Open access • Proceedings Article • DOI:10.1109/CEC.2012.6256519

Handling many-objective problems using an improved NSGA-II procedure

— Source link

Kalyanmoy Deb, Himanshu Jain

Institutions: Indian Institute of Technology Kanpur

Published on: 10 Jun 2012 - Congress on Evolutionary Computation

Topics: Population and Optimization problem

WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia
IEEE CEC

Handling Many-Objective Problems Using an Improved NSGA-II Procedure

Kalyanmoy Deb, IEEE Fellow and Himanshu Jain
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
PIN 208016, U.P., India
Email: {deb,hjain}@iitk.ac.in

***Abstract*—Handling many-objective problems is one of the primary concerns to EMO researchers. In this paper, we discuss a number of viable directions for developing a potential EMO algorithm for many-objective optimization problems. Thereafter, we suggest a reference-point based many-objective NSGA-II (or MO-NSGA-II) that emphasizes population members which are non-dominated yet close to a set of well-distributed reference points. The proposed MO-NSGA-II is applied to a number of many-objective test problems having three to 10 objectives (constrained and unconstrained) and compared with a recently suggested EMO algorithm (MOEA/D). The results reveal difficulties of MOEA/D in solving large-sized and differently-scaled problems, whereas MO-NSGA-II is reported to show a desirable performance on all test-problems used in this study. Further investigations are needed to test MO-NSGA-II's full potential.**

I. Introduction

Many-objective optimization is one of the main research activities in EMO, mainly due to the fact that most existing EMO algorithms are not efficient in handling many (more than three or four) objectives. Both the domination principle and concept of maintenance of diversity – the two factors that made EMO popular in late nineties and in the beginning of 2000s – are questionable for solving many-objective problems. Most real-world problems involve many objectives, however due to lack of suitable algorithms they are down-sized to two or three objectives and solved. Thus, despite the need for solving many-objective optimization problems in practice and the challenges offered due to large dimensionality of many-objective optimization problems, the research in this direction has been so far quite lukewarm. However, some efforts in this direction are providing valuable insights [1], [2], [3], [4], [5], [6].

In this paper, we outline some viable directions for developing an efficient EMO algorithm for many-objective optimization problems. Following one of the principles, we then propose a many-objective NSGA-II (MO-NSGA-II) that uses a predefined set of reference points and systematically emphasizes population members that are non-dominated and close to each reference point. The number of obtained trade-off points at the end depends on the number of chosen reference points and hence the issues of dimensionality and a large proportion of non-dominated solutions are somewhat alleviated. Results on unconstrained and constrained problems up to 10 objectives show encouraging results and suggest its further application to more complex problems.

II. Many-Objective Problems

We discuss difficulties that an EMO algorithm usually face in handling many objectives and investigate if EMO algorithms are useful at all in handling a large number of objectives.

A. Difficulties in Handling Many Objectives

It has been discussed elsewhere [7], [8] that the current state-of-the-art EMO algorithms that work under the principle of domination may face following difficulties:
1. A large fraction of population are non-dominated: It is well-known [9] that with an increase in number of objectives, an increasingly larger fraction of a randomly generated population becomes non-dominated. Since most EMO algorithms emphasize non-dominated solutions in a population, in handling many-objective problems, there are not much room for creating new solutions in a generation. This slows down the search process and therefore the overall EMO algorithm becomes inefficient.
2. Evaluation of diversity measure becomes computationally expensive: To determine the extent of crowding of solutions, the identification of neighbors becomes computationally expensive in a large-dimensional space. Any compromise or approximation in diversity estimate to make the computations faster may cause an unacceptable distribution of solutions at the end.
3. Recombination operation may be inefficient: In a many-objective problem, if only a handful of solutions are to be found, the solutions are likely to be widely separated from each other. In a population having distant solutions, the effect of recombination operator (which is considered as a key search operator in an EMO) becomes questionable. Thus, although parent solutions may be close to the Pareto-optimal front, their offspring solutions need not be near the front.
4. Representation of trade-off surface is difficult: To represent a higher dimensional trade-off surface, exponentially more points are needed. Thus, the population size needed to represent the Pareto-optimal front becomes large. Although with the computing hardware available today a large population can be used in an EMO, it is certainly difficult for a decision-maker to consider a large number of trade-off points.

5. Visualization is difficult: Finally, although it is not a matter for optimization, eventually a visualization of a higher-dimensional trade-off front may be difficult for many-objective problems.

The first two difficulties can only be alleviated by major modifications to the existing EMO methodologies. The third difficulty can be taken care of by the use of a mating restriction scheme or special recombination scheme (such as SBX with a large distribution index [10]). The fourth and fifth difficulties are common to all many-objective optimization problems and are of concerns to decision-making tasks usually followed after a set of Pareto-optimal solutions are found.

B. Are EMO Hopeless for Many Objectives?

The above descriptions may provide an impression that EMO algorithms may not be useful for handling many-objective optimization problems. But we suggest different remedies for the above difficulties.

First, existing EMO algorithms may still be useful in finding a preferred subset of solutions from the complete Pareto-optimal set. Although the preferred subset will still be many-dimensional, since the targeted solutions are focused in a small region on the Pareto-optimal front, most of the above difficulties will be alleviated by this principle [11].

Second, many problems in practice, albeit having many objectives, often degenerate to result in a low-dimensional Pareto-optimal front [7], [3], [12]. In such problems, identification of redundant objectives can be integrated with an EMO to find the Pareto-optimal front that is low-dimensional.

Third, despite the use of existing EMO techniques to the above two scenarios, the principle of EMO methodologies can be modified in at least the following two ways to solve generic many-objective optimization problems.

1. Use of special domination principle: The first difficulty mentioned above can be alleviated by using a special domination principle that will adaptively discretize the Pareto-optimal front and find a well-distributed set of points. For example, the use of ϵ-domination principle [13], [14] will make all points within ϵ distance from a Pareto-optimal points ϵ-dominated and hence the process will generate into a finite number of Pareto-optimal points as target. Such a consideration will also alleviate the second difficulty of diversity preservation.

2. Use of predefined multiple targeted search: This principle directly addresses the second difficulty mentioned above. Instead of adaptively creating a diverse set of solutions, multiple predefined targeted searches can be set by the algorithm. Since optimal points are found corresponding to each of the targeted search task, the first difficulty of dealing with a large non-dominated set does not arise. Recombination issue can be alleviated by using a mating restriction scheme in which two solutions from neighboring targets are participated in the recombination operation. There are at least two ways to implement the predefined multiple targeted search idea:

(a) A set of predefined search directions spanning the entire range of Pareto-optimal front can be specified beforehand and multiple searches can be performed along each direction. Since the search directions are widely distributed, the obtained optimal points are also likely to be widely distributed on the Pareto-optimal front in most problems. To ensure the true range of the Pareto-optimal set, ideal and nadir points [15] must be known. However, if the objectives are somehow normalized to take non-negative values (a trick often used in multi-objective optimization studies), a uniformly distributed set of search directions on the first quadrant can be used. Recently proposed MOEA/D procedure [2] uses this concept.

(b) Instead of multiple search directions, multiple predefined reference points can be specified for this purpose. Thereafter, an achievement scalarizing function (or other MCDM techniques) formed at each reference point can be solved to find set of widely distributed set of Pareto-optimal points. One such implementation was proposed recently [6] and this paper suggests another approach following this principle.

MOEA/D [2] uses a predefined set of weight vectors to maintain a diverse set of trade-off solutions. To start with, every population member (with a size same as the number of weight vectors) is associated with a weight vector randomly. Thereafter, two solutions from neighboring weight vectors are mated and an offspring solution is created. The offspring is then associated with one or more weight vectors based on a performance metric that uses either the Tchebyshev measure or a penalized distance measure from the ideal point. An external population maintains the non-dominated solutions. The first two difficulties mentioned above are negotiated by using an explicit set of weight vectors to find points and the third difficulty is alleviated by using a mating restriction scheme. However, MOEA/D requires two parameters – penalty parameter and a niching parameter defining the extent of neighborhood – that must be set right. Moreover, authors of MOEA/D have not suggested any efficient procedure for handling constraints using MOEA/D, but a penalty based approach exists [16]

A recent study [6] extended the NSGA-II procedure to suggest a hybrid NSGA-II (HN algorithm) for handling three and four-objective problems. Combined population members are projected on a hyper-plane and a clustering operation is performed on the hyper-plane to select a desired number of clusters (user-defined). Thereafter, based on the diversity of the population, either a local search operation on a random cluster member is used to move the solution closer to the Pareto-optimal front or a diversity enhancement operator is used to choose population members from all clusters. Since no targeted and distributed search is used, the approach is more generic than that in MOEA/D or even the procedure suggested in this paper. However, the efficiency of HN algorithm for problems having more than four objectives is yet to be investigated to suggest its use for many-objective problems.

III. Proposed Algorithm: MO-NSGA-II

The basic framework of the proposed many-objective NSGA-II (or MO-NSGA-II) remains similar to the original NSGA-II algorithm [17] with significant changes in its selection mechanism. We first present a brief description of the original NSGA-II algorithm.

Let us consider t-th generation of NSGA-II algorithm. Suppose parent population at this generation is P_t and its size is N, while the child population is Q_t which will also have N members. The first step is to choose the best N members from the combined parent and child population $R_t = P_t \cup Q_t$ (of size $2N$), thus preserving the elite population members. To achieve this, first, the combined population R_t is sorted according to different non-domination levels (F_1, F_2 and so on). Then, each non-domination level is selected one at a time, starting from F_1, till no further level can be included without increasing the population size of P_{t+1}. Let us say the last level included is the l-th level. Thus, all solutions from level $(l+1)$ onwards are neglected. In most situations, the last accepted level (l-th level) cannot be completely included. In such a case, only those solutions are kept that will maximize the diversity in chosen solutions. In NSGA-II, this is achieved through a computationally quick niche-preservation operator which computes the *crowding distance* for every last level member as the summation of objective-wise distance between two neighboring solutions. Thereafter, the solutions having larger crowding distance values are chosen.

We now discuss the modifications made on the NSGA-II procedure. As mentioned before, MO-NSGA-II uses a predefined set of reference points to ensure diversity in obtained solutions. For M objectives, Das and Dennis [18] suggested a systematic procedure of creating a set of reference points on a hyper-plane that lies on the first quadrant and makes equal angle with each objective axis. The intercept on each axis is at one and p divisions are considered along each objective. The total number of reference points is given by:

$$H = \binom{M+p-1}{p}. \tag{1}$$

For a three-objective problem ($M = 3$), the reference points are created on a triangle with apex at $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. If four divisions ($p = 4$) are chosen for each objective axis, $H = \binom{6}{4}$ or 15 reference points will be created. In the proposed MO-NSGA-II, in addition to emphasizing non-dominated solutions, we then emphasize population members which are *closest* to each of these reference points, thereby ensuring a wide diversity among obtained solutions. We describe the procedure in the following paragraphs.

First, all population members from non-dominated front level 1 to level $(l-1)$ are included in P_{t+1}. For choosing the remaining $k = N - \sum_{i=1}^{l-1} F_i$ population members from the last front F_l, all population members from level 1 to level l are considered to constitute the set S_t and following steps are used:

1. Creation of hyper-plane: The ideal point of S_t is determined by identifying minimum value ($\bar{f}_i$) of each objective function and by constructing the ideal point $\bar{z} = (\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_M)$. Each objective value of S_t is then translated by subtracting objective f_i by $\bar{f}_i$, so that the ideal point of translated S_t becomes a zero vector. We denote this translated objective as $f_i'(\mathbf{x}) = f_i(\mathbf{x}) - \bar{f}_i$. Thereafter, the extreme point in each objective axis (with weight vector $\mathbf{w}$ being the axis direction) is identified by finding the solution ($\mathbf{x} \in S_t$) that makes the following achievement scalarizing function minimum:

$$ASF(\mathbf{x}, \mathbf{w}) = \max_{j=1}^{M} f_i'(\mathbf{x})/w_i, \quad \text{for } \mathbf{x} \in S_t. \tag{2}$$

For $w_i = 0$, we replace it with 10^{-6}. For each objective direction, this will result in an extreme solution. These M extreme solutions are then used to constitute a hyper-plane, which is then extended to reach the translated objective axes. The reference points (H of them) are then re-calculated at this hyper-plane. Since the hyper-plane is constructed at each generation from S_t population members, the proposed MO-NSGA-II procedure adaptively builds a suitable hyper-planeb at every generation. This enables MO-NSGA-II to solve problems having a Pareto-optimal front whose objective values may be differently scaled. The ideal and extreme points are updated if a better ideal or extreme point is found.

2. Emphasize less-crowded solutions: To determine an extent of crowding of solutions (S_t) near each reference point, we project all solutions on the created hyper-plane, as shown in Figure 1 and each projected solution is associated with a reference point that is closest to the projected solution. Thus, for each reference point, there will be a cluster of solutions that are associated with it. With H reference points, ideally a population of size N should have an average $\rho_{\text{ideal}} = N/H$ solutions per reference point. If any cluster around a reference point has less solutions than ρ_{ideal}, the reference point is said to be *deficient* and is assigned a deficiency count equal to $(\rho_{\text{ideal}} - \rho_{\text{actual}})$, where ρ_{actual} is the actual count of projected points associated with the reference point. Thereafter, the most deficient reference point is chosen (tie is broken by choosing a random cluster) and the solution in F_l that has minimum ASF value with respect to the most deficient reference point is selected. The deficiency count of this reference point is then reduced by one and the next most deficient reference point is chosen and the closest solution is identified. This process is continued till all k members from F_l are chosen to fill up N members of P_{t+1}. As a final consideration, the cluster for each reference point is checked and if it is empty, the reference point is considered *defunct*, and its allocated ideal cluster size

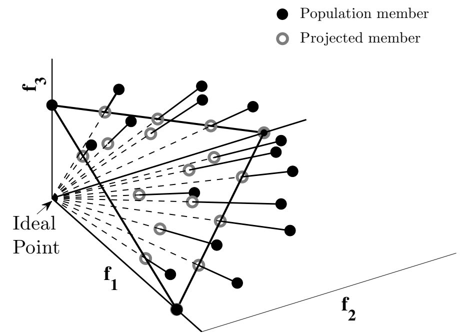

Fig. 1. Projecting points on to the hyper-plane.

ρ_{ideal} is distributed among its neighboring reference points (say m of them). In other words, the deficiency count of each neighboring reference point is increased by ρ_{ideal}/m. This aspect of our proposed algorithm helps solve problems having a disjointed, discontinuous, or biased Pareto-optimal front. The clustering procedure is demonstrated in Figure 2.

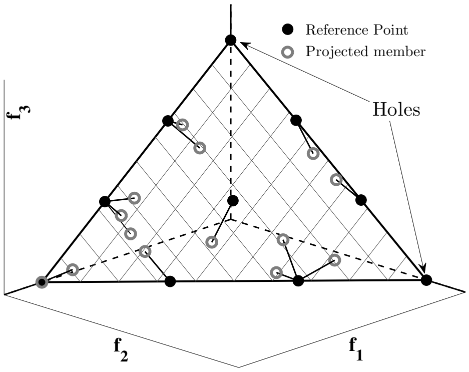

Fig. 2. Clustering points on the hyper-plane.

3. Modify tournament selection: After P_{t+1} is formed, it is used to create a new offspring population Q_{t+1} by applying tournament selection, recombination and mutation operators. The tournament selection operator considers two solutions from P_{t+1} and selects the better solution. This is where we use a hierarchy of considerations. First, if one solution belongs to a better non-dominated level than the other, the former is clearly chosen. Second, if both solutions being compared belong to the same non-domination level but lie on clusters of different reference points, the deficiency counts of their respective reference points are compared. The one lying on a cluster having larger deficiency count is chosen. Third, if two solutions being compared lie on the same non-domination level and also on the same cluster, their ASF values are computed using the corresponding reference point and the one having smaller value is chosen. All these careful comparisons make sure non-dominated solutions are emphasized, followed by solutions that are associated with less-represented reference points, finally followed by solutions are that have better ASF value for the reference points.

4. Recombination with a bias for near-parent solutions: Since reference points are likely to be widely separated from each other, the selected population members P_{t+1} are also likely to be distant from each other. Therefore, we suggest using a recombination operator that will create offspring solutions closer to the parents than away from them. For all our simulations we have used the simulated binary crossover (SBX) operator [10] with a large distribution index so that there is a significant probability of creating a child solution close to the parent solutions.

After the offspring population Q_{t+1} is created, a combined population $R_{t+1} = P_{t+1} \cup Q_{t+1}$ is formed and the above procedure is applied again. The overall procedure is stopped when a pre-specified number of generations have elapsed.

TABLE I
NUMBER OF CLUSTERS AND POPULATION SIZES USED FOR DIFFERENT MANY-OBJECTIVE OPTIMIZATION PROBLEMS.

M	No. of Clusters	MO-NSGA-II-II pop. size	MOEA/D pop. size
3	91	92	91
5	126	128	126
8	330	332	330
10	220	220	220

The popularity of NSGA-II is attributed to its parameter-less approach. The MO-NSGA-II proposed here also does not require any additional parameter except an idea of the total number of trade-off points ($\approx H$) desired by the user. For the given number of objectives (M) in the problem, the corresponding number of divisions (p) can be computed using Equation 1.

IV. RESULTS

In this section, we present simulation results of MO-NSGA-II. Wherever applicable, we compare its performance with MOEA/D. The extended MOEA/D for constraint handling [16] involves further parameters and is not considered here. Besides, we shows a few problem classes in which MOEA/D does not work well, whereas M-MSGA-II has an edge.

In all simulations, we have used a population of size N that is smallest multiple of four higher than H. Table I shows the number of reference points and population size used for different test-problems used in this study. We have also set $p_c = 1$, $p_m = 1/n$ (n is the number of variables), $\eta_c = 20$ and $eta_m = 20$. In all cases, we apply all algorithms 20 times from different initial populations. In order to have a fair comparison for both the algorithms, weight vectors for MOEA/D are derived from the reference points used in MO-NSGA-II.

A. Unconstrained Problems

First, we consider the well-known DTLZ1, DTLZ2, DTLZ3 and DTLZ4 problems [19]. We vary the number of objectives from three to 10. Figure 3 shows the final solutions for the three-objective DTLZ1 problem using $p = 12$. Equation 1 reveals that this will form $\binom{H=3+12-1}{12}$ or 91 points. For this problem, we have chosen a population of size 92. The figure demonstrates the efficacy of MO-NSGA-II.

Similar results are shown for three-objective DTLZ2 with $p = 12$. Again, a nicely distributed set of points are obtained. Results in three, five, eight and 10-objective problems are shown in Table II. Since visual illustration is not possible for more than three-objective problems, we present measures that will test both convergence and diversity of obtained solutions. The column showing the `Convergence Error` presents the worst, average, and best convergence error computed as follows. Since the exact Pareto-optimal front is known for these problems in the form of $\Phi(\mathbf{f}^*) = 0$, the worst, average, and best $|\Phi(\mathbf{f}^*)|$ for all obtained non-dominated solutions are presented. A small value is desired for a well-converged set of solutions. The next five columns represent a diversity

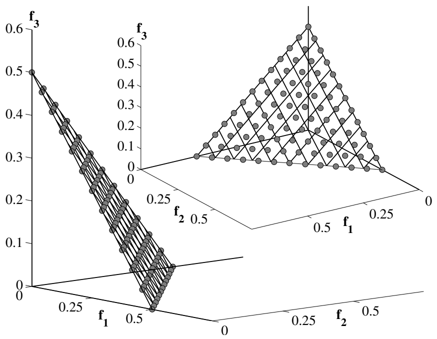

Fig. 3. Trade-off front obtained by MO-NSGA-II for DTLZ1.

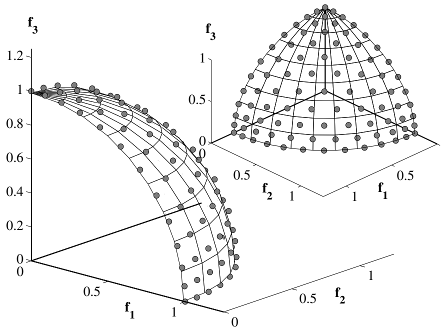

Fig. 4. Trade-off front obtained by MO-NSGA-II for DTLZ2.

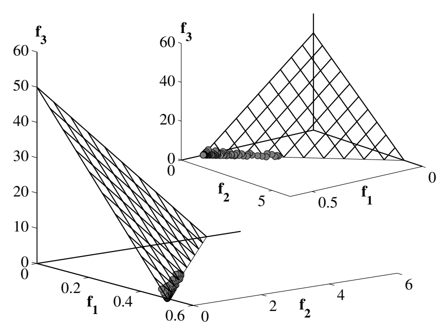

Fig. 5. Front obtained by MOEA/D for scaled three-objective DTLZ1.

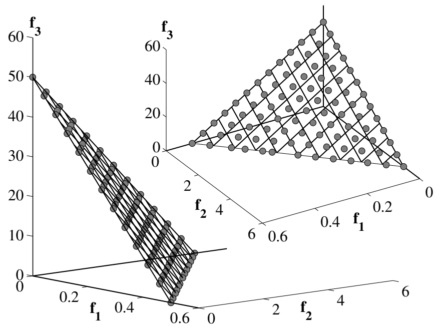

Fig. 6. Front obtained by MO-NSGA-II for scaled three-objective DTLZ1.

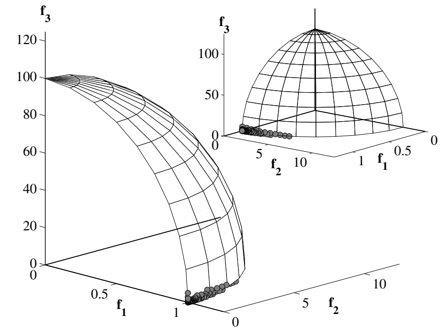

Fig. 7. Front obtained by MOEA/D for scaled three-objective DTLZ2.

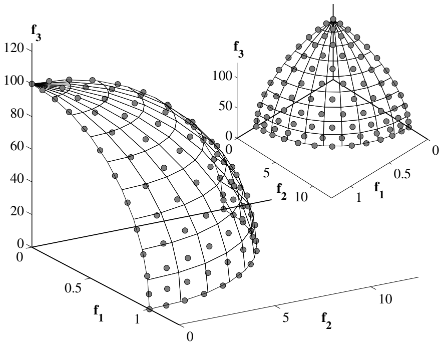

Fig. 8. Front obtained by MO-NSGA-II for scaled three-objective DTLZ2.

estimate of solutions. All H reference points are projected on to the Pareto-optimal surface ($\Phi(\mathbf{f}^*) = 0$). Thereafter, the Euclidean distance between the projected reference point and the closest obtained point is computed. An average distance ($\bar{d}$) between two neighboring reference points is calculated. The table shows the number of points out of H points that are within ζ% of $\bar{d}$ and ζ is varied from 5 to 100. Closer the value to H, better is the diversity in solutions. In DTLZ1, DTLZ2 and DTLZ3 problems, MOEA/D with PBI approach seems to perform well up to 8-objective problems, but MO-NSGA-II performs better on 10-objective problems, in general. For DTLZ4, the performance of MOEA/D is poor. Also, in most occasions, the convergence of solutions to the Pareto-optimal front is better with MO-NSGA-II.

B. More Unconstrained Problems

The above DTLZ problems have one aspect common – all objectives are equally scaled for the Pareto-optimal solutions. To investigate whether the performance of both MOEA/D and MO-NSGA-II methods are due to this property of the DTLZ1-4 problems, we modify DTLZ1 and DTLZ2 to have differently scaled objective values. Objective f_i gets multiplied by 10^{i-1}, so that for the three-objective DTLZ1, the first objective is multiplied by one and the third objective is multiplied by 100.

Both MO-NSGA-II and MOEA/D (with PBI approach) are applied on these two problems with identical parameter settings. Figure 5 shows final front obtained by MOEA/D and Figure 8 shows the same by MO-NSGA-II on scaled DTLZ1 problem. The strength of MO-NSGA-II in handling scaled problems is clear from this study. Apparently, MOEA/D suffers from scaling of objectives. Similar results are observed for the scaled DTLZ2 problem as well (see Figures 7 and 8). Table III shows convergence and diversity measures for problems up to 10 objectives of the scaled DTLZ1 and DTLZ2 problems. For the 10-objective problems, we have used a scaling factor of 3^{i-1}, so that f_{10} is multiplied by 3^9 or 19,683. The original MOEA/D study [2] also suggested a normalized Tchebyshev approach for handling scaled problems. We apply the modified MOEA/D algorithm to the above scaled problems using similar population size and other parameters as before. Table IV shows that for three-objective DTLZ1 and DTLZ2 problems, out of 91 weight vectors, only 4-12 of them could find a representative solution. Based on this poor outcome, we do not show results of this approach on higher-objective scaled problems.

Next, we consider a three-objective WFG-illustrated problem [20] for which the Pareto-optimal front is inverted to that in DTLZ2. While the vertices of the triangular hyperplane lies on the objective axes, the vertices of the Pareto-optimal front lie on the three orthogonal planes. All parameter settings are same as before. Figure 9 shows the obtained Pareto-optimal solutions, all 91 reference points, the defunct reference points (*holes*), and neighboring reference points to the defunct reference points. It is interesting to observe how

TABLE II
CONVERGENCE AND DIVERSITY MEASURES FOR MO-NSGA-II AND MOEA/D ON DTLZ PROBLEMS.

	M/		MO-NSGA-II						MOEA/D					
			Convergence	Diversity estimate: No. of Points					Convergence	Diversity estimate: No. of Points				
Problem	H	Gen.	Error	5%	10%	20%	50%	100%	Error	5%	10%	20%	50%	100%
DTLZ1			$\mathbf{4.11 \times 10^{-4}}$	85	88	90	90	**91**	7.75×10^{-4}	**89**	**91**	**91**	**91**	**91**
	3	750	$\mathbf{1.43 \times 10^{-4}}$	87	90	**91**	**91**	**91**	4.00×10^{-4}	**91**	**91**	**91**	**91**	**91**
	$H = 91$		$\mathbf{1.64 \times 10^{-5}}$	89	90	**91**	**91**	**91**	1.50×10^{-4}	**91**	**91**	**91**	**91**	**91**
			$\mathbf{2.72 \times 10^{-5}}$	124	**126**	**126**	**126**	**126**	1.75×10^{-4}	**126**	**126**	**126**	**126**	**126**
	5	1500	$\mathbf{2.70 \times 10^{-6}}$	**126**	**126**	**126**	**126**	**126**	4.95×10^{-5}	**126**	**126**	**126**	**126**	**126**
	$H = 126$		$\mathbf{2.70 \times 10^{-7}}$	**126**	**126**	**126**	**126**	**126**	1.28×10^{-5}	**126**	**126**	**126**	**126**	**126**
			$\mathbf{2.00 \times 10^{-7}}$	234	301	322	329	**330**	1.89×10^{-5}	**330**	**330**	**330**	**330**	**330**
	8	2500	$\mathbf{1.00 \times 10^{-7}}$	251	307	324	**330**	**330**	1.74×10^{-6}	**330**	**330**	**330**	**330**	**330**
	$H = 330$		$\mathbf{4.82 \times 10^{-11}}$	266	310	325	**330**	**330**	3.74×10^{-7}	**330**	**330**	**330**	**330**	**330**
			$\mathbf{1.40 \times 10^{-6}}$	**219**	**220**	**220**	**220**	**220**	3.07×10^{-3}	212	**220**	**220**	**220**	**220**
	10	5000	$\mathbf{1.00 \times 10^{-7}}$	**220**	**220**	**220**	**220**	**220**	3.81×10^{-5}	219	**220**	**220**	**220**	**220**
	$H = 220$		$\mathbf{3.82 \times 10^{-12}}$	**220**	**220**	**220**	**220**	**220**	9.25×10^{-4}	**220**	**220**	**220**	**220**	**220**
DTLZ2			$\mathbf{2.78 \times 10^{-6}}$	82	88	89	91	**91**	3.00×10^{-6}	**91**	**91**	**91**	**91**	**91**
	3	750	$\mathbf{1.22 \times 10^{-6}}$	83	89	**91**	**91**	**91**	2.74×10^{-6}	**91**	**91**	**91**	**91**	**91**
	$H = 91$		$\mathbf{5.56 \times 10^{-7}}$	85	90	**91**	**91**	**91**	9.79×10^{-7}	**91**	**91**	**91**	**91**	**91**
			2.03×10^{-6}	121	124	**126**	**126**	**126**	$\mathbf{4.17 \times 10^{-7}}$	**126**	**126**	**126**	**126**	**126**
	5	1500	8.61×10^{-7}	123	125	**126**	**126**	**126**	$\mathbf{5.21 \times 10^{-7}}$	**126**	**126**	**126**	**126**	**126**
	$H = 126$		4.34×10^{-7}	124	**126**	**126**	**126**	**126**	$\mathbf{2.31 \times 10^{-7}}$	**126**	**126**	**126**	**126**	**126**
			3.38×10^{-7}	328	**330**	**330**	**330**	**330**	$\mathbf{2.10 \times 10^{-7}}$	**330**	**330**	**330**	**330**	**330**
	8	3500	1.97×10^{-7}	329	**330**	**330**	**330**	**330**	$\mathbf{1.00 \times 10^{-8}}$	**330**	**330**	**330**	**330**	**330**
	$H = 330$		9.33×10^{-8}	**330**	**330**	**330**	**330**	**330**	$\mathbf{1.20 \times 10^{-11}}$	**330**	**330**	**330**	**330**	**330**
			$\mathbf{6.95 \times 10^{-7}}$	**220**	**220**	**220**	**220**	**220**	2.00×10^{-5}	200	200	200	200	200
	10	4000	3.28×10^{-7}	**220**	**220**	**220**	**220**	**220**	$\mathbf{2.22 \times 10^{-7}}$	209	209	210	210	210
	$H = 220$		1.40×10^{-7}	**220**	**220**	**220**	**220**	**220**	$\mathbf{2.07 \times 10^{-8}}$	210	210	210	210	210
DTLZ3			$\mathbf{7.75 \times 10^{-3}}$	0	24	79	89	90	9.56×10^{-3}	0	**32**	**85**	**91**	**91**
	3	1000	$\mathbf{3.60 \times 10^{-3}}$	**36**	78	86	**91**	**91**	6.83×10^{-3}	4	**86**	**91**	**91**	**91**
	$H = 91$		$\mathbf{1.40 \times 10^{-3}}$	61	81	88	**91**	**91**	3.18×10^{-3}	**91**	86	**91**	**91**	**91**
			9.12×10^{-4}	122	124	125	**126**	**126**	$\mathbf{2.99 \times 10^{-4}}$	**126**	**126**	**126**	**126**	**126**
	5	3000	6.23×10^{-4}	123	125	**126**	**126**	**126**	$\mathbf{1.34 \times 10^{-4}}$	**126**	**126**	**126**	**126**	**126**
	$H = 126$		1.66×10^{-4}	124	**126**	**126**	**126**	**126**	$\mathbf{7.53 \times 10^{-5}}$	**126**	**126**	**126**	**126**	**126**
			1.69×10^{-5}	313	323	327	**330**	**330**	$\mathbf{1.16 \times 10^{-5}}$	**330**	**330**	**330**	**330**	**330**
	8	4500	2.51×10^{-6}	317	326	328	**330**	**330**	$\mathbf{1.00 \times 10^{-6}}$	**330**	**330**	**330**	**330**	**330**
	$H = 330$		$\mathbf{1.67 \times 10^{-7}}$	320	329	**330**	**330**	**330**	2.10×10^{-7}	**330**	**330**	**330**	**330**	**330**
			$\mathbf{1.98 \times 10^{-4}}$	**216**	**216**	**217**	**220**	**220**	7.01×10^{-3}	149	161	169	170	170
	10	6000	$\mathbf{3.76 \times 10^{-5}}$	**219**	**220**	**220**	**220**	**220**	1.81×10^{-4}	156	170	178	179	180
	$H = 220$		$\mathbf{2.83 \times 10^{-6}}$	**220**	**220**	**220**	**220**	**220**	3.20×10^{-3}	186	195	196	197	200
DTLZ4			$\mathbf{5.80 \times 10^{-8}}$	**8**	**8**	**8**	**9**	**15**	1.87×10^{-4}	1	1	1	1	1
	3	1500	$\mathbf{7.58 \times 10^{-9}}$	**83**	**88**	**89**	**90**	**91**	6.66×10^{-8}	1	1	1	1	1
	$H = 91$		$\mathbf{1.00 \times 10^{-10}}$	**86**	**89**	**90**	**91**	**91**	1.00×10^{-10}	1	1	1	1	1
			$\mathbf{5.99 \times 10^{-8}}$	**41**	**41**	**41**	**41**	**66**	6.18×10^{-7}	1	1	1	1	1
	5	3000	$\mathbf{4.23 \times 10^{-9}}$	**47**	**48**	**48**	**48**	**71**	2.83×10^{-7}	1	1	1	1	1
	$H = 126$		$\mathbf{1.00 \times 10^{-10}}$	**125**	**126**	**126**	**126**	**126**	1.00×10^{-10}	55	55	55	55	55
			$\mathbf{1.26 \times 10^{-7}}$	**205**	**210**	**210**	**210**	**213**	7.08×10^{-7}	1	1	1	1	1
	8	4000	$\mathbf{5.34 \times 10^{-8}}$	**321**	**327**	**329**	**330**	**330**	1.40×10^{-7}	35	35	35	35	41
	$H = 330$		$\mathbf{1.00 \times 10^{-10}}$	**325**	**329**	**330**	**330**	**330**	1.00×10^{-8}	145	145	145	145	154

multiple reference points become defunct with no associated Pareto-optimal solutions and how neighboring reference points to the defunct ones help find boundary Pareto-optimal points. Figure 10 shows that MO-NSGA-II obtained solutions that are close to the true Pareto-optimal surface.

C. Constrained Problems

To test how our proposed MO-NSGA-II performs on constrained problems, we consider two three-objective problems:

`Constr1` :

$$\begin{array}{lll} \text{Minimize} & f_i = x_i, & i = 1,2,3 \\ \text{subject to} & g_j = (x_j/2)^2 + \left(\sum_{i=1,i\neq j}^{3} x_i^2\right) \geq 1, & j = 1,2,3. \end{array} \qquad (3)$$

All Pareto-optimal solutions make at least one constraint active and lie on specific patches of the ellipsoids.

The second constrained problem `Constr2` is the DTLZ1 problem with the following constraint:

$$g(\mathbf{x}) = \sum_{i=1}^{M} (f_i(\mathbf{x}) - \lambda)^2 - 0.25M\lambda^2 \geq 0, \qquad (4)$$

where $\lambda = \frac{\sum_{i=1}^{M} f_i}{M}$. The constraint makes an infeasible hole in the middle of the triangular Pareto-optimal front.

To handle constraints, we use the constrained domination principle [17], instead of the original domination principle in deciding non-dominated front members. The rest of the MO-NSGA-II is identical to that described before. MO-NSGA-

TABLE III
CONVERGENCE AND DIVERSITY MEASURES IN THE CASE OF SCALED DTLZ PROBLEMS.

			MO-NSGA-II						MOEA/D					
	M/		Convergence	Diversity estimate: No. of Points					Convergence	Diversity estimate: No. of Points				
Problem	H	Gen.	Error	5%	10%	20%	50%	100%	Error	5%	10%	20%	50%	100%
DTLZ1			2.54×10^{-4}	**88**	**91**	**91**	**91**	**91**	5.25×10^{-4}	1	1	2	3	6
	3	750	7.11×10^{-5}	**90**	**91**	**91**	**91**	**91**	2.49×10^{-4}	1	1	2	3	6
	$H = 91$		1.67×10^{-5}	**90**	**91**	**91**	**91**	**91**	1.15×10^{-4}	1	1	2	3	6
			6.32×10^{-5}	**126**	**126**	**126**	**126**	**126**	2.50×10^{-4}	15	21	29	46	56
	5	1500	5.20×10^{-6}	**126**	**126**	**126**	**126**	**126**	1.97×10^{-4}	15	21	29	46	56
	$H = 126$		4.31×10^{-7}	**126**	**126**	**126**	**126**	**126**	1.23×10^{-4}	15	21	29	46	56
			7.15×10^{-5}	**326**	**330**	**330**	**330**	**330**	4.67×10^{-6}	120	126	161	203	210
	8	2500	1.45×10^{-6}	**329**	**330**	**330**	**330**	**330**	1.50×10^{-6}	120	126	164	203	210
	$H = 330$		1.56×10^{-7}	**330**	**330**	**330**	**330**	**330**	6.55×10^{-7}	120	126	164	203	210
			4.62×10^{-6}	**220**	**220**	**220**	**220**	**220**	1.27×10^{-3}	55	77	83	119	156
	10	5000	3.10×10^{-7}	**220**	**220**	**220**	**220**	**220**	1.15×10^{-2}	55	77	83	119	156
	$H = 220$		8.94×10^{-9}	**220**	**220**	**220**	**220**	**220**	1.21×10^{-2}	55	77	83	119	156
DTLZ2			1.47×10^{-6}	**57**	**75**	**85**	**90**	**91**	6.11×10^{-7}	1	1	1	1	1
	3	750	6.22×10^{-7}	**59**	**76**	**87**	**91**	**91**	3.34×10^{-7}	1	1	1	1	1
	$H = 91$		2.92×10^{-7}	**62**	**79**	**88**	**91**	**91**	1.00×10^{-10}	1	1	1	1	1
			3.05×10^{-6}	**61**	**67**	**112**	**126**	**126**	2.49×10^{-7}	5	6	11	19	21
	5	1500	1.15×10^{-6}	**61**	**70**	**115**	**126**	**126**	1.40×10^{-7}	5	6	11	19	21
	$H = 126$		5.68×10^{-7}	**63**	**78**	**121**	**126**	**126**	1.00×10^{-10}	5	6	11	19	21
			1.26×10^{-6}	**129**	**148**	**168**	**186**	**211**	8.05×10^{-8}	70	70	105	126	126
	8	3500	3.46×10^{-7}	**144**	**156**	**174**	**192**	**214**	1.00×10^{-10}	70	70	105	126	126
	$H = 330$		1.60×10^{-7}	**148**	**161**	**176**	**195**	**216**	1.00×10^{-10}	70	70	105	126	126
			9.14×10^{-7}	**166**	**178**	**220**	**220**	**220**	1.23×10^{-8}	35	50	61	84	112
	10	4000	4.78×10^{-7}	**173**	**204**	**220**	**220**	**220**	3.25×10^{-9}	35	50	61	84	112
	$H = 220$		1.89×10^{-7}	**175**	**219**	**220**	**220**	**220**	2.78×10^{-9}	35	50	61	84	112

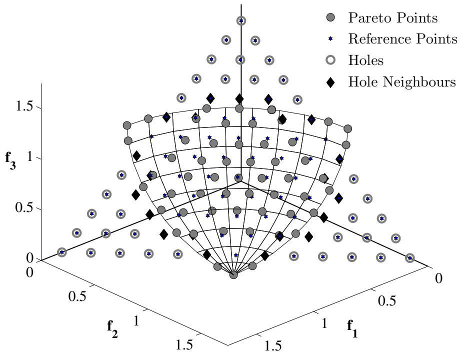

Fig. 9. Reference points and obtained solutions by MO-NSGA-II for WFG-illustrated problem.

II parameters are also same as before. Figures 11 and 12 demonstrate clearly that MO-NSGA-II is able to find the feasible Pareto-optimal solutions close to the chosen reference points.

Table V shows the performance of MO-NSGA-II on `Constr1` problem having up to 10 objectives. MO-NSGA-II approach is also able to find feasible front for higher-objective version of `Constr2` problem as well, but due to space restrictions, we do not show those results here.

V. CONCLUSIONS

There is a growing need for efficient algorithms for handling many-objective optimization problems. In this paper, we have suggested difficulties of existing domination-based EMO algorithms for solving many-objective problems and proposed a

TABLE IV
CONVERGENCE AND DIVERSITY MEASURES FOR THREE-OBJECTIVE SCALED DTLZ PROBLEMS USING NORMALIZED TCHEBYSHEV METRIC BASED MOEA/D.

	MOEA/D-TCH					
	Convergence	Diversity estimate: Points out of 91				
Problem	Error	5%	10%	20%	50%	100%
DTLZ1	2.44×10^{-4}	4	6	6	6	8
3-obj	7.08×10^{-5}	4	6	6	6	8
	4.80×10^{-5}	6	6	8	8	10
DTLZ2	1.26×10^{-7}	6	10	10	10	12
3-obj	1.00×10^{-8}	8	10	10	10	12
	9.78×10^{-9}	8	10	10	10	12

TABLE V
PERFORMANCE OF MO-NSGA-II ON `Constr1` PROBLEM HAVING THREE TO 10 OBJECTIVES. MORE POINTS, BETTER IS THE DIVERSITY.

		Convergence	Diversity estimate: No. of Points				
M	Gen.	Error	5%	10%	20%	50%	100%
		2.51×10^{-3}	7	16	33	62	76
3	750	8.03×10^{-4}	9	18	35	66	79
		2.05×10^{-4}	10	22	40	71	87
		6.99×10^{-3}	27	51	81	115	124
5	1500	1.71×10^{-3}	28	54	86	116	124
		4.72×10^{-4}	33	58	90	117	125
		1.28×10^{-2}	83	175	254	309	323
8	3000	3.72×10^{-3}	100	185	257	313	325
		1.01×10^{-3}	108	189	262	315	327
		2.81×10^{-3}	158	197	208	214	218
10	5000	7.77×10^{-4}	188	202	209	216	218
		1.84×10^{-4}	195	203	211	217	219

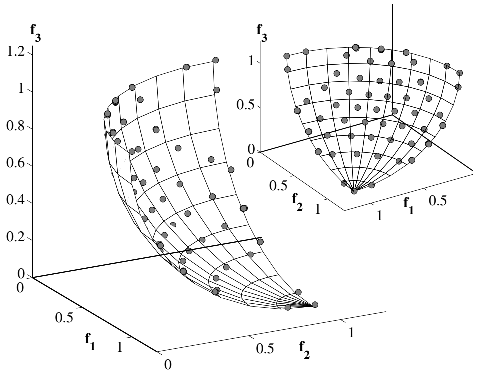

Fig. 10. Trade-off front obtained by MO-NSGA-II for WFG-illustrated problem.

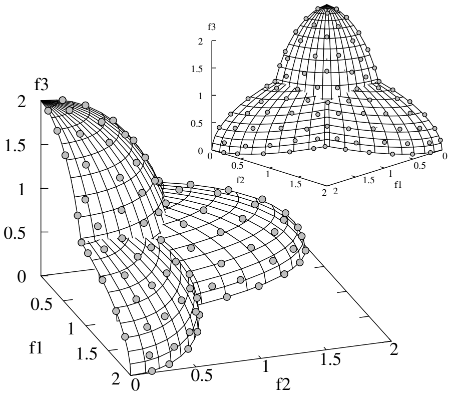

Fig. 11. Trade-off front obtained by MO-NSGA-II for problem Constr1.

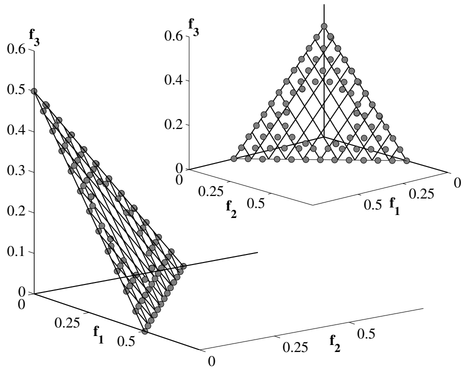

Fig. 12. Trade-off front obtained by MO-NSGA-II for problem Constr2.

reference-point based NSGA-II (or MO-NSGA-II) algorithm. The procedure uses an adaptive reference-point update scheme, a clustering based selection scheme and preference for less-crowded reference points for finding and maintaining trade-off solutions. On a number of unconstrained and constrained problems having three to 10 objectives, MO-NSGA-II has consistently produced well-converged and well-diversified solutions. A nice feature of the proposed MO-NSGA-II algorithm is that it does not require any additional parameter. With these proof-of-principle studies, more rigorous tests are now needed to fully investigate the potential of MO-NSGA-II.

ACKNOWLEDGMENT

The authors would like to thank Michigan State University, East Lansing, for supporting this work during their stay.

REFERENCES

[1] E. J. Hughes, "Evolutionary many-objective optimisation: many once or one many?" in *IEEE Congress on Evolutionary Computation (CEC-2005)*, 2005, pp. 222–227.

[2] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.

[3] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, in press.

[4] J. A. López and C. A. Coello Coello, "Some techniques to deal with many-objective problems," in *Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*. New York, NY, USA: ACM, 2009, pp. 2693–2696.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proceedings of Congress on Evolutionary Computation (CEC-2008)*, 2008, pp. 2424–2431.

[6] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, in press.

[7] K. Deb and D. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, 2006, pp. 3352–3360.

[8] M. Garza-Fabre, G. T. Pulido, and C. A. C. Coello, "Ranking methods for many-objective optimization," in *Proceedings of Mexican International Conference on Artificial Intelligence (MICAI-2009)*. Springer, 2009, pp. 633–645.

[9] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[10] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[11] K. Deb, J. Sundar, N. Uday, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research (IJCIR)*, vol. 2, no. 6, pp. 273–286, 2006.

[12] D. Brockhoff and E. Zitzler, "Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem," in *Operations Research Proceedings 2006*, K. H. Waldmann and U. M. Stocker, Eds. Springer, 2007, pp. 423–429.

[13] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.

[14] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization," *Evolutionary Computation Journal*, in press.

[15] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston: Kluwer, 1999.

[16] M. A. Jan and Q. Zhang, "Moea/d for constrained multiobjective optimization: Some preliminary experimental results," in *2010 UK Workshop on Computational Intelligence (UKCI)*, 2010.

[17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[18] I. Das and J. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal of Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds. London: Springer-Verlag, 2005, pp. 105–145.

[20] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proceedings of the Evolutionary Multi-Criterion Optimization (EMO-2005)*. Berlin, Germany: Springer, 2005.