Eliminating Non-dominated Sorting from NSGA-III

Balija Santoshkumar[1[0000−0002−4829−148X]],
Kalyanmoy Deb[1[0000−0001−7402−9939]], and Lei Chen[2]

[1] Computational Optimization and Innovation (COIN) Laboratory, Michigan State University, East Lansing, MI 48864, USA, {balijasa, kdeb}@egr.msu.edu,
https://www.coin-lab.org

[2] Guangdong University of Technology, Guangzhou, China, chenaction@126.com

COIN Report Number 2022011

Abstract. The series of non-dominated sorting based genetic algorithms (NSGA-series) has clearly shown their niche in solving multi- and many-objective optimization problems since mid-nineties. Of them, NSGA-III was designed to solve problems having three or more objectives efficiently. It is well established that with an increase in number of objectives, an increasingly large proportion of a random population stays non-dominated, thereby making only a few population members to remain dominated. Thus, in many-objective optimization problems, the need for a non-dominated sorting (NDS) procedure is questionable, except in early generations. In support of this argument, it can also be noted that most other popular evolutionary multi- and many-objective optimization algorithms do not use the NDS procedure. In this paper, we investigate the effect of NDS procedure on the performance of NSGA-III. From simulation results on two to 10-objective problems, it is observed that an elimination of the NDS procedure from NSGA-III must accompany a penalty boundary intersection (PBI) type niching method to indirectly emphasize best non-dominated solutions. Elimination of the NDS procedure from NSGA-III will open up a number of avenues for NSGA-III to be modified for different scenarios, such as, for parallel implementations, surrogate-assisted applications, and others, more easily.

Keywords: Non-dominated sorting · multi-objective optimization · evolutionary computation · NSGA-III

1 Introduction

The first non-dominated sorting based genetic algorithm (NSGA) was proposed in 1995 [15]. It ushered in a new era of computational optimization methods for handling two-objective problems along with a few other contemporary algorithms [6, 8, 9]. In 2002, an elitist and parameter-less version of NSGA, called NSGA-II, was proposed to solve primarily two and three-objective problems [4]. Thereafter, in 2014, a reference vector based extension, called NSGA-III, was proposed to handle three and more objective problems. They all have one operation in common: non-dominated sorting (NDS) of the population based on the partial ordering of their objective vectors. The NSGA-series of procedures require that every population member to be classified into a different NDS level. To achieve the sorting procedure within a population of solutions, pairwise comparison of solutions are made with their objective vectors to identify the set of population members which are not dominated by any other population member. This set of non-dominated solutions belong to the first NDS-level. To obtain the second NDS-level members, the first NDS-level members are discounted from the population and another round of pair-wise domination check is performed. The members which do not get dominated by any remaining population members belong to the second NDS-level. This process is continued until all population members are classified into a distinct NDS level. The NSGA series of procedures were based on these sorted classes of population members and emphasized a lower NDS level to be infinitely more important than the next higher NDS level. All NSGA operations were applied by keeping the hierarchy of NDS level of population members. Thus, NDS is intricately linked to the core of NSGA series of algorithms.

It has also been established that when many-objective optimization problems (with more than three conflicting objectives) are to solved, a randomly created population contains increasingly more NDS-level one solutions. For a 10-objective problem, the number of non-dominated (ND) solutions in a random population of size 100 is about 95 [6], The argument can be extended to state that number of NDS level-two members will be significantly small compared to NDS level-one members, and so on. Thus, the effectiveness of executing the NDS procedure for many-objective problems can be questionable. Whether a solution belongs to second or third level of non-domination may not matter on the overall progress of the search algorithm, as there are not many population members exist in the dominated levels altogether. Moreover, two dominated solutions of different levels may stay close in the objective space, hence a classification of one solution to a relatively lower class and the other to a higher class may not produce any significant difference in the performance of the search algorithm.

Thus, it is worth an investigation to eliminate NDS from the NSGA-III procedure and classify the entire population into two classes: non-dominated and dominated classes. If the performance stays similar to the original NSGA-III procedure, the modified search procedure can be beneficial in a number of scenarios. First, the NDS sorting procedure takes $O(MN^2)$ [4] (where M is the number of objectives and N is the population size), which is more complex than identifying the NDS level one members ($O(MN(\log N)^{M-2})$) [13]. Second, when an EMO or EMaO algorithm is to be implemented with surrogates for handling computationally expensive problems, the NDS procedure may have to be performed on a population evaluated with a mix of high-fidelity and surrogate-assisted evaluations. In such a scenario, a classification of every population member to a precise NDS level may be an overkill, particularly since the objective values are noisy.

Having made the argument against a full effectiveness of NDS in a search algorithm, the next important question pertains to the dependencies of NSGA-III's other operators on the NDS procedure. In this paper, we eliminate the NDS procedure from NSGA-III and study the changes that must be introduced in other NSGA-III operators to bring the modified NSGA-III's performance at least at par to that of the original NSGA-III procedure.

In the rest of the paper, we present the modified NSGA-III procedure (without NDS procedure) in Section 2. Results on two to 10-objective problems of the modified NSGA-III procedure is presented and compared against the original NSGA-III and other EMaO algorithms in Section 3. Conclusions are drawn in Section 4.

2 Proposed Algorithm: N$\tilde{\text{S}}$GA-III(NSGA-III\NDS)

The basic framework of the proposed algorithm is similar to NSGA-III [7] with significant changes in the way (i) domination check is executed, (ii) the survival selection operator is modified, and (iii) a few other minor changes are adopted resulting from the change in domination check procedure.

Instead of using Das-Dennis method of creating reference vectors, we use Riesz s-Energy based method proposed in [2,3]. This allows any population size (N) to be used for any objective dimension. Like in NSGA-III [7], first, we initialize the population P_t of size N. We generate an offspring population Q_t of size N with the standard genetic operators and without care of each population member's association to any reference vector. However, the mating selection operator requires the domination status of a population member, which we describe in the next paragraph. The combined population (parent and offspring) is R_t of size $2N$. The survival selection operator then chooses N solutions from R_t and save to P_{t+1}. Iterations proceed until a termination criterion is satisfied. The overall algorithm is provided in Algorithm 1.

2.1 Classification of pop. members

We classify all the population members (R_t) into three hierarchical classes: Class 1: non-dominated and feasible solutions; Class 2: feasible and dominated solutions, and Class 3: infeasible solutions. Notice that all feasible solutions from the second ND front are combined into Class 2. Figure 1 illustrates the classification process. We use a hierarchical classification process for both mating and survival selection operators. A solution belonging to a lower class is better. Thus, for binary tournament selection in the mating selection operator above classification helps to select the better candidate.

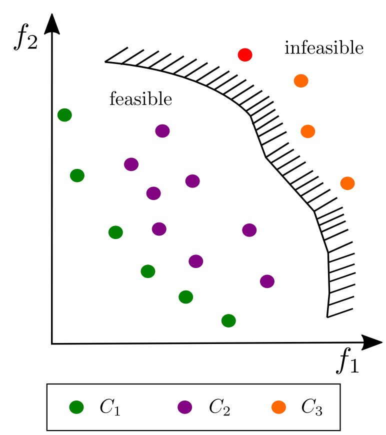


Fig. 1: Classification of population members into three hierarchical classes.

Algorithm 1: Generation t in N$\tilde{\text{S}}$GA-III=NSGA-IIInNDS

Data: Predefined set of reference directions W_r and parent population P_t
Result: P_{t+1}

$S_t = \emptyset$, no of selections remaining $(n_r) = N$
/* off spring population generation */
Q_t = Recombination + Mutation (P_t)
$R_t = P_t \cup Q_t$
I=find-non-dominated (R_t)
/* finding ideal point and nadir point for normalization of objective space */
ideal, nadir=hyperplane-boundary-estimation(R_t, I)
/* Association each s in R_t with a reference direction */
[(s); d_2(s)]=associate-to-niches(R_t, W_r, ideal, nadir)
/* (s)=closest reference direction, d_2(s)= perpendicular distance between (s) and s */
d(s) =pbi-decomposition(R_t, W_r, = 5, ideal, nadir)
/* $d(\mathrm{s}) = d_1 +\ \ d_2$ distance between (s) and s */
Set attribute to each population member non-dominated $ND = 1$ and dominated $ND = 0$
Set attributes each population member $R_t(CV; ND; d(\mathbf{s}))$
/* classifying population members */
Class 1 (C_1) : $R_t(CV = 0 \cap ND = 1)$
Class 2 (C_2) : $R_t(CV = 0 \cap ND = 0)$
Class 3 (C_3) : $R_t(CV > 0)$
/* Class 1 selection */
if *($|C_1| \leq N$)* **then**
    $S_t = S_t \cup C_1$; $n_r = N - |S_t|$
else
    $S_t; n_r$= class-survivor-selection($C_1; S_t; n_r; W_r;$ (s)$; d$(s));
    $P_{t+1} = S_t$, break
end if
/* Class 2 selection */
if $n_r > 0$ **then**
    $S_t; n_r$= class-survivor-selection($C_2; S_t; n_r; W_r;$ (s)$; d$(s));
else
    $P_{t+1} = S_t$, break
end if
/* Class 3 selection for constrained problems */
if $n_r > 0$ **then**
    S_t= tournament-selection(C_3);
else
    $P_{t+1} = S_t$, break
end if

2.2 Association of population members

Population members are normalized using the same procedure as in NSGA-III. Thereafter, each population member is associated with a particular reference vector based on the d_2 distance metric, which is the perpendicular distance from the population member's normalized objective vector to the reference line. This association principle is followed for all classes of population members.

2.3 Class-wise mating selection

In mating selection, two population members are picked at random and a winner must be selected to act as a parent for mating. A lower class member is always the winner. This allows a feasible solution to be better than infeasible solution and a non-dominated feasible solution to be better than a feasible dominated solution. But if both picked members belong to Class 3, the one with smaller overall normalized constrained violation (CV) value [6] wins. For highly constrained problems, it is likely that most population members are infeasible. In this case, preferring a smaller constraint violated solution provides a good signal to the EMO to gradually progress toward the PO front. When both solutions belong to either Class 1 or Class 2, one is randomly chosen as a parent. This disallows any competition between solutions from different classes. Since the number of reference vectors (desired number of final PO solutions) is identical to the population size (N), mating selection, involving exactly N members, must not encourage any competition among N population members.

2.4 Reference vector based niching in survival selection

In the survival selection, there are $2N$ population members and exactly N better diverse members must be chosen as the next generation's starting population. Competitions between associated feasible solutions of a reference vector may be allowed here. For this purpose, we follow a similar niching procedure as in NSGA-III and choose a single member for each reference vector.

For this purpose, all associated members of a reference vector are considered and instead of the d_2 metric, following procedure is used. Associated members are considered in a hierarchical manner based on their class (from Class 1 to Class 3 in the order). Among all the members of the best class, the best solution is chosen as follows. If the best class is 3 (meaning all infeasible associated members), the member with smallest CV is chosen. If the best class is 1 or 2, all associated Class 1 or Class 2, as the case may be, members are compared with the PBI distance metric: $d(\mathbf{s}) = d_1 + \theta d_2$, where d_1 is the distance along reference direction to origin (equivalent to the ideal point), θ (5 used here) is a parameter, and d_2 is the

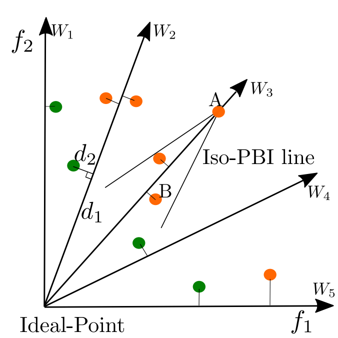


Fig. 2: Association of population members to reference directions ($\mathbf{W}$) and their distance metrics.

perpendicular distance (see Figure 2). We choose the one having smaller $d(\mathbf{s})$ as the winner. After performing the above process for all N reference vectors (note that some may not have any associated population member), the process is repeated to pick the next round of solutions. This is repeated until N members are selected. The reason for choosing the PBI metric for selection can be given from Figure 2. Since all dominated classes of solutions are clubbed together, the association principle can allow a distant member lying close to the reference vector may be judged to be better based on the d_2 metric. But the use if PBI metric makes a combined weighted distance of d_1 and d_2 and thus, even being close to the reference vector, a near ideal-point member may be judged better. When NDS was conducted in NSGA-III, member A was never allowed to compare against member B for closeness to the reference vector (as B dominates A and they will be in different non-dominated levels), hence PBI metric was not needed. But without the NDS, a distance metric with combined d_1 and d_2 is must to compare two associated population members for the same reference vector. Hence, we replace NSGA-III's d_2 vector with the PBI metric for choosing the winner.

3 Experimental Results

In this section, we present the simulation results of the proposed method N$\tilde{\text{S}}$GA-III with NSGA-III [7,12] and MOEA/D [17] on ZDT [18], BNH [1], OSY [14], SRN [15], TNK [16], DTLZ test suite [5] and WFG test suite [10] with objectives ranging from 2 to 10. To support the use of the PBI metric in the survival selection operator, we also replace it with the d_2 metric (call it N$\tilde{\text{S}}$GA-III-d_2) and compare with N$\tilde{\text{S}}$GA-III.

For each problem, we run all the algorithms 31 times with different initial population members. The population size for the 2 and 3-objective problems is set to 100, for the 5 and 8-objective-problems to 200, and for the 10-objectives problem to 300. Each run is executed for a maximum of 100,000 solution evaluations (SEs). We have used the number of reference directions the same as the population size. We have used IGD+ [11] as the performance metric, as it measures both convergence and diversity. For all algorithm, the final generation members are used to compute the IGD+ metric. We have used Wilcoxon signed-rank test with at most $p = 0.05$ to determine the best and statistically similar methods.

3.1 Unconstrained Problems

Two-objective problems: The performance metrics for ZDT problems are given in Table 1. The best performing method is marked in bold, and the other methods which are statistically similar to the best method are in marked in italics. The representative objective vectors for some representative ZDT problems are presented in Figure 3. It can be observed that N$\tilde{\text{S}}$GA-III-d_2 performs the best in three out of the six problems, despite not executing NDS. Interestingly, removal of NDS operation from NSGA-III is found to be more effective than performing the NDS operation, but the use of NDS is not found too detrimental.

The bottom-right figure shows that although in initial generations NS̃GA-III-d_2 converges slowly because of the d_2 metric selection but after a certain number of generations (around 150) it performs better.

Table 1: ZDT problems IGD+ performance metrics for 31 runs. The best performing method is in bold and the other methods which are statistically similar to the best method are in italics with Wilcoxon signed-rank test having $p = 0.05$.

Problem	M	MOEA/D-TCH	NSGA-III	NS̃GA-III-d_2	NS̃GA-III
ZDT1	2	3.9720e-3	4.2770e-3	**2.8750e-3**	3.2900e-3
ZDT2	2	**2.6940e-3**	4.2370e-3	3.0290e-3	3.7760e-3
ZDT3	2	3.2490e-3	3.1050e-3	**2.1120e-3**	*1.9180e-3*
ZDT4	2	7.4550e-3	4.0090e-3	**3.0570e-3**	3.9250e-3
ZDT5	2	8.7824e-2	**8.3809e-2**	9.1272e-2	*9.6077e-2*
ZDT6	2	2.7570e-3	**2.3190e-3**	2.3440e-3	2.3570e-3
best/similar/total →		1/0/6	2/0/6	**3**/0/6	0/2/6

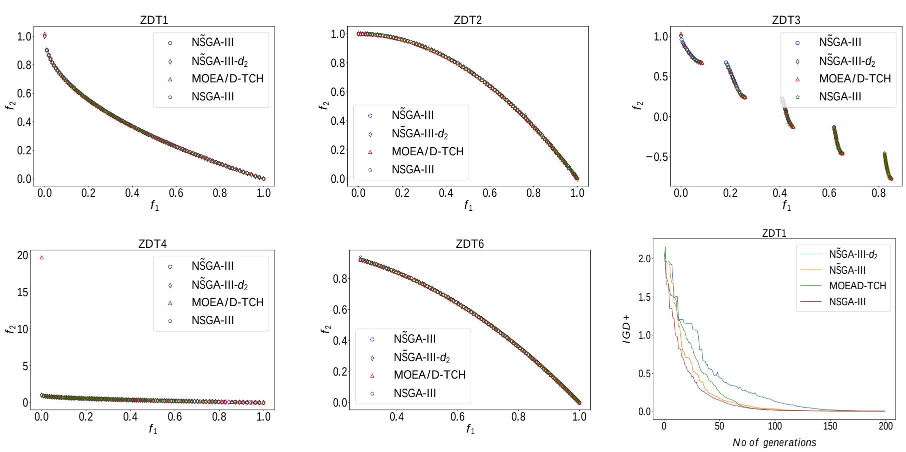


Fig. 3: Obtained solutions by all methods on some ZDT problems.

Three and many-objective problems: The number of variables for DTLZ test suite is chosen with $k = 10$ where $k = (n - M + 1)$ and for WFG we have chosen $n = 30$ with $k = 2(M - 1)$ and $k + l = n - M$. The performance metrics for DTLZ and WFG problems are presented in Tables 2 and 3, respectively. Representative solutions on some DTLZ and WFG problems are shown in Figures 4 and 5, respectively.

It is reported (and consistent with the literature) that MOEA/D performs better than NSGA-III on DTLZ problems mainly due to the similarly-scaled objective values for all objectives. MOEA/D reports all ND solutions from the final generation as an outcome, while NSGA-III reports a single best population member for each active reference line from the final generation. While the number of reported solutions from both these algorithms are more or less identical for DTLZ2 type of problems, MOEA/D will report more ND solutions for DTLZ5 (degenerate PO front dimension or constrained problems) than NSGA-III. It is important to highlight that NSGA-III's final population may have more ND solutions than its reported number of solutions, but one solution per reference vector is reported to provide a widely distributed set of ND solutions. The IGD+ values shown for DTLZ5 and DTLZ6 (degenerate problems) in Table 2 with brackets are the IGD+ values computed by taking only PBI-metric associated solutions for each active reference vectors instead of taking all final ND members. Comparative IGD+ values with N$\tilde{\text{S}}$GA-III are now observed.

It is also clear from Table 3 that MOEA/D does not work well on WFG problems, mainly due to non-uniform scaling of objectives in these problems. N$\tilde{\text{S}}$GA-III works better than all other methods on WFG problems (26 best of 36 problems) and the combined DTLZ and WFG problems (36 best of 64 problems), followed by N$\tilde{\text{S}}$GA-III-d_2 method (14 best of 64 problems). Thus, it is interesting to conclude from the two tables that NDS was not a very important operation for NSGA-III for solving three and many-objective problems. While in two-objective problems N$\tilde{\text{S}}$GA-III-d_2 works better, for three and many-objective problems, N$\tilde{\text{S}}$GA-III works much better with the PBI metric, rather than d_2 metric.

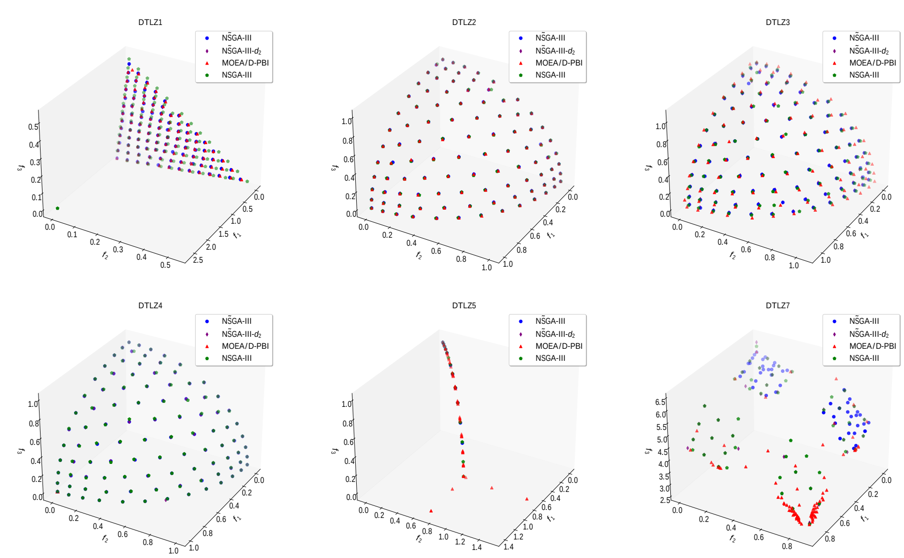


Fig. 4: Obtained solutions by all methods on some DTLZ problems.

With more objectives, the proportion of non-dominated members in a finite population becomes more [6]. It also likely that at later generations, most ref-

Table 2: IGD+ performance metrics for 31 runs on DTLZ problems.

Problem	M	MOEA/D-PBI	NSGA-III	N$\tilde{S}$GA-III-d_2	N$\tilde{S}$GA-III
DTLZ1	3	3.4964e-2	3.3075e-2	**2.7376e-2**	3.0468e-2
	5	1.3433e-2	2.4264e-2	2.3384e-2	**9.8980e-3**
	8	3.5368e-2	6.8785e-2	4.4221e-2	**2.0986e-2**
	10	8.0652e-2	1.0557e-1	7.2443e-2	**5.3484e-2**
DTLZ2	3	2.6866e-2	2.6114e-2	**2.5251e-2**	2.5400e-2
	5	4.8030e-3	4.2260e-3	*2.8530e-3*	**2.5850e-3**
	8	9.5400e-3	1.0530e-2	1.1239e-2	**7.5900e-3**
	10	**1.1309e-2**	3.0248e-2	2.9952e-2	1.7584e-2
DTLZ3	3	4.1858e-2	3.5138e-2	**2.9578e-2**	3.1881e-2
	5	2.6342e-2	6.3278e-2	6.0382e-2	**1.9904e-2**
	8	6.5717e-1	1.4607e-1	9.8674e-2	**3.9972e-2**
	10	*7.6384e-1*	2.4370e-1	2.2173e-1	**8.2418e-2**
DTLZ4	3	2.3108e-1	2.5742e-2	**2.5200e-2**	2.5293e-2
	5	1.2806e-1	7.1360e-3	**2.4350e-3**	4.7470e-3
	8	1.6967e-1	1.2310e-2	**8.4650e-3**	1.4287e-2
	10	1.9387e-1	*2.2136e-2*	**2.2127e-2**	2.4918e-2
DTLZ5	3	**1.0821e-2** (1.9361e-2)	1.8919e-2	8.3213e-2	7.3286e-2
	5	**5.4780e-3** (4.3105e-2)	1.7890e-1	1.7765e-1	5.6062e-2
	8	**1.4013e-2** (3.6413e-1)	3.7251e-1	3.2480e-1	9.8550e-2
	10	**2.0389e-2** (3.7156e-1)	3.9726e-1	3.0289e-1	8.0261e-2
DTLZ6	3	**1.0750e-2** (1.8531e-2)	1.9141e-2	2.4254e-2	1.9901e-2
	5	**6.8450e-3** (3.2681e-2)	2.3767e+0	2.2249e+0	8.4766e-2
	8	**1.3850e-2** (3.6379e-1)	4.1552e+0	4.0039e+0	5.5816e-1
	10	**2.0448e-2** (3.7158e-1)	5.7369e+0	5.5902e+0	1.8406e+0
DTLZ7	3	5.9116e-2	**3.3175e-2**	3.5356e-2	3.5393e-2
	5	1.3871e-1	1.1134e-1	1.1303e-1	**1.0425e-1**
	8	1.2302e+0	1.8461e-1	1.8533e-1	**1.7900e-1**
	10	1.6129e+0	*1.8760e-1*	**1.8816e-1**	*1.8847e-1*
best/similar/total →		9/1/28	1/2/28	8/1/28	**10**/1/28

erence vectors will have a single associated member, particularly for problems having every reference vector leading to a distinct PO solution. In such cases, the use of d_2 or PBI metric may not matter much. However, early on, this may not be the case and the difference between d_2 and PBI metric may show up. If N$\tilde{S}$GA-III and N$\tilde{S}$GA-III-d_2 IGD+ metric values are compared for the DTLZ5 problem having a few active reference vectors leading to a PO solution, many associated population members are expected for each of the active reference vectors. The performance of N$\tilde{S}$GA-III is better than N$\tilde{S}$GA-III-d_2. To support this argument, we plot the variation of IGD+ value versus generations in Figure 6 for five-objective DTLZ2 and DTLZ5 problems. It can observed that while the performance of all three NSGA-III methods are more or less the same (with a slight edge for N$\tilde{S}$GA-III), for DTLZ5, N$\tilde{S}$GA-III performs the best.

3.2 Constrained Problems

Next, we apply all three NSGA-III methods to constrained problems. Since MOEA/D is not usually used for constrained problems, we ignore it here. Our N$\tilde{S}$GA-III method includes constraint violation as Class 3 solutions and are well-equipped to solve constrained problems.

Two-objective problems: First, we consider two-objective test problems: BNH, OSY, SRN and TNK [6]. Results are presented in Table 4. It is clear

Table 3: IGD+ performance metrics for 31 runs on WFG problems.

Problem	M	MOEA/D-PBI	NSGA-III	N$\tilde{S}$GA-III-d_2	N$\tilde{S}$GA-III
WFG1	3	4.3446e-1	4.1909e-1	3.7651e-1	**3.6318e-1**
	5	4.5475e-1	5.1397e-1	4.8717e-1	**4.7428e-1**
	8	**4.5695e-1**	5.4412e-1	5.0332e-1	4.8024e-1
	10	4.8536e-1	4.8279e-1	4.3790e-1	**4.1891e-1**
WFG2	3	7.7834e-2	2.7083e-2	1.6556e-2	**1.3154e-2**
	5	1.3579e-1	4.4112e-2	4.4695e-2	**3.0848e-2**
	8	1.5377e-1	4.4840e-2	4.5205e-2	**3.2744e-2**
	10	1.6877e-1	*2.8194e-2*	**2.6693e-2**	2.9477e-2
WFG3	3	1.6463e-1	4.7558e-2	6.1073e-2	**4.4320e-2**
	5	2.3672e+0	3.5530e-1	2.5779e-1	**2.0911e-1**
	8	6.5851e+1	1.9058e+0	4.9633e-1	**3.7363e-1**
	10	5.8560e+2	2.8201e+1	4.3064e+0	**3.5071e+0**
WFG4	3	5.9218e-2	4.5232e-2	3.0611e-2	**2.9883e-2**
	5	2.8312e-1	9.9648e-2	**8.8945e-2**	*8.9776e-2*
	8	7.1960e-1	1.5102e-1	*1.4690e-1*	**1.4659e-1**
	10	7.4449e-1	3.4704e-1	*3.1872e-1*	**3.1720e-1**
WFG5	3	6.2305e-2	5.6573e-2	*4.4542e-2*	**4.4398e-2**
	5	1.8382e-1	8.3703e-2	**7.5884e-2**	*7.6385e-2*
	8	8.3541e-1	1.7123e-1	*1.6990e-1*	**1.6945e-1**
	10	1.9304e+0	1.8594e-1	*1.8349e-1*	**1.8225e-1**
WFG6	3	6.5468e-2	5.4180e-2	*4.0747e-2*	**3.9925e-2**
	5	2.7583e-1	7.6170e-2	*6.5350e-2*	**6.4881e-2**
	8	6.5052e-1	1.2266e-1	**1.1600e-1**	*1.1656e-1*
	10	7.7136e-1	1.5115e-1	*1.4879e-1*	**1.4749e-1**
WFG7	3	6.9492e-2	3.8074e-2	2.7420e-2	**2.7256e-2**
	5	2.2374e-1	5.6599e-2	**4.8126e-2**	*4.8975e-2*
	8	6.3725e-1	1.1505e-1	*1.1071e-1*	**1.1030e-1**
	10	7.5967e-1	1.5021e-1	1.4735e-1	**1.4209e-1**
WFG8	3	8.8971e-2	7.3944e-2	5.9074e-2	**5.8133e-2**
	5	2.4557e-1	1.2901e-1	*1.2277e-1*	**1.2269e-1**
	8	8.6388e-1	**2.0446e-1**	2.1285e-1	*2.0935e-1*
	10	1.0092e+0	**2.1857e-1**	2.5549e-1	2.5674e-1
WFG9	3	9.1279e-2	6.4961e-2	*4.9177e-2*	**4.6800e-2**
	5	1.8287e-1	1.1333e-1	*1.1344e-1*	**1.0975e-1**
	8	5.2175e-1	**1.5261e-1**	1.6783e-1	*1.5377e-1*
	10	7.2420e-1	*1.9624e-1*	**1.9613e-1**	*1.9981e-1*
best/similar /total →		1/0/36	3/2/36	6/12/36	**26**/7/36
DTLZ + WFG →		10/1/64	4/4/64	14/13/64	**36**/8/64

that N$\tilde{S}$GA-III-d_2 performs the best for the two-objective problems, as in the case of unconstrained two-objective problems, shown in Table 1.

Three and many-objective problems: Table 5 presents the results on three and many-objective (5, 8 and 10-obj.) constrained optimization problems. It is clear that both NSGA-III versions without NDS operation works better than the original NSGA-III, with a slight edge for N$\tilde{S}$GA-III (10 best out of 20 problems), followed by N$\tilde{S}$GA-III-d_2 (8 best out of 20 problems).

Figure 7 shows the performance of all NSGA-III versions on the C2DTLZ2 problem with 10-objectives, showing similar distributions, but convergence by N$\tilde{S}$GA-III is slightly better (see also Table 5).

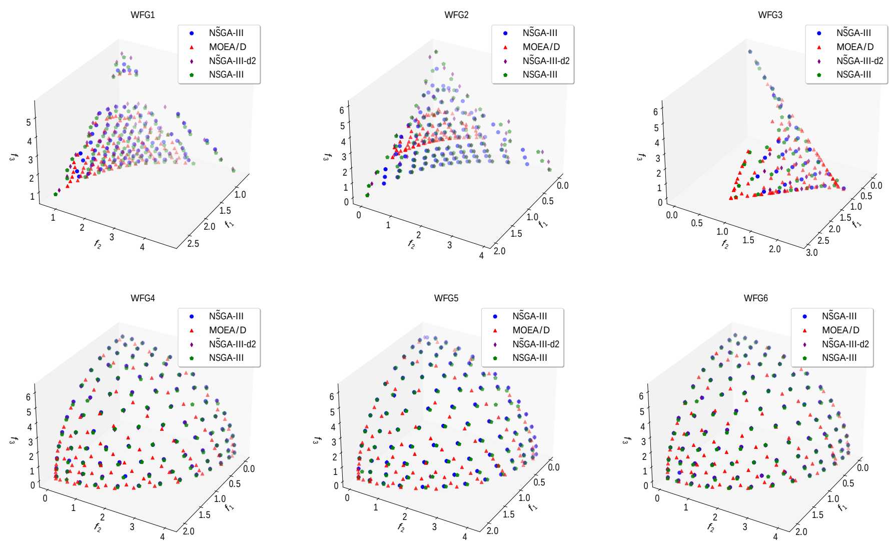


Fig. 5: Obtained solutions by all methods on some WFG problems.

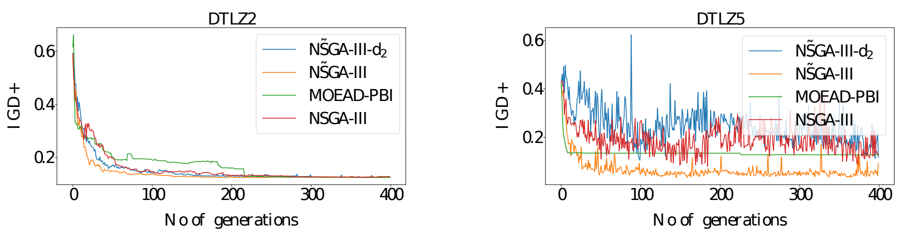


Fig. 6: Convergence history of algorithms on DTLZ2-5obj and DTLZ5-5obj problems. NS̃GA-III produces better IGD+ values.

4 Conclusions

This paper has questioned the use of non-dominated sorting (NDS) operation in NSGA-III method for solving two to 10-objective problems. Since population members are not divided into different non-dominated levels for performing mating and survival selection operators, the choice of an appropriate solution within the associated members for a reference direction becomes important. We have investigated two approaches: (i) NS̃GA-III-d_2, which uses the original orthogonal distance metric d_2 and (ii) NS̃GA-III, which uses the well-known PBI metric. Based on 87 different problems, following two conclusions can be made:

– The NDS operation is not absolutely necessary and for many-objective problems, NSGA-III without NDS performs better than the original NSGA-III.
– For two objective problems, NS̃GA-III with the d_2-metric has a slow progress in the beginning, but can catch up with the performance of NS̃GA-III or the original NSGA-III with enough generations.

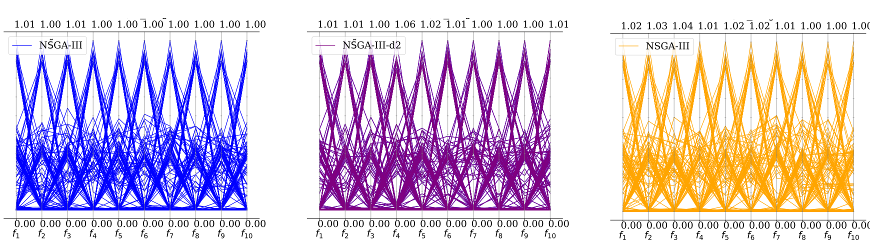
Table 4: IGD+ performance metrics for 31 runs on two-objective constrained test problems.

Problem	M	NSGA-III	NS̃GA-III-d_2	NS̃GA-III
BNH	2	**2.8870e-3**	2.9970e-3	5.4130e-3
OSY	2	2.4169e-2	**4.7080e-3**	9.3770e-3
SRN	2	*2.9650e-3*	3.1560e-3	**2.9390e-3**
TNK	2	4.3210e-3	**3.5450e-3**	5.1930e-3
best/similar /total →		1/1/4	**2**/0/4	1/0/4

Table 5: IGD+ performance metrics for 31 runs on three and many-objective constrained problems.

Problem	M	NSGA-III	NS̃GA-III-d_2	NS̃GA-III
C1DTLZ1	3	3.8573e-2	**2.3388e-2**	3.7922e-2
	5	6.3775e-2	*5.5396e-2*	**4.7830e-2**
	8	*8.0516e-2*	*8.3178e-2*	**7.7365e-2**
	10	**1.2233e-1**	*1.2774e-1*	*1.3327e-1*
C1DTLZ3	3	8.0129e+0	**8.0080e+0**	8.0109e+0
	5	1.1587e+1	1.1583e+1	**1.1570e+1**
	8	1.1677e+1	1.1673e+1	**1.1620e+1**
	10	1.1726e+1	1.1721e+1	**1.1665e+1**
C2DTLZ2	3	1.8170e-3	**3.1800e-4**	8.0800e-4
	5	3.8650e-3	**2.1900e-3**	3.1260e-3
	8	5.6720e-3	5.0310e-3	**1.7410e-3**
	10	1.3350e-2	1.2366e-2	**5.4420e-3**
C3DTLZ1	3	**4.5512e-1**	*5.8319e-1*	5.8754e-1
	5	5.3615e-1	5.3443e-1	**5.2738e-1**
	8	5.8136e-1	5.7790e-1	**5.4731e-1**
	10	5.6870e-1	5.5841e-1	**4.7341e-1**
C3DTLZ4	3	9.7930e-3	**3.3400e-3**	9.3730e-3
	5	*1.9092e-2*	**1.7264e-2**	2.0938e-2
	8	*2.5991e-2*	**2.5469e-2**	2.8631e-2
	10	*3.4809e-2*	**3.4504e-2**	*3.7774e-2*
best/similar/total →		2/4/20	8/4/20	**10**/2/20

These observations are important for making NSGA-III more computationally efficient. Since NDS operation is not essential, domination check can be completed with a smaller computational time. Moreover, since convergence rate is faster for NS̃GA-III, it can be used with more effectiveness to build better surrogate-assisted NSGA-III methods with a limited number of solution evaluations. We plan to pursue some of these extensions in the near future.



(a) NS̃GA-III (b) NS̃GA-III-d_2 (c) NSGA-III

Fig. 7: Obtained solutions using PCP plot for CDTLZ2 problem.

References

1. Binh, T.T., Korn, U.: MOBES: A multiobjective evolution strategy for constrained optimization problems. In: In Proceedings Of The Third International Conference On Genetic Algorithmsc(MENDEL97). pp. 176–182 (1997)
2. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020). https://doi.org/10.1109/access.2020.2990567
3. Blank, J., Deb, K., Dhebar, Y., Bandaru, S., Seada, H.: Generating well-spaced points on a unit simplex for evolutionary many-objective optimization. IEEE Transactions on Evolutionary Computation **25**(1), 48–60 (feb 2021). https://doi.org/10.1109/tevc.2020.2992387
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (apr 2002). https://doi.org/10.1109/4235.996017
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600). IEEE. https://doi.org/10.1109/cec.2002.1007032
6. Deb, K.: Multi-objective optimization using evolutionary algorithms. wiley, London (2001)
7. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (aug 2014). https://doi.org/10.1109/tevc.2013.2281535
8. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms. pp. 416–423. San Mateo, CA: Morgan Kaufmann (1993)
9. Horn, J., Nafpliotis, N., Goldberg, D.: A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. IEEE. https://doi.org/10.1109/icec.1994.350037
10. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Lecture Notes in Computer Science, pp. 280–295. Springer Berlin Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_20
11. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Modified distance calculation in generational distance and inverted generational distance. In: Lecture Notes in Computer Science, pp. 110–125. Springer International Publishing (2015). https://doi.org/10.1007/978-3-319-15892-1_8
12. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. IEEE Transactions on Evolutionary Computation **18**(4), 602–622 (aug 2014). https://doi.org/10.1109/tevc.2013.2281534
13. Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. Journal of the ACM **22**(4), 469–476 (oct 1975). https://doi.org/10.1145/321906.321910
14. Osyczka, A., Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. Structural optimization **10**(2), 94–99 (1995)

15. Srinivas, N., Deb, K.: Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation **2**(3), 221–248 (sep 1994). https://doi.org/10.1162/evco.1994.2.3.221
16. Tanaka, M., Watanabe, H., Furukawa, Y., Tanino, T.: GA-based decision support system for multicriteria optimization. In: 1995 IEEE international conference on systems, man and cybernetics. Intelligent systems for the 21st century. vol. 2, pp. 1556–1561. IEEE (1995)
17. Zhang, Q., Li, H.: MOEA/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (dec 2007). https://doi.org/10.1109/tevc.2007.892759
18. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2), 173–195 (jun 2000). https://doi.org/10.1162/106365600568202