Generating Uniformly Distributed Points on a Unit Simplex for Evolutionary Many-Objective Optimization

Kalyanmoy Deb[1][0000−0001−7402−9939], Sunith Bandaru[2], and Haitham Seada[3]

[1] Michigan State University, East Lansing, Michigan, USA
[2] University of Skövde, Sweden
[3] Ford Motor Company, Dearborn, USA
kdeb@egr.msu.edu,sunith.bandaru@his.se,seadahai@msu.edu

COIN Report Number 2018010

Abstract. Most of the recently proposed evolutionary many-objective optimization (EMO) algorithms start with a number of predefined reference points on a unit simplex. These algorithms use reference points to create reference directions in the original objective space and attempt to find a single representative near Pareto-optimal point around each direction. So far, most studies have used Das and Dennis's structured approach for generating a uniformly distributed set of reference points on the unit simplex. Due to the highly structured nature of the procedure, this method does not scale well with an increasing number of objectives. In higher dimensions, most created points lie on the boundary of the unit simplex except for a few interior exceptions. Although a level-wise implementation of Das and Dennis's approach has been suggested, EMO researchers always felt the need for a more generic approach in which any arbitrary number of uniformly distributed reference points can be created easily at the start of an EMO run. In this paper, we discuss a number of methods for generating such points and demonstrate their ability to distribute points uniformly in 3 to 15-dimensional objective spaces.

Keywords: Many-objective optimization · Reference points · Das and Dennis points · Diversity preservation.

1 Introduction

Recent evolutionary many-objective optimization algorithms (EMO) use a set of reference directions as guides to parallely direct their search to find a single Pareto-optimal solution along each direction. These so-called decomposition-based EMO methods, such as MOEA/D [18], NSGA-III [5], DBEA [1] are gaining popularity due to their success in handling three to 15-objective problems involving convex, non-convex, multi-modal, disjointed, biased density based, and non-uniformly scaled problems.

One of the requirements of these algorithms is the initial supply of a set of reference directions, a matter which has not been pursued much in the literature. Most studies use Das and Dennis's [4] structured method in which first a set of points are initialized on a M-dimensional unit simplex (where M is the number of objectives): $\mathbf{z} \in [0,1]^M$ satisfying $\sum_{i=1}^{M} z_i = 1$. Thereafter, a reference direction is constructed by a vector originating from the origin and connected to each of these points. The number of points on the unit simplex is determined by a parameter p, which indicates the number of divisions along each objective axis. It turns out that the total number of points on the unit simplex is $\binom{M+p-1}{p}$. For example, if $p = 10$ is chosen for an $M = 3$-objective problem, then the total number of points on the unit simplex is $\binom{12}{10}$ or 66. The 66 points are well distributed on the unit simplex. If an EMO algorithm works well to find a single Pareto-optimal solution for each of these 66 reference lines (obtained by a vector originating from the origin and passing through each point), a well-distributed set of Pareto-optimal solutions will be expected at the end. If more points are desired, p can be increased by one (or, $p = 11$), and the total number of points must jump to 78. In other words, if exactly 70 points are desired on the unit simplex, there is no way we can use Das and Dennis's method to achieve them.

Besides the inability to construct an arbitrary number of points, there is another issue with Das and Dennis's method, which has been problematic. As p increases, the total number of points on the unit simplex increases rapidly, as shown for $M = 10$ in Figure 1 – sublinear plot in the semilog scale indicates less than exponential behavior. This requires a large population size to find a single Pareto-optimal solution for each direction. Moreover, most of the structured points lie on the boundary of the unit simplex and very few points lie in the interior of the simplex. Calculations reveal that, when $p < M$, there is no interior point, and when $p = M$, there is exactly one interior point. With $p > M$ more points are in the interior, but the number of such points is only $\binom{p-1}{p-M}$, which is only a tiny fraction of all Das and Dennis's points. Figure 1 shows that for $M = 10$-objective problem, the proportion of interior Das and Dennis's points grow with p. For $p < M$, the proportion is zero and then it starts to grow, but the proportion is still very low compared to the total number of points created. For example, for $p = 15$, there are a total of 1,307,504 points, of which only 0.15% (or only 2,002) points are in the interior. The rest of the points lie on the boundary of the unit simplex. A fix-up to the above problem has been suggested in the literature [5] by applying the Das and Dennis's method *layer-wise*. In every layer, a small p is chosen to create a few points, but layers are shrunk consecutively so that more interior points are created in the process. Even with this layer-wise procedure, any arbitrary number of points cannot be created.

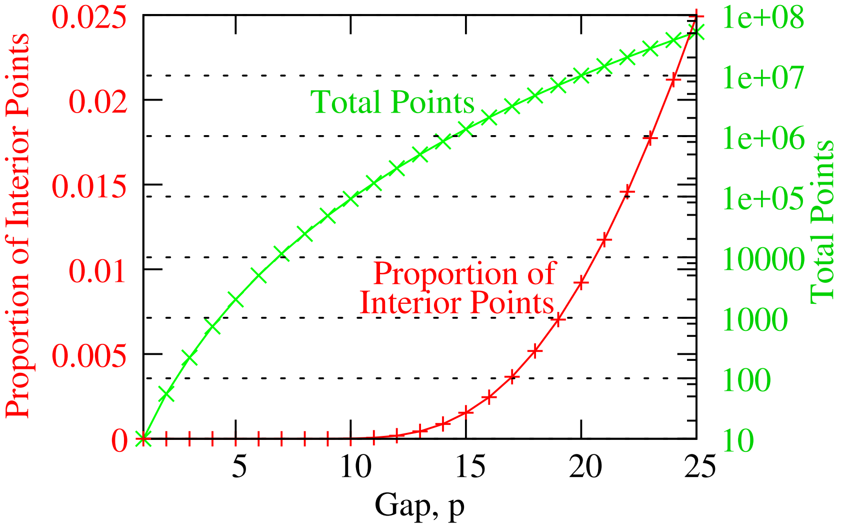

Fig. 1. Proportion of interior points compared to total Das and Dennis's points for $M = 10$-objective problem.

In this paper, we discuss a number of methods by which an arbitrary number of well-distributed points can be created on the unit simplex. Methods based on filling, construction and elimination are first described. The methods are then applied in 3 to 15-dimensional objective spaces to show their effectiveness. Section 2 describes a few standard space filling methods which can be used for the purpose. Section 3 then describes bottom-up approaches that build a set of reference points from a few seed points. Section 4 proposes an opposite scenario, in which a well-distributed set of reference points are chosen from an initial large set of points on the unit simplex. Results using some of these methods are shown in terms of the hypervolume measure in Section 5. Finally, conclusions are drawn in Section 6.

1.1 Motivation

The choice of reference directions in decomposition-based EMO methods is important, since the distribution of resultant Pareto-optimal solutions largely depends on them. If a near-uniform distribution of reference points can be supplied on a unit simplex, the corresponding reference directions are expected to produce a well-distributed set of Pareto-optimal points. In this paper, we propose a number of philosophies of finding n reference points $\mathbf{z}^{(i)}, i = 1, \ldots, n$ on a standard $(M-1)$-simplex[4], such that $\sum_{k=1}^{M} z_k^{(i)} = 1$ holds for each i. Being a preliminary study, we do not consider generating reference points on an arbitrary simplex [8] or on convex and concave manifolds [12].

2 Filling Methods

In these methods, the set of n points will be created on the unit simplex by using a standard filling technique. We describe a few techniques for this purpose.

2.1 Methods Based on Design of Experiments

In the design and analysis of computer experiments, the goal is to capture the effect of inputs on one or more outputs as best as possible with as few expensive experiments as possible. This problem can be restated as: "how to sample computer experiments in the input space such that the input-output relationship are maximally captured?". The chosen set of experiments or samples is called an experimental design. Classical experimental designs, which were originally developed for physical experiments, include the following:

1. **Full and fractional factorial designs:** Full factorial design [7] places a sample at every possible input configuration so that all main effects and interaction effects can be captured. This approach can easily become intractable for a high number of dimensions. Fractional factorial designs are more practical as they use only a subset of complete factorial designs to capture just the main effects and low-order interactions.
2. **RSM designs:** Response surface methodology (RSM) [2] uses polynomials of various degrees to model the input-output relationship. RSM employs a variety of designs, such as full and fractional factorial, central composite design, Box-Behnken design and sequential method, which are aimed at minimizing errors at the design points and keeping biases in the estimated coefficients small.
3. **Optimal designs:** Optimal designs are aimed at optimizing various statistical criteria related to estimation and prediction. For example, D-optimal and A-optimal designs, respectively, minimize the determinant and trace of $X^T X$ (where X is the design matrix), which reduce estimation variance. On the other hand, G-optimal and Q-optimal designs respectively minimize the maximum and average prediction variance over the design points [3]. Here, we consider D-optimal design as a representative method of this class.

2.2 Space Filling Methods

Classical experimental designs described above are also called *model-dependent* designs because they require the knowledge of an underlying model. When no prior information about the input-output relationship is known, the general strategy is to assume that important features of the input-output relationship are equally likely to be present in all parts of the input space. In order to capture these features through an experimental design, the samples are spread evenly throughout the input space. Such designs are called *model-independent* or *space-filling* designs. The three important categories of space-filling designs are:

[4] A standard $(M-1)$-simplex has M vertices in $\mathbb{R}^M$, each of which is one unit from the origin along each axis.

1. **Orthogonal arrays:** An orthogonal array [14] of *strength* d and *index* λ for κ factors ($\kappa > d$), each with s levels, is an experimental design that, upon projection onto any subset of d dimensions, resembles a full factorial, with each design point replicated λ times. The total number of designs required is therefore, $n = \lambda s^d$. The corresponding orthogonal array is denoted as (n, κ, s, d). $(s^3, 4, s, 2)$. Since, the number of points come as in a structured manner and cannot be set arbitrarily, we do not consider this method here.
2. **Latin hypercube design and sampling (LHS):** A Latin hypercube design is any orthogonal array of strength $d = 1$ and index $\lambda = 1$. Since $N = s$, this design gives the flexibility to generate an arbitrary number of samples when the factors are continuous. Latin hypercube sampling [13] involves dividing each factor into N equal intervals. Each of the N required designs is obtained by randomly selecting a previously unselected interval in each factor and sampling a single value from it.
3. **Number theoretic methods:** Number theoretic methods, originally developed for quasi-Monte Carlo integration, are aimed at creating points uniformly in the input space by minimizing different discrepancy metrics, which are measures of uniformity. Various low-discrepancy (also called quasi-random) sequences are available in literature. Popular among them are Halton set [9], Hammersley set [10], Sobol set [17] and Faure [6] sequences. When using low-discrepancy sequences, randomization of samples can be achieved by skipping over used sequences completely or partially. Here, we use three methods as representatives of this class.

Mapping Onto Unit Simplex All filling methods described above generate points in an M-dimensional hypercube ($z_k \in [0, 1]$). In order to map these points onto the $(M-1)$-dimensional unit simplex, we adopt the following approach, suggested in [16]:

1. Generate points $\mathbf{z}^{(i)}, i = 1, \ldots, n$ in an $M - 1$ dimensional unit hypercube.
2. Set $i \leftarrow 1$
3. Sort ordinates $\{z_1^{(i)}, z_2^{(i)}, \ldots, z_{M-1}^{(i)}\}$ of $\mathbf{z}^{(i)}$ in ascending order.
4. Let $\{y_1^{(i)}, y_2^{(i)}, \ldots, y_{M-1}^{(i)}\}$ be the sorted ordinates.
5. Define $y_0^{(i)} = 0$ and $y_M^{(i)} = 1$, so that $y_0^{(i)} < y_1^{(i)} < y_2^{(i)} < \ldots < y_{M-1}^{(i)} < y_M^{(i)}$.
6. Then, set $z_k^{(i)} \leftarrow y_k^{(i)} - y_{k-1}^{(i)}$ for $k = 1, \ldots, M$ to form the mapped point $\mathbf{z}_k^{(i)}$.
7. If $i < n$, then set $i \leftarrow (i + 1)$ and go to Step 3, else Stop.

Note that $\sum_{k=1}^{M} z_k^{(i)} = 1$ is satisfied for each i.

In this paper, we will use the following filling methods because they allow us to choose an arbitrary number of points: D-optimal designs (DOD), Latin hypercube sampling (LHS), Halton set (HAL), Hammersley set (HAM) and Sobol set (SOB).

2.3 Structured Filling Methods

Das and Dennis's Method (DAS) As mentioned in Section 1, Das and Dennis's method is a structured approach which requires an integer gap parameter $p(\geq 1)$ and then creates $\binom{M+p-1}{p}$ points. The method is scalable to any number of objectives (M), but has several drawbacks:

- Number of points cannot be set arbitrarily,
- Most points lie on the boundary of the unit simplex, which may not be of interest to decision-makers, and
- The approach is not easily moldable to incorporate preference information.

Due to popular use of this method, we consider it in our comparison base and call this method DAS.

Layer-wise Das and Dennis's Method Layer-wise construction process is illustrated in Figure 2. A relatively small value of p ($< M$) is used for each layer. The first layer covers the entire unit simplex, but the subsequent layers use a shrunk unit simplex, as shown in the figure. For example, instead of using $p = 12$ for a three-objective ($M = 3$) problem totalling 91 points, the use of three layers, as shown in the figure, with layer-wise $p = (3, 2, 1)$ requires (10, 6, 3) or 19 points in total of which only 9 points are on the boundary. Points from Layer 2 and above are guaranteed to lie in the interior in such a construction, but the uniformity of the points gets lost in the process. In addition, the layered approach can end up with different layouts for the same number of points. For example, with 5 objectives, generating 50 reference points can be done either using $p = (3, 2)$ or $p = (2, 2, 2, 1)$. Moreover, although using layers allows more flexibility in number of points than the original method, but still any arbitrary number of points is not possible to be created.

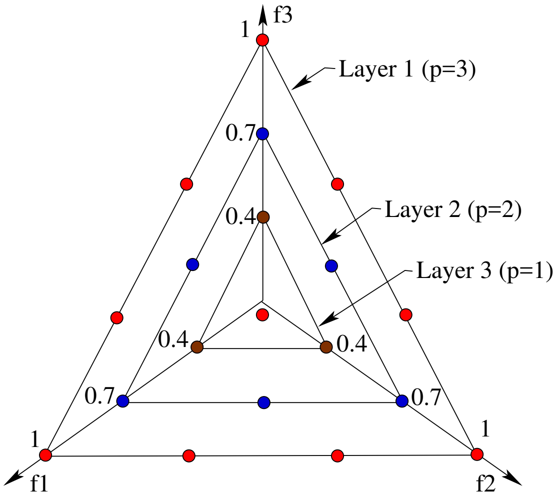

Fig. 2. Layer-wise construction of Das and Dennis's points for $M = 3$-objective problem.

2.4 Probabilistic Filling Methods

A structured probability distribution for each objective can be chosen such that the sum of the objective values is exactly one. One such method was proposed in [11], in which the first objective z_1 is chosen in [0, 1] with a probability distribution proportional to $(1-z_1)^{M-2}$. This can be achieved by first choosing a random number $u_1 \in [0, 1]$ and then computing $z_1 = 1 - \sqrt[M-1]{u_1}$. Thereafter, z_k is computed with another random number $u_k \in [0, 1]$, as $z_k = \left(1 - \sum_{j=1}^{k-1} z_j\right)\left(1 - \sqrt[M-k]{u_k}\right)$. The process is continued until z_{M-1} and the final objective value is computed as $z_M = 1 - \sum_{j=1}^{M-1} z_j$. We call this method JAS in this paper.

Other Structured Methods There exists a number of conformal mapping methods [15] in which uniformly distributed points on a hyperbox can be mapped into a unit simplex. Since we are interested in a near uniform distribution of points in the unit simplex, it may not be easy to use such methods efficiently.

3 Construction Methods

Construction methods uses a bottom-up approach in which the procedure starts with a single point $\mathbf{z}^{(0)}$ on the unit simplex. Thereafter, points are added one by one in stages until a set of n points are obtained with a near uniform distribution. The addition of points can be achieved by using a predefined procedure or by a sophisticated optimization procedure of maximizing the uniformity of points at every stage. At a stage, when k points are already found, the following optimization procedure can be applied to obtain the $(k+1)$-th point $\mathbf{z}^{(k+1)}$:

$$\begin{aligned}
\text{Maximize } & \min_{i=1}^{k} \text{Dist}(\mathbf{z}^{(k+1)}, \mathbf{z}^{(i)}), \\
\text{Subject to } & \sum_{j=1}^{M} z_j^{(k+1)} = 1, \\
& 0 \le z_j^{(k+1)} \le 1, \quad \text{for } j = 1, \ldots, M.
\end{aligned} \tag{1}$$

In this problem, there are only M variables $z_i^{(k+1)}$ and one constraint. Other diversity metrics can also be used instead of the minimum Euclidean distance to all existing k points. The only drawback of this approach is that the above optimization needs to be applied $(n-1)$ times and the computation gets expensive with increasing k. The final outcome of n points will depend on the initial point chosen, which can be a random point on the unit simplex, or its centroid.

3.1 Maximally Sparse Creation Method (MSC)

Instead of starting with a single initial point, the above procedure can be seeded with more than one well-distributed points on the unit simplex. For example, the process can be started with M vertices as initial points or with m points from Layer 1 specification (with a small p $(< M)$), as described in Section 2.3. The remaining $(n-M)$ or $(n-m)$ points, as the case may be, can be created by using the above optimization procedure in stages. In this study, we use the vertices as initial points and call this method as MSC. The optimization problem in (1) is solved using MATLAB's `fmincon()` function. The constraint is relaxed to an inequality $\sum_{j=1}^{M-1} z_j^{(k+1)} \leq 1$ by solving the problem for first $M-1$ variables. The last variable is set to $z_M^{(k+1)} = 1 - \sum_{j=1}^{M-1} z_j^{(k+1)}$ once `fmincon()` terminates.

4 Elimination-based Methods

Contrary to construction methods, a completely opposite process can be devised. Starting with a large set $(\mathcal{S})$ of structured or random points on the unit simplex, a procedure can be devised to eliminate neighboring points. This can be done either by eliminating one point at a time similar to a pruning method or by eliminating multiple points at a time.

4.1 Maximally Sparse Selection Methods (MSS)

This approach starts by filling $\mathcal{S}$ with a large number of uniformly randomly generated points. Then $\mathcal{W}$ is initialized as the set of all extreme points $(1,0,\ldots,0)^T$, $(0,1,\ldots,0)^T$, …, $(0,0,\ldots,1)^T$. Finally, the procedure selects the rest $(n-M)$ points one at a time. The point which is maximally away from the already selected set of points points $\mathcal{W}$ is picked, added to $\mathcal{W}$ and removed from $\mathcal{S}$. The procedure continues until a total of n reference points is reached. Thus, at a stage in which k points are already obtained, we choose the next point $\mathbf{z}^{(k+1)}$, as follows:

$$(k+1) = \text{argmax}_{j\in\mathcal{S}, i\in\mathcal{W}} \sum_{i=1}^{k} \text{Dist}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}). \tag{2}$$

If the original large set $\mathcal{S}$ is a random set of points on the unit simplex [19], we call this method MSS-R and when the original set $\mathcal{S}$ is a large set of points created by Das and Dennis's method with a large p, we call it MSS-D.

Instead of starting with an extreme point, the point closest to the centroid of the entire set $\mathcal{S}$ can be used to start the procedure, as an alternative method.

4.2 Reductive Methods (RED)

In this method, we cluster the large set of points $\mathcal{S}$ into n separate clusters based on Euclidean distance. Then, we choose one representative point from each cluster to select exactly n points. If $\mathcal{S}$ is a random set of points on the unit simplex, we call it RED-R and if $\mathcal{S}$ is chosen using Das and Dennis's method with a large p, we call it RED-D.

These methods may lose boundary points. One way to overcome this issue, is to ensure that for boundary clusters, we choose a boundary point in order to have maximum coverage over the entire unit simplex. Alternatively, the point closest to the centroid of each cluster can be chosen first and then the set of n points can be stretched to extend to the unit simplex boundary. In RED-R and RED-D, once the un-stretched-yet points are generated, $\mathcal{W}'$, the smallest value for each objective i is subtracted from all i components of all the points in $\mathcal{W}'$. This step keeps pushing the points – that used to be on the unit simplex – towards the origin, until for each objective i at least one point exists whose i-th component is Zero. Then all the points are normalized again to fall back on the unit simplex, and these form the targeted $\mathcal{W}$. Since stretching does not guarantee keeping extreme points, we added an additional step to insert them (RED-DS) in place of their closest neighbors.

5 Results

In this section, we present results obtained from 13 different methods described in previous sections on M=$\{3, 5, 8, 10, 15\}$-objective problems for finding $n = \{50, 100, 150, 200, 250, 300\}$ points. The methods used in our study are (i) D-optimal designs (DOD), (ii) Latin hypercube sampling (LHS), (iii) Halton (HAL), (iv) Hammersley (HAM), (v) Sobol (SOB), (vi) Jaszkiewicz (JAS), (vii) Das and Dennis (DAS), (viii) Maximally Sparse Creation (MSC), and (ix) Maximally Sparse Selection with random initial set (MSS-R), (x) Maximally Sparse Selection with Das and Dennis's initial set (MSS-D), (xi) Reduction method with random initial set (RED-R), (xii) Reduction method with Das and Dennis's initial set (RED-D), and (xiii) Reduction method with Das and Dennis's initial set and guaranteed extreme points (RED-DS).

To compare the performance of the methods, we have used the hypervolume measure using the vector $(1.01, \ldots, 1.01)^T$ as the reference vector, as we know that the unit simplex has a nadir point $(1, \ldots, 1)^T$. In all plots, we show a normalized hypervolume metric obtained by dividing the obtained hypervolume value for a method with the maximum hypervolume, computed as follows: N-HV $=$ HV/HV$_{\max}$, where HV$_{\max} = 1.01^M - 1/M!$.

Figure 3 compares all 13 methods in terms of box-plots for $M = 3$ objectives and $n = 20$ points. When $n = 50$ points are required, a different distribution will occur and resulting hypervolume values are plotted in Figure 4 for all 13 methods. It is clear that the hypervolume is the best for RED methods.

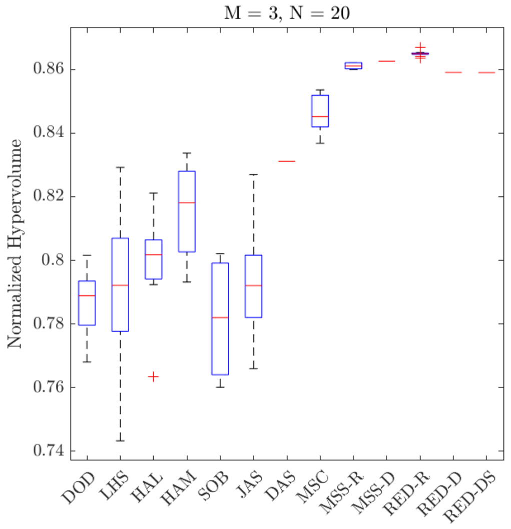

Fig. 3. Normalized hypervolume for $M = 3$ and $n = 20$.

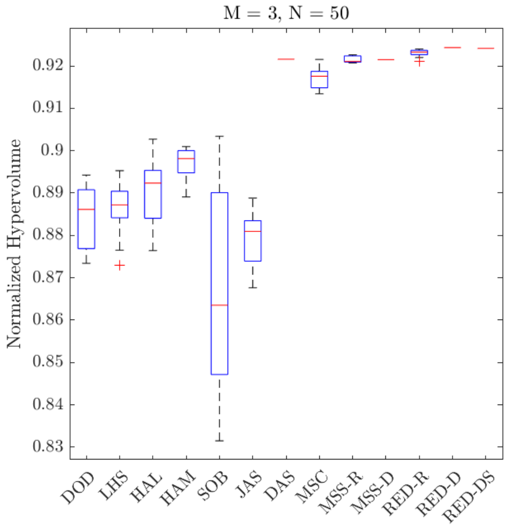

Fig. 4. Normalized hypervolume for $M = 3$ and $n = 50$.

In order to show the sensitivity of the obtained distribution on the desired number of points (n), we use two elimination methods – RED-DS and MSS-D – $n = 50$ and 51 points for a three-objective problem, and obtain two distributions of points. Figures 5 and 6 show the difference in their distributions using RED-DS and MSS-D, respectively. The first plot indicates that an addition of an extra point in the set changes the arrangement in the intermediate part. Since MSS-D uses a sequential and deterministic selection method, the first 50 points for the $n = 51$ case will exactly be identical to the $n = 50$ case. Although both produce a well-distributed set of points, the clustering approach (in RED) has the dependence on n, and may be a better approach.

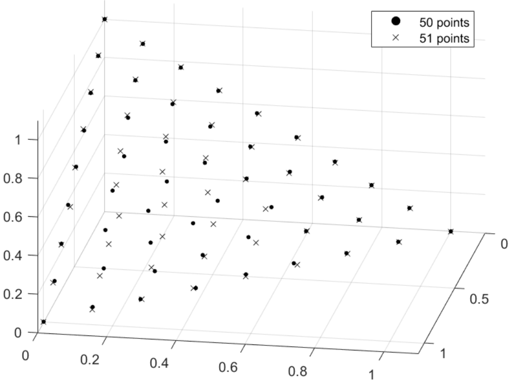

Fig. 5. Distribution of 50 and 51 points using RED-DS method for $M = 3$.

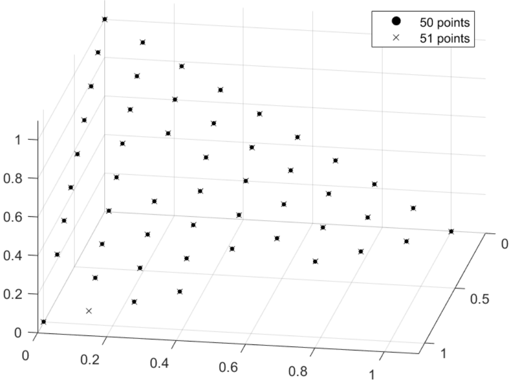

Fig. 6. Distribution of 50 and 51 points using MSS-D method for $M = 3$.

Figures 7 to 10 show box plots of normalized hypervolume for 5 to 15-objective problems with different n values. All plots show how MSS and MSC has the most robust results across all dimensions. At higher dimensions, the performance of RED methods degrades due to the tendency of clustering to avoid boundary points.

6 Conclusions

In this paper, we have attempted to address an important issue related to evolutionary many-objective optimization (EMO) algorithms. While most EMO methods use a structured approach for allocating a set of reference points for leading the search, EMO researchers have always felt the need to make the process more flexible in specifying an arbitrary number of points on the unit simplex. In this paper, we have discussed a number of filling, construction and elimination methods for this purpose. Through extensive experiments in 3 to 15-dimensional objective spaces, we have compared the performance of a few of the proposed methods. Results based on the hypervolume metric have indicated the following:

1. Elimination methods (MSS-R, MSS-D, RED-R, and RED-D) are, in general, better than other methods considered in this paper at lower dimensions.
2. MSS-D procedure performs the best overall.
3. Construction method MSC and structured method DAS also perform well, particularly for higher dimensions.

In the future, we plan to make a comparison based on the computational complexity. Other methods, such as a simultaneous optimization of all n points, can also be included as global methods. The study can be extended to find a biased distribution of reference points, particularly if preference information is available. In some EMO applications, users may be interested in focusing on a particular

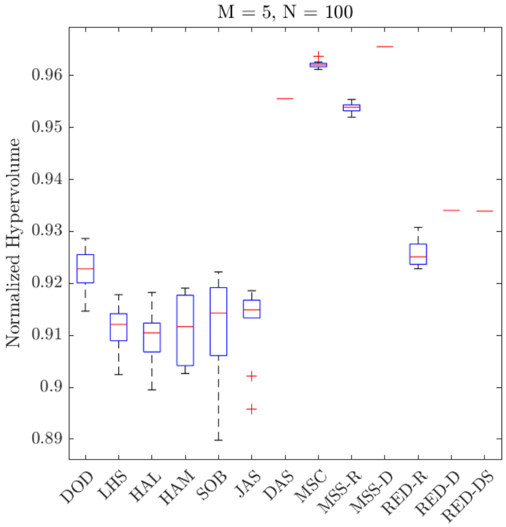

Fig. 7. Normalized hypervolume for $M = 5$ and $n = 100$.

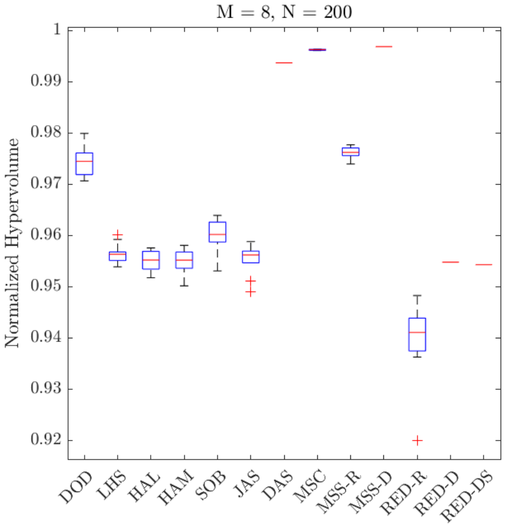

Fig. 8. Normalized hypervolume for $M = 8$ and $n = 200$.

part of the Pareto-optimal front. Methods that are able to create reference points on a specific region of the unit simplex will be another useful extension of this study.

References

1. Asafuddoula, M., Ray, T., Sarker, R.: A decomposition-based evolutionary algorithm for many objective optimization. IEEE Transactions on Evolutionary Computation **19**(3), 445–460 (2015)
2. Box, G.E.P., Wilson, K.B.: On the experimental attainment of optimum conditions. Journal of the Royal Statistical Society. Series B (Methodological) **13**(1), 1–45 (1951)
3. Chen, V.C., Tsui, K.L., Barton, R.R., Allen, J.K.: A review of design and modeling in computer experiments. Handbook of Statistics **22**, 231–261 (2003)
4. Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization **8**(3), 631–657 (1998)
5. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
6. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arithmetica **41**(4), 337–351 (1982)
7. Fisher, R.A.: The arrangement of field experiments. Journal of the Ministry of Agriculture of Great Britain **33**, 503–513 (1926)
8. Grimme, C.: Picking a uniformly random point from an arbitrary simplex. Tech. rep. (2015), University of Münster
9. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik **2**(1), 84–90 (1960)
10. Hammersley, J.M.: Monte carlo methods for solving multivariable problems. Annals of the New York Academy of Sciences **86**(3), 844–874 (1960)
11. Jaszkiewicz, A.: On the performance of multiple-objective genetic local search on the 0/1 knapsack problem – A comparative experiment. IEEE Transactions on Evolutionary Computation **6**(4), 402–412 (2002)
12. Martínez, S.Z., Hernández, V.A.S., Aguirre, H., Tanaka, K., Coello, C.A.C.: Using a family of curves to approximate the pareto front of a multi-objective optimization problem. In: International Conference on Parallel Problem Solving from Nature. pp. 682–691. Springer (2014)
13. McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)

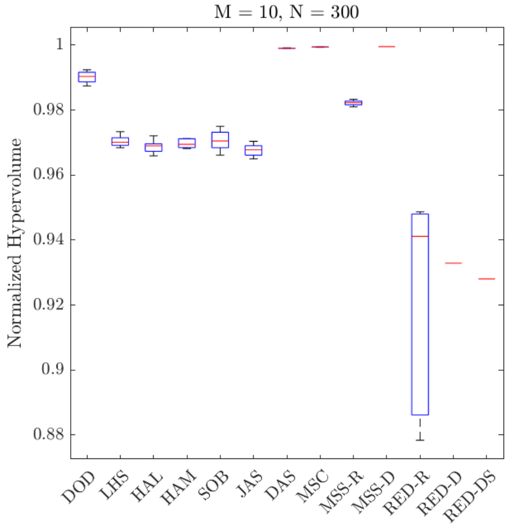

Fig. 9. Normalized hypervolume for $M = 10$ and $n = 300$.

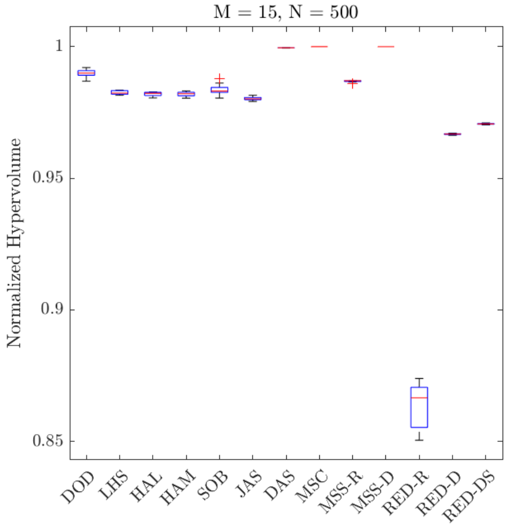

Fig. 10. Normalized hypervolume for $M = 15$ and $n = 500$.

14. Rao, C.R.: Factorial experiments derivable from combinatorial arrangements of arrays. Supplement to the Journal of the Royal Statistical Society **9**(1), 128–139 (1947)
15. Schinzinger, R., Laura, P.A.A.: Conformal Mapping: Methods and Applications. Dover Books on Mathematics (2003)
16. Smith, N.A., Tromble, R.W.: Sampling uniformly from the unit simplex. Tech. rep. (2004), Johns Hopkins University
17. Sobol, I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics **7**(4), 86–112 (1967)
18. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. Evolutionary Computation, IEEE Transactions on **11**(6), 712–731 (2007)
19. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: IEEE Congress on Evolutionary Computation - CEC 2009. pp. 203–208. IEEE (2009)