First-Order Methods and Principled Frameworks for Optimization

Sofiane Tanji

Advisor: Prof. François Glineur

Public PhD defense · 20 November 2026

UCLouvain

How to read this talk

Part I first-order methods

snacks

Large-scale kernel support vector machines

ConvexHullPricing.jl

Dual methods for convex hull pricing

Part II principled frameworks

Argo.jl

Guarantee-based recommendation of first-order methods

badr

Fairness-informed Pareto optimization

How to read this talk



Part I first-order methods

snacks

Large-scale kernel support vector machines

ConvexHullPricing.jl

Dual methods for convex hull pricing

Part II principled frameworks

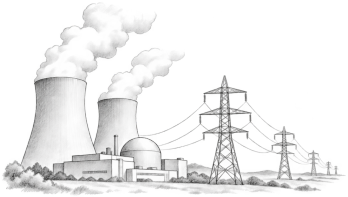
Argo.jl

Guarantee-based recommendation of first-order methods

badr

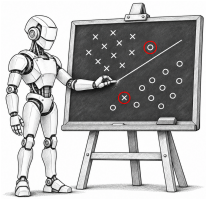
Fairness-informed Pareto optimization

What is optimization?



Power systems

choose the prices
minimize the side payments

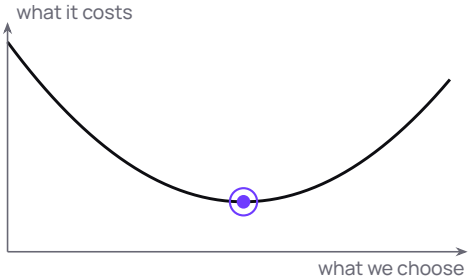


Machine learning

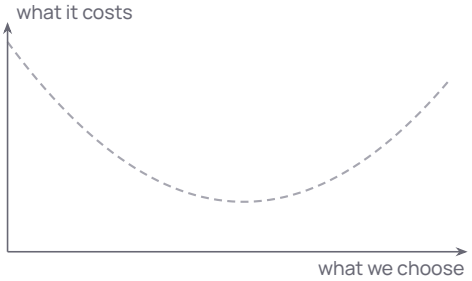
choose the model
minimize its errors

$$\min_{x \in \mathcal{X}} \; f(x)$$

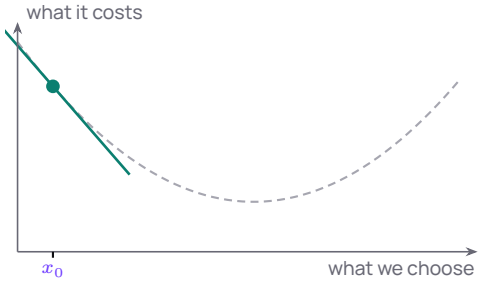
Black-box oracle framework and iterative methods



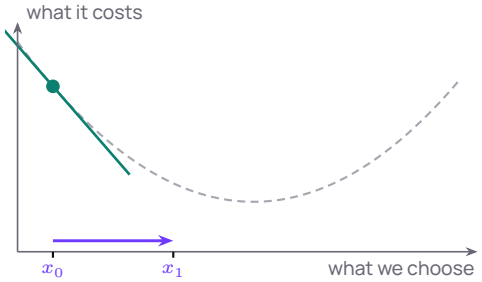
Black-box oracle framework and iterative methods



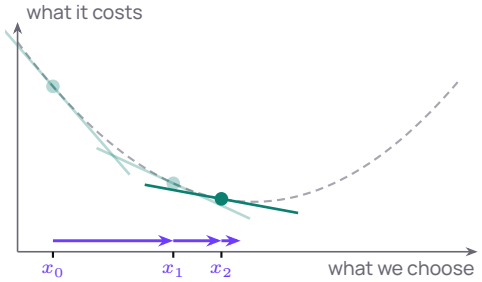
Black-box oracle framework and iterative methods



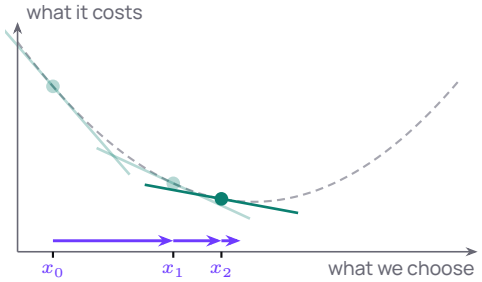
Black-box oracle framework and iterative methods



Black-box oracle framework and iterative methods

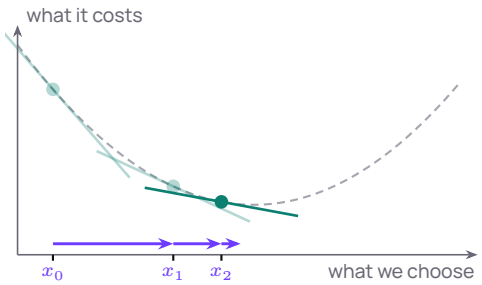


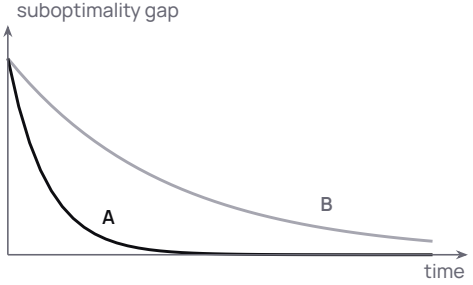
Black-box oracle framework and iterative methods



number of questions × cost of one question

Black-box oracle framework and iterative methods





number of questions × cost of one question

snacks

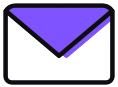
Large-scale kernel support vector machines
with Andrea Della Vecchia, François Glineur and Silvia Villa

Learning to classify

From labelled examples, learn f: object $\mapsto$ class

Learning to classify

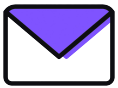
From labelled examples, learn f: object $\mapsto$ class



Emails
spam or not spam

Learning to classify

From labelled examples, learn f: object $\mapsto$ class



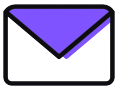
Emails
spam or not spam



Face unlock
the owner's face, or not

Learning to classify

From labelled examples, learn f: object $\mapsto$ class



Emails
spam or not spam

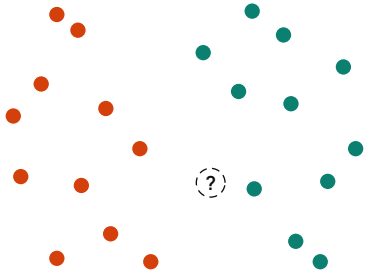


Face unlock
the owner's face, or not

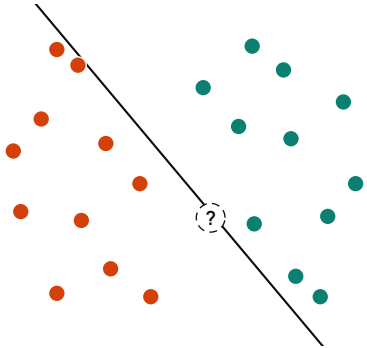


Points in the plane
red or green

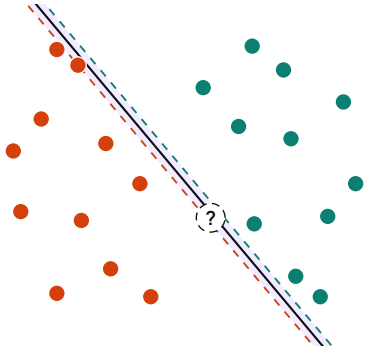
Which line to choose?



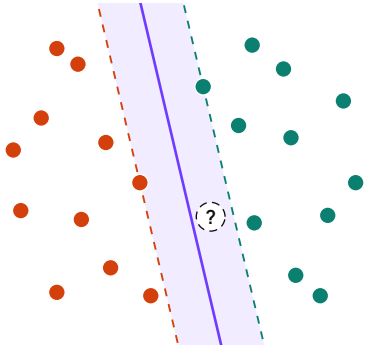
Which line to choose?



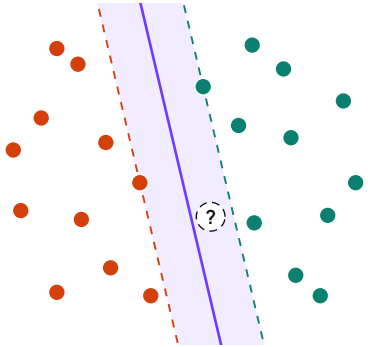
Which line to choose?



Which line to choose?



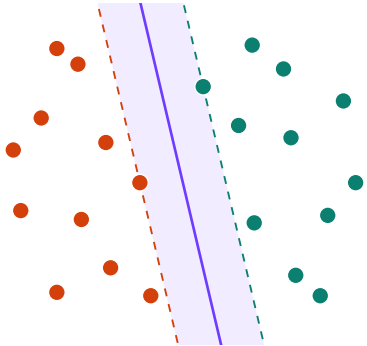
Which line to choose?



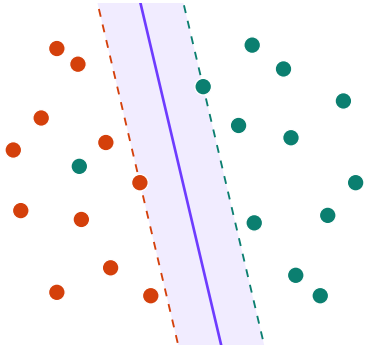
Maximize the width
of the band

Constraint: every point stays beyond the edge of its colour.

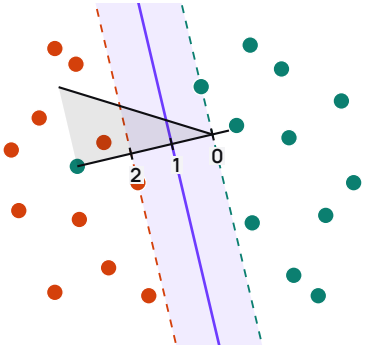
Allow violations, give them a cost



Allow violations, give them a cost

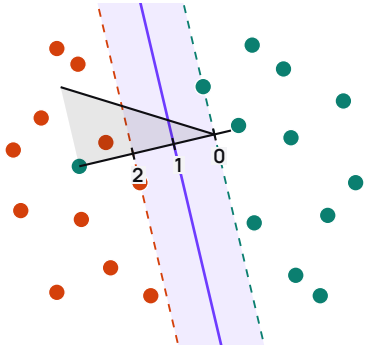


Allow violations, give them a cost



0 at the right edge · 1 on the boundary
2 at the wrong edge
unit: half the width of the band

Allow violations, give them a cost



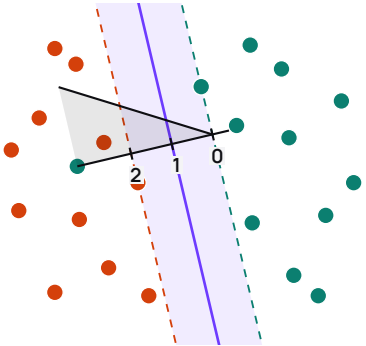
0 at the right edge · 1 on the boundary
2 at the wrong edge

unit: half the width of the band

$$\min_{u} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i z_i^\top u\}}_{\text{average cost}}$$

z_i a point
y_i $+1$ for green, -1 for red
u sets the line and its band

Allow violations, give them a cost



0 at the right edge · 1 on the boundary
2 at the wrong edge

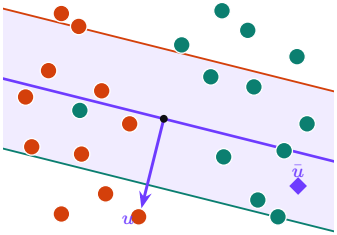
unit: half the width of the band

$$\min_{u} \quad \underbrace{\tfrac{\lambda}{2}\|u\|^2}_{\text{favour a wide band}} \quad + \underbrace{\tfrac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i z_i^\top u\}}_{\text{average cost}}$$

z_i a point
y_i $+1$ for green, -1 for red
u sets the line and its band

width of the band $= 2/\|u\|$

A step of snacks

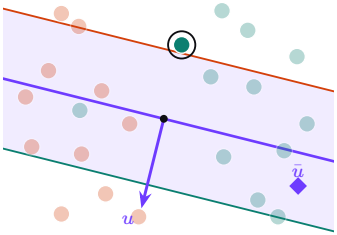


objective $F(u)$ 1.58
misclassified 14

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\substack{\text{back to} \\ \text{the anchor}}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\substack{\text{if } i \text{ is in the band} \\ \text{or on the wrong side:} \\ \text{add a bit of } y_i\, z_i \text{ to } u}}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks

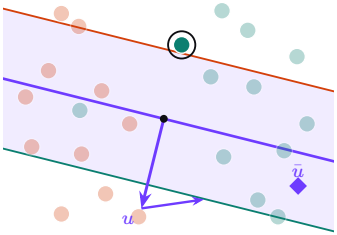


objective $F(u)$ 1.58
misclassified 14

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\substack{\text{back to} \\ \text{the anchor}}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\substack{\text{if } i \text{ is in the band} \\ \text{or on the wrong side:} \\ \text{add a bit of } y_i\, z_i \text{ to } u}}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks

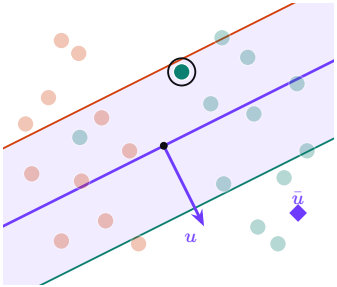


objective $F(u)$ 1.58
misclassified 14

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\text{back to the anchor}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\text{if } i \text{ is in the band or on the wrong side: add a bit of } y_i\, z_i \text{ to } u}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks

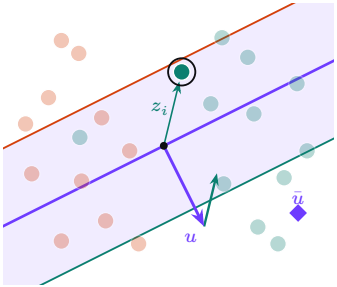


objective $F(u)$ 1.58
misclassified 14

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\text{back to the anchor}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\text{if } i \text{ is in the band or on the wrong side: add a bit of } y_i\, z_i \text{ to } u}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks

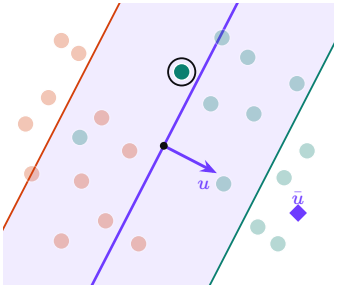


objective $F(u)$ 1.58
misclassified 14

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\text{back to the anchor}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\text{if } i \text{ is in the band or on the wrong side: add a bit of } y_i\, z_i \text{ to } u}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks

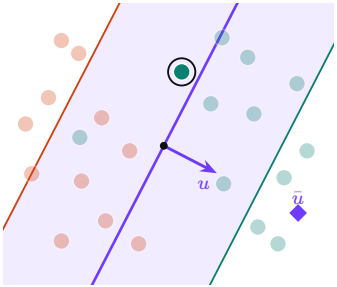


objective $F(u)$ 1.58 → 0.52
misclassified 14 → 3

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\text{back to the anchor}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\text{if } i \text{ is in the band or on the wrong side: add a bit of } y_i\, z_i \text{ to } u}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

A step of snacks



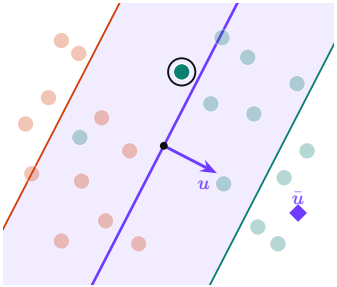
objective $F(u)$ 1.58 → 0.52
misclassified 14 → 3

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\text{back to the anchor}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\text{if } i \text{ is in the band or on the wrong side: add a bit of } y_i\, z_i \text{ to } u}$$

$\bar{u}$: anchor from the previous phase θ, β: step sizes

one example per step: a stochastic first-order oracle
cost: depends on the number of reference points, not on n

A step of snacks



objective $F(u)$ 1.58 → 0.52
misclassified 14 → 3

$$u^+ = u + \underbrace{\theta\,(\bar{u} - u)}_{\substack{\text{back to} \\ \text{the anchor}}} + \underbrace{\theta\beta\, y_i\, z_i\, \mathbf{1}[y_i\, z_i^\top u < 1]}_{\substack{\text{if } i \text{ is in the band} \\ \text{or on the wrong side:} \\ \text{add a bit of } y_i\, z_i \text{ to } u}}$$

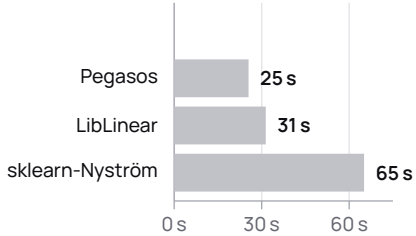
$\bar{u}$: anchor from the previous phase θ, β: step sizes

one example per step: a stochastic first-order oracle
cost: depends on the number of reference points, not on n

snacks repeats this step in phases; each phase moves the anchor to the previous phase's average.

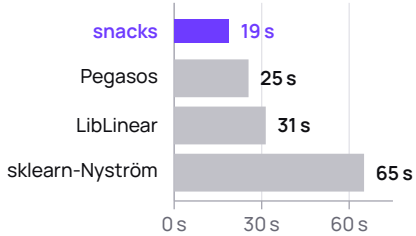
Faster than the usual solvers

9 datasets, 800 to 8.4 million examples
whole training time, geometric mean



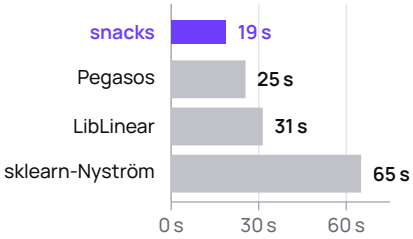
Faster than the usual solvers

9 datasets, 800 to 8.4 million examples
whole training time, geometric mean

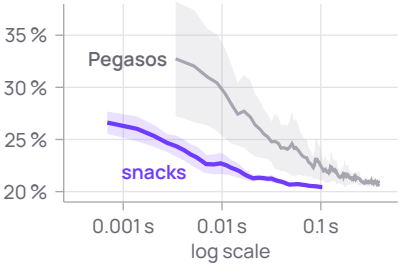


Faster than the usual solvers

9 datasets, 800 to 8.4 million examples
whole training time, geometric mean

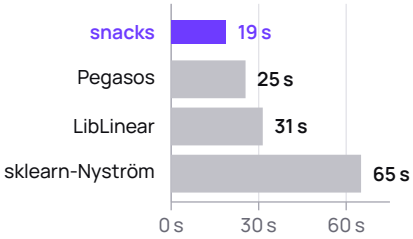


1 dataset, 3.6 million examples
test error over solver time

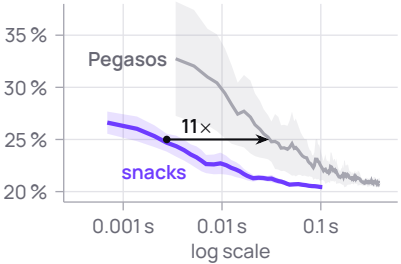


Faster than the usual solvers

9 datasets, 800 to 8.4 million examples
whole training time, geometric mean



1 dataset, 3.6 million examples
test error over solver time



ConvexHullPricing.jl

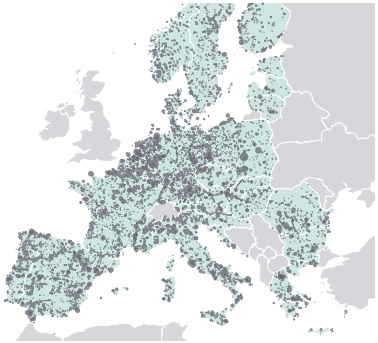
Pricing electricity when plants have start-up costs
with Yassine Kamri, François Glineur and Mehdi Madani

One algorithm sets Europe's prices

Every day, Euphemia sets tomorrow's prices.

one price per zone, per quarter-hour

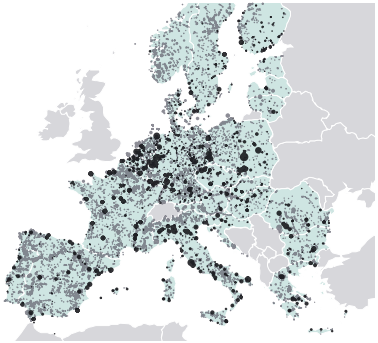
One algorithm sets Europe's prices



Every day, Euphemia sets tomorrow's prices.

- one price per zone, per quarter-hour
- plants of 10 MW or more, by capacity

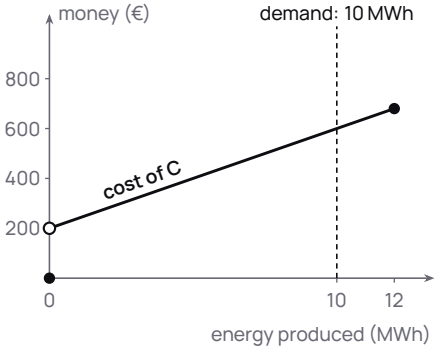
One algorithm sets Europe's prices



Every day, Euphemia sets tomorrow's prices.

- one price per zone, per quarter-hour
- plants of 10 MW or more, by capacity
- **thermal**: pays to start up

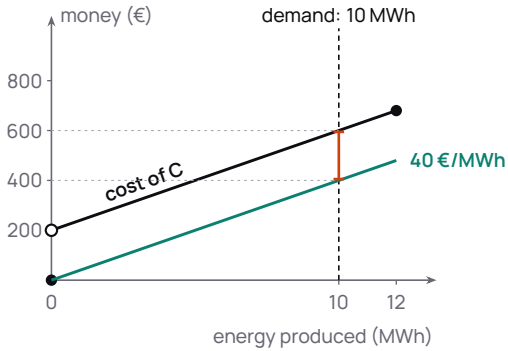
A plant that loses money



Plant C

switch on	**200 €**
each MWh	**40 €**, up to 12 MWh

A plant that loses money



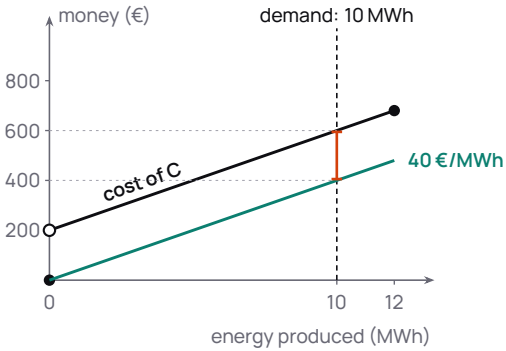
Plant C

switch on	**200 €**
each MWh	**40 €**, up to 12 MWh

At 40 €/MWh

earns	400 €
spends	600 €

A plant that loses money



Plant C

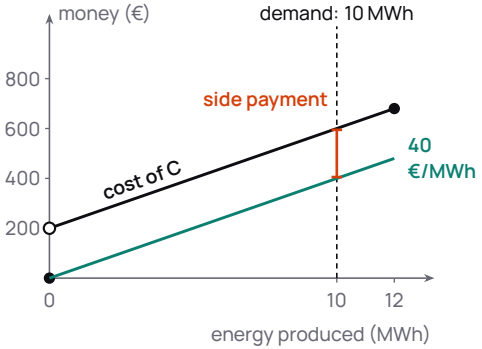
switch on	**200 €**
each MWh	**40 €**, up to 12 MWh

At 40 €/MWh

earns	400 €
spends	600 €

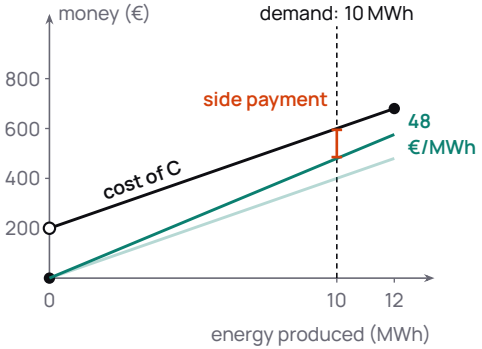
C loses 200 €: we pay a side payment, as small as possible.

Which price is best?



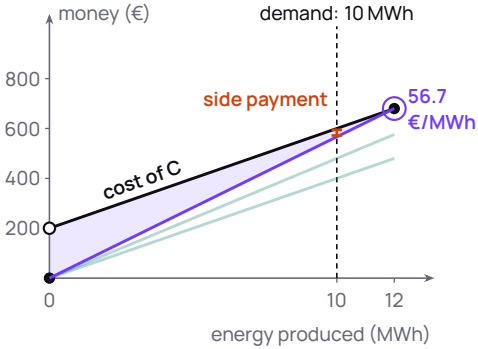


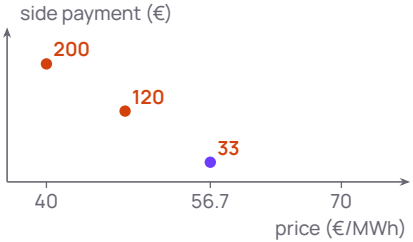
Which price is best?



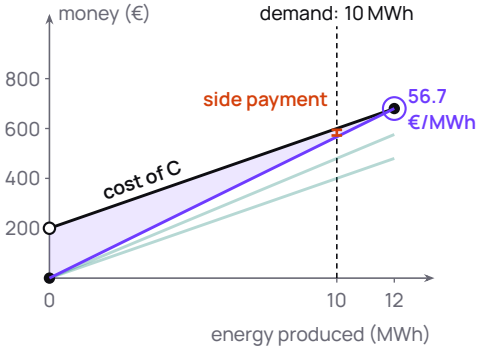


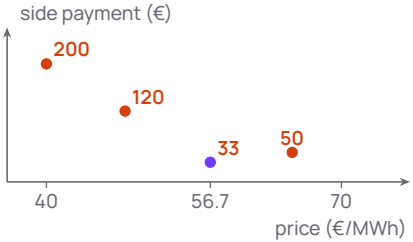
Which price is best?



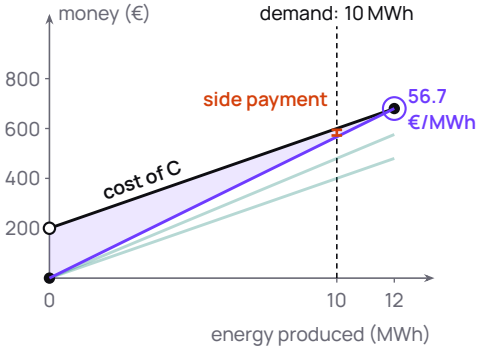


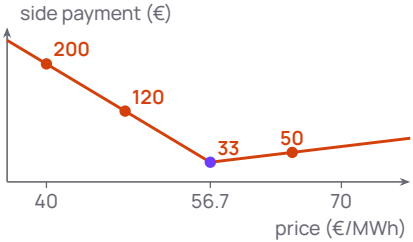
Which price is best?





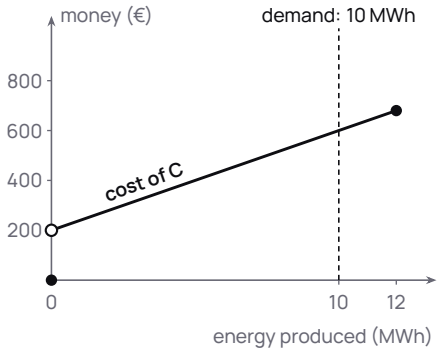
Which price is best?





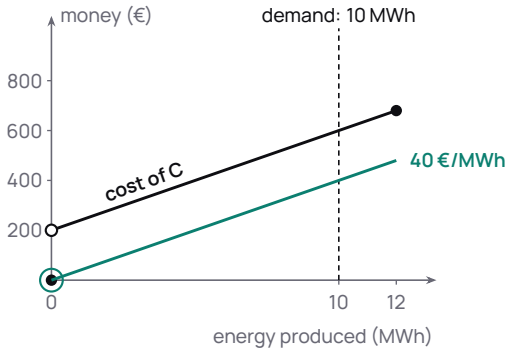
Best price: 56.7 €/MWh
the *convex hull price*

Asking each plant: the oracle



"At these prices, what would you produce?"

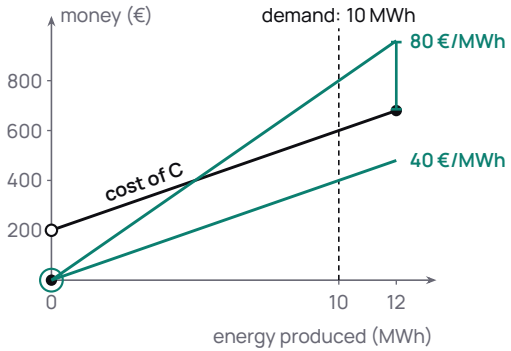
Asking each plant: the oracle



"At these prices, what would you produce?"

40 €/MWh 0 MWh raise the price

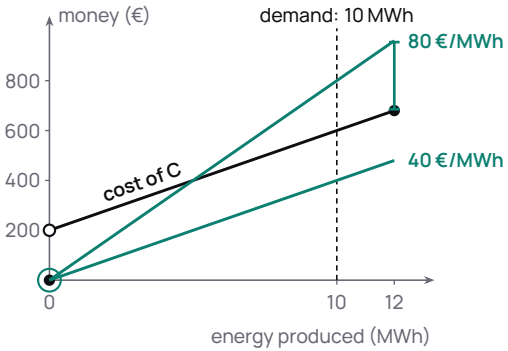
Asking each plant: the oracle



"At these prices, what would you produce?"

40 €/MWh	0 MWh	raise the price
80 €/MWh	12 MWh	lower the price

Asking each plant: the oracle



"At these prices, what would you produce?"

40 €/MWh	0 MWh	raise the price
80 €/MWh	12 MWh	lower the price

production — demand: **which way to move each price**

What's the cost of an oracle call?

In snacks

One point, one line



A few operations

Here

A day's schedule, per plant

What's the cost of an oracle call?

In snacks

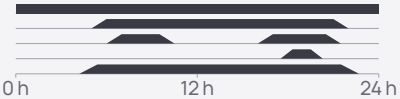
One point, one line



A few operations

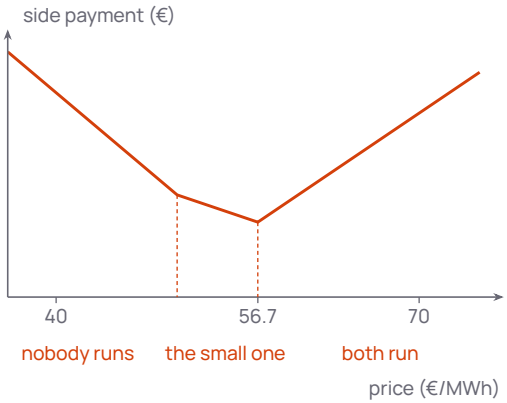
Here

A day's schedule, per plant

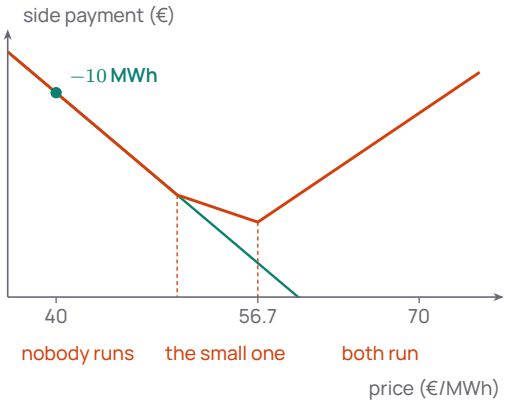


Far longer, more expensive

Keep every answer: a model



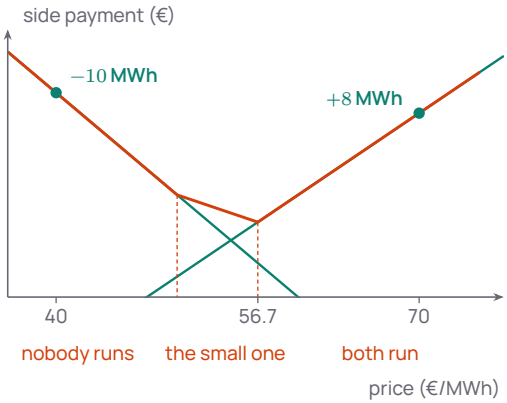
Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

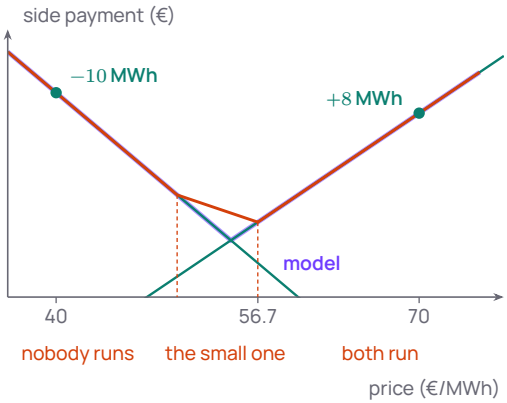
Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

Keep every answer: a model



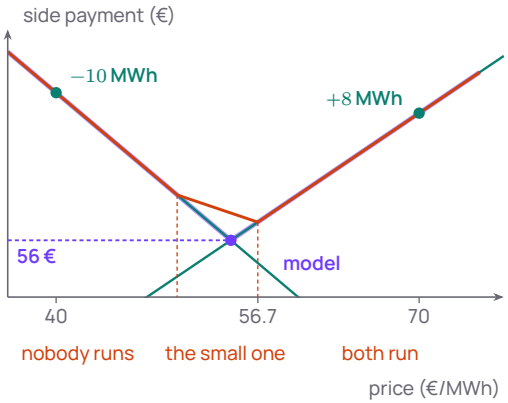
Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

Keep every line: the model.

$$m(\pi) = \max_i \left(f(\pi_i) + g_i\,(\pi - \pi_i)\right)$$

Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

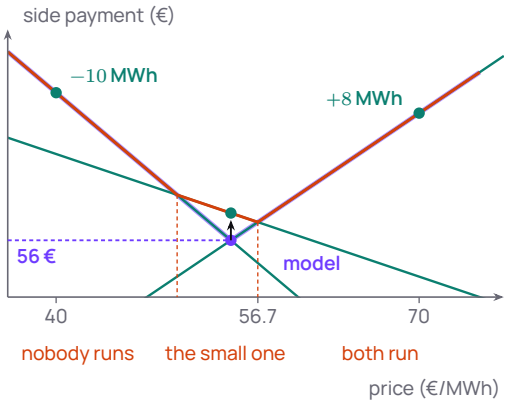
Keep every line: the model.

$$m(\pi) = \max_i \left(f(\pi_i) + g_i\,(\pi - \pi_i)\right)$$

Its lowest point: a proved lower bound.

oracle calls: **2** best so far: **180 €**
lower bound: **56 €** gap: **124 €**

Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

Keep every line: the model.

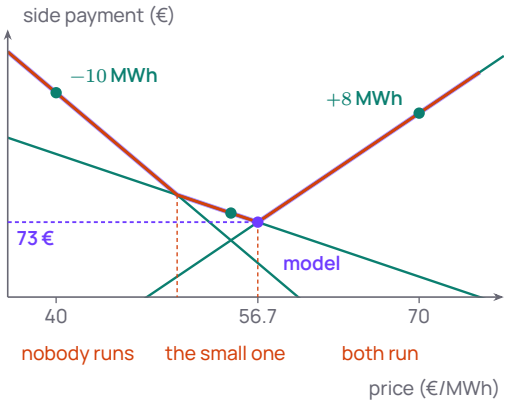
$$m(\pi) = \max_i \left(f(\pi_i) + g_i\,(\pi - \pi_i)\right)$$

next price: $\pi_{k+1} = \operatorname{argmin}_\pi m(\pi)$

Its lowest point: a proved lower bound.

oracle calls: **3** best so far: **82 €**
lower bound: **56 €** gap: **26 €**

Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

Keep every line: the model.

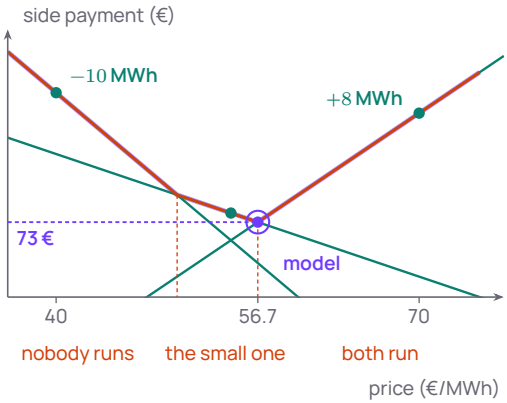
$$m(\pi) = \max_i \left(f(\pi_i) + g_i\,(\pi - \pi_i)\right)$$

$$\text{next price: } \pi_{k+1} = \operatorname{argmin}_\pi\, m(\pi)$$

Its lowest point: a proved lower bound.

oracle calls: **3** best so far: **82 €**
lower bound: **73 €** gap: **9 €**

Keep every answer: a model



Each answer: a line below f.

$$f(\pi) \geq f(\pi_i) + g_i\,(\pi - \pi_i)$$

Keep every line: the model.

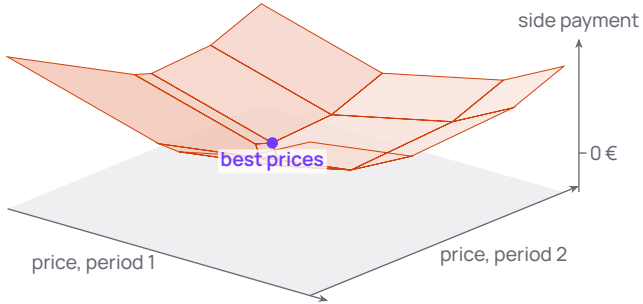
$$m(\pi) = \max_i \left(f(\pi_i) + g_i\,(\pi - \pi_i)\right)$$

$$\text{next price: } \pi_{k+1} = \operatorname{argmin}_\pi\, m(\pi)$$

Its lowest point: a proved lower bound.

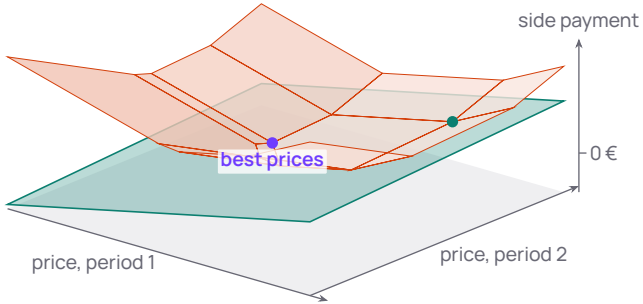
oracle calls: **4** best so far: **73 €**
lower bound: **73 €** gap: **0**, optimal

The model can promise too much



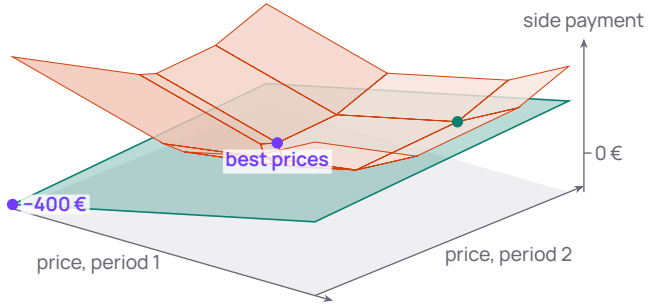
Two periods: the side payment is a **valley**.

The model can promise too much



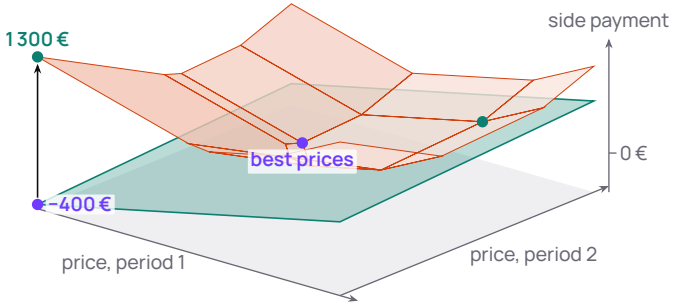
Each answer: a plane below the valley.

The model can promise too much



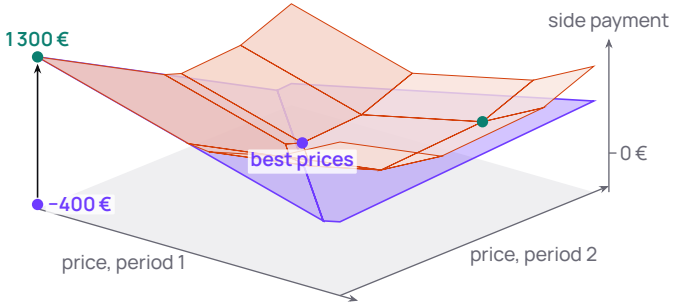
The plane's lowest point: a corner, −400 €.

The model can promise too much



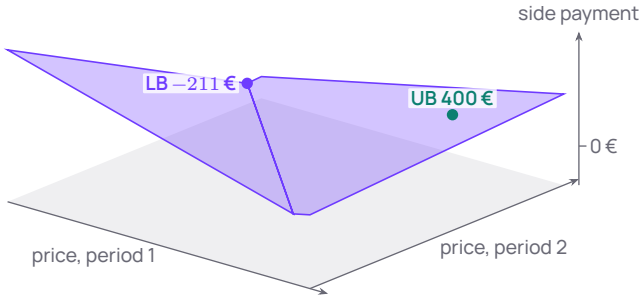
The answer there: 1300 €.

The model can promise too much



Keep both planes: the model.

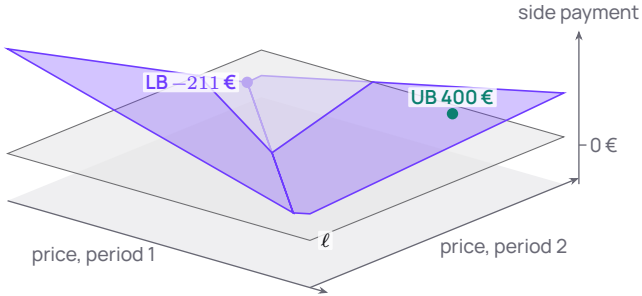
A level between the bounds



UB: the best answer.

LB: the model's lowest point.

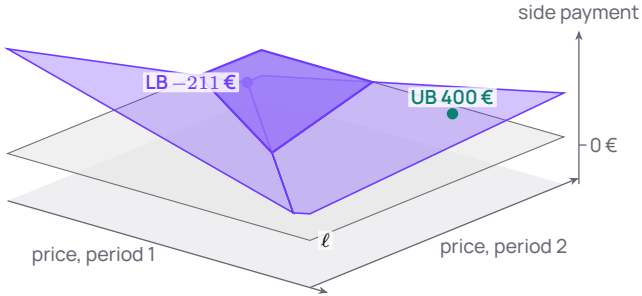
A level between the bounds



A level in between:

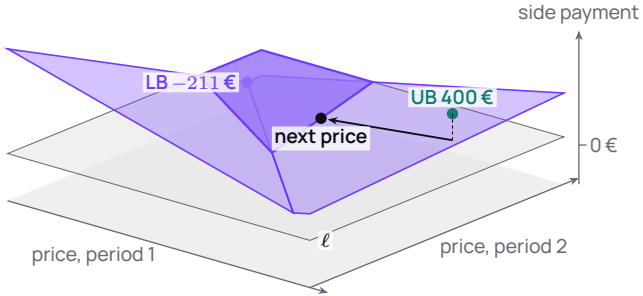
$\ell = \mathrm{LB} + \alpha\,(\mathrm{UB} - \mathrm{LB})$.

A level between the bounds



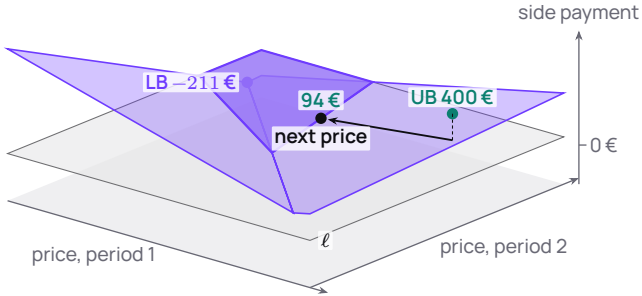
Candidates: the only prices whose side payment can be below ℓ.

A level between the bounds



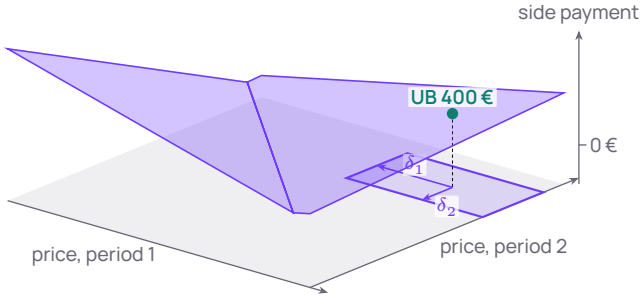
Next price: the candidate **closest** to the UB price.

A level between the bounds



The answer there: **94 €**, not 1 300 €.

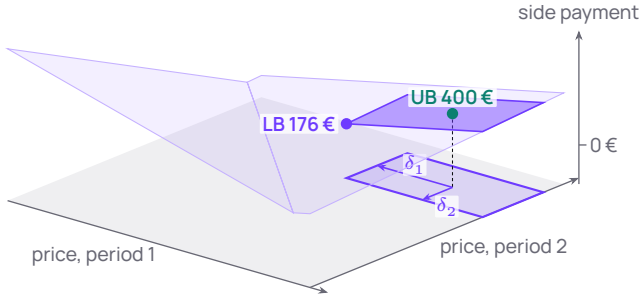
An adaptive box around the best price



A box around the best price, one width per period:

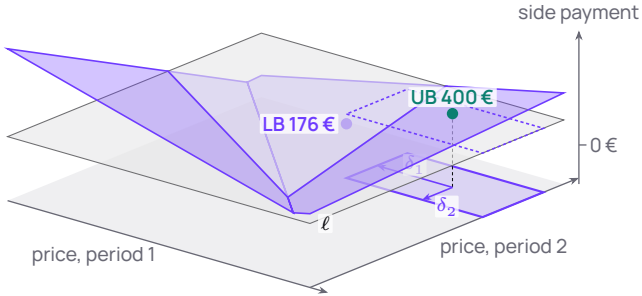
$|\pi_t - \hat{\pi}_t| \leq \delta_t$.

An adaptive box around the best price



LB: the model's lowest point in the box.

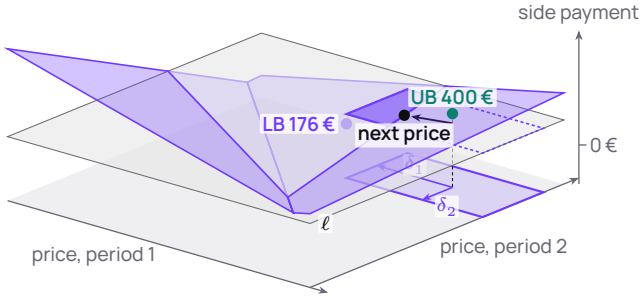
An adaptive box around the best price



The level rises:

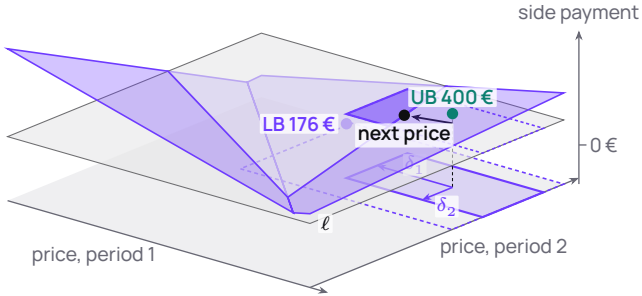
$\ell = \mathrm{LB} + \alpha\,(\mathrm{UB} - \mathrm{LB})$.

An adaptive box around the best price



Next price: the **closest** one in the box, below the level.

An adaptive box around the best price



At the edge, better: twice as wide.

At the edge three times, no better: half as wide.

Our method: PC-BPLM

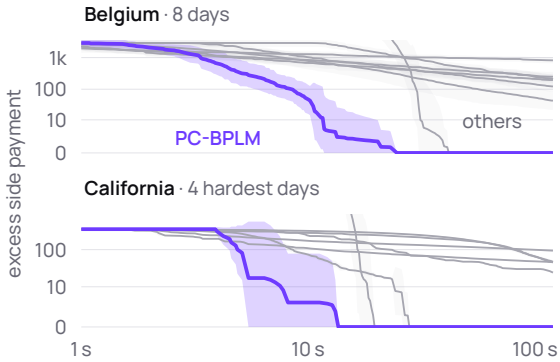
Algorithm PC-BPLM

Given: $\pi_1,\ \Delta = \ell = +\infty$

for $k = 1, 2, \ldots$ until $\mathrm{UB} - \mathrm{LB} \le \varepsilon$

- ask the oracle at π_k: $f(\pi_k)$ and g_k — polyhedral lower model
- $m(\pi) \leftarrow \max_{i \le k}(f(\pi_i) + g_i\,(\pi - \pi_i))$
- $\hat{\pi} \leftarrow$ best π_i so far, $\mathrm{UB} \leftarrow f(\hat{\pi})$
- $B \leftarrow$ box around $\hat{\pi}$, half-width δ_t in each period t — adaptive box
- $\mathrm{LB} \leftarrow \min_B m$
- **if** $\mathrm{UB} - \mathrm{LB} < (1-\alpha)\,\Delta$: $\Delta \leftarrow \mathrm{UB} - \mathrm{LB},\ \ell \leftarrow \mathrm{LB} + \alpha\,\Delta$ — level-set stabilization
- **else:** $\ell \leftarrow \min\{\ell,\ \mathrm{LB} + \alpha\,(\mathrm{UB} - \mathrm{LB})\}$
- $\pi_{k+1} \leftarrow$ projection of $\hat{\pi}$ onto $\{\pi \in B : m(\pi) \le \ell\}$

Faster than every other method



	Belgium	California
PC-BPLM	6.7	6.5
next best	36.3	7.0
others	≥ 79.6	8–70

seconds to within 0.001 %

Two principled frameworks

Part I first-order methods

snacks

ConvexHullPricing.jl

Part II principled frameworks

Argo.jl

a structured problem
→ certified methods

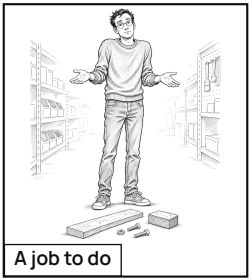
badr

any fairness metric
→ the fairest Pareto-efficient model

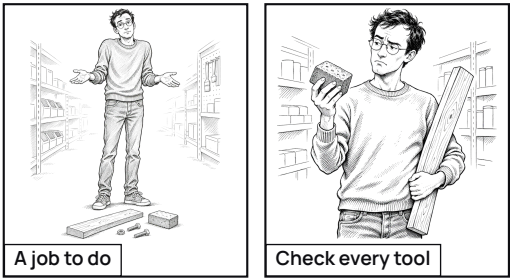
Argo.jl

Guarantee-based recommendation of first-order methods
with François Glineur

Which tool for my job?



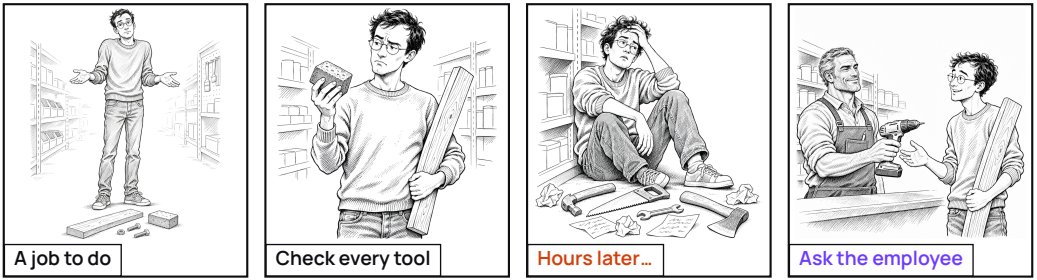
Which tool for my job?



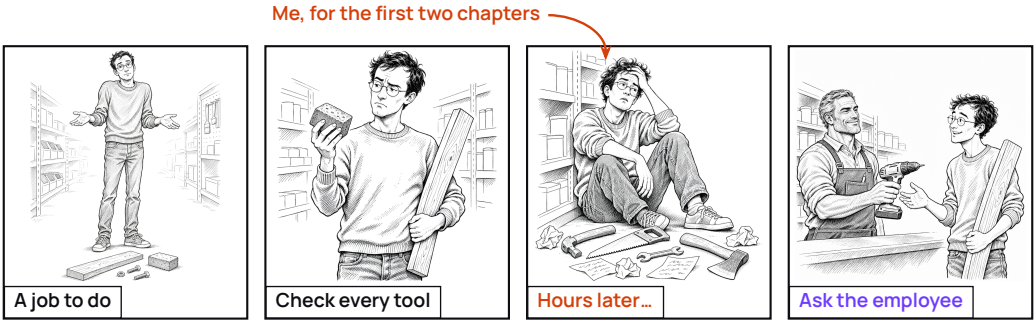
Which tool for my job?



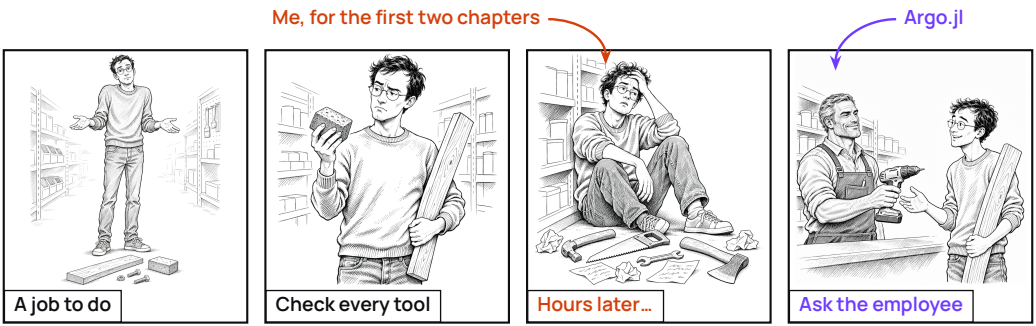
Which tool for my job?



Which tool for my job?



Which tool for my job?



Three questions

$$F(x) = \ell(x) + \lambda_1 \|x\|_1 + \tfrac{\lambda_2}{2}\|x\|^2$$

- Davis–Yin
- Frank–Wolfe

- FISTA
- SC-FISTA
- SAPG

Three questions

Q1 · which methods apply?

$$F(x) = \ell(x) + \lambda_1 \|x\|_1 + \tfrac{\lambda_2}{2}\|x\|^2$$

- Davis–Yin
- Frank–Wolfe
- FISTA
- SC-FISTA
- SAPG

Three questions

Q1 · which methods apply?

$$F(x) = \ell(x) + \lambda_1 \|x\|_1 + \tfrac{\lambda_2}{2}\|x\|^2$$

- Davis–Yin
- Frank–Wolfe

$\equiv$ Q2 · which other writings?

$$\left[\ell(x) + \tfrac{\lambda_2}{2}\|x\|^2\right] + \lambda_1 \|x\|_1$$

- FISTA
- SC-FISTA
- SAPG

Three questions

Q1 · which methods apply?

$$F(x) = \ell(x) + \lambda_1 \|x\|_1 + \tfrac{\lambda_2}{2}\|x\|^2$$

- Davis–Yin
- Frank–Wolfe

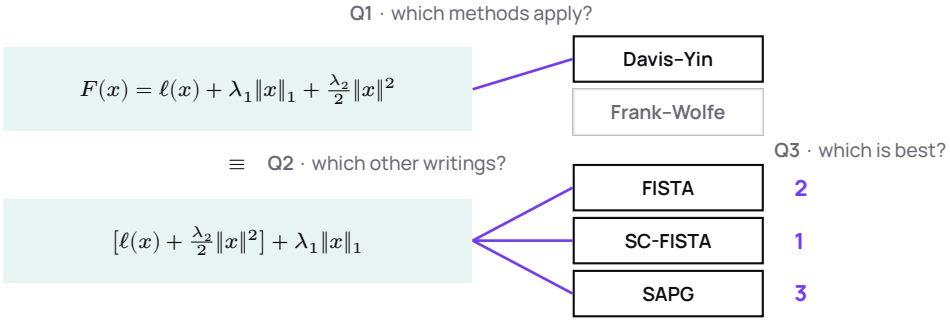
$\equiv$ Q2 · which other writings?

$$[\ell(x) + \tfrac{\lambda_2}{2}\|x\|^2] + \lambda_1 \|x\|_1$$

Q3 · which is best?

- FISTA — 2
- SC-FISTA — 1
- SAPG — 3

Three questions



☐ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

Every theorem has the same skeleton

Template problem	Method	Guarantee

- [ ] Q1 · which methods apply?
- [ ] Q2 · which other writings?
- [ ] Q3 · which is best?

Every theorem has the same skeleton

Template problem	Method	Guarantee
$F = f + g$ f convex, smooth(L), ∇ g convex, prox	FISTA step $1/L$	$F(x_k) - F^\star \le \dfrac{2L\,\lVert x_0 - x^\star\rVert^2}{(k+1)^2}$

- [ ] Q1 · which methods apply?
- [ ] Q2 · which other writings?
- [ ] Q3 · which is best?

Every theorem has the same skeleton

Template problem	Method	Guarantee
$F = f + g$ f convex, smooth(L), ∇ g convex, prox	**FISTA** step $1/L$	$F(x_k) - F^\star \leq \frac{2L\|x_0 - x^\star\|^2}{(k+1)^2}$

- [ ] **Q1** · which methods apply?
- [ ] Q2 · which other writings?
- [ ] Q3 · which is best?

Every theorem has the same skeleton

Template problem

$F = f + g$

f convex, smooth(L) ∇

g convex, prox

f smooth(L) ∇, strongly convex(μ)

g convex, prox

Method

FISTA

step $1/L$

SC-FISTA

Guarantee

$$F(x_k) - F^\star \leq \frac{2L\|x_0 - x^\star\|^2}{(k+1)^2}$$

$$\mathcal{E}_0 = F(x_0) - F^\star + \tfrac{\mu}{2}\|x_0 - x^\star\|^2$$

	$\|x_0 - x^\star\|^2$	$\mathcal{E}_0$	$F(x_0) - F^\star$
$F(x_k) - F^\star$	●	●	●
$\|x_k - x^\star\|^2$	●	●	
$\|G(x_k)\|^2$	●	●	

☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A problem is a labelled tree

$$F(x) = \ell(x) + \underbrace{\lambda_1 \|x\|_1}_{r_1(x)} + \underbrace{\frac{\lambda_2}{2}\|x\|^2}_{r_2(x)}$$

Argo.jl

```
x = variable(:x; shape=[d])
ℓ = logistic_loss(A, b;
        mean=true)(x)
r₁ = l1(λ₁; n=d)(x)
r₂ = λ₂ * sum_squares()(x)
problem = minimize(ℓ + r₁ + r₂)
```

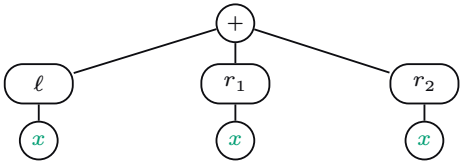
☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A problem is a labelled tree

$$F(x) = \ell(x) + \underbrace{\lambda_1 \|x\|_1}_{r_1(x)} + \underbrace{\frac{\lambda_2}{2}\|x\|^2}_{r_2(x)}$$

Argo.jl

```
x = variable(:x; shape=[d])
ℓ = logistic_loss(A, b;
        mean=true)(x)
r₁ = l1(λ₁; n=d)(x)
r₂ = λ₂ * sum_squares()(x)
problem = minimize(ℓ + r₁ + r₂)
```



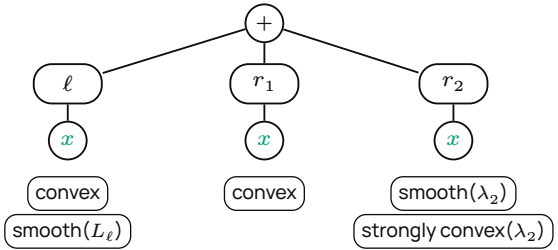
☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A problem is a labelled tree

$$F(x) = \ell(x) + \underbrace{\lambda_1 \|x\|_1}_{r_1(x)} + \underbrace{\frac{\lambda_2}{2}\|x\|^2}_{r_2(x)}$$

Argo.jl

```
x = variable(:x; shape=[d])
ℓ = logistic_loss(A, b;
        mean=true)(x)
r₁ = l1(λ₁; n=d)(x)
r₂ = λ₂ * sum_squares()(x)
problem = minimize(ℓ + r₁ + r₂)
```



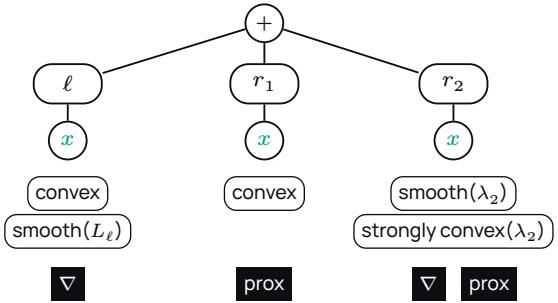
☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A problem is a labelled tree

$$F(x) = \ell(x) + \underbrace{\lambda_1 \|x\|_1}_{r_1(x)} + \underbrace{\frac{\lambda_2}{2}\|x\|^2}_{r_2(x)}$$

Argo.jl

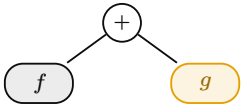
```
x = variable(:x; shape=[d])
ℓ = logistic_loss(A, b;
        mean=true)(x)
r₁ = l1(λ₁; n=d)(x)
r₂ = λ₂ * sum_squares()(x)
problem = minimize(ℓ + r₁ + r₂)
```

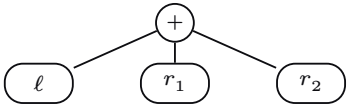


☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A writing against a theorem

SC-FISTA





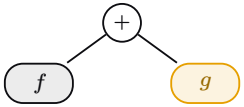
f smooth(L) strongly convex(μ) ∇

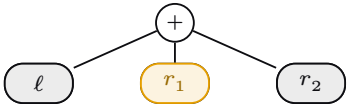
g convex prox

☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A writing against a theorem

SC-FISTA





f smooth(L) strongly convex(μ) ∇

g convex prox

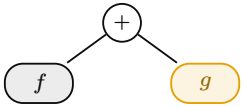
smooth($L_\ell + \lambda_2$) strongly convex(λ_2) ∇

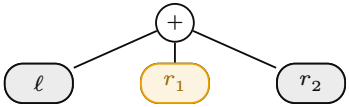
convex prox

- [ ] Q1 · which methods apply?
- [ ] Q2 · which other writings?
- [ ] Q3 · which is best?

A writing against a theorem

SC-FISTA



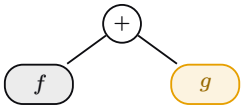


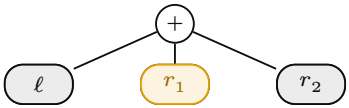
f	smooth(L) strongly convex(μ) ∇	✓ smooth$(L_\ell + \lambda_2)$ strongly convex(λ_2) ∇
g	convex prox	✓ convex prox

- [ ] Q1 · which methods apply?
- [ ] Q2 · which other writings?
- [ ] Q3 · which is best?

A writing against a theorem

SC-FISTA





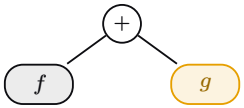
f	smooth(L) strongly convex(μ) ∇	✓ smooth$(L_\ell + \lambda_2)$ strongly convex(λ_2) ∇
g	convex prox	✓ convex prox

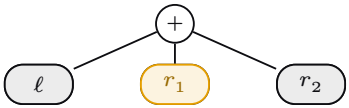
$[\ell + r_2] + [r_1]$ $\quad F(x_k) - F^\star \le \frac{L_\ell + \lambda_2}{2}(1+\sqrt{q})(1-\sqrt{q})^k \|x_0 - x^\star\|^2$ $\quad q = \lambda_2/(L_\ell + \lambda_2)$

☐ Q1 · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

A writing against a theorem

SC-FISTA





f	smooth(L) · strongly convex(μ) · ∇	✓ smooth$(L_\ell + \lambda_2)$ · strongly convex(λ_2) · ∇
g	convex · prox	✓ convex · prox

$[\ell + r_2] + [r_1]$ $\quad F(x_k) - F^\star \le \frac{L_\ell + \lambda_2}{2}(1+\sqrt{q})(1-\sqrt{q})^k \|x_0 - x^\star\|^2$ $\quad q = \lambda_2/(L_\ell + \lambda_2)$

☑ **Q1** · which methods apply? → match the tree against each theorem

One problem, several exact decompositions

	FISTA	SC-FISTA	SAPG
$[\ell] + [r_1 + r_2]$			
$[r_1] + [\ell + r_2]$			
$[r_2] + [\ell + r_1]$			
$[\ell + r_1] + [r_2]$			
$[\ell + r_2] + [r_1]$			
$[r_1 + r_2] + [\ell]$			

☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

One problem, several exact decompositions

		FISTA	SC-FISTA	SAPG
$[\ell] + [r_1 + r_2]$				
$[r_1] + [\ell + r_2]$	×			
$[r_2] + [\ell + r_1]$	×			
$[\ell + r_1] + [r_2]$	×			
$[\ell + r_2] + [r_1]$				
$[r_1 + r_2] + [\ell]$	×			

☑ **Q1** · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

One problem, several exact decompositions

		FISTA	SC-FISTA	SAPG
$[\ell] + [r_1 + r_2]$		$L = L_\ell$		$L = L_\ell$
$[r_1] + [\ell + r_2]$	×			
$[r_2] + [\ell + r_1]$	×			
$[\ell + r_1] + [r_2]$	×			
$[\ell + r_2] + [r_1]$		$L = L_\ell + \lambda_2$	$\mu = \lambda_2$	$L = L_\ell + \lambda_2$
$[r_1 + r_2] + [\ell]$	×			

☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

One problem, several exact decompositions

		FISTA	SC-FISTA	SAPG
$[\ell] + [r_1 + r_2]$		$L = L_\ell$		$L = L_\ell$
$[r_1] + [\ell + r_2]$	×			
$[r_2] + [\ell + r_1]$	×			
$[\ell + r_1] + [r_2]$	×			
$[\ell + r_2] + [r_1]$		$L = L_\ell + \lambda_2$	$\mu = \lambda_2$	$L = L_\ell + \lambda_2$
$[r_1 + r_2] + [\ell]$	×			

☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

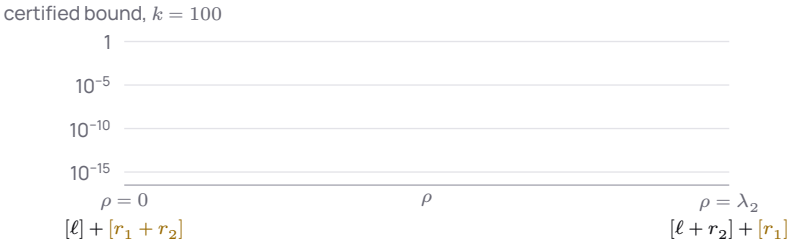
Rewriting: a family of writings

$$F = [\ell + \tfrac{\rho}{2}\|x\|^2] + [r_1 + \tfrac{\lambda_2 - \rho}{2}\|x\|^2], \qquad 0 \le \rho \le \lambda_2$$

☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

Rewriting: a family of writings

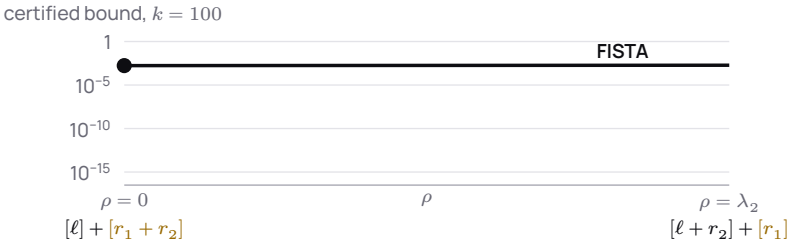
$$F = [\ell + \tfrac{\rho}{2}\|x\|^2] + [r_1 + \tfrac{\lambda_2 - \rho}{2}\|x\|^2], \qquad 0 \le \rho \le \lambda_2$$



☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

Rewriting: a family of writings

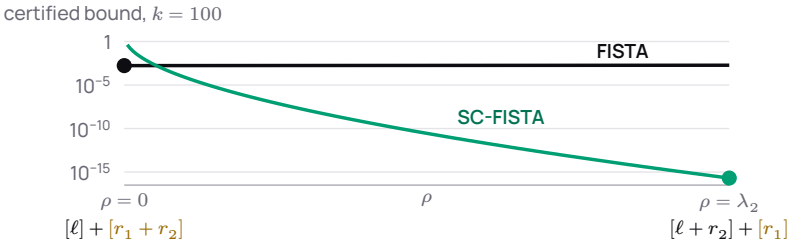
$$F = [\ell + \tfrac{\rho}{2}\|x\|^2] + [r_1 + \tfrac{\lambda_2 - \rho}{2}\|x\|^2], \qquad 0 \le \rho \le \lambda_2$$



☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

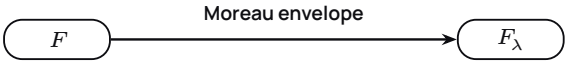
Rewriting: a family of writings

$$F = [\ell + \tfrac{\rho}{2}\|x\|^2] + [r_1 + \tfrac{\lambda_2 - \rho}{2}\|x\|^2], \qquad 0 \le \rho \le \lambda_2$$



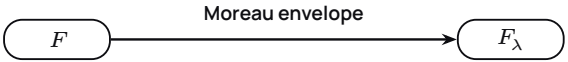
☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ **Q3** · which is best?

When are two writings the same?



- [x] **Q1** · which methods apply?
- [ ] **Q2** · which other writings?
- [ ] **Q3** · which is best?

When are two writings the same?

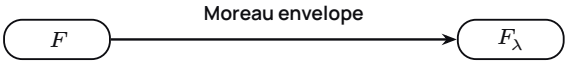


Guarantee transfer

$F(y) - F^\star \leq \varepsilon \quad \longleftarrow \quad F_\lambda(x) - F^\star \leq \varepsilon$

☑ **Q1** · which methods apply? ☐ **Q2** · which other writings? ☐ Q3 · which is best?

When are two writings the same?

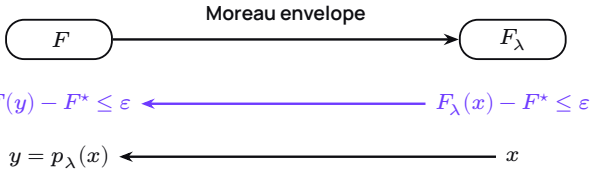


Guarantee transfer $F(y) - F^\star \leq \varepsilon$ ← $F_\lambda(x) - F^\star \leq \varepsilon$

Output map $y = p_\lambda(x)$ ← x

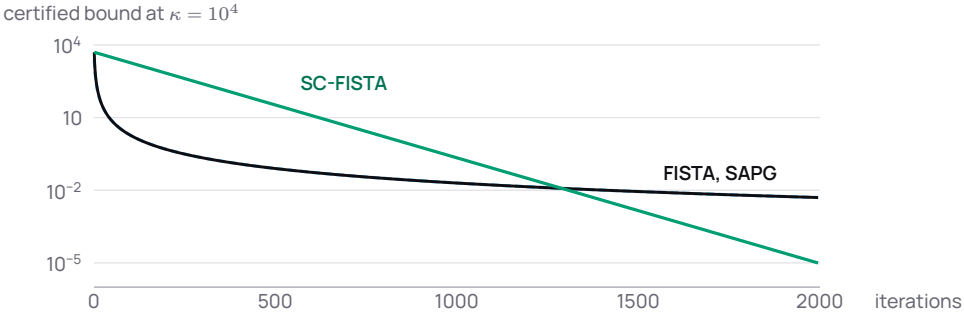
☑ **Q1** · which methods apply? ☐ Q2 · which other writings? ☐ Q3 · which is best?

When are two writings the same?



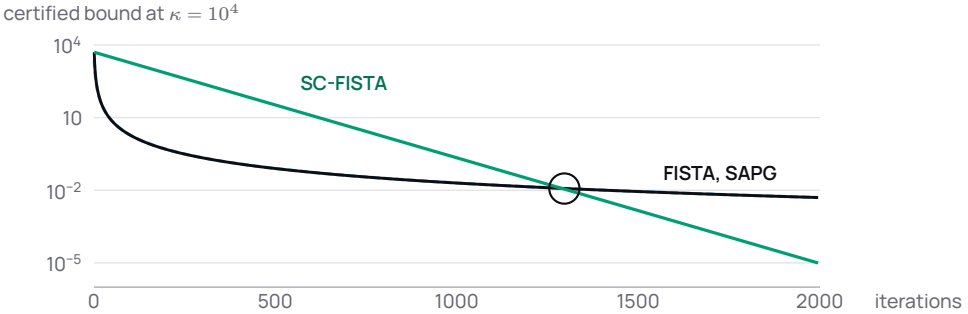
☑ **Q2** · which other writings? → calculus and reformulation tricks

Rank the certified bounds



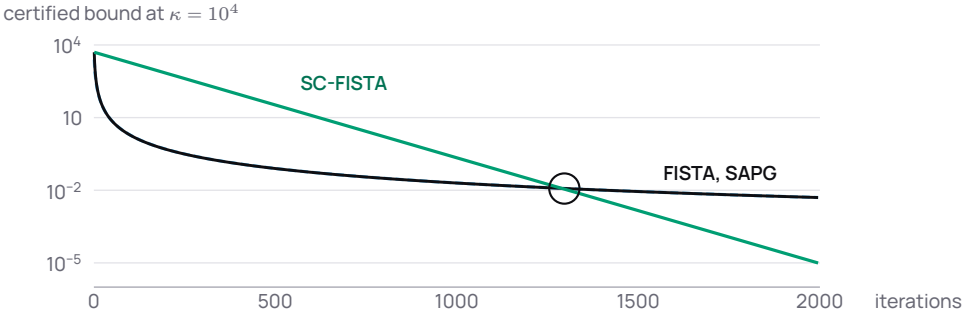
☑ **Q1** · which methods apply? ☑ **Q2** · which other writings? ☐ **Q3** · which is best?

Rank the certified bounds



☑ **Q1** · which methods apply? ☑ **Q2** · which other writings? ☐ **Q3** · which is best?

Rank the certified bounds



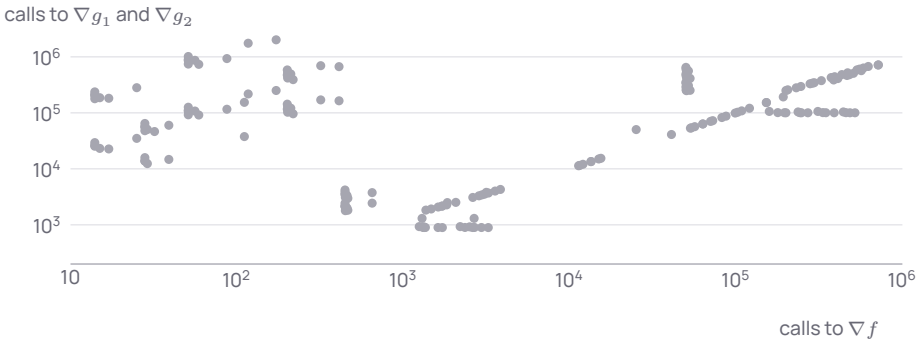
☑ **Q3** · which is best? → it depends on your budget

When costs are unknown, a front

$\min_x \; f(x) + g_1(x) + g_2(x)$

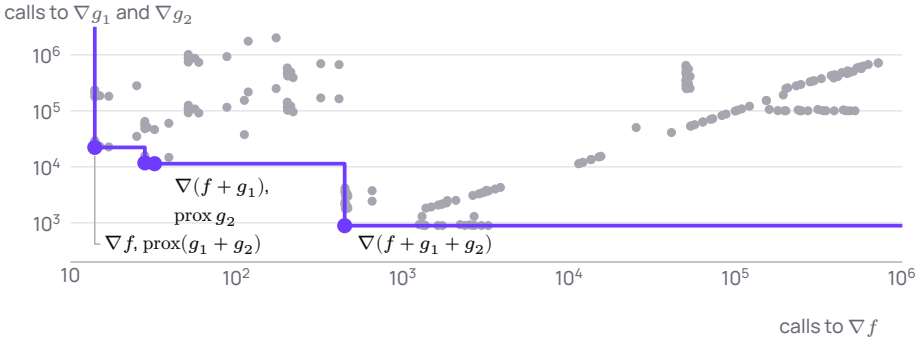
When costs are unknown, a front

$\min_x \; f(x) + g_1(x) + g_2(x)$



When costs are unknown, a front

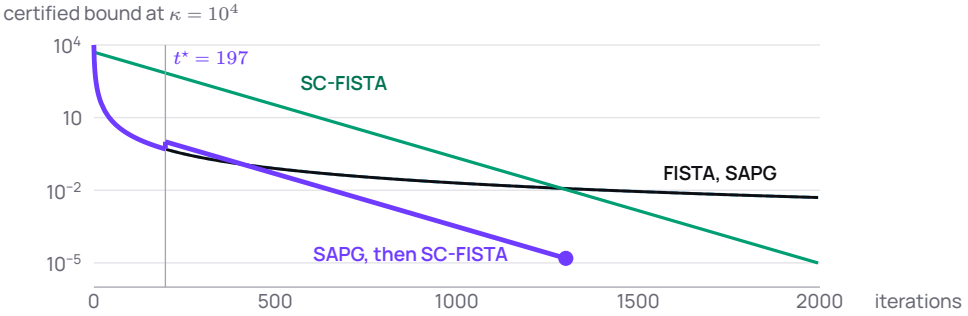
$\min_x \; f(x) + g_1(x) + g_2(x)$



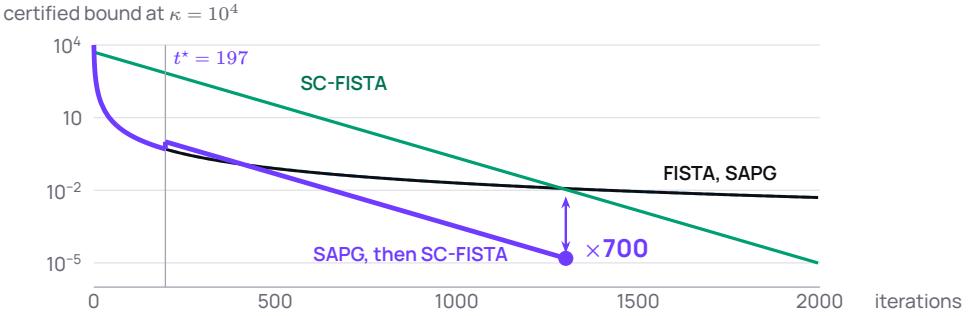
Playing with the certificates

- ☑ **Q1** · which methods apply? → match the tree against each theorem
- ☑ **Q2** · which other writings? → calculus and reformulation tricks
- ☑ **Q3** · which is best? → it depends on your budget

Chain two methods



Chain two methods



badr

Fairness-informed Pareto optimization
with Yassine Laguel and Samuel Vaiter

snacks | ConvexHullPricing.jl | Argo.jl | **badr**

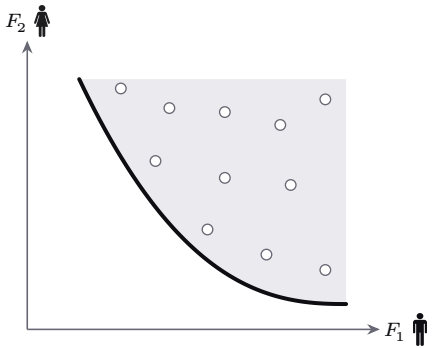
AI already decides about people



> “Dutch scandal serves as a warning for Europe over risks of using algorithms”
>
> POLITICO Europe, on the Dutch childcare benefits scandal

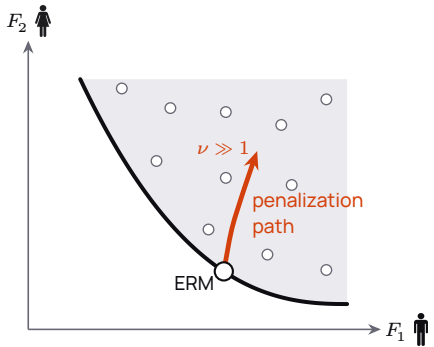
Penalizing unfairness is not efficient

$$\min_{\theta} \text{average loss}(\theta) + \nu \mathcal{U}(\theta)$$



Penalizing unfairness is not efficient

$$\min_{\theta} \text{ average loss}(\theta) + \nu \mathcal{U}(\theta)$$

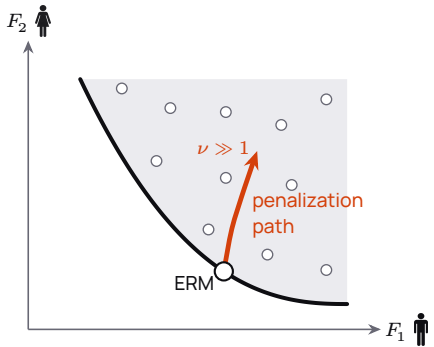


BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

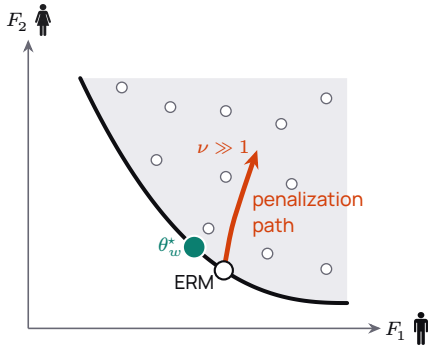


BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

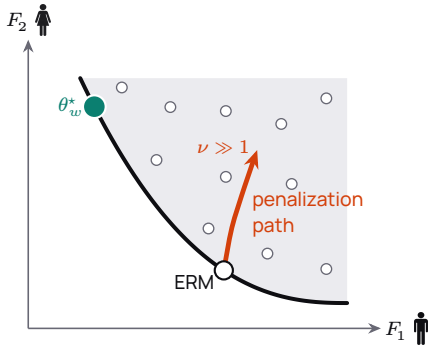


BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

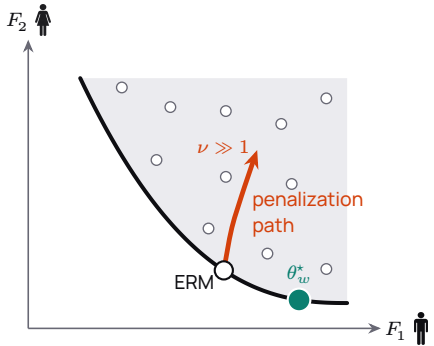


BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1



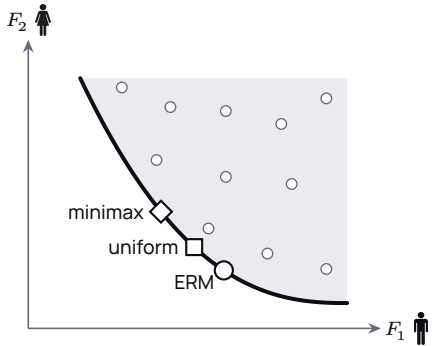
BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

- ERM $w_a = n_a/n$
- uniform $w_a = 1/S$
- minimax $\min_\theta \max_a F_a(\theta)$



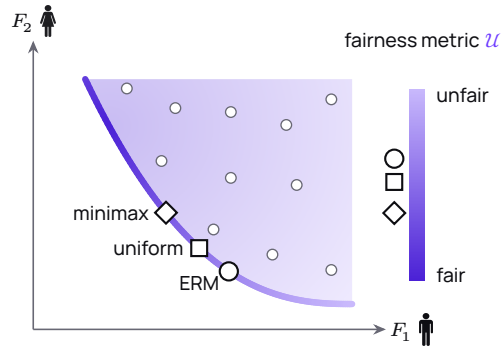
BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

ERM	$w_a = n_a/n$
uniform	$w_a = 1/S$
minimax	$\min_\theta \max_a F_a(\theta)$



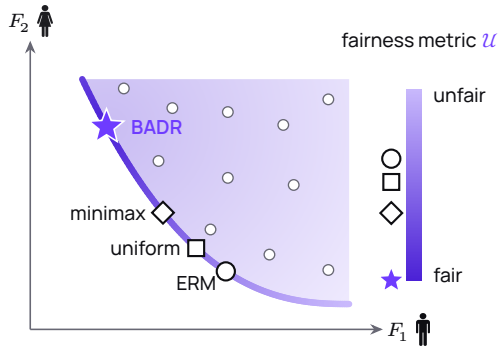
BADR: the fairest model on the front

$$\min_{\theta} \sum_{a} w_a F_a(\theta), \qquad w \in \Delta$$

w_1

0 1

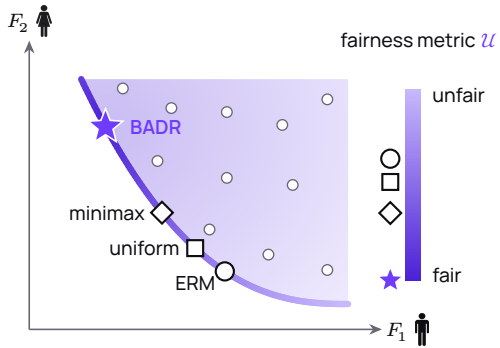
ERM	$w_a = n_a/n$
uniform	$w_a = 1/S$
minimax	$\min_\theta \max_a F_a(\theta)$



BADR: the fairest model on the front

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$



The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)])$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)])$$

Chain rule

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)])$$

Chain rule, then implicit function theorem

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

$$\partial_{w_a} \theta^\star_w = -H_w^{-1} \nabla F_a(\theta^\star_w), \; H_w = \textstyle\sum_a w_a \nabla^2 F_a(\theta^\star_w)$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)])$$

Chain rule, then implicit function theorem

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

$$\partial_{w_a} \theta^\star_w = -H_w^{-1} \nabla F_a(\theta^\star_w), \; H_w = \textstyle\sum_a w_a \nabla^2 F_a(\theta^\star_w)$$

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla F_a(\theta^\star_w)^\top v, \; v = -H_w^{-1} \nabla_\theta \mathcal{U}(\theta^\star_w)$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)])$$

Chain rule, then implicit function theorem

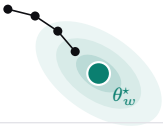
$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

$$\partial_{w_a} \theta^\star_w = -H_w^{-1} \nabla F_a(\theta^\star_w), \; H_w = \textstyle\sum_a w_a \nabla^2 F_a(\theta^\star_w)$$

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla F_a(\theta^\star_w)^\top v, \; v = -H_w^{-1} \nabla_\theta \mathcal{U}(\theta^\star_w)$$

1 $\theta^\star_w$ cannot be queried

only approximated



The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta\left(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)]\right)$$

Chain rule, then implicit function theorem

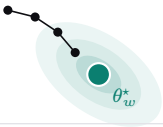
$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

$$\partial_{w_a} \theta^\star_w = -H_w^{-1} \nabla F_a(\theta^\star_w), \; H_w = \textstyle\sum_a w_a \nabla^2 F_a(\theta^\star_w)$$

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla F_a(\theta^\star_w)^\top v, \; v = -H_w^{-1} \nabla_\theta \mathcal{U}(\theta^\star_w)$$

1 $\theta^\star_w$ cannot be queried

only approximated



2 a linear system for v

$$H_w \, v = -\nabla_\theta \mathcal{U}(\theta^\star_w)$$

The implicit approach and its obstacles

$$\min_{w \in \Delta} \mathcal{U}(\theta^\star_w)$$

$$\theta^\star_w \in \operatorname*{argmin}_\theta \sum_a w_a F_a(\theta)$$

Gradient descent on the weights

$$w \leftarrow \operatorname{proj}_\Delta\left(w - \gamma \nabla_w [\mathcal{U}(\theta^\star_w)]\right)$$

Chain rule, then implicit function theorem

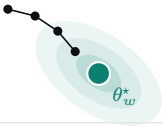
$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla_\theta \mathcal{U}(\theta^\star_w)^\top \partial_{w_a} \theta^\star_w$$

$$\partial_{w_a} \theta^\star_w = -H_w^{-1} \nabla F_a(\theta^\star_w),\ H_w = \textstyle\sum_a w_a \nabla^2 F_a(\theta^\star_w)$$

$$\partial_{w_a} \mathcal{U}(\theta^\star_w) = \nabla F_a(\theta^\star_w)^\top v,\ v = -H_w^{-1} \nabla_\theta \mathcal{U}(\theta^\star_w)$$

1 $\theta^\star_w$ cannot be queried

only approximated



2 a linear system for v

$$H_w\, v = -\nabla_\theta \mathcal{U}(\theta^\star_w)$$

3 stochastic oracles

$$\tilde{\nabla} F_a(\theta),\ \tilde{\nabla}^2 F_a(\theta)\, v,\ \tilde{\nabla}_\theta \mathcal{U}(\theta)$$

BADR-SGD: a single loop

Algorithm BADR-SGD

Given: $\theta_0,\ v_0,\ w_0$; step sizes τ, ρ, γ; threshold C

for $t = 0$ **to** $T-1$

$\quad \theta_{t+1} \leftarrow \theta_t - \tau \sum_a w_{t,a}\, \tilde{\nabla} F_a(\theta_t)$

$\quad v_{t+1} \leftarrow v_t - \rho(\tilde{\nabla}\, \mathcal{U}(\theta_t) + \sum_a w_{t,a}\, \tilde{\nabla}^2 F_a(\theta_t)\, v_t)$

$\quad w_{t+1} \leftarrow \mathrm{proj}_\Delta(w_t - \gamma\, \mathrm{clip}(\tilde{\nabla} F(\theta_t)\, v_t))$

return $\theta_T,\ w_T$

$\tilde{\nabla} F(\theta)\, v = (\tilde{\nabla} F_a(\theta)^\top v)_a$

1. $\theta_w^\star$ approximate
2. linear system for v
3. stochastic oracles

BADR-SGD: a single loop

Algorithm BADR-SGD

Given: $\theta_0,\ v_0,\ w_0$; step sizes τ, ρ, γ; threshold C

for $t = 0$ **to** $T-1$

$\quad \theta_{t+1} \leftarrow \theta_t - \tau \sum_a w_{t,a}\, \tilde{\nabla} F_a(\theta_t)$ 1

$\quad v_{t+1} \leftarrow v_t - \rho(\tilde{\nabla}\, \mathcal{U}(\theta_t) + \sum_a w_{t,a}\, \tilde{\nabla}^2 F_a(\theta_t)\, v_t)$

$\quad w_{t+1} \leftarrow \mathrm{proj}_{\Delta}(w_t - \gamma\, \mathrm{clip}(\tilde{\nabla} F(\theta_t)\, v_t))$

return $\theta_T,\ w_T$

$\tilde{\nabla} F(\theta)\, v = (\tilde{\nabla} F_a(\theta)^\top v)_a$

- ✓ 1 $\theta_w^\star$ approximate
- 2 linear system for v
- 3 stochastic oracles

BADR-SGD: a single loop

Algorithm BADR-SGD

Given: $\theta_0,\ v_0,\ w_0$; step sizes τ, ρ, γ; threshold C

for $t = 0$ **to** $T-1$

$\theta_{t+1} \leftarrow \theta_t - \tau \sum_a w_{t,a}\, \tilde{\nabla} F_a(\theta_t)$ **1**

$v_{t+1} \leftarrow v_t - \rho(\tilde{\nabla}\, \mathcal{U}(\theta_t) + \sum_a w_{t,a}\, \tilde{\nabla}^2 F_a(\theta_t)\, v_t)$ **2**

$w_{t+1} \leftarrow \mathrm{proj}_{\Delta}(w_t - \gamma\, \mathrm{clip}(\tilde{\nabla} F(\theta_t)\, v_t))$

return $\theta_T,\ w_T$

$\tilde{\nabla} F(\theta)\, v = (\tilde{\nabla} F_a(\theta)^\top v)_a$

- ✓ **1** $\theta_w^\star$ approximate
- ✓ **2** linear system for v
- **3** stochastic oracles

BADR-SGD: a single loop

Algorithm BADR-SGD

Given: $\theta_0,\ v_0,\ w_0$; step sizes τ, ρ, γ; threshold C

for $t = 0$ **to** $T-1$

$\theta_{t+1} \leftarrow \theta_t - \tau \sum_a w_{t,a}\, \tilde{\nabla} F_a(\theta_t)$ **1**

$v_{t+1} \leftarrow v_t - \rho(\tilde{\nabla}\, \mathcal{U}(\theta_t) + \sum_a w_{t,a}\, \tilde{\nabla}^2 F_a(\theta_t)\, v_t)$ **2**

$w_{t+1} \leftarrow \mathrm{proj}_\Delta(w_t - \gamma\ \mathrm{clip}(\tilde{\nabla} F(\theta_t)\, v_t))$

return $\theta_T,\ w_T$

$\tilde{\nabla} F(\theta)\, v = (\tilde{\nabla} F_a(\theta)^\top v)_a$

- ✓ **1** $\theta^\star_w$ approximate
- ✓ **2** linear system for v
- ✓ **3** stochastic oracles

  minibatch

  $\tilde{\nabla} F_a(\theta),\ \tilde{\nabla}^2 F_a(\theta)\, v$

  $\tilde{\nabla}\, \mathcal{U}(\theta)$

BADR-SGD converges

Theorem 1 Tanji, Vaiter, Laguel 2026

BADR-SGD with explicit step sizes τ, ρ, γ satisfies, for all $T \geq 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|D_t\| + \tilde{\alpha}^{1/2}\,\mathbb{E}\|\theta_t - \theta_t^\star\| \;\leq\; \mathcal{S}\Big[\frac{\text{initial conditions}}{T} + M^\top \mathcal{V} + r\,C_\gamma^2\Big]$$

BADR-SGD converges

Theorem 1 Tanji, Vaiter, Laguel 2026

BADR-SGD with explicit step sizes τ, ρ, γ satisfies, for all $T \geq 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \underbrace{\mathbb{E}\|D_t\|}_{\text{stationarity in } w} + \tilde{\alpha}^{1/2} \underbrace{\mathbb{E}\|\theta_t - \theta_t^\star\|}_{\text{lower level}} \leq \mathcal{S}\Big[\frac{\text{initial conditions}}{T} + M^\top \mathcal{V} + r\, C_\gamma^2\Big]$$

BADR-SGD converges

Theorem 1 Tanji, Vaiter, Laguel 2026

BADR-SGD with explicit step sizes τ, ρ, γ satisfies, for all $T \geq 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \underbrace{\mathbb{E}\|D_t\|}_{\text{stationarity in } w} + \tilde{\alpha}^{1/2} \underbrace{\mathbb{E}\|\theta_t - \theta_t^\star\|}_{\text{lower level}} \leq \mathcal{S}\Big[\frac{\text{initial conditions}}{T} + \underbrace{M^\top \mathcal{V}}_{\text{variances of the oracles}} + r\, C_\gamma^2\Big]$$

BADR-SGD converges

Theorem 1 Tanji, Vaiter, Laguel 2026

BADR-SGD with explicit step sizes τ, ρ, γ satisfies, for all $T \geq 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \underbrace{\mathbb{E}\|D_t\|}_{\text{stationarity in } w} + \tilde{\alpha}^{1/2} \underbrace{\mathbb{E}\|\theta_t - \theta_t^\star\|}_{\text{lower level}} \leq \mathcal{S}\Big[\frac{\text{initial conditions}}{T} + \underbrace{M^\top \mathcal{V}}_{\text{variances of the oracles}} + r\, C_\gamma^2\Big]$$

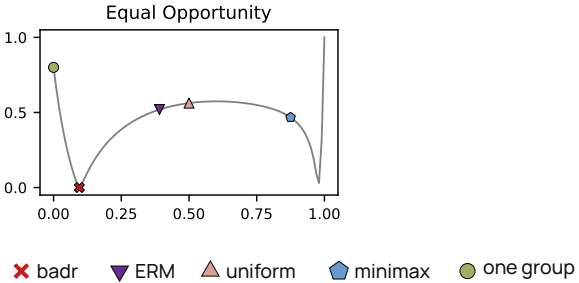
Proof: a new Lyapunov function

$$\varphi_t = \mathbb{E}[\underbrace{\mathcal{U}(\theta_t^\star)}_{\text{objective}} + \alpha \underbrace{\|\theta_t - \theta_t^\star\|^2}_{\text{model error}} + \beta \underbrace{\|v_t - v_t^\star\|^2}_{\text{error on } v} + \delta \underbrace{\|\theta_t - \theta_t^\star\|^2 \|v_t\|^2}_{\text{coupling}}]$$

BADR finds the fairest weights

Two groups

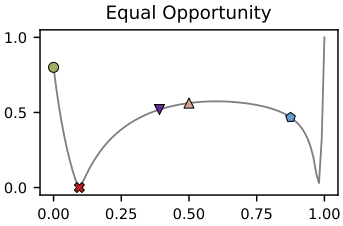
ACS Employment, DE 2014



BADR finds the fairest weights

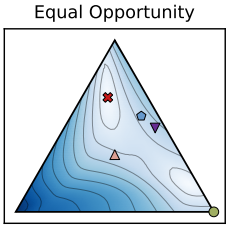
Two groups

ACS Employment, DE 2014



Three groups

ACS Employment, WV 2016

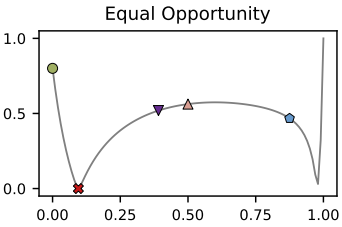


badr ERM uniform minimax one group

BADR finds the fairest weights

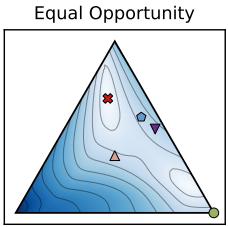
Two groups

ACS Employment, DE 2014



Three groups

ACS Employment, WV 2016



badr ERM uniform minimax one group

badr

an open-source Python toolbox

Datasets

8 public datasets (ACS, COMPAS, etc.), or any dataframe

Fairness metrics

11 (demographic parity, equal opportunity, etc.)

Learning models

3 (logistic regression, ridge regression, smoothed SVM)

scikit-learn interface

Conclusion

We made four contributions towards letting the structure of a problem guide the design and the selection of first-order methods.

Conclusion

We made four contributions towards letting the structure of a problem guide the design and the selection of first-order methods.

1 **snacks**
a solver for large-scale kernel SVMs, the fastest or tied on every benchmark

Conclusion

We made four contributions towards letting the structure of a problem guide the design and the selection of first-order methods.

1. **snacks**
   a solver for large-scale kernel SVMs, the fastest or tied on every benchmark
2. **ConvexHullPricing.jl**
   a bundle method for convex hull pricing, the fastest within market time limits

Conclusion

We made four contributions towards letting the structure of a problem guide the design and the selection of first-order methods.

1. **snacks**
   a solver for large-scale kernel SVMs, the fastest or tied on every benchmark
2. **ConvexHullPricing.jl**
   a bundle method for convex hull pricing, the fastest within market time limits
3. **Argo.jl**
   a modeling language that finds and ranks certified methods for a problem and its rewritings

Conclusion

We made four contributions towards letting the structure of a problem guide the design and the selection of first-order methods.

1. **snacks**
   a solver for large-scale kernel SVMs, the fastest or tied on every benchmark
2. **ConvexHullPricing.jl**
   a bundle method for convex hull pricing, the fastest within market time limits
3. **Argo.jl**
   a modeling language that finds and ranks certified methods for a problem and its rewritings
4. **badr**
   the fairest Pareto-efficient model for any fairness metric, computed at the optimal rate

Thank you


François **Glineur**


Silvia **Villa**


Andrea **Della Vecchia**


Yassine **Kamri**


Mehdi **Madani**


Yassine **Laguel**


Samuel **Vaiter**

UCLouvain

TraDE-OPT
TRAINING DATA-DRIVEN EXPERTS IN OPTIMIZATION
MSCA-ITN 2019

MaLGa
MACHINE LEARNING GENOA CENTER

UNIVERSITÉ CÔTE D'AZUR

N-SIDE