An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic algorithm for Many-Objective Optimization (A^2-NSGA-III)

Himanshu Jain and Kalyanmoy Deb

Department of Mechanical Engineering
Indian Institute of Technology, Kanpur, India
{hjain,deb}@iitk.ac.in
http://www.iitk.ac.in/kangal/

COIN Report Number 2013014

Abstract. NSGA-II and its contemporary EMO algorithms were found to be vulnerable in solving many-objective optimization problems having four or more objectives. It is not surprising that EMO researchers have been concentrating in developing efficient algorithms for many-objective optimization problems. Recently, authors suggested an extension of NSGA-II (NSGA-III) which is based on the supply of a set of reference points and demonstrated its working on three to 15-objective optimization problems. In this paper, NSGA-III's reference point allocation task is made adaptive so that a better distribution of points can be found. The approach is compared with NSGA-III and a previous adaptive approach on a number of constrained and unconstrained many-objective optimization problems. NSGA-III and its adaptive extension proposed here open up new directions for research and development in the area of solving many-objective optimization problems

Keywords: Many-objective optimization, NSGA-III, adaptive optimization, evolutionary optimization.

1 Introduction

Over the years, NSGA-II [2] has been applied to various practical problems and was adopted in various commercial softwares. However, NSGA-II, like other evolutionary multi-objective optimization (EMO) algorithms, suffers from its ability to handle more then three objectives adequately. When the so-called 'curse of dimensionality' thwarted the progress of algorithm development in the EMO field, researchers took interests in devising new methodologies for solving many-objective optimization problems, involving four or more objectives [8, 9, 11, 12]. Progressing towards the Pareto-optimal front and simultaneously arriving at a well-distributed set of trade-off solutions in a high-dimensional space were found to be too challenging tasks for any algorithm to be computationally tractable. Earlier in 2012, authors of this paper suggested a new extension of NSGA-II

(MO-NSGA-II) specifically for solving many-objective optimization problems. MO-NSGA-II [5] starts with a set of automatically or user-defined reference points and then focuses its search to emphasize the EMO population members that are non-dominated in the population and are also "closest" to each of the reference points, thereby finding a well-distributed and well-converged set of solutions. In later studies [3, 4], MO-NSGA-II was further modified and extended to solve constrained problems, this new algorithm was named as NSGA-III. The latter study also suggested an adaptive approach (A-NSGA-III) that was capable of identifying reference points that do not correspond to a well-distributed set of Pareto-optimal points.

In this paper, we extend the concept of relocation of reference points and attempt to remove some of the shortcomings of A-NSGA-III algorithm and suggest an efficient adaptive NSGA-III approach (A^2-NSGA-III) for this purpose. In the remainder of this paper, we first provide a brief overview of NSGA-III and A-NSGA-III approaches. Thereafter, we motivate the reasons for improving the adaptive approach and suggest our proposed procedure (A^2-NSGA-III). Simulation results are shown on constrained and unconstrained problems using the proposed procedure and are compared with original NSGA-III and the A-NSGA-III approaches. Conclusions of this study are then made.

2 Many Objective NSGA-II or NSGA-III

The basic framework of the NSGA-III [3] is similar to the original NSGA-II algorithm [2]. First, the parent population P_t (of size N) is randomly initialized in the specified domain, then the binary tournament selection, crossover and mutation operators are applied to create an offspring population Q_t. Thereafter, both populations are combined and sorted according to their domination level and the best N members are selected from the combined population to be the parent population for the next generation. The fundamental difference between NSGA-II and NSGA-III lies in the way the niche-preservation operation is performed.

Unlike NSGA-II, NSGA-III starts with a set of reference points Z^r. After non-dominated sorting, all acceptable front members and the last front F_l which could not

Algorithm 1 Generation t of NSGA-III procedure

Input: H reference points Z^r, parent population P_t
Output: P_{t+1}

$S_t = \emptyset$, $i = 1$
Q_t = Recombination+Mutation(P_t)
$R_t = P_t \cup Q_t$
$(F_1, F_2, \ldots)$ = Non-dominated-sort(R_t)
repeat
  $S_t = S_t \cup F_i$ and $i = i + 1$
until $|S_t| \geq N$
Last front to be included: $F_l = F_i$
if $|S_t| = N$ **then**
  $P_{t+1} = S_t$, break
else
  $P_{t+1} = \cup_{j=1}^{l-1} F_j$
  Points to be chosen from F_l: $K = N - |P_{t+1}|$
  Normalize objectives
  Associate each member $\mathbf{s}$ of S_t with a reference point: $[\pi(\mathbf{s}), d(\mathbf{s})]$ =`Associate`(S_t, Z^r) % $\pi(\mathbf{s})$: closest reference point, d: distance between $\mathbf{s}$ and $\pi(\mathbf{s})$
  Compute niche count of reference point $j \in Z^r$: $\rho_j = \sum_{\mathbf{s} \in S_t/F_l} ((\pi(\mathbf{s}) = j)\ ?\ 1\ :\ 0)$
  Choose K members one at a time from F_l to construct P_{t+1}: `Niching`$(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
end if

be completely accepted are saved in a set S_t. Members in S_t/F_l are selected right away for the next generation, however the remaining members are selected from F_l such that a desired diversity is maintained in the population. Original NSGA-II used the crowding distance measure for selecting well-distributed set of points, however in NSGA-III the supplied reference points (Z^r) are used to select these remaining members. To accomplish this, objective values and reference points are first normalized so that they have an identical range. Thereafter, orthogonal distance between a member in S_t and each of the reference lines (joining the ideal point and a reference point) is calculated. The member is then associated with the reference point having the smallest orthogonal distance. Next, the niche count ρ for each reference point, defined as the number of members in S_t/F_l that are associated with the reference point, is computed for further processing. The reference point having the minimum niche count is identified and the member in front last front F_l that is associated with the identified reference point is included in the final population. The niche count of the identified reference point is increased by one and the procedure is repeated to fill up population P_{t+1}. The entire procedure is presented in algorithmic form in 1. In NSGA-III, a different tournament selection operator is used. If both competing parents are feasible, then one of them is chosen at random. However, if one is feasible and the other is infeasible, then the feasible one is selected. Finally, if both are infeasible, then the one having the least constraint violation is selected. After applying this tournament selection operator, usual crossover and mutation operations are carried out to create the offspring population and the above-mentioned niching operation is applied again to the combined population. These steps are continued until a termination criterion is satisfied.

Some interesting features of NSGA-III are as follows: (i) it does not require any additional parameter setting, just like its predecessor NSGA-II, (ii) the population size is almost same as the number of reference points, thereby making an efficient computational effort, (iii) it can be used to find trade-off points in the entire Pareto-optimal front or focused in a preferred Pareto-optimal region, (iv) it is extended easily to solve constrained optimization problems, (v) it can be used with a small population size (such as a population of size 100 for a 10-objective optimization problem) and (vi) it can be used for other multi-objective problem solving tasks, such as in finding the nadir point or other special points.

3 Adaptive NSGA-III

A little thought will reveal the fact that not all reference points may be associated with a well-dispersed Pareto-optimal set and carrying on with a predefined set of reference points from start to finish may be a waste of computational efforts. To clarify, let us consider a three-objective optimization problem where the Pareto-optimal front is shown as the shaded portion in Figure 1 and 91 initial reference points are marked as open circles. Clearly, only 28 reference points have a corresponding Pareto-optimal point, while the rest 63 points are then found to be

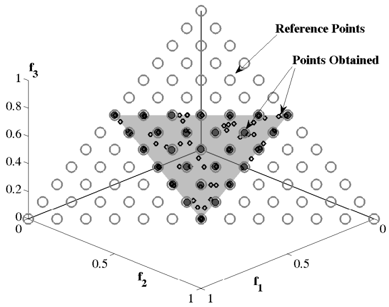

Fig. 1. Only 28 out of 91 reference points find a Pareto-optimal solution.

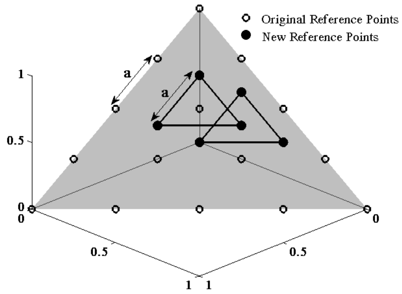

Fig. 2. Addition of reference points.

randomly distributed, as shown in small circles in Figure 1. One possible remedy to this problem is to first identify all those reference points that are not associated with a population member. Then, instead of eliminating these reference points, they can then be relocated so as to find a better distribution of Pareto-optimal points. The following addition and deletion strategies were proposed in A-NSGA-III.

Algorithm 2 `Add Reference Points`(Z^r, ρ, p) procedure

Input: Z^r, p, ρ, $Flag(Z^r)$
Output: $Z^r(updated)$
nref = $|Z^r|$
for $j = 1$ **to** nref **do**
  if $\rho_j > 1$ and $Flag(Z^r_j) = 0$ **then**
    $Z^r_{new} = Structured - points(p = 1)$
    $Z^r_{new} = Z^r_{new}/p + (Z^r_j - 1/(M * P))$
    for $i = 1$ **to** M **do**
      if already-exist($Z^r_{new,i}$) = FALSE and $Z^r_{new,i}$ lie in first quadrant **then**
        $Z^r = Z^r \cup Z^r_{new,i}$
        $Flag(Z^r_{new,i}) = 0$
      end if
    end for
    $Flag(Z^r_j) = 1$
  end if
end for

Note that, in NSGA-III, 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. As the number of reference points (H) is kept almost equal to population size (N), every reference point is expected to be associated with one population member. Thus, if $\rho_j \geq 2$ is observed for any reference point, this means that some other reference point has a zero niche count value. Hence a reference point having zero ρ value is relocated close to the j-th reference point. The relocation procedure is shown in Figure 2. Consider the situation in $M = 3$ objective case. For adding extra reference points, a $(M-1)$-dimensional simplex having M points at its vertices is added. The side length of the simplex is equal to the distance between two consecutive reference points (which is controlled by parameter p) on the original specified hyperplane and the centroid of the simplex is kept on the j-th reference point. 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 added reference point is accepted, two checks are made: (i) if it lies outside the boundary of original simplex, it is not accepted, and (ii) if it already exists in the set of reference points, it is also not accepted. This addition procedure is presented in algorithmic form in 2. Using this procedure it may happen that after some generations too many reference points are added and many of them eventually become *non-useful* again. To avoid this, deletion of non-useful reference points is also carried out simultaneously, as described in the following paragraph:
After the inclusion operation is performed, the niche count of all reference points are updated. Now, if there exist no reference point whose niche count is greater than one (that is, $\rho_j = 0$ or 1 for all reference points), this means that each and every population member is associated with a single unique reference point. In this case, the reference points ($\rho_j = 0$) that are not associated with any population members are simply deleted. In this way the inclusion and deletion operations adaptively relocate reference points based on the niche count values of the respective reference points. The A-NSGA-III worked well on a number of problems in the previous study [4], however, the concept deserves more attention.

4 Limitations of A-NSGA-III

Following limitations of adaptive strategy discussed above are observed:

1. In problems where the entire Pareto-optimal front is concentrated in a small region or in case we start with only few reference points and a sufficiently large population size, the above addition procedure may not be able to introduce enough reference points so that the entire population may be evenly distributed.
2. The above followed addition procedure does not allow introduction of extra reference points around the corner reference points of the hyperplane.
3. Since the addition procedure is carried out right from first generation when the population is far from the actual Pareto-front we may not have allowed enough time for the algorithm to spread the population evenly in various regions which may lead to premature introduction of extra reference points in unwanted regions.
4. Since the removal procedure is only carried out when the ρ value for all the reference points is less than or equal to one, in some cases (specially large-dimensional problems) it may happen that this condition is never satisfied and the algorithm keeps on adding extra reference points, thus increasing the computational cost.

In order to overcome these limitations, we modify the above approach for addition and deletion of reference points in the following section.

5 Efficiently Adaptive NSGA-III Procedure (A^2-NSGA-III)

Let us suppose that the extra reference points are to be added around the j-th reference point (marked as P in Figure 3). As done earlier, here also we use an $(M-1)$-dimensional simplex (shown as ABC) but having a side length equal to the half of the distance between two consecutive reference points on the original normalized hyperplane. This simplex is called the primary simplex that will be added to the reference point P. However, instead of adding the simplex around the reference point as it was done in A-NSGA-III, it is now added by keeping the j-th reference point as one of the corners of the simplex as shown in Figure 3, thus adding $(M-1)$ new reference points. Since there are M points in the simplex, there are a total of M such ways of adding the simplex. To implement, we randomly select one of the corner points and overlay the simplex with the selected corner point falling on the reference point. For example in Figure 3, if we choose the corner A of simplex and coincide it with the reference point P then we get configuration 1 shown in the figure. On the other hand, if we select corner B then we get configuration 2, and so on. Like before, before accepting a configuration, all new locations of reference points are checked for the following

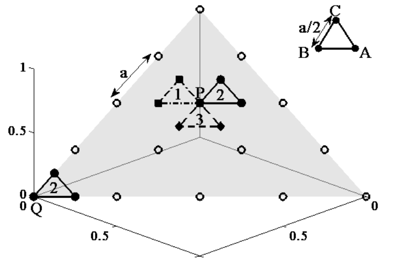

Fig. 3. Proposed approach for adding extra reference points.

Algorithm 3 `Add Reference Points(`Z^r, ρ, p, λ`)` procedure for A^2-NSGA-III

Input: Z^r, p, ρ, $Config(Z^r)$, λ { *%$Config(Z^r)$ contains the configuration number of simplex to be added to each reference point, if $Config(Z^r_k) > M$ it means all configurations are added around k^{th} ref point.*}
Output: Z^r *(updated)*

nref = $|Z^r|$
for $j = 1$ **to** nref **do**
  Flag = 0
  if $\rho_j > 1$ **then**
    while $Config(Z^r_j) \leq M$ and $Flag == 0$ **do**
      $Z^r_{new} = Structured - points(p = 1)$
      $Z^r_{new} = Z^r_{new}/(\lambda * p) + (Z^r_j - Z^r_{new,Config(Z^r_j)}/(\lambda * p))$
      for $i = 1$ **to** M **do**
        if $i \neq Config(Z^r_j)$ and already-exist($Z^r_{new,i}$) = FALSE and $Z^r_{new,i}$ lie in first quadrant **then**
          $Z^r = Z^r \cup Z^r_{new,i}$
          $Config(Z^r_{new,i}) = 1$
          $Flag = 1$
        end if
      end for
      $Config(Z^r_j) = Config(Z^r_j) + 1$
    end while
  end if
end for
if ∃ j(=1:nref) s.t $\rho_j > 1$ and ∀ j s.t $\rho_j > 1$ $Config(Z^r_j) > M$ and no new reference point is added **then**
  $\lambda = \lambda * 2$
end if

two conditions:(i) if any newly located reference point lies outside the original hyperplane, the configuration is not accepted (for example in Figure 3 if extra points are to be added around reference point Q then configurations 1 and 3 are not acceptable), and (ii) if a newly located reference point already exists in the set of reference points, that point is not duplicated. The procedure is continued with all reference points having $\rho_j > 1$. After one simplex is added to such a reference point, if ρ_j is still greater than one, other allowable configurations are continued to be added until all M configurations are added. Thereafter, for further $\rho_j > 1$ occurrences, simplexes half of its current size are introduced one at a time, this is controlled by a scaling factor λ whose initial value is kept to be 2 which denotes simplexes having a side length equal to the half of the distance between two consecutive reference points on the original normalized hyperplane are to be added. Whenever the above stated condition arrives λ value is increased by a factor of 2. This process is repeated till none of the reference point has a niche count greater than one. This improved addition procedure is presented in algorithmic form in 3. The ability to introduce more and more simplexes but with reduced size alleviates the first limitation described above with the previous approach. The above procedure of using simplexes half the size of original simplexes enable more concentrated reference points to be introduced near the vertices of the original hyperplane, thereby alleviating the second limitation mentioned above. To cater the third limitation, we introduce a condition before new reference points can be added. The number of reference points having $\rho > 1$ is monitored and only when the number has settled down to a constant value in the past τ generations, addition of reference points is allowed. This check will ensure that enough time has been spent by the algorithm to evenly spread its population members with the supplied set of reference points before any new reference points are introduced. In this study, we have used $\tau = 10$ generations. Now the last limitation is alleviated by putting a cap over the maximum number of reference points that can ever be handled by the algorithm. Thus, if the total number of reference points shoots up beyond this value the deletion process (described in subsection 3) is carried out, thereby lowering the burden of carrying forward with a large number of reference points. Here, we have used 10 times the number of originally supplied reference points as the cap for maximum number of reference points.

6 Results

We now present the simulation results of both adaptive NSGA-III approaches on a number of three to eight-objective test problems. The population sizes and number of reference points are kept as mentioned in Table 1. Other parameters are kept identical for both approaches: (i) SBX crossover probability of one, (ii) polynomial mutation probability of $1/n$ (where n is the number of variables), (iii) SBX crossover and polynomial mutation indices are kept as 30 and 20,

Table 1. Number of ref. points and population sizes used.

No. of obj. (M)	Ref. pts. (H)	Pop. size (N)
3	91	92
5	210	212
8	156	156

respectively. As a performance metric, we have used the hypervolume indicator as it captures both convergence and distribution ability of an algorithm. In each case, 20 runs are carried out and the best, median and worst hypervolume values are reported.

6.1 Unconstrained Test Problems

Inverted DTLZ1 and DTLZ2 Problem: First of all we take two problems: DTLZ1 and DTLZ2 from scalable DTLZ suite[6] and modify them so that there are certain reference points on normalized hyperplane corresponding to whom there is no point on Pareto-optimal front. To accomplish this in both problems the objective functions are calculated using the original formulation, however after calculating the objective function values, following transformations are made: For DTLZ1:

$$f_i(\mathbf{x}) \leftarrow 0.5(1+g(\mathbf{x})) - f_i(\mathbf{x}), \quad \text{for } i = 1, \ldots, M.$$

For DTLZ2:

$$f_i(\mathbf{x}) \leftarrow (1+g(\mathbf{x})) - f_i(\mathbf{x})^4, \ \text{ for } i = 1, \ldots, (M-1) \text{ and } f_M(\mathbf{x}) \leftarrow (1+g(\mathbf{x})) - f_M(\mathbf{x})^2.$$

where $g(\mathbf{x})$ is calculated as in the original DTLZ1 and DTLZ2 formulation respectively [6]. This transformation inverts the original Pareto-optimal front thereby rendering several reference points as non-useful.

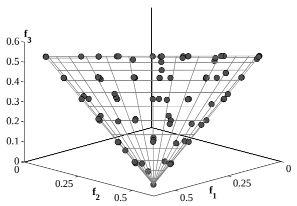

Fig. 4. NSGA-III solutions for Inv-DTLZ1.

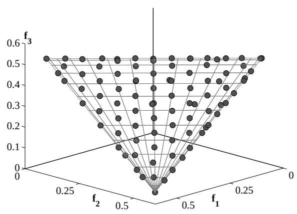

Fig. 5. A-NSGA-III solutions for Inv-DTLZ1.

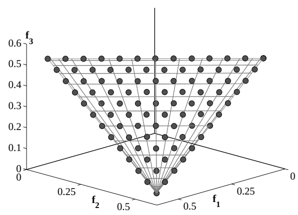

Fig. 6. A^2-NSGA-III solutions for Inv-DTLZ1.

All the approaches are tested against three, five, and eight objective versions of both problems. Pareto-optimal fronts obtained in the case of three-objective version are plotted in Figures 4, 5, 6, 7, 8, and 9 (plotted fronts correspond to median hypervolume values as tabulated in Table 2). As evident from Figure 4 for DTLZ1, in the case of A^2-NSGA-III, all 91 points are uniformly distributed, while with NSGA-III only 28 population members are distributed and with A-NSGA-III, the number increases to 81 and rest are randomly dispersed over the entire front. Similar observation is made in case of inverted DTLZ2 problem. Hypervolume values for three to eight-objective cases are tabulated in Table 2

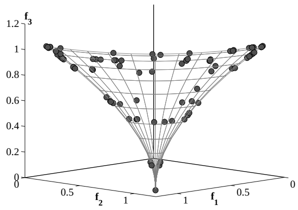

Fig. 7. NSGA-III solutions for Inv-DTLZ2.

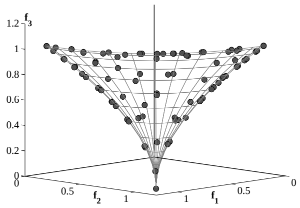

Fig. 8. A-NSGA-III solutions for Inv-DTLZ2.

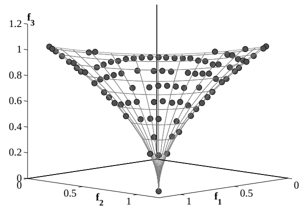

Fig. 9. A^2-NSGA-III solutions for Inv-DTLZ2.

Table 2. Best, median and worst hypervolume values on M-objective inverted DTLZ1 and DTLZ2 problems. Best values are marked in bold.

Function	M	Gen	NSGA-III	A-NSGA-III	A^2-NSGA-III
Inv-DTLZ1			$6.1115e-02$	$6.0945e-02$	$\mathbf{6.3738e-02}$
	3	400	$6.2294e-02$	$6.5404e-02$	$\mathbf{6.5693e-02}$
			$6.3052e-02$	$6.5967e-02$	$\mathbf{6.6206e-02}$
			$1.5475e-03$	$1.9610e-03$	$\mathbf{2.6257e-03}$
	5	600	$1.9821e-03$	$2.1255e-03$	$\mathbf{2.7117e-03}$
			$2.1409e-03$	$2.2413e-03$	$\mathbf{2.8273e-03}$
			$3.0186e-06$	$3.0186e-06$	$\mathbf{4.2066e-06}$
	8	750	$3.3864e-06$	$3.4220e-06$	$\mathbf{4.8353e-06}$
			$3.8361e-06$	$3.8361e-06$	$\mathbf{5.5693e-06}$
Inv-DTLZ2			$1.0530e-01$	$1.1593e-01$	$\mathbf{1.1790e-01}$
	3	250	$1.0951e-01$	$1.1951e-01$	$\mathbf{1.2342e-01}$
			$1.1197e-01$	$1.2137e-01$	$\mathbf{1.2550e-01}$
			$1.5083e-03$	$1.5365e-03$	$\mathbf{2.2707e-03}$
	5	350	$1.9100e-03$	$2.0241e-03$	$\mathbf{3.0822e-03}$
			$2.0894e-03$	$2.2904e-03$	$\mathbf{3.1760e-03}$
			$1.2168e-06$	$1.2168e-06$	$\mathbf{1.2814e-06}$
	8	500	$1.4818e-06$	$1.4818e-06$	$\mathbf{2.3228e-06}$
			$1.7822e-06$	$1.7822e-06$	$\mathbf{3.7236e-06}$

for both the problems. It is clear that in all cases, the use of proposed adaptive method improves the performance of algorithm as compared to the basic and previously suggested adaptive algorithm. The performance of the proposed method gets better with an increase in the number of objectives.

6.2 Constrained Test Problems

After demonstrating the efficacy of proposed approach on a couple of unconstrained test problems, next we consider two constrained problems. These problems are designed by adding constraints to the original scalable DTLZ1 and DTLZ2 problems so that some portions of the original Pareto-optimal front become infeasible.

C-DTLZ1 and C-DTLZ2 Problem: In case of C-DTLZ1 problem, we add a hyper-cylinder (with its central axis passing through the origin and equally inclined to all the objective axes) as a constraint so that the region inside the hyper-cylinder is feasible. In case of C-DTLZ2 we add a constraint to the original DTLZ2 problem, thereby making the entire region of objective space lying in between $0.1 < f_M < 0.9$ infeasible. Due to these changes in both problems, not all reference points initialized on the normalized hyperplane will have an associated feasible Pareto-optimal point. In such problems, the original NSGA-III may waste its computations in dealing with such non-productive reference points and the previously suggested adaptive A-NSGA-III may not be able to fully relocate all reference points to find a well-distributed set of Pareto-optimal points. Figures 10, 11, and 12 show the obtained fronts using the three approaches, respectively, for problem C-DTLZ1. As one can see here the distribution of points obtained using the proposed A^2-NSGA-III is better than that obtained using NSGA-III but relocations of reference points in both A-NSGA-III and A^2-NSGA-III are comparable. The current approach allows a greater density in solutions, but this may not happen uniformly across the entire front, as evident from the Figure 12 which may lead to very less improvement in hypervolume value. Table 3 shows that for three-objective version, the hypervolume is slightly better for A-NSGA-III, while for larger objective cases, A^2-NSGA-III has better hypervolume values.

For C-DTLZ2, as shown in Figures 13, 14, and 15, we get a better distribution of points using A^2-NSGA-III and the same is reflected in the hypervolume values, tabulated in Table 3 for three-objective case. In five-objective version, A^2-NSGA-III performs the best, but in eight-objective version of the problem, A-NSGA-III performs slightly better for the allocated number of population members.

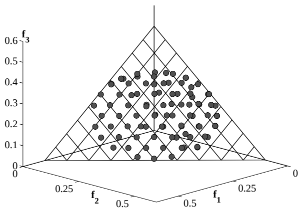

Fig. 10. NSGA-III solutions for DTLZ1.

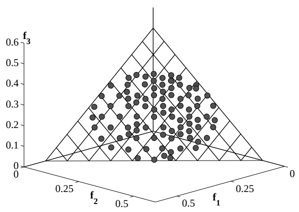

Fig. 11. A-NSGA-III solutions for DTLZ1.

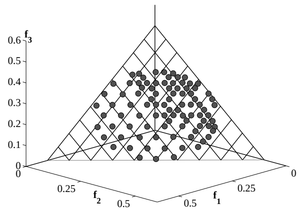

Fig. 12. A^2-NSGA-III solutions for DTLZ1.

6.3 A^2-NSGA-III with Large Population Size

NSGA-III and its adaptive version A-NSGA-III were developed with the principle that a population size almost equal to the number of reference points is to be used. This enabled a direct control of the maximum number of obtained trade-off

Table 3. Best, median and worst hypervolume values on M-objective constrained DTLZ1 and DTLZ2 problems.

Function	M	Gen	NSGA-III	A-NSGA-III	A^2-NSGA-III
C-DTLZ1-hole	3	400	$1.7099e-01$	$\mathbf{1.7135e-01}$	$1.7096e-01$
			$1.7357e-01$	$\mathbf{1.7458e-01}$	$1.7418e-01$
			$1.7501e-01$	$\mathbf{1.7576e-01}$	$1.7535e-01$
	5	600	$7.4929e-02$	$\mathbf{7.4934e-02}$	$7.4846e-02$
			$\mathbf{7.5151e-02}$	$7.5128e-02$	$7.5141e-02$
			$7.5224e-02$	$7.5246e-02$	$\mathbf{7.5310e-02}$
	8	750	$1.5683e-02$	$1.5724e-02$	$\mathbf{1.5878e-02}$
			$1.5916e-02$	$1.5842e-02$	$\mathbf{1.6118e-02}$
			$1.6120e-02$	$1.6022e-02$	$\mathbf{1.6306e-02}$
C-DTLZ2	3	250	$6.2968e-01$	$6.2998e-01$	$\mathbf{6.3056e-01}$
			$6.3091e-01$	$6.3091e-01$	$\mathbf{6.3267e-01}$
			$6.3230e-01$	$6.3187e-01$	$\mathbf{6.3483e-01}$
	5	350	$1.2340e+00$	$\mathbf{1.2350e+00}$	$1.2348e+00$
			$1.2375e+00$	$1.2377e+00$	$\mathbf{1.2400e+00}$
			$1.2398e+00$	$1.2405e+00$	$\mathbf{1.2422e+00}$
	8	500	$1.8209e+00$	$1.9114e+00$	$\mathbf{1.9151e+00}$
			$1.8409e+00$	$\mathbf{1.9245e+00}$	$1.9211e+00$
			$1.8576e+00$	$\mathbf{1.9286e+00}$	$1.9273e+00$

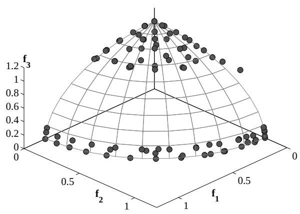

Fig. 13. NSGA-III solutions for DTLZ2.

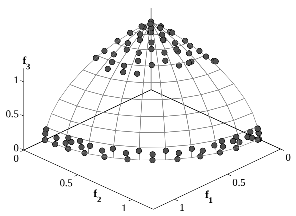

Fig. 14. A-NSGA-III solutions for DTLZ2.

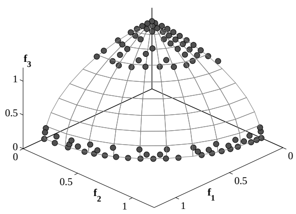

Fig. 15. A^2-NSGA-III solutions for DTLZ2.

points that is expected from the algorithms. If more points are needed, previous algorithms allowed a larger population to be used, but the obtained points may not be distributed well. However, with our proposed A^2-NSGA-III modification, reference points now can be considered as seed points and a much larger set of trade-off points can be obtained by simply using a larger population size. To illustrate, we consider two scalable test problems DTLZ1 and DTLZ2 and use population sizes larger than number of reference points as tabulated in Table 4.

Table 4. Number of ref. points and population sizes used.

No. of obj. (M)	Ref. pts. (H)	Pop. size (N)
3	28	92
5	35	212
8	44	156

We have used three, five, and eight objective versions of both problems. Figures 16, 17, and 18 show the obtained fronts for three-objective DTLZ1 problem using all three ap-

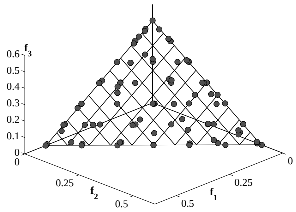

Fig. 16. NSGA-III solutions for DTLZ1.

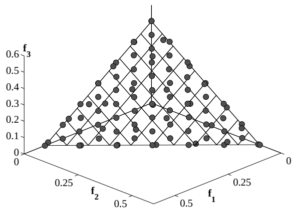

Fig. 17. A-NSGA-III solutions for DTLZ1.

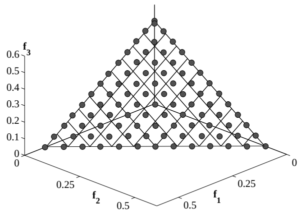

Fig. 18. A^2-NSGA-III solutions for DTLZ1.

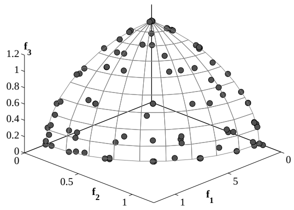

Fig. 19. NSGA-III solutions for DTLZ2.

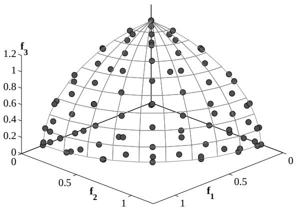

Fig. 20. A-NSGA-III solutions for DTLZ2.

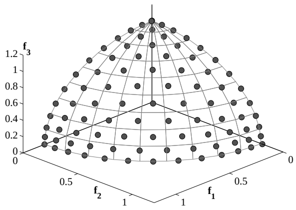

Fig. 21. A^2-NSGA-III solutions for DTLZ2.

proaches. Clearly, with A^2-NSGA-III we get an excellent distribution having all 92 population members, despite the initialization of only 28 reference points on the normalized hyperplane. With A-NSGA-III, the distribution is better than that obtained using NSGA-III, but due to the limitation of only one configuration per reference point, the approach cannot utilize the available pool of population members adequately.

Similarly, for the three-objective DTLZ2, we get the best distribution using A^2-NSGA-III as shown in Figure 21 while with rest of the approaches (Figures 19 and 20) not all the points are well distributed. Table 5 shows the best, median, and worst hypervolume values for three, five and eight objective versions of both problems obtained using all three approaches. A^2-NSGA-III performs the best in all cases in both problems with an identical number of function evaluations, thereby showing the superiority of the proposed adaptive procedure.

6.4 Engineering Optimization Problems

Next, we apply the proposed algorithm on four engineering problems ranging from three to five objectives.

Crash-worthiness Problem: This is a three-objective unconstrained problem aimed at structural optimization of the frontal structure of vehicle for crash-worthiness [10]. Thickness of five reinforced members around the frontal structure are chosen as design variables, while the mass of vehicle, deceleration during the full frontal crash (which is proportional to biomechanical injuries caused to

Table 5. Best, median and worst hypervolume for DTLZ1 and DTLZ2 problems.

Problem	M	Gen	NSGA-III	A-NSGA-III	A^2-NSGA-III
DTLZ1			$1.8569e-01$	$\mathbf{1.8810e-01}$	$1.8671e-01$
	3	400	$1.8674e-01$	$1.8871e-01$	$\mathbf{1.8928e-01}$
			$1.8731e-01$	$1.8912e-01$	$\mathbf{1.8962e-01}$
			$\mathbf{7.6252e-02}$	$7.6252e-02$	$7.6235e-02$
	5	600	$7.6361e-02$	$7.6361e-02$	$\mathbf{7.6726e-02}$
			$7.6449e-02$	$7.6449e-02$	$\mathbf{7.6742e-02}$
			$8.9663e-03$	$8.9663e-03$	$\mathbf{1.6761e-02}$
	8	750	$1.6753e-02$	$1.6753e-02$	$\mathbf{1.6766e-02}$
			$1.6760e-02$	$1.6760e-02$	$\mathbf{1.6770e-02}$
DTLZ2			$7.0956e-01$	$7.2851e-01$	$\mathbf{7.4288e-01}$
	3	250	$7.1800e-01$	$7.3040e-01$	$\mathbf{7.4379e-01}$
			$7.2399e-01$	$7.3320e-01$	$\mathbf{7.4414e-01}$
			$1.2583e+00$	$1.2583e+00$	$\mathbf{1.3017e+00}$
	5	350	$1.2660e+00$	$1.2660e+00$	$\mathbf{1.3039e+00}$
			$1.2780e+00$	$1.2780e+00$	$\mathbf{1.3049e+00}$
			$3.8935e-03$	$3.8935e-03$	$\mathbf{6.4729e-01}$
	8	500	$1.6799e-02$	$1.6799e-02$	$\mathbf{2.1428e+00}$
			$2.0983e+00$	$2.0983e+00$	$\mathbf{2.1435e+00}$

the occupants) and the toe board intrusion in the offset-frontal crash (which accounts for the structural integrity of the vehicle) are taken as objectives. Mathematical formulation for the problem can be found elsewhere [3]. We solve this

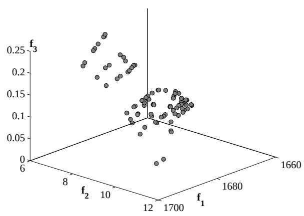

Fig. 22. NSGA-III solutions for Crash-worthiness problem.

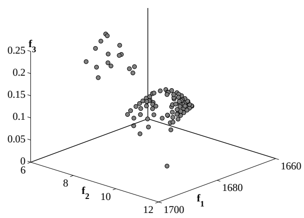

Fig. 23. A-NSGA-III solutions for Crash-worthiness problem.

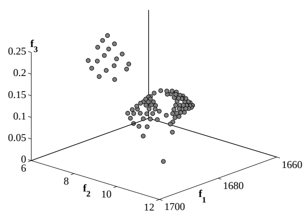

Fig. 24. A^2-NSGA-III solutions for Crash-worthiness problem.

problem using all three approaches keeping a population size of 92 and the number of reference points as 91. Rest all parameters are kept the same as before and each algorithm is run for 500 generations. Figures 22, 23, and 24 show the obtained front using NSGA-III, A-NSGA-III, and A^2-NSGA-III, respectively. Clearly, the distribution obtained using A^2-NSGA-III is the best as compared to others. This practical problem demonstrates that even in a complicated shape of non-dominated front, the proposed A^2-NSGA-III can be effective.

Car Side Impact Problem: This is also a three-objective problem but has 10 constraints. The problem aims at minimizing the weight of car, the pubic force experienced by a passenger, and the average velocity of the V-Pillar responsible for withstanding the impact load [3]. We choose 91 reference points and use a population size of 92. We run all the three approaches for 500 generations. The obtained fronts are shown in Figures 25, 26, and 27. As we can see the distributions obtained using A-NSGA-III and A^2-NSGA-III are similar, but are considerably better than that obtained using NSGA-III.

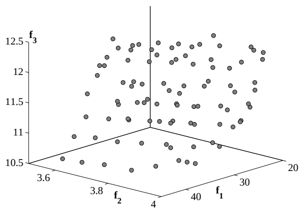

Fig. 25. NSGA-III solutions for car side impact problem.

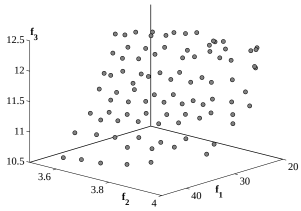

Fig. 26. A-NSGA-III solutions for car side impact problem.

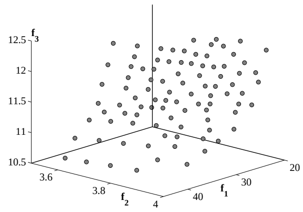

Fig. 27. A^2-NSGA-III solutions for car side impact problem.

Machining Problem: This a four-objective, three-variable constrained problem aimed at optimizing machining performance subject to four constraints [7]. 165 reference points are initialized uniformly over the normalized hyperplane. The problem is solved using all three algorithms keeping a population size of 168. Each algorithm is run for 750 generations and 20 such runs are made. The best, median, and worst hypervolume values of the obtained points are shown in Table 6. A^2-NSGA-III performs much better than the other two approaches, thereby indicating its efficacy.

Table 6. Best, median and worst hypervolume for machining and water problems.

Problem	M-NSGA-II	AM-NSGA-II	A^2M-NSGA-II
Machn.	2.2339	2.2730	**2.2925**
	2.2688	2.2952	**2.3052**
	2.2856	2.3162	**2.3188**
Water	0.5280	0.5349	**0.5402**
	0.5306	0.5365	**0.5429**
	0.5341	0.5396	**0.5455**

Water Problem: Finally, we consider a five-objective, three-variable, seven-constraint problem taken from the literature [1]. 210 reference points and 212 population members are used. 20 different runs are made for all three algorithms and the best, median, and worst hypervolume values are tabulated in Table 6. It is clear from the table that A^2-NSGA-III performs the best.

7 Conclusions

In this paper, we have suggested an adaptive relocation strategy for reference points in the recently proposed many-objective NSGA-III procedure. The proposed method attempts to alleviate the limitations of a previously proposed adaptive strategy. On a number of unconstrained and constrained three to eight-objective optimization problems, it has been found that the proposed strategy is able to find a better distribution of trade-off points on the entire Pareto-optimal front. On a set of four engineering many-objective optimization problems, the proposed A^2-NSGA-III procedure has also been able to find a better distribution of points, both visually and in terms of the hypervolume measure. The suggestion of NSGA-III and results with its current adaptive version open up new directions for handling many-objective optimization problems efficiently.

Acknowledgments: Authors acknowledge the support provided by Academy of Finland under grant number 133387.

References

1. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.
2. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
3. K. Deb and H. Jain. An improved NSGA-II procedure for many-objective optimization Part I: Problems with box constraints. Technical Report 2012009, Indian Institute of Technology Kanpur, 2012.
4. K. Deb and H. Jain. An improved NSGA-II procedure for many-objective optimization, Part II: Handling constraints and extending to an adaptive approach. Technical Report 2012010, Indian Institute of Technology Kanpur, 2012.
5. K. Deb and H. Jain. Handling many-objective problems using an improved NSGA-II procedure. In *Proceedings of World Congress on Computational Intelligence (WCCI-2012)*, pages 1–8, Piscatway: IEEE Press, 2012.
6. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, pages 105–145. London: Springer-Verlag, 2005.
7. M. Ghiassi, R. E. Devor, M. I. Dessouky, and B. A. Kijowski. An application of multiple criteria decision making principles for planning machining operations. *IIE Transactions*, 16:106–114, 1984.
8. E. J. Hughes. Evolutionary many-objective optimisation: Many once or one many? In *IEEE Congress on Evolutionary Computation (CEC-2005)*, pages 222–227, 2005.
9. H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Proceedings of Congress on Evolutionary Computation (CEC-2008)*, pages 2424–2431, 2008.
10. X. Liao, Q. Li, W. Zhang, and X. Yang. Multiobjective optimization for crash safety design of vehicle using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35:561–569, 2008.
11. 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*, pages 2693–2696, 2009. ACM.
12. Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evol. Comput., IEEE Transactions on*, 11(6):712–731, 2007.