An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach

Himanshu Jain and Kalyanmoy Deb, *Fellow, IEEE*

Abstract—In the precursor paper [1], a many-objective optimization method (NSGA-III), based on the NSGA-II framework, was suggested and applied to a number of unconstrained (with box constraints alone) test and practical problems. In this paper, we extend NSGA-III to solve generic constrained many-objective optimization problems. In the process, we also suggest three types of constrained test problems that are scalable to any number of objectives and may provide different types of challenges to a many-objective optimizer. A previously suggested MOEA/D algorithm is also extended to solve constrained problems. Results using constrained NSGA-III and constrained MOEA/D show an edge of the former particularly in solving problems having a large number of objectives. Further, the NSGA-III algorithm is made adaptive in deleting and including new reference points on the fly. The resulting adaptive NSGA-III is shown to provide a denser representation of the Pareto-optimal front compared to the original NSGA-III for an identical computational effort. This and the original NSGA-III paper together suggest and amply test a viable evolutionary many-objective optimization algorithm for handling constrained and unconstrained problems. These studies should encourage researchers to use and pay further attention to the growing interest in evolutionary many-objective optimization.

Index Terms—Many-objective optimization, evolutionary computation, large dimension, NSGA-III, non-dominated sorting, multi-criterion optimization.

I. INTRODUCTION

Evolutionary multi-objective optimization (EMO) methodologies suggested since early nineties have amply demonstrated the use of evolutionary algorithms in solving optimization problems having mostly two and three objectives [2], [3], [4]. The main reason for their popularity was their ability to find multiple trade-off solutions in a single simulation run and their ease and flexibility to focus on any part of the Pareto-optimal frontier. Besides their academic use, EMO has also been practiced in industries, mainly due to the availability of a number of commercial EMO softwares. EMO has also been diversified to hybridize with its contemporary fields,

H. Jain is with Eaton Corporation, Pune 411014, India, email: himanshu.j689@gmail.com.

K. Deb is with Department of Electrical and Computer Engineering. Michigan State University, 428 S. Shaw Lane, 2120 EB, East Lansing, MI 48824, USA, e-mail: kdeb@egr.msu.edu (see http://www.egr.msu.edu/˜kdeb). Prof. Deb is also a visiting professor at Aalto University School of Business, Finland and University of Skövde, Sweden.

such as in multiple criterion decision making (MCDM) and in mathematical multi-objective optimization studies. With all these all-round developments, despite a few studies, one aspect mainly remained unexplored: the issue of handling a large number of objectives. It has been clearly shown in the EMO literature that an EMO methodology that works so well for two or three objectives has the 'curse of dimensionality' in solving more than three-objective problems [5], [6]. The main reasons for their poor working – domination principle being too weak to provide an adequate selection pressure, a large population size requirement, etc. – were not unknown to the EMO researchers, but were difficult to alleviate in an adequate manner. Although a few many-objective evolutionary algorithms have been suggested in the past [6], [7], [8], [9], [10], [11], [12], there is still a need for more efficient algorithms for many-objective optimization, similar to popular two or three-objective EMO methods, such as NSGA-II [13], SPEA2 [14] and others.

In the precursor study [1], we suggested an evolutionary many-objective optimization algorithm by extending the NSGA-II framework. Realizing the computational challenge associated with a population-based optimization algorithm in converging to Pareto-optimal front and simultaneously spread its population along the entire front, in NSGA-III, the latter task is aided by supplying a set of pre-defined reference points. The algorithm was then expected to focus its search to find an associated Pareto-optimal solution for each reference point. Keeping NSGA-II's emphasis on non-dominated solutions intact, its elitist selection mechanism was modified to incorporate three new operations – *normalization* of objective vectors and the supplied reference points so as to have both sets within a single range, *association* of every population member with a particular reference point based on a proximity measure, and *niching* of accepted population members in order to ensure a diverse set of solutions. The results on several test problems and practical problems have amply demonstrated NSGA-III's usefulness in solving two to 15-objective unconstrained problems with specified variable bounds. Since the supplied reference points were chosen as a diverse set, the obtained trade-off solutions were also likely to be diverse. Since multiple Pareto-optimal points were targeted to be found simultaneously in a single simulation run, NSGA-III provided an efficient parallel search.

In the earlier paper, NSGA-III was restricted to solve prob-

lems having box constraints alone. In this paper, we extend NSGA-III to solve constrained many-objective optimization problems of the following type:

$$
\begin{array}{ll}
\text{Minimize} & (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\
\text{subject to} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \ldots, J, \\
& h_k(\mathbf{x}) = 0, \quad k = 1, 2, \ldots, K, \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n.
\end{array}
\tag{1}
$$

One advantage of using an evolutionary algorithm for solving the above problem is that the box constraints (the last set of constraints on variables alone) can be handled automatically by initializing all population members satisfying the bounds and by ensuring that the creation of offspring solutions are always within the specified lower and upper bounds. Thus, a procedure to handle inequality and equality constraints remains to be incorporated with the NSGA-III algorithm. A linear equality constraint, if present in a problem, can be used to eliminate one variable using the constraint. Thus, linear constraints, in general, help reduce the dimensionality of the search space.

In this paper, we modify certain operators of NSGA-III to emphasize feasible solutions more than the infeasible solutions in a population. Two main changes are suggested in the original algorithm for this purpose. The modifications suggested still keep the overall algorithm parameter-less (besides the need of usual genetic parameters). Another aspect of the extension is that if all population members are feasible or an unconstrained problem is supplied, the constrained NSGA-III reduces to the original unconstrained NSGA-III algorithm. To evaluate its performance, the proposed constrained NSGA-III procedure is applied to a number of many-objective test problems, suggested here for the first time, and two practical many-objective problems.

The constrained NSGA-III approach is also applied with a few preferred reference points to find a handful of solutions on a preferred region on the Pareto-optimal set. On a test problem and on a practical problem, the approach is able to find five or 10 trade-off solutions corresponding five or 10 supplied preferred reference points.

During the course of the earlier study [1] and this study on constraint-handling using NSGA-III, we have realized that in certain problems, not all specified reference points will correspond to a Pareto-optimal solution. In such a scenario, the processing of these 'non-useful' reference points causes a waste of computations. In this paper, we rectify this difficulty by suggesting an adaptive NSGA-III that identifies non-useful reference points and adaptively deletes and includes new reference points in addition to the supplied reference points. Simulation results on a number of many-objective test problems and practical problems support the modifications made and demonstrate the usefulness of the proposed procedure.

In the remainder of this paper, we provide a brief overview of existing many-objective constraint handling procedures in Section II. Thereafter, the constrained NSGA-III is described in detail by first providing a brief description of the original NSGA-III in Section III. As an alternative algorithm, we extend the MOEA/D-DE algorithm proposed in [15] to solve

constrained problems in the next section. The resulting C-MOEA/D and proposed constrained NSGA-III algorithms are then applied to three types of scalable constrained test problems in Section V. This section also applies the constrained NSGA-III algorithm to two engineering design problems. Next, to show NSGA-III's ability to be hybridized with a decision-making technique, NSGA-III is applied with a few preferred reference points. Results on two problems are shown in Section VI. Thereafter, in Section VII, we have proposed an adaptive NSGA-III algorithm in detail and applied it to solve many-objective test problems and practical problems. Conclusions of the extensive study are then drawn in Section IX. The appendix contains the optimization problem formulations of two engineering design problems considered in this study.

II. Existing Many-Objective Constraint-Handling Procedures

There is not enough literature on handling constraints in a many-objective optimization algorithm, as most existing many-objective EA studies handled unconstrained problems only. MOEA/D, after its suggestion [7] in 2008, was extended to include the differential evolution (DE) operator [16], and later suggested to address constraints using the MOEA/D-DE approach [15]. We briefly describe the procedure below.

The constrained MOEA/D-DE algorithm [15] is different from its unconstrained version in the following ways: (i) it uses a penalty function to handle constraints, but introduces two penalty parameters s_1 and s_2 for the purpose, (ii) it restricts the number of reference directions a new child solution can be associated with by introducing a limiting parameter n_r, (iii) it chooses a mating partner of a solution based on a probability distribution involving a parameter δ, and (iv) it uses the differential evolution [17] to create new solutions which involves two parameters CR and p_m. The results were reported to depend on the choice of the penalty parameters. Moreover, fixing six parameters adequately for a problem is the main drawback of the above constrained MOEA/D-DE approach. In Section IV, we suggest a different constrained version of MOEA/D based on the principles of our proposed approach that may remain as a viable pragmatic extension of MOEA/D for handling constraints.

The constrained handling approaches proposed by Fonseca and Fleming [18] and by Deb et al. [19], [13] do not require any additional parameters. By making pairwise comparisons between population members, feasible and less constraint-violated solutions were emphasized. Although these methods were suggested for multi-objective optimization problems, they can very well be tried for solving many-objective optimization problems. Our proposed NSGA-III approach, described next, uses these ideas for handling constraints.

III. Proposed NSGA-III with a Constraint Handling Approach

Before we describe the constraint handling procedure, we present a brief outline of recently proposed many-objective NSGA-II procedure described in the original paper [1].

NSGA-III starts with description of a set of reference points Z. The current parent population P_t (at generation t) is used to create an offspring population Q_t by using genetic operations. The combined population $R_t = P_t \cup Q_t$ is sorted into different levels of non-domination. All population members up to the last front (F_l) that could not be fully accommodated are saved in a set S_t and remaining members of R_t are rejected. Members in $S_t \backslash F_l$ are already selected for the next generation and the remaining population slots are selected from F_l. In the original NSGA-II, the last front members having the largest *crowding distance* values (providing widest diversity) were chosen. The crowding distance operation does not work well for many-objective problems [20] and here we modify the selection mechanism by performing a more systematic analysis of members of S_t with respect to the supplied reference points.

Objective values and supplied reference points are first *normalized* so that they have an identical range. This way, the ideal point of the set is the zero vector. Each member of S_t is then *associated* with a reference point depending on the proximity of the member with a reference line obtained by joining the ideal point with the reference point. This procedure helps determine the number and indices of population members associated with each supplied reference point in $S \backslash F_l$. Thereafter, a *niching* procedure is used to select population members from F_l that are not well represented in $S_t \backslash F_l$ using the outcome of the above association procedure. The reference points that have the least number of association in $S \backslash F_l$ population are looked for an associated point in F_l set. Such F_l members are then added one at a time to fill the population. Such a careful selection strategy is found to have a slightly larger computational complexity of $O(N^2 \log N)$ compared to $O(N(\log N)^{M-2})$ complexity of NSGA-II, but NSGA-III helped solve problems having a large number of objectives.

We now propose an extension of the above NSGA-III procedure to handle generic equality and/or inequality constraints. We discuss the modifications one by one.

A. Modifications in the Elitist Selection Operator

Recall that the combined population R_t needs to be sorted according to different non-domination levels. For unconstrained problems, the objective function values alone are considered for the domination check between two solutions. But in the presence of constraints, we follow the constraint-domination principle adopted in NSGA-II [13] using the ideas from [18], [19]:

Definition A solution $\mathbf{x}^{(1)}$ is said to constraint-dominate another solution $\mathbf{x}^{(2)}$, if any one of the following conditions is true:

1) if $\mathbf{x}^{(1)}$ is feasible and $\mathbf{x}^{(2)}$ is infeasible,
2) if $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are infeasible and $\mathbf{x}^{(1)}$ has a smaller *constraint violation* value, or
3) if $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are feasible and $\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$ with the usual domination principle [21], [22].

For calculating the constraint violation value ($CV(\mathbf{x})$) of a solution $\mathbf{x}$, we suggest normalizing all constraints by dividing

the constraint functions by the constant present (that is, for $g_j(\mathbf{x}) \geq b_j$, the normalized constraint function becomes $\bar{g}_j(\mathbf{x}) = g_j(\mathbf{x})/b_j - 1 \geq 0$ and similarly $\bar{h}_k(\mathbf{x})$ can also be normalized) and then using the following measure:

$$CV(\mathbf{x}) = \sum_{j=1}^{J} \langle \bar{g}_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} |\bar{h}_k(\mathbf{x})|, \qquad (2)$$

where the bracket operator $\langle \alpha \rangle$ returns the negative of α, if $\alpha < 0$ and returns zero, otherwise.

The population R_t of size $2N$ can be sorted into different non-domination levels according to the above constraint-domination principle. If every population member is infeasible, the non-domination sorting procedure will assign the solution having the smallest CV in the first front, the solution with the next smallest CV in the second front and so on. Thus, there will be a total of $2N$ fronts, unless there exists two solutions having an identical CV value. On the other hand, if all population members are feasible, the non-domination sorting will be identical to that obtained by the usual domination principle. In most cases, the population may have some feasible solutions (set $\mathcal{F}$) and some infeasible solutions (set $\mathcal{I}$). In this case, the above sorting procedure will arrange feasible solutions according to their non-domination levels in the top of the sorted levels and the infeasible solutions will occupy the next levels one (in most cases) in each front starting with the least constraint-violated solution.

Once the combined population R_t is sorted according to constraint-domination, the number of feasible solutions N_f in R_t is counted. If $N_f \leq N$, meaning that there are at most N feasible points in R_t, we definitely select all feasible solutions for P_{t+1} and the remaining $(N - |P_{t+1}|)$ population slots are filled with top levels of the infeasible solutions (having smaller CV values). However, if $N_f > N$, meaning that there are more feasible solutions in R_t than required, we do not consider the infeasible solutions at all and follow the unconstrained NSGA-III selection procedure with feasible solution set $R_t \backslash \mathcal{I}$.

In either case, we then update the population ideal ($\mathbf{z}^{\min}$) and nadir points ($\mathbf{z}^{\max}$) using the objective values of feasible solutions for the normalization procedure, discussed in the original NSGA-III paper.

B. Modification in Creation of Offspring Population

The original NSGA-III algorithm suggested to use a population size (N) almost equal to the number of reference points (H). The parameter H is derived from a combinatorial value $\binom{(M+p-1)}{p}$ for a given p. The population size is recommended to be the smallest multiple of four, greater than H. Thus, every population member is likely to be associated with a different reference point and at the end it is desired that there will be at least one Pareto-optimal solution associated with every reference point. Due to this one-member-to-one-reference-point expectation, no additional tournament selection was applied to the parent population P_t to create the offspring population Q_t. However, in the presence of infeasible solutions in the population, there is a need for bringing back the tournament selection operator particularly for emphasizing

a feasible solution over an infeasible solution and a small CV solution over a large CV solution.

For this purpose, we select two members from P_t at random, and a binary tournament selection is applied, as follows, to select a better solution:

Definition The modified tournament selection operation between solutions p_1 and p_2 is defined as follows:

1) if p_1 is feasible and p_2 is infeasible, select p_1 else if p_2 is feasible and p_1 is infeasible, select p_2,
2) if p_1 and p_2 are infeasible then if p_1 has a smaller constraint violation CV, select p_1, else if p_2 has a smaller constraint violation CV, select p_2, and
3) if both p_1 and p_2 are feasible then p_1 or p_2 is chosen at random.

The above conditions for choosing a solution over another is similar to that used in defining constraint-domination, except that when both solutions are feasible, a random solution is now chosen, thereby implying that there is no tournament selection performed in this case.

Similarly, another pair of members are randomly selected from population P_t and the above modified tournament selection is applied to select the second parent. Thereafter, crossover and mutation operators are applied on both parents to produce two offspring solutions as usual. This process is continued till N offspring are created to form the population Q_t.

The overall procedure is presented in a pseudo-code in Algorithm 1. Notice how the procedure becomes similar to the unconstrained NSGA-III selection operator (described in the original study [1]) when there is no infeasible population member or when there is equality or inequality constraints specified in the optimization problem formulation.

Algorithm 1 `Tournament Selection`(p_1, p_2) procedure

Require: p_1, p_2
Ensure: p'
  1: **if** feasible(p_1) = TRUE and feasible(p_2) = FALSE **then**
  2:    $p' = p_1$
  3: **else if** feasible(p_1) = FALSE and feasible(p_2) = TRUE **then**
  4:    $p' = p_2$
  5: **else if** feasible(p_1) = FALSE and feasible(p_2) = FALSE **then**
  6:    **if** $CV(p_1) > CV(p_2)$ **then**
  7:      $p' = p_2$
  8:    **else if** $CV(p_1) < CV(p_2)$ **then**
  9:      $p' = p_1$
10:    **else**
11:      $p' = \text{random}(p_1, p_2)$
12:    **end if**
13: **else**
14:    $p' = \text{random}(p_1, p_2)$
15: **end if**

The rest of the NSGA-III procedure described in the original paper [1] remains the same. A careful analysis will reveal that

the above constrained NSGA-III approach does not introduce any new parameter for handling constraints. This remains as a hallmark feature of our proposed constraint handling approach.

IV. PROPOSED CONSTRAINT-MOEA/D METHOD (C-MOEA/D)

The original MOEA/D approach [7] was extended to include the DE operator to develop MOEA/D-DE approach [16], and subsequently suggested a constrained MOEA/D-DE approach to handle constraints [15], but as discussed in Section II, the constrained MOEA/D-DE approach is based on a penalty function concept that requires two penalty parameters. In addition, the approach also requires four other parameters that are needed to be set right in solving an arbitrary problem. In the original NSGA-III study [1], we have reported that MOEA/D approach, in principle, can perform well in solving many-objective optimization problems, which the developers of MOEA/D did not demonstrate.

Here, we modify the MOEA/D-DE approach with a similar constrained handling approach as described above, hoping that the proposed C-MOEA/D can also become a competing and alternate algorithm for constrained many-objective problem solving. We make the following modifications to the original MOEA/D-DE approach [15], [16].

When a child solution $\mathbf{y}$ is compared with a randomly picked member $\mathbf{x}$ from its neighborhood, instead of replacing the member just based on performance metric (PBI or Tchebycheff), the constraint violation, if any, of both solutions are checked. Following four scenarios can occur:

1) Solution $\mathbf{x}$ is feasible while solution $\mathbf{y}$ is infeasible. Then, $\mathbf{x}$ is not replaced by $\mathbf{y}$. No computation of PBI or Tchebycheff metric is needed for any of these two solutions.
2) Solution $\mathbf{x}$ is infeasible while solution $\mathbf{y}$ is feasible. Then, $\mathbf{x}$ is replaced by $\mathbf{y}$.
3) Both solutions $\mathbf{x}$ and $\mathbf{y}$ are infeasible. If $\mathbf{x}$ has a larger constraint violation than $\mathbf{y}$ (that is, $CV(\mathbf{x}) > CV(\mathbf{y})$, then $\mathbf{x}$ is replaced by $\mathbf{y}$.
4) Both solutions $\mathbf{x}$ and $\mathbf{y}$ are feasible. Here, we propose the use of a performance measure. If PBI (or Tchebycheff) metric value of $\mathbf{x}$ is worse than that of $\mathbf{y}$, then $\mathbf{x}$ is replaced by $\mathbf{y}$.

The above modifications are in tune with that adopted in constrained NSGA-III algorithm and should provide an adequate emphasis for feasible and small-CV solutions in the population. Importantly, no new parameter is introduced in the algorithm. Moreover, in the original study [1], the DE operator for creating offspring solutions did not perform well with the rest of the MOEA/D algorithm. Based on that study, here, we do not use the DE operator, instead a real-coded GA with SBX and polynomial mutation operators are used for creating the offspring population. We also propose the use of PBI metric (instead of the Tchebycheff metric), as PBI metric was found to work better in the original study [1]. We name this version of MOEA/D as constrained MOEA/D or simply C-MOEA/D.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

5

V. RESULTS

In this section, we present simulation results of the proposed constrained NSGA-III and C-MOEA/D approaches. For this purpose, we use a number of constrained test problems having three to 15 objectives, designed to introduce different types of difficulties to an algorithm. The problems are scalable both in the number of objectives and in the number of variables.

For each problem, 20 different runs with different initial populations are carried out and the best, median and the worst IGD performance values (which can only be computed for a test problem with a known Pareto-optimal front) are reported. To compute IGD value, first, we compute the targeted points ($\mathbf{Z}$) on the known Pareto-optimal front from the supplied reference points or directions in the normalized objective space. Then, for an algorithm, we obtain the final non-dominated points (set $\mathbf{A}$) in the objective space. Now, we compute the IGD metric value as the average Euclidean distance of points in set $\mathbf{Z}$ with their nearest members of all points in set A:

$$\text{IGD}(\mathbf{A}, \mathbf{Z}) = \frac{1}{|\mathbf{Z}|} \sum_{i=1}^{|\mathbf{Z}|} \min_{j=1}^{|\mathbf{A}|} d(\mathbf{z}_i, \mathbf{a}_j), \qquad (3)$$

where $d(\mathbf{z}_i, \mathbf{a}_j) = \|\mathbf{z}_i - \mathbf{a}_j\|_2$. For both algorithms, the population members from the final generation are presented and used for computing the above IGD metric. The number of reference points, population size and other parameters are kept in agreement with the original study [1] and are tabulated in Tables I and II. In case of C-MOEA/D, two parameters δ

TABLE I
NUMBER OF REFERENCE POINTS/DIRECTIONS AND CORRESPONDING POPULATION SIZES USED IN CONSTRAINED NSGA-III AND C-MOEA/D ALGORITHMS.

No. of objectives (M)	Ref. pts./ Ref. dirn. (H)	NSGA-III popsize (N)	MOEA/D popsize (N')
3	91	92	91
5	210	212	210
8	156	156	156
10	275	276	275
15	135	136	135

TABLE II
PARAMETER VALUES USED IN CONSTRAINED NSGA-III AND C-MOEA/D. n IS THE NUMBER OF VARIABLES.

Parameters	NSGA-III	MOEA/D
SBX probability [23], p_c	1	1
Polynomial mutation prob. [2], p_m	$1/n$	$1/n$
η_c [23]	30	20
η_m [23]	20	20

(probability with which the parent solutions are selected from neighborhood) and n_r (maximal number of solutions replaced by an offspring solution) are set as 0.9 and 2, respectively, as suggested by the developers [15]. In contrast, the proposed constrained-handling NSGA-III does not require setting of any new parameter.

A. Constrained Problems of Type-1

In Type-1 constrained problems, the original Pareto-optimal front is still optimal, but there is an infeasible barrier in ap-

proaching the Pareto-optimal front. This is achieved by adding a constraint to the original problem. The barrier provides infeasible regions in the objective space that an algorithm must learn to overcome, thereby providing a difficulty in converging to the true Pareto-optimal front. DTLZ1 and DTLZ3 problems [24] are modified according to this principle in this study.

For the type 1 constrained DTLZ1 (or C1-DTLZ1), only a part of objective space that is close to Pareto-front is made feasible, as shown in Figure 1. The objective functions are kept the same as they were in the original DTLZ1 problem, while the following constraint is now added:

$$c(\mathbf{x}) = 1 - \frac{f_M(\mathbf{x})}{0.6} - \sum_{i=1}^{M-1} \frac{f_i(\mathbf{x})}{0.5} \geq 0. \qquad (4)$$

The feasible region and the Pareto-optimal front are shown for a two-objective C1-DTLZ1 problem in Figure 1. In all simulations, we use $k = 5$ variables for the original g-function [24], thereby making a total of $(M + 4)$ variables to the M-objective C1-DTLZ1 problem.

In the case of C1-DTLZ3 problem, a band of infeasible space is introduced adjacent to the Pareto-optimal front, as shown in Figure 2. Again, the objective functions are kept the same as in original DTLZ3 problem [24], while the following constraint is added:

$$c(\mathbf{x}) = \left(\sum_{i=1}^{M} f_i(\mathbf{x})^2 - 16 \right) \left(\sum_{i=1}^{M} f_i(\mathbf{x})^2 - r^2 \right) \geq 0. \qquad (5)$$

where, $r = \{9, 12.5, 12.5, 15, 15\}$ is the radius of the hypersphere for $M = \{3, 5, 8, 10, 15\}$. For C1-DTLZ3, we use $k = 10$, so that total number of variables are $(M + 9)$ in a M-objective problem.
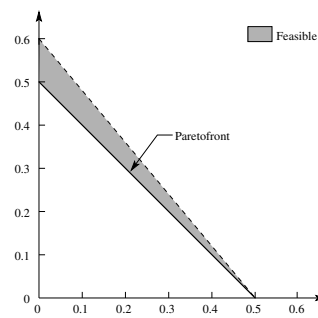
Fig. 1. Two objective version of C1-DTLZ1 problem.
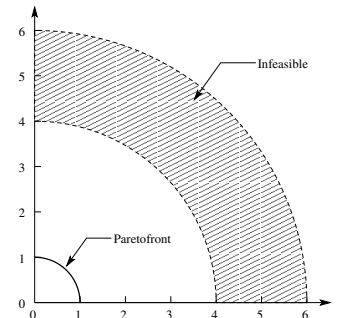
Fig. 2. Two objective version of C1-DTLZ3 problem.

Both algorithms (NSGA-III and C-MOEA/D) are tested on three to 15 objective versions of above two problems. Figure 3 shows that in the case of three-objective C1-DTLZ1 problem, NSGA-III is able to reach the feasible region and find a well-distributed set of points on the entire Pareto-optimal front. C-MOEA/D is also able to find a nice distribution of points (Figure 4). However, as it is evident from Table III, in most of the cases for the C1-DTLZ1 problem, NSGA-III performs better than C-MOEA/D in terms of the IGD metric. Interestingly, the best performance of C-MOEA/D is in most cases better than that of NSGA-III. However, as the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

6

TABLE III
BEST, MEDIAN AND WORST IGD AND GD METRIC VALUES OBTAINED FOR NSGA-III AND C-MOEA/D ON M-OBJECTIVE C1-DTLZ1 AND C1-DTLZ3 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD. IN CASES WHERE ALGORITHM GOT STUCK IN LOCAL PARETO-FRONT THE CORRESPONDING IGD VALUE IS NOT SHOWN INSTEAD THE NUMBER OF SUCCESSFUL RUNS OUT OF 20 ARE SHOWN IN BRACKETS.

Problem	M	MaxGen	NSGA-III		C-MOEA/D	
			IGD	GD	IGD	GD
C1-DTLZ1	3	500	1.229×10^{-3} $\mathbf{4.932 \times 10^{-3}}$ $\mathbf{2.256 \times 10^{-2}}$	1.266×10^{-3} $\mathbf{4.989 \times 10^{-3}}$ $\mathbf{2.222 \times 10^{-2}}$	$\mathbf{6.430 \times 10^{-4}}$ 6.817×10^{-3} 2.461×10^{-2}	$\mathbf{6.430 \times 10^{-4}}$ 6.817×10^{-3} 2.307×10^{-2}
	5	600	2.380×10^{-3} 4.347×10^{-3} $\mathbf{1.024 \times 10^{-2}}$	2.951×10^{-3} 4.727×10^{-3} $\mathbf{1.051 \times 10^{-2}}$	$\mathbf{8.686 \times 10^{-4}}$ $\mathbf{3.637 \times 10^{-3}}$ 1.224×10^{-2}	$\mathbf{8.686 \times 10^{-4}}$ $\mathbf{3.637 \times 10^{-3}}$ 1.224×10^{-2}
	8	800	4.843×10^{-3} 1.361×10^{-2} 4.140×10^{-2}	4.843×10^{-3} 1.361×10^{-2} 4.140×10^{-2}	$\mathbf{4.019 \times 10^{-3}}$ $\mathbf{1.142 \times 10^{-2}}$ $\mathbf{2.380 \times 10^{-2}}$	$\mathbf{4.019 \times 10^{-3}}$ $\mathbf{1.142 \times 10^{-2}}$ $\mathbf{2.380 \times 10^{-2}}$
	10	1000	$\mathbf{3.042 \times 10^{-3}}$ $\mathbf{6.358 \times 10^{-3}}$ 2.762×10^{-2}	3.394×10^{-3} 6.636×10^{-3} 2.806×10^{-2}	3.271×10^{-3} 6.412×10^{-3} $\mathbf{1.747 \times 10^{-2}}$	$\mathbf{3.271 \times 10^{-3}}$ $\mathbf{6.412 \times 10^{-3}}$ $\mathbf{1.747 \times 10^{-2}}$
	15	1500	$\mathbf{4.994 \times 10^{-3}}$ $\mathbf{1.041 \times 10^{-2}}$ $\mathbf{2.930 \times 10^{-2}}$	$\mathbf{5.422 \times 10^{-3}}$ $\mathbf{1.098 \times 10^{-2}}$ $\mathbf{2.988 \times 10^{-2}}$	8.800×10^{-3} 1.258×10^{-2} 3.037×10^{-2}	8.800×10^{-3} 1.258×10^{-2} 3.037×10^{-2}
C1-DTLZ3	3	1000	8.649×10^{-4} $\mathbf{8.139 \times 10^{-3}}$ (13)	8.724×10^{-4} 1.008×10^{-2} $-$	$\mathbf{4.398 \times 10^{-4}}$ (8) $-$	$\mathbf{4.398 \times 10^{-4}}$ $-$ $-$
	5	1500	1.028×10^{-3} $\mathbf{5.101 \times 10^{-2}}$ (15)	1.684×10^{-3} 1.144×10^{-1} $-$	$\mathbf{2.651 \times 10^{-4}}$ (8) $-$	$\mathbf{2.651 \times 10^{-4}}$ $-$ $-$
	8	2500	$\mathbf{1.656 \times 10^{-3}}$ $\mathbf{1.196 \times 10^{-2}}$ (14)	$\mathbf{1.656 \times 10^{-3}}$ 1.423×10^{-2} $-$	4.998×10^{-1} (1) $-$	5.086×10^{-1} $-$ $-$
	10	3500	2.437×10^{-3} $\mathbf{1.445 \times 10^{-2}}$ (18)	2.834×10^{-3} 1.572×10^{-2} $-$	$\mathbf{4.710 \times 10^{-4}}$ (6) $-$	$\mathbf{4.710 \times 10^{-4}}$ $-$ $-$
	15	5000	$\mathbf{4.541 \times 10^{-3}}$ (9) $-$	$\mathbf{5.212 \times 10^{-3}}$ $-$ $-$	$-$ $-$ $-$	$-$ $-$ $-$

number of objectives increase (10 ad 15-objective problems), the performance of NSGA-III is clearly better.

Additionally, we compute the GD metric value for NSGA-III solutions and tabulate the best, median and worst values in Table III. Small GD values indicate that NSGA-III solutions are close to the true Pareto-optimal fronts in each case. Corresponding GD metric values of MOEA/D with PBI approach are also presented. GD metric values for both methods are similar, although interestingly in most cases whichever algorithm produced a better IGD value also made a smaller GD value. It is important to highlight here that GD metric indicates the convergence property of an algorithm, but cannot reveal the diversity in the solutions. On the other hand, the IGD metric indicates a combined measure of both diversity and convergence and is a more reliable metric for comparing multi-objective optimization algorithms.

Figures 5 and 6 show Pareto-optimal fronts (corresponding to median IGD value) obtained by NSGA-III and C-MOEA/D, respectively, for the three-objective C1-DTLZ3 problem. It is clear that while NSGA-III was able to reach the global Pareto-front, C-MOEA/D (the median-performed run) was not able to cross the infeasibility barrier and instead got stuck in the infeasible region. This problem is a difficult problem and Table III shows that in case of three, five and 10 objectives, although C-MOEA/D finds a better best IGD value but the success rate is less for C-MOEA/D as compared to NSGA-III. In all the cases for C-MOEA/D success rate is less than

50% while for NSGA-III it is more than 60% in all the cases except for 15 objectives where it is 45% and there C-MOEA/D was unable to reach the Pareto-front in any of the 20 runs. This problem was found to be difficult for both algorithms, but the performance of NSGA-III was found to be somewhat better than C-MOEA/D.

B. Effect of η_c on C-MOEA/D

In the above simulations with C-MOEA/D, $\eta_c = 20$ is used, simply because of the preference of this value by the original developers of MOEA/D on unconstrained problems. Since, a somewhat higher value ($\eta_c = 30$) is used with NSGA-III to have a higher probability of creating offspring solutions close to parent solutions in a higher-dimensional space, we rerun C-MOEA/D algorithm with $\eta_c = 30$ for C1-DTLZ1 problem and tabulate results in Table IV. C-MOEA/D results are not significantly different from previous C-MOEA/D results obtained with $\eta_c = 20$.

C. Constrained Problems of Type-2

While type 1 constrained problems introduced difficulties in arriving at the entire Pareto-optimal front, type-2 constrained problems are designed to introduce infeasibility to a part of the Pareto-optimal front. Such problems will test an algorithm's ability to deal with disconnected Pareto-optimal fronts. To accomplish this, DTLZ2 [24] and the convex DTLZ2 problems [1] are modified.
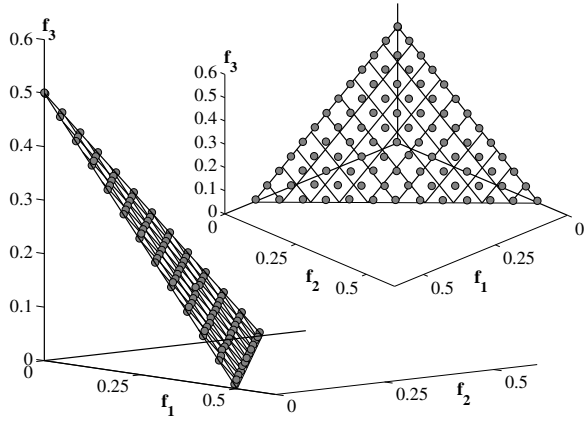
Fig. 3. Obtained solutions using NSGA-III on three-objective C1-DTLZ1 problem.
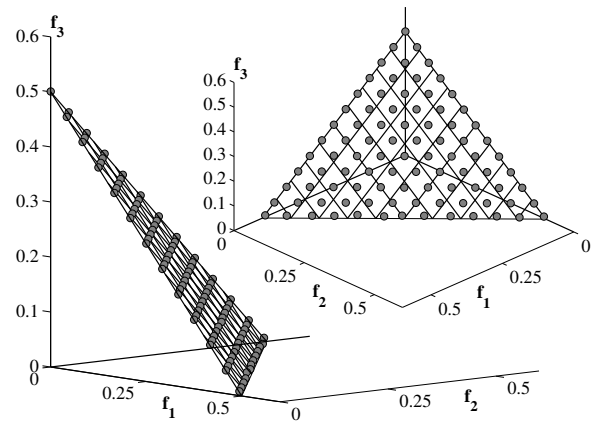

Fig. 4. Obtained solutions using C-MOEA/D approach on three-objective C1-DTLZ1 problem.
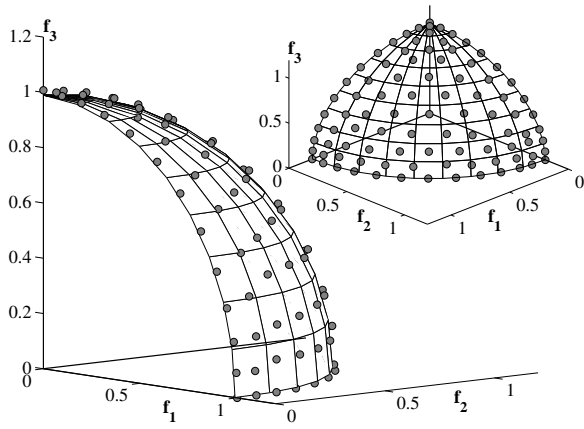

Fig. 5. Obtained solutions using NSGA-III on three-objective constrained C1-DTLZ3 problem.
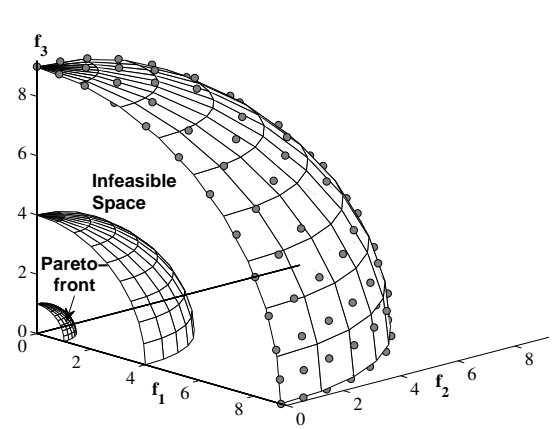

Fig. 6. Obtained solutions using C-MOEA/D on three-objective constrained C1-DTLZ3 problem.

In C2-DTLZ2 problem, only the region of objective space that lies inside each of the $M+1$ hyper-spheres of radius r is made feasible. Of $(M+1)$ hyper-spheres, M are placed at the corners of unit hyper-plane and the $(M+1)$-th is placed at the intersection of the equally-angled line with objective axes and the original Pareto-optimal front. This way, the Pareto-optimal front is disconnected, as shown in Figure 7. Objective
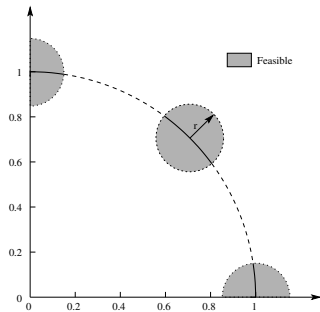

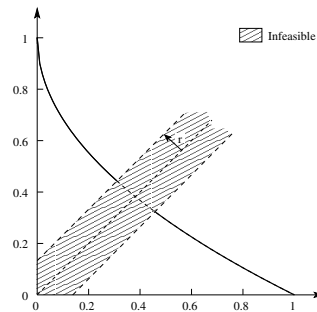Fig. 7. Two-objective version of C2-DTLZ2 problem.

Fig. 8. Two-objective version of convex C2-DTLZ2 problem.

functions are calculated in the same way as in the original

DTLZ2 problem, except that a constraint is now introduced:

$$c(\mathbf{x}) = \max \left\{ \max_{i=1}^{M} \left[(f_i(\mathbf{x}) - 1)^2 + \sum_{j=1, j \neq i}^{M} f_j^2 - r^2 \right], \left[\sum_{i=1}^{M} (f_i(\mathbf{x}) - 1/\sqrt{M})^2 - r^2 \right] \right\},$$

where $r = 0.4$, for $M = 3$ and 0.5, otherwise. For an M-objective C2-DTLZ2 problem, $k = 10$ is used, thereby having a total of $(M+9)$ variables.

For the convex C2-DTLZ2 described in [1], we construct a different feasible region. The region in the objective space lying inside a hyper-cylinder with $(1, 1, \ldots, 1)^T$ as the axis and radius r is kept infeasible, thereby creating an infeasible hole through the objective space. This also produces a hole on the Pareto-optimal front, as demonstrated for a two-objective version of convex C2-DTLZ2 problem in Figure 8. The objective functions are kept the same as before, while the following constraint is added:

$$c(\mathbf{x}) = \sum_{i=1}^{M} (f_i(\mathbf{x}) - \lambda)^2 - r^2 \geq 0, \tag{6}$$

TABLE IV
IGD AND GD VALUES WITH C-MOEA/D ALGORITHM WITH $\eta_c = 30$.

Problem	M	MaxGen	NSGA-III		C-MOEA/D	
			IGD	GD	IGD	GD
C1-DTLZ1	3	500	1.229×10^{-3}	1.266×10^{-3}	$\mathbf{9.624 \times 10^{-4}}$	$\mathbf{9.624 \times 10^{-4}}$
			$\mathbf{4.932 \times 10^{-3}}$	$\mathbf{4.989 \times 10^{-3}}$	6.360×10^{-3}	6.360×10^{-3}
			$\mathbf{2.256 \times 10^{-2}}$	$\mathbf{2.222 \times 10^{-2}}$	2.276×10^{-2}	2.226×10^{-2}
	5	600	2.380×10^{-3}	2.951×10^{-3}	$\mathbf{1.998 \times 10^{-3}}$	$\mathbf{1.998 \times 10^{-3}}$
			4.347×10^{-3}	4.727×10^{-3}	$\mathbf{3.960 \times 10^{-3}}$	$\mathbf{3.960 \times 10^{-3}}$
			1.024×10^{-2}	1.051×10^{-2}	$\mathbf{9.597 \times 10^{-3}}$	$\mathbf{9.597 \times 10^{-3}}$
	8	800	4.843×10^{-3}	4.843×10^{-3}	$\mathbf{3.442 \times 10^{-3}}$	$\mathbf{3.442 \times 10^{-3}}$
			1.361×10^{-2}	1.361×10^{-2}	$\mathbf{9.150 \times 10^{-3}}$	$\mathbf{9.150 \times 10^{-3}}$
			4.140×10^{-2}	4.140×10^{-2}	$\mathbf{3.514 \times 10^{-2}}$	$\mathbf{3.514 \times 10^{-2}}$
	10	1000	$\mathbf{3.042 \times 10^{-3}}$	$\mathbf{3.394 \times 10^{-3}}$	5.042×10^{-3}	5.042×10^{-3}
			$\mathbf{6.358 \times 10^{-3}}$	$\mathbf{6.636 \times 10^{-3}}$	7.960×10^{-3}	7.960×10^{-3}
			2.762×10^{-2}	2.806×10^{-2}	$\mathbf{1.536 \times 10^{-2}}$	$\mathbf{1.536 \times 10^{-2}}$
	15	1500	$\mathbf{4.994 \times 10^{-3}}$	$\mathbf{5.422 \times 10^{-3}}$	8.088×10^{-3}	8.088×10^{-3}
			$\mathbf{1.041 \times 10^{-2}}$	$\mathbf{1.098 \times 10^{-2}}$	1.595×10^{-2}	1.595×10^{-2}
			2.930×10^{-2}	2.988×10^{-2}	$\mathbf{2.893 \times 10^{-2}}$	$\mathbf{2.893 \times 10^{-2}}$
C2-DTLZ2	3	250	1.581×10^{-3}	$\mathbf{1.764 \times 10^{-2}}$	$\mathbf{3.844 \times 10^{-4}}$	2.690×10^{-2}
			2.578×10^{-3}	$\mathbf{1.990 \times 10^{-2}}$	$\mathbf{5.526 \times 10^{-4}}$	2.761×10^{-2}
			$\mathbf{6.733 \times 10^{-3}}$	$\mathbf{2.214 \times 10^{-2}}$	7.014×10^{-3}	5.815×10^{-2}
	5	350	2.762×10^{-3}	1.944×10^{-1}	$\mathbf{5.404 \times 10^{-4}}$	$\mathbf{1.623 \times 10^{-1}}$
			3.873×10^{-3}	1.977×10^{-1}	$\mathbf{7.304 \times 10^{-4}}$	$\mathbf{1.682 \times 10^{-1}}$
			$\mathbf{7.596 \times 10^{-3}}$	2.001×10^{-1}	3.343×10^{-2}	$\mathbf{1.687 \times 10^{-1}}$
	8	500	1.404×10^{-2}	3.576×10^{-1}	$\mathbf{2.926 \times 10^{-3}}$	$\mathbf{3.126 \times 10^{-2}}$
			2.352×10^{-2}	4.728×10^{-1}	$\mathbf{4.975 \times 10^{-3}}$	2.602×10^{-1}
			$\mathbf{8.662 \times 10^{-1}}$	5.126×10^{-1}	1.131	$\mathbf{3.690 \times 10^{-1}}$
	10	750	1.978×10^{-2}	4.637×10^{-1}	$\mathbf{9.661 \times 10^{-4}}$	$\mathbf{1.897 \times 10^{-1}}$
			2.694×10^{-2}	4.717×10^{-1}	$\mathbf{1.491 \times 10^{-3}}$	$\mathbf{1.910 \times 10^{-1}}$
			3.491×10^{-2}	4.765×10^{-1}	8.774×10^{-1}	$\mathbf{2.928 \times 10^{-1}}$
	15	1000	3.117×10^{-2}	3.420×10^{-1}	$\mathbf{1.668 \times 10^{-2}}$	$\mathbf{1.713 \times 10^{-2}}$
			3.544×10^{-2}	3.836×10^{-1}	$\mathbf{2.129 \times 10^{-2}}$	$\mathbf{3.828 \times 10^{-1}}$
			$\mathbf{9.343 \times 10^{-1}}$	$\mathbf{4.363 \times 10^{-1}}$	1.204	4.579×10^{-1}
C3-DTLZ4	3	750	1.862×10^{-2}	1.987×10^{-2}	$\mathbf{6.918 \times 10^{-3}}$	$\mathbf{6.918 \times 10^{-3}}$
			$\mathbf{2.456 \times 10^{-2}}$	$\mathbf{2.930 \times 10^{-2}}$	4.959×10^{-1}	8.371×10^{-2}
			$\mathbf{5.586 \times 10^{-1}}$	$\mathbf{9.158 \times 10^{-2}}$	7.510×10^{-1}	9.478×10^{-2}
	5	1250	3.247×10^{-2}	3.461×10^{-2}	$\mathbf{5.571 \times 10^{-3}}$	$\mathbf{5.571 \times 10^{-3}}$
			$\mathbf{3.854 \times 10^{-2}}$	$\mathbf{4.088 \times 10^{-2}}$	2.207×10^{-1}	8.007×10^{-2}
			$\mathbf{3.466 \times 10^{-1}}$	$\mathbf{1.601 \times 10^{-1}}$	7.209×10^{-1}	2.302×10^{-1}
	8	2000	5.558×10^{-2}	5.558×10^{-2}	$\mathbf{3.782 \times 10^{-1}}$	$\mathbf{2.255 \times 10^{-1}}$
			$\mathbf{2.646 \times 10^{-1}}$	$\mathbf{2.075 \times 10^{-1}}$	8.923×10^{-1}	4.277×10^{-1}
			$\mathbf{8.886 \times 10^{-1}}$	4.976×10^{-1}	1.216	$\mathbf{4.703 \times 10^{-1}}$
	10	3000	4.247×10^{-2}	4.386×10^{-2}	$\mathbf{8.395 \times 10^{-3}}$	$\mathbf{8.395 \times 10^{-3}}$
			$\mathbf{5.927 \times 10^{-2}}$	$\mathbf{6.061 \times 10^{-2}}$	7.008×10^{-1}	4.056×10^{-1}
			$\mathbf{9.092 \times 10^{-1}}$	$\mathbf{4.849 \times 10^{-1}}$	1.156	5.077×10^{-1}
	15	4000	$\mathbf{1.134 \times 10^{-1}}$	$\mathbf{1.172 \times 10^{-1}}$	1.181	3.849×10^{-1}
			$\mathbf{9.325 \times 10^{-1}}$	5.803×10^{-1}	1.450	$\mathbf{5.125 \times 10^{-1}}$
			$\mathbf{1.424}$	$\mathbf{7.206 \times 10^{-1}}$	1.651	8.992×10^{-1}

where $\lambda = \frac{1}{M}\sum_{i=1}^{M} f_i(\mathbf{x})$ and the radius $r = \{0.225, 0.225, 0.26, 0.26, 0.27\}$ for $M = \{3, 5, 8, 10, 15\}$. Total number of variables for this problem are $(M + 9)$.

We now present results of both algorithms on these two problems. Since in these problems only a part of Pareto-optimal front is feasible, there may exist some reference points/directions for which there is no corresponding feasible point on Pareto-optimal front. So while calculating the IGD metric, only the Pareto-optimal points ($\mathbf{Z}$) corresponding to *useful* reference points/directions are used. It could be that some reference points have more than one Pareto-optimal points associated with it. For IGD metric value computation and for plotting the final population, we have considered only one Pareto-optimal point that has the smallest perpendicular distance from the extended reference line corresponding to a reference point/direction.

Table V clearly shows that in the case of C2-DTLZ2 problem, both NSGA-III and C-MOEA/D give similar performance, however in all the cases (3-15 objectives) C-MOEA/D could not come close to the true Pareto-optimal front in all 20 runs (which is indicated by a large value of worst IGD values). However, despite the best performance of NSGA-III is not better than C-MOEA/D, NSGA-III performs well in all 20 runs. The table also shows the number of useful reference points out of total supplied reference points for constrained NSGA-III procedure. The rest of the reference points do not associate with any feasible Pareto-optimal solution, hence not solution is found by constrained NSGA-III for them. GD metric values are calculated and presented in the table. Interestingly, the algorithm performing better in terms of IGD metric also shows a better GD metric value.

Figures 9 and 10 show that both algorithms are able to find

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
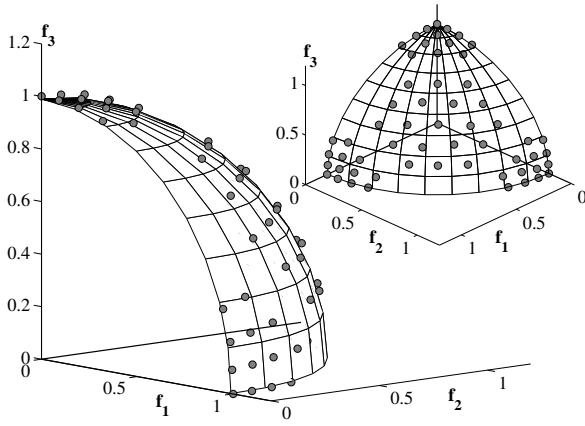
9

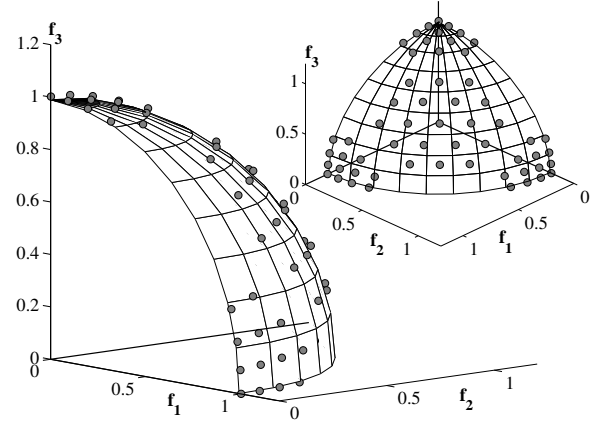Fig. 9. Obtained solutions using NSGA-III on three-objective C2-DTLZ2 problem.

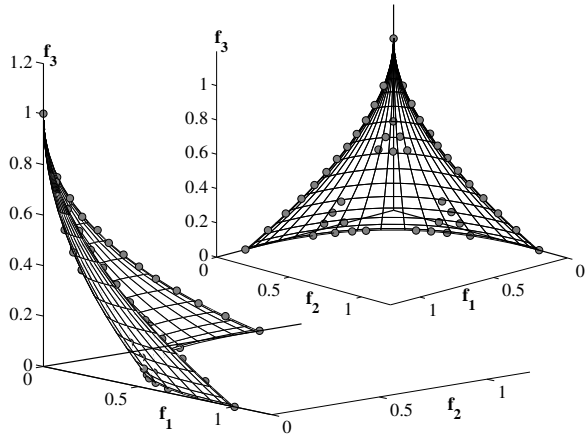Fig. 10. Obtained solutions using C-MOEA/D approach on three-objective C2-DTLZ2 problem.

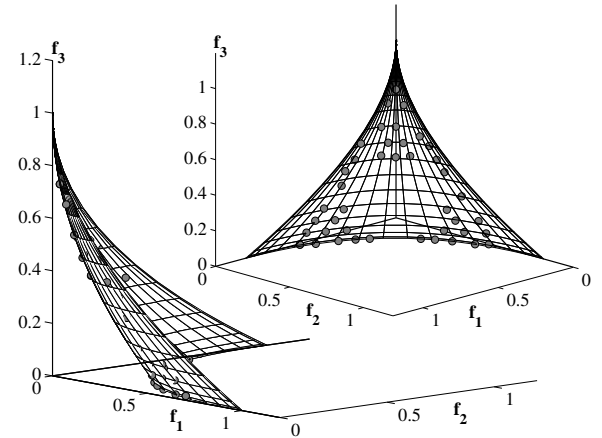Fig. 11. Obtained solutions using NSGA-III on three-objective constraint convex C2-DTLZ2 problem.

Fig. 12. Obtained solutions using C-MOEA/D approach on three-objective constraint convex C2-DTLZ2 problem.

the disconnected Pareto-optimal regions for the three-objective C2-DTLZ2 problem. Column 3 of Table V shows the number of reference points out of total supplied reference points are able to find at least one Pareto-optimal solution. We discuss more about this in Section VII.

However, in the case of convex C2-DTLZ2 problem, as can be seen from Table V, NSGA-III outperforms C-MOEA/D for three to 15-objective versions of the problem. Figure 11 shows that on a three-objective convex C2-DTLZ2 problem, NSGA-III is able to find the feasible Pareto-optimal points. No point in the intermediate infeasible portion of the front is found. However, as shown in Figure 12, although intermediate infeasible points were not found by C-MOEA/D, it could not find the points on the front boundary. A similar observation was also made while solving the convex DTLZ2 problem using the unconstrained MOEA/D algorithm [1].

D. Constrained Problems of Type-3

Type-3 problems involve multiple constraints and the entire Pareto-optimal front of the unconstrained problem need not be optimal any more, rather portions of the added constraint surfaces constitute the Pareto-optimal front. We modify DTLZ1 and DTLZ4 problems for this purpose here by adding

M different constraints. In the case of C3-DTLZ1 problem, objective functions are same as in the original formulation [24], however, following M linear constraints are added:

$$c_j(\mathbf{x}) = \sum_{i=1, i \neq j}^{M} f_j(\mathbf{x}) + \frac{f_i(\mathbf{x})}{0.5} - 1 \geq 0, \quad \forall j = 1, 2, \ldots, M. \quad (7)$$

For C3-DTLZ1 problem, $k = 5$ is used in the original g-function, thereby making a total of $(M + 4)$ variables. Figure 13 shows constraints and feasible region for the two-objective C3-DTLZ1 problem. Notice how the unconstrained Pareto-optimal front is now infeasible by the presence of two constraints.

Similarly, problem DTLZ4 is modified by adding M quadratic constraints of the type:

$$c_j(\mathbf{x}) = \frac{f_j^2}{4} + \sum_{i=1, i \neq j}^{M} f_i(\mathbf{x})^2 - 1 \geq 0, \quad \forall j = 1, 2, \ldots, M. \quad (8)$$

Another difficulty posed by DTLZ4 is that it introduces bias for creating solutions in certain parts of the objective space. For this problem, we have used $n = M + 4$ variables. Figure 13 shows the respective constraints and resulting Pareto-optimal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

10

TABLE V
BEST, MEDIAN AND WORST IGD AND GD METRIC VALUES OBTAINED FOR NSGA-III AND C-MOEA/D ON M-OBJECTIVE C2-DTLZ2 AND CONVEX C2-DTLZ2 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD. U IS THE NUMBER OF USEFUL REFERENCE POINTS THAT GENERATED A PARETO-OPTIMAL SOLUTION AND H IS THE TOTAL NUMBER OF SUPPLIED REFERENCE POINTS.

Problem	M	U/H	MaxGen	NSGA-III		C-MOEA/D	
				IGD	GD	IGD	GD
C2-DTLZ2	3	58/91	250	1.581×10^{-3}	$\mathbf{1.764 \times 10^{-2}}$	$\mathbf{4.600 \times 10^{-4}}$	2.687×10^{-2}
				2.578×10^{-3}	$\mathbf{1.990 \times 10^{-2}}$	$\mathbf{5.842 \times 10^{-4}}$	2.762×10^{-2}
				6.733×10^{-3}	$\mathbf{2.214 \times 10^{-2}}$	6.792×10^{-1}	3.208×10^{-2}
	5	80/210	350	2.762×10^{-3}	1.944×10^{-1}	$\mathbf{7.223 \times 10^{-4}}$	$\mathbf{7.539 \times 10^{-2}}$
				3.873×10^{-3}	1.977×10^{-1}	$\mathbf{8.126 \times 10^{-4}}$	1.684×10^{-1}
				7.596×10^{-3}	2.001×10^{-1}	7.621×10^{-1}	1.695×10^{-1}
	8	72/156	500	1.404×10^{-2}	3.576×10^{-1}	$\mathbf{2.291 \times 10^{-3}}$	$\mathbf{3.525 \times 10^{-2}}$
				2.352×10^{-2}	4.728×10^{-1}	$\mathbf{4.201 \times 10^{-3}}$	$\mathbf{2.637 \times 10^{-1}}$
				$\mathbf{8.662 \times 10^{-1}}$	5.126×10^{-1}	1.111	$\mathbf{3.723 \times 10^{-1}}$
	10	110/275	750	1.978×10^{-2}	4.637×10^{-1}	$\mathbf{1.454 \times 10^{-3}}$	$\mathbf{1.880 \times 10^{-1}}$
				2.694×10^{-2}	4.717×10^{-1}	$\mathbf{1.776 \times 10^{-3}}$	$\mathbf{1.917 \times 10^{-1}}$
				3.491×10^{-2}	4.765×10^{-1}	8.773×10^{-1}	2.912×10^{-1}
	15	30/135	1000	3.117×10^{-2}	3.420×10^{-1}	$\mathbf{1.659 \times 10^{-2}}$	$\mathbf{1.859 \times 10^{-2}}$
				3.544×10^{-2}	3.836×10^{-1}	$\mathbf{2.434 \times 10^{-2}}$	3.829×10^{-1}
				9.343×10^{-1}	$\mathbf{4.363 \times 10^{-1}}$	1.193	4.581×10^{-1}
C2-DTLZ2 Convex	3	47/91	250	$\mathbf{3.134 \times 10^{-3}}$	$\mathbf{2.688 \times 10^{-2}}$	5.880×10^{-2}	6.442×10^{-2}
				$\mathbf{5.857 \times 10^{-3}}$	$\mathbf{2.944 \times 10^{-2}}$	6.335×10^{-2}	7.072×10^{-2}
				$\mathbf{8.554 \times 10^{-3}}$	$\mathbf{3.320 \times 10^{-2}}$	6.561×10^{-2}	8.137×10^{-2}
	5	97/210	750	$\mathbf{6.842 \times 10^{-3}}$	$\mathbf{5.508 \times 10^{-2}}$	1.496×10^{-1}	1.081×10^{-1}
				$\mathbf{1.189 \times 10^{-2}}$	$\mathbf{5.924 \times 10^{-2}}$	1.553×10^{-1}	1.112×10^{-1}
				$\mathbf{1.615 \times 10^{-2}}$	$\mathbf{6.120 \times 10^{-2}}$	1.596×10^{-1}	1.130×10^{-1}
	8	64/156	1500	$\mathbf{1.838 \times 10^{-2}}$	$\mathbf{1.336 \times 10^{-1}}$	2.022×10^{-1}	1.806×10^{-1}
				$\mathbf{2.802 \times 10^{-2}}$	$\mathbf{1.370 \times 10^{-1}}$	2.039×10^{-1}	1.815×10^{-1}
				$\mathbf{4.208 \times 10^{-2}}$	$\mathbf{1.426 \times 10^{-1}}$	2.050×10^{-1}	1.821×10^{-1}
	10	100/275	2500	$\mathbf{1.796 \times 10^{-2}}$	$\mathbf{1.457 \times 10^{-1}}$	1.902×10^{-1}	1.872×10^{-1}
				$\mathbf{2.793 \times 10^{-2}}$	$\mathbf{1.479 \times 10^{-1}}$	1.911×10^{-1}	1.877×10^{-1}
				$\mathbf{6.489 \times 10^{-2}}$	$\mathbf{1.613 \times 10^{-1}}$	1.914×10^{-1}	1.880×10^{-1}
	15	120/135	3500	$\mathbf{1.784 \times 10^{-2}}$	$\mathbf{4.318 \times 10^{-2}}$	1.650×10^{-1}	1.231×10^{-1}
				$\mathbf{2.127 \times 10^{-2}}$	$\mathbf{4.690 \times 10^{-2}}$	1.659×10^{-1}	1.238×10^{-1}
				$\mathbf{2.940 \times 10^{-2}}$	$\mathbf{5.316 \times 10^{-2}}$	1.689×10^{-1}	1.268×10^{-1}

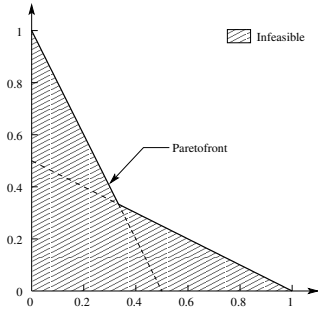
Fig. 13. Two objective version of C3-DTLZ1 problem.
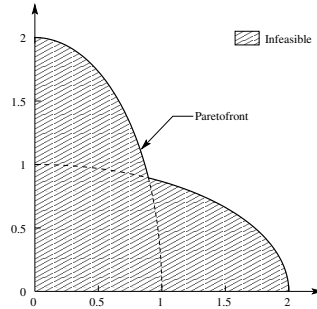
Fig. 14. Two objective version of C3-DTLZ4 problem.

problem corresponding to the run having the median IGD value. Clearly, NSGA-III is able to locate points on all three quadratic constraint surfaces, thereby finding points on the entire Pareto-optimal front. On the contrary, C-MOEA/D is not able to find a single point on two of the three constraint surfaces, as shown in Figure 18. Table VI shows that NSGA-III has performed well on all objective versions of this problem in terms of both IGD and GD metric values. C-MOEA/D, like in C3-DTLZ1 problem, is able to solve three and five-objective version for some runs, but it is not able to solve higher-objective versions of the problem very well.

E. Discussion of the Results

Based on above results on three types of constrained many-objective optimization problems, it can be concluded that the proposed NSGA-III performs fairly well in all types of problems considered in this study. It is also important to note that the successful application of NSGA-III has come without having to fix any additional parameter other than the usual genetic parameters, such as population size, operator probabilities, etc. A careful handling of infeasible and feasible solutions through the constraint-domination principle and in creating offspring population emphasize feasible and less-violated infeasible solutions. NSGA-III's non-dominated sorting, updated normalization process of objectives, association mechanism for linking a population member with a reference

front for the C3-DTLZ4 problem.

For C3-DTLZ1 problem, Table VI shows that NSGA-III outperforms C-MOEA/D mostly in the case of higher-objective problems. Figures 15 and 16 show that both algorithms are able to find a fairly uniformly distributed set of points over the entire Pareto-optimal front for three-objective C3-DTLZ1 problem. It is clear how the three constraints take their share of the Pareto-optimal front in this problem. The GD metric value for three to eight-objective C3-DTLZ1 problems is better for C-MOEA/D algorithm, whereas for higher objective problems GD metric value is better for NSGA-III.

In the case of C3-DTLZ4, Figure 17 shows the distribution of points found by NSGA-III for three-objective C3-DTLZ4

TABLE VI

BEST, MEDIAN AND WORST IGD AND GD METRIC VALUES OBTAINED FOR NSGA-III AND C-MOEA/D ON M-OBJECTIVE C3-DTLZ1 AND C3-DTLZ4 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD.

Problem	M	MaxGen	NSGA-III		C-MOEA/D	
			IGD	GD	IGD	GD
C3-DTLZ1	3	750	5.221×10^{-3}	5.437×10^{-3}	$\mathbf{1.461 \times 10^{-3}}$	$\mathbf{1.461 \times 10^{-3}}$
			9.120×10^{-3}	1.328×10^{-2}	$\mathbf{4.323 \times 10^{-3}}$	$\mathbf{3.894 \times 10^{-3}}$
			$\mathbf{2.058 \times 10^{-2}}$	3.028×10^{-1}	3.284×10^{-2}	$\mathbf{3.261 \times 10^{-2}}$
	5	1250	1.130×10^{-2}	1.152×10^{-2}	$\mathbf{5.482 \times 10^{-4}}$	$\mathbf{5.482 \times 10^{-4}}$
			1.964×10^{-2}	2.312×10^{-2}	$\mathbf{1.115 \times 10^{-2}}$	$\mathbf{7.222 \times 10^{-3}}$
			4.745×10^{-2}	5.318×10^{-2}	$\mathbf{1.713 \times 10^{-2}}$	$\mathbf{9.499 \times 10^{-3}}$
	8	2000	$\mathbf{1.243 \times 10^{-2}}$	$\mathbf{1.243 \times 10^{-2}}$	5.878×10^{-2}	2.293×10^{-2}
			$\mathbf{2.104 \times 10^{-2}}$	$\mathbf{2.207 \times 10^{-2}}$	7.817×10^{-2}	4.224×10^{-2}
			$\mathbf{8.196 \times 10^{-2}}$	1.701×10^{-1}	1.159×10^{-1}	$\mathbf{5.394 \times 10^{-2}}$
	10	3000	$\mathbf{8.450 \times 10^{-3}}$	$\mathbf{8.941 \times 10^{-3}}$	6.053×10^{-2}	2.634×10^{-2}
			$\mathbf{1.509 \times 10^{-2}}$	$\mathbf{1.671 \times 10^{-2}}$	8.968×10^{-2}	3.797×10^{-2}
			$\mathbf{3.753 \times 10^{-2}}$	$\mathbf{3.886 \times 10^{-2}}$	1.104×10^{-1}	5.321×10^{-2}
	15	4000	$\mathbf{4.042 \times 10^{-3}}$	$\mathbf{4.976 \times 10^{-3}}$	2.222×10^{-1}	1.836×10^{-1}
			$\mathbf{1.064 \times 10^{-2}}$	$\mathbf{1.206 \times 10^{-2}}$	3.769×10^{-1}	2.725×10^{-1}
			$\mathbf{2.055 \times 10^{-1}}$	$\mathbf{2.069 \times 10^{-1}}$	4.091×10^{-1}	2.822×10^{-1}
C3-DTLZ4	3	750	1.862×10^{-2}	1.987×10^{-2}	$\mathbf{5.372 \times 10^{-3}}$	$\mathbf{5.372 \times 10^{-3}}$
			$\mathbf{2.456 \times 10^{-2}}$	$\mathbf{2.930 \times 10^{-2}}$	4.948×10^{-1}	8.186×10^{-2}
			5.586×10^{-1}	$\mathbf{9.158 \times 10^{-2}}$	8.320×10^{-1}	9.658×10^{-2}
	5	1250	3.247×10^{-2}	3.461×10^{-2}	$\mathbf{6.610 \times 10^{-3}}$	$\mathbf{6.610 \times 10^{-3}}$
			$\mathbf{3.854 \times 10^{-2}}$	$\mathbf{4.088 \times 10^{-2}}$	2.195×10^{-1}	9.141×10^{-2}
			$\mathbf{3.466 \times 10^{-1}}$	$\mathbf{1.601 \times 10^{-1}}$	8.761×10^{-1}	2.607×10^{-1}
	8	2000	$\mathbf{5.558 \times 10^{-2}}$	$\mathbf{5.558 \times 10^{-2}}$	1.503×10^{-1}	1.103×10^{-1}
			$\mathbf{2.646 \times 10^{-1}}$	$\mathbf{2.075 \times 10^{-1}}$	8.171×10^{-1}	3.870×10^{-1}
			$\mathbf{8.886 \times 10^{-1}}$	4.976×10^{-1}	1.322	$\mathbf{4.675 \times 10^{-1}}$
	10	3000	$\mathbf{4.247 \times 10^{-2}}$	$\mathbf{4.386 \times 10^{-2}}$	6.414×10^{-2}	5.051×10^{-2}
			$\mathbf{5.927 \times 10^{-2}}$	$\mathbf{6.061 \times 10^{-2}}$	4.450×10^{-1}	2.940×10^{-1}
			$\mathbf{9.092 \times 10^{-1}}$	$\mathbf{4.849 \times 10^{-1}}$	1.234	5.106×10^{-1}
	15	4000	$\mathbf{1.134 \times 10^{-1}}$	1.172×10^{-1}	1.126	$\mathbf{3.225 \times 10^{-2}}$
			$\mathbf{9.325 \times 10^{-1}}$	5.803×10^{-1}	1.454	$\mathbf{5.094 \times 10^{-1}}$
			1.424	$\mathbf{7.206 \times 10^{-1}}$	1.645	8.356×10^{-1}
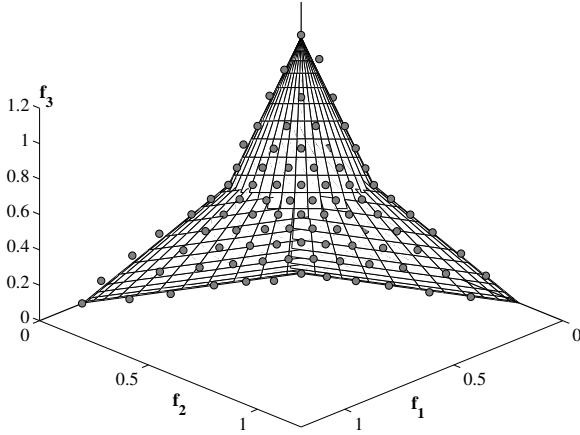
Fig. 15. Obtained solutions using NSGA-III on three-objective C3-DTLZ1 problem.
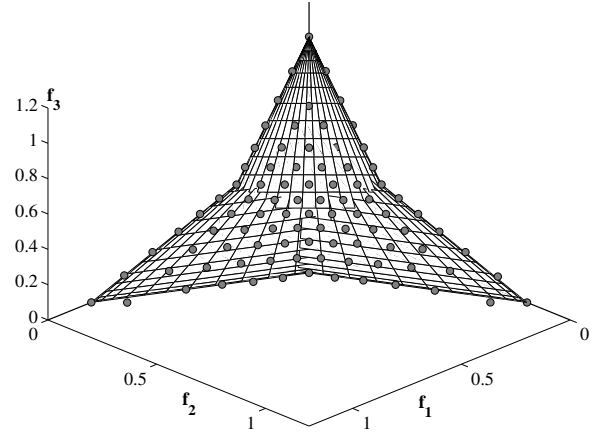
Fig. 16. Obtained solutions using C-MOEA/D approach on three-objective C3-DTLZ1 problem.

point, and the niching operator to carefully choose members from the last accepted non-dominated front are all able to produce a correct signal for providing an adequate emphasis for feasible and infeasible solutions in the population and help in progressing towards the Pareto-optimal front on all three types of constrained search regions. NSGA-III has repeatedly shown its successful performance on three to 15-objective versions of these challenging problems.

Based on the principles used for emphasizing feasible and infeasible population members in constrained NSGA-III, we

have also suggested a constrained MOEA/D algorithm (C-MOEA/D) that has also been found to perform well on most of these test problems. However, in more challenging problems, particularly having a large number of objectives ($M > 5$), it has not been able to perform as well as NSGA-III. It is worth noting here that C-MOEA/D requires *four* extra parameters to be set properly in a problem. In this study, we have used the values suggested in the original MOEA/D study [15], but a parametric study is needed to determine if some other values would allow the proposed C-MOEA/D to perform better
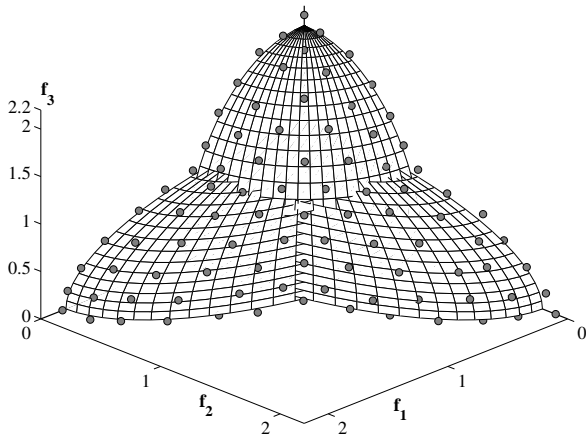
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

12

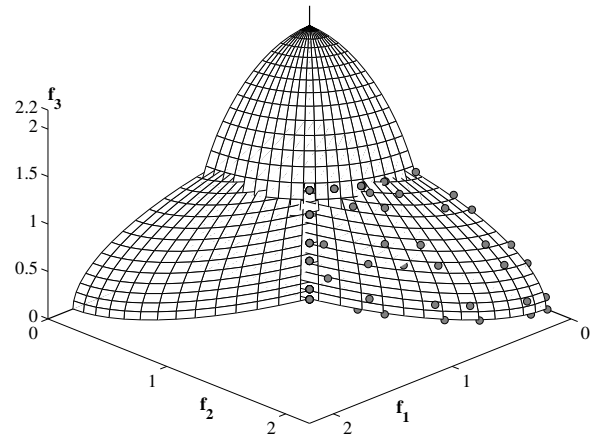Fig. 17. Obtained solutions using NSGA-III on three-objective C3-DTLZ4 problem.

Fig. 18. Obtained solutions using C-MOEA/D approach on three-objective C3-DTLZ4 problem.

on these challenging problems. For brevity, we belabor this parametric study for C-MOEA/D and leave further study with the C-MOEA/D approach and investigate how the proposed NSGA-III algorithm will perform on some practical many-objective optimization problems in the next section.

F. Engineering Constrained Optimization Problems

Having tested NSGA-III's ability in solving various kinds of constrained test problems, it is now applied to two engineering design optimization problems. The first problem has three objectives and ten constraints, while the second one has five objectives and seven constraints.

1) Car Side Impact Problem: This problem aims at minimizing the weight of car and at the same time minimize the pubic force experienced by a passenger and the average velocity of the V-Pillar responsible for withstanding the impact load. All the three objectives are conflicting, therefore, a three-dimensional trade-off front is expected. There are ten constraints involving limiting values of abdomen load, pubic force, velocity of V-Pillar, rib deflection, etc. There are 11 design variables describing thickness of B-Pillars, floor, cross-members, door beam, roof rail, etc. Mathematical formulation for the problem is given in the Appendix.

For this problem, $p = 16$ is chosen so that there are $H = \binom{3-1+16}{16}$ or 153 reference points in total. The reference points are initialized on the entire normalized hyper-plane on the three-objective space. NSGA-III is applied with 156 population members and run for 500 generations. Other parameters are kept the same as before. Out of 153 reference points 95 unique solutions corresponding to 95 reference points are found. These points are shown in Figure 19. No solution with an association to other 58 reference points is found, meaning that these reference points may not correspond to any feasible trade-off points.

These results are next tested against a classical generative method. For this purpose, 6,216 reference points are created (by taking $p = 110$). Thereafter, the ideal and the nadir points are estimated using the 95 solutions obtained by NSGA-III. The ideal and the nadir points are then used to normalize the objectives. Now, corresponding to each of 6,216 reference
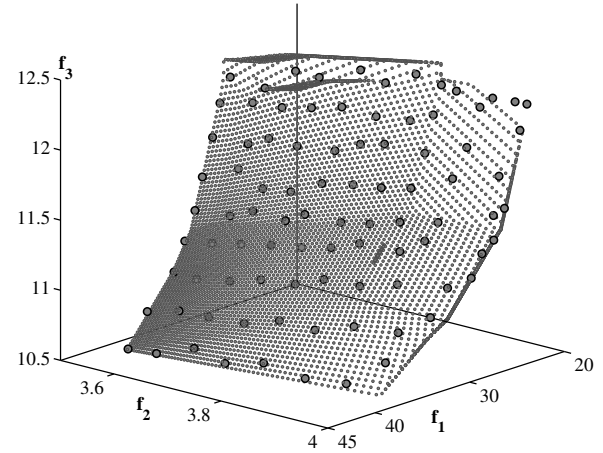
Fig. 19. 95 unique solutions are found on the entire front on the three-objective car-side impact problem.

points, the PBI metric is minimized using Matlab's fmincon routine, which uses a classical single-objective constrained optimization method. The resulting 6,216 points are then collected and dominated points are removed. Remaining non-dominated points are then shown in Figure 19 in small dots. The figure clearly shows that all the points found by NSGA-III are nicely distributed over the entire surface formed by the classical generative procedure. To investigate the closeness of NSGA-III points with that obtained by the classical generative procedure, the convergence metric (average distance of NSGA-III points from the closest fmincon optimized points) is computed and the minimum, median and maximum convergence metric values are found to be $9.80(10^{-4})$, $1.10(10^{-3})$, and $1.30(10^{-3})$, respectively. These values are small and they clearly indicate that NSGA-III is able to converge close to the true trade-off front of this problem. The spread of solutions is demonstrated visually through Figure 19.

We have observed that in solving the constrained test problems of Type 2 (refer Table V), not all reference points resulted in a feasible solution. We observed a similar phenomenon occurring in a practical optimization problem in the original study [1] as well. Applying an algorithm on carefully designed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

13

test problems help evaluate the algorithm's performance on challenging problems before they are applied to a practical problem. Since NSGA-III performed fairly well on such test problems, it is interesting to note that NSGA-III is able to solve a practical problem exhibiting a similar challenge. However, if many problems in practice have such a property, some of NSGA-III's effort would go waste in trying to find a solution corresponding to a reference point that do not end up creating any feasible solution. We address this issue later in Section VII and suggest an adaptive NSGA-III algorithm for automatically identifying such non-useful reference points.

2) Water Problem: This is a five objective problem taken from literature [2], [25]. There are three design variables and seven constraints. Mathematical formulation for the problem is given in the appendix.

210 reference points are created using $p = 6$ and NSGA-III algorithm (with $N = 212$) is ran for 1,000 generations for this problem. To show a five-objective trade-off front, we first identify the ideal and nadir points from a set of NSGA-III points obtained with 20 different runs from different initial populations. Then, the objective values are normalized and presented on a value path plot in Figure 20. The figure shows that NSGA-III is able to find 210 well-distributed set of trade-off points.
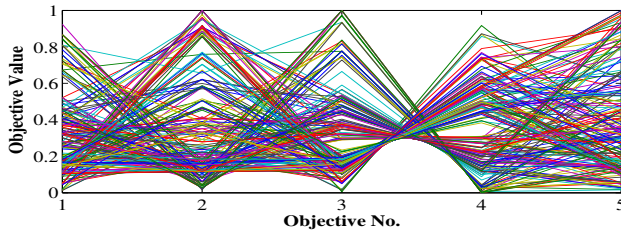
Fig. 20. A value path plot for the five-objective water problem shows that NSGA-III is able to find a well-distributed set of trade-off solutions.

To investigate the near-optimality of constrained NSGA-III solutions, we solve the same problem using Matlab's fmincon method in a generative manner for 4,845 reference points one at a time. Of them, 4,503 obtained trade-off solutions are found to be feasible. These fmincon solutions and the constrained NSGA-III solutions are shown in the scatter matrix plot in Figure 21. The lower left plots are shown for fmincon results and the upper right plots are for constrained NSGA-III solutions. For convenience of comparison of two similar plots, the (i, j)-th $(i > j)$ should be compared with (j, i)-th plot, for which the axes are interchanged for an easier viewing. Notice that the 210 solutions found by the constrained NSGA-III are widely distributed on the entire Pareto-optimal front.

VI. CONSTRAINED NSGA-III WITH PREFERRED REFERENCE POINTS

So far, we have demonstrated constrained NSGA-III's ability to find a well-distributed set of points on the entire Pareto-optimal front. For this purpose, we started with a set of reference points that are uniformly distributed on the normalized hyper-plane using Das and Dennis's structured approach [26]. However, in some practical scenario and for the purpose of
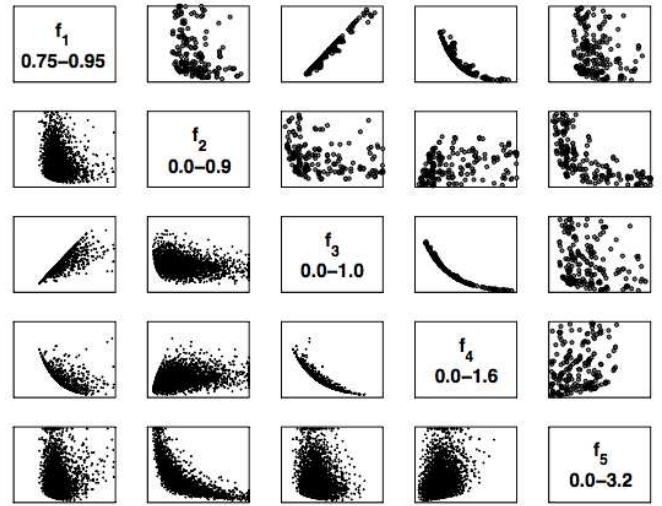
Fig. 21. A scatter plot showing constrained NSGA-III results (top-right plots) vis-a-vis classical generative results obtained using fmincon (bottom-left plots).

decision-making, only a few solutions (≈ 10 or so) may be desired to be found on a preferred part of the Pareto-optimal front. We demonstrate here the constrained NSGA-III's ability for such a purpose.

In the case of finding a preferred set, the user will supply a set of preferred reference points (H_p) in the region of his/her preference. In addition, we include M extreme reference points $(1, 0, 0, \ldots, 0)^T$, $(0, 1, 0, \ldots, 0)^T$ and so on, to make the normalization process to work well and make a total of $|H_p| + M$ reference points (set H). These extreme points are needed to ensure the ideal and nadir points of the population members are properly calculated for the normalization purpose in the NSGA-III algorithm. We then run the NSGA-III algorithm as it is.

First, we solve the constrained type 1 DTLZ1 problem (C1-DTLZ1) introduced in this paper. Recall that this problem introduces difficulty for an algorithm to approach the Pareto-optimal front, as only the region close to the Pareto-optimal front is feasible. Only five reference points (H_p) are chosen in the middle of the normalized hyper-plane, as shown in Figure 22. Three more extreme points are added to make a total of eight reference points (set H). The figure shows a typical outcome of the NSGA-III procedure run with 48 population members for 750 generations and using the reference set H. After eight solutions are obtained, only the ones corresponding to the supplied preferred reference points (H_p) are reported. However, if the user would like to know the extreme Pareto-optimal points, they can also be reported, as these solutions already exist in the final population.

Since C1-DTLZ1 problem is scalable in terms of number of objectives, next we try a 10-objective version of C1-DTLZ1 problem. In this case, only 10 preferred reference points are randomly generated in the intermediate portion ($f_i \in [0.4, 0.6]$). 10 extreme points are added to the set in creating the reference point set H and constrained NSGA-III procedure is applied with a population size 100 and run for
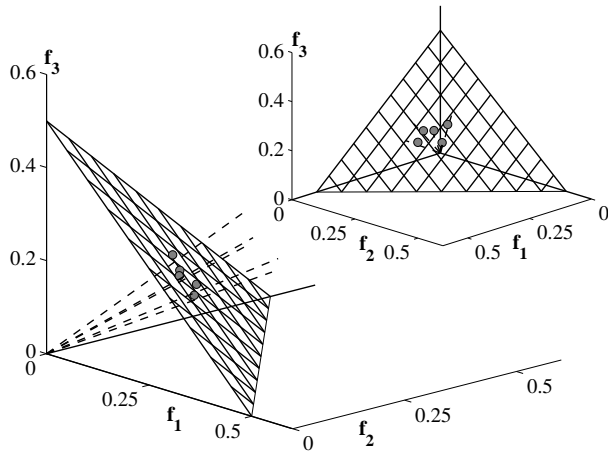
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

14

Fig. 22. A preferred set of reference points find corresponding Pareto-optimal solutions for C1-DTLZ1 problem.
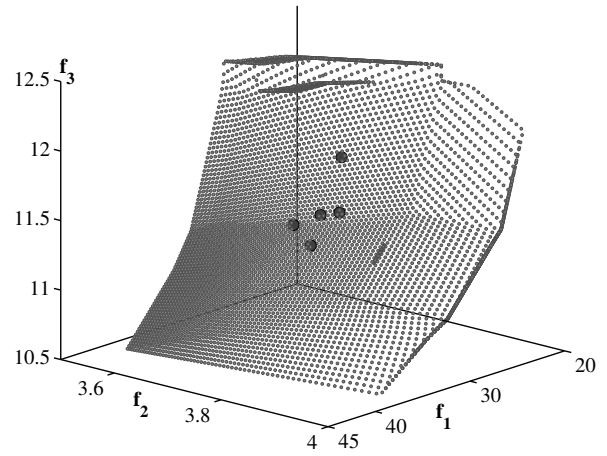

Fig. 23. Five preferred set of reference points find five corresponding trade-off solutions for the car side impact problem. Points shown with dots are obtained earlier using classical fmincon method.

TABLE VII
BEST, MEDIAN AND WORST IGD VALUES FOR C1-DTLZ1 PROBLEM
WITH RANDOMLY SUPPLIED REFERENCE POINTS.

Problem	M	N	MaxGen	NSGA-III
C1-DTLZ1				3.100×10^{-3}
	3	48	750	1.420×10^{-2}
				8.69×10^{-1}
				8.100×10^{-3}
	10	100	1500	2.390×10^{-2}
				4.762×10^{-1}

1,500 generations. Table VII shows the IGD values obtained using the expected Pareto-optimal points from the reference points and the NSGA-III obtained points. Small IGD values indicate that near-Pareto-optimal points are obtained in the case of 10-objective problem.

Next, we apply the preference-based NSGA-III procedure to the car side impact problem discussed in Subsection V-F1. In this case as well, we specify five reference points at the intermediate portion of the normalized hyper-plane ($f_i \in [0.4, 0.6]$). The constrained NSGA-III procedure is run with 28 population members for 500 generations and with a total of (5+3) or 8 reference points. Figure 23 shows the obtained solutions on the trade-off front obtained using Matlab's fmincon procedure (discussed earlier). It can be seen that obtained solutions lie on the trade-off frontier.

VII. A-NSGA-III: AN ADAPTIVE APPROACH TO NSGA-III

NSGA-III requires a set of reference points to be supplied before the algorithm can be applied. We have suggested the following in the original NSGA-III study. If the user has no particular preference on a particular part of the Pareto-optimal front, a structured set of points can be created by using Das and Dennis's approach [26] on a normalized hyper-plane — a hyper-plane that is equally inclined to all objective axes and intersects each axis at one. For a three-objective problem, this means that the supplied reference points are spread uniformly over the triangle with its apexes at $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$. A reference line for each reference point can be defined as a line joining the origin and the reference point.

NSGA-III is designed to find Pareto-optimal points that are closer to each of these reference points in the sense that their perpendicular distance from the extended reference line is minimum. Now, in many constrained or even unconstrained problems, not every extended reference line may intersect with the Pareto-optimal front. Thus, there will be some reference points with no Pareto-optimal point associated with them while others will have more than one point associated with them, and hence NSGA-III may not end up distributing all population members uniformly over the entire Pareto-optimal front. We have witnessed this in certain problems in the original study [1] while solving the practical problems and also in Section V-F of this study.

To illustrate further, let us consider the three-objective inverted DTLZ1 problem (which we describe later in Subsection VIII-A) shown in Figure 24. The corresponding Pareto-
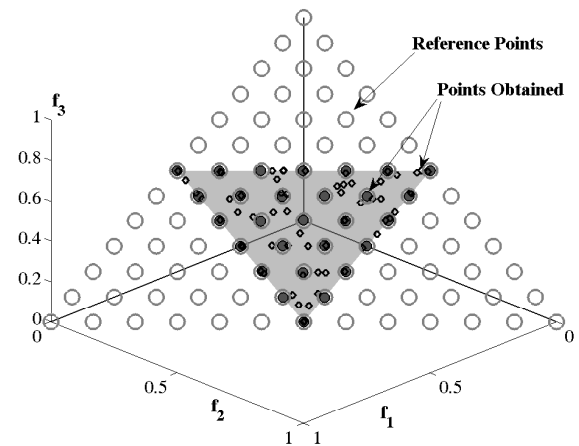

Fig. 24. Only 28 out of 91 reference points find a Pareto-optimal solution.

optimal front is shown as a shaded triangle. If we use a set of structured reference points (shown with open circles), they will lie uniformly on the normalized triangle (hyper-plane). It is clear that the reference lines originating from many of these reference points do not intersect with the Pareto-optimal front.

When optimized, these reference points will end up associated with no Pareto-optimal point. We call these reference points as 'useless' reference points and those that will generate a Pareto-optimal points are called 'useful' reference points, for our discussions here. To discuss further, we use 91 reference points (with $p = 12$) as shown in the Figure 24, and run the original NSGA-III procedure [1] with a population size of 92 to find the Pareto-optimal points. As shown in the figure, 28 Pareto-optimal points (shown in big solid circles) are found from 28 different useful reference points, but the remaining 63 reference points could not associate a Pareto-optimal solution. The solutions marked in small open circles are duplicate solutions to the 28 useful reference points. Since location of these solutions are not used in any significant way in the algorithm, their distribution is somewhat random. The presence of these random points makes the distribution of the final population non-uniform, but importantly it causes a waste in computational efforts in processing these solutions.

One possible remedy to this problem is to increase the number of supplied reference points (H) by increasing p, so that relatively more points can now appear on the Pareto-optimal front, but this is particularly not a viable suggestion, as this will require a larger population size yielding more computational efforts and still there will be several population members which are just randomly distributed over Pareto-optimal front. Ideally, it would be good to allocate all reference points in a such a manner so as to generate a uniformly distributed set of Pareto-optimal points, but the knowledge of which reference points will create Pareto-optimal solutions in an arbitrary problem is not known a priori. To alleviate the difficulty, we suggest here an adaptive NSGA-III procedure to adaptively identify non-useful reference points and re-allocate them in the hope of creating a Pareto-optimal solution for each of them at the end. There are two modifications made on the NSGA-III procedure after the new population P_{t+1} of size N is created:

1) Addition of new reference points, and
2) Deletion of existing reference points.

We describe the criterion and modus operandi of each of these two operations in the following subsections.

A. Addition of Reference Points

Note that after the niching operation, P_{t+1} population is created and the niche count ρ_j (the number of population members that are associated with j-th reference point) for each reference point is updated. Recall that the population size (N) is kept more or less equal to the number of chosen reference points (H). Thus, it is expected that if all reference points are useful in finding a non-dominated point, $\rho_j = 1$ for all reference points. But if $\rho_j \geq 2$ is observed for any (j-th) reference point (considered crowded), this has probably happened at the expense of some other reference point (say k-th one) for which $\rho_k = 0$. If k-th reference point is supposed to be a useful one, the NSGA-III procedure will eventually find an associated population member for it. But if k-th reference point is useless, then NSGA-III will never find an associated population member. It is then better to replace the k reference

point with a new reference point close to the crowded j-th reference point. However, we do not have a priori knowledge about the eventual usefulness of the k-th reference point. In this case, we simply add a set of reference points centering around the crowded j-th reference point. The procedure is described in Figure 25.
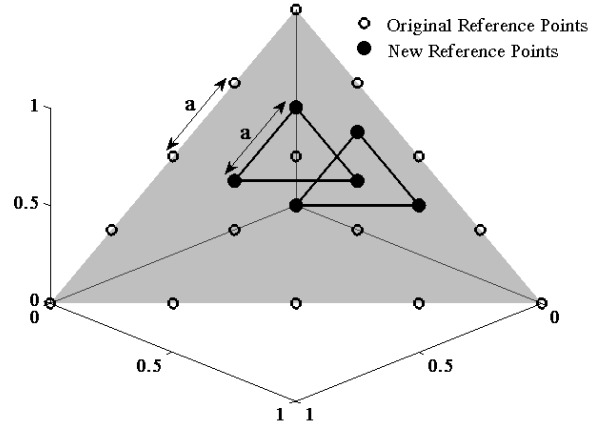

Fig. 25. Addition of reference points.

We simply introduce a simplex of M points (obtained using $\binom{M+p-1}{p}$ with $p = 1$) having a distance between them same as the distance between two consecutive reference points on the original simplex. For example, for $M = 3$ objectives, three new points will be added around the j-th reference point, as shown in the figure. If there are more than one reference points for which $\rho_j \geq 2$, the above inclusion step is executed for each of these reference points. Before a new reference point is accepted, two checks are made: (i) if it does not lie on the first quadrant, it is not accepted, and (ii) if it already exists in the set of reference points, it is not accepted. The j-th reference point is not allowed to have another inclusion operation until all original reference points have a chance to be operated for inclusion as above.

As it may have become clear from above discussion that in many cases it might happen that too many reference points are added and many of them eventually become non-useful, that is no population member is associated with them. Also if too many reference points exist, they will slow down the algorithm. Thus, we consider the possibility of deleting non-useful reference points, as described in the following subsection.

B. Deletion of Reference Points

After the inclusion operation is performed, the niche count of all reference points are updated. Note that $\sum_{j=1}^{|H|} \rho_j = N$. Now, if there exists exactly N reference points having $\rho_j = 1$, that is, we have a scenario in which N of the reference points have one associated member from P_{t+1}, we have a perfect scenario in which points are well-distributed among the reference points. We then arrange to delete all included reference points (excluding the original reference points) having $\rho_j = 0$. Thus, the original reference points are always kept (even if their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

16

niche count is zero) and all those included reference points that have an niche count exactly one.

The inclusion and deletion operations adaptively relocate reference points based on the niche count values of the respective reference points. We now show the results of this adaptive NSGA-III algorithm to a number of challenging problems including a couple of practical optimization problems.

VIII. RESULTS WITH A-NSGA-III

First, we consider two test problems – the first problem is an unconstrained inverted DTLZ1 problem and the second problem is a constrained type 2 problem. For both problems, three and five-objective versions are tried and the hypervolume indicator is used as a performance measure.

A. Inverted DTLZ1 Problem

The DTLZ1 problem is modified so that the corresponding Pareto-optimal front is inverted. The problem is such that minimum of each objective function has a unique solution. It is called an inverted function because it is in disagreement with our defined hyper-plane for which the maximum (and not minimum) point of each objective among all points on the hyper-plane is a unique point. One feature of this problem is that many reference points on created on the normalized hyper-plane will not have an associated Pareto-optimal point. Therefore, the use of A-NSGA-III may turn out to be an useful algorithm.

The objective functions are calculated using the original formulation of DTLZ1 problem, however after calculating the objective function values, the following transformation is made:

$$f_i(\mathbf{x}) \leftarrow 0.5(1 + g(\mathbf{x})) - f_i(\mathbf{x}), \quad (9)$$

where $g(\mathbf{x})$ was defined in DTLZ1 formulation [24].

Three and five-objective versions of this problem are solved using A-NSGA-III. To show the usefulness of the adaptive method, if any, we also compare it with the original NSGA-III procedure, where no update of reference points is made. In case of three-objective inverted DTLZ1 problem, NSGA-III is able to find only 28 well-distributed points (as shown in Figure 26). Here, for visual clarity, only the closest population member for each useful reference point (having $\rho > 0$) is shown, while rest members which are just randomly distributed are not shown. Although 91 reference points were supplied, only 28 of them could find a representative Pareto-optimal point and hence out of 92 population members only 28 are well distributed. Figure 27 shows the distribution of solutions with A-NSGA-III. 81 well-distributed points are now found. This is a remarkable performance of our proposed adaptive procedure.

Next, we show the results on five-objective inverted DTLZ1 problem using the hyper-volume metric. Table VIII shows the best, median and worst hyper-volume values obtained in 20 runs for both the algorithms. Clearly, the use of the adaptive approach is able to find an increased hyper-volume value in both three and five-objective cases. Both NSGA-III and A-NSGA-III approaches use an identical number of population

TABLE VIII
BEST, MEDIAN AND WORST HYPER-VOLUME VALUE OBTAINED FOR NSGA-III AND A-NSGA-III ON M-OBJECTIVE INVERTED DTLZ1 AND C2-DTLZ2 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD.

M	MaxGen	NSGA-III	A-NSGA-III
Inverted DTLZ1 Problem			
3	400	9.845×10^{-2}	$\mathbf{1.010 \times 10^{-1}}$
		9.722×10^{-2}	$\mathbf{9.907 \times 10^{-2}}$
		9.598×10^{-2}	$\mathbf{9.852 \times 10^{-2}}$
5	600	3.011×10^{-2}	$\mathbf{3.144 \times 10^{-2}}$
		2.950×10^{-2}	$\mathbf{3.014 \times 10^{-2}}$
		2.861×10^{-2}	$\mathbf{2.975 \times 10^{-2}}$
Constrained DTLZ2 Problem			
3	250	4.374×10^{-1}	$\mathbf{4.533 \times 10^{-1}}$
		4.347×10^{-1}	$\mathbf{4.464 \times 10^{-1}}$
		4.324×10^{-1}	$\mathbf{4.438 \times 10^{-1}}$
5	350	4.838×10^{-2}	$\mathbf{5.839 \times 10^{-2}}$
		4.716×10^{-2}	$\mathbf{5.697 \times 10^{-2}}$
		4.475×10^{-2}	$\mathbf{4.983 \times 10^{-2}}$

members for computing the hyper-volume metric. In the case of NSGA-III, some reference points may have more than one associated population members. Since only the one closest to each reference point is used for the niching purpose, other associated population members may not be well-distributed on their own. On the other hand, A-NSGA-III reallocates non-useful reference points in a structured manner so that each reference point can find an associated population member for obtaining a better diversity of points. Thus, the hyper-volume metric value is better for A-NSGA-III.

B. Type-2 constrained DTLZ2 Problem (C2-DTLZ2)

As mentioned in Section V-C, in C2-DTLZ2 problem, the Pareto-optimal front is disconnected, that is, there are natural gaps in Pareto-optimal front. Thus, there may exist some original reference points with no associated Pareto-optimal point. In such a case, not all points obtained by NSGA-III will be well-spread, however as discussed above, with the adaptive approach a better distribution of points can be achieved.

A set of 91 reference points are supplied initially. Figure 29 clearly shows that on three-objective C2-DTLZ2 problem, A-NSGA-III finds 91 well-distributed set of points on the feasible part of the Pareto-optimal front, whereas the constrained NSGA-III finds only 58 such points (Figure 28). The superior performance of A-NSGA-III is also clear from the larger hyper-volume value depicted in Table VIII for three and five-objective inverted DTLZ1 and constrained DTLZ2 problems.

C. Two Problems from Practice

Having shown the improved performance of A-NSGA-III on two three and five-objective test problems, we now test the method to a couple of practical optimization problems.

1) Crash-worthiness in Design of Vehicles: This is a three-objective unconstrained problem considered in [1]. NSGA-III results are re-plotted here for convenience in Figure 30. Although 91 reference points were chosen, only 40 of them are able to find associated trade-off solutions. We now apply A-NSGA-III with an identical parameter setting. As shown in Figure 31, 83 unique solutions are now found. A comparison
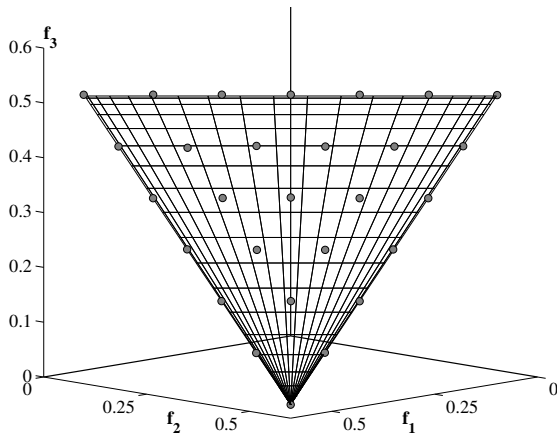
Fig. 26. Obtained solutions using NSGA-III on three-objective inverted DTLZ1 problem (only the closest solution for every useful reference point is shown).
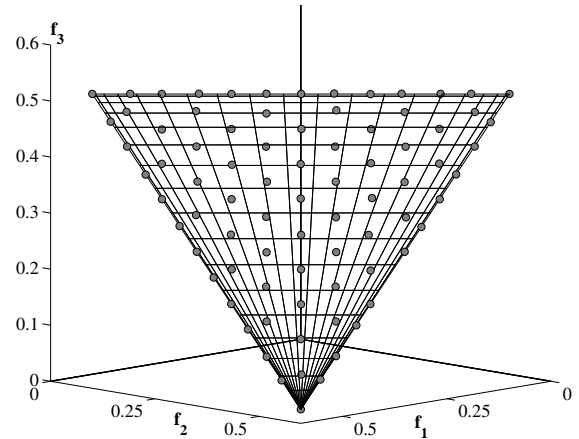

Fig. 27. Obtained solutions using adaptive reference points based NSGA-III approach on three-objective inverted DTLZ1 problem (only the closest solution for every useful reference point is shown).
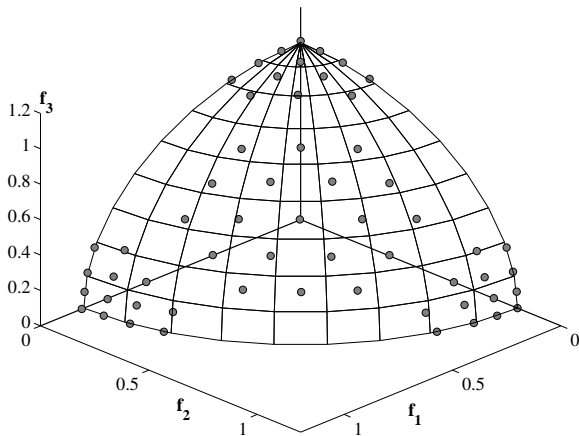

Fig. 28. Obtained solutions using NSGA-III on three-objective C2-DTLZ2 problem (only the closest representative feasible point for each reference point is shown).
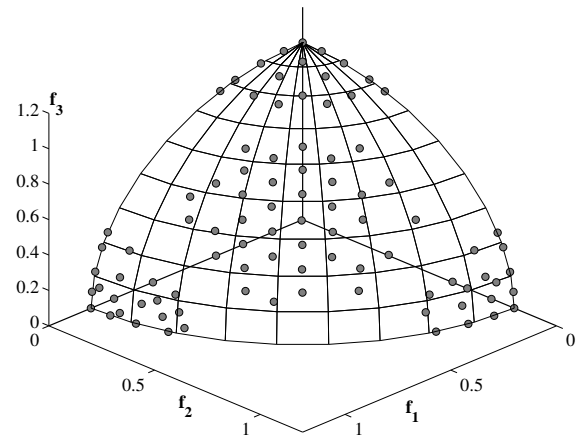
Fig. 29. Obtained solutions using A-NSGA-III approach on three-objective C2-DTLZ2 problem (only the closest representative feasible point for each reference point is shown).

of the two figures reveal that the points found by A-NSGA-III are more dense and describes the nature of the trade-off front more clearly than that with NSGA-III mainly due to former's ability to re-allocate reference points more uniformly on the fly. It is worth mentioning that A-NSGA-III does not require any additional parameter than what are needed in NSGA-III. A-NSGA-III adaptively re-allocates the supplied reference points so that more feasible trade-off points can be discovered.

2) Car-side Impact Problem: This is a constrained optimization problem and was discussed in Section V-F. The trade-off front obtained earlier had a different shape than the chosen normalized reference plane on which the reference points are supplied. The obtained front is reproduced here in Figure 32. 95 of 156 reference points could find a feasible trade-off solution using NSGA-III.

Identical parameter values are used with A-NSGA-III and obtained trade-off front is shown in Figure 33. Now, all 156 reference points find an associated trade-off point. A comparison of the two figures reveals that A-NSGA-III points are more dense and provide a better picture of the trade-off front than that obtained using NSGA-III. Importantly, all 156 reference

points were considered in NSGA-III in all generations and the procedure has resulted in only 95 feasible and well-distributed trade-off points, whereas by processing 156 reference points, A-NSGA-III is able to find 156 different and well-distributed set of trade-off points. These results amply show the usefulness of A-NSGA-III approach for solving practical many-objective optimization problems.

Above results on the test problems and two practical problems clearly show the efficacy of the adaptive NSGA-III approach. With an increase in number of objectives, real-world problems are likely to possess complicated fronts, as it is evident from several practical test problems used in the original study [1] and in this study. Thus, in such cases and in cases where no information about the shape, orientation, discontinuity, convexity etc. of the Pareto-optimal front is known beforehand, the A-NSGA-III approach may be found to be useful.

IX. CONCLUSIONS

In this paper, we have extended the recently proposed NSGA-III approach for solving many-objective optimization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
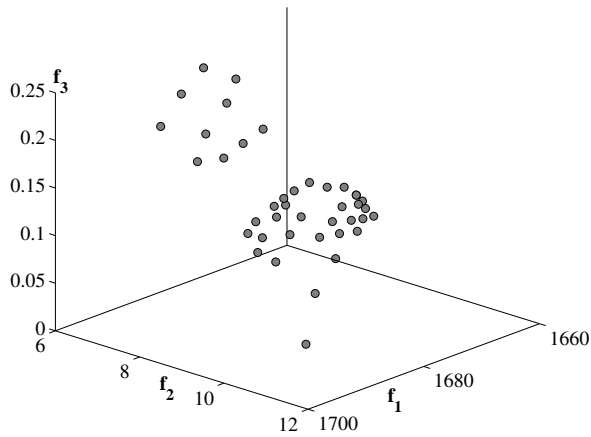
18

Fig. 30.   Obtained solutions using NSGA-III on the three-objective crash-worthiness problem.
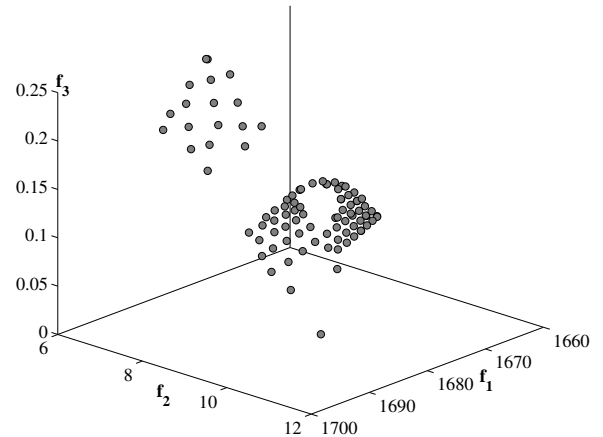

Fig. 31.   Obtained solutions using A-NSGA-III approach on the three-objective crash-worthiness problem.
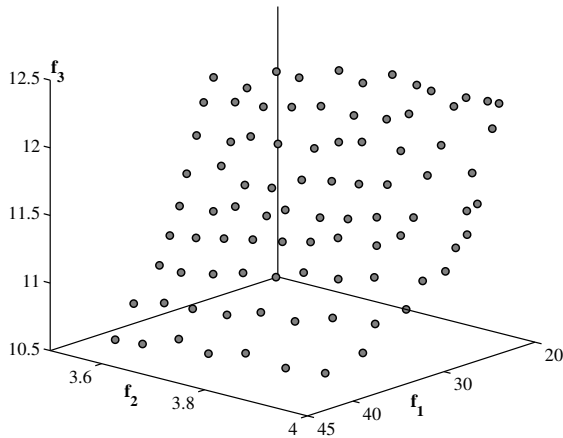

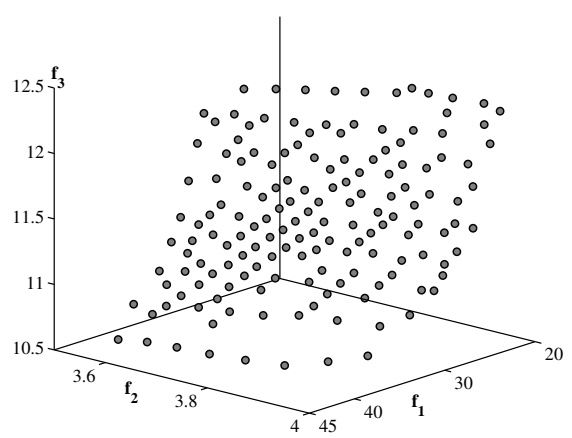Fig. 32.   Obtained solutions using NSGA-III on three-objective car-side impact problem.

Fig. 33.   Obtained solutions using adaptive reference points based NSGA-III approach on three-objective car-side impact problem.

problems with box constraints to generic many-objective constrained optimization problems. The constraint-domination principle, instead of the usual domination principle, has been suggested for classifying population members into different non-dominated fronts. Furthermore, a modified tournament selection operator has been applied along with recombination and mutation operators for generating the offspring population. The constrained algorithm is such that when there is no constraint in a problem or when all population members are feasible in a particular generation, the approach is identical to the original NSGA-III approach developed for solving box-constrained problems in [1]. Like the original NSGA-III approach, the proposed constrained NSGA-III algorithm does not require any additional parameter. This remains as a hallmark property of the proposed algorithm.

Additionally, this study has suggested an efficient extension of MOEA/D algorithm for handling constraints. Although our proposed constraint handling part of the algorithm does not require any parameter, C-MOEA/D's other operations require four parameters. To evaluate both constraint handling algorithms, this paper has also suggested three types of test problems providing three different kinds of challenges to any many-objective constraint handling algorithm. Both algorithms

have shown their ability to solve most of the test problems involving three to 15 objectives, but the proposed C-MOEA/D algorithm has shown its weakness in solving problems having a large number of objectives. A parametric study is then needed to improve its performance. However, on most problems the proposed constrained NSGA-III has been able to converge and find a well-distributed set of points up to 15-objective problems.

As a by-product of the development of the above two algorithms, this paper has also suggested three different types of scalable constrained test problems for many-objective optimization. Hopefully, these test problems and their subsequent modifications will offer adequate challenges to many-objective EMO algorithms in the years to come.

Based on the success of NSGA-III on test problems, it is then applied to two engineering design problems involving three and five objectives. In both cases, NSGA-III is able to find a well-distributed set of trade-off solutions. The constrained NSGA-III algorithm has also been shown to perform satisfactorily on problems where a few preferred reference points are supplied, thereby suggesting the practical usefulness of the constrained NSGA-III algorithm.

It has been observed that reference point or reference

direction based approaches may, in some problems, fail to find a unique associated Pareto-optimal solution for each supplied reference points, particularly if theoretically there does not exist such a Pareto-optimal point for a reference point. This feature has been found to be commonly present in real-world many-objective constrained problems. We have then suggested an adaptive NSGA-III approach which adaptively adds and deletes reference points depending on the crowding of population members on different parts of the current non-dominated front. The approach has been tested on two different types of problems having three and five objectives. In all problems, the proposed A-NSGA-III has been found to discover more and well-distributed points on the Pareto-optimal fronts. An application of A-NSGA-III to two engineering design problems have also shown the usefulness of the proposed adaptive approach.

With the two-part study ([1] and this paper) presenting the development and application of an evolutionary many-objective optimization algorithm based on the framework of NSGA-II, we believe that we have addressed a long-awaited issue in the area of evolutionary multi-objective optimization (EMO). Testing on a number of unconstrained problems having box constraints alone in [1] and on generic constrained problems in this study amply suggest that evolutionary methods can be useful as well to solve many-objective optimization problems (shown up to 15 objectives in both studies). Although further improvements are possible, the proposed adaptive NSGA-III approach should also remain as a useful algorithm for adaptively relocating reference points in the relevant part of the Pareto-optimal front. Importantly, these two extensive studies have paved directions for future research and should motivate EMO researchers to develop further and better algorithms for many-objective optimization in the near future.

ACKNOWLEDGMENT

REFERENCES

[1] K. Deb and H. Jain, "An improved NSGA-II procedure for many-objective optimization Part I: Problems with box constraints," Indian Institute of Technology Kanpur, Tech. Rep. 2012009, 2012.

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

[3] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.

[4] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*. London, UK: Springer-Verlag, 2005.

[5] 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.

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

[7] 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.

[8] 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.

[9] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, p. in press, 2011.

[10] J. A. López and C. A. C. 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: ACM, 2009, pp. 2693–2696.

[11] 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.

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

[13] 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.

[14] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Athens, Greece: International Center for Numerical Methods in Engineering (CIMNE), 2001, pp. 95–100.

[15] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *Proceedings of 2010 UK Workshop on Computational Intelligence*, 2010, pp. 1–6.

[16] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans on Evolutionary Computation*, vol. 12, no. 2, pp. 284–302, 2009.

[17] R. Storn and K. Price, "Differential evolution – A fast and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[18] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms–Part II: Application example," *IEEE Transactions on Systems, Man, and Cybernetics: Part A: Systems and Humans*, vol. 28, no. 1, pp. 38–47, 1998.

[19] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, 2001, pp. 284–298.

[20] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *Proceedings of the Congress on Evolutionary Computation (CEC-06)*, 2006, pp. 1179 – 1186.

[21] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. New York: North-Holland, 1983.

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

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

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

[25] T. Ray, K. Tai, and K. C. Seow, "An evolutionary algorithm for multiobjective optimization," *Engineering Optimization*, vol. 33, no. 3, pp. 399–424, 2001.

[26] 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

20

APPENDIX

CAR-SIDE IMPACT PROBLEM FORMULATION

Mathematical formulation of the three-objective problem is given below. All objectives are to be minimized.

$f_1(\mathbf{x}) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5$
$\qquad + 0.00001x_6 + 2.73x_7,$

$f_2(\mathbf{x}) = F$

$f_3(\mathbf{x}) = 0.5(V_{MBP} + V_{FD})$

$g_1(\mathbf{x}) = 1.16 - 0.3717x_2x_4 - 0.0092928x_3 \leq 1$

$g_2(\mathbf{x}) = 0.261 - 0.0159x_1x_2 - 0.06486x_1 - 0.019x_2x_7 + 0.0144x_3x_5$
$\qquad + 0.0154464x_6 \leq 0.32$

$g_3(\mathbf{x}) = 0.214 + 0.00817x_5 - 0.045195x_1 - 0.0135168x_1$
$\qquad + 0.03099x_2x_6 - 0.018x_2x_7 + 0.007176x_3$
$\qquad + 0.023232x_3 - 0.00364x_5x_6 - 0.018x_2^2 \leq 0.32$

$g_4(\mathbf{x}) = 0.74 - 0.61x_2 - 0.031296x_3 - 0.031872x_7 + 0.227x_2^2 \leq 0.32$

$g_5(\mathbf{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 1.27296x_6 - 2.68065x_7 \leq 32$

$g_6(\mathbf{x}) = 33.86 + 2.95x_3 - 5.057x_1x_2 - 3.795x_2 - 3.4431x_7$
$\qquad + 1.45728 \leq 32$

$g_7(\mathbf{x}) = 46.36 - 9.9x_2 - 4.4505x_1 \leq 32$

$g_8(\mathbf{x}) \equiv F = 4.72 - 0.5x_4 - 0.19x_2x_3 \leq 4$

$g_9(\mathbf{x}) \equiv V_{MBP} = 10.58 - 0.674x_1x_2 - 0.67275x_2 \leq 9.9$

$g_{10}(\mathbf{x}) \equiv V_{FD} = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 \leq 15.7$

Variable bounds are given given as follows:

$0.5 \leq x_1 \leq 1.5, \quad 0.45 \leq x_2 \leq 1.35, \quad 0.5 \leq x_3 \leq 1.5,$
$0.5 \leq x_4 \leq 1.5, \quad 0.875 \leq x_5 \leq 2.625, \quad 0.4 \leq x_6 \leq 1.2,$
$0.4 \leq x_7 \leq 1.2$

WATER PROBLEM FORMULATION

Mathematical formulation of the five-objective problem is given below. All objectives are to be minimized.

$f_1(\mathbf{x}) = 106780.37(x_2 + x_3) + 61704.67$

$f_2(\mathbf{x}) = 3000.0x_1$

$f_3(\mathbf{x}) = 30570 * 0.02289.0x_2/(0.06 * 2289.0)^{0.65}$

$f_4(\mathbf{x}) = 250.0 * 2289.0exp(-39.75x_2 + 9.9x_3 + 2.74)$

$f_5(\mathbf{x}) = 25.0((1.39/(x_1x_2)) + 4940.0x_3 - 80.0)$

$g_1(\mathbf{x}) = 0.00139/(x_1x_2) + 4.94x_3 - 0.08 \leq 1$

$g_2(\mathbf{x}) = 0.000306/(x_1x_2) + 1.082x_3 - 0.0986 \leq 1$

$g_3(\mathbf{x}) = 12.307/(x_1x_2) + 49408.24x_3 + 4051.02 \leq 50000$

$g_4(\mathbf{x}) = 2.098/(x_1x_2) + 8046.33x_3 - 696.71 \leq 16000$

$g_5(\mathbf{x}) = 2.138/(x_1x_2) + 7883.39x_3 - 705.04 \leq 10000$

$g_6(\mathbf{x}) = 0.417(x_1x_2) + 1721.26x_3 - 136.54 \leq 2000$

$g_7(\mathbf{x}) = 0.164/(x_1x_2) + 631.13x_3 - 54.48 \leq 550$

$0.01 \leq x_1 \leq 0.45$

$0.01 \leq x_2, x_3 \leq 0.10.$

Himanshu Jain Himanshu Jain was born in Nehtaur, India, on December 6, 1989. He received his dual Bachelor's and Master's degree in Mechanical Engineering from Indian Institute of Technology Kanpur in 2012. He was a member of the Kanpur Genetic Algorithms Laboratory (KanGAL) from 2009. He is currently working as an Engineer at Eaton Corporation, Pune. His research interests include evolutionary computation and machine learning.

Kalyanmoy Deb Kalyanmoy Deb is the Koenig Endowed Chair Professor at the Department of Electrical and Computer Engineering at Michigan State University (MSU), East Lansing, USA. He is also a professor of Computer Science and Engineering, and Mechanical Engineering at MSU. Prior to this position, he was at Indian Institute of Technology Kanpur in India. Prof. Deb also has a visiting position at Aalto University School of Economics, Finland and University of Skövde, Sweden. Prof. Deb's main research interests are in evolutionary optimization algorithms and their application in optimization and machine learning. He is largely known for his seminal research in Evolutionary Multi-Criterion Optimization. He was awarded the prestigious 'Infosys Prize' in 2012, 'TWAS Prize' in Engineering Sciences in 2012, 'CajAstur Mamdani Prize' in 2011, 'JC Bose National Fellowship' in 2011, 'Distinguished Alumni Award' from IIT Kharagpur in 2011, 'Edgeworth-Pareto' award in 2008, Shanti Swarup Bhatnagar Prize in Engineering Sciences in 2005, 'Thomson Citation Laureate Award' from Thompson Reuters. His 2002 IEEE-TEC NSGA-II paper is now judged as the Most Highly Cited paper and a Current Classic by Thomson Reuters having more than 4,000+ citations. He is a fellow of IEEE. He has written two text books on optimization and more than 340 international journal and conference research papers. He is in the editorial board on 18 major international journals, including IEEE TEC.