Boundary-Rejection-Sampling for Computing Pareto Fronts Efficiently and Uniformly

Markus Herrmann-Wicklmayr and Kathrin Flaßkamp

Systems Modeling and Simulation, Saarland University, Saarbrücken, Germany
{markus.herrmannwicklmayr,kathrin.flasskamp}@uni-saarland.de

Abstract. This paper presents a novel method for efficiently sampling the Pareto fronts of three-objective optimization problems. The core principles are (1) approximating the Pareto front *boundary*, (2) formulating the *rejection* criterion, and (3) efficiently *sampling* scalarized auxiliary problems for Pareto front computation. Starting from the individual minima, the boundary of the Pareto front is sampled and then projected to a hyperplane. This plane is then uniformly sampled within the bounded region. Outside samples are discarded, since they would lead to redundant Pareto optimal solutions. Finally, only the remaining samples are used to determine parameters for the Pascoletti-Serafini scalarization method by which an approximately uniform coverage of the Pareto front can be computed. Results from several test problems demonstrate superior efficiency and uniformity. Our approach is faster on almost all test problems, improves the averaged uniformity metric by a factor of about three, and reduces redundant calculations by a factor greater than fourteen.

Keywords: Multi-objective optimization · Pareto front sampling · Rejection sampling

1 Introduction

Multi-objective optimization problems (MOOPs) are ubiquitous in engineering, involving the simultaneous minimization of conflicting objectives. Since a single solution optimizing all objectives rarely exists, the goal is to identify the set of nondominated points, known as the Pareto front (PF). For high-dimensional nonlinear problems, e.g., in optimal control, generating a uniform approximation of the PF is a significant challenge [13,7].

To approximate the PF, scalarization methods transform the MOOP into a sequence of single-objective subproblems. While the classical weighted sum method is limited to convex problems, advanced techniques such as the ϵ-constraint method, normal boundary intersection, and the modified Polak Problem [9, Chapter 2.5.1-2.5.3] handle nonconvex fronts. However, methods like normal boundary intersection may fail at boundary regions of the convex hull [16,14]. A generalized framework is the Pascoletti-Serafini (PS) scalarization [19], which subsumes many of these approaches [9, Chapter 2.5]. The PS method computes points of the PF by sampling the shooting origin (SO) and the shooting direction vector (SDV).

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

In this paper, our focus is on deriving sampling strategies that avoid parameter choices leading to redundant solutions. In contrast to works focusing on revised scalarization techniques with slack variables [7,1] or specialized formulations for disconnected sets [2,3], our objective is the systematic derivation of sampling parameters. We restrict our comparison to gradient-based methods and exclude evolutionary algorithms, as they often yield uneven or incomplete coverage [5].

To improve efficiency, *boundary-based* strategies first approximate the PF boundary and then sample the interior. The Successive Pareto Optimization (SPO) [18] and the Modified Successive Pareto Optimization (MSPO) [5] are, to our knowledge, the only methods in this class so far. However, these methods suffer from geometric limitations. As noted in [14], SPO and MSPO assume the boundary is described by convex combinations of individual minima (IMs). This fails if IMs are non-unique or coplanar, potentially missing PF regions.

Recent efforts address this by extensive decision space sampling. Dolatnezhadsomarin et al. [6] proposed using a large point cloud to guide shooting parameters. While avoiding pitfalls of IM interpolation, this relies on uniform sampling [17], which is computationally intractable for high-dimensional problems. For $n \geq 100$ variables, as typical in optimal control, the *curse of dimensionality* renders decision space sampling unusable.

A gap in the current state of research remains. Thus, we propose a novel PS-based algorithm that addresses shortcomings of SPO and MSPO without succumbing to the curse of dimensionality. Our approach combines three key elements, namely the individual minima and the nadir point, a robust boundary generation scheme avoiding the issues described in [14], and a rejection scheme for filtering parameters. More specifically, the boundary samples are projected onto a plane, where they define a generally nonconvex polygon. A uniformly sampled outer approximation of this polygon is then used rejection step to generate new PS parameters.

Notation. Bold symbols $\mathbf{0}$ and $\mathbf{1}$ denote vectors of zeros and ones with context-dependent dimensions. We use the operators $\mathrm{sum}(v) := \sum_i v_i$ for the sum of all elements of a vector v; $\mathrm{diag}(v)$ for a diagonal matrix with diagonal v; $\min_{\mathrm{row}}(\cdot)$ for the row-wise minimum; $\mathrm{round}(\cdot)$ and $\mathrm{ceil}(\cdot)$ for rounding to the nearest and next larger integer; and $\mathrm{flip}_i(\cdot)$, which reverses the order of elements along the i-th dimension. The angle between two vectors u and v in degrees is computed by $\measuredangle(u, v)$. For a vector $v \in \mathbb{R}^n$, inequalities like $v \geq 0$ should be understood component-wise, i.e. $v_i \geq 0$ for all $i \in \{1, \ldots, n\}$.

2 Preliminaries

For the reader's convenience, we repeat the problem setting and the characterization of important quantities in MOOPs as originally introduced in [12].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

2.1 Multiobjective Optimization Problem

Consider the MOOP

$$\min_{x \in \mathbf{X}} \ \mathcal{J}(x) \tag{$\mathcal{P}$}$$

with the vector $\mathcal{J}(x) := \left[J_1(x) \,\middle|\, \ldots \,\middle|\, J_{n_J}(x)\right]^\intercal$ of n_J objectives $J_i : \mathbf{X} \to \mathbb{R}$, $i = 1, \ldots, n_J$. The set $\mathbf{X} \subseteq \mathbb{R}^{n_x}$ denotes the feasible set. The vector-valued minimization in $(\mathcal{P})$ is clarified by definitions and conventions adopted from [20].

Definition 1 (Pareto optimality, nondominance). *A point $x^\star \in \mathbf{X}$ is a Pareto optimal solution (POS) to the multi-objective optimization problem $(\mathcal{P})$ if there does not exists any feasible $x \in \mathbf{X}$ such that*

$$J_i(x) \leq J_i\left(x^\star\right) \text{ for all } i \in \{1, \ldots, n_J\} \text{ and}$$
$$J_k(x) < J_k\left(x^\star\right) \text{ for at least one } k \in \{1, \ldots, n_J\}.$$

The respective image value $\mathcal{J}\left(x^\star\right)$ is called nondominated. The set of all nondominated points is the nondominated set or Pareto front $\mathbf{J}_\mathcal{P} := \left\{\mathcal{J}(x) \;\middle|\; x \in \mathbf{X}_\mathcal{P}\right\}$, with the Pareto set $\mathbf{X}_\mathcal{P} := \arg\min_{x \in \mathbf{X}} \mathcal{J}(x) = \left\{x \in \mathbf{X} \;\middle|\; x \text{ is a POS to } (\mathcal{P})\right\}$.

2.2 Characteristic Quantities

We derive characteristic quantities of an MOOP. We denote by x_i^*, $i = 1, \ldots, n_J$ the individual minima (IMs), i.e. solutions to the single-objective optimization problem $\min_{x \in \mathbf{X}} \ J_i(x), \ i = 1, \ldots, n_J$. Evaluating the full objective vector $\mathcal{J}$ at all IM defines the pay-off matrix (see, e.g. [8]) $\Phi = (\mathcal{J}(x_1^*), \ldots, \mathcal{J}(x_{n_J}^*))$.

Definition 2 (Utopia & nadir point). *We define the utopia point (UP) $\mathcal{J}_\mathrm{UP}$ and the nadir point (NP) $\mathcal{J}_\mathrm{NP}$ as the row-wise minimum and maximum of Φ*[1]*.*

Definition 3 (Normalized image space). *Let $C_\mathrm{NP,UP}$ denote the positive definite, diagonal matrix $C_\mathrm{NP,UP} := \mathrm{diag}(\mathcal{J}_\mathrm{NP} - \mathcal{J}_\mathrm{UP})^{-1}$. Then, the operation $\bar{\mathcal{J}}(x) = C_\mathrm{NP,UP}(\mathcal{J} - \mathcal{J}_\mathrm{UP})$ shifts and scales the image space such that the PF is contained in the unit box (or hypercube) spanned by $\mathbf{0}$ and $\mathbf{1}$. We call this the normalized image space. The bar accent indicates quantities in this normalized image space, e.g. $\bar{\mathcal{J}}_\mathrm{UP} = \mathbf{0}$ and $\bar{\mathcal{J}}_\mathrm{NP} = \mathbf{1}$ denote the UP and NP, respectively.*

2.3 Pascoletti-Serafini Scalarization

Applying PS scalarization [9, Chapter 2.1], [19], for which we choose the nonempty, closed, pointed, convex cone $\mathsf{K} = \mathbb{R}^{n_J}_{\geq 0}$, to $(\mathcal{P})$ yields the scalar optimization problem

$$\min_{x \in \mathbb{X},\, l \in \mathbb{R}} \ l \quad \text{s.t.} \quad 0 \leq \hat{\mathcal{J}}_\mathrm{SO} + l\hat{d} - \mathcal{J}(x), \tag{$\tilde{P}(\hat{\mathcal{J}}_\mathrm{SO}, \hat{d})$}$$

parameterized in the SO $\hat{\mathcal{J}}_\mathrm{SO}$ and the SDV $\hat{d}$.

[1] Note that, in literature, it is sometimes further distinguished between pseudo and real NPs [8, Chapter 2.2].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

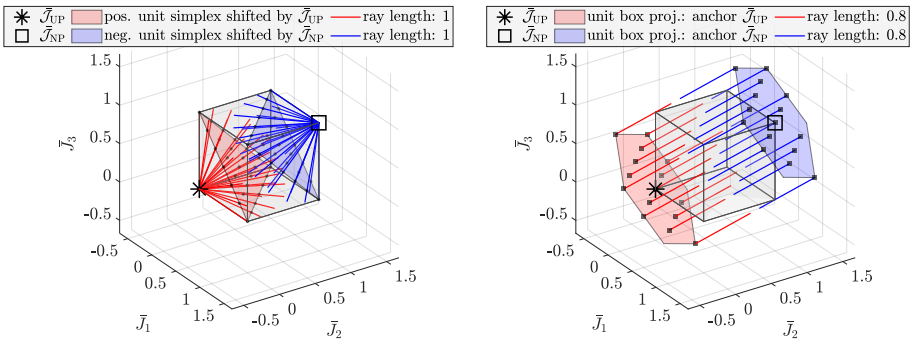

Fig. 1: **Left:** Projecting the unit cube along $\eta = \bar{\mathcal{J}}_{\mathrm{NP}} - \bar{\mathcal{J}}_{\mathrm{UP}} = \mathbf{1}$ on hyperplane (described by $\eta^\intercal x + b = 0$, $x \in \mathbb{R}^{n_J}$) with bias $b_{\mathrm{UP}} = -\eta^\intercal \bar{\mathcal{J}}_{\mathrm{UP}} = 0$ and $b_{\mathrm{NP}} = -\eta^\intercal \bar{\mathcal{J}}_{\mathrm{NP}} = -3$ yields two hexagons; SOs (square markers) lay on hexagons; SDV fixed to η (red) and $-\eta$ (blue).
Right: SO fixed (either $\bar{\mathcal{J}}_{\mathrm{UP}}$ or $\bar{\mathcal{J}}_{\mathrm{NP}}$); SDVs point from SO through points on the (positive/negative and shifted) unit simplex (dot markers).

Convention 2.1. We express a specific parameter realization using a *check* accent instead of the *hat* accent. For example, $\check{d}$ is a specific realization of $\hat{d}$.

To simplify derivations and cover sub-problems with a single formulation, we introduce additional parameters to augment the problem formulation of $(\tilde{P}(\hat{\mathcal{J}}_{\mathrm{SO}}, \hat{d}))$ and define

$$\min_{x \in \mathbb{X},\, l \in \mathbb{R}} \ \hat{s}_l\, l \quad \text{s.t.} \quad 0 \leq \hat{A}_{\mathrm{PS}} \left(\hat{\mathcal{J}}_{\mathrm{SO}} + l\hat{d} - \tilde{\mathcal{J}}(x) \right) \tag{$P(p)$}$$

with $\tilde{\mathcal{J}}(x) = \hat{T}_{\mathcal{J}}(\mathcal{J}(x) - \hat{\mathcal{J}}_{\mathrm{shift}})$ and $p = \{\hat{\mathcal{J}}_{\mathrm{SO}}, \hat{d}, \hat{s}_l, \hat{A}_{\mathrm{PS}}, \hat{T}_{\mathcal{J}}, \hat{\mathcal{J}}_{\mathrm{shift}}\}$.

Appropriately setting these parameters effectively implements the weighted-sum (WS) method, normalizes the image space or performs the PS method in projections of the entire image space. Depending on the SO and the SDV one can set $\check{s}_l = -1$ to maximize l and $\check{s}_l = +1$ to minimize l.

2.4 Determining Shooting Origin and Shooting Direction Vector

For the "standard" PS scalarization $(\tilde{P}(\hat{\mathcal{J}}_{\mathrm{SO}}, \hat{d}))$ as well as for our general formulation $(P(p))$, we must choose realizations of the SO and SDV. Many algorithms either select a fixed SO and vary SDV [15,11], or a varying SO, but fixed SDV [18,5,6,10].

We propose to always start with normalizing the image space, resulting in the PF being contained in the unit box spanned by $\bar{\mathcal{J}}_{\mathrm{UP}} = \mathbf{0}$ and $\bar{\mathcal{J}}_{\mathrm{NP}} = \mathbf{1}$. Then, we provide two alternative parameter realizations, implementing the radial and the parallel shooting approach, respectively. These are shown in Figure 1, where a.o. the unit simplex $\mathbb{F}_{\mathrm{US}} := \{\gamma \in [0,1]^{n_J} \mid \mathrm{sum}(\gamma) = 1\}$ is depicted. Note that the parallel shooting approach using hexagons has similarities to the "parameter set restriction" from [9, Chapter 2.3].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

2.5 Formal Problem Setting

Regardless of choosing the radial shooting, the parallel shooting, or any other method, the goal is to appropriately selecting parameters which allow for an efficient and uniform sampling of a PF. Naive parameter selection often leads to *redundant* computations, since one PF sample can often be obtained by several parameter realizations. In the PS method, with fixed SO or SDV, we define a solution as redundant if the inequality constraint

$$0 \leq \hat{\mathcal{J}}_{\mathrm{SO}} + l\hat{d} - \tilde{\mathcal{J}}(x) \tag{1}$$

in $\tilde{P}(\hat{\mathcal{J}}_{\mathrm{SO}}, \hat{d})$ is not fulfilled with equality. This happens if rays are not shot towards a nondominated point. Consider the case with a fixed SDV. If (1) is not fulfilled with equality, then there exists a different SO for which equality holds, yielding the same nondominated point [9, Theorem 2.21.]. Hence, redundancy, as defined, is tightly coupled to inefficient sampling. Thus, avoiding redundant computations increases the efficiency of the PF sampling algorithm.

3 State of the Art Algorithms Exploiting the Pareto Front Boundary Information

For subsequent comparison of our new methods with existing boundary-based methods we revisit the two published member of this class.

3.1 Successive Pareto Optimization Algorithm

Successive Pareto Optimization (SPO), proposed by Mueller-Gritschneder et al. [18], uses a fixed SDV, categorizing it as a parallel shooting approach.

Generating new SOs $\mathcal{J}_{\mathrm{SO},k}$ for interior samples relies on mapping target weight vectors $\mathbf{w}_k$ via a linear combination of pre-calculated boundary sets. First, boundary weight vectors and their corresponding Pareto-optimal performance vectors are arranged into matrices $\mathbf{W}_B$ and $\mathbf{F}_B$. To determine the coefficients for the linear combination, denoted as vector $\mathbf{a}_k$, the approach solves a linear program subject to $\mathbf{W}_B\mathbf{a}_k = \mathbf{w}_k$. The objective function minimizes a weighted L_1 norm based on a distance metric $\mathbf{d}_k$, which prioritizes boundary points geometrically close to the target weight vector. This strategy aims to ensure an even distribution of the resulting SOs and prevent clustering of solutions. Finally, the optimal coefficients $\mathbf{a}_k^*$ are applied to the boundary performance matrix to compute the new SO via $\mathcal{J}_{\mathrm{SO},k} = \mathbf{F}_B\mathbf{a}_k^*$.

3.2 Modified Successive Pareto Optimization Algorithm

Modified Successive Pareto Optimization (MSPO), proposed by Dolatnezhadsomarin and Khorram [5], fits neither of the shooting approaches, since it repeatedly uses radial shooting to obtain samples from the PF, but the SO changes multiple times during the process.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

Algorithm 1 Modified Successive Pareto Optimization (MSPO)

(0) Initialization

Compute the individual minima, form Φ and derive the utopia point $\mathcal{J}_{\mathrm{UP}}$ from Φ. If applicable, derive the nadir point $\mathcal{J}_{\mathrm{NP}}$ from Φ; otherwise compute it with a different method. Compute the normalization matrix $C_{\mathrm{NP,UP}}$. In p, set $\tilde{T}_{\mathcal{J}} = C_{\mathrm{NP,UP}}$, $\tilde{\mathcal{J}}_{\mathrm{shift}} = \mathcal{J}_{\mathrm{UP}}$, $\check{s}_l = +1$. Select $\mathcal{J}_{\mathrm{SO}} \leq \mathcal{J}_{\mathrm{UP}}$ and compute $\bar{\mathcal{J}}_{\mathrm{SO}} = C_{\mathrm{NP,UP}}\left(\mathcal{J}_{\mathrm{SO}} - \mathcal{J}_{\mathrm{UP}}\right)$. Choose the number samples N "between" first and second IM.

(1) Parameter settings $((i,j) \in P = \{(1,2),(2,3),(1,3)\})$

$d^k := \bar{\mathcal{J}}^k - \bar{\mathcal{J}}_{\mathrm{SO}}, \quad k = 1,2,3 \quad (\bar{\mathcal{J}}^k$ is the k-th IM in the normalized image space)

β-approach	original	$\beta_{i,j} := \measuredangle\left(d^i, d^j\right)$
	corrected	$\beta_{i,j} := \measuredangle\left(d^i_{\{i,j\}}, d^j_{\{i,j\}}\right)$

$\gamma := (\beta_{2,3} + \beta_{1,3})/(N-1), \quad N_{1,2} := N,$

$N_{2,3} := \mathrm{round}(\beta_{2,3}/\gamma) + 1, \quad N_{1,3} := \mathrm{round}(\beta_{1,3}/\gamma) + 1$

ξ-approach	original	$\xi_{i,j} := \beta_{i,j}/(N_{i,j}-1)$
	corrected	$\xi_{i,j} := (\pi/2)/(N_{i,j}-1)$

$\xi_{\mathrm{interior}} := \xi_{1,2}$, **unclear to which ξ it is referred to in original algorithm description** $A_{\mathrm{PS},i,j} := \mathrm{diag}(e_i + e_j) \quad A_{\mathrm{PS},I} := I_3.$

(2) Approximation of the Pareto front boundaries

For each $(i,j) \in P$, initialize $\delta\bar{F}^{\{i,j\}} := \emptyset$.

For $k = 0$ **to** $N_{i,j} - 1$:

Set $\alpha := k \cdot \xi_{i,j}$, $w_1 := \sin^2(\alpha)$, $w_2 := \cos^2(\alpha)$, $\tilde{d} := w_1 d^i + w_2 d^j$. In p, set $\check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}_{\mathrm{SO}}$, $\check{d} = \tilde{d}$, $\check{A}_{\mathrm{PS}} = A_{\mathrm{PS},i,j}$. Then, solve $(P(p))$ and add $\bar{\mathcal{J}}(x^*)$ to the set $\delta\bar{F}^{\{i,j\}}$.

(3) Construct auxiliary sets and compute interior points

Construct the sets A and B:

$A := \delta\bar{F}^{\{1,2\}} \setminus \{\bar{\mathcal{J}}^1, \bar{\mathcal{J}}^2\}, \; B := \left(\mathrm{flip}_2\left(\delta\bar{F}^{\{2,3\}}\right) \setminus \{\bar{\mathcal{J}}^2, \bar{\mathcal{J}}^3\}\right) \cup \bar{\mathcal{J}}^3 \cup \left(\delta\bar{F}^{\{1,3\}} \setminus \{\bar{\mathcal{J}}^1, \bar{\mathcal{J}}^3\}\right),$

$\bar{\mathcal{J}}^{k,A} :=$ "k-th point in A", $\quad \bar{\mathcal{J}}^{k,B} :=$ "k-th point in B".

For $k = 1$ **to** $|A|$:

Compute interior parameters:

$\bar{\mathcal{J}}^{A,B}_{\mathrm{UP}} := \min_{\mathrm{row}}\left((\bar{\mathcal{J}}^{k,A}, \bar{\mathcal{J}}^{k,B})\right), \quad \tilde{d}^A := \bar{\mathcal{J}}^{k,A} - \bar{\mathcal{J}}''_{\mathrm{SO}}, \quad \tilde{d}^B := \bar{\mathcal{J}}^{k,B} - \bar{\mathcal{J}}''_{\mathrm{SO}},$

$\theta := \measuredangle(\tilde{d}^1, \tilde{d}^2), \quad M := \mathrm{ceil}(\theta/\xi_{\mathrm{interior}}) + 2.$

If $M \geq 3$:

Define $\mu := (\pi/2)/(M-1)$.

For $m = 1$ **to** $M - 2$ (skipping boundaries):

Set $\varphi := m \cdot \mu$, $\tilde{d} := \sin^2(\varphi)\tilde{d}^A + \cos^2(\varphi)\tilde{d}^B$. In p, set $\check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}^{A,B}_{\mathrm{UP}}$, $\check{d} = \tilde{d}$, $\check{A}_{\mathrm{PS}} = A_{\mathrm{PS},I}$. Then, solve $(P(p))$ and add $\bar{\mathcal{J}}(x^*)$ to the interior samples.

Reproducing MSPO was challenging due to ambiguities in the algorithmic description in [5]. Thus, a compact version of MSPO is presented in Algorithm 1. In [5], crucial details regarding normalization and precise update rules for angular parameters (specifically β and ξ in Steps 2 and 3 of Algorithm 1) are either unspecified or yield geometrically inconsistent results when implemented.

To address these issues and ensure fair comparison, we analyzed Algorithm 1 on the convex unit sphere problem, i.e. test problem 4 in [5]). Specifically, we combine different realizations of Algorithm 1 (resulting from different choices for the β- and ξ-approach) with feasible choices for the SO.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

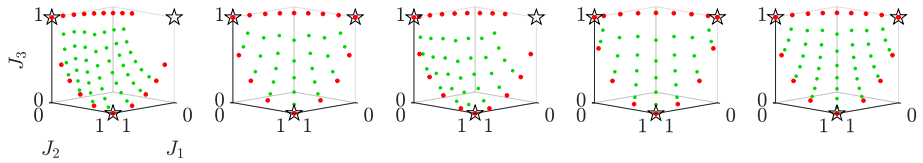

Fig. 2: MSPO with $N = 9$ applied to the unit sphere problem. From left to right:
(a) Original β-approach; original ξ-approach; $\mathcal{J}_{\mathrm{SO}} = \mathbf{0} = \mathcal{J}_{\mathrm{UP}}$.
(b) Corrected β-approach; original ξ-approach; $\mathcal{J}_{\mathrm{SO}} = \mathbf{0} = \mathcal{J}_{\mathrm{UP}}$.
(c) Corrected β-approach; original ξ-approach; $\mathcal{J}_{\mathrm{SO}} = -0.3 \cdot \mathbf{1} < \mathcal{J}_{\mathrm{UP}}$.
(d) Corrected β-approach; corrected ξ-approach; $\mathcal{J}_{\mathrm{SO}} = -0.3 \cdot \mathbf{1} < \mathcal{J}_{\mathrm{UP}}$.
(e) Original β-approach; corrected ξ-approach; $\mathcal{J}_{\mathrm{SO}} = \mathbf{0} = \mathcal{J}_{\mathrm{UP}}$.

Our exact implementation of MSPO as described in [5] cannot reproduce their Figure 6(e), cf. Figure 2 (a). This results from β values smaller than $\pi/2$, causing the interpolation to stop prematurely. Using the corrected β-approach, the β values coincidentally equal $\pi/2$ and we can reproduce Figure 6(e) from [5], cf. Figure 2 (b). However, when choosing a SO smaller than the UP, resulting in β values smaller than $\pi/2$, this implementation yields undesired results again, cf. Figure 2 (c). This can be fixed by choosing the corrected ξ-approach, cf. Figure 2 (d). This approach is similar to what the authors have done to determine the angular step size μ in step (3) of Algorithm 1. Note, that we observed that with the original β-approach, corrected ξ-approach[2] and $\mathcal{J}_{\mathrm{SO}} = \mathbf{0} = \mathcal{J}_{\mathrm{UP}}$, cf. Figure 2 (e), we obtain a different number of interior samplings compared to Figure 6(e) from [5]. However, it can be concluded from Figure 2 (d) that the corrected ξ-approach is required to always obtain an appropriate sampling. Based on this, we conclude that the algorithm with the original β-approach and corrected ξ-approach likely reflects the authors' intent, despite deviations. Consequently, all subsequent comparisons utilize this robust, altered version of MSPO.

4 Boundary-Rejection-Sampling

Now, our proposed Boundary-Rejection-Sampling (BRS) method is presented. Let us assume that the IMs, which form the basis for the algorithm, are computed. Then, we construct the pay-off matrix Φ to obtain the UP $\mathcal{J}_{\mathrm{UP}}$. The NP $\mathcal{J}_{\mathrm{NP}}$ is either assumed to be known or reconstructable from Φ.

We fix the parameter realizations

$$\check{A}_{\mathrm{PS}} = I_3, \quad \check{T}_{\mathcal{J}} = C_{\mathrm{NP,UP}}, \quad \check{\mathcal{J}}_{\mathrm{shift}} = \mathcal{J}_{\mathrm{UP}}. \tag{2}$$

for all shooting approaches in the BRS algorithm to work in the full, normalized image space.

[2] With these choice for the two approaches, the β values only affect the number of boundary and interior samples.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

4.1 Sampling the Pareto Front Boundary

To sample the PF boundary, we use radial shooting. Specifically, we sample points on the boundary points of the unit simplex $\beta \in \mathbb{F}_{\mathrm{US}}$ with $\min(\beta) = 0$. Then, a shooting direction is given by $\tilde{d}(\beta) := \beta/\|\beta\|$ and we can set

$$\check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}_{\mathrm{UP}}, \quad \check{d} = \tilde{d}(\beta), \quad \check{s}_l = 1 \tag{3a}$$

$$\text{or} \quad \check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}_{\mathrm{NP}}, \quad \check{d} = -\tilde{d}(\beta), \quad \check{s}_l = -1. \tag{3b}$$

These two cases are displayed in Figure 1 (right). However, the figure also shows points sampled in the interior of the unit simplices. We choose the approach (3b) to sample the PF boundary. The boundary sampling algorithm is shown in Algorithm 2.

Algorithm 2 Boundary Sampling

(0) Initialization
Set parameters of $(P(p))$, except $\hat{d}$, as in (2) and (3b). Choose N_{facet} and compute $N^{\mathrm{init}}_{\mathrm{boundary}}$.

(1) Parameter generation
Sample the boundary of the unit simplex uniformly using $N^{\mathrm{init}}_{\mathrm{boundary}}$ points.

(2) Boundary sampling
For each $k \in \{1, \dots, N^{\mathrm{init}}_{\mathrm{boundary}}\}$, set $\check{d} = -\tilde{d}(\beta_k)$ and solve $(P(p))$.

The BRS method aims at sampling the boundary such that neighboring points have a maximum distance $\Delta\bar{\mathcal{J}}_{\max}$. Hence, we choose the number of samples on *one facet* such that this is achieved. Assuming the PF is a unit sphere segment, each of the three boundary segments has length $\frac{1}{4}2\pi$. Thus, we set $N_{\mathrm{facet}} = \mathrm{ceil}(c_{N_{\mathrm{facet}}} \frac{1}{4} 2\pi / \Delta\bar{\mathcal{J}}_{\max})$ with $c_{N_{\mathrm{facet}}} \geq 1$. Then, the total number of boundary samples is $N^{\mathrm{init}}_{\mathrm{boundary}} = 3N_{\mathrm{facet}} - 3$.

Boundary sample solutions may not satisfy (1) with equality. While Subsection 2.4 discusses the existence of a different SO to achieve equality, a similar argument holds for a fixed SO regarding existence of a suitable SDV. Geometrically, any point in the unit box is reachable via a scaled ray from the NP through the simplex, as illustrated in Figure 1 (right).

Assume that a parameter β with corresponding SDV $\check{d} = -\tilde{d}(\beta)$ yields a boundary sample $\bar{\mathcal{J}}(\beta)$ in the normalized image space. To find the corresponding β_{eq} such that (1) holds with equality, we perform a perspective projection of the PF sample on the unit simplex with projection center $\bar{\mathcal{J}}_{\mathrm{UP}}$. This is achieved by $\beta_{\mathrm{eq}} = g^{\mathrm{UP}}_{\beta_{\mathrm{eq}}}(\bar{\mathcal{J}}) := (\bar{\mathcal{J}} - \bar{\mathcal{J}}_{\mathrm{UP}}) / \operatorname{sum}(\bar{\mathcal{J}} - \bar{\mathcal{J}}_{\mathrm{UP}})$ with $\bar{\mathcal{J}} = \bar{\mathcal{J}}(\beta)$. Alternatively, when shooting from the NP, one can derive $\beta_{\mathrm{eq}} = g^{\mathrm{NP}}_{\beta_{\mathrm{eq}}}(\bar{\mathcal{J}}) := (\bar{\mathcal{J}}_{\mathrm{NP}} - \bar{\mathcal{J}}) / \operatorname{sum}(\bar{\mathcal{J}}_{\mathrm{NP}} - \bar{\mathcal{J}})$.

In a similar manner we can perform parallel projections on the hexagons depicted in Figure 1 and obtain $\bar{\mathcal{J}}_{\mathrm{SO}} = g^{\mathrm{UP}}_{\bar{\mathcal{J}}_{\mathrm{SO}}}(\bar{\mathcal{J}}) := \bar{\mathcal{J}} - \frac{\eta^\intercal \bar{\mathcal{J}} + b_{\mathrm{UP}}}{\|\eta\|^2}\eta$ and $\bar{\mathcal{J}}_{\mathrm{SO}} = g^{\mathrm{NP}}_{\bar{\mathcal{J}}_{\mathrm{SO}}}(\bar{\mathcal{J}}) := \bar{\mathcal{J}} - \frac{\eta^\intercal \bar{\mathcal{J}} + b_{\mathrm{NP}}}{\|\eta\|^2}\eta$, where η denotes the plane normal and $b_{\mathrm{UP}}, b_{\mathrm{NP}}$ the corresponding plane offsets.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

Remark 1 (Boundary Refinement). Since not every PF is a unit sphere segment, one cannot expect that the initial number of boundary samples provide the desired maximal distance $\Delta\bar{\mathcal{J}}_{\max}$. While a boundary refinement step had been implemented to handle such cases, it is not described further and is not used in this paper due to space constraints.

4.2 Sampling the Pareto Front Interior

First, the PF boundary samples are projected either perspectively onto the unit simplex or in parallel onto the hexagon plane (cf. Figure 1). On the corresponding plane, the ordered samples form a polygon, which we call the *boundary*. Continuing the examples of Figure 1, the resulting boundaries are shown in Figure 3. Second,

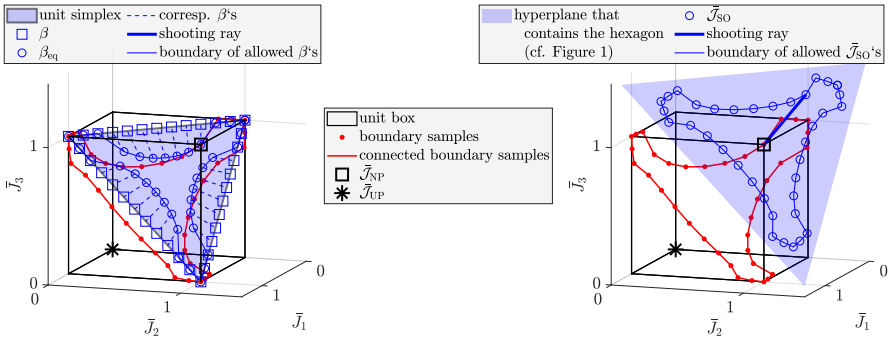

Fig. 3: Left: Perspective projection of boundary samples on the unit simplex; note that the shooting is normal to the viewing plane and is hence not visible. Right: Parallel projection of boundary samples on the plane with normal vector $\eta = \mathbf{1}$ and bias $b = -\eta^\intercal \bar{\mathcal{J}}_{\mathrm{NP}} = -3$; this plane contains the blue hexagon of Figure 1.

uniform samples are generated on the projection plane. Third, the boundary polygon is used to define a rejection rule, as depicted in Figure 4 showing the 2D projection of the continued example from Figure 3. All samples which are not on or in the polygon derived from the projected Pareto front boundary are rejected. A detailed description of the interior sampling can be found in Algorithm 3. Our two algorithms, BRS-radial and BRS-parallel, are composed of Algorithm 2 and Algorithm 3 with variant a) or b).

As the key steps of the overall algorithm, namely projection and rejection, are computationally inexpensive and decoupled from the complexity of the underlying MOOP, the additional overhead introduced by the method is small compared to repeatedly computing redundant PF samples.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

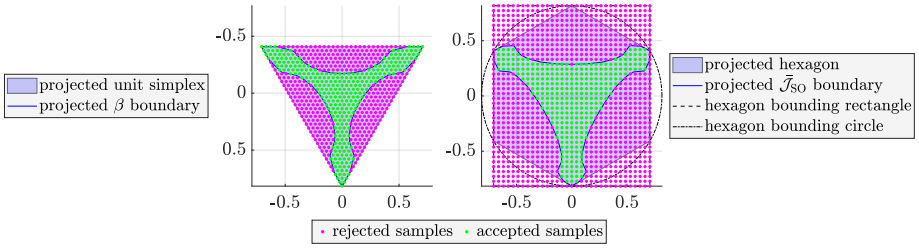

Fig. 4: Idea behind rejection sampling. Quantities related to β (radial shooting) are shown on the left and to $\bar{\mathcal{J}}_{\mathrm{SO}}$ (parallel shooting) on the right.

Algorithm 3 Interior Sampling using a) Radial or b) Parallel Shooting

(0) Initialization
Choose the desired number of total samples $N_{\mathrm{total}}^{\mathrm{des}}$. Set the parameters of $(P(p))$ as shown in (2) and a) $\check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}_{\mathrm{NP}}$, $\check{s}_l = -1$ or b) $\check{d} = -1$, $\check{s}_l = -\mathbf{1}$.

(1) Projection
Use a) $g_{\beta_{\mathrm{eq}}}^{\mathrm{NP}}$ or b) $g_{\bar{\mathcal{J}}_{\mathrm{SO}}}^{\mathrm{NP}}$ to project the boundary samples on the a) unit simplex or b) hexagon. Compute the ratio q of the area of the resulting polygon and the a) unit simplex or b) hexagon.

(2) Rejection sampling
Sample the a) unit simplex uniformly (as e.g. described in [18]) or b) hexagon bounding rectangle uniformly (other choices as e.g. the bounding circle are equally valid) such that at least $\mathrm{ceil}((N_{\mathrm{total}}^{\mathrm{des}} - N_{\mathrm{boundary}})/q)$ points are obtained. Discard all samples that do not lay in the polygon. Let a) $\mathcal{S} = \{\beta_k\}_{k=1}^{N_{\mathrm{int}}}$ be the set of remaining unit simplex samples or b) $\mathcal{S} = \{\bar{\mathcal{J}}_{\mathrm{SO},k}\}_{k=1}^{N_{\mathrm{int}}}$ be the set of remaining bounding rectangle samples.

(3) Interior sample computation
For each a) $\beta \in \mathcal{S}$, set $\check{d} = \check{d}(\beta)$ or b) $\bar{\mathcal{J}}_{\mathrm{SO}} \in \mathcal{S}$, set $\check{\mathcal{J}}_{\mathrm{SO}} = \bar{\mathcal{J}}_{\mathrm{SO}}$ and solve $(P(p))$ to compute an interior sample.

5 Numerical Evaluation

We apply the four algorithms 1) BRS-radial, 2) BRS-parallel, 3) SPO, 4) MSPO to test problems and evaluate results using various metrics. The formulations of the problems are done using CasADi in MATLAB. We use IPOPT [21] running on a laptop with a *12th Gen Intel(R) Core(TM) i7-1260P* processor with 2.10 GHz.

5.1 Test Problems

We consider the six test problems depicted in Table 1: The generic "convex sphere" problem (feasible images are contained in a sphere), four problems from literature and one newly designed problem ("ellipsoid cutout"). Due to the non-unique IMs in test problem 2) we fix them to $\Phi = \sqrt{2}/2 \cdot (1_{3\times 3} - I_3)$ and also provide the NP $\mathcal{J}_{\mathrm{NP}} = \mathbf{1}$. For test problem 5) we provide the NP $\mathcal{J}_{\mathrm{NP}} = (2, 2, 73.5)^\intercal$.

5.2 Evaluation Metrics

To evaluate results objectively, we use five metrics. Three are based on the evenness metric ξ introduced by Messac and Mattson [16]. Calculation of ξ is based on geometric properties of the point distribution in normalized image space.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

1) convex sphere	$\min_{x\in\mathbb{R}^3} \mathcal{J}(x) = x + \mathbf{1} \quad \text{s.t. } \|x\|_2^2 \leq 1$
2) concave sphere [14]	$\min_{x\in\mathbb{R}^2} \mathcal{J}(x) = \begin{pmatrix} \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2) \\ \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2) \\ \sin(\frac{\pi}{2}x_1) \end{pmatrix} \quad \text{s.t. } 0 \leq x \leq 1$
3) bent sphere [18]	$\min_{x\in\mathbb{R}^3} \mathcal{J}(x) = x \quad \text{s.t. } x_1^2 + x_2^2 + x_3 \geq 119,\ x_1x_2x_3 \geq 300,\ 0 \leq x \leq 10$
4) ellipsoid cutout	$\min_{x\in\mathbb{R}^3} \mathcal{J}(x) = \text{diag}(1,10,100)x \quad \text{s.t. } \|x\|_2^2 \leq 1,\ 1 \leq g_{\text{excl,k}}(x),\ k = 1,2,3$
5) comet [4]	$\min_{x\in\mathbb{R}^3} \mathcal{J}(x) = \begin{pmatrix} (1+x_3)(x_1^3x_2^2 - 10x_1 - 4x_2) \\ (1+x_3)(x_1^3x_2^2 - 10x_1 + 4x_2) \\ 3(1+x_3)x_1^2 \end{pmatrix} \quad \begin{matrix} 1 \leq x_1 \leq 3.5 \\ \text{s.t. } -2 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 1 \end{matrix}$
6) OCP [12]	The considered PF is the same as in Figure 6. from [12]; a detailed description on how to construct underlying MOOP can be found in this reference.
Definitions of $g_{\text{excl,k}}(x)$ in "Ellipsoid Cutout" problem: Pairs (i,j) are $(1,2),(2,3),(3,1)$ for $k = 1,2,3$. Let $e_i \in \mathbb{R}^{n_J}$ be the i-th unit vector. Set $\tau = 0.1$ and $b = 0.4$. Construct the anchor points $p_a = -(e_i + \tau(e_j - e_i))$, $p_b = -(e_i + (1-\tau)(e_j - e_i))$. Then we can compute the center $c_k = \frac{p_a+p_b}{2}$, main axis $u = \frac{p_b-p_a}{\|p_b-p_a\|}$, semi-axes $a = \frac{\|p_b-p_a\|}{2}$ and the shape matrix $Q_k = \frac{1}{b^2}I + (\frac{1}{a^2} - \frac{1}{b^2})uu^\intercal$. We can now define $g_{\text{excl,k}}(x) := (x-c_k)^\intercal Q_k (x-c_k)$.	

Table 1: Six test problems considered in our numerical evaluation.

For each point μ^i in the set, two specific circles (or hyperspheres in n dimensions) are constructed. The first is the smallest circle connecting μ^i to any other point, with a diameter denoted as d_l^i. The second is the largest circle connecting μ^i to another point such that the circle contains no additional points from the set; its diameter is denoted as δ_u^i. These diameters are collected into a single set δ. The evenness ξ is then defined as the ratio of the standard deviation σ_d to the mean $\bar{\delta}$ of the set δ, expressed by the formula $\xi = \sigma_\delta/\bar{\delta}$. A value of $\xi = 0$ indicates an exactly even distribution.

The five performance metrics are all designed such that small values are desirable:

1. Sample computation time t_{sample} is the algorithm execution time t_{algo} divided by the number of PF samples: Thereby, t_{sample} consists of time consumed by all auxiliary computations, as e.g. solving linear programs in SPO, and all optimizations.
2. Percentage of redundant interior solutions: First, we check for each interior solution if $|\hat{\mathcal{J}}_{\text{SO}} + l\hat{d} - \bar{\mathcal{J}}(x)| > 10^{-5}$ holds for any component. Then, the number of solutions passing this check is divided by the total number of interior samples.
3. Evenness of all points ξ_{all}.
4. Evenness of the interior points ξ_{int}.
5. Percentage share of local evenness values $\xi_{\text{int,local}}$ that deviate from ξ_{int} by more than 50%. Local neighborhoods are constructed around interior points using a radius of 0.1 in the normalized objective space. Only neighborhoods with more than ten points are considered. For each valid neighborhood, $\xi_{\text{int,local}}$ is computed. The reported share is the proportion of values outside $[0.5 \cdot \xi_{\text{int}}, 1.5 \cdot \xi_{\text{int}}]$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

5.3 Results

For both BRS algorithms we set $\Delta\bar{\mathcal{J}}_{\max} = 0.075$. To allow for a fair comparison, exactly one parameter (SPO D, cf. [18, eq. (7.3)]; MSPO N; BRS $N_{\text{total}}^{\text{des}}$) remains to be chosen for each algorithm. For every problem, it is selected such that the resulting total number of points is slightly larger than and close to 1000. The resulting PF samples are depicted in Figures 5 and 6, and the performance metrics are reported in Table 2.

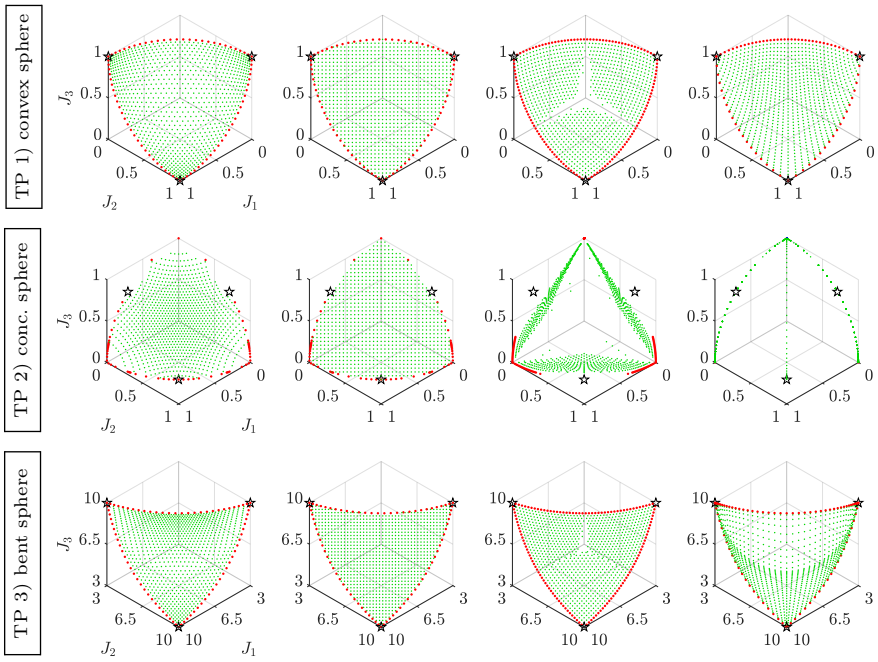

Fig. 5: Top to bottom: different MOOPs; left to right: algorithm 1 to 4. Stars indicate IMs; dot markers: red → boundary samples, green → non-redundant interior samples, blue → redundant interior samples.

For both BRS algorithms, the percentage of redundant interior solutions is always below 1%, even for test problems 5) and 6). Our approach is faster for all test problems except 3), where MSPO is slightly faster but has substantially worse evenness (cf. ξ_{all} and ξ_{int}). Most notably, the average evenness improves by a factor of approximately three, and the share of redundant calculations is reduced by a factor greater than fourteen, while also reducing runtime.

Comparing the two BRS variants, BRS-perspective approach yields better evenness for test problems 1) and 2), but its local interior evenness deviates more strongly from the corresponding global interior evenness. Considering all values

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

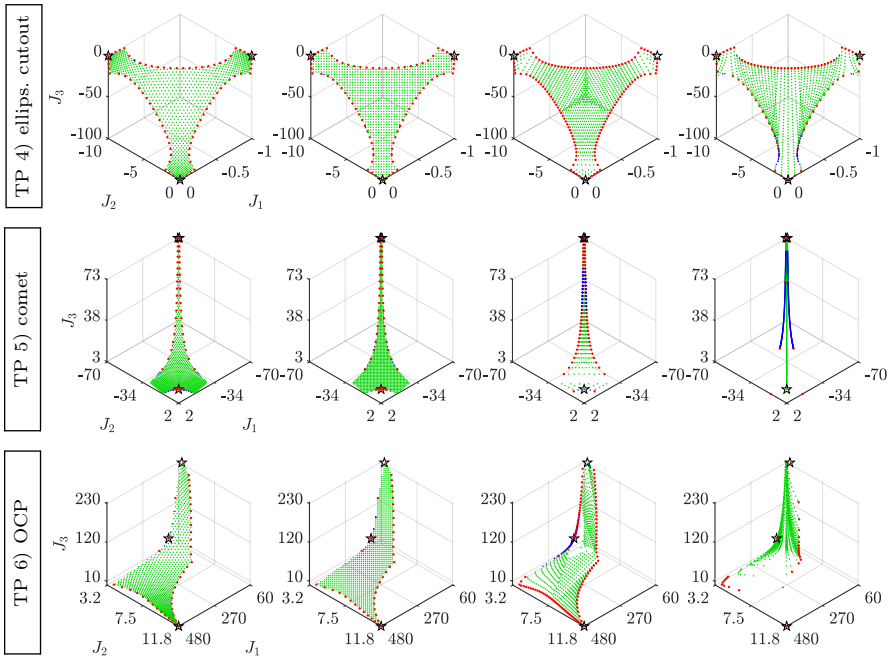

Fig. 6: Top to bottom: different MOOPs; left to right: algorithm 1 to 4. Stars indicate IMs; dot markers: red → boundary samples, green → non-redundant interior samples, blue → redundant interior samples.

in Table 2, BRS-parallel appears to be the more robust choice for approximating a PF in terms of efficiency and uniformity.

6 Conclusion

In this paper, we presented the Boundary-Rejection-Sampling (BRS) algorithm, a novel method for approximating the Pareto front of three-objective optimization problems. By exploiting the boundary information of the feasible objective space, the algorithm generates interior samples through a rejection sampling strategy on projected subspaces. We introduced two variants of the algorithm: BRS-radial and BRS-parallel, which differ in their underlying projection and shooting strategies.

We compared the proposed methods against state-of-the-art algorithms, specifically the Successive Pareto Optimization (SPO) and the Modified Successive Pareto Optimization (MSPO). Our numerical evaluation on a diverse set of test problems demonstrated that the BRS algorithms significantly reduce the number of redundant computations compared to existing methods. Particularly, the BRS-parallel variant proved to be the most robust approach, consistently delivering a high degree of evenness in the distribution of Pareto optimal points while maintaining competitive computational efficiency.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

		TP 1	TP 2	TP 3	TP 4	TP 5	TP 6	⌀
total number of points	BRS-radial	1027	1089	1027	1039	983	1036	
	BRS-parallel	1047	1012	1054	1029	1046	1013	
	SPO	1078	1125	1078	1078	1078	1078	
	MSPO	1034	1083	1026	1076	1378	1028	
t_{sample} **normalized by max.** t_{sample} **of each TP**	BRS-radial	0.523	0.368	0.816	0.842	0.882	**0.835**	0.711
	BRS-parallel	**0.499**	**0.332**	0.753	**0.711**	**0.596**	1.000	**0.648**
	SPO	1.000	0.524	1.000	0.926	1.000	0.907	0.893
	MSPO	0.502	1.000	**0.640**	1.000	0.611	0.845	0.766
redundant interior solutions in %	BRS-radial	**0.000**	**0.000**	**0.000**	0.626	1.507	0.909	0.507
	BRS-parallel	**0.000**	**0.000**	**0.000**	0.633	**1.008**	1.344	**0.498**
	SPO	**0.000**	**0.000**	**0.000**	**0.000**	23.362	23.362	7.787
	MSPO	**0.000**	26.389	**0.000**	6.701	41.673	**0.103**	12.477
ξ_{all}	BRS-radial	**0.192**	0.460	0.311	0.255	0.298	0.362	0.313
	BRS-parallel	0.232	**0.230**	0.245	**0.247**	**0.245**	**0.316**	**0.253**
	SPO	0.285	1.132	**0.159**	0.445	1.765	0.764	0.758
	MSPO	0.343	2.632	0.632	0.594	1.613	1.979	1.299
ξ_{int}	BRS-radial	**0.169**	0.341	0.301	0.211	0.239	0.337	0.266
	BRS-parallel	0.182	**0.181**	0.187	**0.182**	**0.208**	**0.296**	**0.206**
	SPO	0.289	1.141	**0.162**	0.423	1.792	0.753	0.760
	MSPO	0.307	2.628	0.580	0.529	1.306	1.295	1.108
percentage share of $\xi_{\text{int,local}}$ **being 50% above or 50% below** ξ_{int}	BRS-radial	50.211	60.505	41.966	22.651	42.626	61.919	46.647
	BRS-parallel	**12.642**	**8.742**	14.491	**16.878**	**5.847**	**1.656**	**10.043**
	SPO	23.729	89.817	20.507	17.442	78.857	57.370	47.954
	MSPO	14.647	67.460	**5.680**	23.330	17.021	63.246	31.897

Table 2: Performance metrics for all algorithms applied to the test problems.

References

1. Akbari, F., Ghaznavi, M., Khorram, E.: A Revised Pascoletti–Serafini Scalarization Method for Multiobjective Optimization Problems. Journal of Optimization Theory and Applications **178**(2), 560–590 (Aug 2018), https://doi.org/10.1007/s10957-018-1289-2
2. Burachik, R.S., Kaya, C.Y., Rizvi, M.M.: A New Scalarization Technique to Approximate Pareto Fronts of Problems with Disconnected Feasible Sets. Journal of Optimization Theory and Applications **162**(2), 428–446 (Aug 2014), https://doi.org/10.1007/s10957-013-0346-0
3. Burachik, R.S., Kaya, C.Y., Rizvi, M.M.: A New Scalarization Technique and New Algorithms to Generate Pareto Fronts. SIAM Journal on Optimization **27**(2), 1010–1034 (Jan 2017), https://doi.org/10.1137/16M1083967
4. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer-Verlag, London (2005), https://doi.org/10.1007/1-84628-137-7_6
5. Dolatnezhadsomarin, A., Khorram, E.: Two efficient algorithms for constructing almost even approximations of the Pareto front in multi-objective optimization problems. Engineering Optimization **51**(4), 567–589 (Apr 2019), https://doi.org/10.1080/0305215X.2018.1479405
6. Dolatnezhadsomarin, A., Khorram, E., Yousefikhoshbakht, M.: Numerical algorithms for generating an almost even approximation of the Pareto front in nonlinear

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19

multi-objective optimization problems. Applied Soft Computing **165**, 112001 (Nov 2024), https://doi.org/10.1016/j.asoc.2024.112001
7. Ehrgott, M., Ruzika, S.: Improved epsilon-Constraint Method for Multiobjective Programming. Journal of Optimization Theory and Applications **138**(3), 375–396 (Sep 2008), https://doi.org/10.1007/s10957-008-9394-2
8. Ehrgott, M.: Multicriteria Optimization. Springer, Berlin, 2nd ed edn. (2005), https://doi.org/10.1007/3-540-27659-9
9. Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Vector Optimization, Springer Berlin, Berlin Heidelberg (2008), https://doi.org/10.1007/978-3-540-79159-1
10. Eichfelder, G.: An Adaptive Scalarization Method in Multiobjective Optimization. SIAM Journal on Optimization **19**(4), 1694–1718 (Jan 2009), https://doi.org/10.1137/060672029
11. Herrmann-Wicklmayr, M., Flaßkamp, K.: Accelerating Multi-Objective Model Predictive Control Using High-Order Sensitivity Information. In: Berberich, J., Iannelli, A., Allgöwer, F. (eds.) Systems Theory in Data and Optimization. pp. 215–230. Springer Nature Switzerland, Cham (Oct 2025), https://doi.org/10.1007/978-3-031-83191-1_14
12. Herrmann-Wicklmayr, M., Flaßkamp, K.: Individual Minima-Informed Multi-Objective Model Predictive Control for Fixed Point Stabilization (Oct 2025), http://arxiv.org/abs/2510.23454
13. Kaya, C.Y., Maurer, H.: A numerical method for nonconvex multi-objective optimal control problems. Computational Optimization and Applications **57**(3), 685–702 (Apr 2014), https://doi.org/10.1007/s10589-013-9603-2
14. Khaledian, K., Soleimani-damaneh, M.: A new approach to approximate the bounded Pareto front. Mathematical Methods of Operations Research **82**(2), 211–228 (Oct 2015), https://doi.org/10.1007/s00186-015-0510-4
15. Khorram, E., Khaledian, K., Khaledyan, M.: A numerical method for constructing the Pareto front of multi-objective optimization problems. Journal of Computational and Applied Mathematics **261**, 158–171 (May 2014), https://doi.org/10.1016/j.cam.2013.11.007
16. Messac, A., Mattson, C.A.: Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. AIAA Journal **42**(10), 2101–2111 (Oct 2004), https://doi.org/10.2514/1.8977
17. Morovati, V., Pourkarimi, L.: Extension of Zoutendijk method for solving constrained multiobjective optimization problems. European Journal of Operational Research **273**(1), 44–57 (Feb 2019), https://doi.org/10.1016/j.ejor.2018.08.018
18. Mueller-Gritschneder, D., Graeb, H., Schlichtmann, U.: A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems. SIAM Journal on Optimization **20**(2), 915–934 (Jan 2009), https://doi.org/10.1137/080729013
19. Pascoletti, A., Serafini, P.: Scalarizing Vector Optimization Problems. Journal of Optimization Theory and Applications **42**(4), 499–524 (Apr 1984), https://doi.org/10.1007/BF00934564
20. Stieler, M.: Performance Estimates for Scalar and Multiobjective Model Predictive Control Schemes. Ph.D. thesis (Mar 2018), https://nbn-resolving.org/urn:nbn:de:bvb:703-epub-3783-4
21. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (Mar 2006), https://doi.org/10.1007/s10107-004-0559-y

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_19