Projected Hypercube Sampling with Parallel Reference Vectors for Uniform Coverage of Irregular Pareto-Optimal Fronts

Kannan Sekar
The University of New South Wales
Canberra, Australia
k.sekar@unsw.edu.au

Ankush Kapoor
The University of New South Wales
Canberra, Australia
ankush.kapoor@unsw.edu.au

Hemant Kumar Singh
The University of New South Wales
Canberra, Australia
h.singh@unsw.edu.au

Tapabrata Ray
The University of New South Wales
Canberra, Australia
t.ray@unsw.edu.au

***Abstract*—Evolutionary algorithms commonly utilize search along reference vectors to solve multi/many-objective optimization problems. However, it is now well known that their effectiveness strongly depends on the geometric compatibility between reference vectors and the geometry of the Pareto front (PF). Conventional simplex-based systematic sampling (SS) implicitly assumes simplex-aligned PFs, which may lead to missing intersections and poor coverage when the front is highly curved, inverted or disconnected. Adaptation of reference vectors during the search is one potential approach explored in the literature to counter this. This paper investigates an alternate approach, referred to as Projected Hypercube Sampling (PHS). It is a reference vector generation strategy intended to work across a wider range of PF shapes. It uniformly samples in a unit hypercube and projects the samples onto a reference hyperplane, producing a set of well-spaced parallel reference vectors that span the full PF-relevant region. PHS is evaluated on analytically defined convex and non-convex PFs with controllable curvature across up to ten objectives. Numerical experiments show that PHS consistently achieves better coverage and inverted generational distance compared to SS over a range of front geometries. The performance of PHS also improves monotonically with increasing reference-point density, unlike SS where the performance saturates if the shape is not aligned to the simplex. These observations demonstrate that PHS provides a reliable and scalable reference vector generation mechanism to deal with many-objective optimization problems.**

***Index Terms*—Multi-objective optimization, Many-objective optimization, reference vectors, systematic sampling, projected hypercube sampling**

I. Introduction and Background

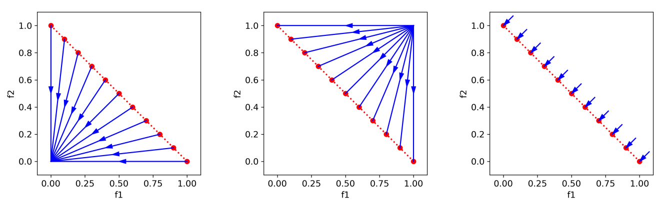
Many real-world optimization problems require the simultaneous optimization of multiple conflicting objectives. These problems are typically formulated as multi-objective optimization problems (MOPs), in which no single solution can optimize all objectives at once. Instead, the aim is to find the Pareto-optimal set (PS), whose image yields a trade-off manifold called the Pareto-optimal front (PF) [1] in the objective space. An improvement in any objective for a solution on the PF leads to degradation in at least one other objective.

Evolutionary algorithms (EAs) have been widely adopted for solving MOPs due to their population-based nature, which allows identification of multiple trade-off solutions in a single run. The majority of early evolutionary multi-objective optimization (EMO) algorithms rely on Pareto dominance as the fundamental selection criterion, e.g. NSGA-II [1], SPEA2 [2], and PAES [3]. These methods combine non-dominance based ranking combined with diversity preservation mechanisms to approximate the PF. Indicator-based approaches, such as SMS-EMOA [4] and HypE [5], employ an alternate strategy, wherein a set-based measure such as hypervolume (HV) is utilized for environmental selection.

While dominance-based approaches work well for two or three objectives, their effectiveness is shown to reduce in many cases for higher number of objectives. The proportion of mutually non-dominated solutions grows rapidly, reducing selection pressure and limiting the effectiveness of Pareto dominance as a guiding mechanism [6], [7]. Owing to this challenge, problems with four or more objectives are often referred to as many-objective optimization problems (MaOPs) in the EMO community.

To address MaOPs, decomposition-based methods [8] using reference vectors have been widely studied. These approaches decompose a MaOP into subproblems typically defined using reference vectors representing different search directions. A representative algorithm is MOEA/D [9], which uses scalarizing functions such as weighted Tchebycheff or penalty-based boundary intersection (PBI) to balance convergence and diversity. Earlier methods like Normal Boundary Intersection (NBI) [10] provide the theoretical foundation through systematic directional sampling of Pareto-optimal solutions. NSGA-III [11] integrates reference points with non-dominated sorting, while RVEA [12] employs reference vectors and angle-based selection to maintain diversity in high-dimensional spaces. The algorithms utilize geometric relationships between solutions and reference vectors, through, e.g. parallel and perpendicular components, or angles, to guide the population towards a well-distributed PF approximation.

Several mechanisms have been proposed for defining the direction of the reference vector for many-objective optimization, including pull-based, push-based, and parallel strategies. Considering a minimization problem, pull-based methods have the reference vectors originating from (or converging to) the ideal point and attempt to attract the solutions towards it (Fig. 1a). This strategy performs well on linear simplex, non-convex or orthant type PFs such as DTLZ2 [9]. In contrast, push-based methods(Fig. 1b) originate from the nadir point and repel solutions away from it, making them more suitable for convex and inverted orthant type fronts [13]–[15]. Parallel reference vector mechanisms (Fig. 1c) adopt a relatively more geometry-agnostic design, where vectors are uniformly distributed and aligned parallel to a fixed direction, typically $\mathbf{0} \rightarrow \mathbf{1}$ in normalized objective space. By guiding solutions along parallel trajectories rather than toward anchor points, they are less sensitive to PF curvature [16]. Note that all these strategies depend on good estimation of ideal and nadir points, which is challenging in early search stages or for complex and disconnected fronts.



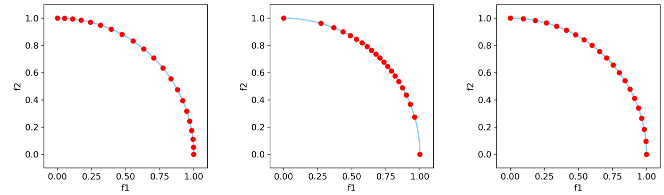
(a) Pull-based (b) Push-based (c) Parallel

Fig. 1. Different reference vector search mechanisms in the objective space

The effect of search mechanisms is illustrated on benchmark problems in Fig. 2. For the non-convex PF of DTLZ2, pull-based reference vectors align well with the front, yielding relatively uniform distributions (Fig. 2a). In contrast, push-based mechanisms exhibit non-uniformity with central bias (Fig. 2b), while parallel mechanisms maintain relatively uniform distributions (Fig. 2c).

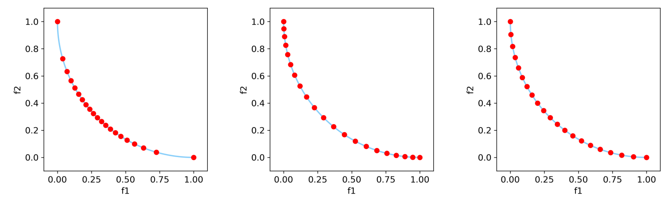
For the inverted (convex) PF of minus-DTLZ2, push-based mechanism provides a better alignment and coverage (Fig. 3a), while pull-based mechanism concentrates the solutions in the central regions (Fig. 3b). Parallel reference vectors offer comparatively better coverage across the PF (Fig. 3c).

Another critical component of reference vector-based methods is the generation of reference points that the vectors pass through. The most common approach is uniform sampling on a unit simplex, as proposed in the NBI work [10], which has been adopted in MOEA/D, NSGA-III, and RVEA among others. However, the performance of these decomposition-based algorithms strongly depends on PF geometry [17]. Despite its popularity, simplex-based sampling has key limitations. Firstly, sampling within the simplex introduces bias towards the central regions of the PF, especially the PFs that extend beyond the simplex. This central bias tends to worsen as the number of objectives increases. Secondly, it typically assumes a continuous PF well mapped by the simplex. For irregular PFs, such as highly curved, inverted or disconnected PFs, many vectors fail to intersect the front, leading to poor coverage and wasted computational effort. To address this, adaptive reference vector methods have been proposed [18], [19]. Such methods they typically require additional user-defined parameters to control adaptation.



(a) Pull-based (b) Push-based (c) Parallel

Fig. 2. DTLZ2 nondominated solutions under different search mechanisms



(a) Pull-based (b) Push-based (c) Parallel

Fig. 3. minus-DTLZ2 nondominated solutions in the normalized space under different search mechanisms

A potential alternative to reference vector adaptation is to develop an improved reference vector generation strategies that provide uniform and comprehensive coverage without relying on simplex geometry or strong assumptions about the PF shape. An effective many-objective sampling approach should avoid central bias, scale with the number of objectives, and remain robust to irregular and disconnected fronts.

Towards the above line of inquiry, this paper investigates *Projected Hypercube Sampling* (PHS) as an alternative to simplex-based systematic sampling. Instead of sampling on a simplex, PHS uniformly samples the points in the unit hypercube and projects those points onto a reference hyperplane spanning the PF-relevant region. Parallel reference vectors constructed from these points enable uniform coverage and reliable intersection with the PF, regardless of its geometry. This strategy eliminates central bias and has the potential to achieve consistent performance across diverse PF shapes. To note, sampling on the projected hypercube bounding box has also been explored in a previous work [20].

However, it focuses on iterative search along selected directions, whereas this study targets reference vector generation for concurrent search, more aligned with decomposition-based EMO algorithms. The sampling design and benchmarking framework are also tailored to evolutionary settings, providing new insights into performance of hypercube based sampling vs conventional simplex-based sampling. The performance of the proposed method is evaluated using well-known set-based performance indicators, including Inverted Generational Distance (IGD) [21] and coverage error [20], [22].

II. Methodology

This section outlines the key steps involved in the PHS approach.

A. Uniform Sampling in the Unit Hypercube

Let M denote the number of objectives. A set of candidate points $\mathcal{C}$ is uniformly sampled in the unit hypercube:

$$\mathcal{C} = \{\mathbf{c}_j \in \mathbb{R}^M \mid \mathbf{c}_j \in [0,1]^M,\ j = 1, \dots, N_c\},$$

where N_c is a user-defined parameter chosen to ensure dense coverage. Uniform sampling in $[0,1]^M$ provides an unbiased representation of extreme and interior trade-offs. For this study, N_c is set to 500,000. Additionally, a user-defined number of points N_e can be sampled along each edge of the unit hypercube to improve boundary representation. In this study, $N_e = 50$ samples are generated on each edge of the hypercube.

B. Projection onto the Reference Hyperplane

Each candidate point is projected normally onto the hyperplane $\sum_{i=1}^{M} f_i = M$. This maps all samples onto a common $(M-1)$-dimensional manifold that uniformly spans the PF relevant region. For $M = 3$, the projected region forms a regular hexagon, while in higher dimensions it forms a convex polytope. This construction encloses any arbitrary shape of PF, including convex, non-convex, and disconnected Pareto fronts.

C. Equidistant Reference Point Selection

To obtain a computationally tractable reference set, a subset of N_r points is selected from the projected samples to achieve approximately uniform spacing. For initialization, a random solution (say s_0) is first considered and the point furthest to it (s_1) is selected. Then, the point (s_2) furthest to s_1 is selected. Thus, $\{s_1, s_2\}$ form the initial pair of selected points. Then a greedy maximin strategy is used, iteratively selecting the point that maximizes its minimum Euclidean distance to the current set, until the required number N_r is achieved. The approach is also referred to as distance-based subset selection [23]. This yields a well-distributed reference set without discretization parameters or combinatorial search.

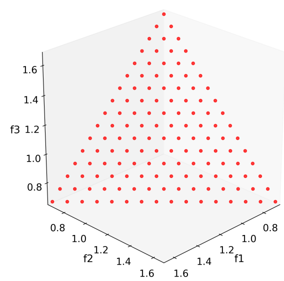
D. Parallel Reference Direction Construction

Each selected reference point $\mathbf{r}$ defines a reference direction parallel to the hyperplane normal $\mathbf{n} = (1, 1, \dots, 1)$:

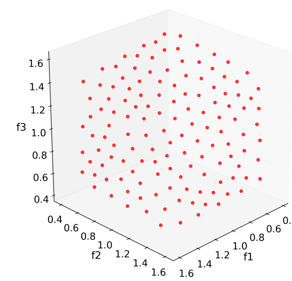
$$\mathbf{d}(t) = \mathbf{r} + t\,\mathbf{n}, \quad t \in \mathbb{R}.$$

All reference vectors are parallel and differ only in their offsets. With sufficient reference point density (sufficiently high N_r), every region of the PF can be intersected by at least one reference direction, independent of its shape.

For intuitive understanding, the difference in reference points generated using simplex-based systematic sampling (SS) and PHS is visualized on the same projection plane, centered at the nadir point $(1, 1, 1)$ in Fig. 4.



(a) Systematic sampling



(b) Projected hypercube sampling

Fig. 4. Comparison of sampling methods for $M = 3$, with $N_r = 136$

As evident, PHS works without strong assumptions on PF geometry, while ensuring that no part of the PF is outside the region reachable by the parallel reference vectors on the hyperplane. The evenly spaced reference points on the projection hyperplane aid uniform coverage of the PFs. The approach is scalable to higher dimensions. It offers a viable alternative to sophisticated reference vector adaptation methods for solving MaOPs with irregular PFs.

III. Numerical Experiments

This section evaluates the proposed PHS against SS on analytically defined PFs with controllable curvature. Performance is measured using IGD and coverage error. To isolate the efficacy of the sampling methods, no search algorithms are used. Instead, the PF approximations are obtained by directly intersecting the parallel reference vectors with the analytical PF, ensuring all sampled points are Pareto-optimal.

A. PF Construction and Experimental Setup

For numerical experiments, a family of PFs, covering both convex and non-convex geometries with controllable curvature are considered. The curvature is controlled using a parameter α, with higher values of α leading to stronger convexity/non-convexity. The PF construction is discussed below.

Let $\mathbf{u} = (u_1, \dots, u_M)$ be a point uniformly sampled (e.g., using SS) on the unit simplex, $\sum_{i=1}^{M} u_i = 1, \quad u_i \geq 0$. The corresponding ℓ_α-norm can be defined as:

$$\|\mathbf{u}\|_\alpha = \left(\sum_{i=1}^{M} u_i^\alpha\right)^{1/\alpha}. \quad (1)$$

The above points can then be transformed to an $\alpha-$ parametrized PF, contained within the unit hypercube as:

- **Non-convex Pareto front:**

$$\mathbf{f} = \frac{\mathbf{u}}{\|\mathbf{u}\|_\alpha}, \quad (2)$$

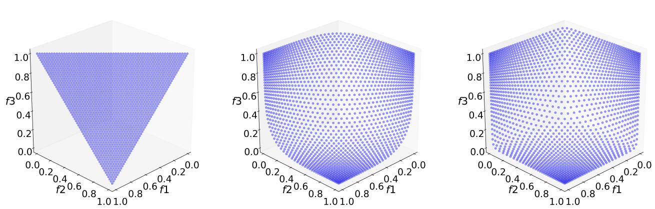
- **Convex Pareto front:**

$$\mathbf{f} = \mathbf{1} - \frac{\mathbf{u}}{\|\mathbf{u}\|_\alpha}, \quad (3)$$

Here, $\mathbf{1}$ denotes a vector of M ones. For $\alpha = 1$, the non-convex front coincides with the unit simplex, exactly matching the geometric assumption of SS. Increasing α introduces higher curvature (e.g., $\alpha = 2$ corresponds to PF of DTLZ2), progressively deviating from the simplex. As for convex PFs, they remain geometrically mismatched with SS even at $\alpha = 1$, as it forms an inverted (or rotated) simplex. Representative PF examples for $\alpha = 1$, 5, and 30 are shown in Figs. 5 and 6.

Experiments are conducted for the number of objectives $M \in \{3, 5, 8, 10\}$, on both convex and non-convex PFs generated using the above formulation for $\alpha \in \{1, 2, 5, 10, 15, 20, 25, 30\}$. To compute the performance metrics, reference PF sets are required (noting the full PFs are continuous). The reference PF sets are constructed using the above approach with sizes of 1953 ($M = 3$), 4845 ($M = 5$), 11440 ($M = 8$), and 48620 ($M = 10$).

For each M, solution sets (PF approximations) with different densities are considered. For SS, this is done by specifying the spacing parameter H [10]. Specifically, we use 91, 136, 231, 351, and 496 reference points for $M = 3$; 210, 495, 715, 1001, and 1365 for $M = 5$; 330, 792, 1716, 3432, and 6435 for $M = 8$; and 715, 2002, 5005, 11440, and 24310 for $M = 10$. For each M, same set of reference points are used across all α values. For PHS, same number of reference points N_r are sampled within the projected hypercube to enable a fair comparison with SS.

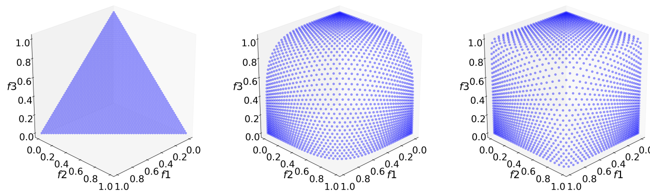


(a) $\alpha = 1$ (b) $\alpha = 5$ (c) $\alpha = 30$

Fig. 5. Convex PFs for different curvature parameter α settings

B. Results

As indicated before, the metrics are computed based on the intersection of the parallel reference directions passing through the reference points constructed for SS/PHS with the true PFs. The IGD values for convex PFs, which have misaligned shape from the regular simplex, are summarized in Table I. For $M = 3$, it can be seen that SS exhibits strong center bias, increasingly missing extreme regions as curvature grows, leading to consistently large and saturated IGD values even with dense reference sets across all α, indicating persistent geometric misalignment. In contrast, PHS achieves well distributed intersections across the PF, achieving lower IGD that further improves with reference point density. For non-convex fronts (Table II), SS performs best at $\alpha = 1$, but its IGD values degrade with increasing curvature, while PHS maintains relatively stable IGD across all α.



(a) $\alpha = 1$ (b) $\alpha = 5$ (c) $\alpha = 30$

Fig. 6. Non-convex PFs for different curvature parameter α settings

The trends for $M = 3$ also persist in higher dimensions, with performance increasingly influenced by reference-point density. SS scales poorly with increasing M, with IGD increasing significantly, especially for convex fronts. In contrast, PHS shows monotonic improvement with increasing reference-point density, achieving lower and more stable IGD in higher dimensions. As all sampled points lie on the true PF, IGD reflects distribution quality rather than convergence, making it an appropriate measure of geometric coverage in this setting.

The results of coverage error metric are shown in Tables III and IV. They reflect broadly similar trends as IGD. For brevity, we have shown the results only for the best (minimum) coverage error achieved across different numbers of reference-points for each M.

C. Effect of Curvature Parameter α

In this section, we visualize the effect of α on the performance of the two sampling strategies, using $M = 5$ as the representative case. Increasing α amplifies PF curvature, highlighting the geometric limitations of simplex-based sampling. SS performs well only at non-convex $\alpha = 1$ (linear PF) due to perfect alignment, but degrades rapidly with increasing curvature for both convex (Figure 7a) and non-convex cases (Figure 7b). In contrast, PHS shows low sensitivity to α, as hyperplane projection of unit hypercube in the normalized objective space enables relatively uniform coverage of the PF-relevant region, resulting in stable performance.

D. Effect of Reference Point Density

Reference-point density affects the two methods differently (Tables I–IV) depending on the PF. For SS, increasing density yields limited improvement where geometric misalignment is present, leading to early saturation in IGD and coverage error.

TABLE I

IGD OBTAINED BY SS AND PHS FOR CONVEX PFS. BETTER IGD VALUES ARE INDICATED IN BOLDFACE, FOR EACH (N_r, M, α).

M	Ref. points (N_r)	α															
		1		2		5		10		15		20		25		30	
		SS	PHS	SS	PHS	SS	PHS	SS	PHS	SS	PHS	SS	PHS	SS	PHS	SS	PHS
3	91	0.0791	**0.0663**	0.1574	**0.0738**	0.2450	**0.0877**	0.2625	**0.0884**	0.2650	**0.0861**	0.2657	**0.0862**	0.2660	**0.0865**	0.2662	**0.0855**
	136	0.0730	**0.0519**	0.1522	**0.0580**	0.2406	**0.0680**	0.2591	**0.0676**	0.2615	**0.0659**	0.2622	**0.0660**	0.2625	**0.0662**	0.2627	**0.0654**
	231	0.0671	**0.0373**	0.1482	**0.0419**	0.2371	**0.0494**	0.2558	**0.0503**	0.2581	**0.0502**	0.2588	**0.0504**	0.2590	**0.0506**	0.2591	**0.0500**
	351	0.0640	**0.0301**	0.1457	**0.0340**	0.2356	**0.0402**	0.2547	**0.0410**	0.2569	**0.0408**	0.2576	**0.0407**	0.2579	**0.0409**	0.2580	**0.0403**
	496	0.0610	**0.0249**	0.1442	**0.0280**	0.2340	**0.0326**	0.2535	**0.0330**	0.2561	**0.0331**	0.2568	**0.0330**	0.2570	**0.0332**	0.2572	**0.0329**
5	210	**0.1821**	0.2509	0.3745	**0.3125**	0.6486	**0.3085**	0.7868	**0.3055**	0.8391	**0.3045**	0.8650	**0.3043**	0.8801	**0.3046**	0.8902	**0.2998**
	495	**0.1641**	0.2121	0.3721	**0.2362**	0.6454	**0.2298**	0.7847	**0.2264**	0.8367	**0.2278**	0.8626	**0.2258**	0.8778	**0.2261**	0.8879	**0.2230**
	715	**0.1658**	0.1880	0.3622	**0.2057**	0.6452	**0.2065**	0.7849	**0.2020**	0.8371	**0.2032**	0.8630	**0.2015**	0.8783	**0.2002**	0.8884	**0.1984**
	1001	**0.1527**	0.1732	0.3637	**0.1810**	0.6446	**0.1898**	0.7837	**0.1862**	0.8358	**0.1875**	0.8620	**0.1806**	0.8773	**0.1795**	0.8874	**0.1776**
	1365	**0.1517**	0.1589	0.3610	**0.1625**	0.6448	**0.1719**	0.7841	**0.1702**	0.8361	**0.1713**	0.8621	**0.1643**	0.8774	**0.1629**	0.8874	**0.1622**
8	330	**0.2903**	0.3941	**0.6022**	0.6587	1.1018	**0.6573**	1.3900	**0.6453**	1.4999	**0.6504**	1.5561	**0.6543**	1.5906	**0.6557**	1.6139	**0.6570**
	792	**0.2594**	0.3941	**0.5865**	0.6009	1.0999	**0.5788**	1.3873	**0.5247**	1.4996	**0.5228**	1.5584	**0.5242**	1.5940	**0.5255**	1.6173	**0.5266**
	1716	**0.2781**	0.3941	0.5778	**0.4855**	1.0931	**0.4595**	1.3797	**0.4384**	1.4915	**0.4355**	1.5506	**0.4359**	1.5872	**0.4367**	1.6118	**0.4370**
	3432	**0.3292**	0.3941	0.5792	**0.4441**	1.0968	**0.4302**	1.3840	**0.4135**	1.4958	**0.4130**	1.5544	**0.4139**	1.5898	**0.4146**	1.6133	**0.4152**
	6435	**0.2291**	0.3647	0.5725	**0.4006**	1.0948	**0.3872**	1.3823	**0.3795**	1.4941	**0.3811**	1.5522	**0.3824**	1.5869	**0.3833**	1.6103	**0.3836**
10	715	–	**0.4100**	**0.6629**	0.8604	1.3042	**1.0506**	1.6757	**0.9980**	1.8188	**1.0031**	1.8934	**1.0138**	1.9394	**1.0178**	1.9706	**1.0214**
	2002	**0.2813**	0.4100	**0.6647**	0.8297	1.3058	**0.8099**	1.6756	**0.7788**	1.8227	**0.7787**	1.8996	**0.7825**	1.9454	**0.7853**	1.9752	**0.7866**
	5005	**0.2647**	0.4100	**0.6470**	0.7754	1.3010	**0.6776**	1.6694	**0.6694**	1.8163	**0.6725**	1.8946	**0.6759**	1.9430	**0.6780**	1.9754	**0.6792**
	11440	**0.2684**	0.4100	0.6480	**0.6167**	1.3027	**0.5827**	1.6726	**0.5830**	1.8192	**0.5865**	1.8958	**0.5893**	1.9421	**0.5911**	1.9734	**0.5885**
	24310	**0.2937**	0.4100	0.6436	**0.5230**	1.3013	**0.5176**	1.6713	**0.5239**	1.8171	**0.5290**	1.8927	**0.5320**	1.9387	**0.5337**	1.9698	**0.5281**

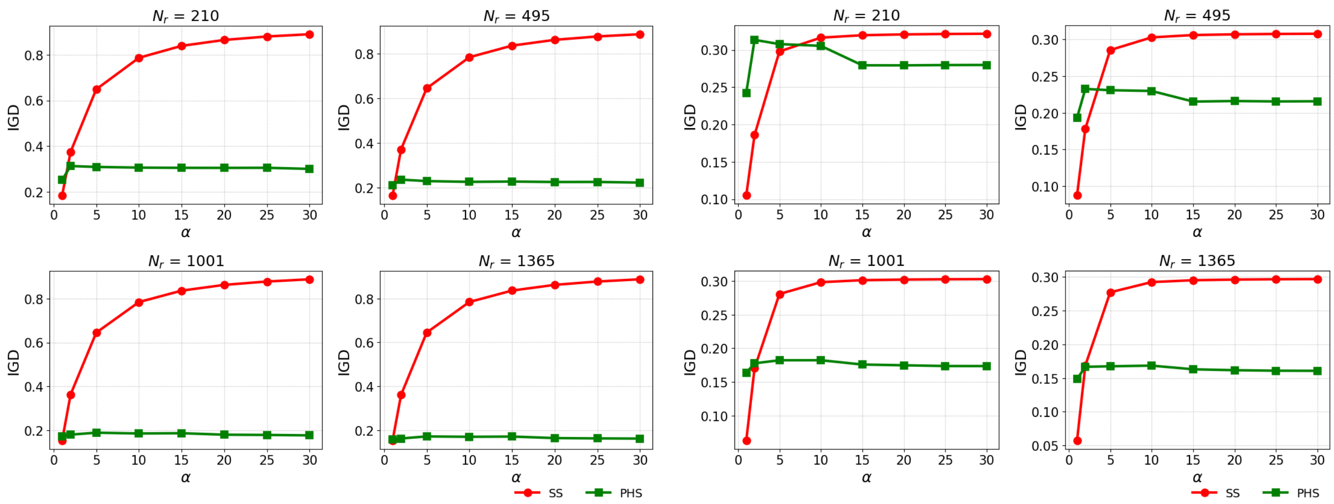
TABLE II

IGD OBTAINED BY SS AND PHS FOR NON-CONVEX PFS. BETTER IGD VALUES ARE INDICATED IN BOLDFACE, FOR EACH (N_r, M, α).

M	Ref. points (N_r)	α															
		1		2		5		10		15		20		25		30	
		SS	PHS	SS	PHS	SS	PHS	SS	PS	SS	PHS	SS	PHS	SS	PHS	SS	PHS
3	91	**0.0408**	0.0627	**0.0676**	0.0784	0.1037	**0.0932**	0.1105	**0.0959**	0.1117	**0.0884**	0.1122	**0.0854**	0.1124	**0.0825**	0.1125	**0.0820**
	136	**0.0326**	0.0509	0.0611	**0.0554**	0.0969	**0.0636**	0.1035	**0.0666**	0.1047	**0.0647**	0.1051	**0.0644**	0.1053	**0.0638**	0.1054	**0.0632**
	231	**0.0245**	0.0374	0.0549	**0.0439**	0.0905	**0.0505**	0.0968	**0.0511**	0.0979	**0.0503**	0.0982	**0.0499**	0.0984	**0.0493**	0.0985	**0.0491**
	351	**0.0196**	0.0299	0.0514	**0.0341**	0.0864	**0.0402**	0.0921	**0.0405**	0.0932	**0.0396**	0.0935	**0.0395**	0.0937	**0.0393**	0.0938	**0.0390**
	496	**0.0163**	0.0249	0.0491	**0.0284**	0.0842	**0.0326**	0.0910	**0.0334**	0.0923	**0.0328**	0.0927	**0.0329**	0.0928	**0.0328**	0.0929	**0.0325**
5	210	**0.1049**	0.2424	**0.1860**	0.3134	**0.2979**	0.3078	0.3164	**0.3054**	0.3196	**0.2793**	0.3208	**0.2792**	0.3213	**0.2796**	0.3217	**0.2798**
	495	**0.0873**	0.1931	**0.1783**	0.2325	0.2855	**0.2308**	0.3028	**0.2297**	0.3059	**0.2153**	0.3070	**0.2160**	0.3074	**0.2155**	0.3077	**0.2157**
	715	**0.0702**	0.1703	**0.1757**	0.2006	0.2837	**0.2042**	0.2992	**0.2049**	0.3019	**0.1946**	0.3029	**0.1934**	0.3034	**0.1924**	0.3038	**0.1923**
	1001	**0.0630**	0.1638	**0.1704**	0.1778	0.2806	**0.1823**	0.2981	**0.1824**	0.3010	**0.1760**	0.3020	**0.1749**	0.3024	**0.1738**	0.3027	**0.1737**
	1365	**0.0570**	0.1485	0.1686	**0.1667**	0.2773	**0.1675**	0.2925	**0.1685**	0.2952	**0.1631**	0.2962	**0.1617**	0.2967	**0.1609**	0.2969	**0.1608**
8	330	**0.1933**	0.4110	**0.3299**	0.7912	**0.5350**	0.7711	**0.5624**	0.6560	**0.5677**	0.6661	**0.5700**	0.6680	**0.5713**	0.6701	**0.5723**	0.6715
	792	**0.1547**	0.4110	**0.3173**	0.7546	**0.5239**	0.6366	0.5506	**0.5292**	0.5572	**0.5291**	0.5603	**0.5302**	0.5622	**0.5313**	0.5635	**0.5317**
	1716	**0.1236**	0.4110	**0.3115**	0.5462	0.5049	**0.4852**	0.5332	**0.4432**	0.5379	**0.4429**	0.5398	**0.4426**	0.5410	**0.4430**	0.5418	**0.4437**
	3432	**0.1026**	0.4110	**0.3083**	0.4883	0.4969	**0.4482**	0.5210	**0.4149**	0.5263	**0.4155**	0.5288	**0.4155**	0.5304	**0.4162**	0.5314	**0.4170**
	6435	**0.0836**	0.3860	**0.3051**	0.4201	0.4979	**0.4017**	0.5235	**0.3806**	0.5293	**0.3831**	0.5316	**0.3840**	0.5330	**0.3849**	0.5340	**0.3857**
10	715	**0.2122**	0.4099	**0.3809**	0.8604	**0.6327**	1.0253	**0.6712**	0.8359	**0.6808**	0.8491	**0.6853**	0.8560	**0.6873**	0.8592	**0.6884**	0.8618
	2002	**0.1697**	0.4099	**0.3682**	0.8604	**0.6070**	0.8337	**0.6393**	0.6932	**0.6462**	0.6637	**0.6494**	0.6659	**0.6514**	0.6676	**0.6528**	0.6689
	5005	**0.1342**	0.4099	**0.3641**	0.8307	**0.6022**	0.6739	**0.6349**	0.6150	0.6433	**0.5962**	0.6475	**0.5988**	0.6500	**0.6000**	0.6518	**0.6010**
	11440	**0.1099**	0.4099	**0.3625**	0.6528	0.5970	**0.5873**	0.6309	**0.5549**	0.6382	**0.5445**	0.6410	**0.5473**	0.6426	**0.5488**	0.6437	**0.5500**
	24310	**0.0879**	0.4099	**0.3566**	0.5332	0.5857	**0.5211**	0.6156	**0.5005**	0.6224	**0.4945**	0.6257	**0.4976**	0.6278	**0.4993**	0.6292	**0.5006**

In contrast, PHS consistently benefits from denser reference sets, as each point defines a distinct and valid parallel search direction (Figure 8a and 8b).

Although PHS is geometrically reliable, higher dimensions require greater density for competitive performance, particularly for $\alpha = 1$ (Table II). Here, the flat PF and large projected



(a) Convex PFs

(b) Non-convex PFs

Fig. 7. Effect of α on IGD metric for different number of reference points N_r, for $M = 5$

TABLE III
BEST (MINIMUM) COVERAGE ERROR ACHIEVED ACROSS DIFFERENT NUMBERS OF REFERENCE-POINTS USED FOR CONVEX PARETO FRONTS. LOWER VALUES ARE HIGHLIGHTED IN BOLD, INDICATING BETTER COVERAGE ERROR BETWEEN SS AND PHS.

M	Sampler	α							
		1	2	5	10	15	20	25	30
3	SS	0.4082	0.5284	0.6911	0.7068	0.7071	0.7071	0.7071	0.7071
	PHS	**0.0692**	**0.0942**	**0.0999**	**0.0993**	**0.1009**	**0.1015**	**0.1016**	**0.1016**
5	SS	0.6708	0.8534	1.2279	1.3892	1.4394	1.4611	1.4741	1.4829
	PHS	**0.4390**	**0.3973**	**0.4787**	**0.4814**	**0.4814**	**0.3676**	**0.3677**	**0.3698**
8	SS	**0.8018**	1.0243	1.6148	1.9295	2.0481	2.1103	2.1485	2.1736
	PHS	0.9171	**0.7132**	**0.6830**	**0.6402**	**0.6701**	**0.6708**	**0.6709**	**0.6710**
10	SS	**0.8515**	1.0842	1.7878	2.1942	2.3525	2.4364	2.4882	2.5211
	PHS	1.0231	**0.8215**	**0.8859**	**0.9187**	**0.9192**	**0.9192**	**0.9192**	**0.9192**

TABLE IV
BEST (MINIMUM) COVERAGE ERROR ACHIEVED ACROSS DIFFERENT NUMBERS OF REFERENCE-POINTS USED FOR NON-CONVEX PARETO FRONTS. LOWER VALUES ARE HIGHLIGHTED IN BOLD, INDICATING BETTER COVERAGE ERROR BETWEEN SS AND PHS.

M	Sampler	α							
		1	2	5	10	15	20	25	30
3	SS	**0.0268**	0.1953	0.3672	0.4109	0.4205	0.4252	0.4280	0.4299
	PHS	0.0947	**0.1159**	**0.1191**	**0.1079**	**0.1080**	**0.1080**	**0.1080**	**0.1015**
5	SS	**0.0937**	0.3073	0.5813	0.6367	0.6501	0.6550	0.6625	0.6671
	PHS	0.4416	**0.4616**	**0.4320**	**0.4330**	**0.3701**	**0.3783**	**0.3818**	**0.3834**
8	SS	**0.1094**	**0.4343**	0.8444	0.9251	0.9564	0.9743	0.9884	0.9982
	PHS	0.9538	0.7308	**0.7170**	**0.6661**	**0.6626**	**0.6600**	**0.6664**	**0.6699**
10	SS	**0.1179**	**0.4822**	0.9493	1.0653	1.1072	1.1307	1.1457	1.1560
	PHS	0.9598	0.8985	**0.8617**	**0.8628**	**0.8134**	**0.8148**	**0.8151**	**0.8152**

region can result in insufficient intersections under sparse sampling. This is more evident for $M = 8$ and $M = 10$, where PHS initially shows higher IGD values, but improves rapidly with density. Overall, PHS exhibits consistent improvements with increasing density, indicating a density requirement rather than a geometric limitation.

E. Additional Experiments, Codes and Data

The source code and data are available from this Github link. In addition to the experiments discussed here, further data is presented to support the observations, including:

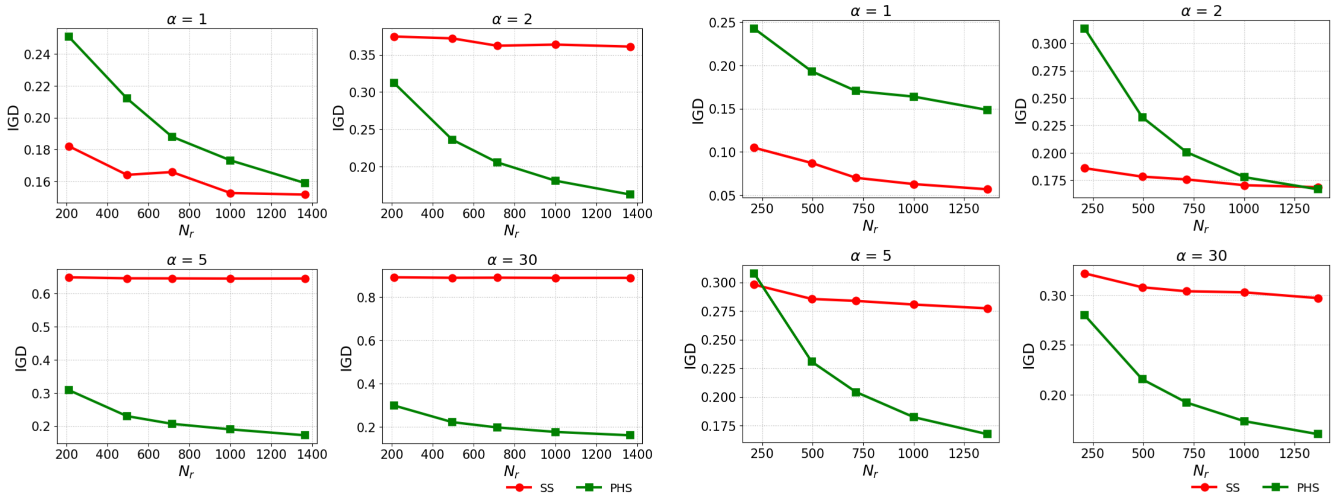
- experiments on 3-objective DTLZ7 to demonstrate efficacy of PHS for disconnected PFs,
- experiments with 31 different PHS intializations and comparison with SS to establish statistical reliability, and
- experiments with $M = 15$, to verify further scalability of the method.

IV. CONCLUSION AND FUTURE WORK

This study examined the geometric limitations of simplex-based systematic sampling (SS) and investigated PHS as a geometry-independent alternative for reference vector generation for MOP/MaOPs. Using analytically defined convex and non-convex PFs with controllable curvature, PHS was systematically compared with SS across objective dimensions up to $M = 10$.

The results illustrate that simplex-based sampling is highly sensitive to PF geometry. SS performs well only when the PF coincides well with the simplex (non-convex $\alpha = 1$), but its performance deteriorates rapidly as curvature increases or when the front is convex, leading to poor coverage error and IGD. Increasing reference point density offers limited improvement for cases of geometric mismatch.

In contrast, PHS provides reliable geometric coverage across a wide range of PF shapes by uniformly spanning the PF-relevant region through hyperplane projection. Although PHS requires a higher reference-point density in higher dimensions to achieve better results for linear PFs, its performance improves monotonically with increasing density. With sufficient density, PHS achieves lower and more stable performance than SS across the tested scenarios. Future work would involve



(a) Convex PFs

(b) Non-convex PFs

Fig. 8. Effect of number of reference points N_r on IGD metric for different values of curvature parameter α, for $M = 5$

incorporating PHS into prominent decomposition-based EMO frameworks to further investigate their performance.

Acknowledgement

This research was supported by the Commonwealth through an Australian Government Research Training Program Scholarship [DOI: https://doi.org/10.82133/C42F-K220] and the Australian Research Council (ARC) Discovery Project DP220101649. Microsoft Copilot and OpenAI ChatGPT were used for editorial improvements.

References

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

[2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," ETH Zurich, TIK Report, Tech. Rep., 2001.

[3] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[4] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[5] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[6] E. J. Hughes, "Fitness assignment methods for many-objective problems," in *Multiobjective Problem Solving from Nature: From Concepts to Applications*, 2007, pp. 307–329.

[7] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[8] K. Li, "A survey of multi-objective evolutionary algorithm based on decomposition: Past and future," *IEEE Transactions on Evolutionary Computation*, 2024, early access.

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

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

[11] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[12] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[13] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi- and many-objective optimization," in *ACM Genetic and Evolutionary Computation Conference*, 2014, pp. 645–652.

[14] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, and L. Jiao, "On the use of two reference points in decomposition-based multiobjective evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 34, pp. 89–102, 2017.

[15] K. S. Bhattacharjee, H. K. Singh, T. Ray, and Q. Zhang, "Decomposition based evolutionary algorithm with a dual set of reference vectors," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 105–112.

[16] M. Elarbi, S. Bechikh, C. A. C. Coello, M. Makhlouf, and L. B. Said, "Approximating complex Pareto fronts with predefined normal-boundary intersection directions," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 809–823, 2019.

[17] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.

[18] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2017.

[19] X. Ma, Y. Yu, X. Li, Y. Qi, and Z. Zhu, "A survey of weight vector adjustment methods for decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 634–649, 2020.

[20] B. S. Saini, H. K. Singh, B. Shavazipour, and K. Miettinen, "An efficient iterative approach for uniformly representing Pareto fronts," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2025, pp. 241–256.

[21] C. A. Coello Coello and M. Reyes Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Mexican International Conference on Artificial Intelligence*, 2004, pp. 688–697.

[22] S. Sayın, "Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming," *Mathematical Programming*, vol. 87, no. 3, pp. 543–560, 2000.

[23] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance-based subset selection for benchmarking in evolutionary multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 904–912, 2018.