Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization

Julian Blank, Kalyanmoy Deb, Yashesh Dhebar, Sunith Bandaru, and Haitham Seada

***Abstract*—Most evolutionary many-objective optimization (EMaO) algorithms start with a description of a number of predefined set of reference points on a unit simplex. So far, most studies have used the Das and Dennis's structured approach for generating well-spaced reference points. Due to the highly structured nature of the procedure, this method cannot produce an arbitrary number of points, which is desired in an EMaO application. Although a layer-wise implementation has been suggested, EMO researchers always felt the need for a more generic approach. Motivated by earlier studies, we introduce a metric for defining well-spaced points on a unit simplex and propose a number of viable methods for generating such a set. We compare the proposed methods on a variety of performance metrics such as hypervolume, deviation in triangularized simplices, distance of the closest point pair, and variance of the geometric means to nearest neighbors in up to 15-dimensional spaces. We show that an iterative improvement based on Riesz s-Energy is able to effectively find an arbitrary number of well-spaced points even in higher-dimensional spaces. Reference points created using the proposed Riesz s-Energy method for a number of standard combinations of objectives and reference points as well as a source code written in Python are available publicly at https://www.egr.msu.edu/coinlab/blankjul/uniform.**

***Index Terms*—Many-objective optimization, Reference points, Das-Dennis points, Diversity preservation, Riesz s-Energy.**

I. Introduction

RECENT evolutionary many-objective optimization algorithms (EMaO) use a set of reference directions as guides to parallelly direct their search to find a single Pareto-optimal solution along each direction. These so-called decomposition-based EMaO methods, such as MOEA/D [1], NSGA-III [2], [3], 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 is initialized on an 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 gaps between two consecutive points along an objective axis. It turns out that the total number of points (n) on the unit simplex is

$$n = C_p^{M+p-1}. \tag{1}$$

For example, if $p = 10$ is chosen for a three-objective problem ($M = 3$), then the total number of points on the unit simplex is $C_{10}^{3+10-1} = \binom{12}{10}$ or 66. These 66 points are well-spaced with an identical distance to their nearest neighbor on the unit simplex. If an EMaO 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-spaced 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-Dennis* approach to achieve them. The jump becomes severe with higher objectives, such as for a 10-objective problem, p values from 2 to 6 requires 55, 220, 715, 2,002, and 5,005 points.

Besides the inability to construct an arbitrary number of points, there is another issue with the *Das-Dennis* approach. As M increases, the number of total points on the unit simplex increases rapidly, as shown for $M = 10$ in the right vertical axis in Figure 1. Although a sub-linear plot in the semi-log scale indicates weaker than exponential behavior, an extremely large number of reference points is generated for a reasonable value of p. Since the population size of an EMaO is almost the same as the number of reference points, this requires an extremely large population size.

Another crucial problem for a large value of M is that all points in a reasonably sized reference set 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, irrespective of the size of M. With $p > M$ points begin to appear in the interior, but the number of such interior points is $n_I = \binom{p-1}{p-M}$, which is only a tiny fraction

$$\rho_I = \frac{n_I}{n} = \frac{p!\,(p-1)!}{(p-M)!\,(M+p-1)!} \tag{2}$$

of all n *Das-Dennis* points. Figure 1 shows that for a 10-objective problem ($M = 10$), the proportion of interior *Das-Dennis* points grows with p, but the proportion (left vertical axis) is small. To provide an example, with $p = 15$, there are a total of 1,307,504 reference points, of which only 0.15% (or only 2,002) points are in the interior. The rest of the points lie on boundary of the unit simplex.

Julian Blank, Kalyanmoy Deb and Yashesh Dhebar are at Michigan State University, East Lansing, USA (email: {blankjul,kdeb,dhebarya}@msu.edu). Sunith Bandaru is with University of Skövde, Sweden (email: sunith.bandaru@his.se), and Haitham Seada is with Ford Motor Company, Dearborn, USA (email: seadahai@msu.edu).

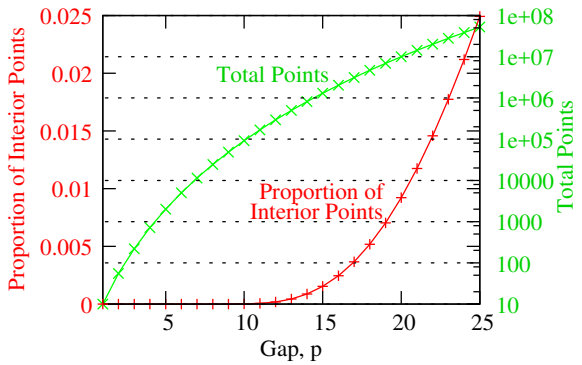


Fig. 1. Proportion of interior points compared to total Das-Dennis's points for 10-objective problem ($M = 10$).

In this paper, we extend a previous preliminary study [5] in light of existing literature to propose a number of efficient methods to generate an arbitrary number of well-spaced point-set on the unit simplex. Section II describes a number of existing *space-filling* methods which can be used to create points on a hypercube, and a few methods to map those points onto the unit simplex. In Section III we describe a way to measure uniform spacing of a point-set on a unit simplex and describe different approaches to create such points. In Section IV the well-spaced properties of the points obtained by three proposed methods from 3 up to 15-dimensional problems are evaluated by using a variety of metrics. Some practical matters are addressed in Section V and conclusions are drawn in Section VI.

II. Related Work

The task of creating a well-spaced set of points on the unit simplex has two different aspects as the problem definition already implies: First, the points should be uniformly and as evenly distributed as possible so that no sparse or crowded areas exist. Second, all points must lie on the unit simplex in the first quadrant, which means that each point's objective values have to be positive and their sum must equal to one. Creating points uniformly without restricting them to be on the unit simplex has been studied extensively and is applied during the design and analysis of computer experiments, where the goal is to capture the effect of inputs on one or more outputs in the best possible manner with as few expensive experiments as possible.

The literature distinguishes between model-dependent and model-independent methods. Model-dependent methods, such as full and fractional factorial designs [6], RSM designs [7], and D-optimal or G-optimal designs [8] are restricted to small dimensional problems.

A. Model-independent Methods

With no prior information about the input-output relationship, these methods 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. Three important categories of space-filling designs are:

1) **Orthogonal arrays:** An orthogonal array [9] 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.
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 [10] 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 the literature. Popular among them are Halton set [11], Hammersley set [12], Sobol set [13] and Faure [14] sequences. When using low-discrepancy sequences, randomization of samples can be achieved by skipping over used sequences completely or partially.

Each of these methods require a way to transform the points inside a hypercube to a unit simplex [15].

B. Mapping onto Unit Simplex

All filling methods described above generate points in an M-dimensional hypercube ($z_k \in [0, 1]$). However, all points need to lie on the unit simplex to be a feasible solution for our problem. Assuming all coordinates are already positive, a naive approach is to divide each point's (k-th) coordinate by the sum of all coordinates:

$$z_k^{(i)} \leftarrow z_k^{(i)} \Big/ \sum_{j=1}^{M} z_j^{(i)}. \tag{3}$$

A study [16] has shown that this mapping function is not appropriate, because, despite creating a uniformly distributed set of points on the hypercube, the well-spaced nature of points in the unit simplex is not guaranteed. In order to map these points onto the $(M-1)$-dimensional unit simplex, the authors proposed:

1) Generate points $\mathbf{z}^{(i)}, i = 1, \ldots, n$ in an $(M-1)$-dimensional unit hypercube.
2) Set $i \leftarrow 1$.
3) Sort coordinates $\{z_1^{(i)}, z_2^{(i)}, \ldots, z_{M-1}^{(i)}\}$ of $\mathbf{z}^{(i)}$ in ascending order.

4) Let $\{y_1^{(i)}, y_2^{(i)}, \ldots, y_{M-1}^{(i)}\}$ be the sorted coordinates.
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 coordinate $z_k^{(i)}$.
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 point i.

C. Probabilistic Filling Methods

Instead of creating a point-cloud in the hypercube and then using a mapping function to project it on a unit simplex, points can be directly sampled on the desired unit simplex. This can be achieved by choosing a structured probability distribution for each objective, so that the sum of the objective values is exactly one. One such method was proposed in [17], in which the first objective z_1 is chosen in [0,1] with a probability $\left(1 - z_1{}^{M-2}\right)$. 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_j is computed with another random number $u_2 \in [0,1]$, as $z_j = \left(1 - \sum_{j=1}^{k-1} z_j\right)\left(1 - \sqrt[M-j]{u_2}\right)$. The process is continued until z_{M-1} and the final objective value is computed as $z_M = 1 - \sum_{i=1}^{M-1} z_i$. There exists a number of conformal mapping methods [18] in which uniformly distributed points on a hypercube can be mapped onto a unit simplex. Since we are interested in a well-spaced distribution of points on the unit simplex, it may not be easy to use such methods efficiently. A suggestion of creating each member of an M-dimensional vector using an exponential probability distribution and then normalizing it is one way to generate a well-spaced distribution on a unit simplex [19]. Other methods have been proposed to meet the uniform distribution property, however, they cannot be used easily to generate a finite number of well-spaced points on any dimensional unit simplex.

D. Das-Dennis's Approach

In 1958, Scheffé [20] first proposed the "simplex-lattice" approach to generate a well-spaced point-set on a unit simplex in the context of experimental design. Fang and Wang [15] generalized the approach to create well-spaced points on a contracted unit simplex. However, Das and Dennis [4] used the simplex-lattice approach for multi-objective optimization for the first time, hence we call the approach in their name.

As mentioned in Section I, *Das-Dennis* approach [4] is a structured approach which requires an integer gap parameter p (≥ 1) and then creates $\binom{M+p-1}{p}$ structured points. The method is scalable to any number of objectives (M), but has a few 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 changeable to incorporate preference information.

As an improvisation, a layer-wise construction process is implemented as shown 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 (i.e. down-scaled) unit simplex. For instance, instead of using $p = 12$ for a three-objective ($M = 3$) problem totalling 91 points, the use of three layers 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 $p = (2, 2, 2, 1)$. Creation of a well-spaced distribution with arbitrary number of points is not possible by this method. Also, the user needs to define the shrinkage factor for each layer, which is not an easy task. Hughes [21] proposed a single-objective optimization method in which a large number of random reference vectors are evaluated based on the proximity of associated non-dominated population members to choose a final subset of reference vectors. However, the approach is quite different in principle to the one-time identification of an arbitrary and controlled number of reference points, which is the main goal of this paper.

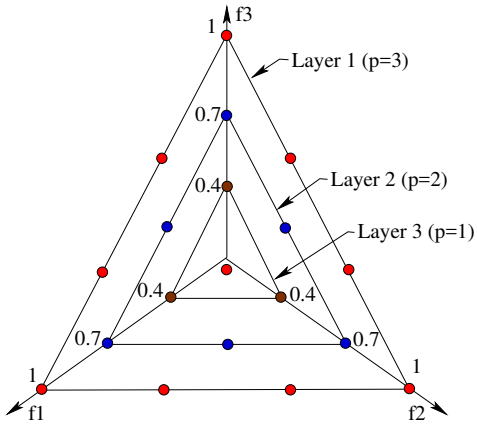


Fig. 2. Layer-wise construction of Das-Dennis's points for three-objective problem ($M = 3$).

III. Proposed Methodologies

In EMaO methods, the choice of reference points on the respective unit simplex is an important matter. For an arbitrary number of desired reference points on the unit simplex, a perfect equi-spaced distribution of them is not possible, but a well-spaced distribution is still desired. Elsewhere [22], a well-spaced distribution of n points (set, $\mathbf{z}$) over a unit simplex was defined using a mean-squared error (MSD) measure, which was proposed to compute by solving an optimization problem. First, a large number of uniformly-distributed points ($\mathbf{y}^{(j)}$, $j = 1, \ldots, K$) on a unit simplex is created. Then, the desired set of n well-spaced points ($\mathbf{z}^{(i)}$, $i = 1, \ldots, n$) is obtained by minimizing the following MSD measure:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{K}\sum_{j=1}^{K} \min_{i=1}^{n} \left\|\mathbf{y}^{(j)} - \mathbf{z}^{(i)}\right\|^2, \\ \text{subject to} \quad & \sum_{m=1}^{M} z_m^{(i)} = 1, \quad i = 1, \ldots, n, \\ & z_m^{(i)} \geq 0, \quad i = 1, \ldots, n, \ m = 1, \ldots, M. \end{aligned} \tag{4}$$

The final outcome depends on the chosen initial points and optimization becomes difficult due to the min-operator. Other number-theoretic concepts were used to suggest a discrepancy measure to theoretically define an 'equi-distributed' or 'uniformly-scattered' point-set [15], but they did not suggest any simple procedure for arriving at a well-spaced point-set for any arbitrary dimension.

Here, we define a closest neighbor distance (CND) among n points on a unit simplex by maximizing the Euclidean distance ($D^{\min}$) between the closest pair of points:

$$\begin{aligned} \text{Maximize} \quad & D^{\min}(\mathbf{z}) = \min_{i,j\in\{1,\ldots,n\},i\neq j} \left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|, \\ \text{subject to} \quad & \sum_{m=1}^{M} z_m^{(i)} = 1, \quad i = 1,\ldots,n, \\ & z_m^{(i)} \geq 0, \quad i = 1,\ldots,n,\ m = 1,\ldots,M. \end{aligned} \tag{5}$$

The constraints force each point $\mathbf{z}^{(i)}$ to lie on the unit simplex. The advantage of this formulation is that it does not require any initial point-set (such as $\mathbf{y}$). However, the above problem involves $n\times M$ variables (set, $\mathbf{z}$), a non-differentiable objective function involving $n(n-1)/2$ quadratic functions, n linear equality constraints and non-negativity constraints of all $n \times M$ variables. A smoother CND optimization problem can be written as follows, by introducing an additional variable ϵ:

$$\begin{aligned} \text{Maximize} \quad & \epsilon, \\ \text{subject to} \quad & \epsilon \leq \left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|, \ \forall i,j \in \{1,\ldots,n\}, i \neq j, \\ & \sum_{m=1}^{M} z_k^{(i)} = 1, \quad i = 1,\ldots,n, \\ & z_m^{(i)} \geq 0, \quad i = 1,\ldots,n,\ m = 1,\ldots,M. \end{aligned} \tag{6}$$

The objective function is linear, but there are now $n(n-1)/2$ quadratic constraints, in addition to n linear equality constraints and non-negativity of variables. There is no provable algorithm for solving such a nonlinear optimization problem. Hence, finding a well-spaced distribution for an arbitrary set of points (n) on a unit simplex remains as a challenging task. Note that for *Das-Dennis* commensurate n points given by Equation 1 for a specific p, by construction, the *Das-Dennis* approach produces every point exactly the same distance from its neighbor. This is precisely a property of an optimal solution. Since every boundary line of the unit simplex has a length of $\sqrt{2}$, p gaps will make two neighboring points to have a distance $D^{\min,*} = \sqrt{2}/p$ between them. When Matlab's fmincon() routine is used to solve the smoother version for a three-dimensional problem, it can find the optimal distribution matching the above $D^{\min,*}$ value for up to $N = 21$ points (with $p = 5$). For larger p, the number of variables to the above problem is too large for fmincon() to find the optimal solution. See Supplementary document for more details.

Instead of trying to solve Equation 6, in this section, we introduce different approaches towards finding a well-spaced distribution and directly present benefits and drawbacks of each method. Also, we present some results to visualize their near-uniformity using standard visualization methods. Moreover, numerical results are presented in Section IV.

A. Sampling Methods

A naive approach is to use any of the space-filling sampling methods described in Section II-A to create points in a hypercube and map them onto the unit simplex by using the method described in Section II-B. In Figure 3a, we show a mapping of 91 points using the Latin Hypercube Sampling. Clearly, the resulting distribution shows a lack of uniformity. We have also observed similar outcome with other methods discussed in Section II-A. This indicates that using a space-filling method and mapping is not sufficient to obtain a well-spaced distribution on the *unit simplex*.

However, Figure 3b indicates that if a large number of points is sampled and then mapped to the unit simplex, the simplex can be covered apparently uniformly. We exploit this observation to propose reduction-based methods (see Section III-C).

B. Construction-based Methods

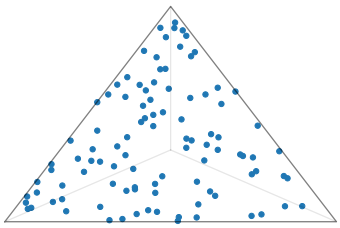
Instead of finding all n points at a time, the construction method uses a bottom-up approach in which the procedure starts with a single point $\mathbf{z}^{(0)}$ (or a number of well-spaced points using methods of Sections II-B or II-C) on the unit simplex. Thereafter, points are added one by one in stages until a set of n points is obtained with a well-spaced distribution. The final outcome of n points will depend on the initial point(s) chosen. The addition of points can be achieved by using a pre-defined procedure or by an optimization procedure of maximizing the uniformity of points at every stage. At a stage, when k points are already found, the following optimization procedure (P_{k+1}), simplified from Equation 5, can be solved to obtain the $(k+1)$-th point $\mathbf{z}^{(k+1)}$ on the unit simplex:

$$\begin{aligned} \text{Maximize} \quad & \min_{i=1}^{k} \left\|\mathbf{z}^{(k+1)} - \mathbf{z}^{(i)}\right\|, \\ \text{subject to} \quad & \sum_{m=1}^{M} z_m^{(k+1)} = 1, \\ & z_m^{(k+1)} \geq 0, \quad m = 1,\ldots,M. \end{aligned} \tag{7}$$

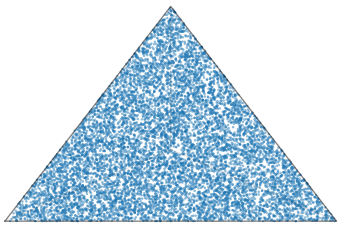
In the above optimization problem for finding a single new point $\mathbf{z}^{(k+1)}$, there are only M variables and one equality constraint. However, the structure of the problem is same as in Equation 5, but fewer variables may allow each problem P_{k+1} to get solved with near-optimality within a reasonable computational effort.

Instead of starting with a single initial point, the above procedure can be seeded with more than one well-spaced point on the unit simplex. For example, the process can start with M vertices as initial points or with m points from Layer 1 specification (with a small p where $p < M$), as described in Section II-D. This is a reasonable choice of starting points because the extreme points should exist in the final point-set.

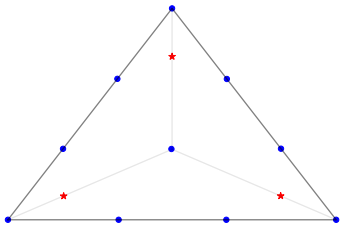
The construction method has two obvious drawbacks: (i) the above optimization method needs to be solved $(n-1)$ times and the computation of the objective function gets more time consuming with increasing k, (ii) each insertion of a new point is dealt independently, and (iii) the procedure may not produce a well-spaced distribution if a small number of points (n) is desired. Figure 3c visualizes the problem for 10 points (blue) which are already added and 3 points (red) which could be added with exactly the same distance to other existing points. It is obvious that if only 11 points are desired, the algorithm



(a) Latin Hypercube Sampling of 91 Points mapped onto the unit simplex.



(b) 10,000 randomly sampled points mapped onto the unit simplex.



(c) Illustration of *Construction* where 10 points are already obtained (blue) and one more additional point need to be added. Three possible locations are marked in red.

Fig. 3. Illustrations of Unit Simplices by using different Methods

will not converge to a well-spaced distribution (corresponding to the optimum of Equation 5) because of the greedy selection at each stage.

C. Reduction-based Methods

Contrary to the construction-based methods, a completely opposite process can also be devised. First, a set of well-spaced boundary points $\mathcal{W}_\mathcal{B}$ of size n_B on the unit simplex is generated and then another well-spaced interior point-set $\mathcal{W}_\mathcal{I}$ of size $n_I = n - n_B$ (where n is the total number of desired points on the unit simplex) is obtained by using k-means algorithm. The final point-set is given by $\mathcal{W}_\mathcal{B} \cup \mathcal{W}_\mathcal{I}$. Our investigations revealed that this two-step procedure of first estimating $\mathcal{W}_\mathcal{B}$ and then $\mathcal{W}_\mathcal{I}$ performs significantly better than optimizing the combined distribution of points $\mathcal{W}_\mathcal{B} \cup \mathcal{W}_\mathcal{I}$ all at once.

In order to obtain $\mathcal{W}_\mathcal{B}$, we use the concept proposed in [23] which is based on the partition number p used in the *Das-Dennis* method. We find the largest p (say, $\bar{p}$) which produces *Das-Dennis* points of size smaller than or equal to n, satisfying the following condition:

$$C_{M-1}^{\bar{p}+M-1} \leq n < C_{M-1}^{\bar{p}+M}. \tag{8}$$

Thus, the number of chosen boundary points n_B ($< n$) is

$$n_B = C_{\bar{p}}^{M+\bar{p}-1} - C_{\bar{p}-M}^{\bar{p}-1}. \tag{9}$$

For instance, for creating a total of $n = 100$ points in a three-objective space ($M = 3$), $\bar{p} = 12$ satisfies the above criterion, because $C_2^{14} = 91 \leq 100 < C_2^{15} = 105$. Therefore, $n_B = C_{12}^{14} - C_9^{11} = 36$ and $n_I = n - n_B = 100 - 36 = 64$.

The creation of $\mathcal{W}_\mathcal{B}$ boundary points is executed using the *Das-Dennis* approach, but the creation of $\mathcal{W}_\mathcal{I}$ interior points is not straightforward and we describe it next. First, the procedure starts with a large sample $\mathcal{S}$ of structured or random points on the unit simplex and we set $\mathcal{W}_\mathcal{I} = \emptyset$ and $\mathcal{W}_\mathcal{B}$ having n_B *Das-Dennis* boundary points. Then, the point $\bar{\mathbf{s}} \in \mathcal{S}$ which is maximally away (in the Euclidean distance sense) from the already selected points in $\mathbf{w} \in \mathcal{W} = \mathcal{W}_\mathcal{B} \cup \mathcal{W}_\mathcal{I}$ is identified using:

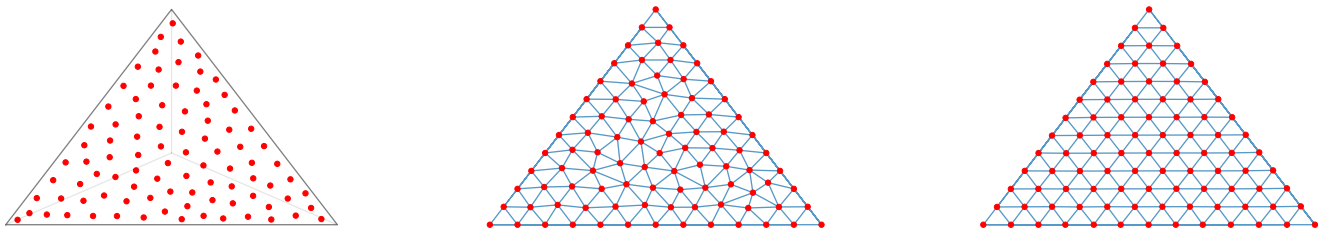
$$\bar{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s} \in \mathcal{S}} \min_{\mathbf{w} \in \mathcal{W}} \left\| z^{(\mathbf{s})} - z^{(\mathbf{w})} \right\|. \tag{10}$$

The point $\bar{\mathbf{s}}$ is then added to $\mathcal{W}_\mathcal{I}$ and is also removed from $\mathcal{S}$ for further processing. The procedure is continued until n_I points are chosen from $\mathcal{S}$. Please note that this initialization of points is performed similarly by Singh et. al [24] to select a posteriori subset of points from an archive obtained by an MOEA. The resulting distribution of points largely depends on the starting sample set $\mathcal{S}$. A large set will produce a good distribution of interior points, but with an increasing size of $\mathcal{S}$ this method also becomes computationally more expensive. However, the points can be improved by executing a few iterations of the well-known k-means algorithm [25]. In an implementation, all n points can be adjusted to find a well-spaced distribution on the unit simplex, but the execution of regular k-means algorithm results into the loss of boundary points, as shown in Figure 4a. This happens because the randomly sampled point-set S does not contain any points outside of the unit simplex and, therefore, cluster centers have no incentive to stay on the boundary and are pushed towards the center (see Figure 4a).

To overcome this effect, we use points in $\mathcal{W}_\mathcal{I}$ only for cluster-assignment but exclude them from the cluster-center-update step of k-means. The k-means algorithm terminates whenever the average movement of all points is below a certain threshold. We have set this threshold to 10^{-4} in our implementation. Figure 4b shows that the resulting distribution of points is nearly uniform. However, inspecting the point-set from Figure 4a more carefully reveals that some minor irregularities exist, and the alignment of points is not as well-spaced as produced by the *Das-Dennis* method. Nevertheless, the *Reduction* procedure allows any arbitrary number of points to be distributed well, but not to the extent desired. The next subsection discusses a method which optimizes the outcome of the *Reduction* procedure to get a better well-spaced distribution (see the Supplementary document for a comparison).

D. Riesz s-Energy Method

The motivation behind the use of an energy concept to obtain a well-spaced distribution comes from nature. Multi-body and interacting physical systems eventually settle on a state which corresponds to the minimum overall potential energy. For two bodies, the potential energy is proportional to the inverse of the distance between them. The minimum poten-



(a) *Reduction* method, when Edges are not fixed.

(b) *Reduction* method with fixed Edges.

(c) *s-Energy* method with $s = 9$.

Fig. 4. Visualization of distribution of 91 points in 2-D unit simplex obtained by different methods.

tial solution corresponds to a diverse distribution of multiple bodies in the three-dimensional physical space. While dealing with a many-dimensional (s) space, we use a generalization of potential energy called *Riesz s-Energy* [26] which is defined between two points ($\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$) as

$$U(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \frac{1}{\left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|^s}. \tag{11}$$

In our context, it is not clear how the dimension s should depend on the number of objectives (M), but some trial-and-error studies (shown in the Supplementary document) motivated us to set $s = M^2$ for all simulations here. For multiple (n) points, the overall s-Energy can be written as follows:

$$U_T(\mathbf{z}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{1}{\left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|^s}, \quad \mathbf{z} \in \mathbb{R}^{n\times M}. \tag{12}$$

The concept for our energy method is to find the $\mathbf{z}$-matrix of size $n \times M$ which minimizes U_T subject to every $\mathbf{z}^{(i)}$ vector to lie on the unit simplex, that is, $\sum_{m=1}^{M} z_m^{(i)} = 1$. We employ a gradient-based optimization method (Adam [27]) here. Due to very large magnitude of U_T, we take a logarithm of U_T and then compute the partial derivative of $F_E = \log U_T$ with respect to $z_m^{(i)}$, as follows:

$$\frac{\partial F_E}{\partial z_m^{(i)}} = -\frac{d}{U_T}\left[\sum_{\substack{j=1,\\ j\neq i}}^{n} \frac{\left(z_m^{(i)} - z_m^{(j)}\right)}{\left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|^{s+2}}\right]. \tag{13}$$

To make sure all points stay on the unit simplex, gradients are projected onto the unit simplex. Figure 4c shows the outcome for $M = 3$ with 91 points.

Riesz s-Energy has been used before to successfully distribute points uniformly on a sphere or other multi-dimensional surfaces [28], [29]. It has also been shown that an inappropriate choice of s can lead to non well-spaced results. However, to the best of our knowledge a concrete implementation of optimizing the objective function and applying the concept to the unit simplex has not been proposed yet. Moreover, it is worth mentioning that *Riesz s-Energy* has recently been utilized as a performance metric for multi-objective optimization in [30] and as a reduction method for choosing a set of reference points from a large set in [29].

The overall outline of the *s-Energy* method is as follows:

1) Generate an initial point-set $\mathbf{z}^{(i)}$ $(i = 1, \ldots, n)$ on the unit simplex using the *Reduction* method.
2) Until maximum number of iterations is reached or a convergence criteria is met:
   a) Calculate gradient with respect to $\mathbf{z}^{(i)}$: $\nabla_{\mathbf{z}^{(i)}} F_E = \left(\frac{\partial F_E}{\partial z_1^{(i)}}, \frac{\partial F_E}{\partial z_2^{(i)}}, \ldots, \frac{\partial F_E}{\partial z_M^{(i)}}\right)$ for all points $i = 1, 2, \ldots, n$.
   b) Project the gradient onto the unit simplex: $\nabla^{proj}_{\mathbf{z}^{(i)}} F_E = \nabla_{\mathbf{z}^{(i)}} F_E - \frac{1}{M}\left(\sum_{m=1}^{M} \frac{\partial F_E}{\partial z_m^{(i)}}\right)\hat{u}$, where $\hat{u}$ is M-dimensional vector of ones. This guarantees $\sum_{m=1}^{M} z_m^{(i)} = 1, \forall i$.
   c) Update each $z_m^{(i)}$ (for n points and for M dimensions) using the gradient-based optimizer *Adam*. Whenever a point $\mathbf{z}^{(i)}$ is outside of the unit simplex, project it back onto the unit simplex to make sure it satisfies $z_m^{(i)} \geq 0, \forall i, m$.

As mentioned before, since the final distribution of points depends on the initial configuration, we use *Reduction* to generate an initial set of points. We define the termination criterion based on the average movement of all points (below 10^{-5}) and perform a restart of *Adam* in case no improvement has been made for 50 iterations. Moreover, the maximum number of iterations is set to 3,000.

IV. Results

In the following, we evaluate the performance of our proposed methods for two different scenarios. In the first scenario, we choose the desired number of points in a manner admissible by the *Das-Dennis* method. We present these results mainly to compare other proposed methods of this paper with the *Das-Dennis* method. In the second scenario, we consider any desired number of reference points. Since the *Das-Dennis* method cannot be applied for such cases, we only compare the proposed methods of this paper with each other.

A. Scenario 1: Das-Dennis Commensurate Points

We compare our proposed methods – *Sampling*, *Construction*, *Reduction*, and *s-Energy* – with the *Das-Dennis* method for a three-objective problem with five different number of reference points (n), as shown in Table I. The obtained points on the unit simplex are compared with five different performance metrics, which we describe next.

TABLE I
EXPERIMENTAL RESULTS FOR 3-DIMENSIONAL UNIT SIMPLICES WITH VARYING NUMBER OF POINTS. *Das-Dennis* PROVIDES THE KNOWN OPTIMUM ($D^{\min,*} = \sqrt{2}/p$), THE BEST PERFORMING METHOD (EXCEPT *Das-Dennis*) IS SHOWN IN BOLD. RIESZ S-ENERGY METHOD PRODUCES CLOSEST $D^{\min,*}$ AND NEAR-ZERO IGD, TRI, AND VGM COMPARED TO OTHER METHODS.

		21 ($p=5$)	%	**36** ($p=7$)	%	**91** ($p=12$)	%	**153** ($p=16$)	%	**325** ($p=24$)	%
HV	Sampling	1.3980450	100.00	1.2063700	100.00	1.0323220	100.00	0.9788900	100.00	0.9311150	100.00
	Construction	1.4440990	7.81	1.2372800	26.41	1.0578780	6.28	0.9982060	1.76	0.9419730	10.80
	Reduction	1.4479780	0.04	1.2483510	0.05	**1.0596600**	**-0.25**	**0.9985990**	**-0.24**	0.9432230	0.53
	Riesz s-Energy	**1.4480000**	**0.00**	**1.2483730**	**0.00**	1.0595890	0.01	0.9985500	0.01	**0.9432740**	**0.11**
	Das-Dennis	1.4480000	0.00	1.2483730	0.00	1.0595910	0.00	0.9985520	0.00	0.9432870	0.00
$D^{\min}$	Sampling	5.230e-02	100.00	2.578e-02	100.00	9.831e-03	100.00	5.611e-03	100.00	2.683e-03	100.00
	Construction	1.540e-01	55.90	1.500e-01	29.55	8.207e-02	33.13	4.721e-02	49.74	3.959e-02	34.37
	Reduction	2.680e-01	6.45	1.829e-01	10.87	8.556e-02	29.90	5.923e-02	35.23	3.519e-02	42.20
	Riesz s-Energy	**2.819e-01**	**0.43**	**2.004e-01**	**0.91**	**1.142e-01**	**3.42**	**8.467e-02**	**4.49**	**5.204e-02**	**12.24**
	Das-Dennis	2.828e-01	0.00	2.020e-01	0.00	1.179e-01	0.00	8.839e-02	0.00	5.893e-02	0.00
IGD	Sampling	1.053e-01	100.00	8.423e-02	100.00	5.301e-02	100.00	4.018e-02	100.00	2.732e-02	100.00
	Construction	8.979e-02	85.23	7.287e-02	86.51	4.169e-02	78.64	3.089e-02	76.87	2.139e-02	78.31
	Reduction	2.836e-03	2.69	5.174e-03	6.14	1.785e-02	33.67	1.731e-02	43.08	1.505e-02	55.10
	Riesz s-Energy	**2.673e-04**	**0.25**	**5.092e-04**	**0.60**	**1.693e-03**	**3.19**	**2.178e-03**	**5.42**	**2.486e-03**	**9.10**
	Das-Dennis	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00
Tri	Sampling	3.019e-01	100.00	2.134e-01	100.00	1.203e-01	100.00	8.716e-02	100.00	5.459e-02	100.00
	Construction	8.948e-02	29.64	8.834e-02	41.40	4.884e-02	40.59	2.126e-02	24.40	1.908e-02	34.96
	Reduction	9.390e-03	3.11	1.287e-02	6.03	2.544e-02	21.15	2.278e-02	26.13	1.712e-02	31.37
	Riesz s-Energy	**4.104e-04**	**0.14**	**5.637e-04**	**0.26**	**1.260e-03**	**1.05**	**1.229e-03**	**1.41**	**1.941e-03**	**3.56**
	Das-Dennis	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00
VGM	Sampling	5.853e-03	100.00	2.777e-03	100.00	8.847e-04	100.00	4.578e-04	100.00	1.978e-04	100.00
	Construction	2.905e-03	49.63	7.980e-04	28.74	2.931e-04	33.13	1.913e-04	41.79	1.413e-05	7.15
	Reduction	1.374e-05	0.23	2.102e-05	0.76	5.280e-05	5.97	3.583e-05	7.83	2.183e-05	11.04
	Riesz s-Energy	**7.484e-08**	**0.00**	**1.501e-07**	**0.01**	**5.906e-07**	**0.07**	**4.343e-07**	**0.09**	**1.076e-06**	**0.54**
	Das-Dennis	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00	0.000e+00	0.00

- **Hypervolume (HV):** The hypervolume measure [31] is frequently used for comparing results of multi and many-objective optimization algorithms. It measures the area or volume of the collective dominated region with the help of a reference point **r** which is pre-specified by the user. A suitable reference point $\mathbf{r} = (r, r, \ldots, r)^T$ is essential to provide equal importance to all points in the point-set. We use a suggested value [23]:

$$r = 1 + 1/p, \tag{14}$$

  where p is the number of gaps used by *Das-Dennis* method. The larger the HV, the better is the performance of a method, but we also note that an improper choice of **r** may not cause the *Das-Dennis* method to produce the largest HV.
- **Inverted Generational Distance (IGD):** The IGD indicator [32] is another popular performance metric used in EMaO studies. It measures the average Euclidean distance (in the objective space) for each supplied Pareto-optimal solution to the corresponding closest solution in the provided solution set. Thus, this metric can only be used in cases where the location of the entire Pareto-optimal front is known. Moreover, since the supplied Pareto-set must be a well-spaced set on the unit simplex, we are only able to use IGD for *Das-Dennis* commensurate scenarios.
- **Distance of Closest Pair of Points ($D^{\min}$):** For a perfectly well-spaced point-set, the distance ($D^{\min}$) between the closest pair of points (described in Equation 5) is maximum compared to any other point-set. This metric ignores the distribution of all points except a single isolated pair of points which has the shortest distance along the unit simplex. However, for *Das-Dennis* commensurate n points with a specific p, this is an appropriate metric and has an upper bound on the objective value of $\sqrt{2}/p$.
- **Triangulation (Tri):** This is another metric which can be used to gauge deviation of the obtained distribution from the optimal *Das-Dennis* point-set. The obtained point-set is first triangulated and then the greatest difference between smallest and largest sides of triangles is used as a measure of uniformity of the triangles. Here, we use *Delauney-Triangulation* [33]. A point-set obtained by *Das-Dennis* will make every triangle equilateral, similar to the plot in Figure 4c, thereby making the *Tri* measure equal to zero. Thus, the smaller the *Tri* measure, the better is the performance of the method. It is also clear that this metric is governed by a few neighboring points and it fails to capture the uniformity measure of the entire point-set. Since triangulation method is computationally expensive for more than three dimensions, this metric will be difficult to extend for many-objective problems.

- **Variance of Geometric Mean of k-Nearest Neighbors (VGM):** To alleviate the singularity associated with the $D^{\min}$ metric, we propose to measure the variance of geometric means (VGM). For an M-objective problem, we compute Euclidean distances from k-nearest neighboring points ($k = M - 1$) for each obtained point. We then compute the *geometric-mean* of the k distances. The geometric mean, in this case, is a measure of a representative size of the hypervolume of the local simplex formed by a point touching its neighbors. For a well-spaced set of points (such as *Das-Dennis* method), the variance of the neighborhood size (geometric mean of the hypervolume) of each point will be zero. For other distributions, a smaller value of the variance may indicate a more well-spaced distribution. Interestingly, this metric can be extended for any dimension. However, a smaller variance does not ensure a better spread of points on the entire unit simplex. Thus, we propose to use this metric (a small value) along with $D^{\min}$ metric (with a large value) to determine the superiority of a particular point creation method proposed in this paper.

In Table I, the performance measures on a three-objective unit simplex are shown for 21, 36, 91, 153, and 325 points. Corresponding values of p to obtain the above set of points using *Das-Dennis* method are 5, 7, 12, 16, and 24, respectively. We report the median values of each performance metric over 101 runs, followed by its normalized value in [0, 100]% scale, where 0% corresponds to the *Das-Dennis* and 100% corresponds to the worst performing point creation method. The best performing method (excluding *Das-Dennis* method, due to its perceived superiority for chosen commensurate points) is marked in bold. It is observed that *s-Energy* method outperforms the other methods in almost all cases. Also, from the values of relative performance, it can be observed that *s-Energy* produces a very similar metric value as that of *Das-Dennis* method. Except the HV values, *s-Energy* and *Das-Dennis* methods are visibly better, hence we do not perform any statistical significant test to establish the superiority of these two methods.

For 91 and 153 points, the *Reduction* method obtains slightly better HV values. A negative relative performance implies that the *Reduction* method is able to outperform even the *Das-Dennis* method. A similar behavior was recently observed in [34] where the authors have shown that in many cases uniformly-distributed objective vectors are not optimal regarding multi-objective performance indicators (including HV). This implies that HV does not necessarily reflect the uniformity of a point-set and, therefore, might not be a reliable performance indicator for the purpose of this paper.

The minimum distance $D^{\min}$ between the closest point pair is a fairly strict metric because one point, not being appropriately adjusted, can have a big influence on the resulting value. However, *s-Energy* is able to successfully obtain distributions with larger values of $D^{\min}$, whereas the other approaches do not reflect well on this strict metric. Increasing the number of points increases the overall density of the distribution, thereby resulting into smaller $D^{\min}$ values. A similar observation is made considering the *Tri* metric.

Small values of *VGM*, *Tri*, *IGD* and large values of *HV* demonstrated by the *s-Energy* method with the metric values close to that of *Das-Dennis* method makes it a suitable reference point creation method for three-objective *Das-Dennis* commensurate points. Next, we investigate how *s-Energy* method performs for arbitrary number of desired points for three as well as for larger dimensional spaces.

B. Scenario 2: Arbitrary Number of Points

Here, we consider three to 15-objective cases involving arbitrary number of desired points on the unit simplex, as shown in Table II. For this scenario, we cannot apply the *Das-Dennis* method, as no integer p value will correspond to an arbitrary number of desired points for a specific M-dimensional unit simplex. Thus, we restrict our comparison to four point-creation methods – *Sampling*, *Construction*, *Reduction*, and *s-Energy*. Also, since an equi-spaced point-set may not be available for an arbitrary n (in fact, finding a well-spaced set of points on the unit simplex is the goal of this study), we cannot use the *IGD* metric. The weakness of the *HV* metric demonstrated in Table I prohibits us to use it as well for the many-objective problems. Also, the computational complexity involved with the *Tri* metric deters us from using it here. Thus, we use $D^{\min}$ and *VGM* metrics to compare the four point-creation methods. Their combination (large $D^{\min}$ and small *VGM*) will ensure that points are spaced with a large distance from each other and for each point, its k-closest neighbors are almost equi-spaced.

Table II shows the results regarding $D^{\min}$. The best performing method is marked in bold. The first column for each method marks the median $D^{\min}$ value of 101 runs and the second column indicates its relative performance among all four methods – 0% is marked for the best and 100% for the Sampling method. Other methods are assigned a relative percentage value based on their normalized $D^{\min}$ values. The results clearly indicate that *s-Energy* is able to outperform other creation methods in all cases by producing the largest median $D^{\min}$ value. The reduction of the *Riesz s-Energy* function with iteration of *s-Energy* method for $M = 3$ with 100 points is shown in the Supplementary document.

Table II further shows that the *Reduction* method is not able to perform well, particularly for higher dimensional spaces. For instance, for the 15-dimensional case, its relative performance (about 93% to 100%) is almost similar to the *Sampling* method (100%), while for three and five-dimensional cases, its relative performance is below 50%, indicating a reasonable performance. This can be attributed to the fact that we have used a fixed number (10,000) of sampled points on the unit simplex to start the *Reduction* method. However, for higher dimensional spaces, a larger number of points may be required to uniformly fill the unit simplex. This is not computationally practical due to memory and running time limitations.

However, the point-set obtained by the *Reduction* method can serve as an initial starting set for the *s-Energy* method. Results indicate that *s-Energy* is able to morph a point-set to form a well-spaced distribution eventually. Even though

TABLE II
COMPARISON OF FOUR METHODS USING THE CLOSEST DISTANCE PAIR OF POINTS METRIC (D^{MIN}) FOR THREE TO 15-DIMENSIONAL UNIT SIMPLICES.

		D^{min}							
M	**n**	**Sampling**		**Construction**		**Reduction**		**Riesz s-energy**	
3	50	1.753e-02	100.00	1.483e-01	0.21	1.247e-01	18.20	**1.485e-01**	**0.00**
	100	9.340e-03	100.00	8.115e-02	22.69	7.866e-02	25.37	**1.022e-01**	**0.00**
	250	3.457e-03	100.00	4.132e-02	36.22	4.193e-02	35.18	**6.281e-02**	**0.00**
	500	1.673e-03	100.00	2.077e-02	52.74	2.562e-02	40.74	**4.208e-02**	**0.00**
5	100	4.417e-02	100.00	2.050e-01	29.59	1.736e-01	43.34	**2.726e-01**	**0.00**
	250	2.628e-02	100.00	1.487e-01	27.26	1.333e-01	36.42	**1.946e-01**	**0.00**
	500	1.927e-02	100.00	1.187e-01	27.45	1.089e-01	34.56	**1.563e-01**	**0.00**
8	200	6.758e-02	100.00	2.632e-01	29.56	1.461e-01	71.73	**3.453e-01**	**0.00**
	500	5.293e-02	100.00	2.078e-01	30.86	1.243e-01	68.16	**2.770e-01**	**0.00**
	1000	4.308e-02	100.00	1.803e-01	27.76	1.134e-01	62.97	**2.330e-01**	**0.00**
10	300	7.721e-02	100.00	2.795e-01	28.03	1.303e-01	81.12	**3.582e-01**	**0.00**
	600	6.571e-02	100.00	2.402e-01	33.92	1.097e-01	83.33	**3.297e-01**	**0.00**
	1000	5.874e-02	100.00	2.172e-01	28.82	1.121e-01	76.04	**2.813e-01**	**0.00**
15	300	1.011e-01	100.00	3.628e-01	26.46	1.013e-01	99.94	**4.570e-01**	**0.00**
	600	8.792e-02	100.00	3.115e-01	29.27	1.007e-01	95.95	**4.040e-01**	**0.00**
	1000	8.188e-02	100.00	2.827e-01	27.47	1.015e-01	92.92	**3.588e-01**	**0.00**

Construction is able to scale significantly better than *Reduction*, it is not able to match the performance of the *s-Energy* method. Moreover, it is worth mentioning that with increasing number of dimensions and points, the overall running time of *Construction* increases significantly, since it demands a dedicated optimization-run for obtaining each single point, one at a time.

Table III shows the performance of all four methods using the *VGM* metric. Results indicate that *s-Energy* is able to obtain significantly smaller *VGM* values compared to other approaches. Also, similar to the trend observed for D^{min}, *Reduction* is not able to perform well with increasing number of dimensions – for eight-objective and above, it performs significantly worse than even the *Sampling* method. *Construction* performs the second best, after *s-Energy* method. The combined results of both performance metrics, D^{min} and *VGM*, clearly validate the suitability of our proposed *s-Energy* method to effectively determine well-spaced point-set on the unit simplex on three to 15-objective problems. Further comparative results are presented in the Supplementary document.

In addition to the performance comparisons made above, visualization of points can provide further validation of the uniformity of obtained points by the *s-Energy* method. It is worth noting here that the *s-Energy* point-sets can have an *arbitrary* size, which has been the primary motivation of this study.

1) First, we present results from *Das-Dennis* commensurate case with $p = 12$ for a three-objective unit simplex, requiring 91 points. Figure 5 shows the distribution of 91 points on the unit simplex in circles obtained using the *s-Energy* method. Next, we create a total of 92 points afresh, a case which is not possible to achieve by the *Das-Dennis* method. The points are marked using a cross. It is interesting to observe from the figure that only a few points towards the lower right corner of the simplex get readjusted to accommodate the extra (92nd) point. The choice of adjustment at the lower right corner is arbitrary and probably is an outcome of the starting distribution of points (see the Supplementary document for a slightly different distribution obtained in another run of the *s-Energy* method). The major portion of the overall distribution remains nearly similar to the configuration obtained for $n = 91$ points. Visually, the resulting point-set with 92 points is uniform, even though it is known to the best of our knowledge that no perfectly well-spaced distribution of points with 92 or any arbitrary number of points exists. This study demonstrates the effectiveness of the proposed *s-Energy* method.

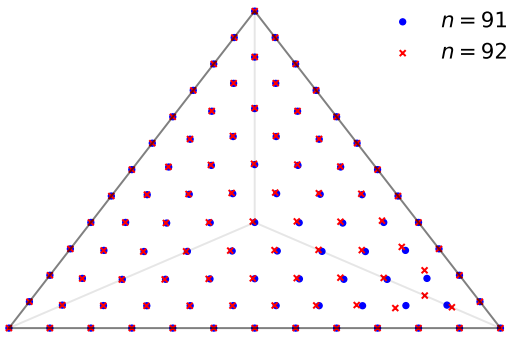


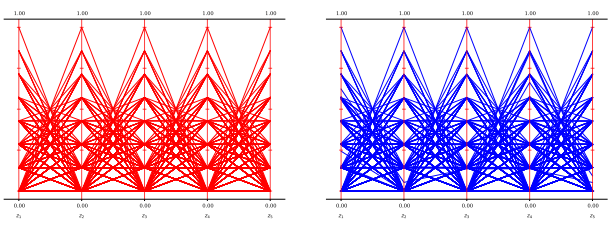
Fig. 5. Distribution of points obtained by *s-Energy* with 91 (circle) and 92 (cross) points. Notice how an additional point placed on the bottom-right corner adjusts neighboring points to make a good distribution.

2) Next, we visually compare the distribution of points obtained by *s-Energy* for more than three dimensions. We present the points using the parallel coordinate plot (PCP) [35], which is one of the common visualization techniques in many-objective optimization. All axes are plotted vertically and represent different objectives. Each

TABLE III
COMPARISON OF FOUR METHODS USING *VGM* METRIC FOR THREE TO 15-DIMENSIONAL UNIT SIMPLICES.

		VGM							
M	**n**	**Sampling**		**Construction**		**Reduction**		**Riesz s-energy**	
3	50	1.855e-03	100.00	6.130e-05	0.85	1.892e-04	7.92	**4.583e-05**	**0.00**
	100	7.899e-04	100.00	1.835e-04	22.33	7.904e-05	8.95	**9.130e-06**	**0.00**
	250	2.692e-04	100.00	6.638e-05	24.40	3.075e-05	11.12	**9.174e-07**	**0.00**
	500	1.232e-04	100.00	5.475e-05	44.09	1.603e-05	12.46	**7.866e-07**	**0.00**
5	100	4.668e-03	100.00	1.121e-03	23.57	2.227e-03	47.40	**2.695e-05**	**0.00**
	250	1.687e-03	100.00	3.301e-04	18.55	6.422e-04	37.29	**2.099e-05**	**0.00**
	500	8.133e-04	100.00	1.676e-04	19.02	8.579e-05	8.76	**1.594e-05**	**0.00**
8	200	7.512e-03	100.00	2.813e-03	36.98	9.485e-03	126.45	**5.498e-05**	**0.00**
	500	2.758e-03	100.00	8.453e-04	30.31	3.922e-03	142.41	**1.368e-05**	**0.00**
	1000	1.352e-03	100.00	4.327e-04	31.61	1.457e-03	107.87	**7.838e-06**	**0.00**
10	300	7.924e-03	100.00	3.305e-03	41.02	9.759e-03	123.44	**9.272e-05**	**0.00**
	600	3.747e-03	100.00	1.516e-03	40.14	9.803e-03	262.45	**1.996e-05**	**0.00**
	1000	2.223e-03	100.00	8.517e-04	37.95	4.799e-03	216.58	**1.298e-05**	**0.00**
15	300	1.714e-02	100.00	4.445e-03	23.36	3.921e-02	233.17	**5.751e-04**	**0.00**
	600	8.323e-03	100.00	3.120e-03	36.62	2.248e-02	272.45	**1.145e-04**	**0.00**
	1000	4.887e-03	100.00	2.290e-03	45.76	1.577e-02	327.16	**9.879e-05**	**0.00**

solution is illustrated by a line marking the objective value on each vertical axis. The crisscrossing of lines between two consecutive axes indicates the existence of trade-offs between objective values. Figure 6 shows two PCP plots comparing point-sets obtained using *Das-Dennis* and *s-Energy* in a five-dimensional space. *Das-Dennis* is initialized with a partition number $p = 7$ which results in 330 points (see Figure 6a). Seven gaps on each axis is clear from the plot. *s-Energy* also produces an identical distribution, but is not shown here for brevity. Instead, we reduce the number of points to 320, apply *s-Energy* method (as *Das-Dennis* method is no more applicable), and present the results in Figure 6b. Again, seven gaps on each axis is clearly visible except some minor adjustments in the first and fourth objective values. But apparent crisscrossing patterns are very similar between the two plots, despite the right plot having 10 points less. This study demonstrates the efficacy of our proposed *s-Energy* method to generate a well-spaced point-set even in higher dimensions.



(a) *Das-Dennis Sampling* with $p = 7$ resulting in 330 points. (b) *s-Energy* generating 320 points.

Fig. 6. Well-spaced point-set on the unit simplex in five dimensions.

To make well-spaced point-sets directly available for the EMO community, we provide pre-generated results for up to 15 dimensions with some standard size of points online[1]. A source code of the *s-Energy* method is available in the multi-objective optimization framework pymoo [36].

V. PRACTICAL MATTERS

When no preference information is available, creating a well-spaced set of points is a recommended approach. However, with some known preference information, biased distributions are desired. We describe here two biased approaches based on our *s-Energy* method, which can be of more practical value.

A. Points in Region of Interest (ROI)

Often, in an optimization scenario, some parts of the objective space are of more interest than other parts. This implies that in addition to creating an overall well-spaced distribution, some preferred regions must be represented with more points. For such scenarios, *s-Energy* method is suitable because it can be reapplied separately on such regions to create a local distribution with arbitrary number of points.

For this purpose, we restrict the shape of ROI as a down-scaled version of the original unit simplex. The size of i-th such preferred region can be defined using a *scaling factor* S_i. To make this even more customized, a user can supply the number of points N_i to be added to the i-th ROI, in addition to the initial set of n points to be distributed uniformly on the entire unit simplex, thereby making the total number of points equal to $n_T = n + \sum_{i=1}^{K} N_i$, where K is the number of ROIs. Figure 7 illustrates a case with two user-specified ROIs with two reference points R_i on a three-dimensional unit simplex: (i) $R_1 = (0.15, 0.70, 0.15)$ with $S_1 = 0.2$ and $N_1 = 12$, and (ii) $R_2 = (0.33, 0.33, 0.33)$ with $S_2 = 0.35$ and $N_2 = 20$. The initial set contains $n = 90$ points. The *s-Energy* method

[1]https://www.egr.msu.edu/coinlab/blankjul/uniform/

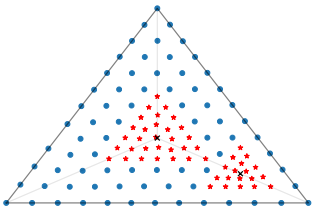


Fig. 7. Distribution of 112 points with two focused sets using the *s-Energy* method.

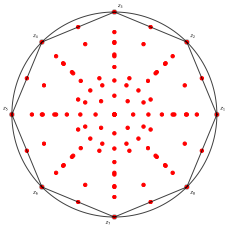


Fig. 8. Semi-structured approach with 112 points in eight dimensions on a RadViz plot.

is modified as follows. First, n points are located using the *s-Energy* method. Let us call this set $\mathcal{S}$. Thereafter, for i-th ROI, we isolate the points from $\mathcal{S}$ in its domain (centered around R_i with shrinkage factor S_i). Let us say there are n_i points from original n points in its domain. If additional N_i points are desired in i-th ROI, we apply the *s-Energy* method in the ROI to find the location of $(n_i + N_i)$ points, but the energy is now computed from all points and by fixing the location of points outside the ROI. The above procedure is repeated for each ROI one at a time to obtain the final distribution of n_T points. Another case is shown in the Supplementary document.

B. Semi-Structured Point-set

Another way to ensure a sufficient number of intermediate points is to use a layer-wise *Das-Dennis* approach described in Subsection II-D. However, instead of a user specifying the shrinkage factor for each layer, the *s-Energy* method can be used to find the optimal shrinkage factors so that Riesz s-Energy of all n points is maximized. Thus, the user has to only supply information about the gap number p of each layer in an ordered array $\mathbf{P}$, wherein the first element represents gap number of the outermost layer and the last element of the inner-most layer. If the user desires to have around 100 points in an 8-objective problem, then p must fall into the range between 0 and 3. Thus, $\mathbf{P} = \{2, 2, 1, 1, 1, 1\}$ is one of the feasible settings which will render a total of 112 points $(2 \times 36 + 5 \times 8)$ distributed across seven layers. It is recommended to choose a larger p value for outer layers than for the inner layers. The *scaling factors* for layers are then determined by optimizing the energy function. The points obtained by the method are shown in Figure 8 in a RadViz plot. It is visually evident that the resulting point-set is *structured*, however the points are not well-spaced. For the layer-wise approach, D^{min} and *VGM* metric values are 9.9×10^{-2} and 4.5×10^{-3}, respectively, whereas the *s-Energy* method produces much improved values: 4.1×10^{-1} and 1.6×10^{-4} (a RadViz plot is presented in the Supplementary document).

VI. Conclusions

In this paper, we have addressed an important issue related to decomposition-based evolutionary many-objective optimization (EMaO) algorithms: creation of arbitrary number of well-spaced reference points for any M dimensional objective space. After defining near-uniformity of a set of points through the outcome of an optimization problem, we have discussed fundamentally different approaches for creating such a set for three to 15-dimensional problems: a baseline approach (*Sampling*), where points are created in a space-filling manner and mapped onto the unit simplex; a bottom-up approach (*Construction*) where a point is incrementally added to an existing point-set; a top-down approach (*Reduction*) where a large number of points is reduced to a desired number; and a point-set refinement approach (*s-Energy*) that incrementally improves a given point-set with respect to the Riesz s-Energy function. We have compared the results with the existing state-of-the-art method *Das-Dennis*. Results indicate that the *s-Energy* method is able to efficiently find a well-spaced distribution on the unit simplex, based on a number of performance metrics suggested in this study. Moreover, we have provided a scale up study to observe the behavior of our methods to higher dimensional spaces for up to 15 objectives. Across all experiments with different number of points and dimensions, *s-Energy* method has outperformed other methods. Additionally, we have addressed two practical matters, such as obtaining localized well-spaced distribution in the regions of interest and the generation of a semi-structured point-set through a layer-wise approach, each optimized using the *s-Energy* method.

This study fills a long-standing gap in the study of EMaO algorithms. While so far researchers have concentrated in developing efficient algorithms by using structured reference directions, this study allows the algorithms to choose an arbitrary number of well-spaced reference directions for any dimensions with and without any preference biases. Thus, it can be used to generate a set of arbitrary number of reference points for computing performance indicators, such as IGD or IGD+. Further studies now should be spent on extending the *s-Energy* method to other preference-biasing approaches and to reallocate directions adaptively and uniformly to parts of the unit simplex.

References

[1] 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.

[2] K. Deb and H. Jain, "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*, vol. 18, no. 4, pp. 577–601, 2014.

[3] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. on Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

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

[5] K. Deb, S. Bandaru, and H. Seada, "Generating uniformly distributed points on a unit simplex for evolutionary many-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Cham: Springer International Publishing, 2019, pp. 179–190.

[6] R. A. Fisher, "The arrangement of field experiments," *Journal of the Ministry of Agriculture of Great Britain*, vol. 33, pp. 503–513, 1926.

[7] G. E. P. Box and K. B. Wilson, "On the experimental attainment of optimum conditions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 1, pp. 1–45, 1951.

[8] V. C. Chen, K.-L. Tsui, R. R. Barton, and J. K. Allen, "A review of design and modeling in computer experiments," *Handbook of Statistics*, vol. 22, pp. 231–261, 2003.

[9] C. R. Rao, "Factorial experiments derivable from combinatorial arrangements of arrays," *Supplement to the Journal of the Royal Statistical Society*, vol. 9, no. 1, pp. 128–139, 1947.

[10] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[11] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, no. 1, pp. 84–90, 1960.

[12] J. M. Hammersley, "Monte carlo methods for solving multivariable problems," *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960.

[13] I. M. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[14] H. Faure, "Discrépance de suites associées à un système de numération (en dimension s)," *Acta Arithmetica*, vol. 41, no. 4, pp. 337–351, 1982.

[15] Y. Wang and K. T. Fang, "On number-theoretic method in statistics simulation," *Science China Math.*, vol. 53, no. 1, pp. 179–186, 2010.

[16] N. A. Smith and R. W. Tromble, "Sampling uniformly from the unit simplex," 2004, Johns Hopkins University.

[17] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[18] R. Schinzinger and P. A. A. Laura, *Conformal Mapping: Methods and Applications*. Dover Books on Mathematics, 2003.

[19] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods, Wiley Series in Prob. and Stats.* New York: Wiley, 2011.

[20] H. Scheffé, "Experiments with mixtures," *Journal of the Royal Statistica; Societies, Series B*, vol. 20, no. 2, pp. 344–360, 1958.

[21] E. J. Hughes, "Multiple single objective pareto sampling," in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, vol. 4, 2003, pp. 2678–2684 Vol.4.

[22] P. Prescott, "Nearly uniform designs for mixture experiments," *Communications in Statistics—Theory and Methods*, vol. 37, no. 13, pp. 2095–2115, 2008.

[23] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evol. Comput.*, vol. 26, no. 3, pp. 411–440, Sep. 2018.

[24] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance-based subset selection for benchmarking in evol. multi/many-objective optimization," *IEEE Trans. on Evol. Comput.*, vol. 23, no. 5, pp. 904–912, 2019.

[25] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *JSTOR: Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[26] D. Hardin and E. Saff, "Minimal Riesz energy point configurations for rectifiable d-dimensional manifolds," *Advances in Mathematics*, vol. 193, no. 1, pp. 174 – 204, 2005.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, Y. Bengio and Y. LeCun, Eds., 2015.

[28] D. P. Hardin and E. B. Saff, "Discretizing manifolds via minimum energy points," in *Notices of the AMS*, 2004, pp. 1186–1194.

[29] J. G. Falcón-Cardona, H. Ishibuchi, and C. A. C. Coello, "Riesz s-energy-based reference sets for multi-objective optimization," 2020, Research Gate.

[30] J. G. Falcón-Cardona, C. A. Coello Coello, and M. Emmerich, "Criemoa: A Pareto-front shape invariant evolutionary multi-objective algorithm," in *Evolutionary Multi-Criterion Optimization (EMO-2019 Conference Proceedings)*. Springer, 2019, pp. 307–318.

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

[32] D. A. V. Veldhuizen and D. A. V. Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Evolutionary Computation, Tech. Rep., 1999.

[33] J. A. De Loera, J. Rambau, and F. Santos, *Triangulations: Structures for Algorithms and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[34] R. Tanabe and H. Ishibuchi, "An analysis of quality indicators using approximated optimal distributions in a three-dimensional objective space," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2020.

[35] E. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Journal of the Am. Stats. Asso.*, vol. 85, pp. 664–675, 09 1990.

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

Julian Blank is a PhD student in the Department of Computer Science and Engineering at Michigan State University. He received his B.Sc. in Business Information Systems from Otto von Guericke University, Germany in 2010. He was a visiting scholar for six months at the Michigan State University, Michigan, USA in 2015, and, finished his M.Sc. in Computer Science at Otto von Guericke University, Germany in 2016. He is the main developer of pymoo, an open source multi-objective optimization framework in Python. His research interests include evolutionary computation, multi-objective optimization, surrogate-assisted optimization and machine learning.

Kalyanmoy Deb is Fellow, IEEE and the Koenig Endowed Chair Professor with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, Michigan, USA. He received his Bachelor's degree in Mechanical Engineering from IIT Kharagpur in India, and his Master's and Ph.D. degrees from the University of Alabama, Tuscaloosa, USA, in 1989 and 1991. He is largely known for his seminal research in evolutionary multi-criterion optimization. He has published over 535 international journal and conference research papers to date. His current research interests include evolutionary optimization and its application in design, modeling, AI, and machine learning. He is one of the top cited EC researchers with more than 140,000 Google Scholar citations.

Yasheh Dhebar is a PhD student in the Department of Mechanical Engineering at Michigan State University. He did his bachelors and masters in Mechanical Engineering from the Indian Institute of Technology Kanpur and graduated in 2015 with his thesis on modular robotics and their operation in cluttered environments. Currently he is actively pursuing his research in the field of explainable and interpretable artificial intelligence, machine learning and knowledge discovery. He has worked on applied optimization and machine learning projects with industry collaborators under the guidance of Prof. Kalyanmoy Deb.

Sunith Bandaru is Sr. Member, IEEE and an Associate Professor with the Department of Production and Automation Engineering in the School of Engineering Science and Chair of the Committee for Research Education in Informatics at University of Skövde, Sweden. He received his Bachelor's degree (2006) in Mechanical Engineering from Jawaharlal Nehru Technological University, India, and Master's (2008) and Ph.D. (2013) degrees from IIT Kanpur, India. His research interests lie at the intersection of evolutionary computation, machine learning and data mining, where he has been developing methods for knowledge discovery from simulation-based multi-objective engineering optimization problems.

Haitham Seada is a data scientist at AltairPD/Ford Motor Company. He is a Computer Science major with a master's degree from Zagazig University, Egypt, and a PhD from Michigan State university (2017). Dr. Seada worked briefly as a postdoctoral research associate at Brown University before joining Altair/Ford. His current role at Ford involves creating optimization models and algorithms for vehicle routing problems and building software wrappers to the developed algorithms. His research interests involve vehicle routing, linear/non-linear optimization, computational intelligence and machine learning. Dr. Seada's teaching experience spans for over 8 years. In addition, he is also the author of several software libraries.