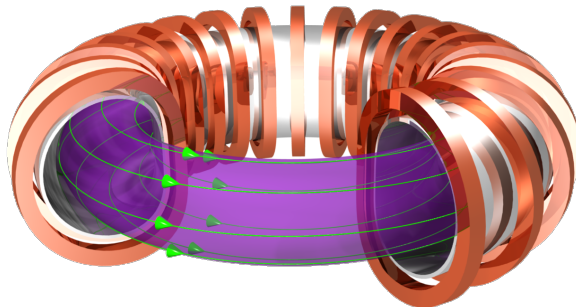


OptiFun: Optimising Nuclear Fusion with Functional Programming

Patrik Jansson

Functional Programming unit, Chalmers University of Technology

2023-06-09



What is “OptiFun”?

- Chalmers Energy Area of Advance grant (3M SEK) for a collaboration project: *“Optimising Fusion with Functional Programming”*
- 2022–2023: joint work with the Physics department
- PIs: Patrik Jansson + Tünde Fülöp
- FP-side: Patrik, Nick Smallbone, Nicola Botta + students

Aim: “... **to combine numeric and symbolic methods** to accelerate first-principles simulations and enable optimisation of fusion confinement designs. ... build a **reliable and versatile toolset** for performance **optimisation** of strongly coupled physical systems **with a large parameter space**, and apply it to explore reliable fusion designs.”

Setting: black-box function with multiple objectives

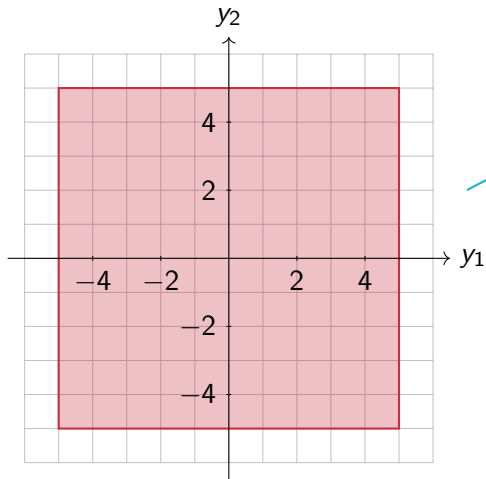
- The object of study is an expensive, black-box objective function $e : \mathbb{R}^n \rightarrow \mathbb{R}^m$
- ... and the aim is to find “stable minima” of e .
- Expensive means that each call to e takes minutes or hours to evaluate.
- Stable means that e should be not vary too much near the minima.
- The optimisation framework has to choose points wisely and use the information from each call well.

Setting: black-box function with multiple objectives

- The object of study is an expensive, black-box objective function $e : \mathbb{R}^n \rightarrow \mathbb{R}^m$
- ... and the aim is to find “stable minima” of e .
- Expensive means that each call to e takes minutes or hours to evaluate.
- Stable means that e should be not vary too much near the minima.
- The optimisation framework has to choose points wisely and use the information from each call well.
- Example: two control inputs and two objectives ($m=n=2$): “ $x_1=\text{cost}$ ” and “ $x_2=\text{error}$ ”.
 - The two inputs are some knobs we can turn to change the settings.
 - In the actual physics scenario the inputs are amounts of two gases injected into the fusion reaction chamber.
 - We are given an interval for each input parameter and the task is to find input parameters which give “stable minima”.

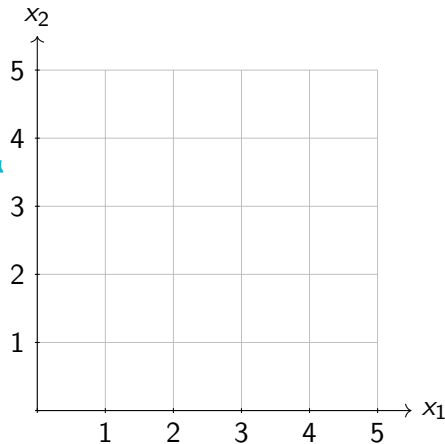
More concrete: black-box function from $\mathbb{R}^2 \rightarrow \mathbb{R}^2$

$Y = \text{controls}$



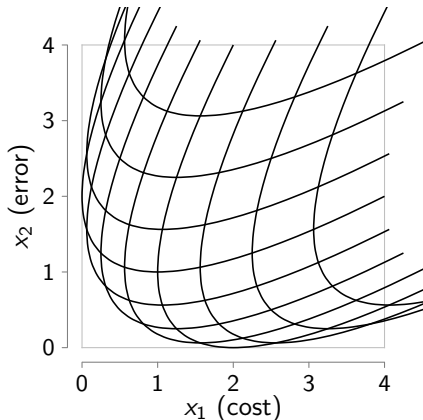
$e: Y \rightarrow X$

$X = \text{objectives}$



- There is rarely a single control point (input vector) that optimises all the objectives.
- Common solution: combine the objectives $\Rightarrow$ single objective optimization.
- Difficult problem: choose relative weight of the objectives.
- Instead we compute the *Pareto front* of e and let domain experts decide.
- The Pareto front is a generalisation of optimum and explained in the coming slides.

Objective space (idealised point cloud)



- Each curve keeps y_1 or y_2 fixed and varies the other control continuously.
- The curves shown are for $y_1 = -1, -0.75, \dots, 1.75, 2$ (same for y_2)

type $\mathbb{R} = \text{Double}$

type $\mathbb{R}^2 = (\mathbb{R}, \mathbb{R})$

type $Y = \mathbb{R}^2$ -- controls

type $X = \mathbb{R}^2$ -- objectives

circles :: $Y \rightarrow X$ -- this is our function *e*

circles *ys* = (*obj*₁ *ys*, *obj*₂ *ys*)

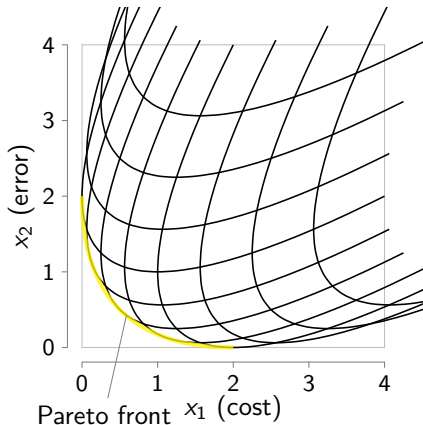
where *obj*₁ = *sqDist* (1, 0)

*obj*₂ = *sqDist* (0, 1)

sqDist :: $\mathbb{R}^2 \rightarrow \mathbb{R}^2 \rightarrow \mathbb{R}$

sqDist (*c*₁, *c*₂) (*y*₁, *y*₂) = (*y*₁ - *c*₁)² + (*y*₂ - *c*₂)²

Pareto front (“the optima” in objective space)



- Each curve keeps y_1 or y_2 fixed and varies the other control continuously.
- The curves shown are for $y_1 = -1, -0.75, \dots, 1.75, 2$ (same for y_2)
- The Pareto front is the set of “optimal points” in objective space.

$circles :: Y \rightarrow X$ -- this is our function e

$circles\ ys = (obj_1\ ys, obj_2\ ys)$

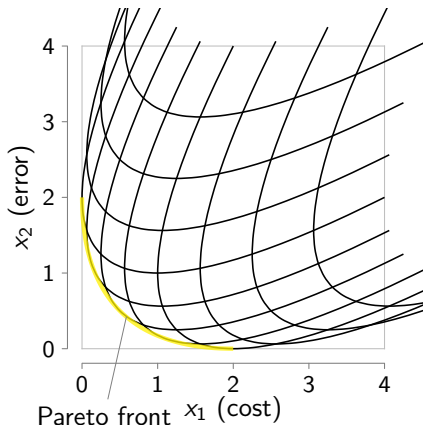
where $obj_1 = sqDist\ (1, 0)$

$obj_2 = sqDist\ (0, 1)$

$sqDist :: \mathbb{R}^2 \rightarrow \mathbb{R}^2 \rightarrow \mathbb{R}$

$sqDist\ (c_1, c_2)\ (y_1, y_2) = (y_1 - c_1)^2 + (y_2 - c_2)^2$

Pareto front (“the optima” in objective space)

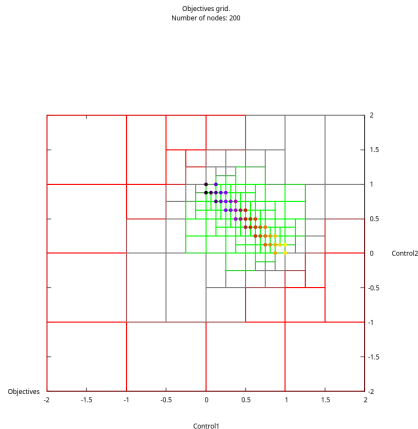


- Each curve keeps y_1 or y_2 fixed and varies the other control continuously.
- The curves shown are for $y_1 = -1, -0.75, \dots, 1.75, 2$ (same for y_2)
- The Pareto front is the set of “optimal points” in objective space.

$$(x_1, x_2) \text{ 'dominates' } (x'_1, x'_2) = (x_1 \leq x'_1) \wedge (x'_2 \leq x_2)$$

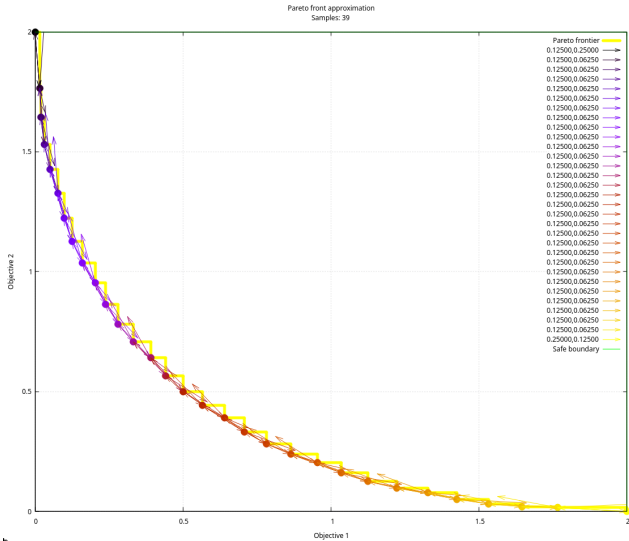
Def.: The *Pareto front* is the set of non-dominated objective points.

Adaptive mesh refinement (AMR)



- Start with a rectangle covering the whole search space $\Rightarrow$ initial queue.
- Check stopping condition (small model error) $\Rightarrow$ maybe done.
- Split the “best” few cells (in the “best” direction)
- Extend the queue with the new cells.
- Recurse.

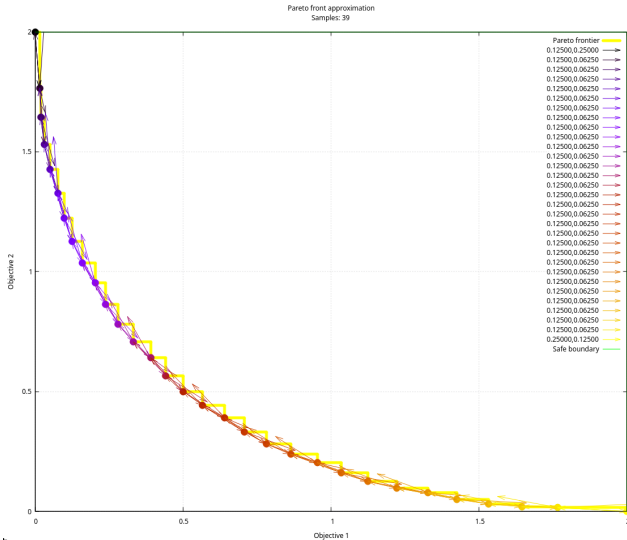
Approximate Pareto front from AMR (for circles example)



Some observations:

- The approximate front is close to the ideal front

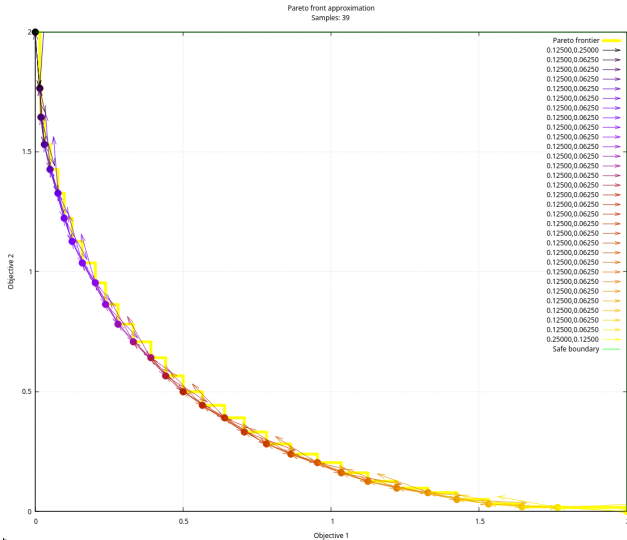
Approximate Pareto front from AMR (for circles example)



Some observations:

- The approximate front is close to the ideal front
- The arrows illustrate the stability: they point to the objective point corresponding to the nearest neighbours of points on the front.

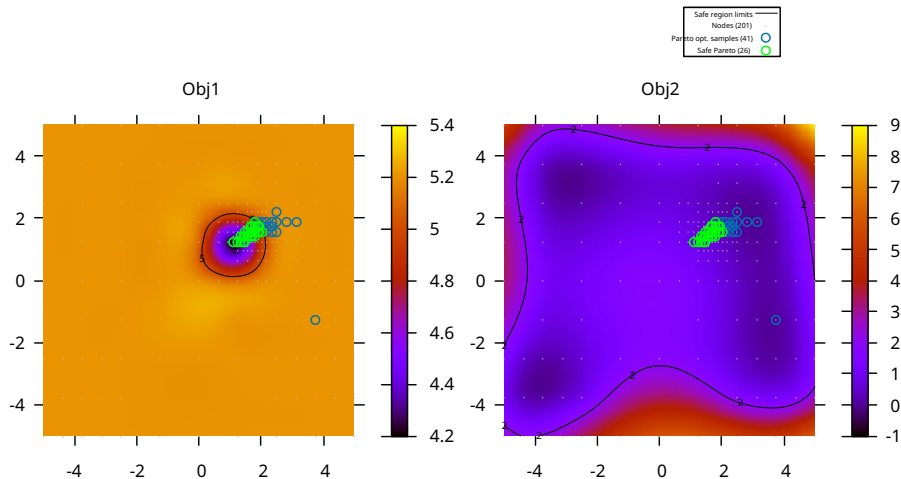
Approximate Pareto front from AMR (for circles example)



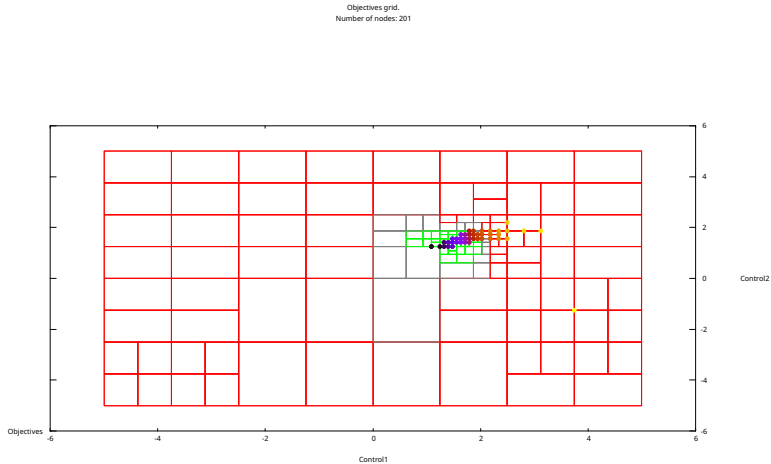
Some observations:

- The approximate front is close to the ideal front
- The arrows illustrate the stability: they point to the objective point corresponding to the nearest neighbours of points on the front.
- This is after 200 function evaluations, and 39 points are “on the front”.

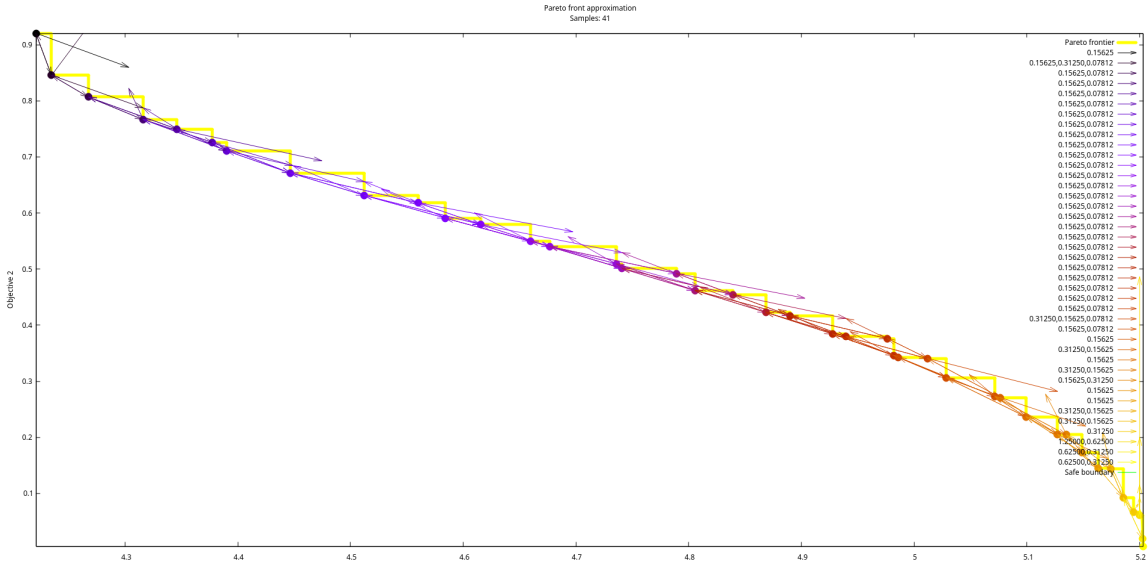
Another example (“valley”): Contour plot



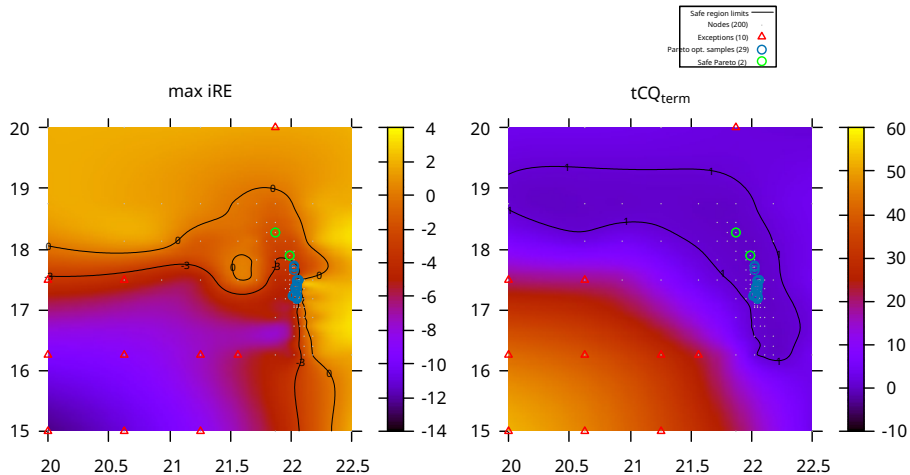
Another example (“valley”): Adaptive grid



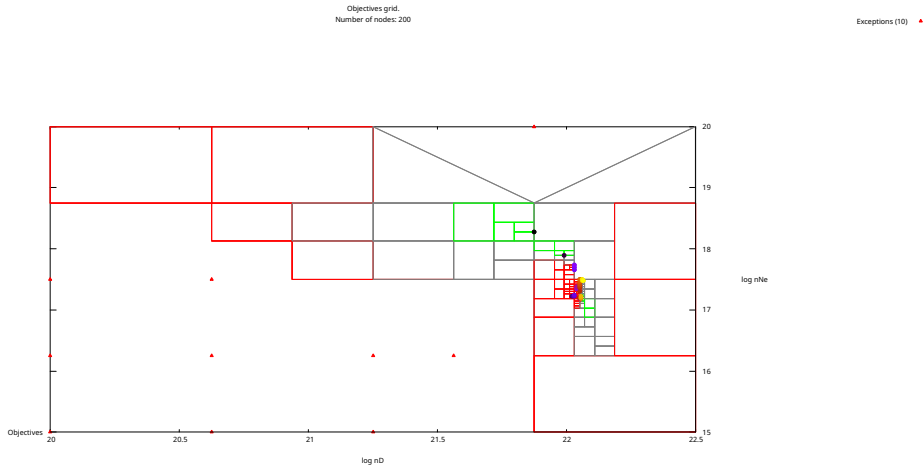
Another example (“valley”): Pareto front



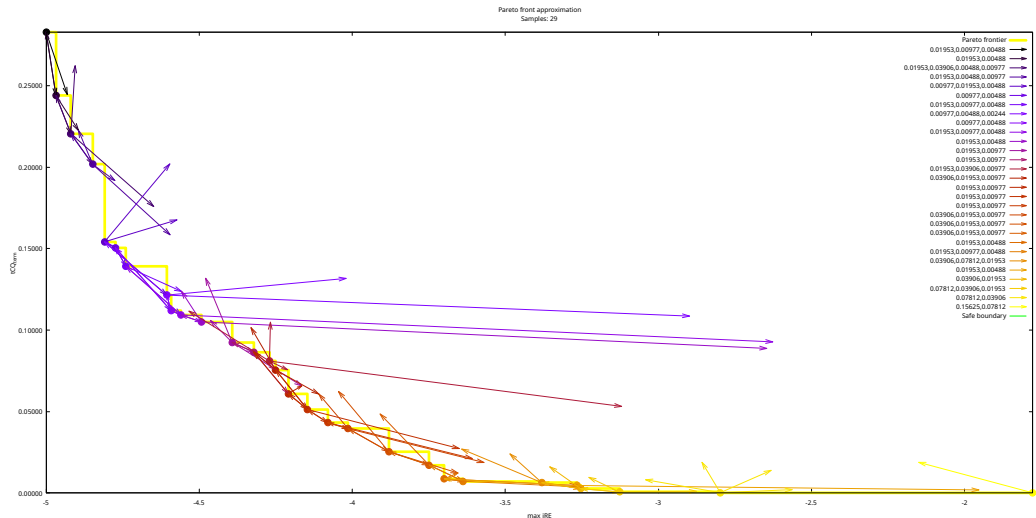
Simulation example (“DREAM”): Contour plot



Simulation example (“DREAM”): Adaptive grid

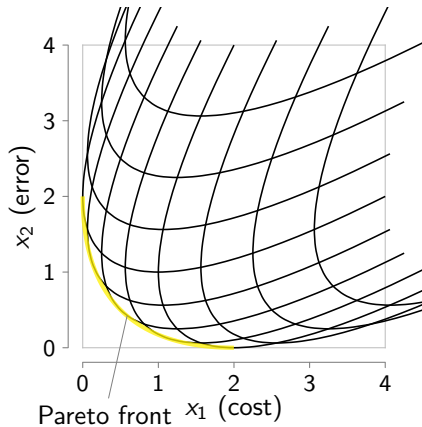


Simulation example (“DREAM”): Pareto front

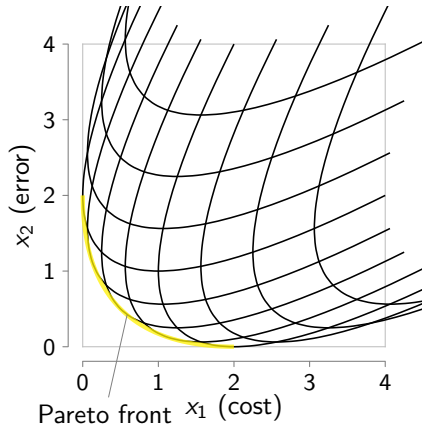


Summary

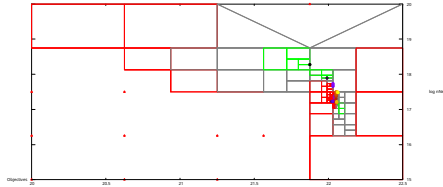
- Multi-objective optimisation $\Rightarrow$ compute Pareto front



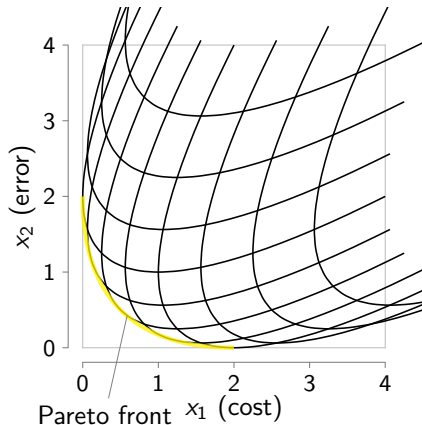
Summary



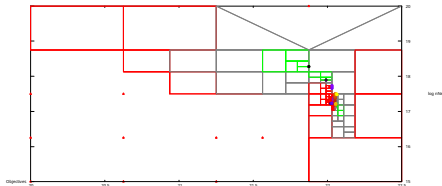
- Multi-objective optimisation $\Rightarrow$ compute Pareto front
- Expensive function $\Rightarrow$ adaptive mesh refinement (AMR)



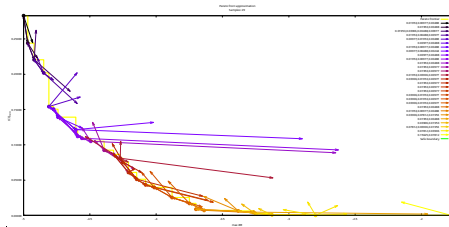
Summary



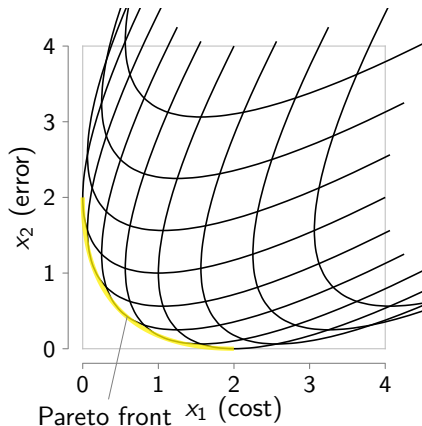
- Multi-objective optimisation $\Rightarrow$ compute Pareto front
- Expensive function $\Rightarrow$ adaptive mesh refinement (AMR)



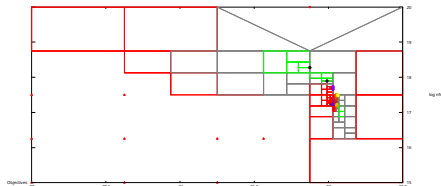
- Stability wanted $\Rightarrow$ check variation near the front



Questions?



- Multi-objective optimisation $\Rightarrow$ compute Pareto front
- Expensive function $\Rightarrow$ adaptive mesh refinement (AMR)



- Stability wanted $\Rightarrow$ check variation near the front

