# The Impact of Population Size, Number of Children, and Number of Reference Points on The Performance of NSGA-III

Ryoji Tanabe and Akira Oyama

Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, Kanagawa, Japan
tanabe@flab.isas.jaxa.jp, oyama@flab.isas.jaxa.jp

**Abstract.** We investigate the impact of three control parameters (the population size $\mu$, the number of children $\lambda$, and the number of reference points $H$) on the performance of Nondominated Sorting Genetic Algorithm III (NSGA-III). In the past few years, many efficient Multi-Objective Evolutionary Algorithms (MOEAs) for Many-Objective Optimization Problems (MaOPs) have been proposed, but their control parameters have been poorly analyzed. The recently proposed NSGA-III is one of most promising MOEAs for MaOPs. It is widely believed that NSGA-III is almost parameter-less and requires setting only one control parameter ($H$), and the value of $\mu$ and $\lambda$ can be set to $\mu = \lambda \approx H$ as described in the original NSGA-III paper. However, the experimental results in this paper show that suitable parameter settings of $\mu$, $\lambda$, and $H$ values differ from each other as well as their widely used parameter settings. Also, the performance of NSGA-III significantly depends on them. Thus, the usually used parameter settings of NSGA-III (i.e., $\mu = \lambda \approx H$) might be unsuitable in many cases, and $\mu$, $\lambda$, and $H$ require a particular parameter tuning to realize the best performance of NSGA-III.

## 1 Introduction

A Multi-Objective continuous Optimization Problem (MOP) which frequently appears in engineering problems can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_M(\boldsymbol{x}))^{\mathrm{T}} \\ \text{subject to} \quad & \boldsymbol{x} \in \mathbb{S} \subseteq \mathbb{R}^D \end{aligned} \tag{1}$$

where, $\boldsymbol{f} : \mathbb{S} \to \mathbb{R}^M$ is an objective function vector which consists of $M$ conflicting objective functions, and $\mathbb{R}^M$ is the objective function space. $\boldsymbol{x} = (x_1, ..., x_D)^{\mathrm{T}}$ is a $D$-dimensional solution vector, and $\mathbb{S} = \Pi_{j=1}^{D}[a_j, b_j]$ is the bound-constrained search space, where $a_j \leq x_j \leq b_j$ for each variable index $j \in \{1, ..., D\}$.

We say that $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$ and denote $\boldsymbol{x}^1 \prec \boldsymbol{x}^2$ if and only if $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)$ for all $i \in \{1, ..., M\}$ and $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$ for at least one index $i$. $\boldsymbol{x}^*$ is a Pareto optimal solution if there exists no $\boldsymbol{x} \in \mathbb{S}$ such that $\boldsymbol{x} \prec \boldsymbol{x}^*$. $\boldsymbol{f}(\boldsymbol{x}^*)$ is also called a Pareto optimal objective function vector. The set of all $\boldsymbol{x}^*$ in $\mathbb{S}$ is the

Pareto optimal solution Set (PS), and the set of all $\boldsymbol{f}(\boldsymbol{x}^*)$ is the Pareto Frontier (PF). The goal of MOPs is finding a set of well-distributed nondominated solutions which close to the PF in the objective function space.

A Multi-Objective Evolutionary Algorithm (MOEA) is one of most promising approaches for solving MOPs [14]. Since MOEAs use a set of individuals (solutions of a given problem) for the search, they can find good nondominated solutions in one single run. NSGA-II [5] proposed in the early 2000's is a representative MOEA. However, NSGA-II performs relatively well on MOPs with $M \leq 3$, but its performance significantly degrades on MOPs with $M \geq 4$ [22,13], where an MOP with $M \geq 4$ is called a Many-Objective continuous Optimization Problem (MaOP). Since MaOPs frequently appear in real-world applications [14], the poor performance of MOEAs is a critical problem. Thus, in the past few years, researchers in the Evolutionary Computation community have studied on the development of MOEAs which can handle a large number of objectives [14].

To the best of our knowledge, while many efficient MOEAs for MaOPs have been proposed in previous work, their control parameters have been poorly analyzed. Almost all previous work only proposed a novel MOEA and evaluated its performance on some benchmark problems, but analysis of their control parameters have been minimal. In particular, effects of the population size $\mu$ and the number of children $\lambda$ on the performance of MOEAs are poorly understood. It is well-known that the performance of EAs and MOEAs significantly depends on control parameter settings [8]. Therefore, users of MOEAs need to appropriately set their parameter values for a target application. General rules of thumb for parameter settings (e.g., the population size should be set to about 20 on six-objective multimodal MOPs) help users with such tedious tasks. In addition, since appropriate parameter settings might differ from each MOEA, the performance comparison between MOEAs with inappropriate ones is unfair [10,12]. An analysis of the impact of control parameter settings is also useful for understanding the performance behavior of MOEAs and designing a novel, efficient MOEA. For example, a small population size leads NSGA-II to the fast convergence [3] and is effective for expensive optimization problems.

In this paper, we analyze the impact of three control parameters (the population size $\mu$, the number of children $\lambda$, and the number of reference points $H$) on the performance of Nondominated Sorting Genetic Algorithm III (NSGA-III) [6]. We exhaustively investigate the performance of NSGA-III with various parameter settings of $\mu$, $\lambda$, and $H$ on the WFG benchmarks [9] with $M \in \{2, 4, 6, 8\}$. NSGA-III, one of most promising MOEAs for MaOPs, is an improved version of NSGA-II [5] by replacing the crowding distance-based selection with the reference vectors-based niching selection. NSGA-III can be categorized into the $(\mu + \lambda)$-MOEA framework. Therefore, NSGA-III essentially has the three control parameters ($\mu$, $\lambda$, and $H$), except for settings of variation operators (e.g., the crossover probability of SBX), and $\mu$, $\lambda$, and $H$ can be set independently of each other. However, it is widely believed that NSGA-III is almost parameterless and requires setting only one control parameter ($H$), and the value of $\mu$ and $\lambda$ can be set to $\mu = \lambda \approx H$ as described in [6]. In fact, $\mu$, $\lambda$, and $H$ of NSGA-III

were set to same values as $\mu = \lambda \approx H$ in almost all previous work (exceptions will be discussed in Section 3). Thus, the performance of NSGA-III with $\mu \neq \lambda \neq H$ has never been examined while parameter studies of MOEAs are important for practical uses as well as algorithm designs as discussed above. In this paper, we show that suitable parameter settings of $\mu$, $\lambda$, and $H$ values differ from each other as well as widely used parameter settings, and the performance of NSGA-III significantly depends on them.

This paper is organized as follows: Section 2 introduces the NSGA-III algorithm. We describe related work in Section 3. Section 4 describes experimental settings, and Section 5 presents the experimental analysis of $\mu$, $\lambda$, and $H$ of NSGA-III. Finally, Section 6 concludes this paper and discusses our future work.

## 2 NSGA-III

This section briefly introduces NSGA-III [6]. A NSGA-III population $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{\mu,t}\}$ is a set of individuals $\boldsymbol{x}^{i,t} = (x_1^{i,t}, ..., x_D^{i,t})^{\mathrm{T}}$, $i \in \{1, ..., \mu\}$. After the initialization of the population $\boldsymbol{P}^t$, the following steps are repeated until the termination criteria are met.

At the beginning of each iteration $t$, children $\boldsymbol{Q}^t = \{\boldsymbol{u}^{1,t}, ..., \boldsymbol{u}^{\lambda,t}\}$ are generated by applying variation operators (e.g., crossover and mutation operators) to randomly selected individuals from $\boldsymbol{P}^t$, where $\boldsymbol{u}^{i,t} = (u_1^{i,t}, ..., u_D^{i,t})^{\mathrm{T}}$, $i \in \{1, ..., \lambda\}$. After the generation of $\boldsymbol{Q}^t$, $\mu$ individuals composing the population in the next iteration $\boldsymbol{P}^{t+1}$ are selected from a union $\boldsymbol{R}^t = \boldsymbol{P}^t \cup \boldsymbol{Q}^t$, where $|\boldsymbol{R}^t| = \mu + \lambda$. The nondominated sorting [5] is applied to $\boldsymbol{R}^t$, and the individuals in $\boldsymbol{R}^t$ are grouped as $\{\boldsymbol{F}^{1,t}, \boldsymbol{F}^{2,t}, ...\}$ according to their nondomination levels. Starting from $i = 1$, the individuals in $\boldsymbol{F}^{i,t}$ is added to a new temporal population $\boldsymbol{S}^t$, and the index counter $i$ is incremented as $i = i + 1$. This procedure is repeated until the size of $\boldsymbol{S}^t$ equals or exceeds $\mu$. The index counter $i$ at the end of the nondomination levels-based selection is recorded as $l$. When $|\boldsymbol{S}^t| = \mu$, $\boldsymbol{P}^{t+1} = \boldsymbol{S}^t$, and this environmental selection procedure is finished. Otherwise (i.e., $|\boldsymbol{S}^t| > \mu$), all individuals in $\boldsymbol{S}^t \backslash \boldsymbol{F}^{l,t}$ are added to $\boldsymbol{P}^{t+1}$, and remaining $K$ individuals ($K = \mu - |\boldsymbol{P}^{t+1}|$) are selected from $\boldsymbol{F}^{l,t}$ so that $|\boldsymbol{P}^{t+1}| = \mu$ according to the following reference vectors-based niching selection.

First, for each $f_i$ ($i \in \{1, ..., M\}$), for each $\boldsymbol{x} \in \boldsymbol{S}^t$, the objective function value $f_i(\boldsymbol{x})$ is normalized into $[0, 1]$ by an originally defined normalization rule in [6]. After the normalization, the niching procedure with the reference vectors is applied to each $\boldsymbol{x} \in \boldsymbol{S}^t$. The reference vector consists of the reference point $\boldsymbol{w} = (w_1, ..., w_M)^{\mathrm{T}}$, $\sum_{j=1}^{M} w_j = 1$ and the ideal point which is identical to the origin $(0, ..., 0)^{\mathrm{T}}$. NSGA-III requires a set of the reference points $\boldsymbol{W} = \{\boldsymbol{w}^1, ..., \boldsymbol{w}^H\}$. $\boldsymbol{W}$ can be generated by using any method, but it is desirable that $\boldsymbol{W}$ consist of uniformly distributed reference points when a user's preference information cannot be utilized for the search.

For each $\boldsymbol{x} \in \boldsymbol{S}^t \backslash \boldsymbol{F}^{l,t}$ and each reference vector, the perpendicular distance between the normalized objective function vector $\boldsymbol{f}'(\boldsymbol{x})$ and the reference vector

is calculated. Then, $\boldsymbol{x}$ is associated with a reference vector which has the minimum distance. For each $\boldsymbol{w}^i$ ($i \in \{1, ..., H\}$), the number of associated individuals is counted as $\rho_{i,t}$. After the association procedure of all members in $\boldsymbol{S}^t \backslash \boldsymbol{F}^{l,t}$, the other members in $\boldsymbol{F}^{l,t} \in \boldsymbol{S}^t$ are also associated in the same manner, but the niche count $\rho$ is not counted at this time.

Here, $K$ individuals in $\boldsymbol{F}^{l,t}$ ($K = \mu - |\boldsymbol{P}^{t+1}|$) is selected for the population in the next iteration $\boldsymbol{P}^{t+1}$. First, $\boldsymbol{w}^{\min,t}$ having the minimum niche count $\rho_{\min,t} = \min_{i=1}^{H}\{\rho_{i,t}\}$ is selected. When there are more than two such reference points, one of them is randomly selected as $\boldsymbol{w}^{\min,t}$. Let $\boldsymbol{I}^t$ be a set of the individuals in $\boldsymbol{F}^{l,t}$ associated with $\boldsymbol{w}^{\min,t}$. In this case, the following three situations (i)–(iii) can be considered: (i) $|\boldsymbol{I}^t| = 0$, (ii) $|\boldsymbol{I}^t| \geq 1$ and $\rho_{\min,t} = 0$, and (iii) $|\boldsymbol{I}^t| \geq 1$ and $\rho_{\min,t} \geq 1$. In the case of (i), $\boldsymbol{w}^{\min,t}$ is temporarily removed from $\boldsymbol{W}$ at the iteration $t$. In the case of (ii), an individual having the minimum perpendicular distance between the normalized objective function vector $\boldsymbol{f}'(\boldsymbol{x})$ and the reference vector is selected from $\boldsymbol{I}^t$ and added to $\boldsymbol{P}^{t+1}$. In the case of (iii), a randomly selected individual in $\boldsymbol{I}^t$ is added to $\boldsymbol{P}^{t+1}$. In any of the situations (i)–(iii), the selected individual is removed from $\boldsymbol{I}^t$, and $\rho_{\min,t} = \rho_{\min,t} + 1$. The niching procedure described above is repeatedly applied until $|\boldsymbol{P}^{t+1}| = \mu$.

## 3 Previous parameter studies of NSGA-III

Here, we review previous parameter studies of three control parameters (the population size $\mu$, the number of children $\lambda$, and the number of reference points $H$) of NSGA-III. In [6], Deb and Jain visually verified that NSGA-III with small $\mu$ and $H$ values can find good nondominated solutions on three-objective functions. Seada and Deb evaluated the convergence performance of U-NSGA-III with $\mu > H$ on two-objective MOPs [18], where U-NSGA-III is a unified version of NSGA-III for single, multi, and many-objective optimization. The experimental results in [18] show that the larger $\mu$ value than $H$ is suitable for U-NSGA-III when solving a multi-modal MOP. Andersson et. al. presented a parameter tuning study of NSGA-II and NSGA-III [1]. $\mu$ ($\approx H$) and four control parameters of the SBX crossover and polynomial mutation (i.e., the crossover/mutation probability $p_c$ and $p_m$, the distribution index $\eta_c$ and $\eta_m$) of NSGA-II and NSGA-III were tuned using a meta-GA. However, the work [1] mainly focused on the latter four control parameters ($p_c$, $p_m$, $\eta_c$, and $\eta_m$) of the variation operators, and the discussion on the tuned $\mu$ ($\approx H$) value was minimal. The experiments in the two previous work [18,1] were performed only on two-objective MOPs, and the impact of $\mu$, $\lambda$, and $H$ was not investigated on MaOPs with $M \geq 4$. Also, $\lambda$ was set as $\lambda = \mu$ in [6,18], and three parameters were set as $\mu = \lambda \approx H$ in [1].

Very recently, Ishibuchi et. al. [12] examined the performance of various MOEAs, including NSGA-III, with various $\mu$ values on the three- and five-objective functions. Their experimental results show that the performance of the MOEAs significantly depends on the settings of $\mu$, and a suitable $\mu$ value differs from each MOEA. However, they set the three parameter values of NSGA-

III as $\mu = \lambda \approx H$ and did not examine the performance of NSGA-III (and the other MOEAs) with $\mu \neq \lambda \neq H$ in the same way as [1].

In summary, the performance of NSGA-III with different $\mu$, $\lambda$, and $H$ values has been poorly understood. Note that the situation is not unique to NSGA-III. Whereas there is much previous work on parameter studies of variation operators [21,25], the impact of $\mu$ and $\lambda$ on the performance of MOEAs has been hardly investigated. While there are several parameter studies on $\mu$ such as [11,23,17,3], they set $\mu$ and $\lambda$ to the same value, and the performance of MOEAs with $\mu \neq \lambda$ has been poorly discussed (the only one exception [7] will be discussed in Section 5.2).

## 4 Experimental settings

We used the nine WFG functions [9] with $M \in \{2, 4, 6, 8\}$. The shapes of the PFs of the WFG1, WFG2, and WFG3 functions are complicated, discontinuous, and degenerate respectively, and those of the others WFG functions are nonconvex. The WFG4 and WFG9 functions are multimodal, and the WFG2, WFG6, WFG8, and WFG9 functions are also nonseparable. As suggested in [9], the position parameter $k$ was set to $k = 2(M - 1)$, and the distance parameter $l$ was set to $l = 20$, where the number of variables $D$ is $D = k + l$.

We used the source code of NSGA-III implemented by Yuan[1]. We used the SBX crossover and polynomial mutation as in the original NSGA-III [6]. As suggested in [6], we set the control parameters of the variation operators as follows: $p_c = 1.0$, $\eta_c = 30$, $p_m = 1/D$, $\eta_m = 20$. The number of independent runs was set to 31 for $M \in \{2, 4, 6\}$ and only 11 for $M = 8$ due to expensive computational cost of the hypervolume calculation. The number of maximum function evaluations was set to $1 \times 10^4$ as [2]. We set the population size $\mu$, the number of children $\lambda$, and the number of reference points $H$ as follows: $\mu \in \{10, 20, 40, 80, 160, 320\}$, $\lambda \in \{1, 10, 20, 40, 80, 160, 320, 640\}$, and $H \in \{10, 20, 40, 80, 160\}$ (i.e., $6 \times 8 \times 5 = 240$ configurations were verified).

We used the hypervolume (HV) indicator [28] for evaluating the quality of a set of obtained nondominated solutions $\boldsymbol{A}$[2]. Before calculating the HV value (as well as the MinSum and SumMin values [13] used in Section 5.3), the objective function vector $\boldsymbol{f}(\boldsymbol{x})$ of each $\boldsymbol{x} \in \boldsymbol{A}$ was normalized using the ideal point $(0, ..., 0)^{\mathrm{T}}$ and the nadir point $(2, ..., 2M)^{\mathrm{T}}$ as in [26]. The reference point for calculating HV was set to $(1.1, ..., 1.1)^{\mathrm{T}}$. Almost all previous work (e.g., [27,2,25,18,26]) used nondominated solutions in the bounded archive (= the population $\boldsymbol{P}$) for calculating the HV value. The bounded archive maintains (nondominated) solutions obtained during the search process, but its size is limited. As pointed out in [10], when using the bounded archive and the HV as the performance indicator, the large $\mu$ values are more beneficial than the small $\mu$ values. Furthermore, a monotonic increase of the hypervolume over time (=

[1] The code was downloaded from http://www.cs.bham.ac.uk/~xin/journal_papers.html

[2] Since the IGD indicator [28] used in [6] is unsuitable for comparing nondominated solution sets of different size as pointed out in [10], we did not use it.

**Algorithm 1:** Generation method of a set of reference points $\boldsymbol{W}$

1 Generate $\boldsymbol{W}^{\mathrm{tmp}} = \{\boldsymbol{w}^{\mathrm{tmp},1}, ..., \boldsymbol{w}^{\mathrm{tmp},H^{\mathrm{tmp}}}\}$ using SLD so that $H^{\mathrm{tmp}} \gg H$;
2 $\boldsymbol{W} \leftarrow \emptyset$, $\boldsymbol{w}^c = (1/M, ..., 1/M)^{\mathrm{T}}$, $\boldsymbol{W} \leftarrow \boldsymbol{W} \cup \{\boldsymbol{w}^c\}$;
3 **for** $i = 1$ **to** $H^{\mathrm{tmp}}$ **do**
4  $d_i \leftarrow \mathrm{EuclideanDistance}(\boldsymbol{w}^{\mathrm{tmp},i}, \boldsymbol{w}^c)$, $V_i \leftarrow TRUE$;
5 **while** $|\boldsymbol{W}| < H$ **do**
6  $k \leftarrow \underset{j|V_j=TRUE}{\arg\max}\ d_j$, $V_k \leftarrow FALSE$, $\boldsymbol{W} \leftarrow \boldsymbol{W} \cup \{\boldsymbol{w}^{\mathrm{tmp},k}\}$;
7  **for** $l = 1$ **to** $H^{\mathrm{tmp}}$ **do**
8  If $V_l = TRUE$, then $d_l \leftarrow \min(\mathrm{EuclideanDistance}(\boldsymbol{w}^{\mathrm{tmp},l}, \boldsymbol{w}^{\mathrm{tmp},k}), d_l)$;

the number of function evaluations) cannot be ensured when using the bounded archive [15,17]. Thus, we cannot perform a fair comparison of MOEAs with various $\mu$ values using the bounded archive. As suggested in [15,17,1,3,12], for the fair comparison of NSGA-III with various $\mu$ values, we used the *unbounded* archive that stores *all* nondominated solutions found during the search process.

As described in Section 2, NSGA-III requires a set of reference points $\boldsymbol{W} = \{\boldsymbol{w}^1, ..., \boldsymbol{w}^H\}$. In the original NSGA-III paper [6], Simplex-Lattice Design (SLD) [4] and Two-layered SLD (TSLD) [6] were used for generating $\boldsymbol{W}$. SLD and TSLD are the widely used weight/reference points generation methods, but they cannot generate $\boldsymbol{W}$ with arbitrary size [20,24]. That is, they are not suitable for our scale-up studies of $H$ and $M$. In addition to SLD and TSLD, generation methods of $\boldsymbol{W}$ with arbitrary size were proposed in [27,20,16]. In this paper, we used a generation method of $\boldsymbol{W}$ inspired by the proposed methods in [27,16]. Algorithm 1 shows the procedure. First, a set of candidate reference points $\boldsymbol{W}^{\mathrm{tmp}} = \{\boldsymbol{w}^{\mathrm{tmp},1}, ..., \boldsymbol{w}^{\mathrm{tmp},H^{\mathrm{tmp}}}\}$ are generated using SLD so that $H^{\mathrm{tmp}} \gg H$. For each $M \in \{2, 4, 6, 8\}$, we set the division number of SLD to $10^5, 83, 24$, and 14 respectively (i.e., $H^{\mathrm{tmp}} = 100\,001$, $102\,340$, $118\,755$, and $116\,280$). Then, after adding a central point $\boldsymbol{w}^c = (1/M, ..., 1/M)^{\mathrm{T}}$ to $\boldsymbol{W}$, a reference point in $\boldsymbol{W}^{\mathrm{tmp}}$ having the largest Euclidean distance value from ones already belonging to $\boldsymbol{W}$ is selected and added to $\boldsymbol{W}$ (lines 2–4 and 5–8 in Algorithm 1 respectively). This trial is repeated until the size of $\boldsymbol{W}$ is equal to $H$.

## 5 Experimental results and discussion

We analyze the impact of the parameter settings of $H, \lambda$, and $\mu$ on the performance of NSGA-III. The results are shown in Section 5.1, Section 5.2, and Section 5.3 respectively. We also discuss how much the performance of NSGA-III can be improved by tuning the $\mu$, $\lambda$, and $H$ values in Section 5.4.

### 5.1 Influence of $H$ and $\mu$ values on the performance of NSGA-III

Figure 1 shows the influence of the $H$ values on the performance of NSGA-III with the various $\mu$ values on the WFG1, WFG2, and WFG4 functions with

$M \in \{2, 4, 6, 8\}$ at the number of function evaluations $= 1 \times 10^4$. We can see that the choice of $\mu$ and $H$ values significantly affects the performance of NSGA-III, and their appropriate values also depend on the function, $M$, and the other parameter values. Due to space constraints, we show only the results on the selected functions, but the results on the WFG3, 7, 9 functions and the WFG5, 6, 8 functions are similar to the WFG1 and WFG4 functions respectively. Note that some HV values in Figure 1 exceed 1 since the reference point for the HV calculation was set to $(1.1, ..., 1.1)^{\mathrm{T}}$.

First, we discuss the impact of the $\mu$ values on the performance of NSGA-III. NSGA-III with $\mu \in \{40, 80\}$ and $\{80, 160\}$ achieves the relatively higher HV values for the WFG1 and WFG2 functions respectively. Large $\mu$ values might be suitable for NSGA-III when solving a MOP with the disconnected PF (e.g., WFG2 function). The results on the multimodal WFG4 function with $M \in \{2, 4\}$ is counterintuitive – NSGA-III with $\mu = 10$ performs best. With an increasing $M$, an appropriate $\mu$ value is also increased as $\mu \in \{20, 40\}$ for $M = 6$ and $\mu \in \{40, 80\}$ for $M = 8$. The reason for this is simply that the objective function space is increasing according to $M$, and large $\mu$ values become suitable for NSGA-III. The reason why NSGA-III with the smallest population size $\mu = 10$ performs well on the WFG4 function with $M \in \{2, 4\}$ can be considered as follows: The WFG4 function is multimodal, but its shape of the PF is not so complex such as that of the WFG1 and WFG2 functions. Therefore, NSGA-III with the smallest $\mu$ value can find good nondominated solutions. NSGA-III with $\mu = 320$ performs poorly in many cases but outperforms NSGA-III using the small $\mu$ values when $M$ increases. Therefore, large $\mu$ values might be suitable for solving MaOPs with a very large $M$ (i.e., $M \geq 10$).

Then, we analyze the impact of the $H$ values on the performance of NSGA-III. As shown in Figure 1, for the WFG functions with $M \in \{2, 4\}$, the increase of the $H$ values deteriorates the performance of NSGA-III with the small population size $\mu \in \{10, 20\}$ while it does not significantly influence the performance of NSGA-III with the large $\mu$ values. On the other hand, for the WFG functions with $M \in \{6, 8\}$, there exists a particular $H$ value which is suitable for NSGA-III with each $\mu$ value. For example, NSGA-III with $\mu = 80$ achieves the highest HV values on the eight-objective WFG1, WFG2 and WFG4 functions when using $H = 160$. Since large $H$ values increase the number of niches in the environmental selection and promote diversity in the population (see Section 2), they might be suitable for NSGA-III when solving MaOPs as shown Figure 1. Seada and Deb [18] reports that the larger $\mu$ value than $H$ is suitable for U-NSGA-III when solving a two-objective multimodal MOP (see Section 3). Although the examined version of NSGA-III is slightly different from [18], the experimental results of this study show that the use of the smaller $\mu$ value than $H$ can improve the performance of NSGA-III on MaOPs with $M \geq 6$.

### 5.2 Influence of $\lambda$ and $\mu$ values on the performance of NSGA-III

Figure 2 shows the influence of the $\lambda$ values on the performance of NSGA-III with the various $\mu$ values. For $M = 2$, the large $\lambda$ values deteriorate the performance

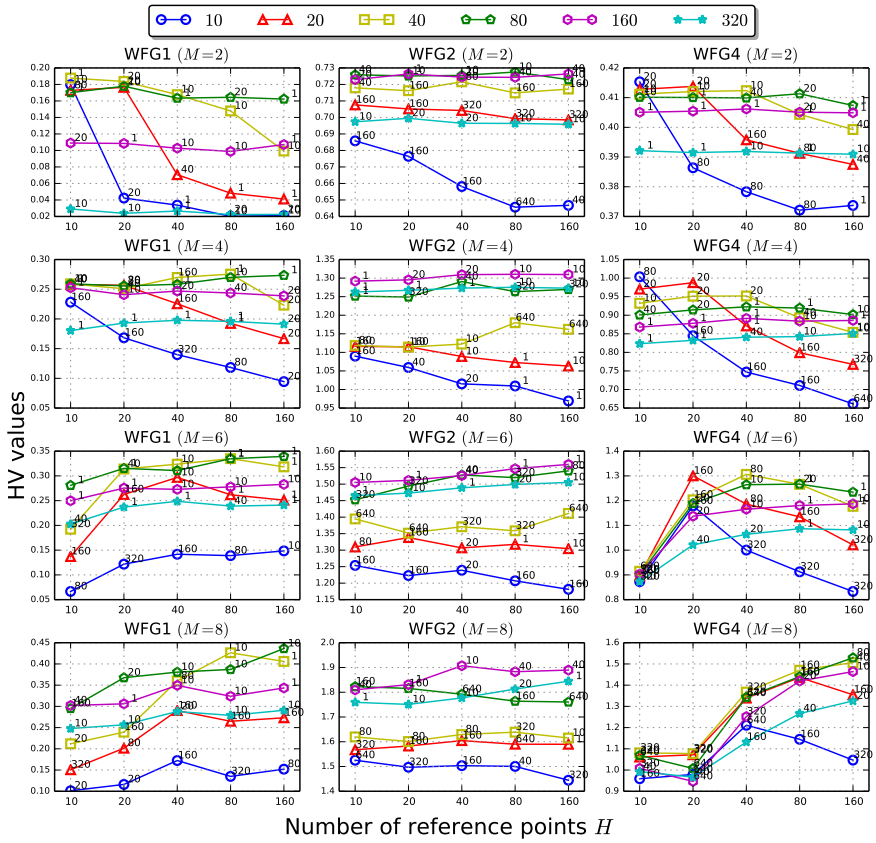


Fig. 1: Influence of the $H$ values on the performance of NSGA-III with the various $\mu$ values on the WFG1, WFG2, and WFG4 functions with $M = \{2, 4, 6, 8\}$. We show the results of NSGA-III with the best configuration of $\mu$, $H$, and $\lambda$ values which achieved the best median HV value at the number of function evaluations $= 1 \times 10^4$. The numbers in the figure represent the corresponding $\lambda$ values.

of NSGA-III. This is consistent with the results reported in the previous work [7]. In [7], Durillo et. al. investigated the performance of the steady state version of NSGA-II and SPEA2 (i.e., $\lambda = 1$) on two-objective MOPs, and the results show their promising performance. This trend can also be seen in the results of NSGA-III with the large $\mu$ values ($\mu \in \{160, 320\}$) on the WFG functions with $M \in \{4, 6, 8\}$. On the other hand, for the WFG functions with $M \in \{6, 8\}$, there is a particular $\lambda$ value for each $\mu$ value which makes NSGA-III with the small $\mu$ values ($\mu \in \{10, 20, 40, 80\}$) efficient. For example, NSGA-III with $\mu = 10$ and $\lambda = 80$ performs best on the WFG4 function with $M = 4$.

In [7], the performance comparison of $(\mu + \mu)$-MOEAs and $(\mu + 1)$-MOEAs was performed only on MOPs with $M = 2$. Our results show that the larger $\lambda$ value than $\mu$ might be suitable for NSGA-III with small $\mu$ values on MaOPs

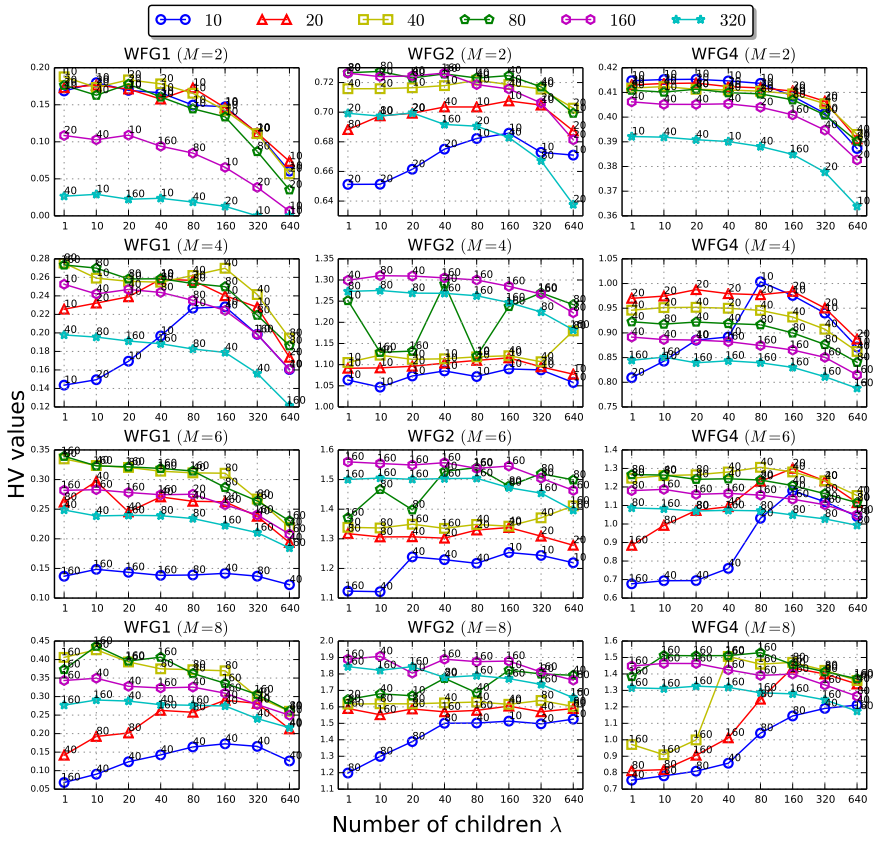


Fig. 2: Influence of the $\lambda$ values on the performance of NSGA-III with the various $\mu$ values on the WFG1, WFG2, and WFG4 functions with $M = \{2, 4, 6, 8\}$. We show the results of NSGA-III with the best configuration of $\mu$, $H$, and $\lambda$ values which achieved the best median HV value at the number of function evaluations $= 1 \times 10^4$. The numbers in the figure represent the corresponding $H$ values.

with $M \geq 4$. Unfortunately, we cannot explain the reason why NSGA-III with $\lambda > \mu$ performs well on MaOPs. A further investigation of the impact of $\lambda$ on the performance of NSGA-III is our future work.

### 5.3 Convergence behavior of NSGA-III with various $\mu$ values

We here analyze the convergence behavior of NSGA-III with the various $\mu$ values. For the analysis, in addition to the HV indicator, we also measured two convergence metrics (MinSum and SumMin [13]) and the number of nondominated solutions stored in the unbounded archive $\boldsymbol{A}$[3]. MinSum indicates only the con-

[3] Since, as far as we know, there is no good diversity indicator for the *unbounded* archive, we could not measure the diversity of the obtained nondominated solutions.

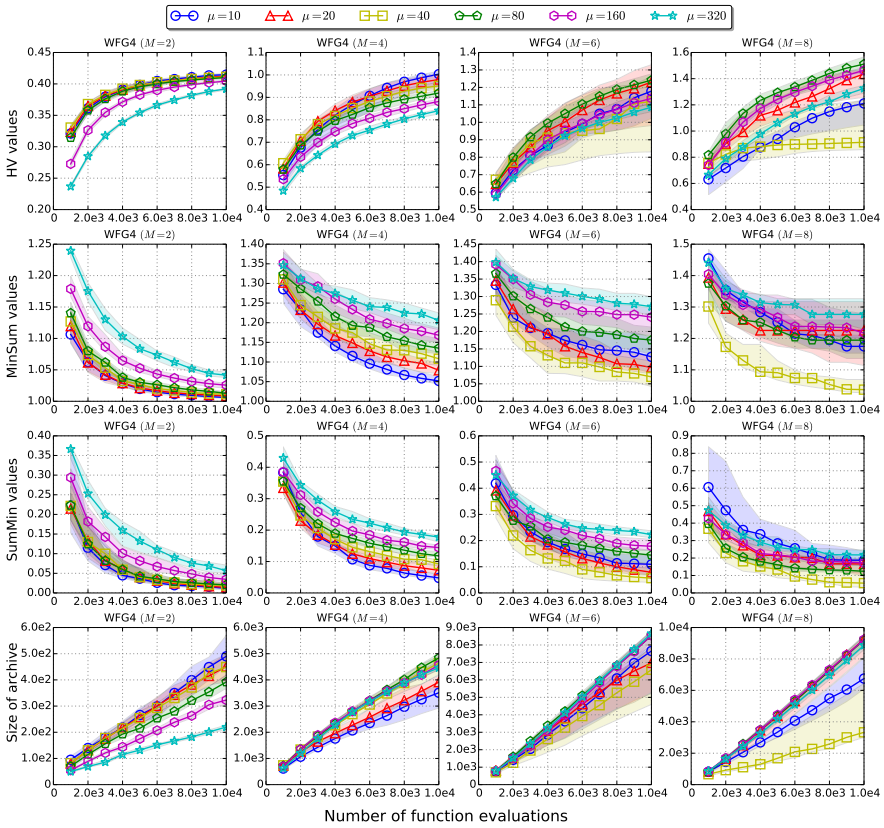


Fig. 3: Convergence behavior of NSGA-III with the various $\mu$ values. The figures show the HV, MinSum, SumMin, and $|\boldsymbol{A}|$ over the number of function evaluations on the WFG4 function with $M \in \{2, 4, 6, 8\}$. We plot the median and quartile values of each indicator across all runs. We show the data of NSGA-III with the best configuration of $\mu$, $H$, and $\lambda$ values which achieved the best median HV value at the number of function evaluations $= 1 \times 10^4$.

vergence quality of $\boldsymbol{A}$ to the center of the PF. The MinSum value of $\boldsymbol{A}$ is calculated as follows: $\mathrm{MinSum}(\boldsymbol{A}) = \min_{\boldsymbol{x} \in \boldsymbol{A}} \{\sum_{j=1}^{M} f_j(\boldsymbol{x})\}$. Also, SumMin evaluates only the convergence performance of $\boldsymbol{A}$ toward the PF around its $M$ edges. The SumMin value of $\boldsymbol{A}$ is obtained as follows: $\mathrm{SumMin}(\boldsymbol{A}) = \sum_{j=1}^{M} \min_{\boldsymbol{x} \in \boldsymbol{A}} \{f_j(\boldsymbol{x})\}$. We also report the number of nondominated solutions stored in the unbounded archive $\boldsymbol{A}$.

Figure 3 shows the HV, MinSum, SumMin, and $|\boldsymbol{A}|$ over the number of function evaluations on the WFG4 function with $M \in \{2, 4, 6, 8\}$. We here only discuss the results on the WFG4 function, but note that the obtained results highly depend on the function instance. The analysis of the convergence behavior of NSGA-III on the other WFG functions is our future work.

Figure 3 shows that the large $\mu$ values make the convergence speed of NSGA-III slow for $M \in \{2, 4, 6\}$. For example, NSGA-III with $\mu = 320$ achieves the worst HV value at any time, and its convergence speed towards the PF is slow as shown in the MinSum and SumMin values. The results are consistent with the previous work by Brockhoff et. al. [3] which investigated the anytime behavior of NSGA-II with various $\mu$ values on two-objective MOPs. NSGA-III with the small $\mu$ values ($\mu \in \{10, 20\}$) show the good anytime behavior on the WFG4 function with $M \in \{2, 4\}$ and the strong convergence to the PF (see the MinSum and SumMin values). On the other hand, the performance of NSGA-III with $\mu = 10$ is poor on the WFG4 function with $M \in \{6, 8\}$, and $\mu = 80$ seems to be suitable. For $M = 8$, NSGA-III with $\mu = 10$ performs worse than NSGA-III with $\mu = 320$. Although NSGA-III with $\mu = 40$ achieves the best MinSum and SumMin value, its HV value is worst. This might be due to its low number of obtained nondominated solutions $|\boldsymbol{A}|$. In summary, the results show that large $\mu$ values make the convergence speed of NSGA-III slow but might be suitable for MaOPs with $M \geq 6$.

### 5.4 Generalist vs. Specialist: How much can the performance of NSGA-III be improved by tuning $\mu$, $\lambda$, and $H$ values?

So far in this paper, we have analyzed the impact of parameter settings on the performance of NSGA-III for each WFG function instance. We here investigate how much the performance of NSGA-III can be improved by tuning the $\mu$, $\lambda$, and $H$ values. We consider the following two parameter tuning scenarios: (1) finding the parameter settings that robustly perform well on a given set of functions, and (2) finding ones that perform well only on a specific function. We say that the parameter settings obtained by the parameter tuning on the scenarios (1) and (2) are *generalist* and *specialist* respectively, where they were named by Smit and Eiben in [19].

First, we deal with the parameter tuning scenario (1) for finding the generalist. We used the Average Performance Score (APS) [2]. Recall that we verified the 240 configurations of $\mu$, $\lambda$, and $H$ in this paper (see Section 4). Suppose that we compare the performance of $n$ algorithms $\{A_1, ..., A_n\}$ on a given problem instance using the HV values obtained in multiple runs ($n = 240$ in this study). For each $i \in \{1, ..., n\}$ and $j \in \{1, ..., n\} \backslash \{i\}$, let $\delta_{i,j} = 1$, if $A_j$ outperforms $A_i$ based on the Wilcoxon rank-sum test with $p < 0.05$, otherwise $\delta_{i,j} = 0$. Then, the performance score $P(A_i)$ is defined as follows: $P(A_i) = \sum_{j \in \{1,...,n\} \backslash \{i\}}^{n} \delta_{i,j}$. $P(A_i)$ represents the number of compared algorithms that outperforms $A_i$. The APS is an average of the $P(A_i)$ values on all problem instances.

Table 1 provides the combination of $\mu$, $\lambda$, and $H$ values which achieved the smallest APS value for each $M$ on all WFG functions, i.e., the generalist for each $M$. We have no intention of claiming that NSGA-III with the generalist shown in Table 1 performs best on all possible problems (it performs well only on the WFG benchmarks). As discussed in [19], obtaining such "best" parameter settings is impossible in practice. We can see that $\mu = 40$ is relatively suitable for NSGA-III for $M \in \{2, 4, 6\}$. On the other hand, NSGA-III with $\mu = 80$ performs well on

Table 1: Generalist for each $M$: Combination of ($\mu$, $\lambda$, $H$) values that achieved the smallest APS value for each $M$ at the number of function evaluations $= 1 \times 10^4$.

| $M = 2$ | $M = 4$ | $M = 6$ | $M = 8$ |
|---|---|---|---|
| $(40, 10, 20)$ | $(40, 10, 40)$ | $(40, 80, 40)$ | $(80, 40, 80)$ |

Table 2: Specialist for each WFG function with $M = 8$: Combination of ($\mu$, $\lambda$, $H$) values that achieved the highest median HV value for each eight-objective WFG function at the number of function evaluations $= 1 \times 10^4$.

| WFG1 | WFG2 | WFG3 | WFG4 | WFG5 | WFG6 | WFG7 | WFG8 | WFG9 |
|---|---|---|---|---|---|---|---|---|
| (80, 10, 160) | (160, 10, 40) | (160, 80, 20) | (80, 80, 160) | (40, 80, 40) | (40, 10, 40) | (40, 80, 40) | (40, 80, 40) | (80, 20, 40) |

the WFG functions with $M = 8$. The results also show that the appropriate $H$ values depend on $M$ as discussed in Section 5.1. Unfortunately, general rules of thumb for the appropriate parameter settings of $\lambda$ cannot be found in Table 1.

Then, we deal with the parameter tuning scenario (2) for finding the specialist. Table 2 shows the combination of ($\mu$, $\lambda$, $H$) values which achieved the best median HV value for each eight-objective WFG function, i.e., the specialist for each WFG function with $M = 8$. The results show that the appropriate parameter settings of $\mu$, $\lambda$, and $H$ significantly depend on a given function. When solving the WFG1 and WFG4 functions, the large $H$ value is suitable, but NSGA-III with the small $H$ value ($H = 40$) performs best on the other WFG functions, except for the WFG3 function. The large $\mu$ value is also suitable for the WFG3 and WFG4 functions.

Although the parameter tuning is not the main concern of this paper, it is interesting to compare the performance of NSGA-III with different parameter configurations. Figure 4 shows the convergence behavior of NSGA-III with the generalist, specialist, and default parameter settings of $(\mu, \lambda, H)$ on the WFG functions with $M = 8$. For the default parameter settings, we set $\mu = \lambda = H = 156$ as suggested in the original NSGA-III paper [6]. We also show the results of NSGA-III with default parameter settings using a set of reference points generated by TSLD (see Section 4), i.e., the original NSGA-III. For the generalist and specialist, we used the parameter settings shown in Table 1 and 2 respectively.

Figure 4 shows that NSGA-III with the specialist clearly outperforms NSGA-III with the three other parameter settings on all WFG functions with $M = 8$. NSGA-III with the generalist performs better than NSGA-III with the default settings, except for the WFG2 and WFG3 functions. Since the shapes of the PFs of the WFG2 and WFG3 functions are complex, NSGA-III requires particular parameter settings (see Table 2). Due to this reason, we consider that NSGA-III with the generalist did not work well on these WFG functions. We cannot find a clear winner on the comparison between NSGA-III with the default parameter settings using a set of reference points $\boldsymbol{W}$ generated by Algorithm 1 and TSLD

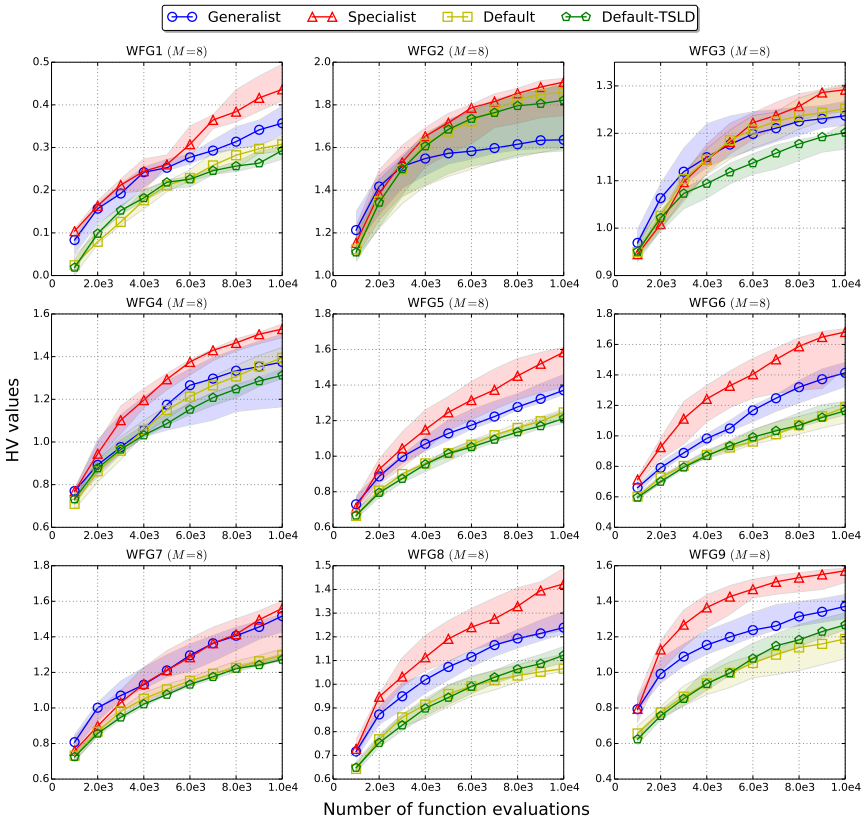


Fig. 4: Convergence behavior of NSGA-III with the generalist, specialist, and default settings of $(\mu, \lambda, H)$ on the WFG functions with $M = 8$. We plot the median and quartile HV values across all 11 runs. For the default settings, we set $\mu = \lambda = H = 156$ as suggested in the original NSGA-III paper [6]. We also show the results of NSGA-III with default settings using a set of reference points generated by TSLD, i.e., the original NSGA-III. For the generalist and specialist, see Table 1 and 2 respectively.

(labeled as "Default" and "Default-TSLD" in Figure 4 respectively). A further performance evaluation of reference points generation methods is an avenue for future work. However, Algorithm 1 has an advantage compared to TSLD – it can generate a set of reference points $\boldsymbol{W}$ with arbitrary size.

In summary, the results show that (i) the performance of NSGA-III can be greatly improved by tuning the parameter settings of $\mu$, $\lambda$, and $H$ individually, and (ii) their appropriate settings depend on the function instance.

## 6 Conclusion

We analyzed the impact of the population size $\mu$, the number of children $\lambda$, and the number of reference points $H$ on the performance of NSGA-III [6].

It was widely believed that NSGA-III was almost parameter-less and requires setting only one control parameter ($H$), and the value of $\mu$ and $\lambda$ could be set to $\mu = \lambda \approx H$ [6]. In contrast, our experimental results showed that appropriate parameter settings of $\mu$, $\lambda$, and $H$ differ from each other as well as widely used parameter settings. Also, the performance of NSGA-III significantly depends on the settings of $\mu$, $\lambda$, and $H$. In fact, NSGA-III with a carefully tuned parameter setting (i.e., the specialist) performs significantly better than NSGA-III with the widely used settings. Thus, the usually used parameter settings of NSGA-III (i.e., $\mu = \lambda \approx H$) might be unsuitable in many cases, and NSGA-III requires a particular parameter tuning to realize its best performance.

Future work will analyze the behavior of NSGA-III with different values of $\mu$, $\lambda$, and $H$. We will also investigate the relationship between good parameter combinations of NSGA-III and the shape of the PF by using some statistical approaches (e.g., main effect plots). While the focus of this study was limited to the bound-constrained WFG functions, an analysis on constrained MOPs is an avenue for future work. We believe that the effect of the parameter settings of $\mu$, $\lambda$, and $H$ is applicable to other reference vectors-based MOEAs such as U-NSGA-III [18], $\theta$-DEA [26], and VaEA [24], and its investigation is also our future direction.

**Acknowledgments.** This research is supported by the HPCI System Research Project "Research and development of multiobjective design exploration and high-performance computing technologies for design innovation" (Project ID:hp160203).

## References

1. M. Andersson, S. Bandaru, A. H. C. Ng, and A. Syberfeldt. Parameter Tuning of MOEAs Using a Bilevel Optimization Approach. In *EMO*, pages 233–247, 2015.
2. J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evol. Comput.*, 19(1):45–76, 2011.
3. D. Brockhoff, T. Tran, and N. Hansen. Benchmarking Numerical Multiobjective Optimizers Revisited. In *GECCO*, pages 639–646, 2015.
4. I. Das and J. E. Dennis. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optimiz*, 8(3):631–657, 1998.
5. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TEVC*, 6(2):182–197, 2002.
6. K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE TEVC*, 18(4):577–601, 2014.
7. J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba. On the Effect of the Steady-State Selection Scheme in Multi-Objective Genetic Algorithms. In *EMO*, pages 183–197, 2009.
8. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE TEVC*, 3(2):124–141, 1999.
9. S. Huband, P. Hingston, L. Barone, and R.L.While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE TEVC*, 10(5):477–506, 2006.

10. H. Ishibuchi, H. Masuda, and Y. Nojima. Comparing solution sets of different size in evolutionary many-objective optimization. In *IEEE CEC*, pages 2859–2866, 2015.
11. H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima. Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations. In *IEEE SMC*, pages 1758–1763, 2009.
12. H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima. How to compare many-objective algorithms under different settings of population and archive sizes. In *IEEE CEC*, pages 1149–1156, 2016.
13. H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE CEC*, pages 2419–2426, 2008.
14. B. Li, J. Li, K. Tang, and X. Yao. Many-Objective Evolutionary Algorithms: A Survey. *ACM Comput. Surv.*, 48(1):13, 2015.
15. M. López-Ibáñez, J. D. Knowles, and M. Laumanns. On Sequential Online Archiving of Objective Vectors. In *EMO*, pages 46–60, 2011.
16. S. Martínez, H. E. Aguirre, K. Tanaka, and C. A. C. Coello. On the low-discrepancy sequences and their use in MOEA/D for high-dimensional objective spaces. In *IEEE CEC*, pages 2835–2842, 2015.
17. A. Radulescu, M. López-Ibáñez, and T. Stützle. Automatically Improving the Anytime Behaviour of Multiobjective Evolutionary Algorithms. In *EMO*, pages 825–840, 2013.
18. H. Seada and K. Deb. A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE TEVC*, 2016 (in press).
19. S. K. Smit and A. E. Eiben. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. In *EvoApplications*, pages 542–551, 2010.
20. Y. Tan, Y. Jiao, H. Li, and X. Wang. MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & OR*, 40(6):1648–1660, 2013.
21. T. Tusar and B. Filipic. Differential Evolution versus Genetic Algorithms in Multiobjective Optimization. In *EMO*, pages 257–271, 2006.
22. T. Wagner, N. Beume, and B. Naujoks. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In *EMO*, pages 742–756, 2007.
23. S. Wessing, N. Beume, G. Rudolph, and B. Naujoks. Parameter Tuning Boosts Performance of Variation Operators in Multiobjective Optimization. In *PPSN*, pages 728–737, 2010.
24. Y. Xiang, Y. Zhou, M. Li, and Z. Chen. A vector angle based evolutionary algorithm for unconstrained many-objective optimization. *IEEE TEVC*, 2016 (in press).
25. Y. Yuan, H. Xu, and B. Wang. An Experimental Investigation of Variation Operators in Reference-Point Based Many-Objective Optimization. In *GECCO*, pages 775–782, 2015.
26. Y. Yuan, H. Xu, B. Wang, and X. Yao. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE TEVC*, 20(1):16–37, 2016.
27. Q. Zhang, W. Liu, and H. Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *IEEE CEC*, pages 203–208, 2009.
28. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE TEVC*, 7(2):117–132, 2003.