Exploring the use of Polynomial Models in the Search Step of BoostDMS

Marta Pozzi[1], Everton Silva[2], Ana Luísa Custódio[2]

[1]Bocconi University [2]NOVA Math, NOVA FCT

Presentation Outline

Problem Features

Multiobjective Derivative-free Optimization

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} F(x) \equiv (f_1(x), f_2(x), \ldots, f_m(x))^\top$$

$$f_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, \; j = 1, \ldots, m \geq 2$$

Problem Features

Multiobjective Derivative-free Optimization

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} F(x) \equiv (f_1(x), f_2(x), \ldots, f_m(x))^\top$$

$$f_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, \; j = 1, \ldots, m \geq 2$$

- several objectives, often conflicting

Problem Features

Multiobjective Derivative-free Optimization

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} F(x) \equiv (f_1(x), f_2(x), \ldots, f_m(x))^\top$$

$$f_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, \; j = 1, \ldots, m \geq 2$$

- several objectives, often conflicting
- expensive function evaluation, possibly subject to noise
- impossible to use or approximate derivatives

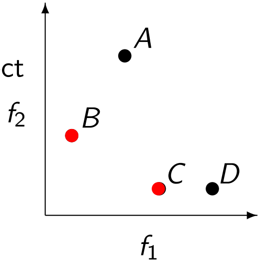
Problem Features

Multiobjective Derivative-free Optimization

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} F(x) \equiv (f_1(x), f_2(x), \ldots, f_m(x))^\top$$

$$f_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, \; j = 1, \ldots, m \geq 2$$

- several objectives, often conflicting
- expensive function evaluation, possibly subject to noise
- impossible to use or approximate derivatives
- make use of Pareto dominance

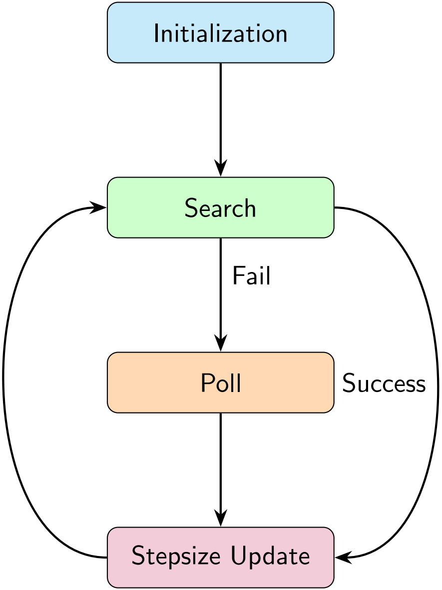


Pareto Front: $\{B, C\}$

Pareto Dominance (x dominates y)

$F(x) \leq F(y)$, with $F(x) \neq F(y)$

BoostDMS: General Framework



- Makes use of the search/poll paradigm
- The search step is used for **exploration**
- **Convergence** is assured by the polling procedure
- Keeps a list of feasible nondominated points
- Poll centers are chosen from the list
- **Successful iterations** correspond to list changes

Polynomial Interpolation $f : \mathbb{R}^n \rightarrow \mathbb{R}$

Polynomial Model

$m(y) = \alpha^\top \phi(y)$ where ϕ is a polynomial basis

Polynomial Interpolation $f : \mathbb{R}^n \to \mathbb{R}$

Polynomial Model

$m(y) = \alpha^\top \phi(y)$ where ϕ is a polynomial basis

Given a sample set $Y = \{y^0, y^1, \ldots, y^p\}$, the model coefficients result from the linear system:

$$M(\phi, Y)\alpha \;=\; f(Y),$$

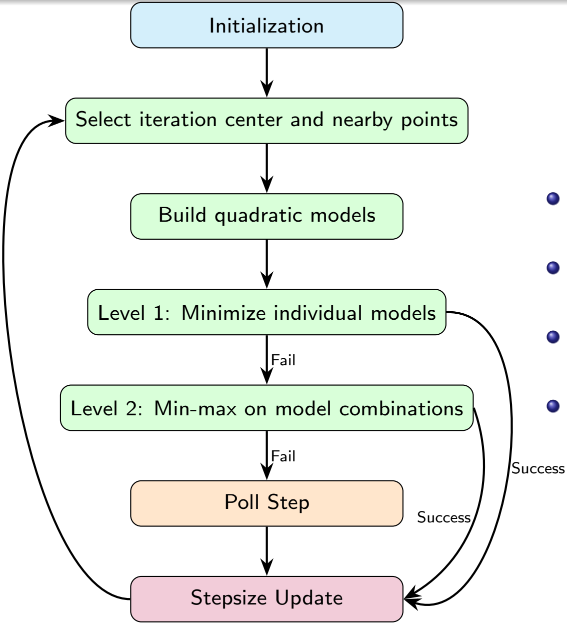
where

$$M(\phi, Y) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_q(y^0) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_q(y^p) \end{bmatrix} \quad f(Y) = \begin{bmatrix} f(y^0) \\ \vdots \\ f(y^p) \end{bmatrix}$$

We use the natural basis of monomials, which in 2D is

$$\phi \;=\; \{1, x_1, x_2, x_1^2/2, x_2^2/2, x_1 x_2\}$$

Model-Based Search Step in BoostDMS



- Select points from a cache within a trust region
- Build local quadratic models per objective
- **Level 1:** Minimize each model separately
- **Level 2:** Use min-max scalarization over combinations of models

Search Step: What's new?

We modified and tested several approaches for the search step:

Search Step: What's new?

We modified and tested several approaches for the search step:

- Normal Boundary Intersection (NBI)
- Epsilon-constraint - adapted

Search Step: What's new?

We modified and tested several approaches for the search step:

- Normal Boundary Intersection (NBI)
- Epsilon-constraint - adapted
- Improved Front Steepest Descent IFSD

Performance Assessment

Performance Assessment

Metrics

- **Hypervolume:** Measures the volume of the dominated region
- **Purity:** Fraction of points that are non-dominated
- **Gamma Spread:** Assesses the largest gap in the approximation of the Pareto front
- **Delta Spread:** Evaluates how evenly distributed the points are along the approximation of the Pareto front

Performance Assessment

Metrics

- **Hypervolume:** Measures the volume of the dominated region
- **Purity:** Fraction of points that are non-dominated
- **Gamma Spread:** Assesses the largest gap in the approximation of the Pareto front
- **Delta Spread:** Evaluates how evenly distributed the points are along the approximation of the Pareto front

Performance Profiles

Cumulative distribution plots, comparing solver performance over a test set. Higher curves indicate better performances.

Performance Assessment

Metrics

- **Hypervolume:** Measures the volume of the dominated region
- **Purity:** Fraction of points that are non-dominated
- **Gamma Spread:** Assesses the largest gap in the approximation of the Pareto front
- **Delta Spread:** Evaluates how evenly distributed the points are along the approximation of the Pareto front

Performance Profiles

Cumulative distribution plots, comparing solver performance over a test set. Higher curves indicate better performances.

Test Set

A comprehensive benchmark of **100 bound-constrained problems** used to assess performance across diverse objective landscapes and difficulty levels.

NBI Method

Mathematical Formulation

F: vector of objectives

F^* : vector of the individual minima

NBI problem:

$$\begin{array}{cl} \max\limits_{x\in\mathbb{R}^n,\ t\in\mathbb{R}} & t \\ \text{subject to} & F(x) - F^* = \Phi\beta + t\cdot\mathbf{n} \\ & x \in \Omega \end{array}$$

where:

- Φ with columns $F(x_i^*) - F^*$ where $x_i^* = argmin_{x\in\Omega}(f_i(x))$
- $\beta \in \mathbb{R}^m$: weights, $\sum \beta_i = 1$, $\beta_i \geq 0$
- $\mathbf{n}$: unit normal vector to the convex hull of the individual minima

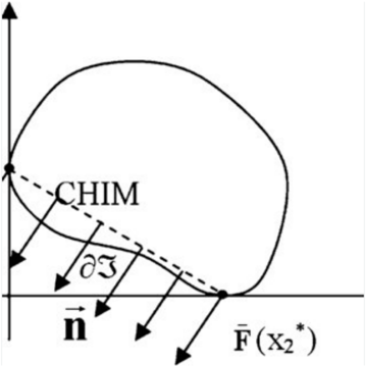


Illustration of CHIM and normal projection

NBI Method in DMS

Context

- The point with the **largest gap** in the Pareto front is selected
- **Quadratic models** are built at this point
- These models are used in the NBI formulation

NBI Method in DMS

Context

- The point with the **largest gap** in the Pareto front is selected
- **Quadratic models** are built at this point
- These models are used in the NBI formulation

Procedure

1. **Extreme points:** Minimize each model to get x_i^* and the *ideal point* F^*
2. **Payoff matrix:** Evaluate all models at x_i^* to build the matrix Φ
3. **CHIM hyperplane:** Define $\Phi\beta = (F(x_i^*) - F^*)\beta$, for $\beta \in \Delta$.
4. **Normal direction:** Compute a normal vector $\mathbf{n}$ to CHIM
5. **Solve NBI modified subproblems**

$$\begin{array}{cl} \max\limits_{x\in\Omega,\ t\in\mathbb{R}} & t \\ \text{subject to} & F(x) - F^* \leq \Phi\beta + t \cdot \mathbf{n} \end{array}$$

NBI Method in DMS

Context

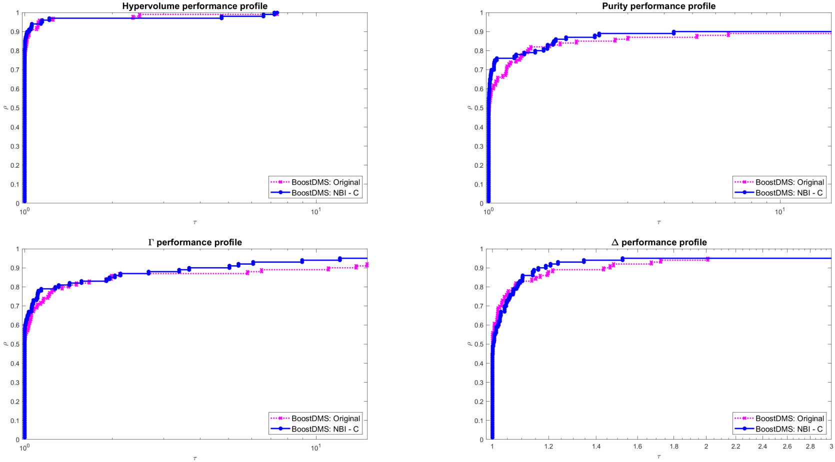
- The point with the **largest gap** in the Pareto front is selected
- **Quadratic models** are built at this point
- These models are used in the NBI formulation

Procedure

1. **Extreme points:** Minimize each model to get x_i^* and the *ideal point* F^*
2. **Payoff matrix:** Evaluate all models at x_i^* to build the matrix Φ
3. **CHIM hyperplane:** Define $\Phi\beta = (F(x_i^*) - F^*)\beta$, for $\beta \in \Delta$.
4. **Normal direction:** Compute a normal vector $\mathbf{n}$ to CHIM
5. **Solve NBI modified subproblems**

$$\begin{array}{rl} \max\limits_{x\in\Omega,\ t\in\mathbb{R}} & t \\ \text{subject to} & F(x) - F^* \leq \Phi\beta + t \cdot \mathbf{n} \end{array}$$

Numerical Results



BoostDMS Original

BoostDMS NBI

The Epsilon-constraint Method

Mathematical Formulation

For $k = 1, ..., m$

$$\begin{aligned} \min_{x \in \Omega} \quad & f_k(x) \\ \text{s.t.} \quad & f_j(x) \leq \varepsilon_j, \quad \forall j = 1, \ldots, m, \quad j \neq k \end{aligned}$$

The Epsilon-constraint Method

Mathematical Formulation

For $k = 1, ..., m$

$$\begin{aligned} \min_{x \in \Omega} \quad & f_k(x) \\ \text{s.t.} \quad & f_j(x) \leq \varepsilon_j, \quad \forall j = 1, \ldots, m, \quad j \neq k \end{aligned}$$

Description

Fix upper bounds ε_j on all objectives except one, and minimize that single objective within these bounds

The Epsilon-constraint Method

Mathematical Formulation

For $k = 1, ..., m$

$$\begin{aligned} \min_{x\in\Omega} \quad & f_k(x) \\ \text{s.t.} \quad & f_j(x) \leq \varepsilon_j, \quad \forall j = 1, \ldots, m, \quad j \neq k \end{aligned}$$

Description

Fix upper bounds ε_j on all objectives except one, and minimize that single objective within these bounds

Intuition

By varying ε_j, different Pareto optimal points are found, effectively exploring the Pareto front

Epsilon-constraint Method in DMS

Context in the Search Step:

Epsilon-constraint method:

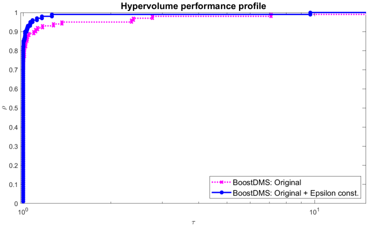
For $k = 1, \ldots, m$

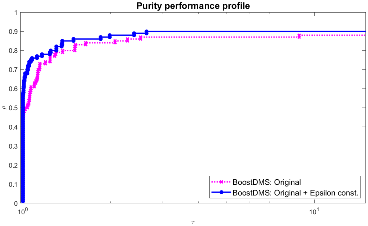
$$\begin{aligned} \min_{h} \quad & m_k(x_0 + h) \\ \text{s.t.} \quad & m_j(x_0 + h) \leq \varepsilon_j = f_j(x_0), \quad j \neq i \\ & x + h \in \Omega, \quad \|h\| \leq \Delta \end{aligned}$$

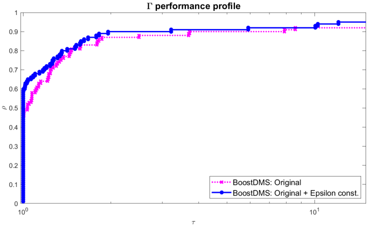
Key points:

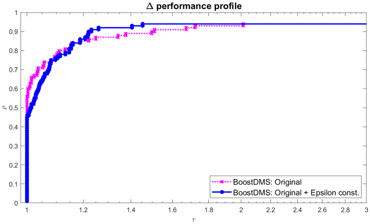
- ε_j fixed at current objective values: $\varepsilon_j = f_j(x_0)$
- Focuses on improving one objective while constraining the others
- Ensures local search within the trust region radius Δ

Numerical Results









BoostDMS Original **BoostDMS Epsilon**

Improved Front Steepest Descent (IFSD): Intuition

Goal

Efficiently generate a set of nondominated solutions by moving a front of points in the decision space

Core Idea

- Start from a set of points approximating the Pareto front
- For each point, try to improve all objectives simultaneously using a steepest descent-like direction
- If joint improvement is not possible, identify promising subsets of objectives and move accordingly
- After each move, update the front by removing dominated solutions

Derivatives are needed to perform the descent step

Improved Front Steepest Descent (IFSD): Algorithm

Step 1: Common Descent Direction

Given x^k, solve:

$$d^k = \arg\min_{d \in \mathbb{R}^n} \max_{j=1,\dots,m} \nabla f_j(x^k)^T d + \frac{1}{2}\|d\|^2$$

If $d^k \neq 0$, project the point in the bound and perform a line search along d^k: $x^{k+1} = x^k + \alpha d^k$

Improved Front Steepest Descent (IFSD): Algorithm

Step 1: Common Descent Direction

Given x^k, solve:

$$d^k = \arg\min_{d\in\mathbb{R}^n} \max_{j=1,\dots,m} \nabla f_j(x^k)^T d + \frac{1}{2}\|d\|^2$$

If $d^k \neq 0$, project the point in the bound and perform a line search along d^k: $x^{k+1} = x^k + \alpha d^k$

Step 2: Partial Descent Directions

For each proper subset $\ell \subset \{1,\dots,m\}$, solve:

$$d^\ell = \arg\min_{d\in\mathbb{R}^n} \max_{j\in I} \nabla f_j(x^{k+1})^T d + \frac{1}{2}\|d\|^2$$

If $d^\ell \neq 0$, project the point in the bound and perform a line search and update: $x^{k+1} \leftarrow x^{k+1} + \alpha d^\ell$

IFSD in DMS

How IFSD is used

- The IFSD algorithm is applied for the multiobjective optimization of the models
- It generates a set of candidates points to improve the Pareto front
- Only the candidates in the trust region are kept

IFSD in DMS

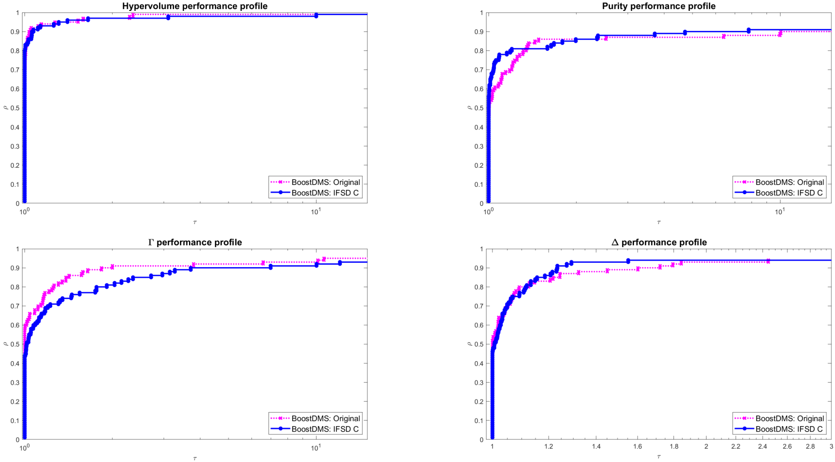
How IFSD is used

- The IFSD algorithm is applied for the multiobjective optimization of the models
- It generates a set of candidates points to improve the Pareto front
- Only the candidates in the trust region are kept

Filtering and Selection

- We evaluate which candidates points are **nondominated** concerning the current model predictions
- Among the nondominated candidates, we select up to $2^m - 1$ points (where m is the number of objectives) based on the crowding distance

Numerical Results



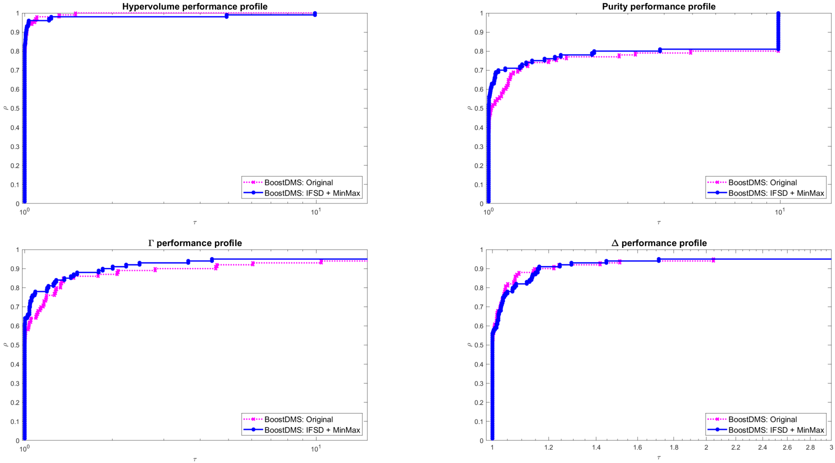
BoostDMS Original **BoostDMS IFSD**

Combined Approaches

We show the effect of combining some of the previous methods:

- We introduce the condition number as a metric to assess the accuracy of the models
- We set the condition number to 1000
- If the model is trustable, we use the IFSD search
- Otherwise, we use a scalarization approach
  - MinMax
  - Epsilon-constraint method

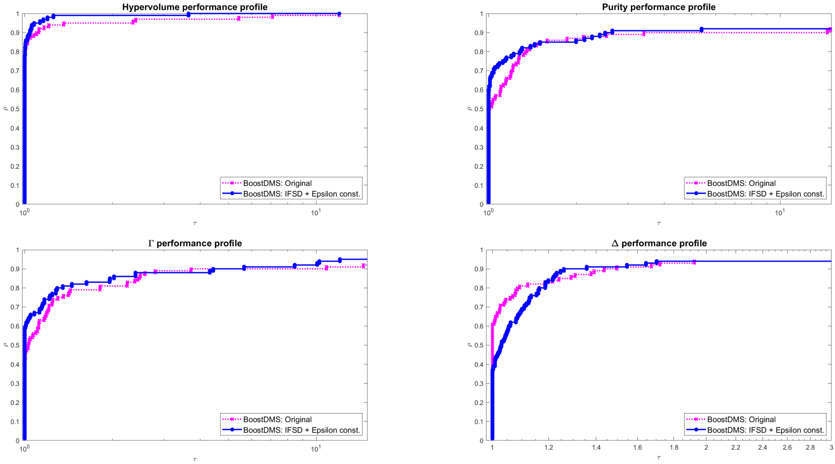
Numerical Results: IFSD + MinMax



BoostDMS Original

BoostDMS IFSD + MinMax

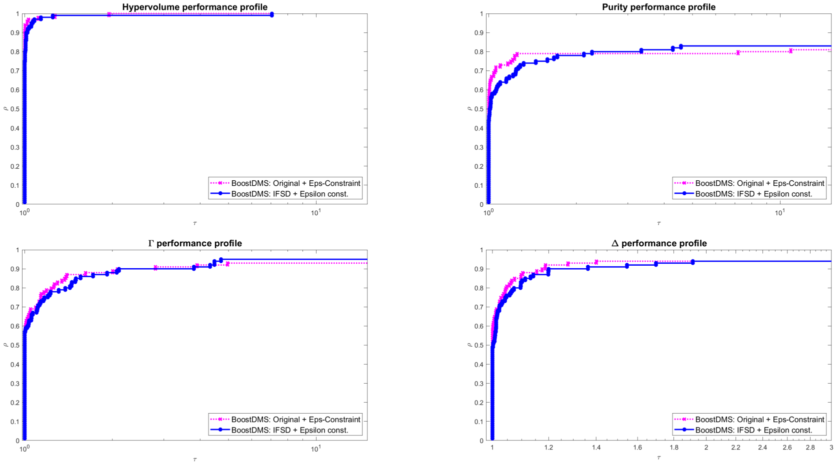
Numerical Results: IFSD + Epsilon-constraint Method



BoostDMS Original

BoostDMS IFSD + Epsilon-constraint

Comparison of the Best Solvers



BoostDMS + Epsilon Constraint

BoostDMS IFSD + Epsilon constraint

Conclusions

- There are advantages in the use of the Epsilon-constraint scalarization, both combined with the original approach, and with the IFSD method

Conclusions

- There are advantages in the use of the Epsilon-constraint scalarization, both combined with the original approach, and with the IFSD method
- There is a small advantage of the Original + Epsilon-constraint over the IFSD + Epsilon-constraint method

Conclusions

- There are advantages in the use of the Epsilon-constraint scalarization, both combined with the original approach, and with the IFSD method
- There is a small advantage of the Original + Epsilon-constraint over the IFSD + Epsilon-constraint method
- Running the IFSD algorithm is costly in terms of computational time

Conclusions

- There are advantages in the use of the Epsilon-constraint scalarization, both combined with the original approach, and with the IFSD method
- There is a small advantage of the Original + Epsilon-constraint over the IFSD + Epsilon-constraint method
- Running the IFSD algorithm is costly in terms of computational time

Given the previous considerations, we conclude that the most competitive version of the algorithm uses the original approach, substituting the MinMax scalarization with the Epsilon-constraint method.

THANK YOU FOR YOUR ATTENTION!