Polytechnique Montréal
Department of Mathematics and Industrial Engineering

A progressive barrier for blackbox/derivative-free multiobjective optimization : very preliminary results

Ludovic Salomon
Jean Bigeon
Sébastien Le Digabel

Before starting

Warnings !!

As the title indicates, this work is under development.

The problem

Multiobjective optimization problem

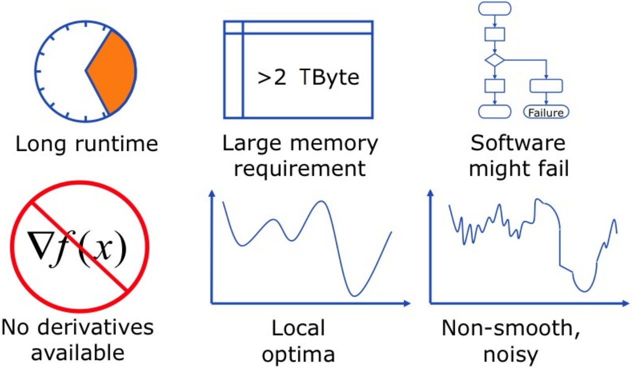
$$\min_{x \in \Omega} f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^\top$$

where

- $\Omega = \{x \in X : c_i(x) \leq 0, \ \forall i \in \mathcal{I}\} \subset \mathbb{R}^n$ is the *feasible set*.
- $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ for $i = 1, 2, \ldots, m$, $m \geq 2$ are *objective functions*.
- $c_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ for $i \in \mathcal{I}$ are *relaxable constraints*.

The f_i for $i = 1, 2, \ldots, m$ and c_i for $i \in \mathcal{I}$ are supposed to be blackboxes.

What is a blackbox ?



Pareto dominance

Given two vectors z^1 and z^2 in the objective space, we write that:
$z^1 \leq z^2 \iff z^2 - z^1 \in \mathbb{R}^m_+ \iff \forall i = 1, 2, \dots, m\,, z^1_i \leq z^2_i.$

Given two decision vectors x^1 and x^2, we write that:

- $x^1 \preccurlyeq x^2$ (x^1 weakly dominates x^2) if and only if $f(x^1) \leq f(x^2)$. ex: $f(x^1) = [-1, 1]^\top$, $f(x^2) = [1, 1]^\top$.
- $x^1 \prec x^2$ (x^1 dominates x^2) if and only if $x^1 \preccurlyeq x^2$ and at least one objective is strictly better than another. ex: $f(x^1) = [-1, 0]^\top$, $f(x^2) = [1, 2]^\top$.
- $x^1 \parallel x^2$ (x^1 and x^2 are incomparable) if neither x^1 weakly dominates x^2 nor x^2 weakly dominates x^1. ex: $f(x^1) = [-1, 0]^\top$, $f(x^2) = [-2, 1]^\top$.

Pareto dominance

$x \in \Omega$ is said to be Pareto-optimal if there is no other vector in Ω that dominates it. The set of Pareto-optimal solutions (decision variables) is called the Pareto set and the image of the Pareto set is called the Pareto front.

Pareto dominance

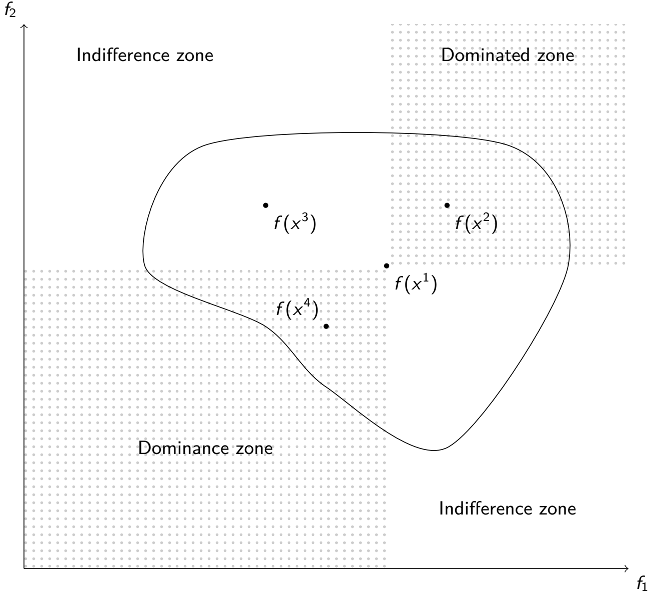
Given two vectors z^1 and z^2 in the objective space, we write that:
$z^1 \leq z^2 \iff z^2 - z^1 \in \mathbb{R}^m_+ \iff \forall i = 1, 2, \ldots, m \,, z^1_i \leq z^2_i$.

Given two decision vectors x^1 and x^2, we write that:

- $x^1 \preccurlyeq x^2$ (x^1 weakly dominates x^2) if and only if $f(x^1) \leq f(x^2)$. ex: $f(x^1) = [-1, 1]^\top$, $f(x^2) = [1, 1]^\top$.
- $x^1 \prec x^2$ (x^1 dominates x^2) if and only if $x^1 \preccurlyeq x^2$ and at least one objective is strictly better than another. ex: $f(x^1) = [-1, 0]^\top$, $f(x^2) = [1, 2]^\top$.
- $x^1 \parallel x^2$ (x^1 and x^2 are incomparable) if neither x^1 weakly dominates x^2 nor x^2 weakly dominates x^1. ex: $f(x^1) = [-1, 0]^\top$, $f(x^2) = [-2, 1]^\top$.

Pareto dominance

$x \in \Omega$ is said to be Pareto-optimal if there is no other vector in Ω that dominates it. The set of Pareto-optimal solutions (decision variables) is called the Pareto set and the image of the Pareto set is called the Pareto front.



An illustration of Pareto dominance for a minimization biobjective problem.

Previously, in mesh adaptive direct search algorithms for multiobjective optimization

We proposed the DMulti-MADS algorithm which:

- is strongly inspired by Direct MultiSearch (DMS) [Custódio et al., 2011] and BiMADS [Audet et al., 2008].
- Handles more than 2 objectives, contrary to BiMADS.
- Converges to a set of locally optimal Pareto points contrary to DMS, under mild assumptions.
- Does not aggregate any of the objective functions.
- Practically, it is competitive according to other state-of-the-art algorithms (NSGAII [Deb et al., 2000], DMS, MOIF [Cocchi et al., 2018], BiMADS).

The question

DMulti-MADS deals with general constraints via an extreme barrier approach, i.e.

$$f_\Omega(x) = \begin{cases} [+\infty, +\infty, \ldots, +\infty] & \text{if } x \notin \Omega \\ f(x) & \text{otherwise.} \end{cases}$$

Could we exploit relaxable constraints c_i, $\forall i \in \mathcal{I}$?

Previously, in mesh adaptive direct search algorithms for multiobjective optimization

We proposed the DMulti-MADS algorithm which:

- is strongly inspired by Direct MultiSearch (DMS) [Custódio et al., 2011] and BiMADS [Audet et al., 2008].
- Handles more than 2 objectives, contrary to BiMADS.
- Converges to a set of locally optimal Pareto points contrary to DMS, under mild assumptions.
- Does not aggregate any of the objective functions.
- Practically, it is competitive according to other state-of-the-art algorithms (NSGAII [Deb et al., 2000], DMS, MOIF [Cocchi et al., 2018], BiMADS).

The question

DMulti-MADS deals with general constraints via an extreme barrier approach, i.e.

$$f_\Omega(x) = \begin{cases} [+\infty, +\infty, \ldots, +\infty] & \text{if } x \notin \Omega \\ f(x) & \text{otherwise.} \end{cases}$$

Could we exploit relaxable constraints c_i, $\forall i \in \mathcal{I}$?

What has the derivative-free/blackbox literature ever done for us ?

- In single-objective convergent-based derivative-free methods:

With derivatives	Without derivatives
Augmented Lagrangian [Nocedal and Wright, 2006]	[Lewis et al., 2006]
Filter approach [Fletcher and Leyffer, 2002]	[Audet and Dennis, 2009]
Merit function [Nocedal and Wright, 2006]	[Gratton and Vicente, 2014]
Penalty function [Nocedal and Wright, 2006]	[Liuzzi and Lucidi, 2009]

- In multiobjective convergent-based derivative-free methods:
  - Scalarization based approaches [Audet et al., 2008, Audet et al., 2010]
  - Merit function based-approaches [Liuzzi et al., 2016]

What has the derivative-free/blackbox literature ever done for us ?

- In single-objective convergent-based derivative-free methods:

With derivatives	Without derivatives
Augmented Lagrangian [Nocedal and Wright, 2006]	[Lewis et al., 2006]
Filter approach [Fletcher and Leyffer, 2002]	[Audet and Dennis, 2009]
Merit function [Nocedal and Wright, 2006]	[Gratton and Vicente, 2014]
Penalty function [Nocedal and Wright, 2006]	[Liuzzi and Lucidi, 2009]

- In multiobjective convergent-based derivative-free methods:
  - Scalarization based approaches [Audet et al., 2008, Audet et al., 2010]
  - Merit function based-approaches [Liuzzi et al., 2016]

The PB-MADS algorithm for constrained single-objective optimization

- At each iteration, MADS attempts to find better points on the mesh

$$M^k = \{x^k + \delta^k Dy : y \in \mathbb{N}^p\} \subset \mathbb{R}^n$$

  where:
  - x^k is the current incumbent solution.
  - $\delta^k > 0$ is the mesh size parameter.
  - $D = GZ$ with $G \in \mathbb{R}^{n \times n}$ inversible and $Z \in \mathbb{Z}^{n \times p}$ such that the columns of Z form a positive spanning set of $\mathbb{R}^n$.
- Each iteration is organized around a poll and a search step.
- Constraints are aggregated using the constraint violation function

$$h(x) = \begin{cases} \sum_{i \in \mathcal{I}} \max\{c_i(x), 0\}^2 & \text{if } x \in X; \\ +\infty & \text{otherwise.} \end{cases}$$

- PB-MADS considered the constrained single objective optimization problem as a biobjective one, where f and h are the two objective functions to optimize.
- Use a threshold value $h^k_{\max}$ to reject not-promising points.
- Towards the iterations, $h^k_{\max}$ is reduced.

The PB-MADS algorithm for constrained single-objective optimization

- At each iteration, MADS attempts to find better points on the mesh

$$M^k = \{x^k + \delta^k D y : y \in \mathbb{N}^p\} \subset \mathbb{R}^n$$

where:

  - x^k is the current incumbent solution.
  - $\delta^k > 0$ is the mesh size parameter.
  - $D = GZ$ with $G \in \mathbb{R}^{n \times n}$ inversible and $Z \in \mathbb{Z}^{n \times p}$ such that the columns of Z form a positive spanning set of $\mathbb{R}^n$.

- Each iteration is organized around a poll and a search step.
- Constraints are aggregated using the constraint violation function

$$h(x) = \begin{cases} \sum_{i \in \mathcal{I}} \max\{c_i(x), 0\}^2 & \text{if } x \in X; \\ +\infty & \text{otherwise.} \end{cases}$$

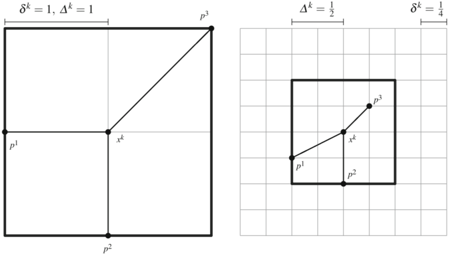
- PB-MADS considered the constrained single objective optimization problem as a biobjective one, where f and h are the two objective functions to optimize.
- Use a threshold value $h^k_{\max}$ to reject not-promising points.
- Towards the iterations, $h^k_{\max}$ is reduced.

The poll I

- The convergence analysis is based on the poll step, where one evaluates the following candidates:

$$P^k = \{x^k + \delta^k d : d \in \mathbb{D}_\Delta^k\} \subset F^k$$

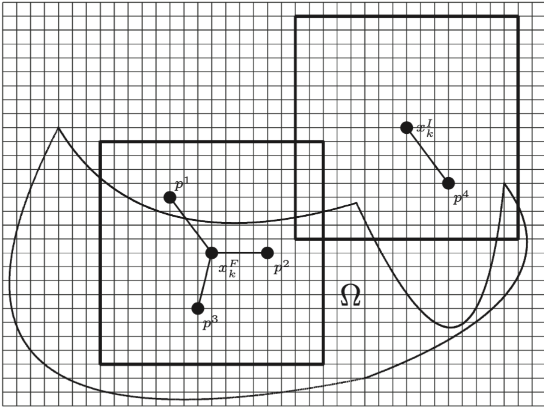
where $\mathbb{D}_\Delta^k$ is a positive spanning set of directions and F^k is the frame centered at x^k of frame size $\Delta^k > 0$.



Example of frames and meshes in $\mathbb{R}^2$ [Audet and Hare, 2017]

The poll II

- The PB-Mads algorithm allows the evaluations of poll candidates around two poll centers, the best feasible one x_F^k and the best infeasible one x_I^k.



Example of poll sets for PB-MADS in $\mathbb{R}^2$ [Audet and Dennis, 2009]

Functioning of the PB-DMulti-MADS algorithm : preliminaries

Constraint violation function

Given the multiobjective optimization problem

$$\min_{x \in \Omega} f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^\top$$

where

$$\Omega = \{x \in X : c_i(x) \leq 0, \ \forall i \in \mathcal{I}\} \subset \mathbb{R}^n,$$

the constraint violation function [Audet and Dennis, 2009] is defined as

$$h(x) = \begin{cases} \sum_{i \in \mathcal{I}} \max\{c_i(x), 0\}^2 & \text{if } x \in X; \\ +\infty & \text{otherwise.} \end{cases}$$

Definition (Extension of the dominance relation for constrained optimization [Audet and Dennis, 2009])

$y \in X$ is said to dominate $x \in X$ if

- Both points are feasible and $y \in \Omega$ dominates $x \in \Omega$, denoted by $y \prec_f x$.
- Both points are infeasible, $f(y) \leq f(x)$ and $h(y) \leq h(x)$ with at least one inequality strictly verified, denoted by $y \prec_h x$.

Functioning of the PB-DMulti-MADS algorithm : preliminaries

Constraint violation function

Given the multiobjective optimization problem

$$\min_{x \in \Omega} f(x) = [f_1(x), f_2(x), \dots, f_m(x)]^\top$$

where

$$\Omega = \{x \in X : c_i(x) \leq 0, \ \forall i \in \mathcal{I}\} \subset \mathbb{R}^n,$$

the constraint violation function [Audet and Dennis, 2009] is defined as

$$h(x) = \begin{cases} \sum_{i \in \mathcal{I}} \max\{c_i(x), 0\}^2 & \text{if } x \in X; \\ +\infty & \text{otherwise.} \end{cases}$$

Definition (Extension of the dominance relation for constrained optimization [Audet and Dennis, 2009])

$y \in X$ is said to dominate $x \in X$ if

- Both points are feasible and $y \in \Omega$ dominates $x \in \Omega$, denoted by $y \prec_f x$.
- Both points are infeasible, $f(y) \leq f(x)$ and $h(y) \leq h(x)$ with at least one inequality strictly verified, denoted by $y \prec_h x$.

Functioning of the PB-DMulti-MADS algorithm: main characteristics

PB-DMulti-MADS

- As in single-objective optimization, is built around a poll and a search.
- Similarly to DMS, keeps a list of non-dominated feasible points called an iterate list L^k

$$L^k = \{(x^j, \Delta^j), x^j \in \Omega, \Delta^j > 0, j = 1, 2, \ldots, |L^k|\}.$$

- Similarly to DMS, choice of the poll center (x^k, Δ^k) among the elements of L^k.

Furthermore,

- Implements a filter-based approach based on the constraint violation function h as for PB-Mads.
- Keep a list of non-feasible incumbent points

$$I^k = \arg\min_{x \in U^k} \{f(x) : 0 < h(x) \leq h^k_{\max}\}$$

where U^k is the set of infeasible non-dominated points and $h^k_{\max}$ a barrier threshold updated at each iteration.

- As k increases, the barrier threshold $h^k_{\max}$ is reduced.

Functioning of the PB-DMulti-MADS algorithm: main characteristics

PB-DMulti-MADS

- As in single-objective optimization, is built around a poll and a search.
- Similarly to DMS, keeps a list of non-dominated feasible points called an iterate list L^k

$$L^k = \{(x^j, \Delta^j), x^j \in \Omega, \Delta^j > 0, j = 1, 2, \ldots, |L^k|\}.$$

- Similarly to DMS, choice of the poll center (x^k, Δ^k) among the elements of L^k.

Furthermore,

- Implements a filter-based approach based on the constraint violation function h as for PB-Mads.
- Keep a list of non-feasible incumbent points

$$I^k = \arg \min_{x \in U^k} \{f(x) : 0 < h(x) \leq h^k_{\max}\}$$

where U^k is the set of infeasible non-dominated points and $h^k_{\max}$ a barrier threshold updated at each iteration.

- As k increases, the barrier threshold $h^k_{\max}$ is reduced.

Functioning of the PB-DMulti-MADS algorithm: the poll step

As for the single-objective case, two poll centers are considered:

- The feasible poll center x_F^k (if it exists) must satisfy:

$$(x_F^k, \Delta^k) \in \arg \max_{(x^j, \Delta^j) \in L^k} \Delta^j.$$

- The infeasible poll center x_I^k (if it exists) must satisfy:

$$(x_I^k, \Delta^k) \in \arg \max_{x \in I^k} h(x).$$

- The optional poll center $x_{opt}^k \in I^k$ (dependent on x_F^k).

Functioning of the PB-DMulti-MADS algorithm: the poll step

As for the single-objective case, three poll centers are considered:

- The feasible poll center x_F^k (if it exists) must satisfy:

$$(x_F^k, \Delta^k) \in \arg \max_{(x^j, \Delta^j) \in L^k} \Delta^j.$$

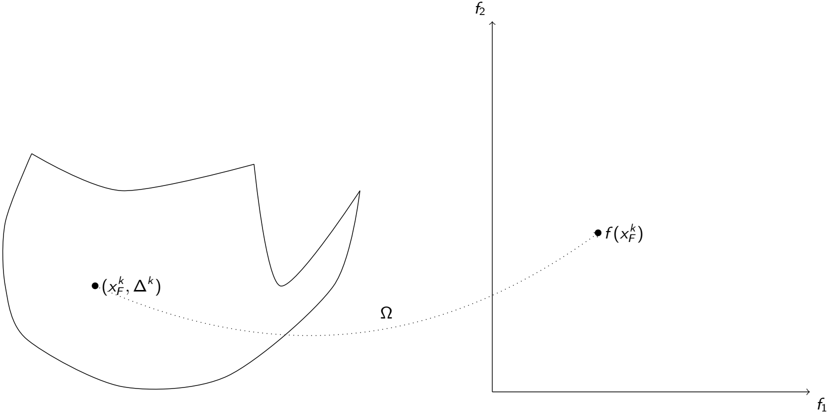
- The infeasible poll center x_I^k (if it exists) must satisfy:

$$(x_I^k, \Delta^k) \in \arg \max_{x \in I^k} h(x).$$

- The optional poll center $x_{opt}^k \in I^k$ (dependent on x_F^k).

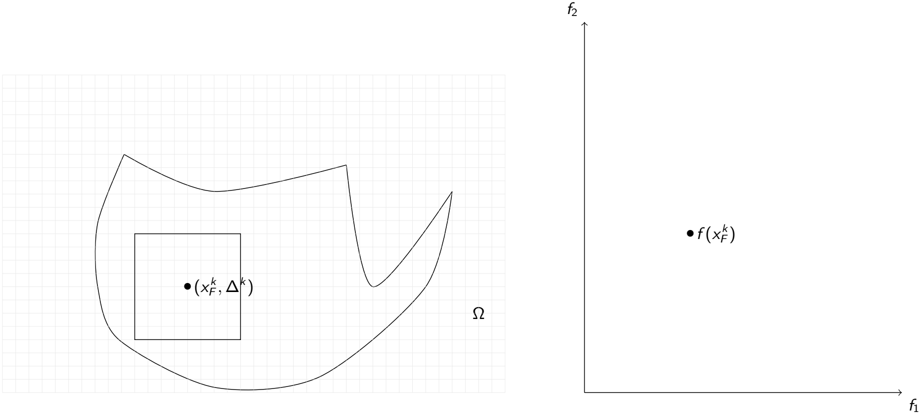
Reminder: an iteration of DMulti-MADS in the feasible case

Initialization of the iteration



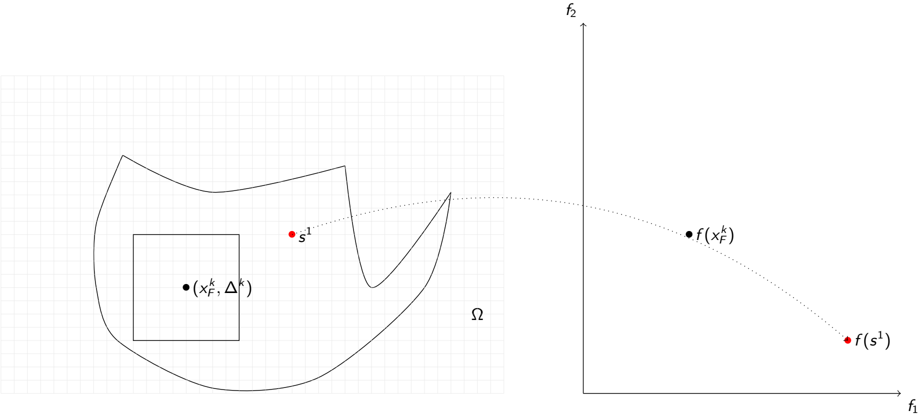
Reminder: an iteration of DMulti-MADS in the feasible case

Frame of parameter Δ^k/corresponding mesh



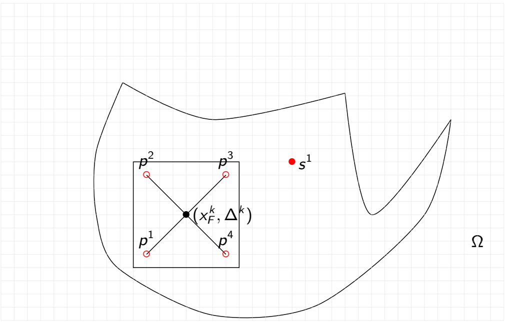
Reminder: an iteration of DMulti-MADS in the feasible case

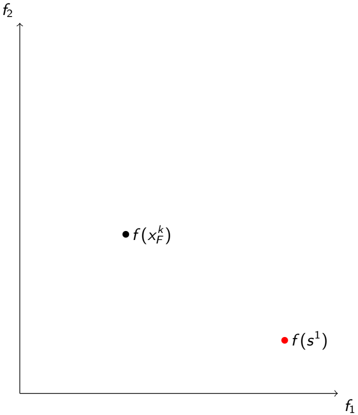
Search step



Reminder: an iteration of DMulti-MADS in the feasible case

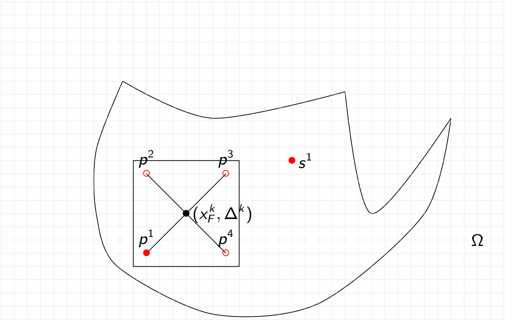
Poll step

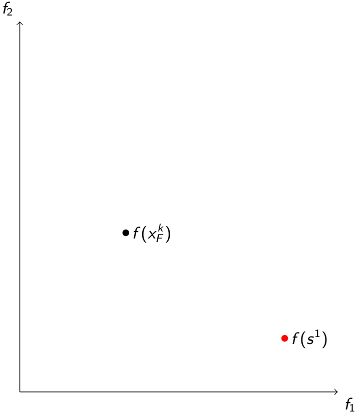




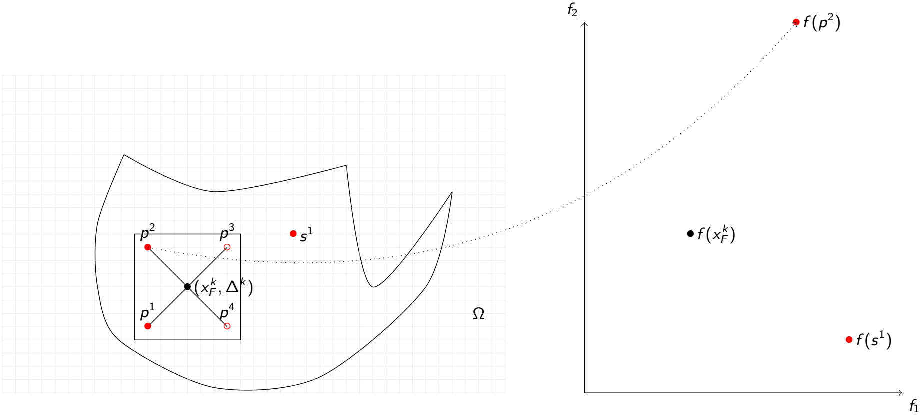
Reminder: an iteration of DMulti-MADS in the feasible case

Evaluation at p^1 fails ! $p^1 \notin X$

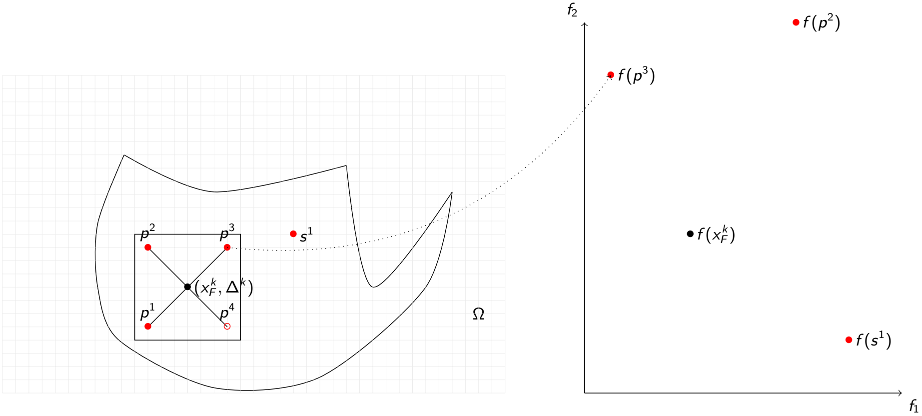




Reminder: an iteration of DMulti-MADS in the feasible case

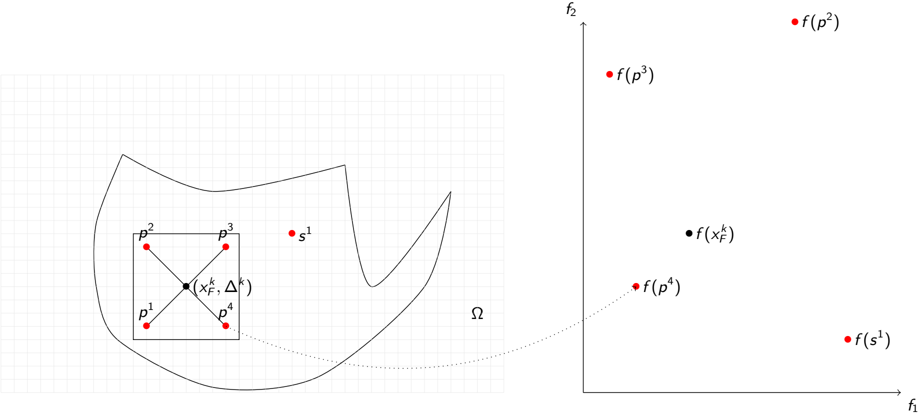


Reminder: an iteration of DMulti-MADS in the feasible case



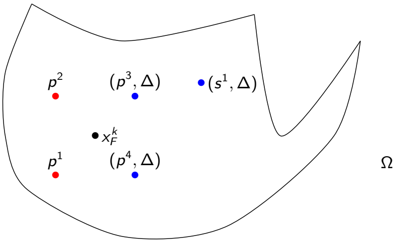
Reminder: an iteration of DMulti-MADS in the feasible case

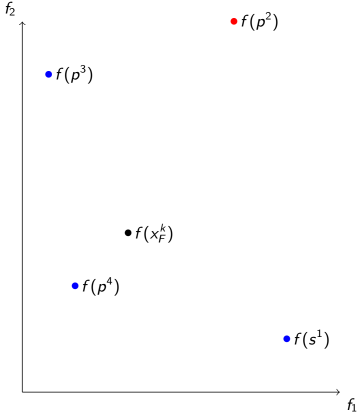
p^4 dominates x_F^k : success !



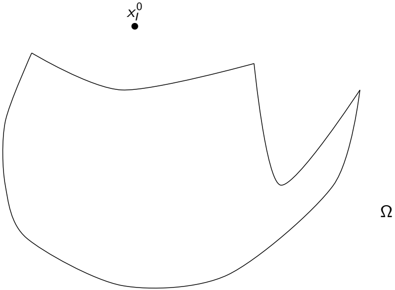
Reminder: an iteration of DMulti-MADS in the feasible case

Keep new non-dominated points: affect them $\Delta \geq \Delta^k$

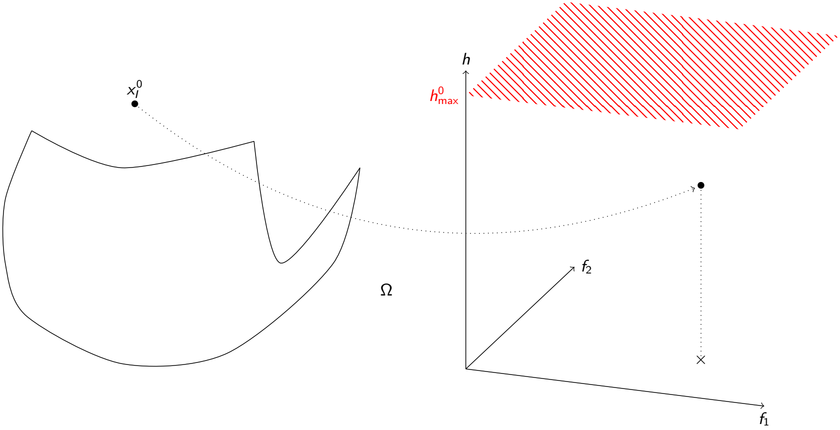




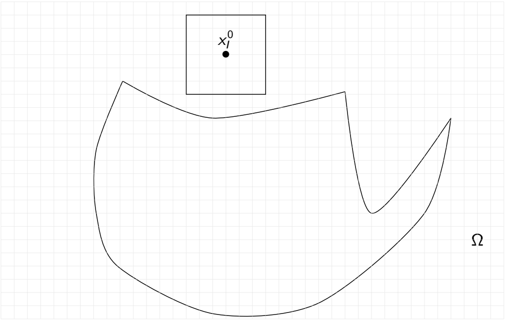
PB-DMulti-MADS: Iteration 0

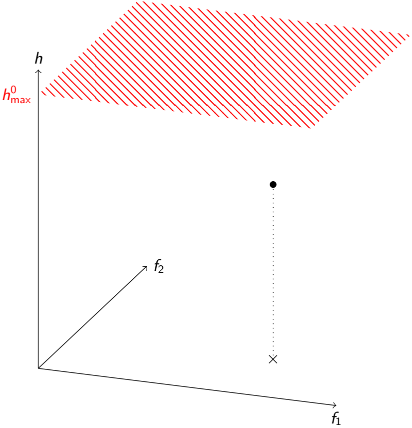


PB-DMulti-MADS: Iteration 0



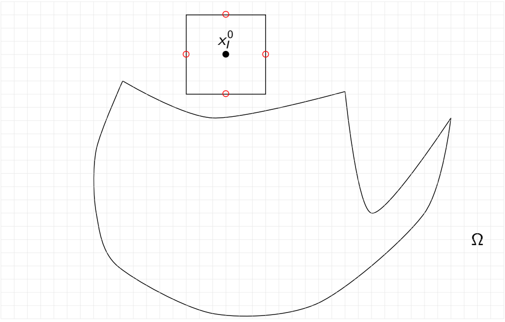
PB-DMulti-MADS: Iteration 0

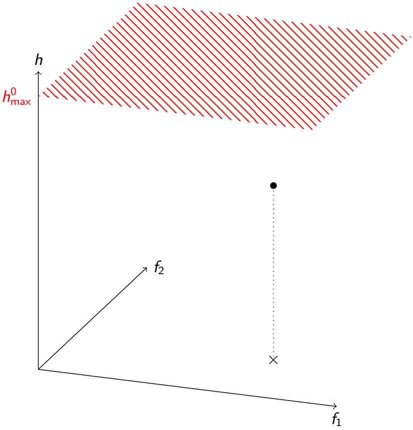




PB-DMulti-MADS: Iteration 0

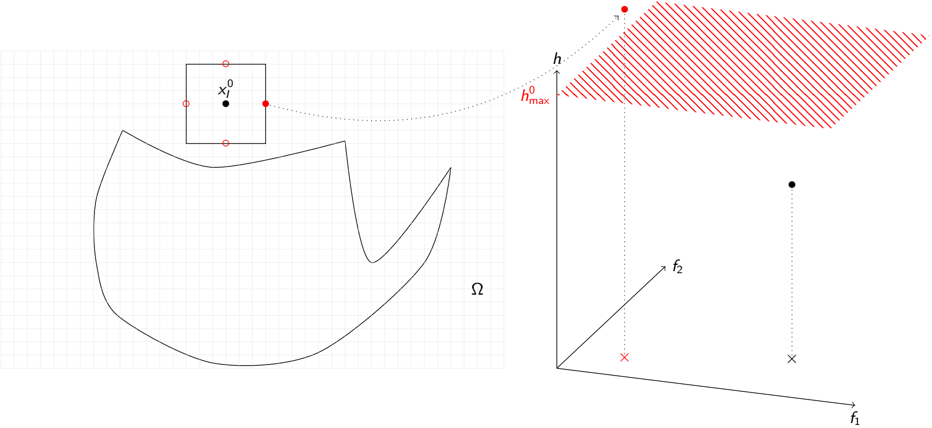
Poll step





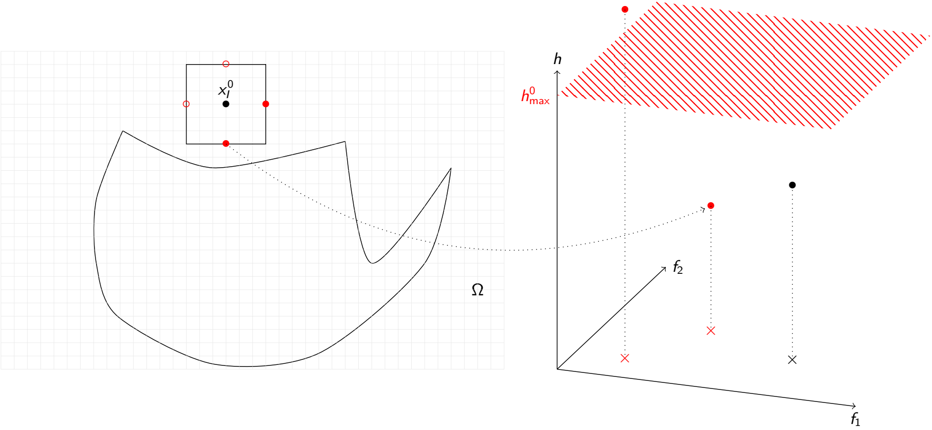
PB-DMulti-MADS: Iteration 0

Reject point (above threshold)



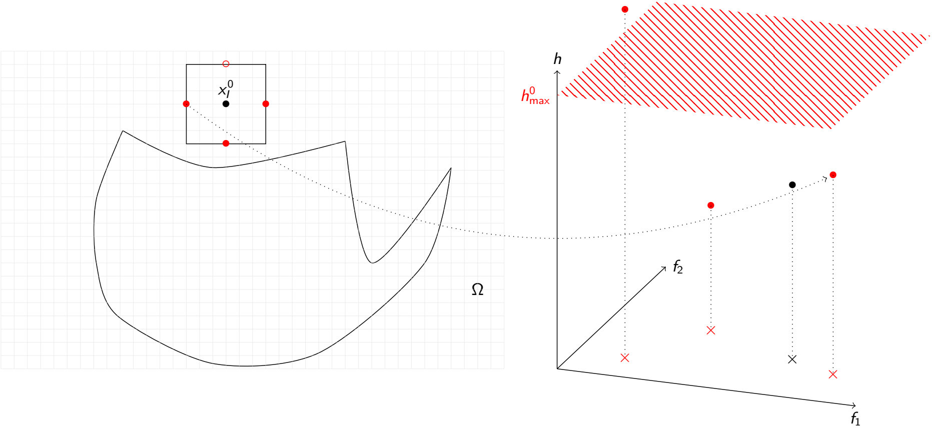
PB-DMulti-MADS: Iteration 0

Better h, f uncomparable



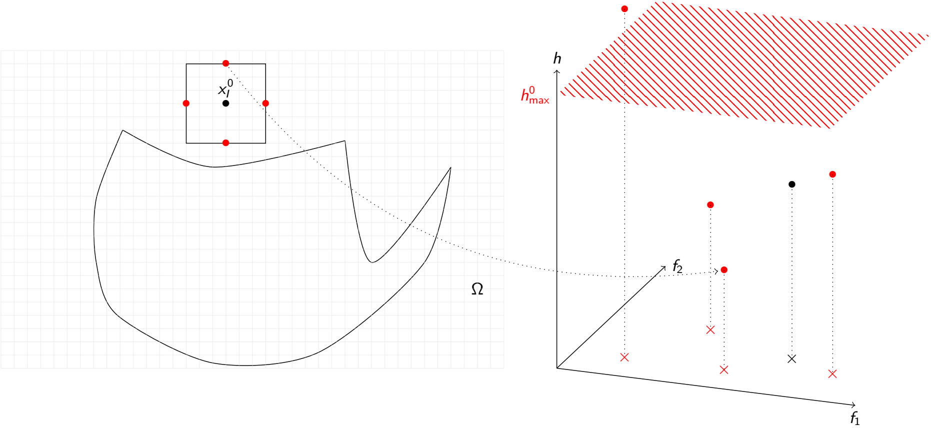
PB-DMulti-MADS: Iteration 0

Worse h, f uncomparable



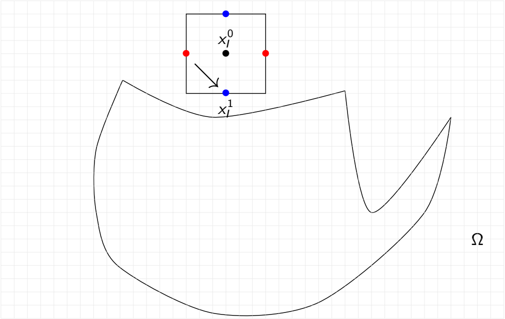
PB-DMulti-MADS: Iteration 0

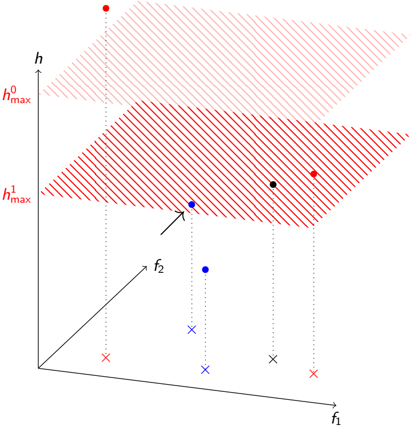
Dominates the current incumbent in terms of h of f values



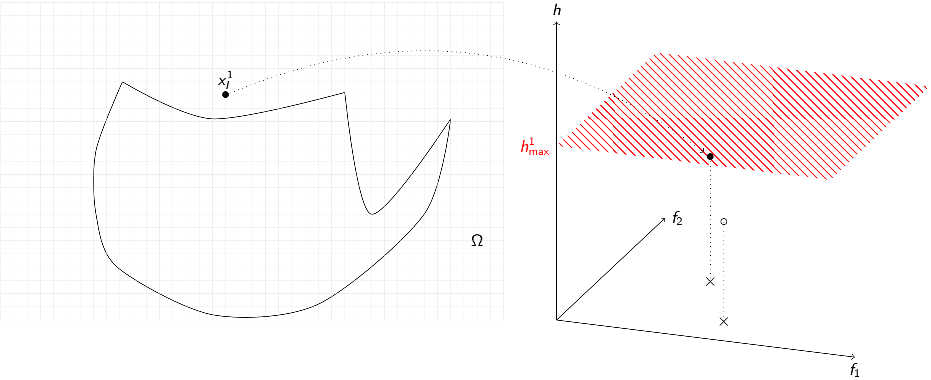
PB-DMulti-MADS: Iteration 0

The new incumbent is the one among the new non-dominated ones (in terms of f) with the highest h-value

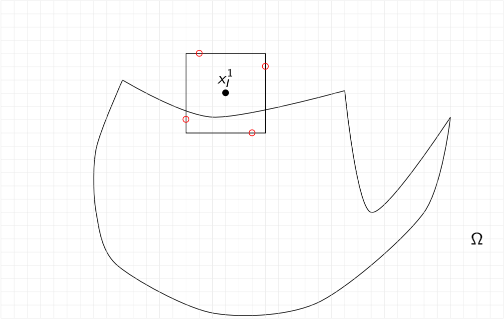


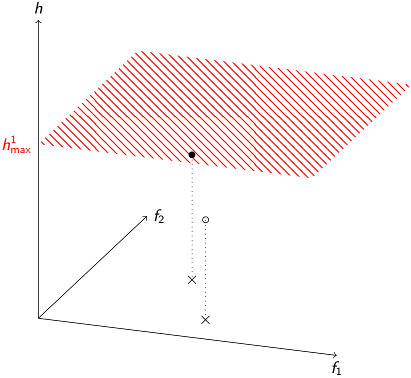


PB-DMulti-MADS: Iteration 1

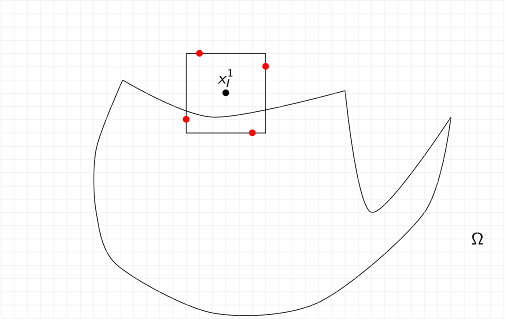


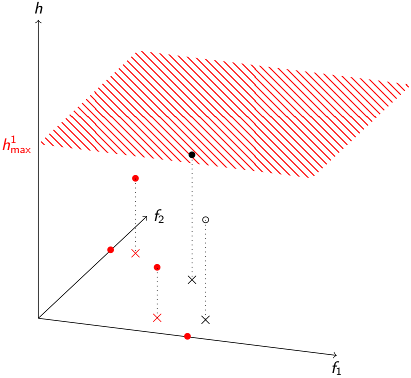
PB-DMulti-MADS: Iteration 1





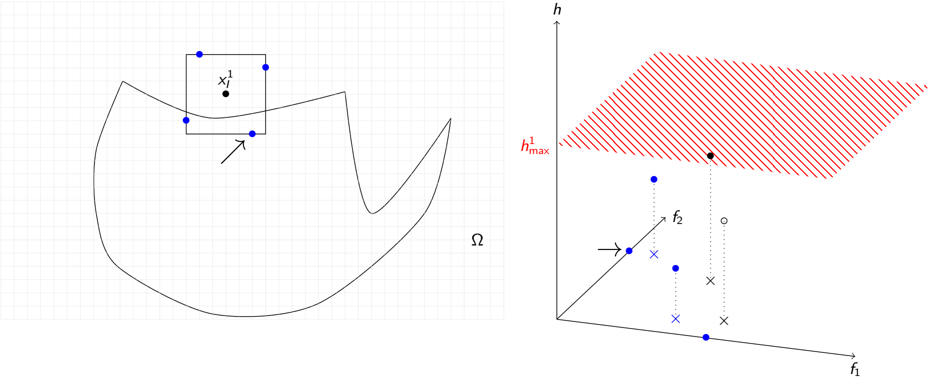
PB-DMulti-MADS: Iteration 1





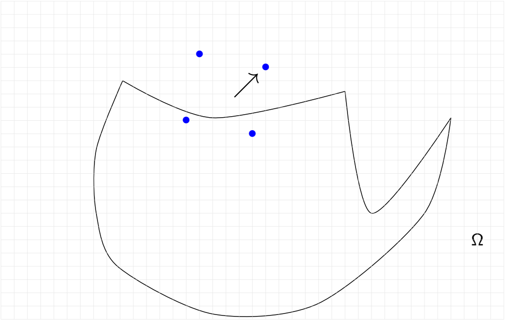
PB-DMulti-MADS: Iteration 1

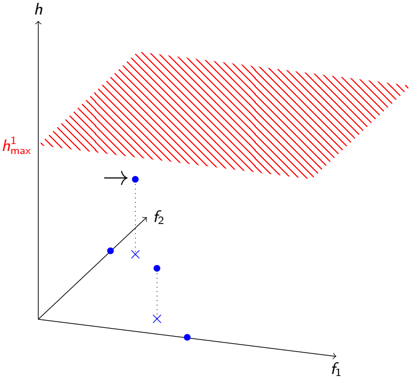
New feasible poll center



PB-DMulti-MADS: Iteration 1

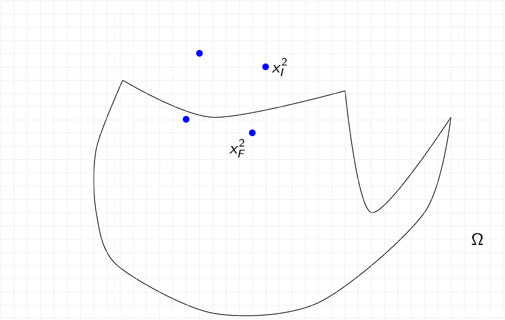
New infeasible poll center

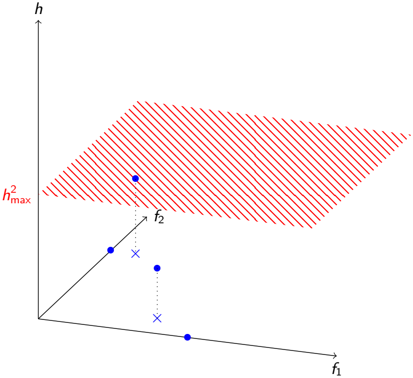




PB-DMulti-MADS: Iteration 1

$h_{\max}^k$ decreases toward iterations





Updating $h^k_{\max}$

Success, dominance and failure [Audet and Dennis, 2009]

At the end of an iteration k, three cases can happen:

- If a new point y is found which satisfies $y \prec_f x_F^k$ or $y \prec_h x_I^k$, the iteration is said to be dominating. In this case, $h^{k+1}_{\max} := h(x_I^k)$.
- If a new point y is found which satisfies $0 < h(x) < h(x_I^k)$, then the barrier threshold value is set to

$$h^{k+1}_{\max} := \max_{x \in V^k} \{0 < h(x) < h(x_I^k)\}$$

  where V^k is the cache at the end of iteration k. The iteration is said to be improving.
- Otherwise, the iteration is declared as unsuccessful, and $h^{k+1}_{\max} := h(x_I^k)$.

A bit of theory I

Assumptions

- Assume a starting point in X.
- All iterates lie at the intersection of a mesh and a compact set.

We introduce a definition taken from [Liuzzi et al., 2016].

Definition (Linked sequence)

Let $\{L^k\}_{k\in\mathbb{N}}$ with $L^k = \left\{(x_F^j, \Delta^j), x_F^j \in \Omega, \Delta^j > 0, j = 1, 2, \ldots, |L^k|\right\}$ be the sequence of current approximated Pareto sets generated by the DMulti-MADS algorithm. A linked sequence is defined as a sequence $\{(x_F^{j_k}, \Delta^{j_k})\}$ such that for any $k = 1, 2, \ldots$, the pair $(x^{j_k}, \Delta^{j_k}) \in L^k$ is generated at iteration $k-1$ of DMulti-MADS by the pair $(x_F^{j_{k-1}}, \Delta^{j_{k-1}}) \in L^{k-1}$.

Under some classical direct search assumptions, we can prove:

A bit of theory II

Theorem (Feasible case)

For each linked sequence $\{(x_F^{j_k}, \Delta^{j_k})\}$, *there exists a subset of indexes* K' *such that* $\{x_F^{j_k}\}_{k \in K'}$ *is a refining subsequence converging to a Pareto-Clarke locally optimal point* $\hat{x}_F^{j}$.

Unfeasible case

Under investigation (similar to [Audet and Dennis, 2009] ?).

Why a filter-based approach ?

- "Intuitive" to understand.
- "No external parameters/optimization hyperparameters".

PB-DMulti-MADS VS !!

Core

- Implemented in `Julia`.
- Speculative search.
- Poll step: $n+1$ directions for the first poll center, 2 directions for the second poll center and 2 directions for the optional poll center, Orthomads strategy.
- Granular and dynamic mesh scaling [Audet et al., 2019].
- Spread strategy.
- Opportunistic.

Competitors

- BiMADS [Audet et al., 2008] implemented in `Nomad.3.9.1` [Le Digabel, 2011]: default parameters, line search initialization.
- DFMO [Liuzzi et al., 2016] implemented in `DFMO`; default parameters.
- NSGA-II [Deb et al., 2000] implemented in `pymoo 0.4.1`; default parameters, 10 different seeds.

Data profiles

- Use of the hypervolume indicator [Zitzler et al., 2003] to build data profiles.
- Use of the constrained benchmark set proposed by [Liuzzi et al., 2016] of functions with $m = 2$, $n \in \{3, \ldots, 30\}$; $|\mathcal{P}| = 103$.
- For each algorithm $a \in \mathcal{A}$, a maximal budget of 20000 evaluations.

Preliminary results I

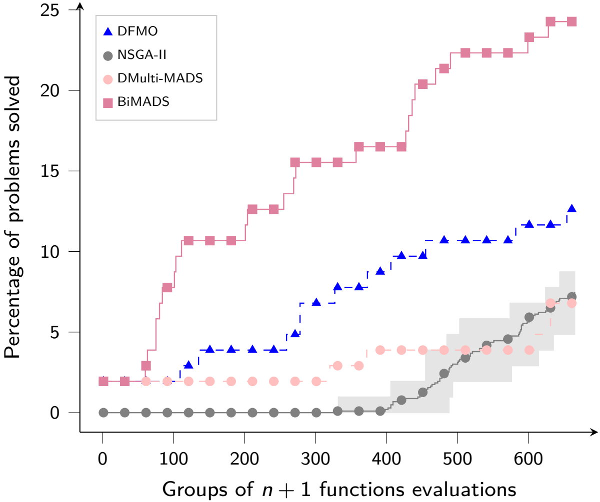


Figure: Data profiles; $\tau = 10^{-2}$

Preliminary results II

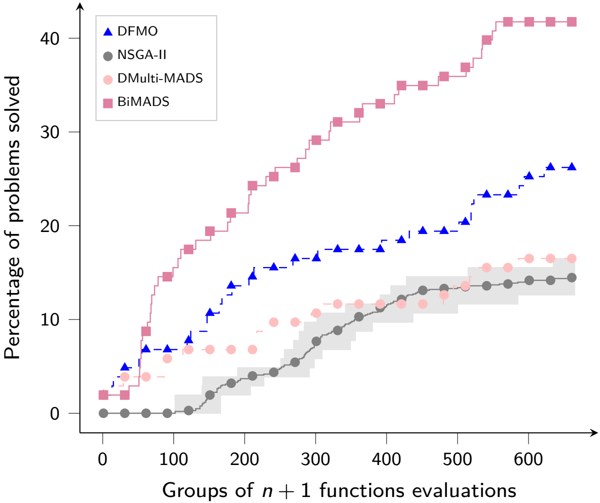


Figure: Data profiles; $\tau = 5 \times 10^{-2}$

Preliminary results III

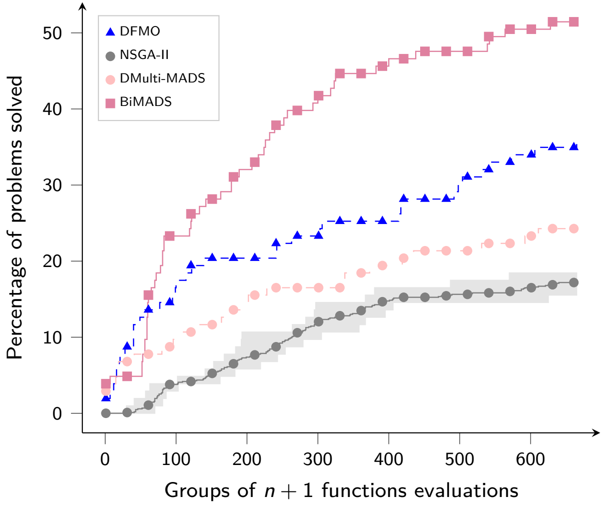


Figure: Data profiles; $\tau = 10^{-1}$

Discussion

Pessimistic view

It is not really efficient: change paradigm (merit function approach) ?

Optimistic view

- Implement a two-phase approach such as BiMADS.
- Deactivate BiMADS models.
- Reinvestigate code.

Once it is done

- Compare to DMS with a penalty-based approach [Liuzzi et al., 2016].
- Test on Styrene (against extreme barrier strategy).

Discussion

Pessimistic view

It is not really efficient: change paradigm (merit function approach) ?

Optimistic view

- Implement a two-phase approach such as BiMADS.
- Deactivate BiMADS models.
- Reinvestigate code.

Once it is done

- Compare to DMS with a penalty-based approach [Liuzzi et al., 2016].
- Test on Styrene (against extreme barrier strategy).

Discussion

Pessimistic view

It is not really efficient: change paradigm (merit function approach) ?

Optimistic view

- Implement a two-phase approach such as BiMADS.
- Deactivate BiMADS models.
- Reinvestigate code.

Once it is done

- Compare to DMS with a penalty-based approach [Liuzzi et al., 2016].
- Test on Styrene (against extreme barrier strategy).

Thank you for your attention ! Do you have any questions ?

Audet, C., Côté, P., Poissant, C., and Tribes, C. (2019).
Monotonic grey box direct search optimization.
Optimization Letters.

Audet, C. and Dennis, J. (2009).
A progressive barrier for derivative-free nonlinear programming.
SIAM Journal on Optimization, 20(1):445–472.

Audet, C. and Hare, W. (2017).
Derivative-Free and Blackbox Optimization.
Springer International Publishing.

Audet, C., Savard, G., and Zghal, W. (2008).
Multiobjective optimization through a series of single-objective formulations.
SIAM Journal on Optimization, 19(1):188–210.

Audet, C., Savard, G., and Zghal, W. (2010).
A mesh adaptive direct search algorithm for multiobjective optimization.
European Journal of Operational Research, 204(3):545–556.

Cocchi, G., Liuzzi, G., Papini, A., and Sciandrone, M. (2018).
An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints.
Computational Optimization and Applications, 69(2):267–296.

Custódio, A. L., Madeira, J. F. A., Vaz, A. I. F., and Vicente, L. N. (2011).
Direct multisearch for multiobjective optimization.
SIAM Journal on Optimization, 21(3):1109–1140.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000).
A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.
In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fletcher, R. and Leyffer, S. (2002).
Nonlinear programming without a penalty function.
Mathematical Programming, 91(2):239–269.

Gratton, S. and Vicente, L. (2014).
A merit function approach for direct search.
SIAM Journal on Optimization, 24(4):1980–1998.

Le Digabel, S. (2011).
Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm.
ACM Trans. Math. Softw., 37(4):44:1–44:15.

Lewis, R. M., Torczon, V. J., and Kolda, T. G. (2006).
A generating set direct search augmented lagrangian algorithm for optimization with a combination of general and linear constraints.
Technical report.

Liuzzi, G. and Lucidi, S. (2009).
A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function.
SIAM Journal on Optimization, 20(1):1–29.

Liuzzi, G., Lucidi, S., and Rinaldi, F. (2016).
A derivative-free approach to constrained multiobjective nonsmooth optimization.
SIAM Journal on Optimization, 26(4):2744–2774.

Nocedal, J. and Wright, S. (2006).
Numerical optimization.
Springer Science & Business Media.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. (2003).
Performance assessment of multiobjective optimizers: an analysis and review.
IEEE Transactions on Evolutionary Computation, 7(2):117–132.