A Running Performance Metric and Termination Criterion for Evaluating Evolutionary Multi- and Many-objective Optimization Algorithms

Julian Blank
Computer Science and Engineering
Michigan State University
East Lansing, United States
blankjul@egr.msu.edu

Kalyanmoy Deb
Electrical and Computer Engineering
Michigan State University
East Lansing, United States
kdeb@egr.msu.edu

***Abstract*—Researchers have spent a considerable effort in evaluating the goodness of a solution set obtained by an evolutionary multi-objective algorithm. However, most performance metrics assume that the knowledge of the exact Pareto-optimal set is available. Also, most metrics evaluate an algorithm's performance based on the final solution set, which fails to capture their performance during intermediate generations. In this paper, we investigate a running performance metric which can be applied to measure the performance at any time during the algorithm execution and no true optimum needs to be known for computing the metric. In general, multi-objective algorithms either improve the convergence based on the dominance relation or the diversity in the solution set. Our proposed running metric makes use of this fact by keeping track of the indicators regarding the extreme points and the ND solution set each generation and derives measures of convergence and diversity. Moreover, by introducing a threshold and comparing the values of indicators a set of termination criteria is also suggested. Finally, we demonstrate how our running performance metric can be used to compare multiple evolutionary multi-objective algorithms with each other. An implementation of the proposed methodology is available at *pymoo*, a multi-objective optimization framework: https://pymoo.org.**

***Index Terms*—Multi-objective Optimization, Performance Indicator, Running Metric.**

I. Introduction

In multi-objective optimization, the convergence and diversity of a *set* of non-dominated (ND) solutions to the true Pareto-optimal (PO) front must be considered in defining a performance metric. To accomplish a fair comparison of two solution sets various performance metrics have been proposed in the past [4], [12], [14], [16]. There are at least two issues with the past performance comparison studies. First, in most EMO studies, the performance of an EMO algorithm was computed only with the final ND set. Such a metric does not provide how the algorithm is able to come up with the final performance – "Was the algorithm gradually and consistently improving its performance from start to the end of a run?", or "Were there sudden spurts of improvements with a long stagnation?". While comparing two or more algorithms, the comparison of the final ND set does not provide many vital dynamics of each algorithm – "Did one algorithm perform better in early generations and then slowed down towards the end?", or "Did one algorithm outperform another from the start to the end?". A static evaluation of algorithms using the final ND points does not reveal any of these important characteristics of them. This calls for a running performance metric, which can provide a generation-wise performance profile so that a more detailed understanding of the algorithm's performance or a relative performance of two or more algorithms can be achieved.

The second issue with most performance metrics is that they require the knowledge of the true PO front. For example, the Inverse Generational Distance (IGD) metric [4] and its extension IGD+ [12] require a reference set of solutions from the true PO front. Clearly, such metrics cannot be applied to a real-world problem for which PO solutions are not available before an EMO is run. The popularly used hypervolume [17] metric requires a reference point (preferably a point close to the *nadir* point). Again, without knowing the PO front, the nadir point information is not available. Many studies have shown that the hypervolume metric value largely depends on the chosen reference point [11]. Due to the lack of knowledge of PO solutions, researchers still use the hypervolume metric with a questionable outcome.

In this paper, we address this vital issue and seek ways to update the above-mentioned metrics without any use of the true PO front and extend them to be used as any-time performance metrics for multi-objective optimization. Besides the description of the running IGD metric, we also propose a set of termination criteria based on convergence and stability of extreme points in the evolving ND set and diversity of ND solutions. We demonstrate the working of the running metric and proposed termination criteria on two-objective ZDT problems (including ZDT5) and three to eight-objective DTLZ problems. We also describe how the proposed running IGD metric and termination criteria can be used to compare two or more EMO algorithms based on the entire history of evolution from start to end of multiple optimization runs and not on the basis of the final ND set only. We argue that such a dynamic running metric plot provides more information about

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

the characteristics of different algorithms on specific problems.

In the remainder of this paper, we provide a brief summary of existing running metrics for multi-objective optimization in Section II. The development of the proposed running metric is provided next in Section III. Results on a number of two to eight-objective problems and on two real-world problems are provided in Section IV. Finally, conclusions of this study are made in Section V.

II. Related Work

A running performance metric for MOEAs was first proposed in [6]. It showed the need to consider not only the final ND set of solutions for evaluating an EMO algorithm's performance but also intermediate solution sets during an optimization run. The authors then suggested to calculate the average normalized distance to the closest PO point for each ND solution in a generation in order to measure the level of convergence. Moreover, a diversity measure based on grid-wise entropy in the objective space was proposed. Both metrics together were used to keep track of generation-wise dynamics of an EMO. These metrics suggested in 2002 assumed that the knowledge of the true PO front is available.

III. Proposed Methodology

A performance metric for evaluating an EMO algorithm must consider the following aspects: (a) handle differently scaled objectives, (b) emphasize a solution set converging as close as possible to the true PO front, and (c) emphasize the diversity of the ND set in the objective space [3]. These aspects need to be addressed by a performance metric, whether it is a running metric for evaluating an EMO's performance at any generation or a static metric which is applied only to the final ND set. While the normalization issue can be addressed easily due to the existence of a set, the convergence and diversity issues require special attention. Depending on the algorithm and characteristics of optimization problems, the above two phases for convergence and diversity might be clearly separable, alternating, or interwoven [15]. Recognizing the above, we refer to the convergence phase to extreme solutions in an EMO run with $\mathbb{C}^E$ and the diversity creating phase with $\mathbb{C}^D$. The convergence phase $\mathbb{C}^E$ is characterized by having found the (optimal) extreme points which is necessary to ensure suitable normalization for improving the diversity.

A. Convergence to Extreme Points

During the convergence phase $\mathbb{C}^E$, an EMO algorithm continually discovers new solutions that dominate existing solutions from one generation to the next. Presumably, this takes place in the early generations of an optimization run. Therefore, the current set of ND solutions gets significantly changed from one generation to the other. A measure of the change can be derived from the movement of the extreme points of the current ND solutions in the objective space. The extreme points in a multi-objective context can be defined with two points: realized ideal point $\boldsymbol{z}^*$ which is a vector constructed with the minimum of all objectives and is usually a non-existent solution, and a realized nadir point $\boldsymbol{z}^{\text{nad}}$ which is constructed with the maximum of ND solutions in the objective space. We refer to a realized ideal or nadir point at generation t by $\boldsymbol{z}^*(t)$ and $\boldsymbol{z}^{\text{nad}}(t)$, respectively. Our method keeps track of the movement of these two extreme points from one generation to the next by considering the maximum absolute difference of each component. For the realized ideal point, the normalized change from $(t-1)$-th to t-th generation is given by:

$$\Delta_{t-1,\,t}\ \boldsymbol{z}^* = \max_{i=1}^{M} \frac{z_i^*(t-1) - z_i^*(t)}{z_i^{\text{nad}}(t) - z_i^*(t)}. \tag{1}$$

By definition it is guaranteed that $z_i^*(t-1) \geq z_i^*(t)$ and $z_i^{\text{nad}}(t) \geq z_i^*(t)$. Therefore, the nominator and denominator are guaranteed to be equal or greater than zero. If the denominator equals to zero, we neglect normalization for the i-th component. For the nadir point z_i^{nad}, the normalized change is defined analogously:

$$\Delta_{t-1,\,t}\ \boldsymbol{z}^{\text{nad}} = \max_{i=1}^{M} \frac{z_i^{\text{nad}}(t-1) - z_i^{\text{nad}}(t)}{z_i^{\text{nad}}(t) - z_i^*(t)}. \tag{2}$$

The proposed metrics, $\Delta_{t-1,t}\ \boldsymbol{z}^*$and $\Delta_{t-1,t}\ \boldsymbol{z}^{\text{nad}}$, are illustrated in Figure 1. ND solutions found in generation

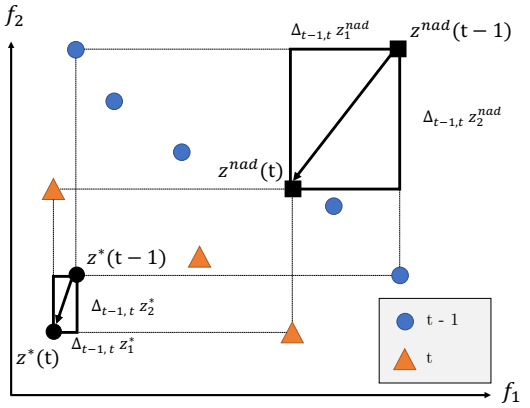


Fig. 1: A visualization of the realized ideal $\boldsymbol{z}^*$ and nadir points $\boldsymbol{z}^{\text{nad}}$ in a transition from generation t (triangles) to $t+1$ (circles) during the convergence phase $\mathbb{C}^E$.

$t-1$ are represented by blue circles and in generation t by orange triangles. The black arrows illustrate the ideal and nadir points movement from one generation to another. In the figure, $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$ have moved significantly which is also indicated by large values of $\Delta_{t-1,t}\ \boldsymbol{z}^*$ and $\Delta_{t-1,t}\ \boldsymbol{z}^{\text{nad}}$. Such a scenario is typical for the first few generations of an EMO algorithm. It is also important to highlight that since $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$ are unstable during the $\mathbb{C}^E$ phase, the normalization of the objectives is unstable. Thus, any running metric will also produce an incoherent measure of an EMO's performance from one generation to the next, particularly when the knowledge of theoretical ideal and nadir points are not available.

However, in a later generation a relatively small $\Delta_{t-1,t}\ \boldsymbol{z}^*$ and $\Delta_{t-1,t}\ \boldsymbol{z}^{\text{nad}}$ will imply that the boundary points have

started to settle down. This will indicate that the extreme points of the ND solution set are not moving significantly anymore and, therefore, the algorithm has achieved its convergence phase $\mathbb{C}^E$ by anchoring the extreme points and is now ready to concentrate on maximizing the diversity of intermediate ND solutions.

B. Enhancing Diversity

When the algorithm enters the diversity phase $\mathbb{C}^D$, the movement of the extreme points is insignificant. From this generation on, the normalization of the objectives will also stabilize. A stable normalization for the $\mathbb{C}^D$ phase is important, since the diversity measure must calculate the distance between ND solutions involving all objectives. However, during the $\mathbb{C}^D$ phase, it is still expected that the boundary points will change slightly. To make a more reliable computation of a diversity metric, we propose a cumulative approach. We accumulate the ND solutions from the initial generation to the current generation (τ) and calculate the realized $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$ points from the accumulated set. Then, we determine the following normalized i-th objective value of j-th ND point at t-th generation $P_i^{(j)}(t)$ using $P(\tau)$ as follows ($0 \leq t \leq \tau$):

$$\bar{P}_i^{\tau,(j)}(t) = \frac{P_i^{(j)}(t) - z_i^*(\tau)}{z_i^{\text{nad}}(\tau) - z_i^*(\tau)}. \tag{3}$$

Note that $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$ points are calculated at the τ-th generation.

Now, that the ND sets at generations $0 \leq t \leq \tau$ are normalized with fixed $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$ points, any existing performance metric requiring a reference set $P^* = \bar{P}^\tau(\tau)$ and an evolving ND set $Q(t) = \bar{P}^\tau(t)$ can be computed to evaluate the algorithm's performance for the above-mentioned generations. For example, the Inverted Generational Distance (IGD) metric is a popularly used metric for this purpose, which in its existing sense, requires the exact knowledge of the true PO front P^*:

$$\text{IGD}(Q(t), P^*) = \frac{1}{|P^*|} \sum_{i=1}^{|P^*|} \left(\min_{j=1}^{|Q(t)|} \left\| P_i^* - Q_j(t) \right\| \right). \tag{4}$$

It measures the average distance from a solution in P^* to the closest solution in $Q(t)$ obtained by the algorithm at generation t, for given τ. However, our above proposal does not require the knowledge of true PO front, but the above IGD metric can be computed as $\text{IGD}(\bar{P}^\tau(t), \bar{P}^\tau(\tau))$ for $0 \leq t \leq \tau$.

To reduce the computational complexity of re-normalization and IGD computations, we follow a simple procedure for our $\mathbb{C}^D$ measure. The average improvement of the IGD metric from generation $(t-1)$ to t is computed as follows:

$$\varnothing_t = \text{IGD}(\bar{P}^t(t-1), \bar{P}^t(t)). \tag{5}$$

Note that, the normalization is performed with ideal and nadir points computed at generation t.

In Figure 2, a solution set in generation $(t-1)$ and generation t during the diversity phase $\mathbb{C}^D$ is illustrated. First, it is worth noting that the realized ideal and nadir points $\boldsymbol{z}^*$ and

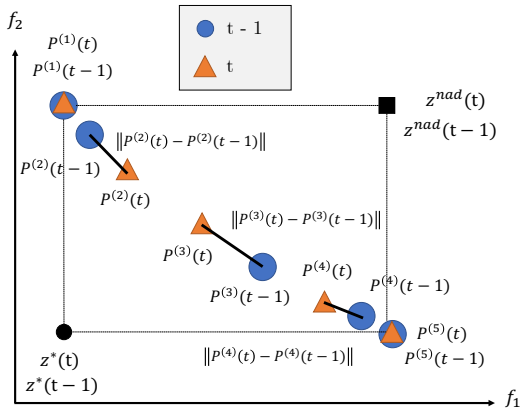


Fig. 2: A visualization of the average change in IGD values ($\varnothing_t$) from generation $t-1$ to t during the diversity phase $\mathbb{C}^D$.

$\boldsymbol{z}^{\text{nad}}$ are unaltered during the transition between generations. Second, the diversity has improved which is obvious by observing that the solution set in generation $(t-1)$ is biased towards the boundary points $P^{(1)}(t-1)$ and $P^{(5)}(t-1)$ and has only a single point $P^{(3)}(t-1)$ at the center. The metric $\varnothing_t$ is calculated by considering for each point in $P(t)$ with its closest point in $P(t-1)$. For the boundary points the distance turns out to be zero because both points exist in $P(t-1)$ and $P(t)$. For the non-boundary points the distances $||P^{(2)}(t) - P^{(2)}(t-1)||$, $||P^{(3)}(t) - P^{(3)}(t-1)||$ and $||P^{(4)}(t) - P^{(4)}(t-1)||$ are summed up and the resulting value divided by the size of ND set at generation t, $|P(t)| = 5$.

C. Termination Criterion

A termination criterion must be set based on an algorithm's satisfactory performance up until the generation. The proposed metrics $\Delta_{t-1,t}$ $\boldsymbol{z}^*$, $\Delta_{t-1,t}$ $\boldsymbol{z}^{\text{nad}}$ and $\varnothing_t$ are capable of providing the information whether the EMO algorithm is in the convergence phase, diversity creation phase or in a phase which does not seem to change the current status of ND solutions. For estimating the status for termination of any of the above scenarios, we propose to use a sliding window (ω generations) to compute the metrics and check against a pre-specified threshold value (ϵ) for termination. While $\Delta_{t-1,t}$ $\boldsymbol{z}^*$ and $\Delta_{t-1,t}$ $\boldsymbol{z}^{\text{nad}}$ checks are done at every generation and the respective $\mathbb{C}^E$ completion is identified, the $\mathbb{C}^D$ completion is checked after the first ω generations are over and then at every generation thereafter. The $\varnothing_t$ metric can be computed for two consecutive generations for the past ω generations. If all ω changes are below or equal to ϵ, the $\mathbb{C}^D$ completion is declared. When both convergence and diversity completions are made, the EMO algorithm is terminated. Figure 3 illustrates an exemplary decision for termination of an EMO.

In this example the algorithm is run up to six generations and, therefore, five transitions exist. The termination is based on a window size of $\omega = 3$ which corresponds to the most recent transitions $3 \rightarrow 4$, $4 \rightarrow 5$, and $5 \rightarrow 6$ (highlighted in green). All values from earlier transitions (namely, $1 \rightarrow 2$ and $2 \rightarrow 3$) are not of interest for current calculations. To

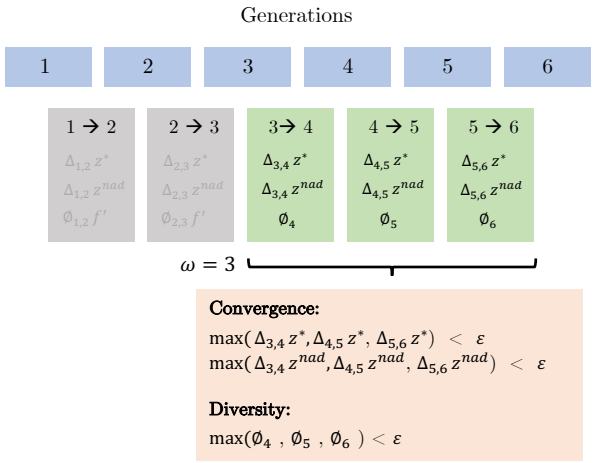


Fig. 3: Termination criterion after six generations ($\tau = 6$) with a sliding window size of three ($\omega = 3$).

check whether the algorithm has completed the *convergence* phase $\mathbb{C}^E$, we verify if $\Delta_{t-1,t}\, \boldsymbol{z}^*$ and $\Delta_{t-1,t}\, \boldsymbol{z}^{\text{nad}}$ have not changed significantly in the past ω generations by comparing their maximum metric values with a threshold ϵ. If *all* previous values are below ϵ, we consider the algorithm has completed $\mathbb{C}^E$. This clearly indicates the impact of a suitable choice of ω and ϵ, because they directly decide if the algorithm has completed $\mathbb{C}^E$ or not. If the algorithm is found to have completed its convergence phase, we further compare the transition of $\varnothing_t$ with ϵ to determine if the diversity creation phase is completed or not. If all past ω values of $\varnothing_t$ do not exceed the threshold ϵ, we declare that the algorithm has additionally completed $\mathbb{C}^D$. The completion of $\mathbb{C}^E$ and $\mathbb{C}^D$ phases implies that the algorithm can be terminated.

D. Visualizing the Running Performance Metric

In a real-world optimization scenario the objective function can be computationally expensive and an optimization run might take a couple of hours or days to complete. Therefore, the continuous visualization of an algorithm's performance is a practical need during an optimization run. This means, instead of the IGD or hypervolume metric values at the final generation, the metrics must be presented generation-wise from the start until the end of an optimization run. In this paper, we call such a metric a running performance metric.

For such a scenario, the computational overhead to present a meaningful visualization at the end of each generation can be demanding, but we argue that in practical problems, such computational overhead will still be negligible compared to the evaluation time of a population of solutions. Even without knowing the true PO front, we propose a novel way to compute the $\mathbb{C}^D$ metric as described below.

For visualization, we re-consider the interval of τ generations, at which the performance plot has to be updated. For an easy implementation, τ can be initialized to ω (the sliding window used for termination condition in the previous subsection). From the start of an EMO run, the user waits until the first τ generations are completed. The ND sets of all past generations are collected to compute the realized ideal and nadir points and all ND sets are normalized using Equation 3. Then, the following IGD metric

$$\varnothing_t^\tau = \text{IGD}(\bar{P}^\tau(t), \bar{P}^\tau(\tau)) \tag{6}$$

is computed for all past generations ($0 \leq t \leq \tau$). For $\tau = \omega = 5$, such a plot is shown in Figure 4 with a blue line. Although it requires re-normalization of ND sets in all previous generations, but it ensures a strictly improving performance of an algorithm. While the algorithm is running

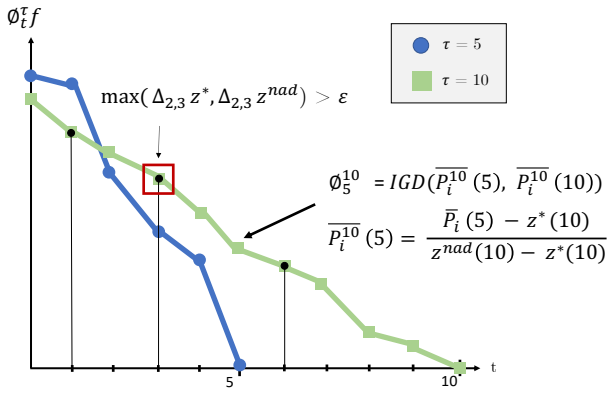
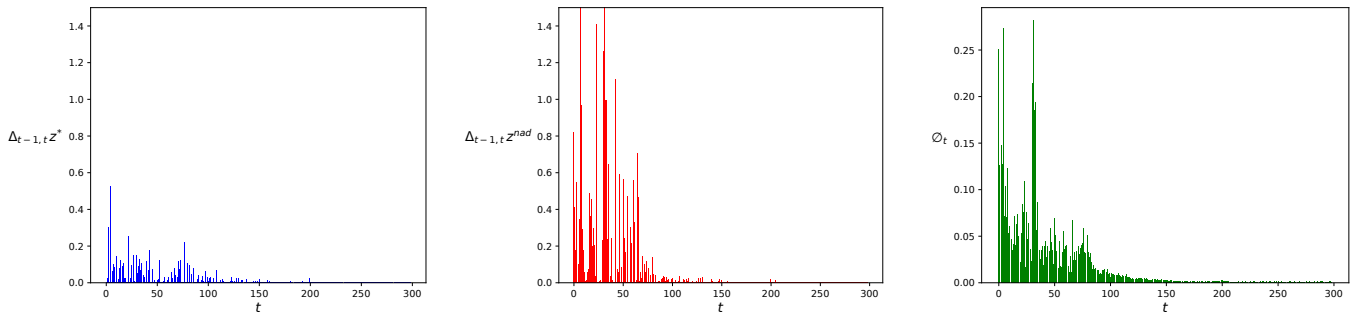


Fig. 4: Visualization of the running performance metric $\varnothing_t^\tau$ at generations $\tau = 5$ and 10.

in the background, the user does not see any plot until $\tau = 5$ generations are elapsed. At generation 5, the plot for $\varnothing_t^5 = \text{IGD}(\bar{P}^\tau(t), \bar{P}^\tau(5))$ (blue line) appears, showing the performance of the EMO for the past $\tau = 5$ generations. To compute this plot, all six ND sets (from generation $t = 0$ to 5) are normalized using the ideal and nadir vectors computed at generation 5 and then IGD values are computed. The plot shows that IGD is steadily improving with generation. The fact that IGD at $t = 5$ is zero is not surpsiring, as the reference set $\bar{P}^\tau(\tau)$ for the blue IGD line is the normalized ND set at generation 5 or $\bar{P}^\tau(5)$ itself.

Then, the algorithm continues and ideally a new IGD line can be created using a re-normalization of all seven ND sets (generations zero to six) and computing IGD with a reference set $\bar{P}^\tau(6)$. But, as mentioned above, this can be time-consuming to re-normalize all ND sets from start, particularly when hundreds of generations have elapsed. Instead of recomputing IGD at every generation, we can stagger τ by another ω generations, so that $\tau \leftarrow \tau + \omega$. The figure shows the next IGD line at $\tau = 10$ in green color. All ND sets are collected until generation 10 and normalized. Thereafter, $\varnothing_t^{10}$ IGD$(\bar{P}^\tau(t), \bar{P}^\tau(10))$ is computed for generations ($0 \leq t \leq 10$). To paint a picture of the progress of the algorithm from generation 5 to 10, both blue and green lines can be shown for the user to comprehend, while the next ω generations can be continued in the background. For not making the IGD plots cluttered, only two most current IGD plots can be shown at a time.

With the above visualization scheme, the termination of a run still takes place using the low computational approach de-



(a) $\Delta_{t-1,t}\ \boldsymbol{z}^*$ (b) $\Delta_{t-1,t}\ \boldsymbol{z}^{\text{nad}}$ (c) $\varnothing_t$=IGD($\bar{P}^t(t-1), \bar{P}^t(t)$)

Fig. 5: Analysis of the proposed indicators during a run of NSGA-II on ZDT6 in 300 generations.

scribed in the previous section and the EMO can be terminated by making a final IGD plot at the terminating generation.

In general, $\varnothing_{t,\tau}$ can be computed with another multi-objective performance indicator, such as hypervolume [17] metric. However, since hypervolume becomes too computationally expensive for higher objectives, we do not pursue it here.

IV. Results

In the following, the behavior of the proposed running performance metric and the proposed termination criteria are tested on a number of test problems and two real-world problems. We have used the state-of-the-art evolutionary multi-objective optimization algorithms: NSGA-II [7] for problems with two objectives and NSGA-III [9] for problems with three or more objectives. All hyper-parameters, such as population size and other evolutionary parameters are set based on the standard recommendations [7], [9].

A. Convergence Behavior

First, we investigate the settlement of the ideal and nadir points during an optimization run using our proposed $\mathbb{C}^E$ metric, given in Equation 3. We choose a representative bi-objective test problem ZDT6 [18] for this purpose. In Figure 5a, the change of the ideal point ($\boldsymbol{z}^*$) and in Figure 5b the change of the nadir point ($\boldsymbol{z}^{\text{nad}}$) is shown.

Clearly, the changes of ideal points are less frequent and significant compared to the changes of the nadir points. This is additionally supported by similar observations made for NSGA-III in [2]. Both figures indicate that the boundary points can be considered to have settled only after 100 generations are elapsed. In Figure 5c, the generational movement $\varnothing_t$ is presented at each generation. A set of ω consecutive values is checked with a pre-specified threshold parameter ϵ for terminating an EMO run. IGD values are computed using two consecutive ND sets and, therefore, no knowledge about the true PO frontier is required.

In Figure 6, IGD values with a normalization achieved with the final generation at $\tau = 300$ are shown. The variation of IGD is more smooth compared to that in Figure 5c. Although a more stable and more reliable termination is possible using the $\varnothing_t^\tau$ metric, but clearly it is a computationally demanding, as all 300 ND sets (one at each generation) must be normalized using the final population extreme values at generation $\tau = 300$.

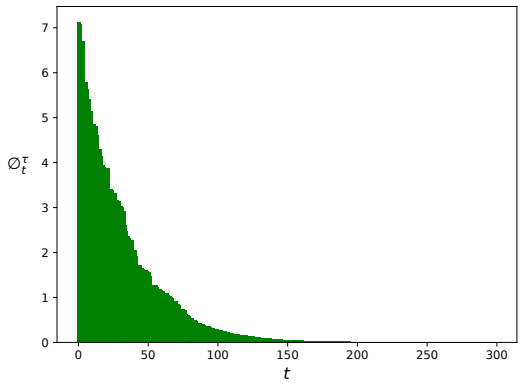


Fig. 6: The running IGD metric $\varnothing_t^\tau$ with $\tau = 300$ on ZDT6.

B. Visualization of Running IGD Metric

The above four figures provided a clear picture of the $\mathbb{C}^E$ completion and $\mathbb{C}^D$ completion for terminating an EMO run on ZDT6 using NSGA-II. We now demonstrate how the running metric can be used to visualize the performance of NSGA-II on ZDT6 as the optimization proceeds. Figure 7 shows the proceedings with $\omega = 5$. The first IGD plot (with $\tau = 5$) on the top figure appears after five generations are elapsed showing a monotonic decrease in IGD values. Then, after five more generations, the next IGD plot (with $\tau = 10$) is shown. Since all 11 ND sets (including the initial set) are re-normalized with 10-th generation ideal and nadir points, the IGD values in the first five generations are now different from that of the previous IGD plot. The IGD at $\tau = 10$ indicates that NSGA-II is able to produce better ND solutions compared to those at 5-th generation. At generation 15, a new IGD plot appears indicating that during generation 11-14, ND sets have not improved much due to almost horizontal nature of the IGD plot. But, at generation 15, a drastic improvement occurred. It is also interesting that for this specific NSGA-II run, at generation 18 a fast drop of IGD has also occurred, as shown in the generation-20 plot, indicating that NSGA-II

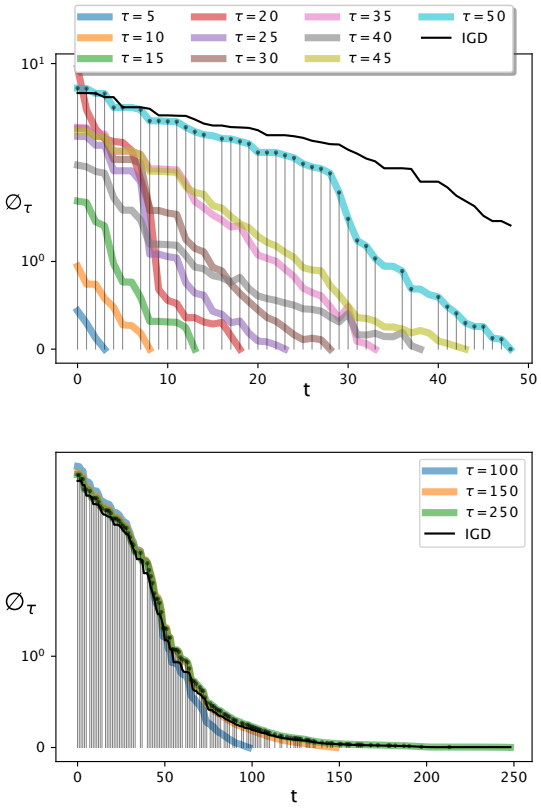


Fig. 7: Convergence of NSGA-II on ZDT6 during an early and a latter phase of the algorithm.

is able to find a much-improved set of extreme points within 16-20 generations. Such dynamics of evolution of ND points gets clear by observing the changes in the running IGD plots. IGD plots at $t = 35$ and 40 indicate that IGD values get worse at around generation 30. This may have happened due to change of ideal or nadir points. For comparison, the true IGD values computed using the true PO points are shown by a black line (marked 'IGD'). The proceedings of the latter generations are shown in bottom part of Figure 7. The true IGD variation is also shown in this figure. The IGD computed without the knowledge of the true PO points is consistent with that calculated with PO points.

C. Termination of NSGA-II Runs

Next, we demonstrate how the proposed $\mathbb{C}^E$ and $\mathbb{C}^D$ completion procedures can be used to terminate an EMO run. We consider all six ZDT problems and set the sliding window size to $\omega = 30$. Thus, the earliest an EMO run can be terminated is at generation 30. According to the procedure described in Subsection III-C, termination criteria are checked at every generation thereafter. To reduce the computational burden, here, we check the termination criteria at every 5-th generation after the first 30 generations by using the past 30 ND sets. We also set the threshold value for termination to $\epsilon = 0.0025$. This implies that an algorithm is considered to be terminated if in the last ω generational transitions, no average difference in the objective space regarding $\Delta_{t-1,t}\ \boldsymbol{z}^*$, $\Delta_{t-1,t}\ \boldsymbol{z}^{\text{nad}}$ and $\varnothing_t$ is above 0.25%. Table I presents the results of NSGA-II with a population size of 100 on the ZDT test problem suite.

TABLE I: Generations at which termination criteria are met on six bi-objective ZDT test problems using NSGA-II based on 51 runs.

Problem	μ_{gen}	σ_{gen}	μ_{IGD}	σ_{IGD}
ZDT1	170.98	23.9378	0.006	0.0007
ZDT2	176.57	15.2804	0.006	0.0008
ZDT3	165.59	18.9907	0.007	0.0078
ZDT4	244.51	28.6401	0.006	0.0013
ZDT5	91.27	59.0834	1.482	1.0608
ZDT6	248.33	10.8474	0.004	0.0002

The table presents (i) μ_{Gen}, which is the average number of generations until termination occurred, (ii) σ_{Gen} which is the standard deviation of terminated generations over 51 runs, (iii) μ_{IGD}, which is the average IGD value of the final population at the terminated generation computed using true PO points as reference points, and (iv) σ_{IGD} which is the standard deviation of IGD values over 51 runs. It can be observed that for ZDT1-4 and ZDT6, the final IGD values are small, indicating that an excellent performance is achieved by NSGA-II over 51 runs. Interestingly, the termination criteria based on an IGD metric which does not use the true PO points is able to produce similar true IGD values dictated by the true PO points. Such an outcome is possible with a threshold of $\epsilon = 0.0025$ on $\varnothing_t$ over $\omega = 30$ generations and is a remarkable achievement. Also, σ_{Gen} in each run is rather small, which indicates that NSGA-II was terminated reliably over multiple runs. A relative comparison of μ_{gen} reveals that ZDT4 and ZDT6 problems require more generations to achieve the desired convergence and diversity compared to ZDT1-3 problems. This was also established earlier [5].

EMO studies have mostly avoided ZDT5, which is defined over a Boolean search space with many deceptive optima. The table indicates that NSGA-II optimizing ZDT5 terminates rather quickly (on an average 91.27 generations), but with a significantly large true IGD value. This indicates that NSGA-II has got stuck in *deceptive* fronts (which are far away from the global PO front in the objective space) early on and could not recover from it before termination.

D. Many-objective Test Problems

Next, we investigate if our proposed running performance metric can be applied suitably to indicate the performance of evolutionary many-objective optimization algorithms on problems having more than two objectives. To investigate this aspect, we apply NSGA-III [9] on the DTLZ test problem suite [10]. For our experiments, we have considered DTLZ1 to DTLZ4 with $M = 3$, $M = 5$ and $M = 8$ objectives. Analogously to the bi-objective experiments, Table II shows the effect of automatic termination using the proposed criteria.

TABLE II: Generations at which termination criteria are met on four many-objective DTLZ test problems using NSGA-III based on 51 runs. N is the population size.

Problem	M	N	μ_{gen}	σ_{gen}	μ_{IGD}	σ_{IGD}
DTLZ1	3	92	305.2	62.20	0.004	0.0026
	5	212	410.39	70.31	0.003	0.0015
	8	156	689.31	174.36	0.003	0.0017
DTLZ2	3	92	191.67	54.72	0.003	0.0016
	5	212	395.98	128.53	0.005	0.0031
	8	156	733.04	164.11	0.004	0.0018
DTLZ3	3	92	549.41	89.02	0.100	0.3393
	5	212	672.75	97.31	0.030	0.0261
	8	156	1143.53	213.60	0.018	0.0119
DTLZ4	3	92	234.69	88.57	0.003	0.0022
	5	212	343.82	102.79	0.006	0.0037
	8	156	414.9	117.88	0.003	0.0018

As expected, the required number of generations to satisfy the strict termination criteria increases with in increase in the number of objectives M. Moreover, the σ_{gen} is also significantly larger. Interestingly, the chosen value of ω makes the true IGD values come closer to the chosen threshold ϵ on all problems, except in DTLZ3. The variation of IGD values over 51 runs is also small, indicating a reliable performance of NSGA-III in solving these problems. For DTLZ3, a similar behavior than for ZDT5 can be observed due to the multimodal nature of the problem. NSGA-III gets stuck in a local PO front in some runs, thereby making the IGD value to be large when computed using the true PO points. A follow-up study using $\omega = 40$ reveals a better performance on DTLZ3. Due to space restrictions, we do not show the results here.

E. Comparison of Multiple EMO Algorithms

Finally, we propose a way to compare two or more EMO algorithms problems using the proposed running IGD performance metric.

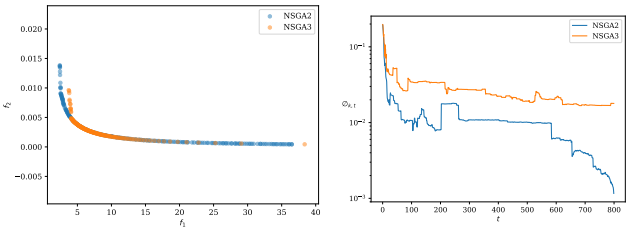
Each algorithm is run with a termination criterion (here, we show results after a pre-defined number of generations (τ) for a fixed population size is completed). Then, all final ND solutions are merged and dominated solutions, if any, are deleted from the merged set. We refer to this filtered set as Q^*. Individual ND sets from each algorithm and Q^* are normalized using the ideal and nadir points of Q^* and, then, the running IGD metric is computed for the normalized ND set $\bar{P}^\tau(k,t)$ for the k-th algorithm at generation t:

$$\varnothing_{k,t} = \text{IGD}(\bar{P}^\tau(k,t), \bar{Q}^*). \quad (7)$$

However, our proposed termination criteria with pre-defined ω and ϵ can also be used to terminate each algorithm.

A more detailed comparison of dynamics of algorithms can be achieved by applying the above procedure after every τ generations by merging ND sets together, removing dominated solutions, and normalizing the sets using the merged set. Then, a visualization of the running IGD metric at generation τ can be updated with a revised IGD at 2τ generations and so on, until all algorithms are terminated based on our proposed termination criteria.

To demonstrate algorithmic comparison, we choose two real-world problems – Welded Beam design problem (WELD) [8] with two objectives and Carside Impact design problem (CAR) [13] with three objectives. We compare the performance of two algorithms: NSGA-II with NSGA-III. For WELD, the final solutions in the objective space are shown in Figure 8a. It can be observed from $\varnothing_{k,t}$ variations that NSGA-II performs significantly better than NSGA-III on WELD. The results show that NSGA-III is not able to find the



(a) Objective points (b) Running IGD metric $\varnothing_{k,t}$

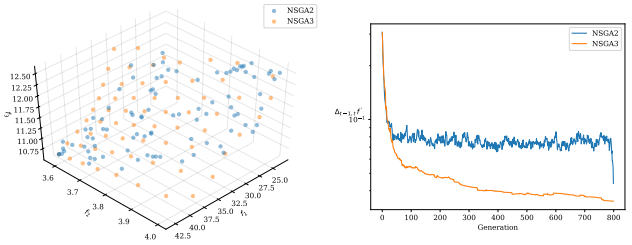
Fig. 8: Results on WELD problem (4 variables, 4 constraints, 2 objectives).

boundary point that minimizes f_1 and was not able to cover the lower right part in the objective space uniformly. While most studies in the past have compared the final ND points among competing algorithms, here we show the use of a running performance metric to have a more comprehensive evaluation.

Figure 8b shows that the running IGD metric ($\varnothing_t$) for NSGA-II is better right from the start and stays better throughout the generations. It was never the case that NSGA-III worked better in any intermediate generation during the runs. Since the cumulative ND sets from both algorithms are used as the reference set at every generation t and IGD of each algorithm at generation $(t-1)$ is computed for the figure, it is clear that NSGA-II sets populated the reference sets for it to produce better running IGD values. Moreover, the fact that the running IGD values gets flattened for NSGA-III in later generations indicates its lack of improvements. NSGA-II keeps on improving its ND points and the IGD value approaches zero. Since the merged set has a few points from NSGA-III (such as the extreme f_2 point) which NSGA-II sets do not have, the running IGD metric value for NSGA-II does not reach zero, as they would with the running IGD metric in a standalone run.

Next, we apply our approach to the CAR problem. It has seven variables, ten constraints, and three objectives. Due to the fact that NSGA-III was originally proposed as an improvement of NSGA-II for handling more than two objectives, a superior performance of NSGA-III is expected. Figure 9a shows the solution sets returned by each algorithm in the objective space. NSGA-III is able to achieve a better distribution of points. Figure 9 clearly shows that NSGA-III has outperformed NSGA-II on this problem throughout a run. The running IGD metric oscillates around an almost fixed IGD

value, except at the last few generations where the performance gets better suddenly. On the other hand, NSGA-III is able to exhibit a steady improvement in the IGD value from the start to the end of the run.



(a) Objective points (b) Running IGD metric $\varnothing_{k,t}$

Fig. 9: Results on CAR problem (7 variables, 10 constraints, 3 objectives).

An implementation of the running performance metric and the termination criterion are publicly available in our multi-optimization framework *pymoo* [1].

V. Conclusions

In this paper, we have addressed the performance evaluation of EMO algorithms during an optimization run. The main advantage of our proposed running metric is that it does not require the knowledge of the true PO points. By comparing the ND points of a generation with the ND points of the next generation as a reference set, a running IGD metric is proposed. Moreover, by estimating the convergence of extreme points in consecutive generations and by comparing the running IGD metric value in a few consecutive generations with a pre-specified threshold, systematic termination criteria are proposed. Thus, this paper has suggested a performance metric that (i) does not require any knowledge of the true PO front, (ii) provides a generation-wise performance measure, and (iii) is capable of terminating an EMO algorithm based on its demonstrated performance thus far. Results on two to eight-objective problems indicate that the proposed method is able to produce similar (running) IGD values by approximation instead of requiring the knowledge of the true PO front. The paper has also demonstrated a way to compare two or more EMO algorithms using the proposed running IGD metric by combining the ND sets from multiple algorithms and using it as a reference set for running IGD computations. Results have been shown on two real-world problems in which NSGA-II performs better on two-objective problem, while NSGA-III performs better on three-objective problem.

We plan to extend this study by investigating the algorithm comparison in more detail. The comparison requires to extract an approximated Pareto front from multiple solutions sets. We have used a naive approach by merging all solution sets together and obtaining a set of ND solutions. However, more accurate predictions regarding performance could be achieved by considering the diversity of the solution set and optimizing this subset selection problem more carefully. Also, we plan to perform a hyper-parameter optimization on the window size ω used for the termination criterion. Finally, the running performance metric concept implemented here with the IGD metric will be extended for other existing metrics, such as hypervolume and IGD+. Nevertheless, the development of the running IGD metric without any use of true PO points, its use as a termination condition, and its use in comparing multiple algorithms stays as a significant contribution for future EMO studies.

References

[1] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 2020. DOI: 10.1109/ACCESS.2020.2990567.

[2] Julian Blank, Kalyanmoy Deb, and Proteek Chandan Roy. Investigating the normalization procedure of NSGA-III. In *Evolutionary Multi-Criterion Optimization - 10th International Conference, EMO-2019, Proceedings*, pages 229–240, 2019.

[3] Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiundefinedski, editors. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008.

[4] Carlos A. Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Humberto Sossa, editors, *MICAI 2004: Advances in Artificial Intelligence*, pages 688–697, Berlin, Heidelberg, 2004. Springer Berlin.

[5] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230, 1999.

[6] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL-02)*, pages 13–20, 2002.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[8] Kalyanmoy Deb. Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, 29(11):2013–2015, Nov 1991.

[9] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. on Evolutionary Computation*, 18(4):577–601, 2014.

[10] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, 2005.

[11] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation*, 26(3), 2018.

[12] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization*, pages 110–125, Cham, 2015. Springer International Publishing.

[13] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. on Evolutionary Computation*, 18(4):602–622, 2014.

[14] N. Riquelme, C. Von Lücken, and B. Baran. Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, pages 1–11, Oct 2015.

[15] H. Seada, M. Abouhawwash, and K. Deb. Multiphase balance of diversity and convergence in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 23(3):503–513, June 2019.

[16] David A. Van Veldhuizen and David A. Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. Technical report, Evolutionary Computation, 1999.

[17] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 292–301, 1998.

[18] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.