HAL open science

Estimating Relevance of Variables for Effective Recombination

Taishi Ito, Hernan Aguirre, Kiyoshi Tanaka, Arnaud Liefooghe, Bilel Derbel, Sébastien Verel

▶ **To cite this version:**

Taishi Ito, Hernan Aguirre, Kiyoshi Tanaka, Arnaud Liefooghe, Bilel Derbel, et al.. Estimating Relevance of Variables for Effective Recombination. EMO 2019 - International Conference on Evolutionary Multi-Criterion Optimization, Mar 2019, East Lansing, Michigan, United States. pp.411-423, 10.1007/978-3-030-12598-1_33 . hal-02064547

HAL Id: hal-02064547

https://hal.science/hal-02064547v1

Submitted on 13 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimating Relevance of Variables for Effective Recombination

Taishi Ito[1,4], Hernán Aguirre[1,4], Kiyoshi Tanaka[1,4],
Arnaud Liefooghe[2,4], Bilel Derbel[2,4], and Sébastien Verel[3,4]

[1] Shinshu University, Faculty of Engineering, Japan
{17w2006d, ahernan, ktanaka}@shinshu-u.ac.jp
[2] Université Lille 1, CRIStAL – Inria Lille-Nord Europe, France
{arnaud.liefooghe,bilel.derbel}@univ-lille.fr
[3] Université du Littoral Cote d'Opale, LISIC, France
verel@lisic.univ-littoral.fr
[4] International Associated Laboratory LIA-MODO

Abstract. Dominance, extensions of dominance, decomposition, and indicator functions are well-known approaches used to design MOEAs. Algorithms based on these approaches have mostly sought to enhance parent selection and survival selection. In addition, several variation operators have been developed for MOEAs. We focus on the classification and selection of variables to improve the effectiveness of solution search. In this work, we propose a method to classify variables that influence convergence and increase their recombination rate, aiming to improve convergence of the approximation found by the algorithm. We incorporate the proposed method into NSGA-II and study its effectiveness using three-objective DTLZ and WFG functions, including unimodal, multimodal, separable, non-separable, unbiased, and biased functions. We also test the effectiveness of the proposed method on a real-world bi-objective problem. Simulation results verify that the proposed method can contribute to achieving faster and better convergence in several kinds of problems, including the real-world problem.

Keywords: Evolutionary multi-objective optimization · Variables classification · Variables selection · Recombination operators.

1 Introduction

Multi-objective evolutionary algorithms[4, 2] (MOEAs) have been used to solve multi-objective optimization problems on all kinds of application domains. Due to their success, MOEAs are being applied to real-world problems of increased complexity. Scalability in decision and objective spaces, epistasis, effectiveness on problems with difficult topologies of the Pareto optimal set, and a limited budget of evaluations due to computationally expensive fitness functions are some of the challenges the new generation of MOEAs have to face.

The enhancement of MOEAs performance is an active research subject. Dominance, extensions of dominance, decomposition, and indicator functions are well-known approaches used to design MOEAs[5], [10], [9], [11], [1]. Algorithms based

on these approaches have mostly sought to enhance parent selection and survival selection. In addition, several variation operators have been developed and incorporated within MOEAs.

Evolutionary multi-objective algorithms commonly select individuals for variation based on their fitness. However, the operators of variation are commonly applied to variables randomly chosen. Typically, an operator rate per variable controls the expected number of variables that will be subject to variation, but the decision of what variables will be modified is left to chance. Modifying a variable of a solution in a multi-objective problem can have one of the following effects. The modification improves one or several objectives without worsening others. This would be the case if the solution subject to the modification is sub-optimal, which is commonly observed in random initial populations and during early stages of the optimization. The modification improves one or more objectives but worsens others. This will typically be observed if the solution being modified is Pareto optimal or if it belongs to a local front. Multi-objective evolutionary algorithms aim to find an approximation of the Pareto optimal set, commonly with good qualities in terms of convergence and diversity in objective space. If the effects a variable has on convergence and diversity can be learned or estimated during the optimization[8], the effectiveness of the search could be enhanced by targeting particular variables for variation to find better approximations of the Pareto optimal set.

We focus on the classification and selection of variables for variation aiming to improve the ability of solution search. In this work, we propose a method to identify variables that influence convergence and increase their selection probabilities, so that recombination can select them more frequently to improve convergence of the approximation found by the algorithm. The proposed method selects randomly a solution from the instantaneous Pareto set and creates variations of it mutating one variable at the time. Variables are classified into `influential` and `uninfluential` based on whether there is a dominance relation or not between the original solution and the corresponding one-variable mutants. The method estimates that `influential` variables affect convergence of solutions in objective space and increase their recombination rate.

In this paper, we incorporate the proposed method into NSGA-II[5] and study its effectiveness using three-objective DTLZ[3] and WFG[6] functions, including unimodal, multimodal, separable, non-separable, unbiased, and biased functions. We also verify the effectiveness of the proposed method on a real-world bi-objective problem [7]. Three ways to determine the trial values of variables to create the mutants are investigated. Simulation results verify that the proposed method can contribute to achieving faster and better convergence in several kinds of problems.

2 Method

In this section, we describe the proposed variable selection method, applied every generation after front sorting before truncation selection. The method first clas-

```
procedure VariableClassification(F_1)
    L = (L_1, L_2, ··· , L_n) = (0, 0, ··· , 0)
    x^trial = (x_1^trial, x_2^trial, ··· , x_n^trial) = ReferenceValues(F_1)
    x = x^j ∈ F_1, j = rand(1, |F_1|)
    for i = 1 to n do
        y = (y_1, y_2, ··· , y_n) = x = (x_1, x_2, ··· , x_n)
        y_i = x_i^trial
        Evaluate(y)
        if x ≻ y OR y ≻ x then
            L_i = 1
        end if
    end for
    return L
end procedure
```

Fig. 1. Classification of variables

sifies variables that can influence convergence and then update their probabilities so that recombination can select them more frequently.

The procedure to classify variables is illustrated in the pseudocode of **Figure 1**. First, we assign a label 0 to each variable, $L_i = 0, i = 1, \cdots, n$. Next, we obtain trial values for each variable $\boldsymbol{x}^{\mathbf{trial}} = (x_1^{\text{trial}}, \cdots, x_n^{\text{trial}})$ from the non-dominated solution set $\boldsymbol{F}_1$ in the population P_t at generation t. We randomly select one solution $\boldsymbol{x} = (x_1, \cdots, x_n)$ from $\boldsymbol{F}_1$. Then, for each variable i, we generate a solution $\boldsymbol{y}$ duplicate of $\boldsymbol{x}$ and modify the i-th variable with its corresponding trial value, $y_i = x_i^{trial}$. Evaluate $\boldsymbol{y}$ and calculate the dominance relation between solutions $\boldsymbol{y}$ and $\boldsymbol{x}$. If either $\boldsymbol{y}$ dominates $\boldsymbol{x}$ or $\boldsymbol{x}$ dominates $\boldsymbol{y}$, we update the corresponding label to 1, $L_i = 1$. This procedure returns the vector of labels $\boldsymbol{L}$ assigned to the variables, where $L_i = 1$ if the change in the i-th variable induced a dominance relation ($\succ$) between the randomly sampled solution and its one-variable mutant. $L_i = 0$ otherwise.

We explore three procedures called `random`, `far`, and `near` to set trial values for variables.

`random` sets x_i^{trial} to the value of the i-th variable of a solution randomly selected from $\boldsymbol{F}_1$. A different solution j is randomly chosen for each variable i,

$$x_i^{trial} = z_i^j \mid \boldsymbol{z}^{\boldsymbol{j}} \in \boldsymbol{F}_1 \wedge j = rand(1, |\boldsymbol{F}_1|). \tag{1}$$

`far` sets x_i^{trial} to the farthest value of x_i in $\boldsymbol{F}_1$. That is,

$$\begin{aligned} k &= \arg\max_{j=1\ldots|\boldsymbol{F}_1|}(|x_i - z_i^j|), \qquad \boldsymbol{z}^j \in \boldsymbol{F}_1, x_i \neq z_i^j \\ x_i^{trial} &= z_i^k. \end{aligned} \tag{2}$$

On the other hand, `near` sets x_i^{trial} to the nearest value of x_i in $\boldsymbol{F}_1$. That is,

$$\begin{aligned} k &= \arg\min_{j=1\ldots|\boldsymbol{F}_1|}(|x_i - z_i^j|), \qquad \boldsymbol{z}^j \in \boldsymbol{F}_1, x_i \neq z_i^j \\ x_i^{trial} &= z_i^k. \end{aligned} \tag{3}$$

In `near` and `far`, if all solutions in $\boldsymbol{F}_1$ have the same value in x_i, then $x_i^{trial} = x_i$ (no change).

After variables have been classified, a recombination rate $p'_{cv,i}$ for each variable i is computed as follows

$$p'_{cv,i} = p_{cv}, \qquad if \quad N_L = 0. \tag{4}$$

$$p'_{cv,i} = \begin{cases} 1, & if \quad L_i = 1 \\ \frac{E-N_L}{n-N_L}, & otherwise, \end{cases}, \qquad if \quad 0 < N_L \leq E. \tag{5}$$

$$p'_{cv,i} = \begin{cases} \frac{E}{N_L}, & if \quad L_i = 1 \\ 0, & otherwise. \end{cases}, \qquad if \quad N_L > E. \tag{6}$$

where $L_i \in \{0, 1\}$ is the label assigned to variables, $N_L = \sum_{i=1}^{n} L_i$ is the number of variables classified with label $L_i = 1$, $E = n \times p_{cv}$ is the expected number of crossed variables using the default rate. If after exploring n one-variable mutants no changes in dominance are observed, $N_L = 0$, the recombination rate $p'_{cv,i}$ is set to the default rate p_{cv} for each variable. If there are changes in dominance and these are less than the expected number of crossed variables when the default crossover rate is used, $0 < N_L \leq E$, then the recombination rate $p'_{cv,i}$ is set to 1 for variables labeled $L_i = 1$ and to $\frac{E-N_L}{n-N_L} < p_{cv}$ for variables labeled $L_i = 0$. Otherwise, if $N_L > E$, $p'_{cv,i}$ is set to $\frac{E}{N_L} \geq p_{cv}$ for variables labeled $L_i = 1$ and to 0 for variables labeled $L_i = 0$. Note that the expected number of recombined variables with p_{cv} and p'_{cv} is the same.

In this work, we use the variable classification procedure to update recombination probabilities of variables. However, this method can be easily extended to influence probabilities for mutation or other variation operators.

3 Test Problems

DTLZ [3] and WFG [6] benchmark multi-objective optimization problems are used to evaluate the proposed method. These problems are scalable in the number of variables and the number of objective functions. From the DTLZ family of problems we use DTLZ2 and DTLZ3, whereas from the WFG family we use WFG1-WFG9. Some properties of these problems are summarized in **Table 1**. We include separability, modality and bias. Separable problems are marked with S and non-separable with NS. Unimodal problems are marked U and multimodal problems with M.

We also test the effectiveness of the proposed method on a bi-objective real-world problem [7]. The problem is to design a platform with a motor mounted on it. The machine setup is simplified as a pin-pin supported beam carrying a weight (motor). A vibratory disturbance is imparted from the motor onto the beam, which is of length L, width b, and symmetrical about its mid-plane. The beam is made of three layers of material. Variables d_1 and d_2, respectively, locate the contact of materials 1 and 2, and 2 and 3. Variable d_3 locates the top of the beam. The values of d_1, d_2, and d_3 are measured from the mid-plane

Table 1. Features of Test Problems. Separability: separable S, non-separable NS. Modality: unimodal U, multimodal M.

Problem	Sep.	Modality	Bias	Other Features
DTLZ2	S	U	-	
DTLZ3	S	M	-	
WFG1	S	U	Yes	Polynomial bias $\alpha = 0.02$. Bias variables towards 1
WFG2	NS	U,M	-	U:$f_1, \cdots, f_{M-1}$, M:f_M, f_M. Discontinuous front
WFG3	NS	U	-	Degenerancy constants $A_1 = 1, A_{2:M-1} = 0$,
	.			front reduces to two dimensions
WFG4	S	M	-	
WFG5	S	M	-	
WFG6	NS	U	-	
WFG7	S	U	Yes	Parameter dependent bias: $z_{i=1:k} \leftarrow z_{i+1}, \ldots, z_n$
WFG8	NS	U	Yes	Parameter dependent bias: $z_{i=k+1:n} \leftarrow z_1, \ldots, z_{i-1}$
WFG9	NS	M	Yes	Parameter dependent bias: $z_{i=1:n-1} \leftarrow z_{i+1}, \ldots, z_n$

of the beam. M_i refers to the material type for layer i ($i = 1, 2, 3$). The mass density (ρ), Young's modulus of elasticity (E), and cost per unit volume (c) for each material type is given [7]. The objective functions are the fundamental frequency, f_1, to be maximized and the cost of the set up, f_2, to be minimized. The complete formulation is as follows,

$$\begin{aligned} f_1(d_1, d_2, d_3, b, L) &= (\pi/2L^2)(EI/\mu)^{1/2}, \\ EI &= (2b/3)[E_{M_1} d_1^3 + E_{M_2}(d_2^3 - d_1^3) + E_{M_3}(d_3^3 - d_2^3)], \\ \mu &= 2b[\rho_{M_1} d_1 + \rho_{M_2}(d_2 - d_1) + \rho_{M_3}(d_3 - d_2)], \end{aligned} \tag{7}$$

$$f_2(d_1, d_2, d_3, b) = 2b[c_{M_1} d_1 + c_{M_2}(d_2 - d_1) + c_{M_3}(d_3 - d_2)], \tag{8}$$

subject to $\mu L - 2800 < 0$, $d_2 - d_1 \leq 0.01$, $d_3 - d_2 \leq 0.01$, $0.35 \leq b \leq 0.5$, $3 \leq L \leq 6$, and $d_1, d_2, d_3 \in [0.01, 0.6]$.

4 Experimental Setup and Performance Measures

In this paper, we use DTLZ2, DTLZ3, WFG1-WFG9 as benchmark problems, setting the number of objectives $M = 3$ and the number of variables $n = 12$ for all problems.The number of position variables is $M - 1$ in DTLZ. Similarly, we set the number of position variables to $k = M - 1$ in WFG. Thus, the number of position variables is 2 and the number of distance variables is 10 in all benchmark problems dealt with in this paper.

We use five algorithms to study and verify the performance of the variable selection method proposed in this work. The base algorithm is conventional NSGA-II [5]. NSGA-II randomly selects variables for recombination with the same probability p_{cv} per variable. In the following, NSGA-II is named `org`. We also use three variations of NSGA-II using the proposed variable selection

method with one of the procedures to set trial variables, as explained in section 2. These variations are named `random`, `far`, and `near`. The fifth algorithm, named `ideal`, knows in advance the correct classification of distance and position variables to compute the recombination rate per variable. Thus, `org` provides a baseline for performance comparison, whereas `ideal` provides the performance reference of an algorithm with a perfect classification of variables.

The algorithms are run for 2000 generations setting the number of individuals to 100, the recombination rate per individual to 1.0, and the mutation rate to $1/n$. The recombination rate per variable is set to $p_{cv} = 0.5$ in `org` and to $\boldsymbol{p'_{cv}}$ in `random`, `far`, `near`, and `ideal`, computed with Eq. (4-6) as explained in section 2. The number of runs is 30 in all experiments.

To verify improvements on convergence we use generational distance GD in the case of DTLZ problems and the value of variable x_M in WFG problems. GD is computed analitycally. The smaller the value of GD, the better the convergence of the set of obtained nondominated solutions. In WFG distance variables are finally aggregated into variable $x_M \in [0.0, 1.0]$. The smaller the value of x_M is, the closer to the true optimum the solution is. When $x_M = 0$, the obtained solution is the true optimal solution. We also use the hypervolume metric to evaluate performance in the real world problem.

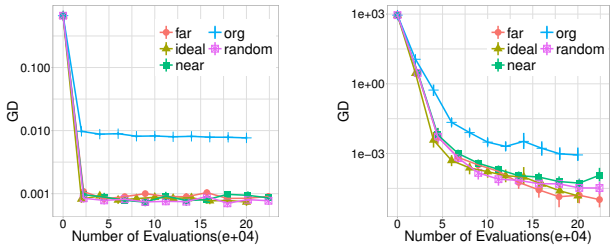
5 Simulation Results and Discussion

5.1 Three-objective Benchmark Problems

First, we apply each algorithm to DTLZ2 and DTLZ3 functions. **Figure 2 (a)** shows the transition of the average GD value in 30 runs over the number of fitness evaluations for DTLZ2. Similarly, **Figure 2 (b)** shows results for DTLZ3. In this experiment, population size is set to 100 and the number of generations to 2,000. Therefore, the total number of evaluations in `org` and `ideal` is 200,000 per run. In the case of `random`, `far`, and `near` the algorithms evaluate an additional trial solution per variable. Since the number of variables is 12, `random`, `far`, and `near` evaluate 112 solutions in each generation and 224,000 evaluations per run.

Comparing GD by the algorithms in **Figure 2 (a)** and **(b)**, it can be seen that `random`, `far`, and `near` approach `ideal` and obtain significantly smaller GD than `org` in both separable problems, the unimodal DTLZ2 and in the multimodal DTLZ3. In DTLZ2, no significant difference is observed among `random`, `far`, and `near`. In the multimodal DTLZ3, during the latest stages of the search, `far` seems to perform better than `random` and `near`, in that order. Note that there is one order of magnitude difference between `far` and `org` in DTLZ2 and two orders of magnitude difference in DTLZ3.

Next, we apply each algorithm to WFG1-WFG9 problems. **Figure 3** shows the transition of the median value of the distance variable x_M in the set of nondominated solutions over the number of evaluations. Comparing the algorithms, it can be seen that `random`, `far`, and `near` approach `ideal` and achieve smaller



(a) DTLZ2 (b) DTLZ3

Fig. 2. Transition of the median value of GD over the number of fitness evaluations in DTLZ2 and DTLZ3 problems
m

x_M values than `org` in WFG1, WFG2, WFG4, WFG5, WFG6, and WFG8, improving significantly convergence. In WFG3, WFG7, and WFG9 x_M value is similar for all algorithms.

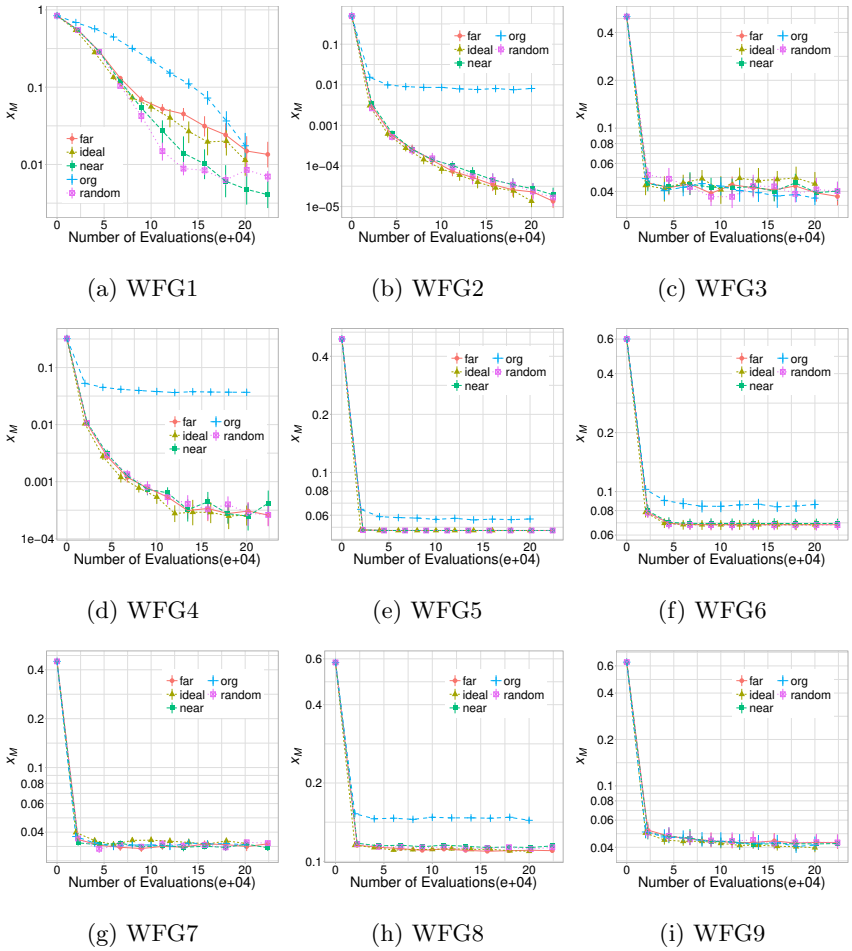
Table 2. Rate of absolute classification of all variables.

	DTLZ2	DTLZ3	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
`far`	99.6	76.6	38.0	99.9	34.6	99.7	99.9	100	4.06	98.1	0.805
`near`	71.6	43.1	14.9	92.3	34.6	98.3	69.5	99.2	6.41	91.3	0.485
`random`	86.9	53.6	13.3	86.7	30.8	89.6	83.5	89.6	2.78	86.7	0.240

Table 2 shows the average rate of absolute classification of variables. A classification is counted as absolute when all position and distance variables are correctly classified. From Table 2, it can be seen that the rates of absolute classification are high in problems DTLZ2, WFG2, WFG4-6, and WFG8, where convergence improves in the algorithms with the proposed method as shown in **Figure 2** and **Figure 3**. Absolute classification rate is low in problems WFG3, WFG7, and WFG9, were convergence did not improve. Note that absolute classification is low in DTLZ3 and WFG1, although the proposed method significantly improves convergence.

To analyze with more detail the classification of variables and its impact on performance, **Figure 4** shows the classification rate per variables over the generations for some problems. For each variable, we compute the percentage it was assigned label 1 in every 100 generations.

In all DTLZ and WFG problems, the classification rates of position variables x_1, x_2 and z_1, z_2 are 0 in nearly all generations. In WFG8, the classification rate of distance variables $x_3, \ldots, x_{12}$ and $z_3, \ldots, z_{12}$ are 1 in almost all generations. A similar situation occurs in problems DTLZ2, WFG2, and WFG4-6 (not shown here). As shown in **Figure 2** and **Figure 3**, the convergence performance im-



(a) WFG1 (b) WFG2 (c) WFG3

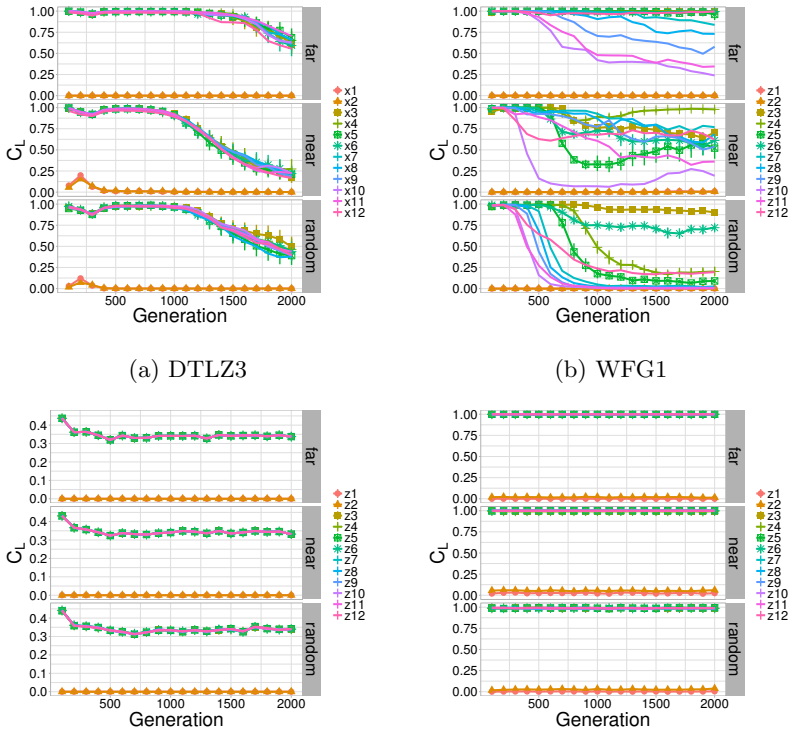
(d) WFG4 (e) WFG5 (f) WFG6

(g) WFG7 (h) WFG8 (i) WFG9

Fig. 3. Transition of the median value of the distance variable x_M over the number of fitness evaluations in WFG problems

proved in the proposed methods compared with the conventional NSGA-II in problems where classification rates of the distance variable are high. In these problems, it is considered that the convergence performance has improved since the distance variables are estimated correctly and they are searched intensively.

Looking at DTLZ3 and WFG1 in **Figure 4**, distance variables were correctly classified in the early generations, but their classification rate decreases in later generations. This is because, in these problems, the value of distance variable tends to converge to the same value, i.e. solutions are trapped in local optima. When this happens, the trial value of the variable in the mutant is the same as the value of the variable in the original solution and therefore a dominance relation does to occur between them. From **Figure 2** and **Figure 3**, DTLZ3



(a) DTLZ3

(b) WFG1

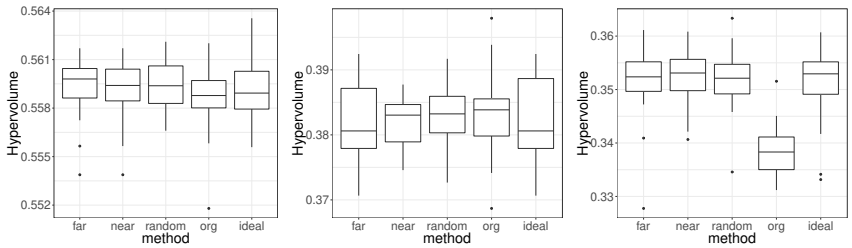
(c) WFG3

(d) WFG8

Fig. 4. Classification rate per variables over the generations

and WFG1, in which distance variables were estimated correctly in the early generations, distance variable converged to values close to optimal in all three proposed methods, so the convergence performance of the solution improved compared with the conventional method. Note that in WFG1, with the passing of generations, the rate of correct classification of distance variables is gradually higher from z_3 to z_{12}. This is because the influence of the distance variables of WFG1 on the convergence of the solution increases from z_3 to z_{12}.

Note from **Figure 4** that the classification rate of the distance variables in WFG3 was low throughout the generations. Something similar happens in problems WFG7 and WFG9. In these problems, distance variables affect position variables as indicated in the Other Features column of **Table 1**. In WFG3, when the value of a distance variable changes, the value of the position variable changes as well due to degeneracy after the transition. In the case of WFG7 ($z_{i=1:k} \leftarrow z_{i+1}, \ldots, z_n$) and WFG9 ($z_{i=1:n-1} \leftarrow z_{i+1}, \ldots, z_n$), a transformation is applied whereby the value of the distance variable is used to bias the value of position variables. As shown in **Figure 3**, the convergence performance of the proposed method on these problems did not differ from the conventional method



(a) WFG3 (b) WFG5 (c) WFG8

Fig. 5. Hypervolume

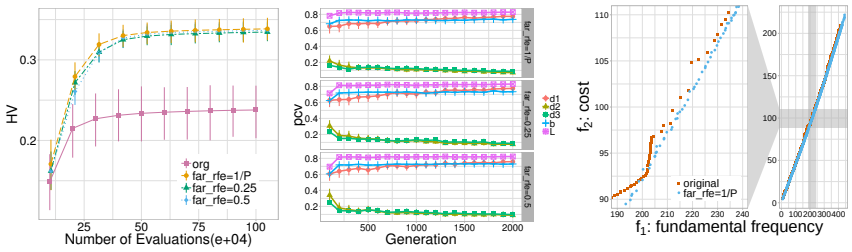
and `ideal`. In these problems, finally note that the classification rate of distance variable was higher in near than random and far. This is probably because the change of the value of the distance variable was small and the values of position variables did not change greatly.

When distance variables also affect position variables, a dominance relation between a solution and its one-variable mutant is less likely to occur and therefore becomes more difficult to classify distance variables in these problems. Note that linkage can be very large in these problems, where some variables are affected by almost all other variables, or some variables affect most other variables. It is worth studying with more detail the effects of linkage on variables identification.

On the other hand, note that variables are correctly classified when position variables affect distance variables ($z_{i=k+1:n} \leftarrow z_1, \ldots, z_{i-1}$), and convergence can be improved as shown by the results on WFG8.

Modality and non-separability of a function seem not to affect the correct classification of variables. Note that there is a high classification rate for uni-modal and multi-modal problems, separable and non-separable, when distance variables do not affect the position of solutions. Also, linkage between distance variables and linkage from position to distance variables are not an issue for correct classification.

In this work, the proposed method favors recombination of variables that can improve convergence. However, in addition to convergence, the aim of a multi-objective optimizer is to also achieve a set of well-distributed solutions. To verify whether the proposed method has a negative impact on diversity we also compute the hypervolume of the solutions found by the algorithms, which measures both convergence and diversity. **Figure 5** shows the hypervolume for WFG3, WFG5, and WFG8. In general, where there is a clear improvement in convergence we also observe an improvement in hypervolume, as illustrated in **Figure 3** (h) and **Figure 5** (c) for WFG8. In cases where there is no difference in convergence or is very small, hypervolume is also similar, as shown in **Figure 3** (c) and **Figure 5** (a) for WFG3 and **Figure 3** (e) and **Figure 5** (b) for WFG5. These results suggest that there is not a serious detriment to diversity of solutions. However, emphasizing variation of variables that improve diversity



(a) Hypervolume (b) Pcv (c) Pareto front

Fig. 6. Results on the vibrating beam problem

would enhance further the performance of the multi-objective algorithm. In the future, we would like to extend the method to focus on diversity as well.

Summarizing, the convergence performance of the proposed method on benchmark problems with `random`, `far`, or `near` procedures to set the trial variables improves compared to `org`, because distance variables are correctly estimated and the frequency of recombining them is increased.

5.2 Vibrating Beam Problem

This bi-objective problem consists of 5 variables and 3 constraints. A random initialization of the population in this problem leads to all solutions being unfeasible. Approximately, 3 in every 10.000 randomly created solutions is feasible. However, a simple constraint handling method allows the algorithm fo find feasible solutions after 5 or 6 generations (500-600 function evaluations) and accumulate them thereafter. Around generation 20 all solutions in the population are feasible. **Figure 6** shows results by NSGA-II with the constraint handling technique, denoted `org`, and by the proposed method `far` applied after 1 solution is feasible, 25% and 50% of the population is feasible. Note from **Figure 6 (a)** that the hypervolume by the proposed method is significantly higher. Also, note from **Figure 6 (b)** that variables d_1, b and L are consistently classified as influential for convergence and their probability p_{cv} for recombination is higher. From **Figure 6 (c)** note that the proposed method improves convergence towards the Pareto front.

6 Conclusions

In this work we have proposed a method to classify variables that influence convergence and increased their selection probabilities to recombine them more often. The classification procedure is based on whether there is a Pareto dominance relation between one-variable mutants. The proposed method was tested

on DTLZ and WFG functions, including unimodal, multimodal, separable, non-separable, unbiased and biased functions. Our experimental results show that the proposed method can improve significantly the performance of the well known NSGA-II algorithm in most instances for 3 objective functions. Modality and non-separability of a function seem not to affect the correct classification of variables. Also, linkage between distance variables and linkage from position to distance variables are not an issue for correct classification. However, performance could not be improved in problems where linkage from distance to position variables is very large. We also verified that the proposed method can improve convergence without affecting diversity on a real world bi-objective problem. In the future we would like to study with more detail the effect of linkage. Also, we would like to study the scalability in objective space, particularly since we use a dominance relation to classify variables. In addition, we would like to study ways to extend the proposed method for large scale problems.

References

1. Aguirre, H., Oyama, A., Tanaka, K.: Adaptive ε-sampling and ε-hood for evolutionary many-objective optimization. In: Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, vol. 7811, pp. 322–336 (2013)
2. Coello, C.C., Lamont, G., van Veldhuizen, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation, Springer, Berlin, Heidelberg, 2nd edn. (2002)
3. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Congress on Evolutionary Computation. pp. 825–830. IEEE Service Center (2002)
4. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In: Proceedings Parallel Problem Solving from Nature - PPSN VI. vol. 1917, pp. 849–858 (2000)
6. Huband, S., Hingston, P., Barone, L., While, R.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2007)
7. Narayanan, S., Azarm, S.: On improving multiobjective genetic algorithms for design optimization. Structural optimization **18**(2), 146–155 (Oct 1999)
8. Sagawa, M., Aguirre, H., Daolio, F., Liefooghe, A., Derbel, B., Verel, S., Tanaka, K.: Learning variable importance to guide recombination. In: IEEE SSCI. pp. 1–7 (2016)
9. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolutionary Computation **11**(6), 712–731 (2007)
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems. pp. 95–100. International Center for Numerical Methods in Engineering (2001)
11. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Parallel Problem Solving from Nature - PPSN VIII. pp. 832–842. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)