Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

Local Dominance Including Control of Dominance Area of Solutions in MOEAs

Hiroyuki Sato Hernán E. Aguirre Kiyoshi Tanaka
Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano, 380-8553 JAPAN
Email: {sato@iplab., ahernan@, ktanaka@}shinshu-u.ac.jp

Abstract— **Local dominance has been shown to improve significantly the overall performance of multiobjective evolutionary algorithms (MOEAs) on combinatorial optimization problems. This work proposes the control of dominance area of solutions in local dominance MOEAs to enhance Pareto selection aiming to find solutions with high convergence and diversity properties. We control the expansion or contraction of the dominance area of solutions and analyze its effects on the search performance of a local dominance MOEA using 0/1 multiobjective knapsack problems. We show that convergence of the algorithm can be significantly improved while keeping a good distribution of solutions along the whole true Pareto front by using local dominance with expansion of dominance area of solutions. We also show that by controlling the dominance area of solutions dominance can be applied within very small neighborhoods, which reduces significantly the computational cost of the local dominance MOEA.**

I. Introduction

Multiobjective evolutionary algorithms (MOEAs) [1], [2] are being increasingly investigated for solving multiobjective optimization problems. Some important features of the latest generation MOEAs are that selection incorporates elitism and it is biased by Pareto dominance and a diversity preserving strategy in objective space. Pareto dominance based selection is thought to be effective for problems with convex and non-convex fronts and has been successfully applied, especially in two and three objectives problems.

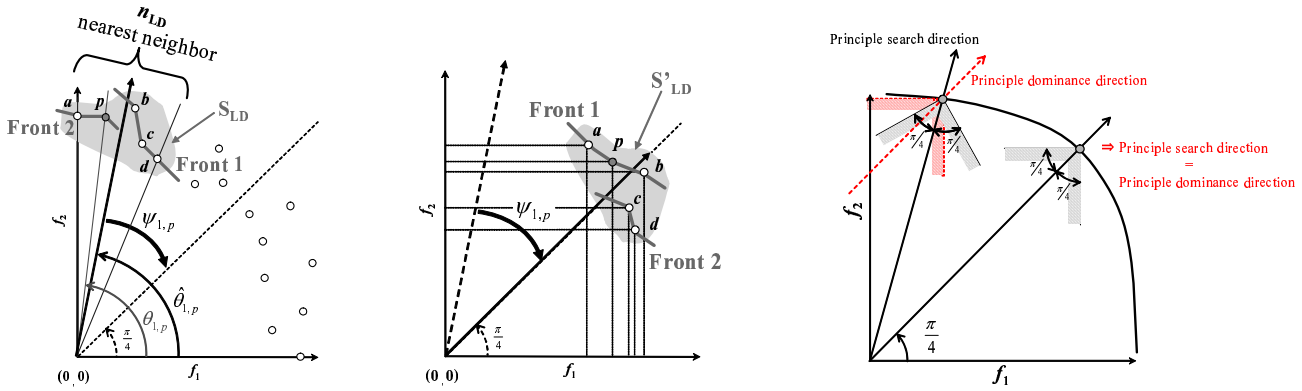
However, some current research reveals that ranking by Pareto dominance on problems with an increased number of objectives might not longer be effective [3], [4], [5]. It has been shown that the characteristics of multi-objective landscapes viewed in terms of non-dominated fronts can change drastically as the number of objectives increases, i.e. the number of fronts reduces substantially and become denser (more solutions per front) just by increasing the number of objectives [5]. In this case, most sampled solutions at a given time turn to be non-dominated. That is, most solutions are assigned the same rank of non-dominance and Pareto selection weakens since it has to discriminate mostly based on diversity of solutions. Another factor that affects the density of the fronts is the complexity of the individual single objective landscapes. It has been shown that the top non-dominated fronts become denser as the complexity of the landscapes reduces, and vice-versa [5], which affects the behavior and effectiveness of conventional Pareto selection. These studies suggest that for selection to be effective a more careful analysis of Pareto dominance relation is required when the number of objectives increases. In addition, for any number of objectives, the dominance relation should be appropriately revised according to the characteristics of the multiobjective landscape.

Aiming to find better ways to perform Pareto selection, in previous works we have proposed the use of local dominance in MOEAs. In [6] we introduced a method that creates a neighborhood around each individual and assigns a local dominance rank after rotating the principal search direction of the neighborhood by using polar coordinates in objective space. Experimental results on combinatorial optimization problems showed that the overall performance of the method in terms of hypervolume is significantly better than the performance of algorithms applying dominance globally as conventional approaches do. Local dominance with alignment of principle search direction induces the algorithm to search for solutions along the whole true Pareto front. However, convergence towards the true Pareto front might slightly deteriorate in some regions. Also, in order to achieve better performance, computational cost of the local dominance algorithm might be greater than a conventional global dominance algorithm.

In order to solve these problems, in this work we propose to include the control of dominance area of solutions [7] in local dominance MOEAs and analyze its effects on search performance. In [7] we showed that the degree of expansion or contraction of the dominance area of solutions can be controlled using a user-defined parameter S. The motivation of the present work is to enhance selection aiming to find solutions with high convergence properties covering the whole true Pareto front on combinatorial optimization problems. We conduct experiments with 0/1 multiple knapsack problems on $m = \{2, 3, 4\}$ objectives showing that convergence of the algorithm can be significantly improved while covering the whole true Pareto front by using local dominance with expansion of dominance area of solutions. We also show that by controlling the dominance area of solutions dominance can be applied within very small neighborhoods, which reduces significantly the computational cost of the algorithm.

II. MOEA using Local Dominance

In this section we explain in detail the local dominance MOEA [6]. The main steps of our method at each generation are as follows:



(a) Before rotation (b) After rotation (c) Alignment

Fig. 1. Neighborhood creation, its rotation, and the obtained fronts after non-dominance sorting in the calculation of Local Dominance (LD), (a) and (b). Alignment of principle dominance direction with principle search direction, (c).

(i) Calculate *Local Dominance* (*LD*) for each individual in the population $\mathcal{P}(t)$ and assign a local non-domination rank.
(ii) Assign a θ-*crowding* factor to each individual.
(iii) Truncate the population $\mathcal{P}(t)$ to obtain the parent population $\mathcal{Q}(t)$ taking into account the local non-dominance ranking and θ-crowding factor.
(iv) Create the offspring population $\mathcal{R}(t)$ from $\mathcal{Q}(t)$ applying *Local Recombination* (*LR*).
(v) Evaluate the offspring population $\mathcal{Q}(t)$.
(vi) Join the parent and offspring population to create the new population $\mathcal{P}(t+1)$ for the next generation.

Note that the method uses local dominance and local recombination with different locality. Below we explain mainly local dominance, the main focus of this work.

A. Local Dominance (LD)

To calculate local dominance of the individuals in the population $\mathcal{P}(t)$ first the fitness vector of each individual is transformed to polar coordinates in the objective function space, i.e. an individual p is expressed by a norm r_p and $m-1$ declination angles $\theta_{1,p}$, $\theta_{2,p}$, $\cdots$, $\theta_{m-1,p}$, where m is the number of objectives. Second, the neighborhood for local dominance S_{LD} of individual p is temporally created as a local sub-population by choosing the n_{LD} closest individuals to p from the entire population $\mathcal{P}(t)$. The closeness $\delta_{p,x}$ between an individual p and another individual x is determined by their declination angles to each axis of objective function as follows[1]

$$\delta_{p,x} = \sum_{i=1}^{m-1} |\theta_{i,p} - \theta_{i,x}| . \quad (1)$$

Note that p is part of its neighborhood since $\delta_{p,p} = 0$. Third, a principle search direction $\{\hat{\theta}_{1,p}, \hat{\theta}_{2,p}, \cdots, \hat{\theta}_{m-1,p}\}$ for the neighborhood S_{LD} is established by calculating the angle difference between extreme individuals in the neighborhood of p. That is,

$$\hat{\theta}_{i,p} = \frac{(\theta^{max}_{i,p} - \theta^{min}_{i,p})}{2} + \theta^{min}_{i,p} \quad (2)$$

where $\theta^{max}_{i,p} = \max\{\theta_{i,x_1}, \theta_{i,x_2}, \cdots, \theta_{i,x_{n_{LD}}}\}$, $\theta^{min}_{i,p} = \min\{\theta_{i,x_1}, \theta_{i,x_2}, \cdots, \theta_{i,x_{n_{LD}}}\}$, and $x_j \in S_{LD}$. Next, the principle search direction is rotated by $\{\hat{\theta}_{1,p} - \pi/4, \hat{\theta}_{2,p} - \pi/4, \cdots, \hat{\theta}_{m-1,p} - \pi/4\}$, so that all its declination angles would be $\pi/4$. Accordingly, all individuals in the local sub-population are rotated by the same rotation angles, so their declination angles would be around $\pi/4$. Finally, a non-domination sorting procedure [8] is applied to the rotated neighborhood S'_{LD}, and p is assigned a rank equal to the non-dominated front it belongs to. **Fig. 1** (a) and (b) illustrate for two objectives the neighborhood creation, its rotation, and the fronts obtained with non-domination sorting before and after rotation. Varying the number of elements in the neighborhood $n_{LD} \leq |\mathcal{P}(t)|$ we can control the degree of locality for dominance. In the extreme, $n_{LD} = |\mathcal{P}(t)|$, we have global dominance as in the case of conventional MOEAs.

B. Motivation and Expected Effect from Local Dominance with Alignment of Principle Search Direction

The motivation to rotate the principle search direction of the neighborhood, before we calculate non-domination rank, is to establish more precisely local non-domination relationships by aligning the direction where the individuals are leading (search direction) with the direction of their area of influence (dominance direction), i.e. local dominance with alignment of principle search direction. The reason to chose $\pi/4$ as the declination angle of the rotated principle search direction of the sub-population is because precisely at that angle the principle search direction is coincident with the principle dominance direction as illustrated in **Fig. 1** (c).

As indicated above, the rotation of the sub-population changes dominance relationships among solutions. This increases the chance of selecting promising solutions rather

[1] **Eq. (1)** can also be calculated by using the inner product and norms of two vectors. However, here we use the declination angles calculated in advance to perform rotation.

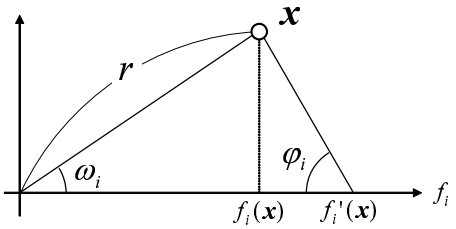


Fig. 2. Fitness modification to change the covered area of dominance

than conventional schemes that apply global dominance. As shown in **Fig. 1** (a), if we calculate dominance with a conventional scheme, say NSGA-II [8], individuals a and p would be assigned a lower rank and dismissed with high probability in the parent selection process since they appear globally dominated by b. On the other hand, if we take into account the principle search direction of S_{LD} and properly rotate declination angles, as shown in **Fig. 1** (b), promising individuals a and p become non-dominated solutions, which would induce a better coverage of under-represented regions in the set of non-dominated solutions found so far. In this example, a and p have the potential to disperse the distribution of solutions to the direction of objective function f_2.

C. Local Recombination (LR) and θ-crowding

In our method offspring are created one at the time. To create one offspring first we specify a random principle search direction $\boldsymbol{\nu} = \{\nu_1, \nu_2, \cdots, \nu_{m-1}\}$, where $0 \leq \nu_i \leq \pi/2$. Second, a neighborhood for local recombination S_{LR} is temporally created as a local sub-population around $\boldsymbol{\nu}$ by choosing the n_{LR} closest individuals to $\boldsymbol{\nu}$ from the parent population $\mathcal{Q}(t)$. The same notion of closeness specified by **Eq. (1)** is used here but using $\boldsymbol{\nu}$ instead of the declination angles of p, $\delta_{\boldsymbol{\nu},x}$. Third, mating is performed within the neighborhood S_{LR} and then recombination followed by mutation are carried out. For mating binary tournament selection is used, but we enforce equal participation in the tournaments. Varying the number of elements in the neighborhood $n_{LR} \leq |\mathcal{Q}(t)|$ we can control the degree of locality for recombination. In the extreme, $n_{LR} = |\mathcal{Q}(t)|$, we have global recombination as in the case of conventional MOEAs.

θ-crowding is used to calculate the crowding factor of each solution in our method [6]. The procedure to calculate θ-crowding is inspired from the crowding distance procedure used by NSGA-II [8], but it is based on declination angles rather than on fitness values. That is, it sorts the population by declination angle θ_j (one at the time, $j = 1, \cdots, m-1$) and estimates the density of the i ordered point by accumulating the angle difference between immediate neighbors $i-1$ and $i+1$.

III. Extension of Pareto dominance

A. Contraction and Expansion of Dominance Area

In this work, we adjust the dominance area of solutions during the process of calculating local dominance. In general, the dominance area is uniquely determined with a fitness vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))$ in the objective space when a solution $\boldsymbol{x}$ is given. To contract and expand the dominance area of solutions, we modify fitness value for each objective function by changing the user defined parameter S_i in the following equation

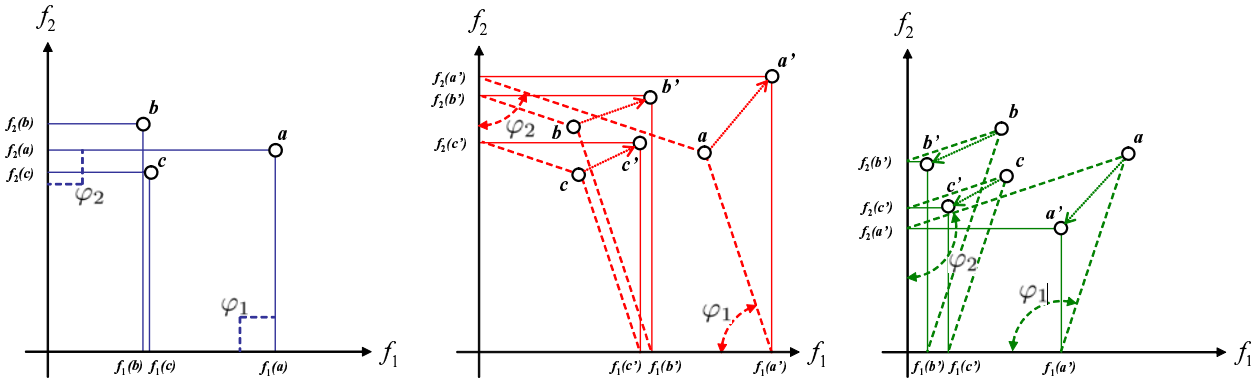
$$f_i'(x) = \frac{r \cdot \sin(\omega_i + S_i \cdot \pi)}{\sin(S_i \cdot \pi)} \qquad (i = 1, 2, \cdots, m) \qquad (3)$$

where $\varphi_i = S_i \cdot \pi$. This equation is derived from the Sine theorem. We illustrate the fitness modification in **Fig. 2**, where r is the norm of $\boldsymbol{f}(\boldsymbol{x})$, $f_i(\boldsymbol{x})$ is the fitness value in the i-th objective, and ω_i is the declination angle between $\boldsymbol{f}(\boldsymbol{x})$ and $f_i(\boldsymbol{x})$. In this example, the i-th fitness value $f_i(\boldsymbol{x})$ is increased to $f_i'(\boldsymbol{x}) > f_i(\boldsymbol{x})$ by using $\varphi_i < \pi/2$ ($S_i < 0.5$). In case of $\varphi_i = \pi/2$ ($S_i = 0.5$), $f_i(\boldsymbol{x})$ does not change and $f_i'(\boldsymbol{x}) = f_i(\boldsymbol{x})$. Thus, this case is equivalent to the conventional dominance. On the other hand, in case of $\varphi_i > \pi/2$ ($S_i > 0.5$), $f_i(\boldsymbol{x})$ is decreased so $f_i'(\boldsymbol{x}) < f_i(\boldsymbol{x})$.

Such fitness modification changes the dominance area of solutions. We show an example in **Fig. 3**, where three solutions $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are distributed in 2-dimensional objective space. In **Fig. 3**(a), $\boldsymbol{a}$ dominates $\boldsymbol{c}$, but $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\boldsymbol{b}$ and $\boldsymbol{c}$ do not dominate each other. However, if we modify fitness values for each solution by using **Eq. (3)**, the location of each solution moves in the objective space, and consequently the dominance relationship among solutions changes. For example, if we use $S_1 = S_2 < 0.5$ as shown in **Fig. 3**(b), the dominance area of solutions $\boldsymbol{a}'$, $\boldsymbol{b}'$ and $\boldsymbol{c}'$ is expanded from the original one of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. This causes that $\boldsymbol{a}'$ dominates $\boldsymbol{b}'$ and $\boldsymbol{c}'$, and $\boldsymbol{b}'$ dominates $\boldsymbol{c}'$. That is, expansion of dominance area by smaller $S_i (< 0.5)$ works to produce a more fine grained ranking of solutions and would strengthen selection especially of solutions with higher projection on the $\pi/4$ direction (middle regions of objective space). On the other hand, if we use $S_1 = S_2 > 0.5$ as shown in **Fig. 3**(c), the dominance area of solutions $\boldsymbol{a}'$, $\boldsymbol{b}'$ and $\boldsymbol{c}'$ is contracted from the original one of $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. This causes that $\boldsymbol{a}'$, $\boldsymbol{b}'$ and $\boldsymbol{c}'$ do not dominate each other. That is, contracting the area of dominance by larger $S_i (> 0.5)$ works to produce a coarse ranking of solutions and would weaken selection by giving high rank to solutions located towards the extreme regions. We refer the reader to other few works on relaxed forms of Pareto dominance, such as ε-dominance [9] and α-domination [10].

B. Effects of Controlling Dominance Area

As indicated above, expanding or contracting the dominance area of solutions change the dominance relation of some solutions and therefore modify the distribution of the fronts (number of fronts and solutions per front). Since front distribution significantly relates to selection, we verify and illustrate the effect of expanding or contracting the dominance area on the distribution of the fronts changing the parameter S_i in **Eq. (3)**. Here, we randomly generate 100 solutions in the 2-dimensional objective space of $[0, 1]^2$, calculate dominance among them after recalculating fitness with **Eq. (3)**, and perform a non-domination sorting to obtain the fronts.



(a) $S_1 = S_2 = 0.5$ (b) $S_1 = S_2 < 0.5$ (c) $S_1 = S_2 > 0.5$

Fig. 3. Conventional dominance and examples of expanding and contracting the dominance area of solutions

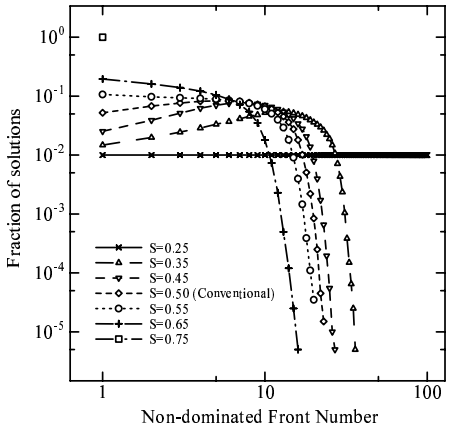


Fig. 4. Solutions per front as we vary the parameter S

We repeat the above steps a 1000 times and calculate the average number of fronts and solutions per front, for each value of S_i. In this work, we use a common parameter $S = S_i (i = 1, 2, \cdots, m)$ for all objective functions, because we assume that all objective functions are normalized. **Figure 4** shows the fraction of number of solutions per front for $S = \{0.25, 0.35, 0.45, 0.5, 0.55, 0.65, 0.75\}$.

From this figure, note that if we gradually expand the area of dominance by decreasing S below 0.5, the number of fronts increases and the ranking of solutions by non-dominance can be fine grained. Note that for maximum expansion of the dominance area $S = 0.25$ there is one single solution per front. On the other hand, if we gradually contract the area of dominance by increasing S above 0.5, the number of fronts decreases and ranking of solutions by non-dominance becomes coarser. Note that for maximum contraction of the dominance area $S = 0.75$ there is only one front that contains all solutions. Since different rankings can be produced, we can expect that the optimum parameter S^* that yields maximum search performance exists for a given kind of problem.

IV. Experimental Results and Discussion

A. Problems, Metrics, and Parameters

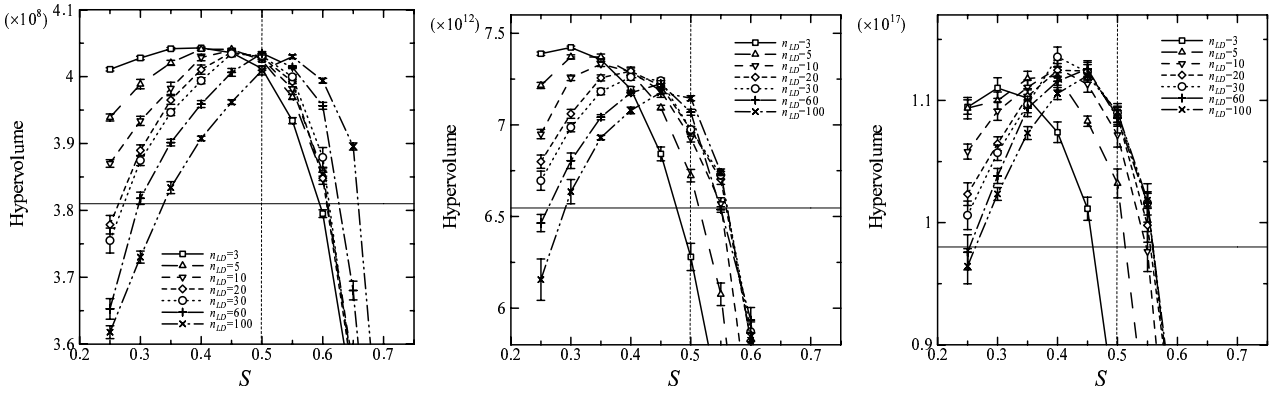
In this paper we use multiobjective 0/1 knapsack problems [11] to study and compare the effects on search performance of controlling dominance area of solutions within the local dominance algorithm [6]. Here we use problems with $m = \{2, 3, 4\}$ objectives, $n = 500$ items and feasibility ratio $\phi = 0.5$ downloaded from [12], for which we know the true Pareto non-dominated set only in case of two objectives $m = 2$.

The hypervolume is used as a metric to evaluate sets of non-dominated solutions obtained by MOEAs. The hypervolume measures the m-dimensional volume of the region in objective space enclosed by the obtained non-dominated solutions and a dominated reference point [13]. Here we use $(f_1, f_2, \cdots, f_m)$=$(0, 0, \cdots, 0)$ as the reference point to calculate the hypervolume. A set of non-dominated solutions showing higher value of hypervolume can be considered as a better set of solutions from both convergence and diversity viewpoints. To provide additional information separately on convergence and diversity of the obtained solutions in this work we also use C metric [14], Generational Distance (GD) [15], Inverse Generational Distance (IGD) [16] and Spread (SP) [1].

In our study we compare the performance of local dominance MOEA [6] without and with control of dominance area of solutions. As a reference, we also include the results by a conventional NSGA-II [8] that uses global dominance. All algorithms adopt two-point crossover with a crossover rate $p_c = 1.0$ for recombination, 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 2,000 generations. Population size is set to $|P| = 200$ on $m = 2$ objectives and $|P| = 600$ on $m = \{3, 4\}$ objectives. The parent and offspring population sizes $|Q|$ and $|R|$ are set to half the population size $|P|$, i.e. $|Q| = |R| = \{100, 300, 300\}$ on $m = \{2, 3, 4\}$, respectively.

B. Results on Hypervolume

Figure 5 (a)-(c) show the hypervolume achieved by the



(a) KP500-2 (b) KP500-3 (c) KP500-4

Fig. 5. Results on Hypervolume

proposed algorithm varying the parameter S at intervals 0.05 in the range $[0.25, 0.75]$. Results are included for dominance neighborhood-size $n_{LD} = \{3, 5, 10, 20, 30, 60, 100\}$ on problems with $m = \{2, 3, 4\}$ objectives. The hypervolume values at $S = 0.5$ marked with a dotted vertical line indicate the performance of the algorithm that includes locality and alignment for dominance, but calculates ranking of solutions using conventional-Pareto dominance. Also, the performance of conventional NSGA-II that uses global Pareto dominance is marked with a horizontal line.

First, looking at conventional Pareto dominance ($S = 0.5$) we can see that the method including locality for dominance can achieve by far better values of hypervolume than NSGA-II. This is in accordance with the previous work [6].

Second, comparing with the results at $S = 0.5$, we can see that local dominance with expansion of dominance area of solutions ($S < 0.5$) can achieve higher values of hypervolume. Note that this effect is more notorious for $m = \{3, 4\}$ objectives than for $m = 2$ objectives. We elaborate on this in **IV-C**.

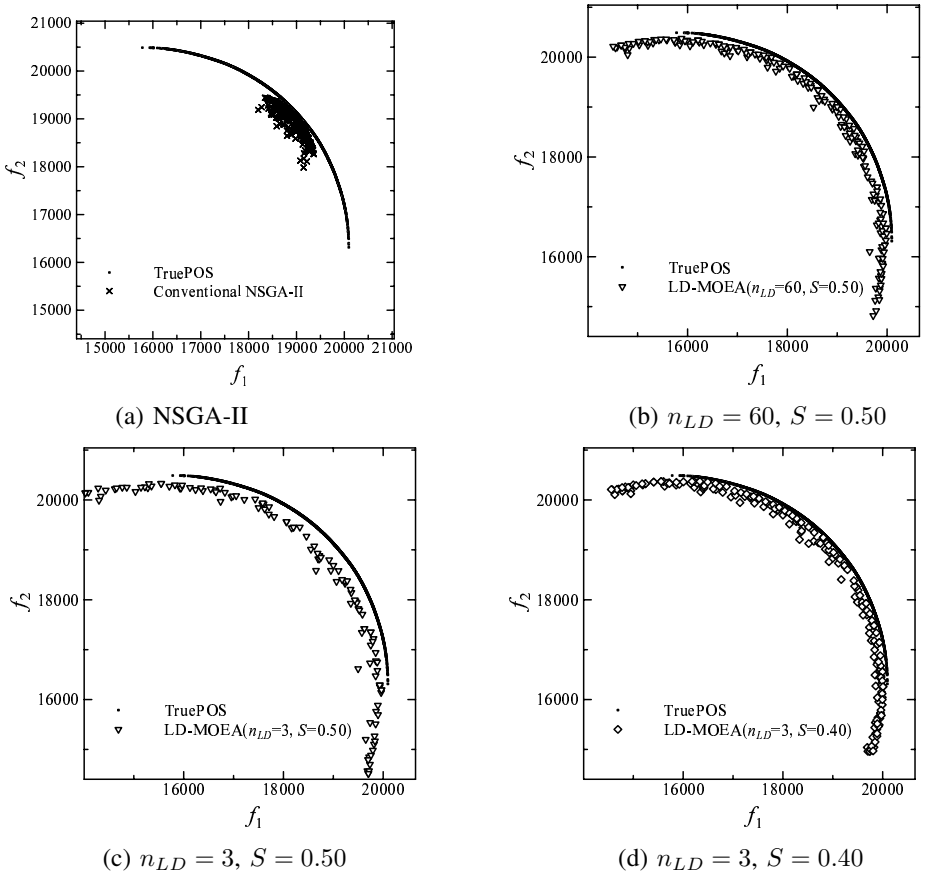
Third, an important observation concerns n_{LD}^{Hmax}, the size of the neighborhood used to calculate dominance that maximizes hypervolume. Note that $n_{LD}^{Hmax} = \{60, 100, 100\}$ by local conventional-Pareto dominance ($S = 0.5$), whereas $n_{LD}^{Hmax} = \{3, 3, 30\}$ by local dominance with expansion of dominance area ($S < 0.5$) for $m = \{2, 3, 4\}$ objectives, respectively. That is, expansion of dominance area of solutions allows significant reductions on the size of the neighborhood. This has an important impact on the reduction of computational cost of the algorithm. See **V** below for details.

Also, it is important to mention that either extreme expansion or extreme contraction of dominance area of solutions deteriorates the performance of the algorithm. We elaborate on this in **IV-D**.

C. Gains on Convergence by Proposed Method

Increases on hypervolume can be due to better diversity and to better convergence. In the following we show that the gains on hypervolume by local dominance with expansion of dominance area of solutions are due to an increase on convergence towards the true Pareto front while keeping diversity along the whole front. **Figure 6 (a)-(d)** show the final population of solutions obtained in a single run by NSGA-II, local conventional-Pareto dominance ($S = 0.5$) for $n_{LD} = 60$ and $n_{LD} = 3$, and local dominance with expanded dominance area of solutions ($S < 0.5$) for $n_{LD} = 3$ on $m = 2$ objectives. From **Figure 6 (a)** note that NSGA-II converges close to the true Pareto front but the solutions found cover only a very narrow region of the true Pareto front. Including local conventional-Pareto dominance ($S = 0.5$) the algorithm is able to find solutions covering the whole true Pareto front, as shown in **Figure 6 (b)** where the algorithm is set to the optimal neighborhood size $n_{LD}^{Hmax} = 60$ that maximizes hypervolume. The algorithm could be set to values of $n_{LD} > 60$ in order to achieve even better convergence, but this requires additional computational time. Also, further reductions on neighborhood size deteriorates convergence of the algorithm, as illustrated in **Figure 6 (c)** where $n_{LD} = 3$. On the other hand, local dominance with expansion of dominance area of solutions ($S < 0.5$) can achieve higher convergence with very small neighborhood sizes, as shown in **Figure 6 (d)** where $S = 0.4$ and $n_{LD}^{Hmax} = 3$. Comparing **Figure 6 (d)** and **(b)** note that local dominance with expansion of dominance area of solutions set to $n_{LD}^{Hmax} = 3$ converges closer to the true Pareto front than local conventional-Pareto dominance set to $n_{LD}^{Hmax} = 60$.

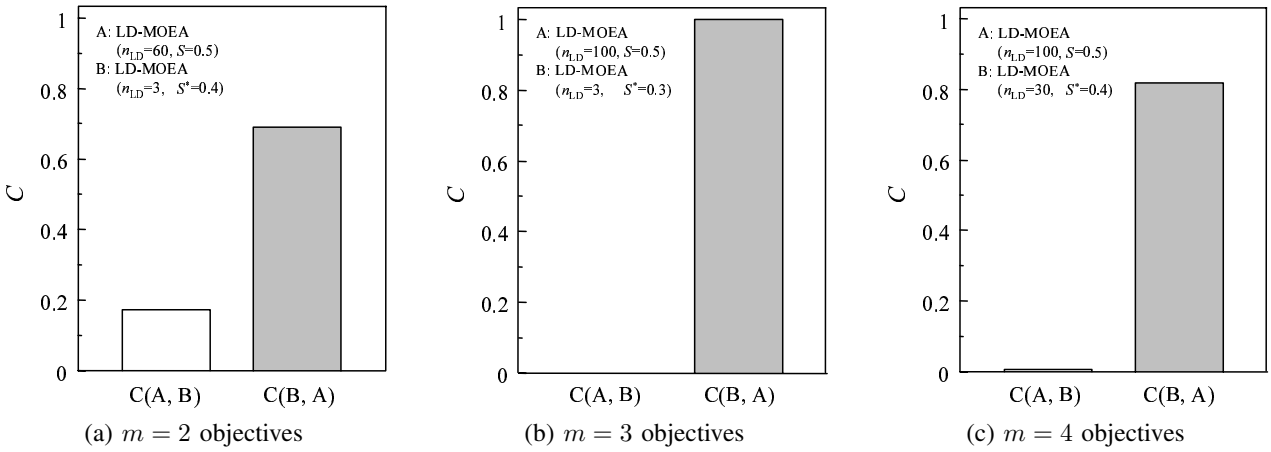
To verify further the gains on convergence, we use the C metric [14] to calculate the fraction of solutions obtained by local dominance without expansion of dominance area of solutions ($S = 0.5$) that are dominated by solutions obtained by local dominance with expansion of dominance area of solutions ($S < 0.5$), and vice versa. **Figure 7 (a)-(c)** show results on C metric obtained with dominance neighborhood sizes that maximize hypervolume n_{LD}^{Hmax} on problems with $m = \{2, 3, 4\}$ objectives, respectively. In the plots we denote the algorithm without expansion as A and the algorithm with expansion as B. From the figure, looking at $C(B, A)$ values, we can see that most solutions obtained without expansion are dominated by solutions obtained with expansion of dominance



(a) NSGA-II (b) $n_{LD} = 60$, $S = 0.50$

(c) $n_{LD} = 3$, $S = 0.50$ (d) $n_{LD} = 3$, $S = 0.40$

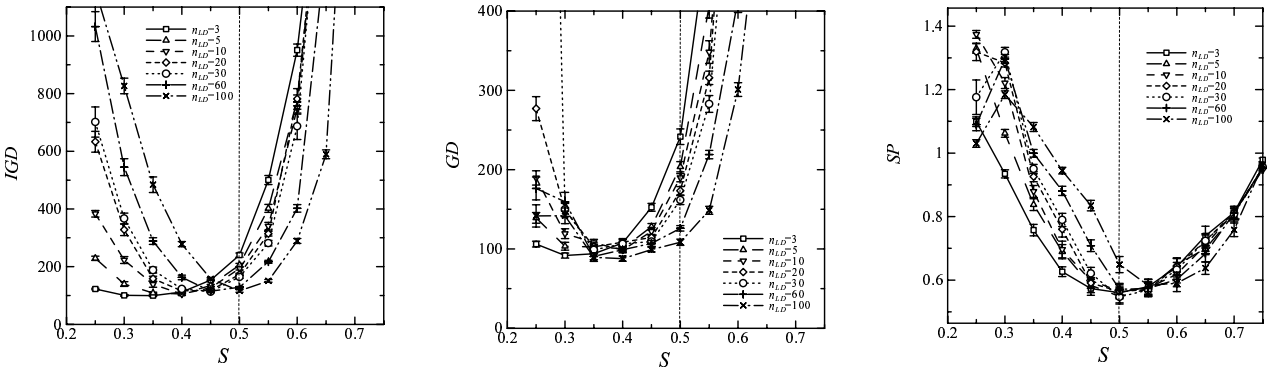
Fig. 6. Final population obtained in a single run



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives

Fig. 7. Results on C metric by local dominance MOEA with and without expansion of dominance area of solutions

area of solutions, i.e. 70%, 100%, and 82% on $m = \{2, 3, 4\}$ objectives, respectively. On the other hand, looking at $C(A, B)$ values, we can see that only a very small fraction of solutions obtained by expansion are dominated by solutions obtained without expansion, i.e. 18%, 0% and 1% on $m = \{2, 3, 4\}$ objectives, respectively. These results are in accordance with the increased difference on hypervolume seen in **Figure 5** especially for 3 and 4 objectives, and confirm that the gain on hypervolume is mostly due to a better convergence by including expansion of area of dominance of solution within the local dominance MOEA.



(a) IGD (b) GD (c) SP

Fig. 8. Results on complementary metrics, KP500-2

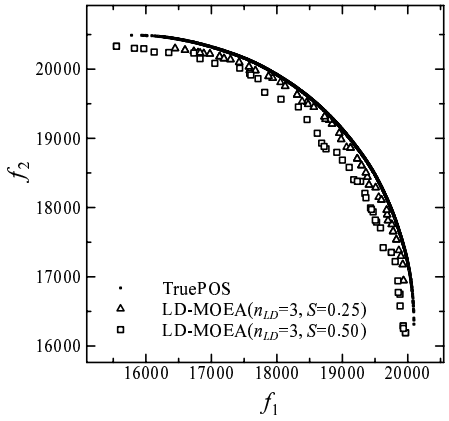


Fig. 9. Obtained non-dominated solutions by local dominance without expansion of dominance area of solutions ($S = 0.5$) and with extreme expansion of dominance area of solutions ($S = 0.25$). Dominance neighborhood $n_{LD} = 3$.

D. Results on Additional Complementary Metrics

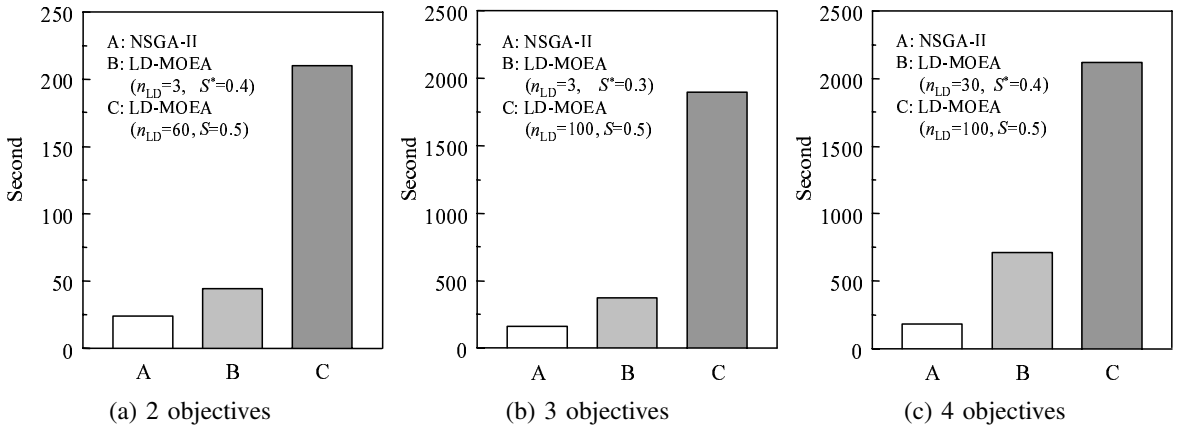
In order to discuss further convergence and diversity, **Figure 8 (a)-(c)** show results on inverse generational distance IGD, generational distance GD, and spread SP, respectively, for $m = 2$ objectives. From **Figure 8(a)** we can see that there is a minimum value for each dominance neighborhood size n_{LD}. Also, note that the minimum moves from $S = 0.5$ for $n_{LD} = 100$ towards $S = 0.3$ for $n_{LD} = 3$. From **Figure 8(b)** note that minimum GD is located in the region $S < 0.5$ for all n_{LD}. That is, convergence towards the true Pareto front can be improved by expansion of dominance area of solutions. We can see this tendency in **Figure 9**, where there is a substantial improvement on convergence of obtained solutions for $n_{LD} = 3$ by local dominance with expansion ($S = 0.25$) compared to local dominance without expansion ($S = 0.5$). On the other hand, from **Figure 8(c)** we can see that a similar best minimum value of SP is achieved at $S = 0.5$ for most dominance neighborhood sizes. Only for $n_{LD} = 100$ the minimum is around $S = 0.55$. Here, it is important to note that local dominance with alignment of search direction does a very good job spreading the population.

Figure 8 (a)-(c) also help explaining the lower values of hypervolume for extreme values of expansion or contraction of dominance area of solutions. For extreme values of contraction of dominance area ($S \approx 0.75$) note that the hypervolume deteriorates mainly due to worsening of convergence towards the true Pareto front as shown by GD on **Figure 8(b)** for $S > 0.5$. On the other hand, for extreme values of expansion of dominance area ($S \approx 0.25$) the hypervolume deteriorates mainly due to worsening of distribution of solution along the true Pareto front as shown by SP on **Figure 8(c)** for $S < 0.5$.

V. Computational Order

In this section we give an estimate of the computational order required to calculate local dominance as it is implemented here. We also present results on CPU time.

To calculate local dominance for each solution, we first create a sub-population with its nearest neighbors, rotate the sub-population to align its search direction, and calculate Pareto dominance within the sub-population, see section **II-A** for details. In the following, N is the population size, m the number of objectives, and n_{LD} the neighborhood size. Thus, the main steps of this process require $(m-1)N$ calculations to compute the distances from one solution to all other solutions, where distance is determined by $m-1$ declination angles, $N \log_2 N$ computations to order the population and determine the neighbors sub-population, mn_{LD} computations to rotate and align the neighbors sub-population, and mn_{LD}^2 calculations to compute local dominance within the sub-population. Thus, the computational order needed to calculate local dominance rank of all individuals is given by $N\{(m-1)N + N\log_2 N + mn_{LD} + mn_{LD}^2\}$. In the proposed method, since we recalculate fitness to include expansion or contraction of dominance area within local dominance calculation, another mNn_{LD} computations are required in addition to local dominance rising the overall number of computations to $N\{(m-1)N + N\log_2 N + 2mn_{LD} + mn_{LD}^2\}$. However, remember the optimal neighborhood sizes $n_{LD} = \{3, 3, 30\}$ for population sizes $N = \{200, 600, 600\}$ on $m = \{2, 3, 4\}$ objectives, respectively. For $m = \{2, 3\}$ objectives clearly $n_{LD}^2 \ll N$, whereas for $m = 4$ objectives $n_{LD}^2 < 2N$.



(a) 2 objectives (b) 3 objectives (c) 4 objectives

Fig. 10. Actual CPU time.

Thus, we can simplify the order of computations reducing to $\mathcal{O}(N^2 log_2 N)$. This is a substantial reduction on computational time compared to the local dominance MOEA proposed in [6], which order is given by $\mathcal{O}(mNn_{LD}^2)$.

Figure 10 shows the actual CPU time by conventional NSGA-II and the local dominance MOEA without and with expansion of dominance area of solutions running the algorithms on a PC (CPU:AthlonXP 2800+, Memory: 1 Gbytes, OS:WindowsXP). From this figure we can see that local dominance MOEA with expansion of dominance area of solutions ($S < 0.5$) substantially reduces CPU time compared to the local dominance MOEA without expansion ($S = 0.5$). Also, note that the measured CPU time is in accordance with the computational order of the algorithms.

VI. Conclusions

In this work we have proposed and analyzed the effects on performance of controlling dominance area of solutions in local dominance MOEAs. We showed that convergence of the algorithm can be significantly improved while keeping a good distribution of solutions along the whole true Pareto front by using local dominance with expansion of dominance area of solutions. We also showed that by controlling the dominance area of solutions dominance can be applied within very small neighborhoods, which reduces significantly the computational cost of the local dominance MOEA. Currently, at each generation we create a neighborhood around each individual in the population to calculate its local dominance ranking. We would like to look into ways to reduce further the number of computations without affecting performance of the algorithm in terms of convergence and diversity. Also, we would like to investigate adaptation to control the dominance area of solutions. Furthermore, we should test the local dominance algorithm on other kind of problems and more number of objectives [17].

References

[1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[2] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, Kluwer Academic Publishers, 2002.

[3] R. C. Purshouse and P. J. Fleming, "Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation", in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (editors), *Proc. Second Intl. Conf. on Evolutionary Multi-Criterion Optimization EMO 2003*, Springer, Lecture Notes in Computer Science, Vol. 2632, pp.16-30, April 2003.

[4] E. J. Hughes, "Evolutionary Many-Objective Optimisation: Many Once or One Many?", *IEEE Congress on Evolutionary Computation CEC'2005*, pp.222-227, Vol.1, IEEE Service Center, Edinburgh, Scotland, September 2005.

[5] H. Aguirre and K. Tanaka, "Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes", *European Journal of Operational Research, Special Issue on Evolutionary Multi-Objective Optimization*, in press.

[6] H. Sato, H. Aguirre and K. Tanaka, "Local Dominance and Local Recombination in MOEAs on 0/1 Multiple Knapsack Problems", *European Journal of Operational Research, Special Issue on Evolutionary Multi-Objective Optimization*, in press.

[7] H. Sato, H. Aguirre and K. Tanaka, 'Controlling Dominance Area of Solutions and its Impact on the Performance of MOEAs", *Proc. Fourth Intl. Conf. on Evolutionary Multi-Criterion Optimization EMO 2007*, Springer, Lecture Notes in Computer Science, to appear, March 2007.

[8] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", *KanGAL report 200001*, 2000.

[9] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization", *Evolutionary Computation*, Vol.10, No.3, pp.263-282, Fall 2002.

[10] K. Ikeda, H. Kita and S. Kobayashi, "Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal?", *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, Vol. 2, IEEE Service Center, Piscataway, New Jersey, pp.957-962, May 2001.

[11] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – a comparative case study", *Proc. of 5th Intl. Conf. on Parallel Problem Solving from Nature (PPSN-V)*, pp.292-301, 1998.

[12] http://www.tik.ee.ethz.ch/~zitzler/testdata.html

[13] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1999.

[14] J. Knowles and D. Corne, "On Metrics for Comparing Non-dominated Sets", *Proc. 2002 IEEE Congress on Evolutionary Computation*, pp.711–716, IEEE Service Center, 2002.

[15] D. A. V. Veldhuizen and G. B. Lamont, "On Measuring Multiobjective Evolutionary Algorithm Performance", *2000 Congress on Evolutionary Computation*, vol.1, pp.204–211, 2000.

[16] H. Sato, H. Aguirre, and K. Tanaka, "Local Dominance Using Polar Coordinates to Enhance Multi-objective Evolutionary Algorithms", *Proc. 2004 IEEE Congress on Evolutionary Computation*, IEEE Service Center, vol. 1, pp. 188-195, 2004.

[17] V. Khare, X. Yao, and K. Deb, "Performance Scaling of Multiobjective Evolutionary Algorithms", *Proc. 2nd Intl. Conf. on Evolutionary Multi-Criterion Optimization EMO 2003*, Springer, Lecture Notes in Computer Science, vol2632, pp.376-390, April 2003.