Using Sparsity in Superquantile-Constrained Optimization

Jake Roth

UMN ISyE

ICCOPT July 24, 2025

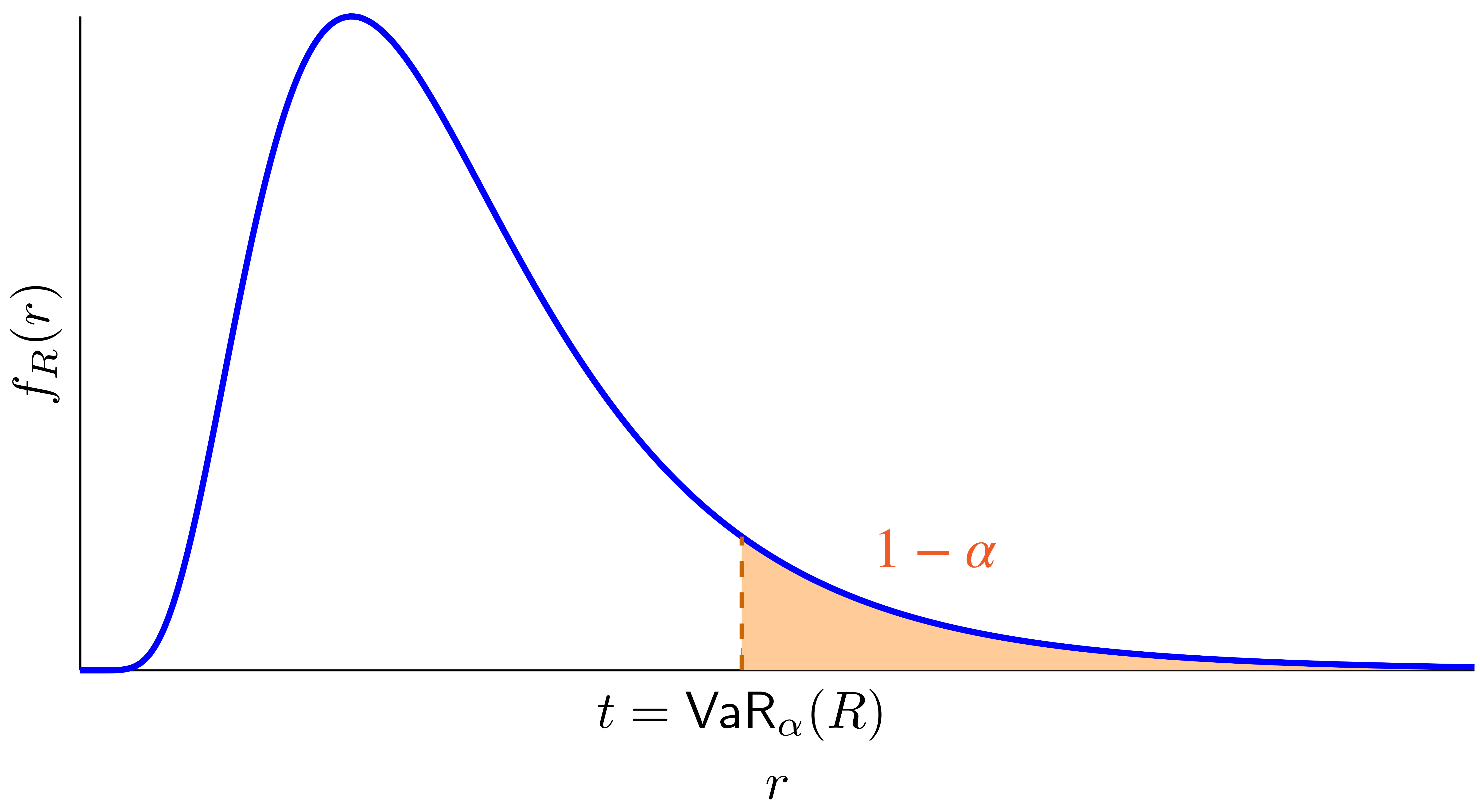
with **Ying Cui** (UCB)
Ignacio Aravena Solis and **Cosmin Petra** (LLNL)

Outline

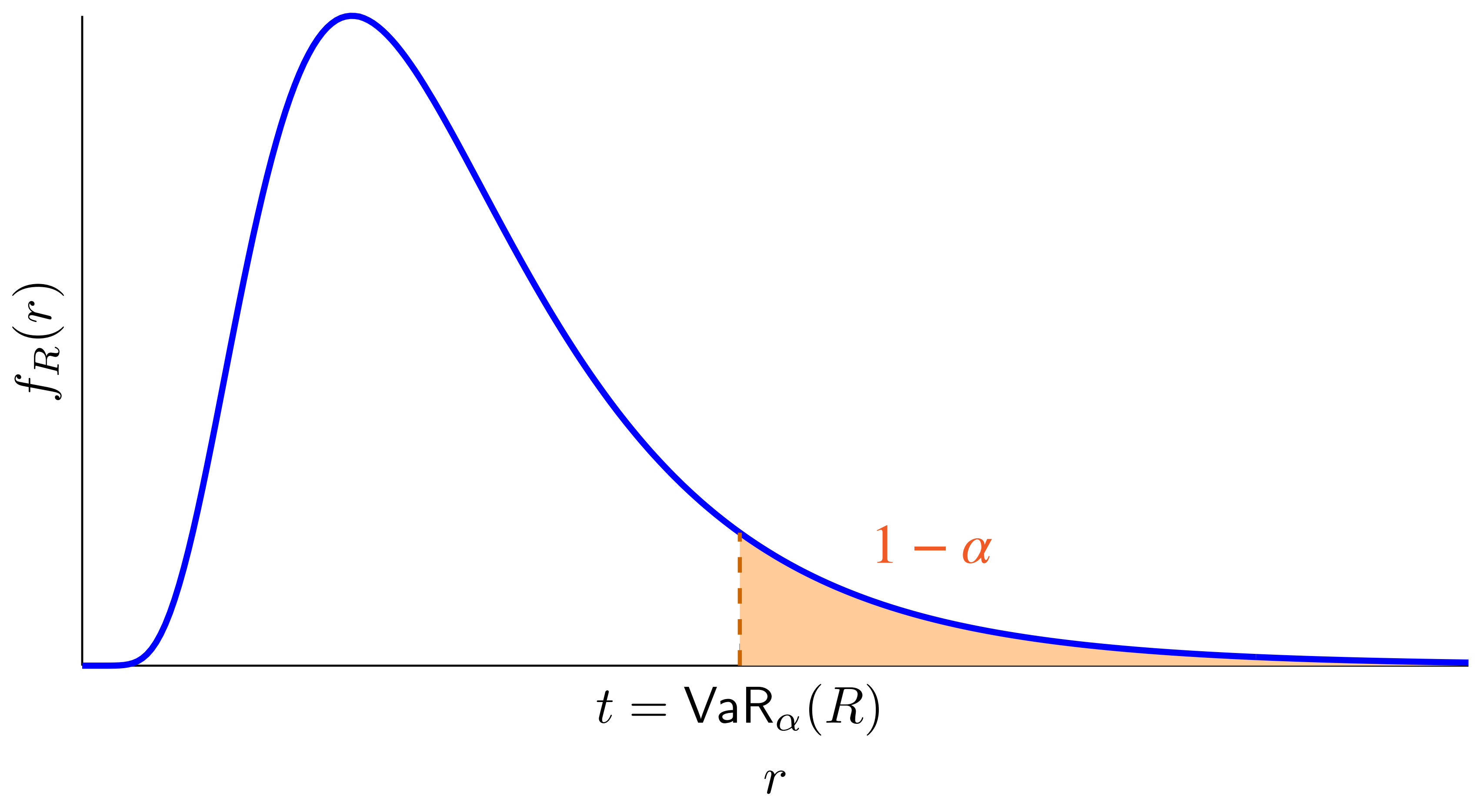
- Problem (uniform SAA)
    - Superquantile
    - Methodology
- Application (weighted SAA)
    - What changes

Superquantile

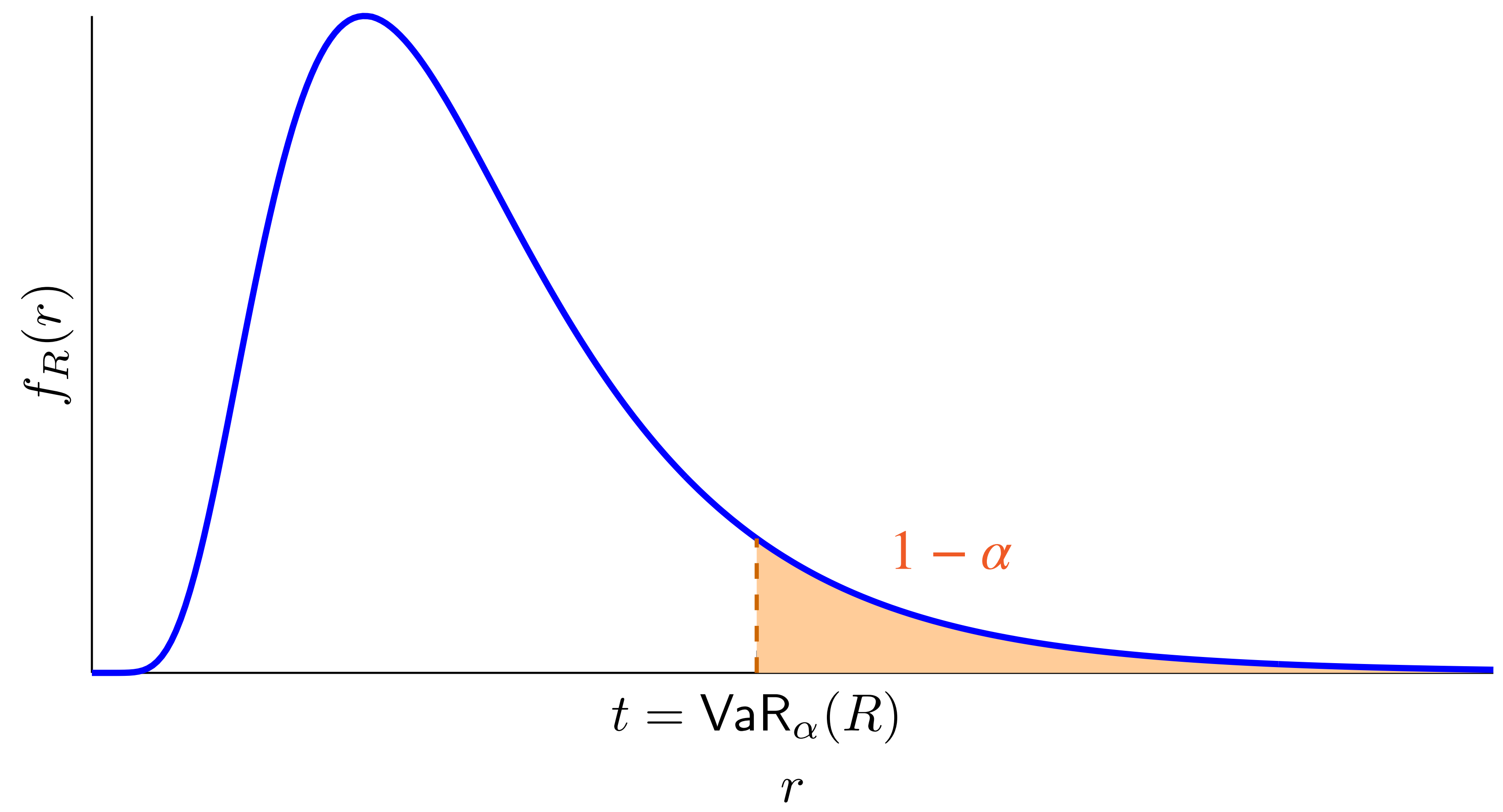
Superquantile



Superquantile

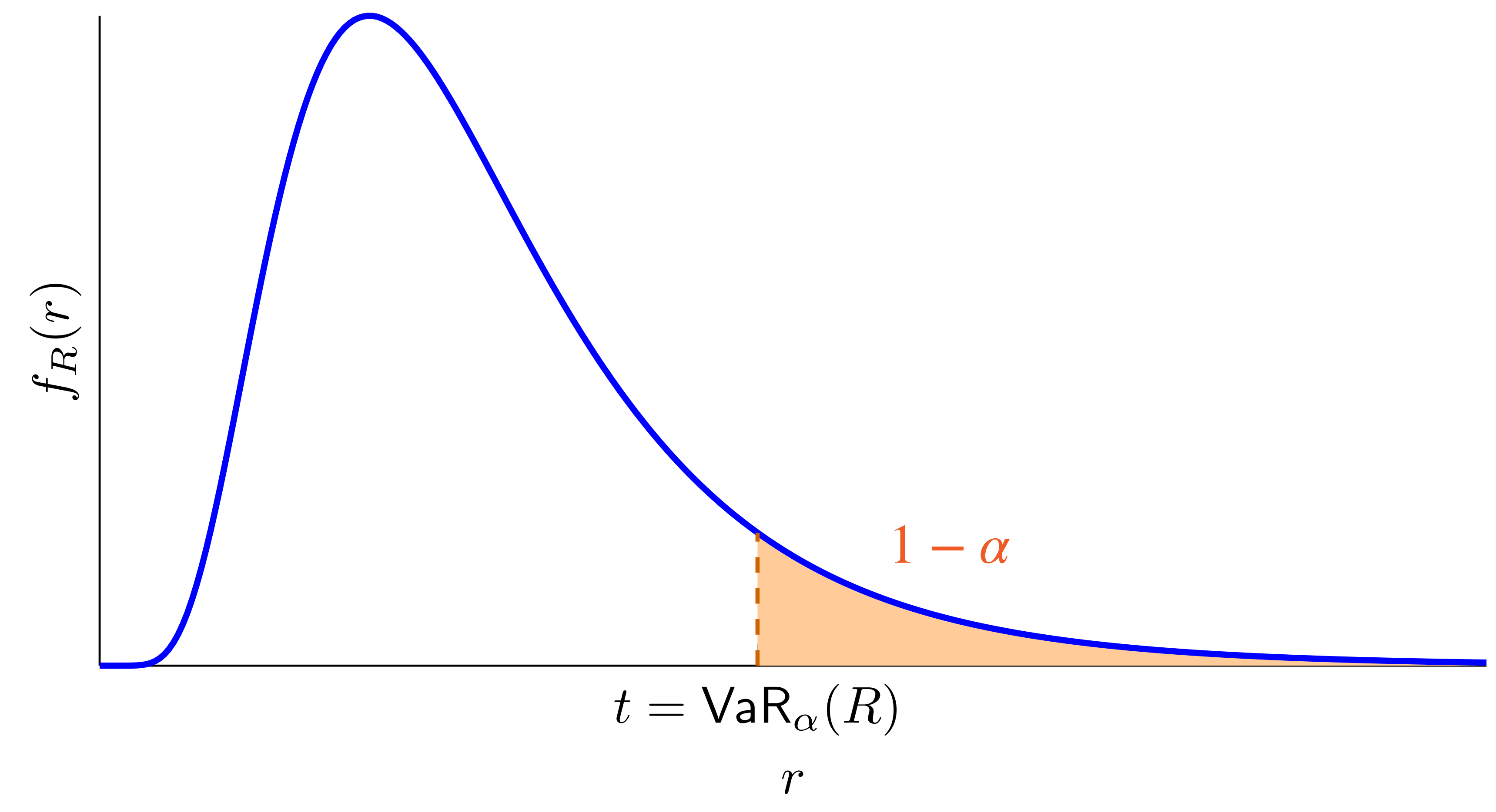


Superquantile



- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]

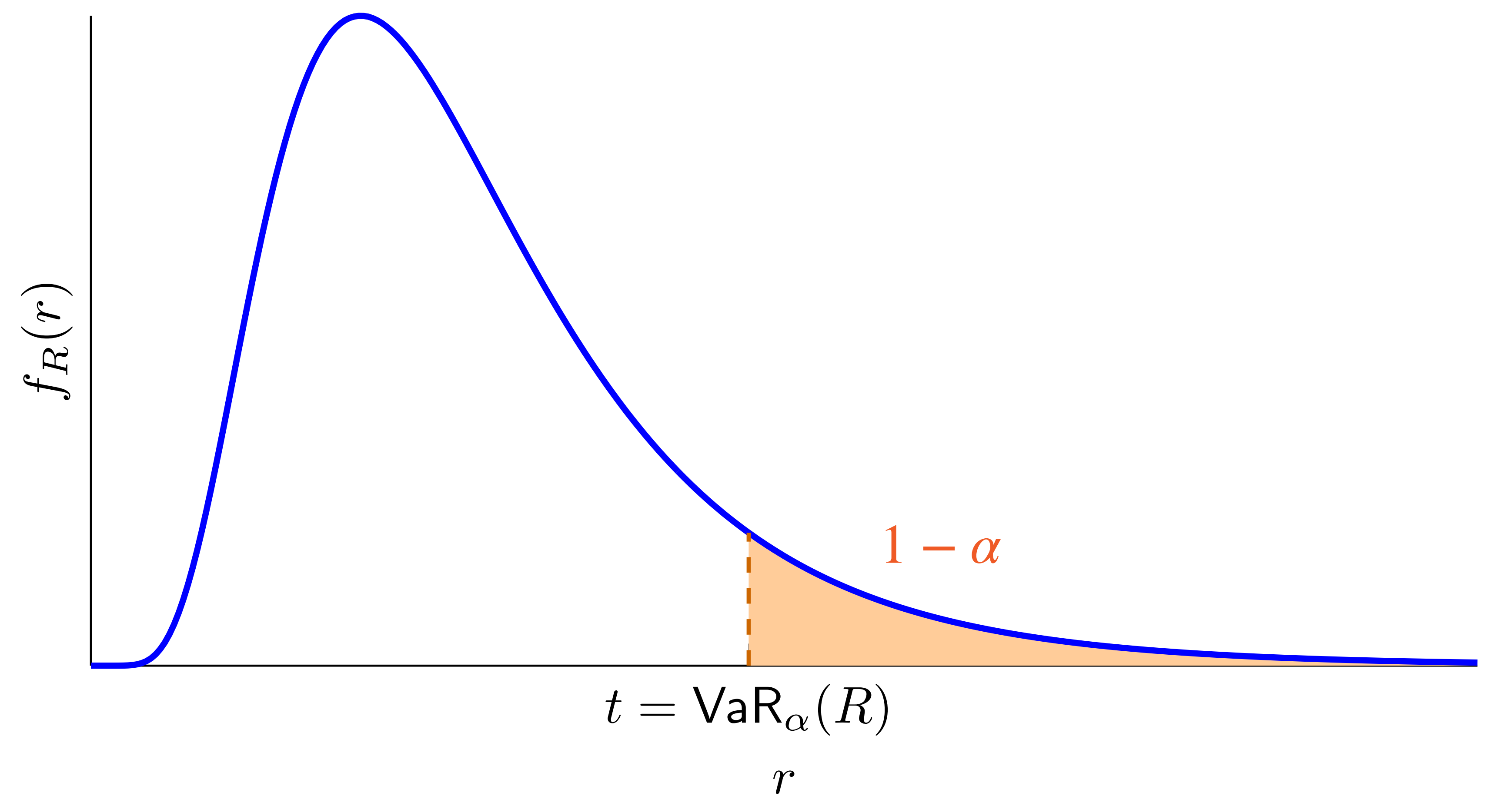
Superquantile



- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]

- Many equivalent representations…

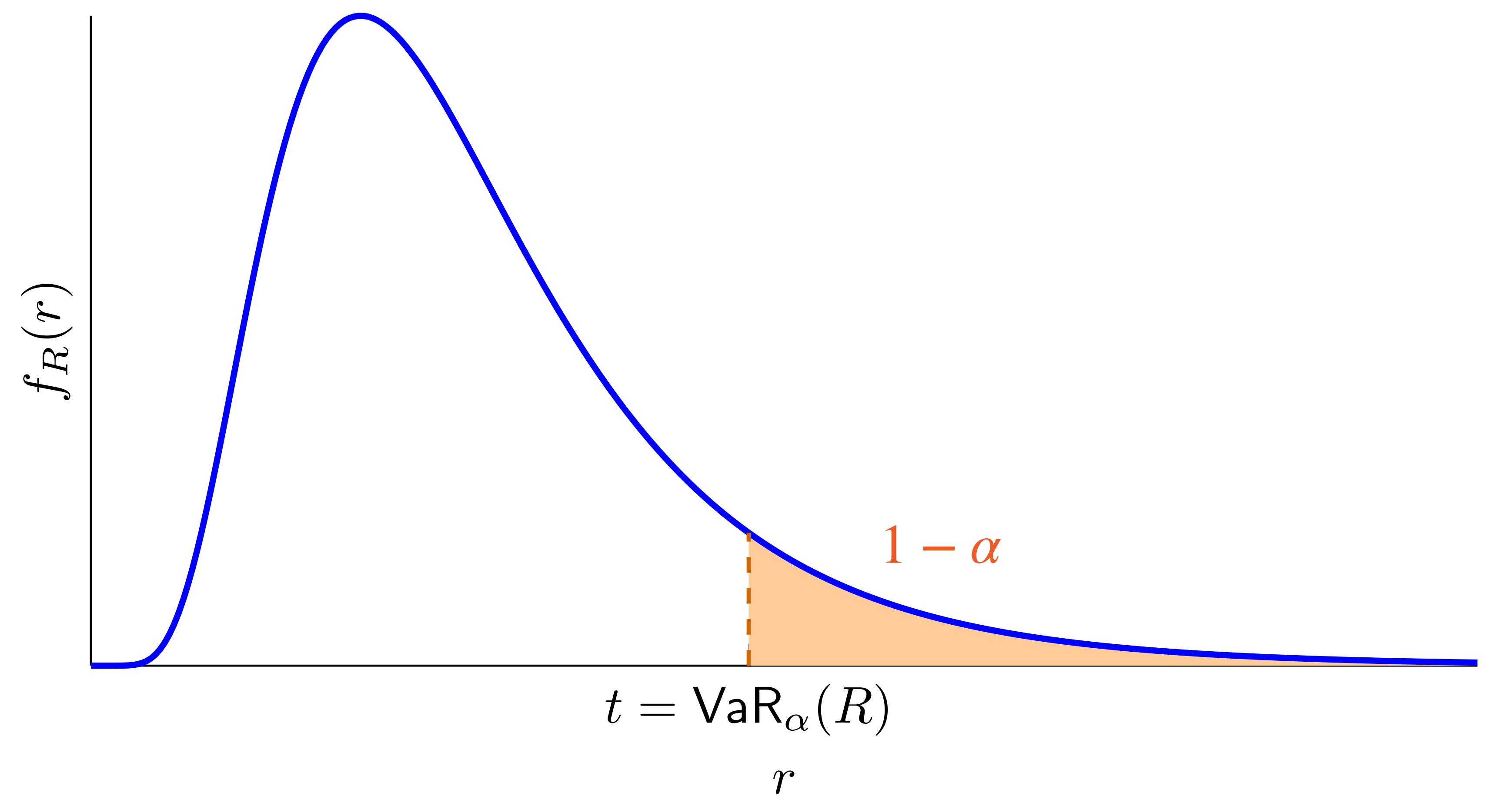
Superquantile



- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]
- Many equivalent representations…

$$\mathsf{CVaR}_\alpha(R) := \min_t \, t + \frac{1}{1-\alpha}\mathsf{E}_P[R-t]_+$$

Superquantile

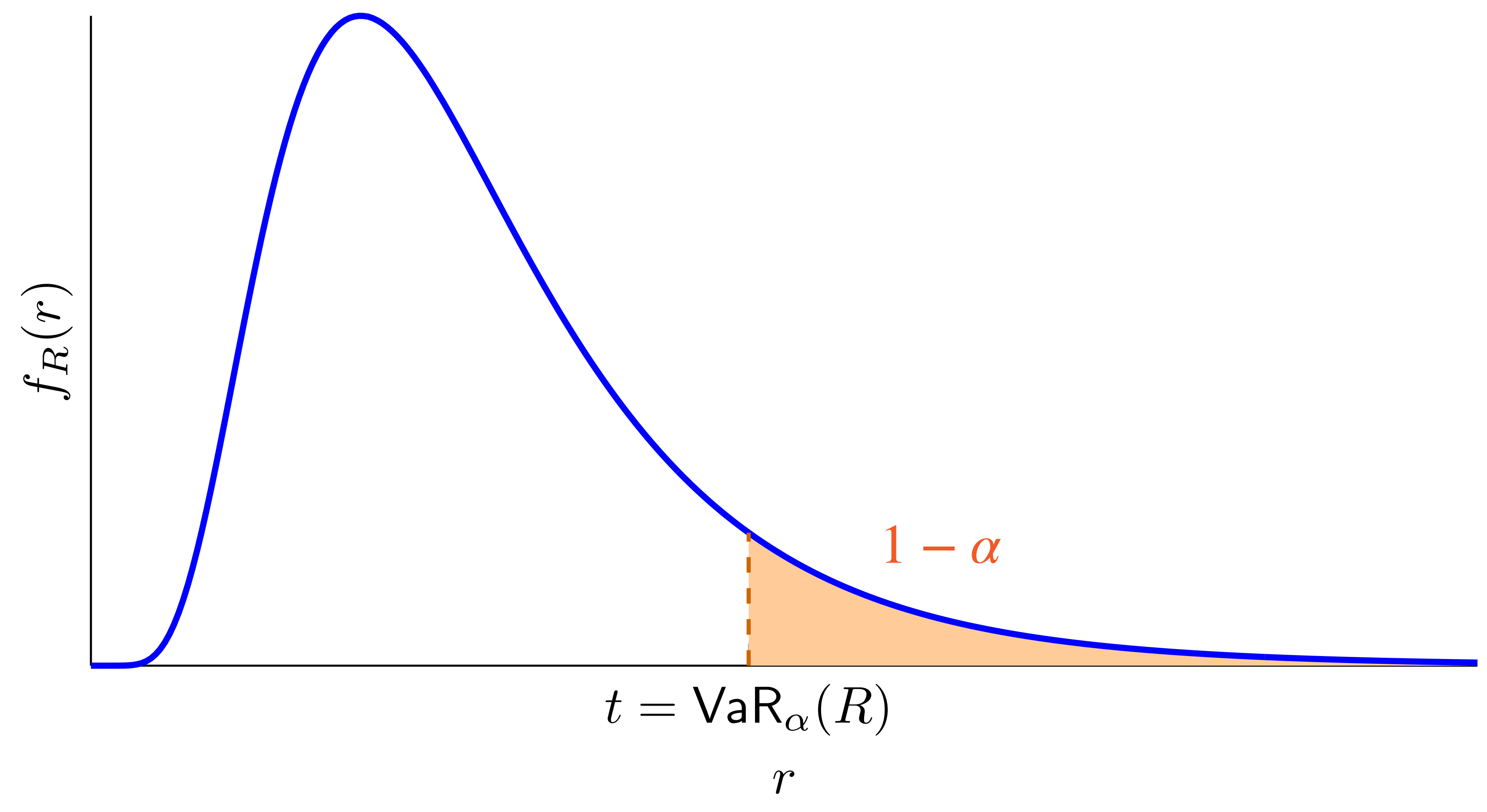


- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]

- Many equivalent representations…

$$\mathsf{CVaR}_\alpha(R) := \min_t \; t + \frac{1}{1-\alpha}\mathsf{E}_P[R-t]_+$$
$$= \min_t \; t + \frac{1}{1-\alpha}\sum_{i=1}^m \frac{1}{m}\cdot[r_i - t]_+ = \max_{q\in Q_{k(\alpha)}} \langle q, r\rangle$$

Superquantile



- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]

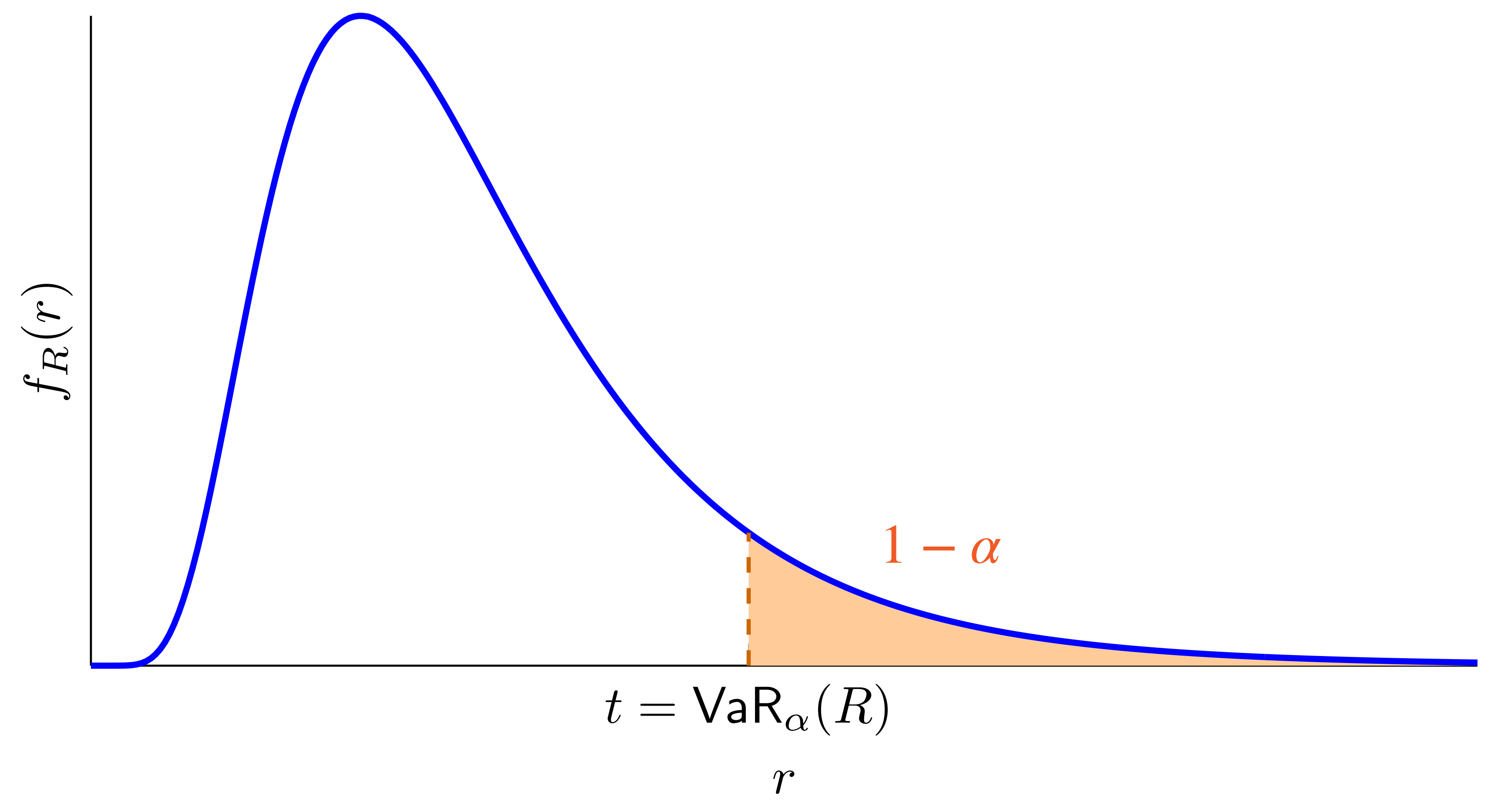
- Many equivalent representations…

$$\mathsf{CVaR}_\alpha(R) := \min_t \, t + \frac{1}{1-\alpha} \mathsf{E}_P[R-t]_+$$

$$= \min_t \, t + \frac{1}{1-\alpha} \sum_{i=1}^m \frac{1}{m} \cdot [r_i - t]_+ = \max_{q \in Q_{k(\alpha)}} \langle q, r \rangle$$

$$Q_{k(\alpha)} := \{ q \in \mathbb{R}^m : 0 \le q \le \frac{1}{k},\ \sum_{i=1}^m q_i = 1 \}$$

Superquantile



- Many names: CVaR (conditional value at risk) / AVaR / superquantile / expected shortfall [Ben-Tal & Teboulle, Rockafellar & Uryasev, Rockafellar & Royset…]

- Many equivalent representations…

$$\mathrm{CVaR}_\alpha(R) := \min_t \; t + \frac{1}{1-\alpha}\mathrm{E}_P[R-t]_+$$
$$= \min_t \; t + \frac{1}{1-\alpha}\sum_{i=1}^m \frac{1}{m}\cdot[r_i - t]_+ = \max_{q\in Q_{k(\alpha)}} \langle q, r\rangle$$
$$Q_{k(\alpha)} := \{q \in \mathbb{R}^m : 0 \le q \le \tfrac{1}{k},\ \textstyle\sum_{i=1}^m q_i = 1\}$$

- In SAA setting: $\mathrm{T}_{(k)}(R) := \sum_{i=1}^k r_{(i)}$ where(1) $k(\alpha) = m\cdot(1-\alpha)$ **i.e., α divides m**

(1) when α doesn't divide m and $k = \lfloor \alpha m \rfloor$, we need to append (and appropriately weight) the $(k+1)^{\text{th}}$ largest term.

Superquantile optimization

Superquantile optimization

- Example: $$\underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} \quad f_0(x) + \mathrm{CVaR}_\alpha(G(x, \xi))$$

  where $G(x; \xi) = \{g_i(x; \xi_i)\}_{i=1}^m$

Superquantile optimization

- Example:

$$\underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} \quad f_0(x) + \text{CVaR}_\alpha(G(x, \xi))$$

  where $G(x; \xi) = \{g_i(x; \xi_i)\}_{i=1}^m$

- General problem:

Superquantile optimization

- Example:

$$\underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} \quad f_0(x) + \mathrm{CVaR}_\alpha(G(x, \xi))$$

where $G(x; \xi) = \{g_i(x; \xi_i)\}_{i=1}^m$

- General problem:

$$\begin{aligned} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} \quad & f_0(x) + \mathrm{CVaR}_\alpha(G^0(x, \xi^0)) \\ \text{subject to} \quad & \mathrm{CVaR}_\alpha(G^\ell(x, \xi^\ell)) \leq 0, \; \ell = 1, \ldots, L \end{aligned}$$

Expectation vs superquantile

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

$$\mathsf{CVaR}_k[g(x;\xi)] \approx \frac{1}{k}\sum_{i=1}^{k} g(x;\xi_{(i)})$$

$$g(x;\xi_{(1)}) \geq g(x;\xi_{(2)}) \geq \cdots \geq g(x;\xi_{(m)})$$

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

$$\mathrm{CVaR}_k[g(x;\xi)] \approx \frac{1}{k}\sum_{i=1}^{k} g(x;\xi_{(i)})$$

$$g(x;\xi_{(1)}) \geq g(x;\xi_{(2)}) \geq \cdots \geq g(x;\xi_{(m)})$$

- **Non-separable**: only tail samples contribute

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

$$\mathrm{CVaR}_k[g(x;\xi)] \approx \frac{1}{k}\sum_{i=1}^{k} g(x;\xi_{(i)})$$

$$g(x;\xi_{(1)}) \geq g(x;\xi_{(2)}) \geq \cdots \geq g(x;\xi_{(m)})$$

- **Non-separable**: only tail samples contribute
- Only samples $g(x;\xi_{k'})$ in tail ($k' \geq k$) give nontrivial (sub)gradient

Expectation vs superquantile

$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

$$\mathrm{CVaR}_k[g(x;\xi)] \approx \frac{1}{k}\sum_{i=1}^{k} g(x;\xi_{(i)})$$

$$g(x;\xi_{(1)}) \geq g(x;\xi_{(2)}) \geq \cdots \geq g(x;\xi_{(m)})$$

- **Non-separable**: only tail samples contribute
- Only samples $g(x;\xi_{k'})$ in tail ($k' \geq k$) give nontrivial (sub)gradient
- Function evaluation may be costly!

Expectation vs superquantile

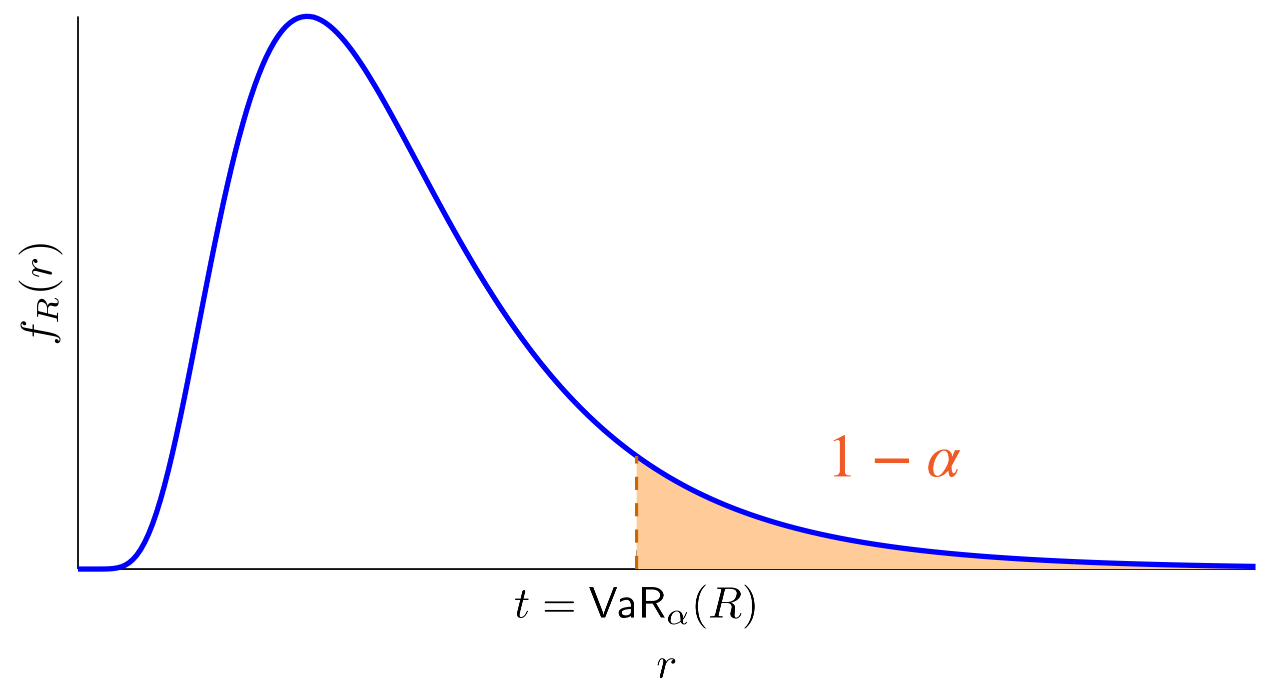
$$\mathbb{E}[g(x;\xi)] \approx \frac{1}{m}\sum_{i=1}^{m} g(x;\xi_i)$$

- **Separable**: all samples contribute equally
- Any sample gives estimate of function's value and (sub)gradient

$$\mathrm{CVaR}_k[g(x;\xi)] \approx \frac{1}{k}\sum_{i=1}^{k} g(x;\xi_{(i)})$$

$$g(x;\xi_{(1)}) \geq g(x;\xi_{(2)}) \geq \cdots \geq g(x;\xi_{(m)})$$

- **Non-separable**: only tail samples contribute
- Only samples $g(x;\xi_{k'})$ in tail ($k' \geq k$) give nontrivial (sub)gradient
- Function evaluation may be costly!



Superquantile optimization

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Expensive to evaluate “Hessian” and solve linear system (with many scenarios)

Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Expensive to evaluate "Hessian" and solve linear system (with many scenarios)

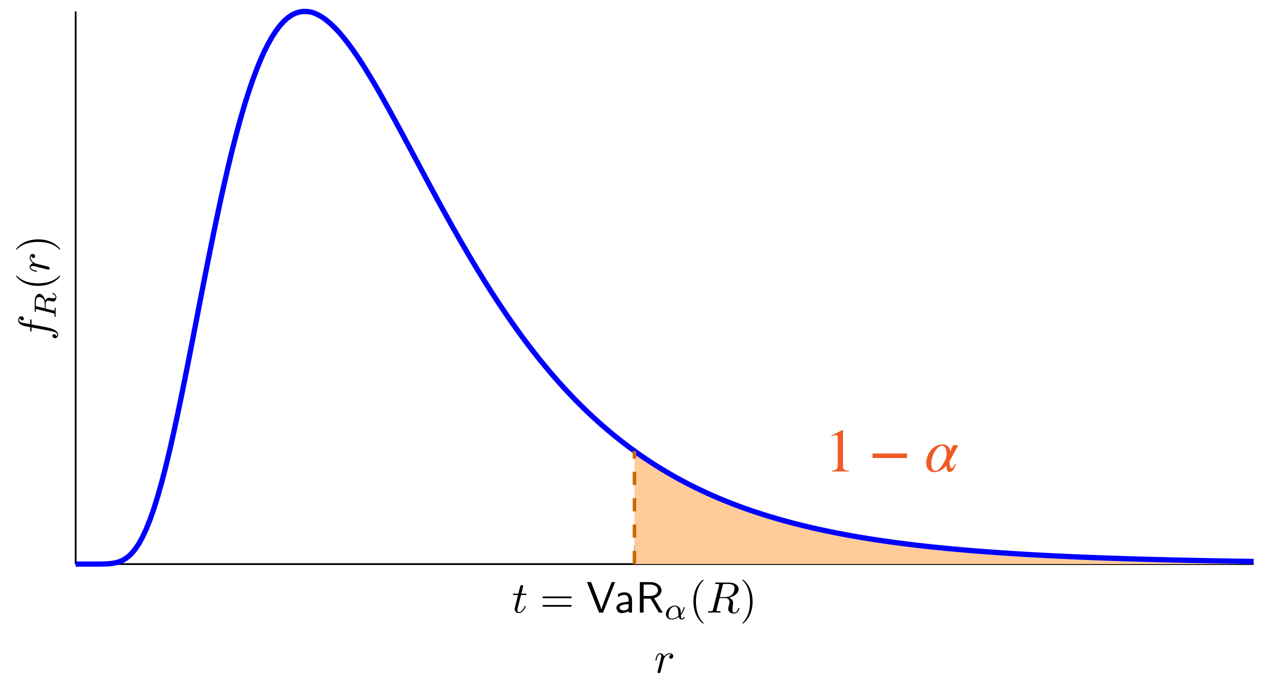
Superquantile optimization

Reduce # of function / (sub)gradient evaluations && obtain high precision

Second order method

Expensive to evaluate "Hessian" and solve linear system (with many scenarios)

only tail scenarios matter!



"Hessian" is sparse

Existing methods

Existing methods

$$\boxed{\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \mathrm{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \mathrm{CVaR}_k(G^\ell(x, \xi^\ell)) \le 0,\ \ell = 1, \ldots, L \end{array}}$$

Existing methods

$$\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \text{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \text{CVaR}_k(G^\ell(x, \xi^\ell)) \leq 0,\ \ell = 1, \ldots, L \end{array}$$

- Linearize the “max”:

$t + \frac{1}{1-\alpha} \sum_{i=1}^{m} \frac{1}{m} z_i$ and $z_i \geq g_i(x, \xi_i) - t,\ z_i \geq 0,\ i = 1, \ldots, m$

Existing methods

$$\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \text{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \text{CVaR}_k(G^\ell(x, \xi^\ell)) \le 0,\ \ell = 1, \dots, L \end{array}$$

- Linearize the "max":

$t + \frac{1}{1-\alpha} \sum_{i=1}^m \frac{1}{m} z_i$ and $z_i \geq g_i(x, \xi_i) - t,\ z_i \geq 0,\ i = 1, \dots, m$

- Off-the-shelf second-order methods

Existing methods

$$\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \mathrm{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \mathrm{CVaR}_k(G^\ell(x, \xi^\ell)) \le 0, \ \ell = 1, \dots, L \end{array}$$

- Linearize the “max”:

$t + \frac{1}{1-\alpha} \sum_{i=1}^m \frac{1}{m} z_i$ and $z_i \ge g_i(x, \xi_i) - t$, $z_i \ge 0$, $i = 1, \dots, m$

- Off-the-shelf second-order methods
  - Numerical scaling in Newton system when constraints/scenarios switch

Existing methods

$$\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \text{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \text{CVaR}_k(G^\ell(x, \xi^\ell)) \leq 0,\ \ell = 1, \ldots, L \end{array}$$

- Linearize the “max”:

$t + \frac{1}{1-\alpha} \sum_{i=1}^{m} \frac{1}{m} z_i$ and $z_i \geq g_i(x, \xi_i) - t,\ z_i \geq 0,\ i = 1, \ldots, m$

- Off-the-shelf second-order methods
    - Numerical scaling in Newton system when constraints/scenarios switch
    - How to avoid computation of derivative information of inactive g_i

Existing methods

$$\begin{array}{ll} \underset{x \in X \subseteq \mathbb{R}^n}{\text{minimize}} & f_0(x) + \mathrm{CVaR}_k(G^0(x, \xi^0)) \\ \text{subject to} & \mathrm{CVaR}_k(G^\ell(x, \xi^\ell)) \leq 0,\ \ell = 1, \ldots, L \end{array}$$

- Linearize the "max":

$t + \frac{1}{1-\alpha} \sum_{i=1}^{m} \frac{1}{m} z_i$ and $z_i \geq g_i(x, \xi_i) - t,\ z_i \geq 0,\ i = 1, \ldots, m$

- Off-the-shelf second-order methods
  - Numerical scaling in Newton system when constraints/scenarios switch
  - How to avoid computation of derivative information of inactive g_i
- Active set: need good heuristics to tune

Augmented Lagrangian Method

Augmented Lagrangian Method

- Simplified problem:

Augmented Lagrangian Method

- Simplified problem:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^\top x \\ \text{subject to} & \mathrm{CVaR}_k(Ax + b) \leq 0 \end{array}$$

$$\boxed{\begin{array}{c} \text{“row” := “scenario”} \\ g_i(x;\xi) := a_i^\top x + b_i \end{array}}$$

Augmented Lagrangian Method

- Simplified problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x \\ \text{subject to} \quad & \mathrm{CVaR}_k(Ax + b) \le 0 \end{aligned}$$

$$\boxed{\begin{array}{c} \text{“row” := “scenario”} \\ g_i(x;\xi) := a_i^\top x + b_i \end{array}}$$

- Reformulate:

Augmented Lagrangian Method

- Simplified problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^\top x$$
$$\text{subject to} \quad \mathrm{CVaR}_k(Ax + b) \le 0$$

$\boxed{\begin{array}{c}\text{“row” := “scenario”} \\ g_i(x;\xi) := a_i^\top x + b_i\end{array}}$

- Reformulate:

$$\underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} \quad c^\top x$$
$$\text{subject to} \quad \mathrm{CVaR}_k(y) \le 0, \;\; y = Ax + b$$

Augmented Lagrangian Method

- Simplified problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^\top x$$
$$\text{subject to} \quad \text{CVaR}_k(Ax + b) \le 0$$

"row" := "scenario"
$g_i(x; \xi) := a_i^\top x + b_i$

- Reformulate:

$$\underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} \quad c^\top x$$
$$\text{subject to} \quad \text{CVaR}_k(y) \le 0, \quad y = Ax + b$$

- Partial AugLag function: $c^\top x + \lambda^\top (y - Ax - b) + \frac{\sigma}{2} \|y - Ax - b\|^2$ with $\text{CVaR}_k(y) \le 0$

Augmented Lagrangian Method

- Simplified problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x \\ \text{subject to} \quad & \mathrm{CVaR}_k(Ax + b) \le 0 \end{aligned}$$

$$\boxed{\begin{array}{c} \text{“row” := “scenario”} \\ g_i(x;\xi) := a_i^\top x + b_i \end{array}}$$

- Reformulate:

$$\begin{aligned} \underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} \quad & c^\top x \\ \text{subject to} \quad & \mathrm{CVaR}_k(y) \le 0, \;\; y = Ax + b \end{aligned}$$

- Partial AugLag function: $c^\top x + \lambda^\top (y - Ax - b) + \frac{\sigma}{2}\|y - Ax - b\|^2$ with $\mathrm{CVaR}_k(y) \le 0$

ALM Subproblem

Augmented Lagrangian Method

- Simplified problem:

$$\begin{aligned}\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x \\ \text{subject to} \quad & \mathrm{CVaR}_k(Ax + b) \le 0\end{aligned}$$

“row” := “scenario”
$g_i(x; \xi) := a_i^\top x + b_i$

- Reformulate:

$$\begin{aligned}\underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} \quad & c^\top x \\ \text{subject to} \quad & \mathrm{CVaR}_k(y) \le 0,\;\; y = Ax + b\end{aligned}$$

- Partial AugLag function: $c^\top x + \lambda^\top(y - Ax - b) + \frac{\sigma}{2}\|y - Ax - b\|^2$ with $\mathrm{CVaR}_k(y) \le 0$

ALM Subproblem

$$\underset{x,\, y}{\text{minimize}} \quad c^\top x + \langle \tilde{\lambda}, y - Ax - b \rangle + \frac{\sigma}{2}\|y - Ax - b\|^2$$

Augmented Lagrangian Method

- Simplified problem:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & c^\top x \\ \text{subject to} & \text{CVaR}_k(Ax+b) \le 0 \end{array}$$

"row" := "scenario"
$g_i(x;\xi) := a_i^\top x + b_i$

- Reformulate:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} & c^\top x \\ \text{subject to} & \text{CVaR}_k(y) \le 0, \;\; y = Ax+b \end{array}$$

- Partial AugLag function: $c^\top x + \lambda^\top(y - Ax - b) + \frac{\sigma}{2}\|y - Ax - b\|^2$ with $\text{CVaR}_k(y) \le 0$

ALM Subproblem

$$\underset{x,y}{\text{minimize}} \quad c^\top x + \langle \tilde{\lambda}, y - Ax - b \rangle + \frac{\sigma}{2}\|y - Ax - b\|^2$$

eliminate y

$$\underset{x}{\text{minimize}} \quad c^\top x + \frac{\sigma}{2}\left\| \Pi_{\text{CVaR}_k(\cdot)\le 0}(Ax + b - \tilde{\lambda}/\sigma) - \left(Ax + b - \tilde{\lambda}/\sigma\right) \right\|^2$$

Augmented Lagrangian Method

- Simplified problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^\top x \qquad \text{subject to} \quad \mathrm{CVaR}_k(Ax+b) \le 0$$

“row” := “scenario”
$g_i(x;\xi) := a_i^\top x + b_i$

- Reformulate:

$$\underset{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m}{\text{minimize}} \quad c^\top x \qquad \text{subject to} \quad \mathrm{CVaR}_k(y) \le 0, \;\; y = Ax + b$$

- Partial AugLag function: $c^\top x + \lambda^\top (y - Ax - b) + \frac{\sigma}{2}\|y - Ax - b\|^2$ with $\mathrm{CVaR}_k(y) \le 0$

ALM Subproblem

$$\underset{x,\,y}{\text{minimize}} \quad c^\top x + \langle \tilde{\lambda}, y - Ax - b \rangle + \frac{\sigma}{2}\|y - Ax - b\|^2$$

eliminate y

$$\underset{x}{\text{minimize}} \quad c^\top x + \frac{\sigma}{2}\left\| \boxed{\Pi_{\mathrm{CVaR}_k(\cdot)\le 0}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \tilde{\lambda}/\sigma\right) \right\|^2$$

ALM Subproblem

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

ALM Subproblem

$$\boxed{\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}}$$

- Tail risk set: $B_k := \{r \in \mathbb{R}^m : \mathrm{CVaR}_k(r) \leq 0\}$

ALM Subproblem

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Tail risk set: $B_k := \{r \in \mathbb{R}^m : \mathrm{CVaR}_k(r) \leq 0\}$
  - B_k is polyhedral (very simple non-symmetric cone)

ALM Subproblem

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Tail risk set: $B_k := \{r \in \mathbb{R}^m : \mathrm{CVaR}_k(r) \leq 0\}$
  - B_k is polyhedral (very simple non-symmetric cone)
  - $\mathrm{dist}(r, B_k)^2$ is continuously differentiable

ALM Subproblem

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Tail risk set: $B_k := \{r \in \mathbb{R}^m : \text{CVaR}_k(r) \leq 0\}$
  - B_k is polyhedral (very simple non-symmetric cone)
  - $\text{dist}(r, B_k)^2$ is continuously differentiable
  - projection is strongly semismooth

ALM Subproblem

$$\boxed{\underset{x}{\text{minimize}} \quad \left\{\varphi(x) := c^\top x + \frac{\sigma}{2}\left\|\boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \tilde{\lambda}/\sigma\right)\right\|^2\right\}}$$

- Tail risk set: $B_k := \{r \in \mathbb{R}^m : \mathrm{CVaR}_k(r) \leq 0\}$
    - B_k is polyhedral (very simple non-symmetric cone)
    - $\mathrm{dist}(r, B_k)^2$ is continuously differentiable
    - projection is strongly semismooth
- Subproblem optimality condition:

$$\nabla\varphi(x) = c + \sigma \cdot A^\top\left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)\right] = 0$$

can be solved by **semismooth Newton method**

Tail risk (B_k) projection: detail

Tail risk (B_k) projection: detail

- Given y^0, calculate

$$\Pi_{B_\alpha}(y^0) = \operatorname{arg\,min}_y \frac{1}{2}\|y - y^0\|_2^2 : y_{(1)} + y_{(2)} + \cdots + y_{(\lfloor \alpha m \rfloor)} \leq 0$$

Tail risk (B_k) projection: detail

- Given y^0, calculate

$$\Pi_{B_\alpha}(y^0) = \operatorname{arg\,min}_y \frac{1}{2}\|y - y^0\|_2^2 : y_{(1)} + y_{(2)} + \cdots + y_{(\lfloor \alpha m \rfloor)} \leq 0$$

- Sort!

$$\Pi_{B_\alpha}(y^{0,\downarrow}) = \operatorname{arg\,min}_y \frac{1}{2}\|y - y^{0,\downarrow}\|_2^2 : y_1 \geq y_2 \geq \cdots \geq y_m,\ \sum_{i=1}^k y_i \leq 0$$

Tail risk (B_k) projection: detail

- Given y^0, calculate

$$\Pi_{B_\alpha}(y^0) = \operatorname{arg\,min}_y \frac{1}{2}\|y - y^0\|_2^2 : y_{(1)} + y_{(2)} + \cdots + y_{(\lfloor \alpha m \rfloor)} \le 0$$

- Sort!

$$\Pi_{B_\alpha}(y^{0,\downarrow}) = \operatorname{arg\,min}_y \frac{1}{2}\|y - y^{0,\downarrow}\|_2^2 : \underbrace{y_1 \ge y_2 \ge \cdots \ge y_m, \ \textstyle\sum_{i=1}^k y_i \le 0}_{\boxed{\text{isotonic + one budget constraint}}}$$

Tail risk (B_k) projection: detail

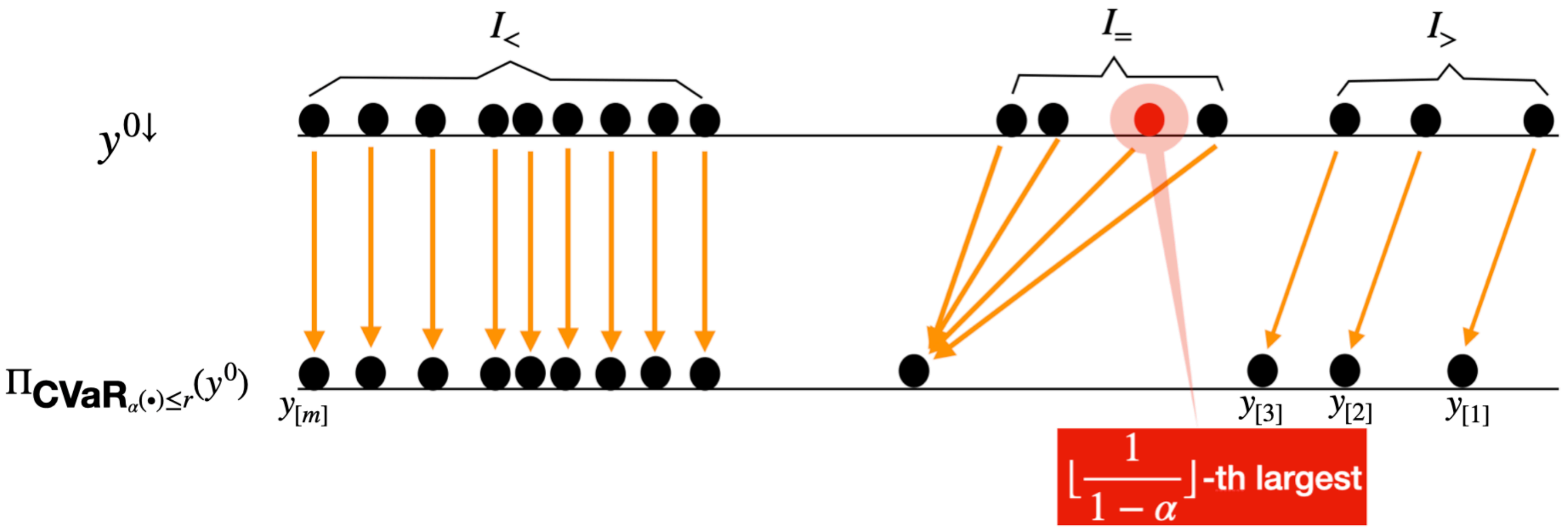
- Given y^0, calculate

$$\Pi_{B_\alpha}(y^0) = \arg\min_y \frac{1}{2}\|y - y^0\|_2^2 : y_{(1)} + y_{(2)} + \cdots + y_{(\lfloor \alpha m \rfloor)} \le 0$$

- Sort!

isotonic + one budget constraint

$$\Pi_{B_\alpha}(y^{0,\downarrow}) = \arg\min_y \frac{1}{2}\|y - y^{0,\downarrow}\|_2^2 : y_1 \ge y_2 \ge \cdots \ge y_m,\ \sum_{i=1}^{k} y_i \le 0$$



equivalent to parametric LCP with tridiagonal Z-matrix $O(m)$

Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

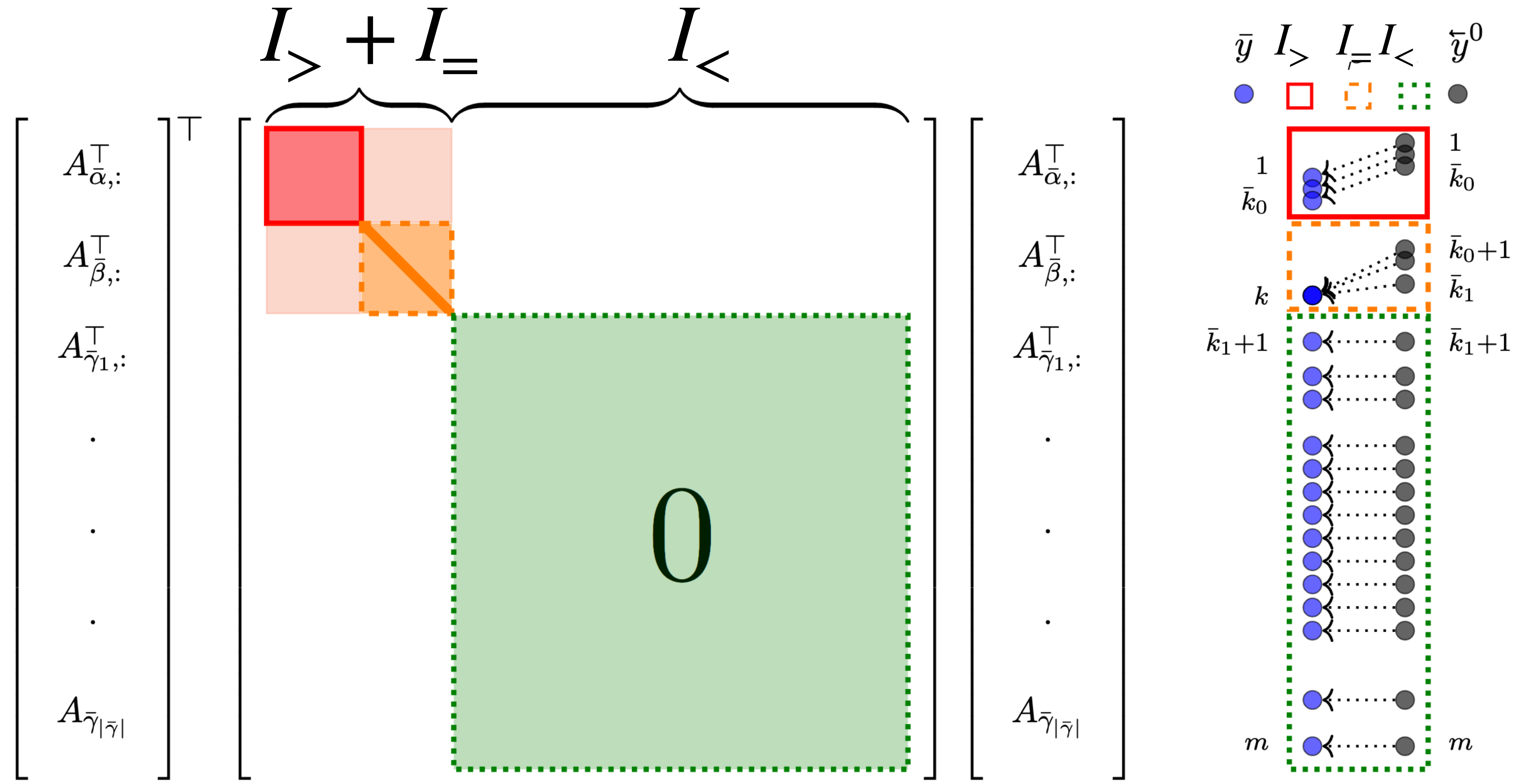
$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$



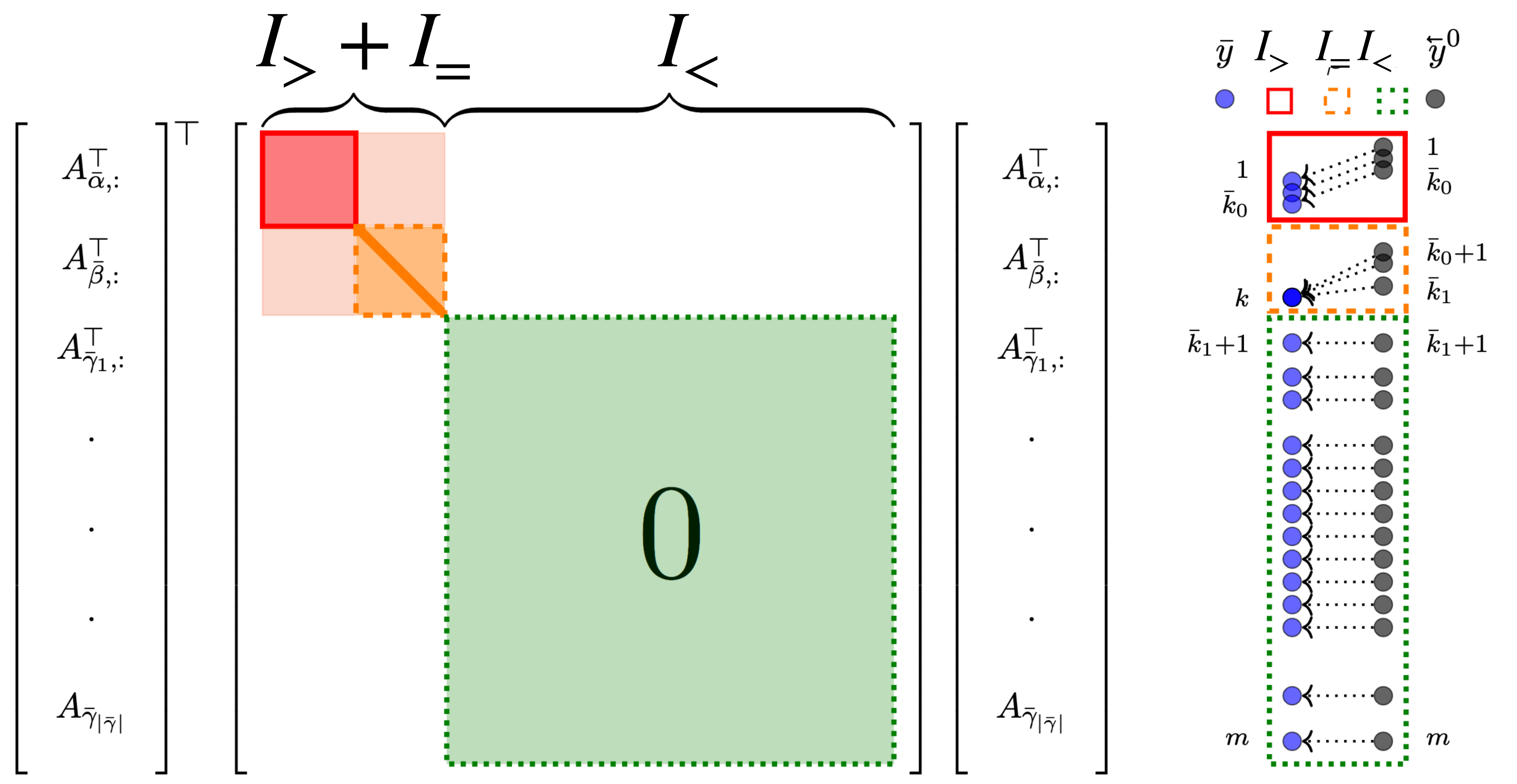
Generalized Jacobian: detail

$$\operatorname*{minimize}_{x} \quad \left\{\varphi(x) := c^\top x + \frac{\sigma}{2}\left\| \boxed{\Pi_{B_k}(Ax+b-\tilde{\lambda}/\sigma)} - \left(Ax+b-\bar{\lambda}/\sigma\right)\right\|^2\right\}$$

- Subproblem objective

$$\nabla\varphi(x) = c + \sigma\cdot A^\top\left[\left(Ax+b-\tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax+b-\tilde{\lambda}/\sigma)\right]$$
$$\hat{\nabla}^2\varphi(x) \ni \sigma\cdot A^\top(I-J)A$$

- Selects $I_> \cup I_=$ samples and reweights



Generalized Jacobian: detail

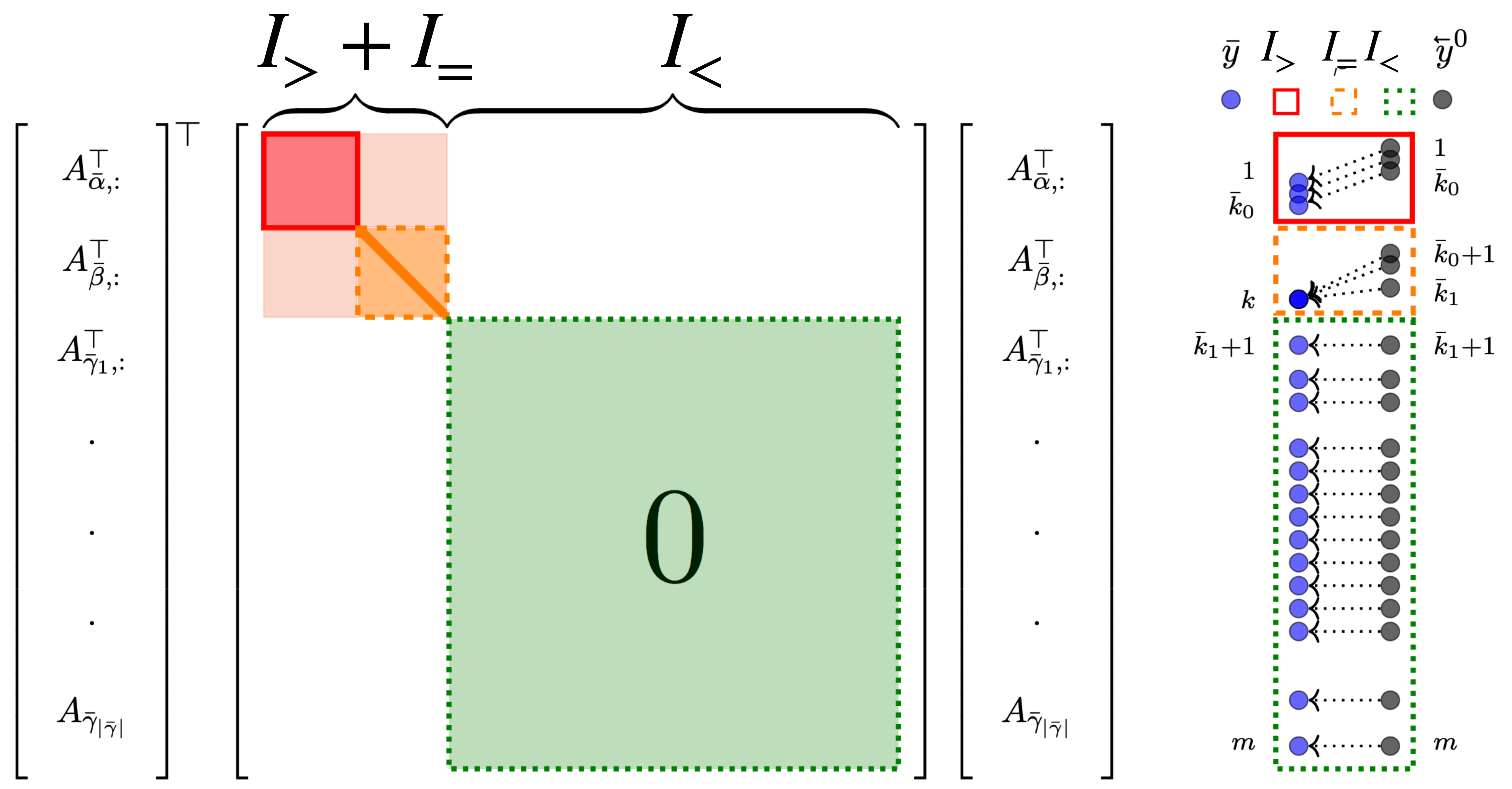
$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

- Selects $I_> \cup I_=$ samples and reweights

- **Often have $|I_> \cup I_=| \approx k \leq m$**



Generalized Jacobian: detail

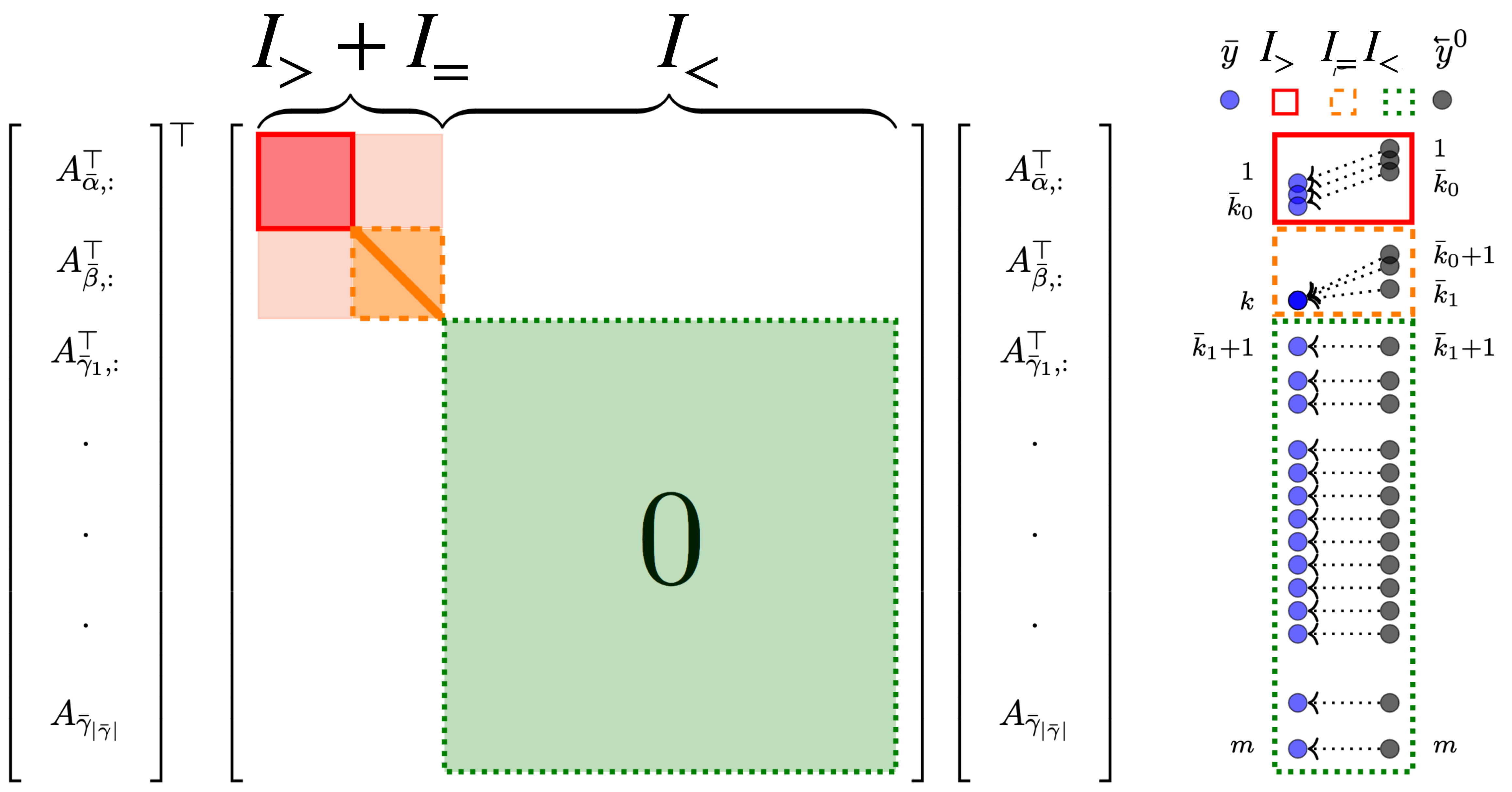
$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - (Ax + b - \bar{\lambda}/\sigma) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma \right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

- Selects $I_> \cup I_=$ samples and reweights
- **Often have $|I_> \cup I_=| \approx k \leq m$**
- **Compute $A^\top (I - J) A = \tilde{A}^\top \tilde{A}$ with**

  $\tilde{A}^\top \tilde{A} \in \mathbb{R}^{(I_= + 1) \times n}$ with $|I_=| \ll m$



Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

- Selects $I_> \cup I_=$ samples and reweights
- **Often have** $|I_> \cup I_=| \approx k \leq m$
- **Compute** $A^\top (I - J) A = \tilde{A}^\top \tilde{A}$ **with**

  $\tilde{A}^\top \tilde{A} \in \mathbb{R}^{(I_= + 1) \times n}$ **with** $|I_=| \ll m$

Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

- Selects $I_> \cup I_=$ samples and reweights
- **Often have** $|I_> \cup I_=| \approx k \leq m$
- **Compute** $A^\top (I - J) A = \tilde{A}^\top \tilde{A}$ **with**

  $\tilde{A}^\top \tilde{A} \in \mathbb{R}^{(I_= + 1) \times n}$ **with** $|I_=| \ll m$

- Gradient evaluation: $O(m \log m + mn)$

Generalized Jacobian: detail

$$\underset{x}{\text{minimize}} \quad \left\{ \varphi(x) := c^\top x + \frac{\sigma}{2} \left\| \boxed{\Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma)} - \left(Ax + b - \bar{\lambda}/\sigma\right) \right\|^2 \right\}$$

- Subproblem objective

$$\nabla \varphi(x) = c + \sigma \cdot A^\top \left[\left(Ax + b - \tilde{\lambda}/\sigma\right) - \Pi_{B_k}(Ax + b - \tilde{\lambda}/\sigma) \right]$$
$$\hat{\nabla}^2 \varphi(x) \ni \sigma \cdot A^\top (I - J) A$$

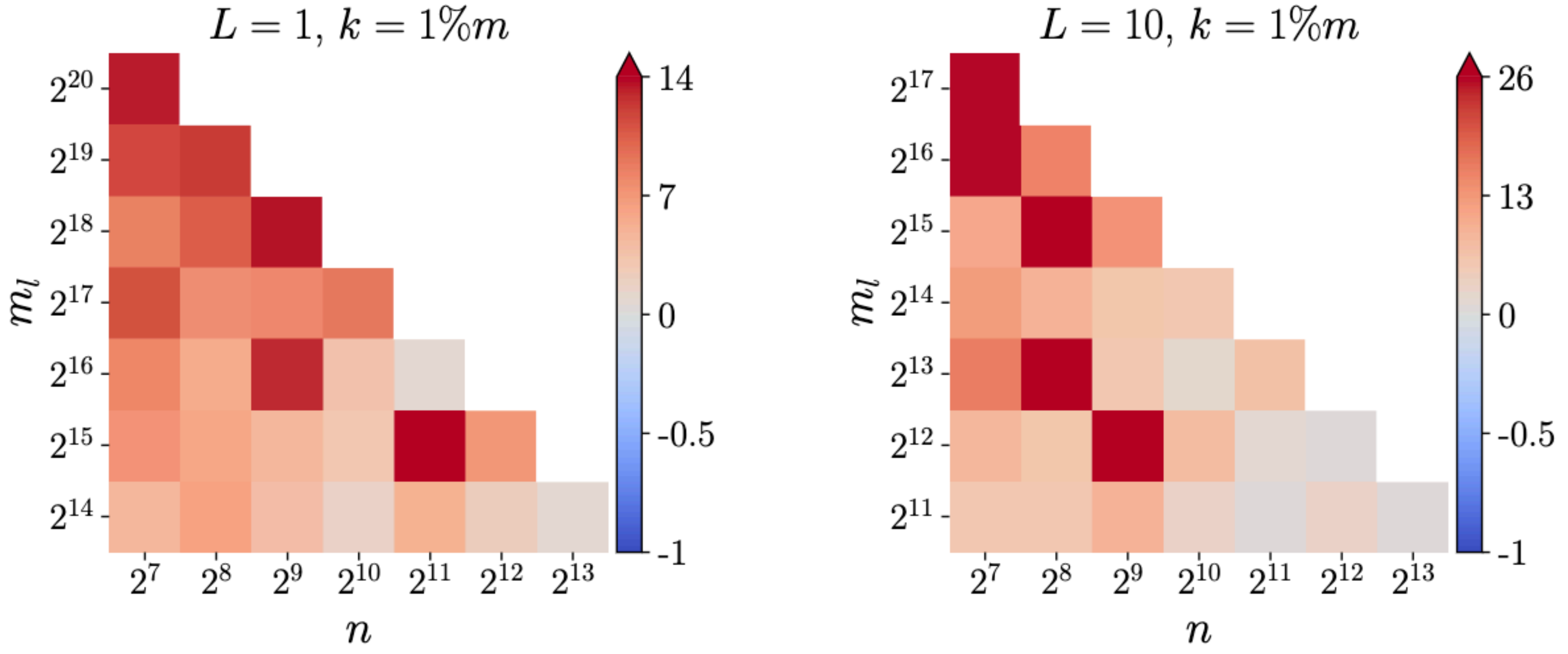
- Selects $I_> \cup I_=$ samples and reweights
- **Often have** $|I_> \cup I_=| \approx k \leq m$
- **Compute** $A^\top (I - J) A = \tilde{A}^\top \tilde{A}$ **with**

  $\tilde{A}^\top \tilde{A} \in \mathbb{R}^{(I_= + 1) \times n}$ **with** $|I_=| \ll m$

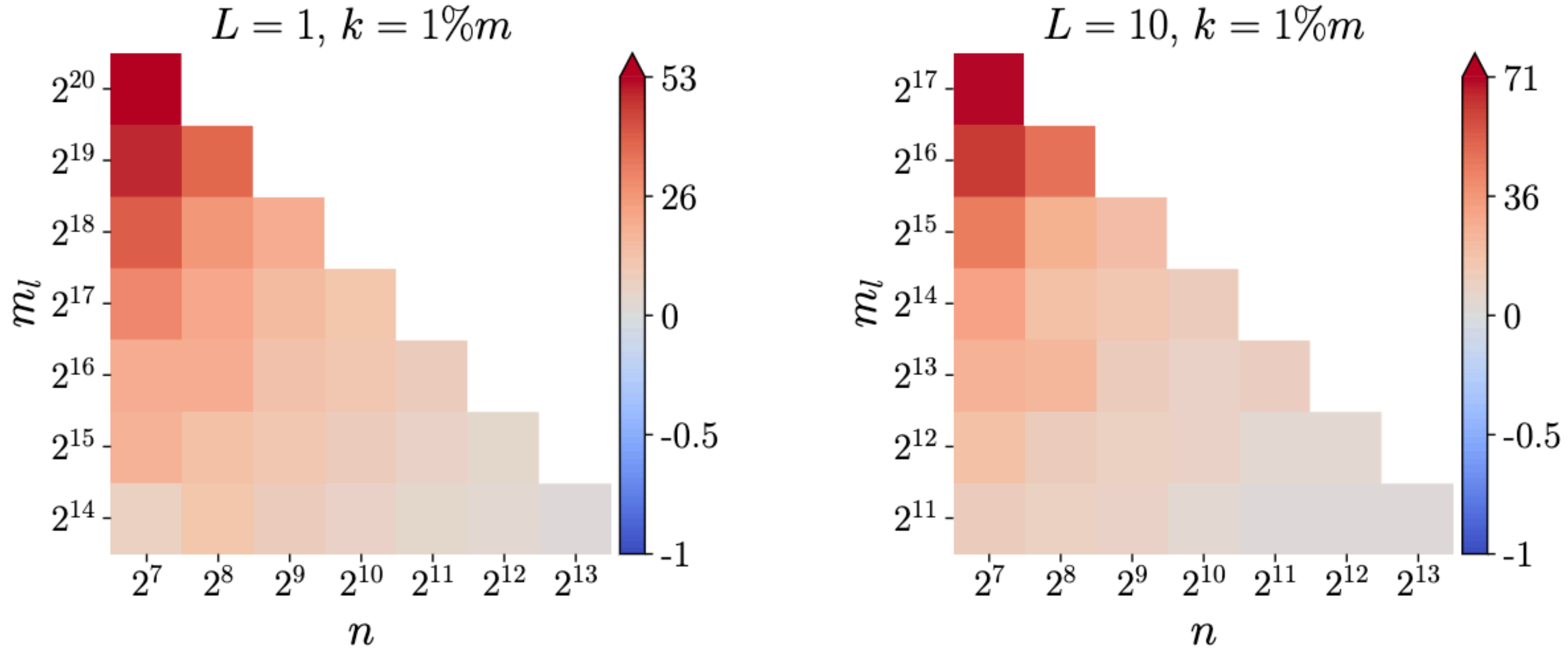
- Gradient evaluation: $O(m \log m + mn)$
- **Newton build+solve:** $O\left(n \cdot |I_= + 1|^2 + |I_= + 1|^3\right)$ **"for free"**

Numerical experiments: synthetic instances

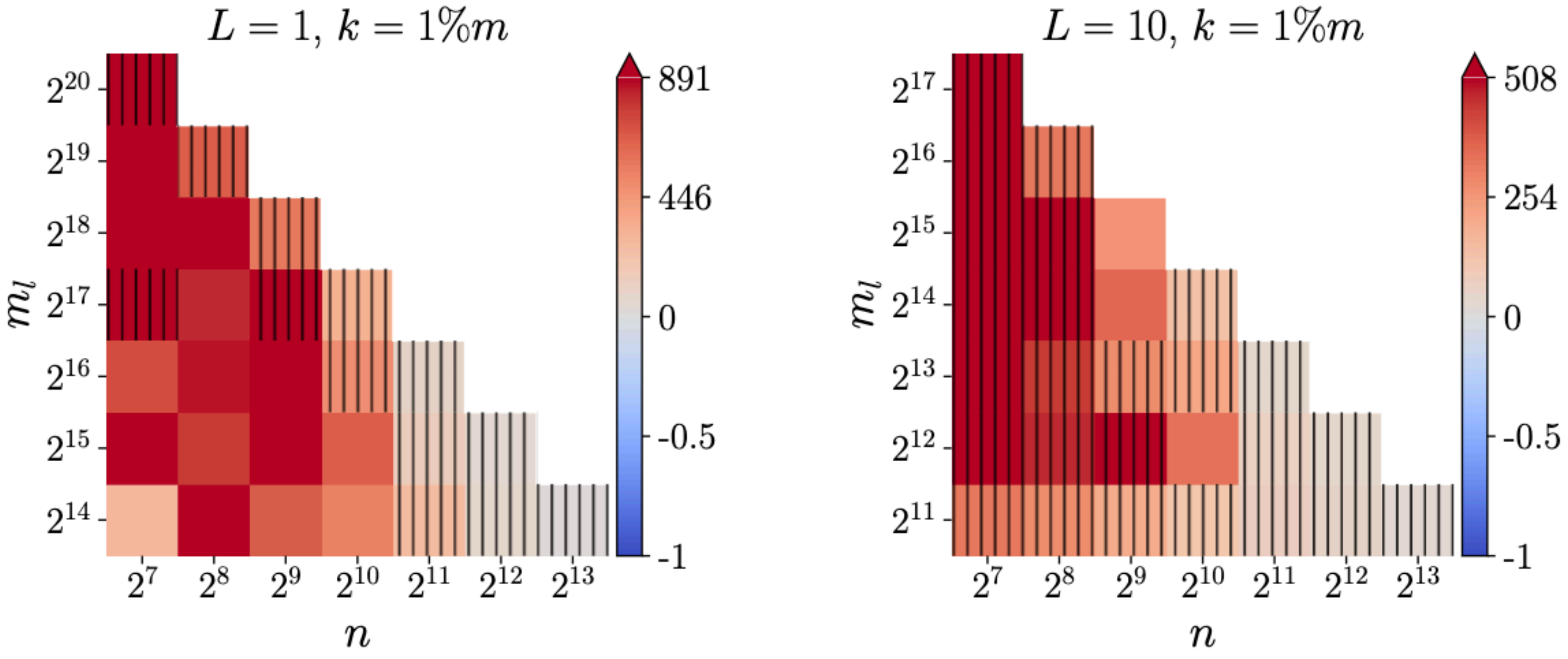
GRB[2]: linear objective



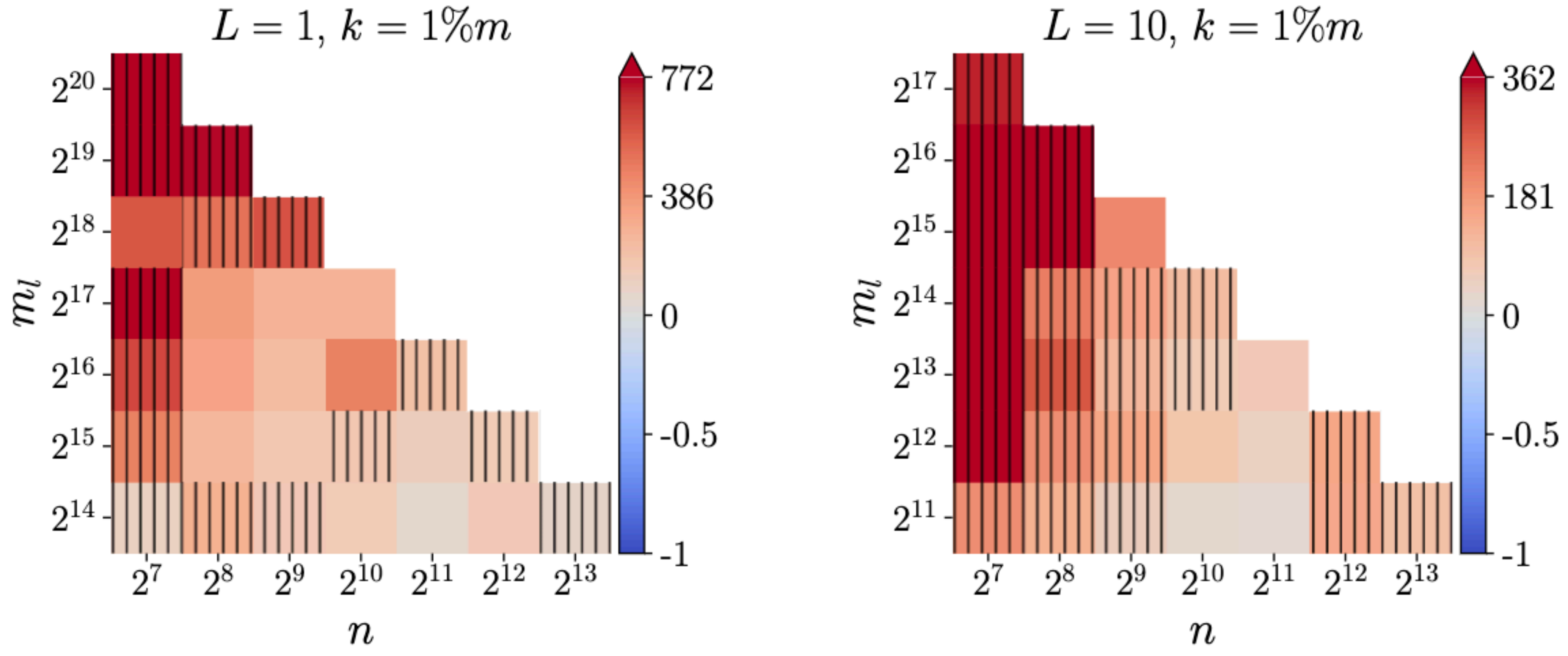
GRB[2]: quadratic objective



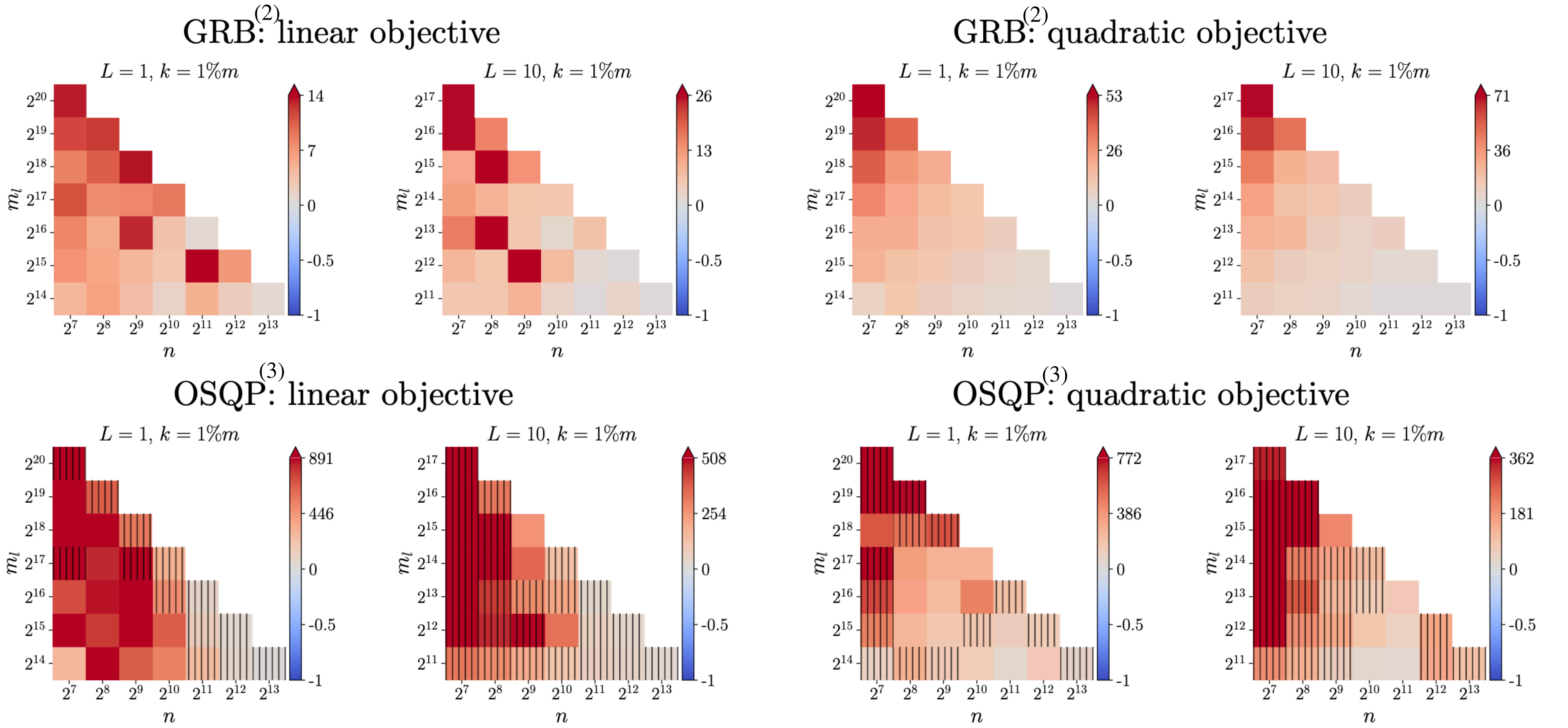
OSQP[3]: linear objective



OSQP[3]: quadratic objective



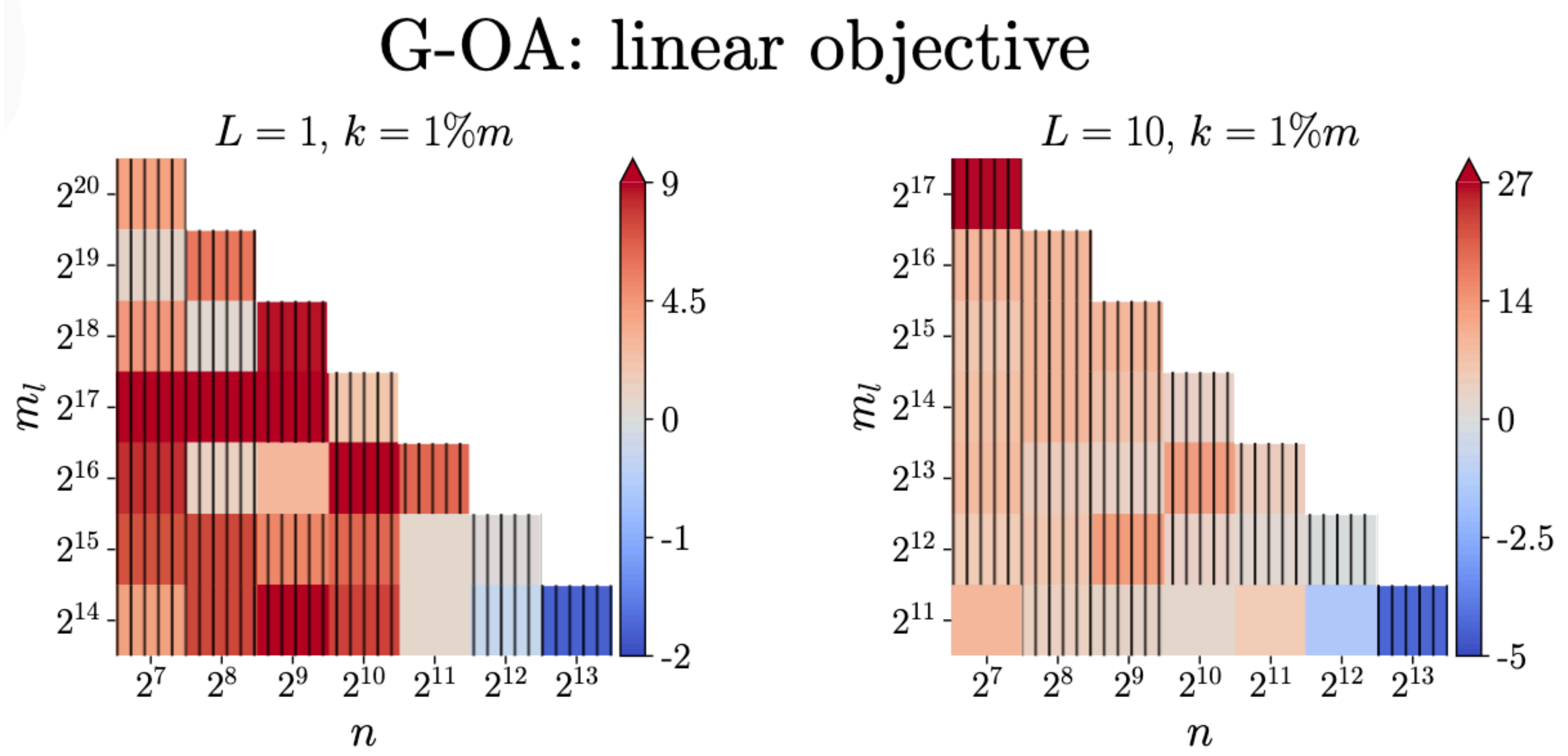
Numerical experiments: synthetic instances

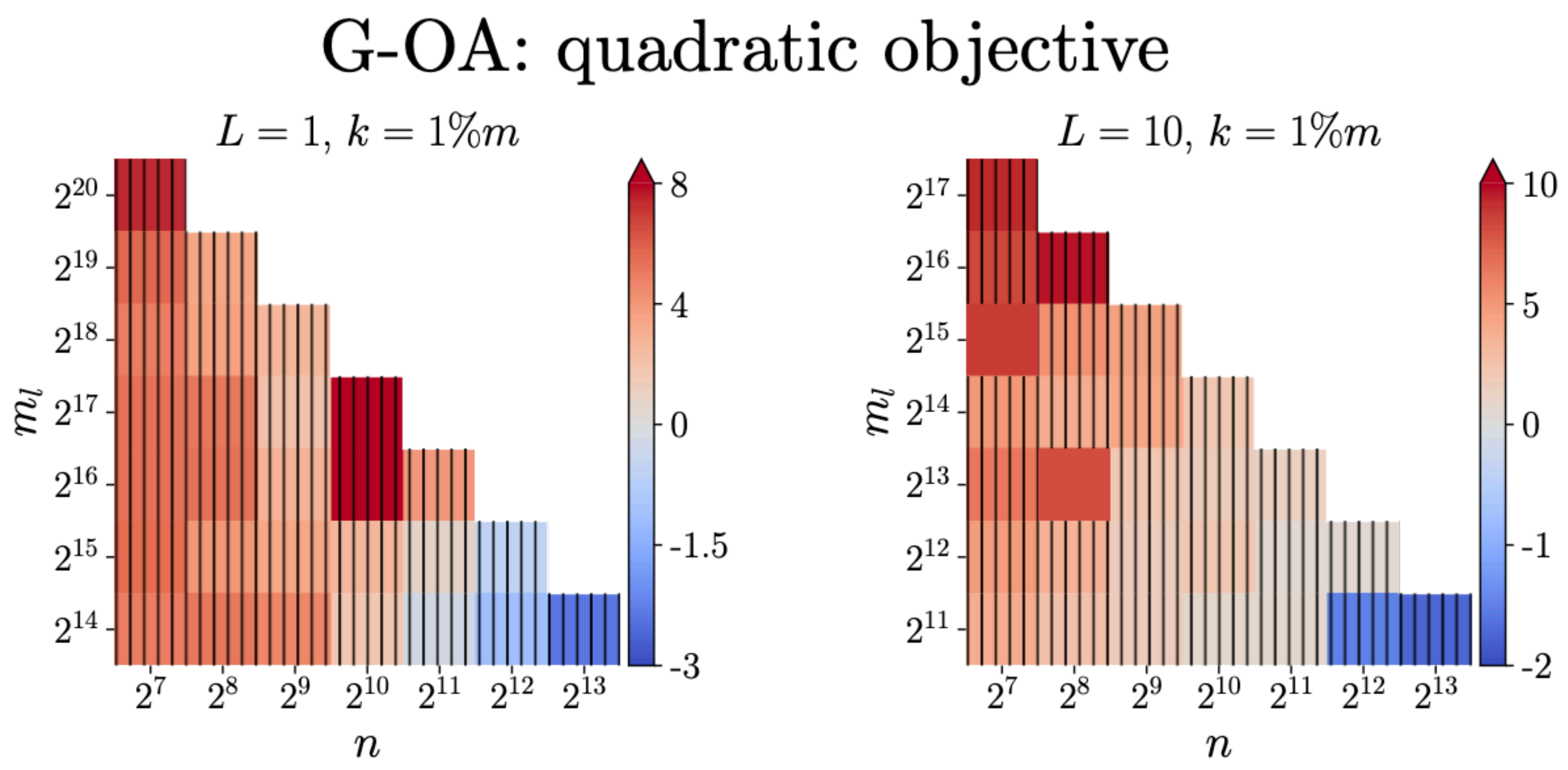


(2) GRB v11. Tolerance = 1e-8. Best of P-simplex, D-simplex, and Barrier.

(3) OSQP v0.6. Tolerance = 1e-3. Conducted with default linear solver. Vertical lines indicate OSQP timed out or failed to achieve tolerance.

Numerical experiments: synthetic instances





Numerical experiments: synthetic instances

Problem	KKT residual		Time_a : $\epsilon = 10^{-3}$		Time_b : $\epsilon = 10^{-6}$		Time_c : $\epsilon = 10^{-8}$		% Time_c: Sort		% Time_c: $V^{-1}v$		% Time_c: $\nabla\varphi$		ALM | SSN iter	
(m, n)	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.
								$L = 1,\ k = 1\%m$								
$(2^{20}, 2^7)$	3.3e-9	1.4e-9	4.0e0	4.7e0	1.0e1	8.8e0	2.6e1	1.1e1	42	16	4	6	12	20	18 | 100	17 | 70
$(2^{19}, 2^7)$	6.7e-9	2.0e-9	1.9e0	1.9e0	7.3e0	3.8e0	1.3e1	4.4e0	34	14	5	7	16	21	19 | 130	17 | 60
$(2^{18}, 2^7)$	1.5e-10	2.6e-9	9.6e-1	1.1e0	2.5e0	1.9e0	5.5e0	2.1e0	39	14	5	7	14	21	18 | 90	16 | 50
$(2^{17}, 2^7)$	2.3e-9	9.7e-9	4.5e-1	4.7e-1	1.4e0	8.7e-1	1.6e0	9.5e-1	22	10	8	8	19	22	15 | 70	15 | 50
$(2^{19}, 2^8)$	7.0e-9	1.2e-9	5.4e0	4.7e0	1.2e1	7.6e0	1.7e1	8.9e0	18	10	8	9	24	27	18 | 120	15 | 70
$(2^{18}, 2^8)$	6.3e-9	9.9e-9	2.1e0	2.6e0	6.8e0	4.1e0	9.8e0	4.4e0	21	11	9	10	23	26	19 | 140	14 | 70
$(2^{17}, 2^8)$	4.6e-10	4.5e-9	1.4e0	1.1e0	2.8e0	1.7e0	4.0e0	1.9e0	18	7	11	11	23	27	18 | 120	13 | 60
$(2^{18}, 2^9)$	4.0e-10	9.0e-10	6.3e0	6.0e0	1.5e1	8.4e0	1.8e1	9.1e0	9	5	16	13	29	31	20 | 160	13 | 80
$(2^{17}, 2^9)$	9.3e-9	7.1e-9	4.0e0	3.3e0	7.4e0	4.7e0	8.4e0	5.0e0	8	4	21	17	27	30	17 | 140	13 | 90
$(2^{17}, 2^{10})$	9.4e-9	1.1e-9	1.1e1	9.1e0	2.1e1	1.1e1	2.3e1	1.2e1	3	2	38	24	24	30	16 | 170	13 | 100

Numerical experiments: synthetic instances

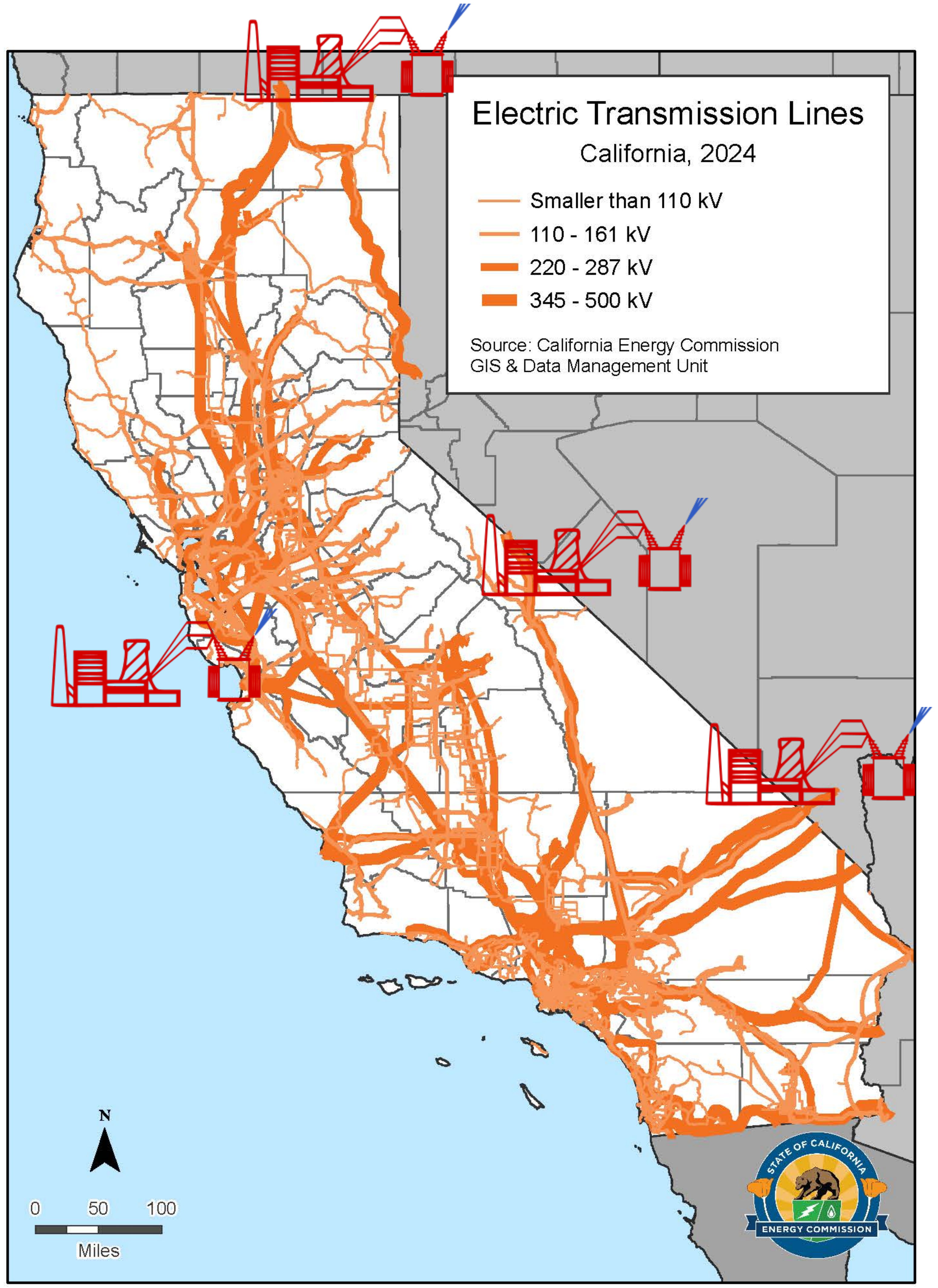
Problem	KKT residual		$\text{Time}_\text{a}: \epsilon = 10^{-3}$		$\text{Time}_\text{b}: \epsilon = 10^{-6}$		$\text{Time}_\text{c}: \epsilon = 10^{-8}$		% Time_c: Sort		% Time_c: $V^{-1}v$		% Time_c: $\nabla\varphi$		ALM | SSN iter	
(m, n)	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.
									$L = 1,\ k = 10\%m$							
$(2^{20}, 2^7)$	7.5e-9	7.7e-9	7.3e0	8.2e0	2.7e1	1.5e1	2.7e1	1.8e1	19	17	30	31	15	14	18 | 110	14 | 80
$(2^{19}, 2^7)$	3.5e-9	3.7e-9	4.7e0	3.9e0	1.1e1	6.9e0	1.3e1	8.4e0	20	17	29	31	14	15	20 | 100	15 | 80
$(2^{18}, 2^7)$	7.2e-9	2.4e-9	2.2e0	2.2e0	7.8e0	4.0e0	8.1e0	4.5e0	16	17	32	31	17	15	19 | 160	15 | 80
$(2^{17}, 2^7)$	7.5e-11	2.1e-9	1.0e0	9.8e-1	2.9e0	1.7e0	2.9e0	2.0e0	22	16	27	30	15	15	18 | 100	15 | 70
$(2^{19}, 2^8)$	1.7e-9	4.9e-9	1.0e1	1.1e1	3.0e1	1.7e1	3.1e1	1.8e1	13	11	37	37	18	18	19 | 160	14 | 100
$(2^{18}, 2^8)$	2.7e-9	8.6e-9	6.3e0	5.5e0	1.3e1	8.3e0	1.4e1	8.9e0	14	13	36	36	18	18	18 | 140	14 | 100
$(2^{17}, 2^8)$	9.9e-10	1.7e-9	3.0e0	2.5e0	7.7e0	3.7e0	8.0e0	4.1e0	16	7	35	40	17	19	17 | 160	14 | 100
$(2^{18}, 2^9)$	5.5e-9	1.9e-9	2.2e1	1.5e1	3.4e1	2.1e1	3.8e1	2.2e1	10	6	42	43	19	20	22 | 220	16 | 130
$(2^{17}, 2^9)$	6.3e-9	1.8e-9	1.1e1	8.3e0	1.7e1	1.1e1	1.8e1	1.1e1	6	6	46	44	19	20	21 | 200	15 | 140
$(2^{17}, 2^{10})$	9.6e-9	6.8e-9	3.4e1	2.0e1	5.3e1	2.6e1	5.6e1	2.7e1	4	3	56	50	17	20	22 | 280	15 | 160

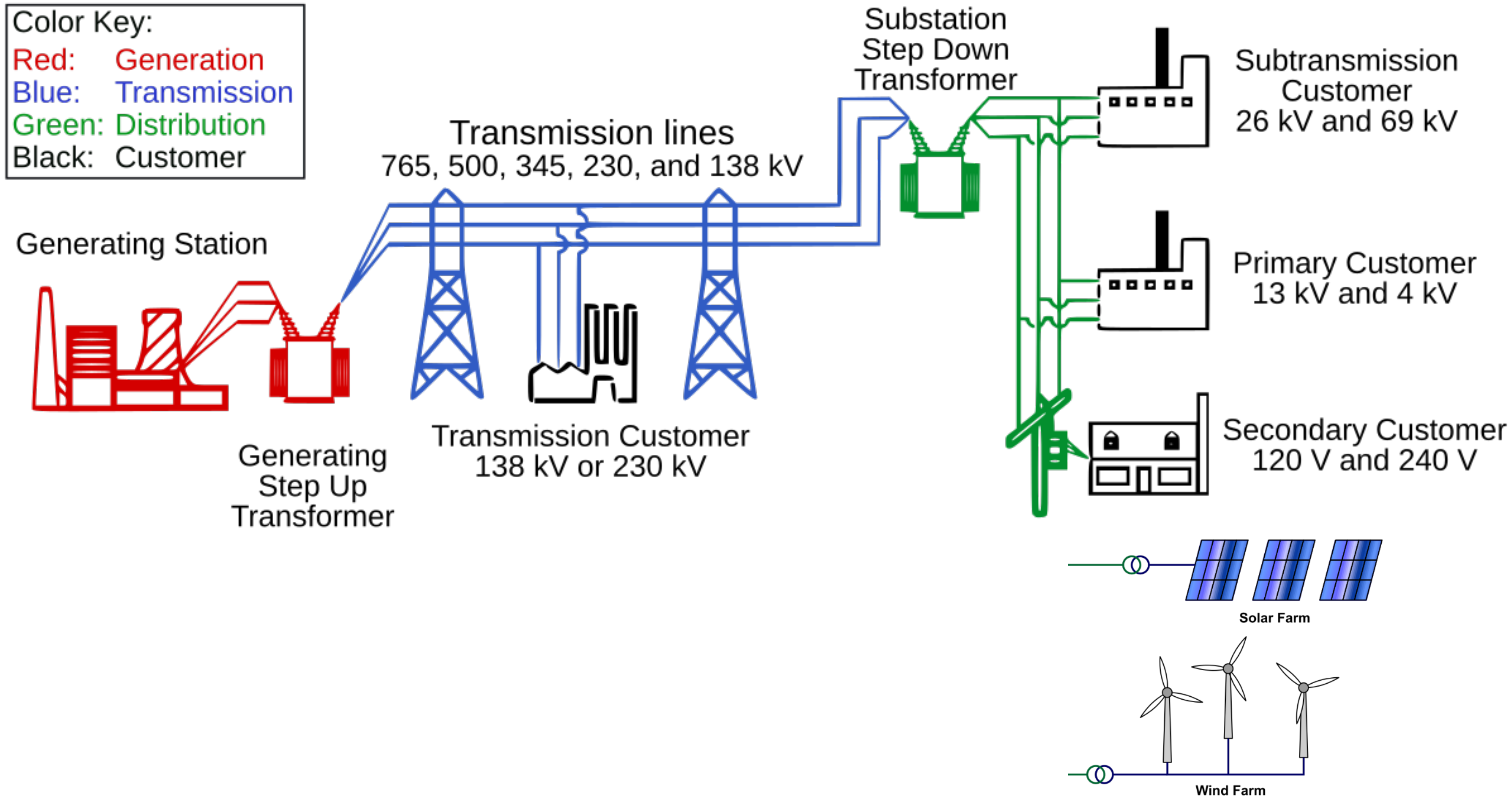
Numerical experiments: synthetic instances

Problem	KKT residual		$\text{Time}_\text{a}: \epsilon = 10^{-3}$		$\text{Time}_\text{b}: \epsilon = 10^{-6}$		$\text{Time}_\text{c}: \epsilon = 10^{-8}$		% Time_c: Sort		% Time_c: $V^{-1}v$		% Time_c: $\nabla\varphi$		ALM | SSN iter	
(m,n)	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.	lin.	quad.
$L=1,\ k=10\%m$																
$(2^{20},2^{7})$	7.5e-9	7.7e-9	7.3e0	8.2e0	2.7e1	1.5e1	2.7e1	1.8e1	19	17	30	31	15	14	18 | 110	14 | 80
$(2^{19},2^{7})$	3.5e-9	3.7e-9	4.7e0	3.9e0	1.1e1	6.9e0	1.3e1	8.4e0	20	17	29	31	14	15	20 | 100	15 | 80
$(2^{18},2^{7})$	7.2e-9	2.4e-9	2.2e0	2.2e0	7.8e0	4.0e0	8.1e0	4.5e0	16	17	32	31	17	15	19 | 160	15 | 80
$(2^{17},2^{7})$	7.5e-11	2.1e-9	1.0e0	9.8e-1	2.9e0	1.7e0	2.9e0	2.0e0	22	16	27	30	15	15	18 | 100	15 | 70
$(2^{19},2^{8})$	1.7e-9	4.9e-9	1.0e1	1.1e1	3.0e1	1.7e1	3.1e1	1.8e1	13	11	37	37	18	18	19 | 160	14 | 100
$(2^{18},2^{8})$	2.7e-9	8.6e-9	6.3e0	5.5e0	1.3e1	8.3e0	1.4e1	8.9e0	14	13	36	36	18	18	18 | 140	14 | 100
$(2^{17},2^{8})$	9.9e-10	1.7e-9	3.0e0	2.5e0	7.7e0	3.7e0	8.0e0	4.1e0	16	7	35	40	17	19	17 | 160	14 | 100
$(2^{18},2^{9})$	5.5e-9	1.9e-9	2.2e1	1.5e1	3.4e1	2.1e1	3.8e1	2.2e1	10	6	42	43	19	20	22 | 220	16 | 130
$(2^{17},2^{9})$	6.3e-9	1.8e-9	1.1e1	8.3e0	1.7e1	1.1e1	1.8e1	1.1e1	6	6	46	44	19	20	21 | 200	15 | 140
$(2^{17},2^{10})$	9.6e-9	6.8e-9	3.4e1	2.0e1	5.3e1	2.6e1	5.6e1	2.7e1	4	3	56	50	17	20	22 | 280	15 | 160

- When $g_i(x;\xi)$ is expensive and has "easy" second-order information, ALM+SSN is expected to perform even better

Application: risk-averse power grid dispatch

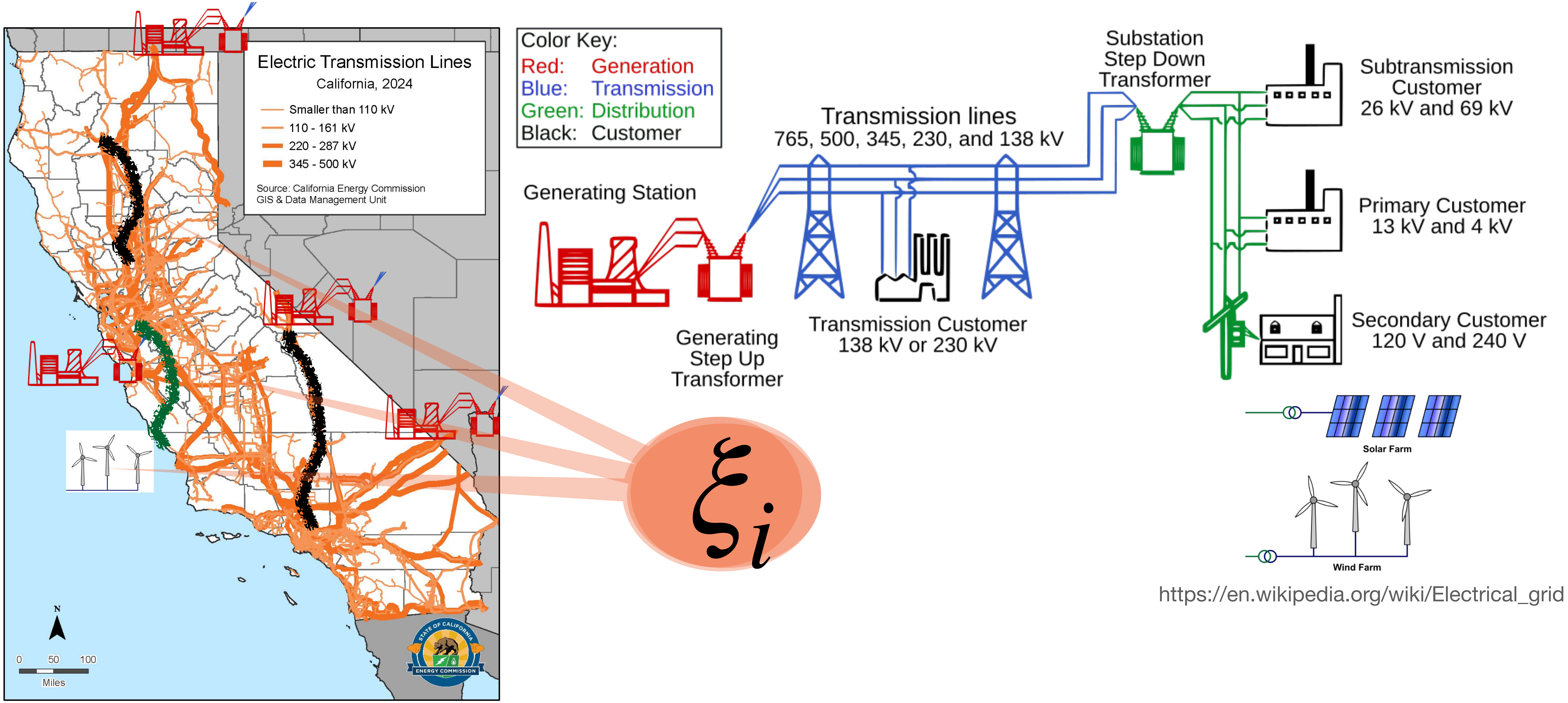




https://en.wikipedia.org/wiki/Electrical_grid

https://www.energy.ca.gov/programs-and-topics/topics/california-transmission-system/transmission-challenge

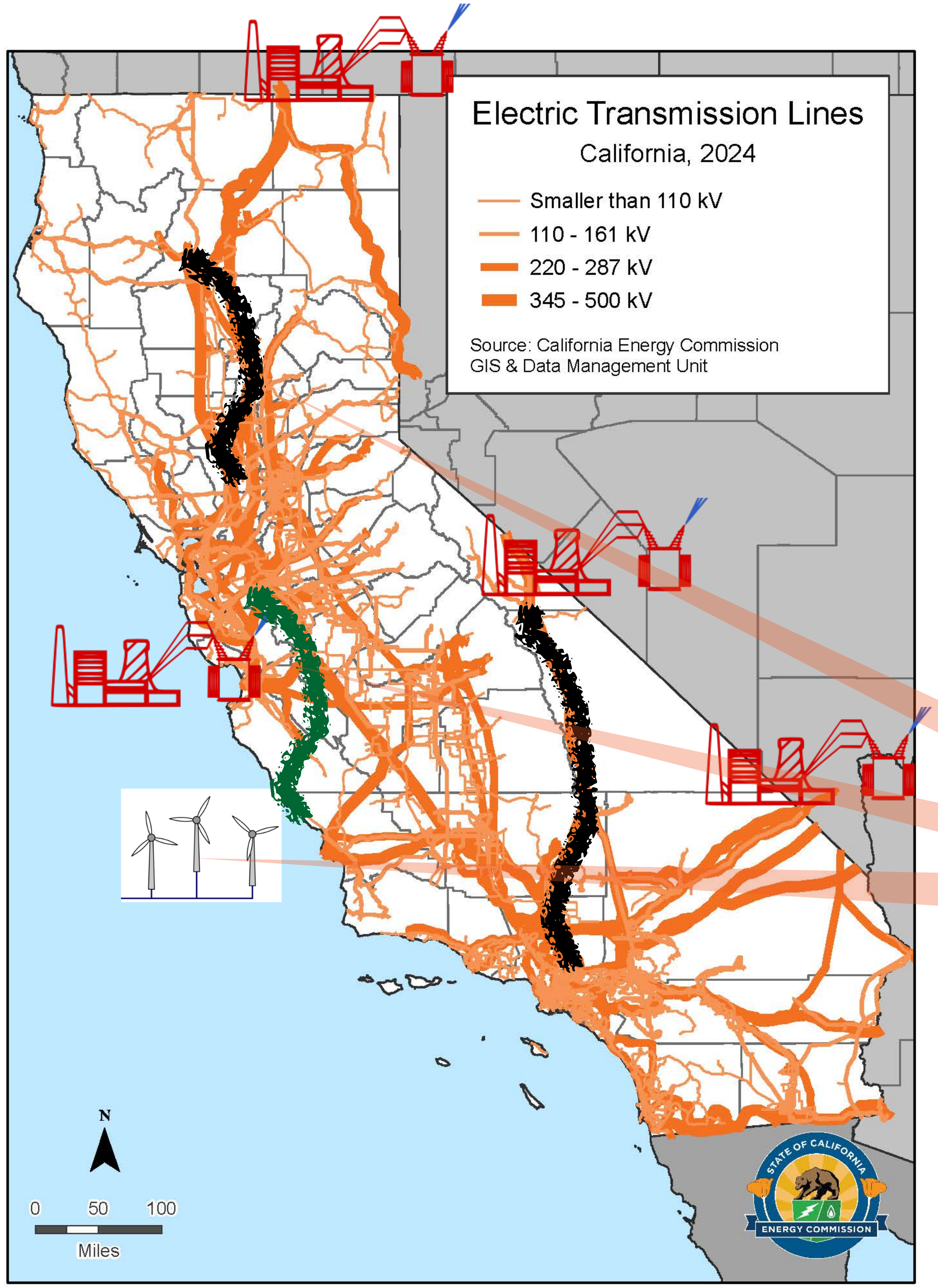
Application: risk-averse power grid dispatch



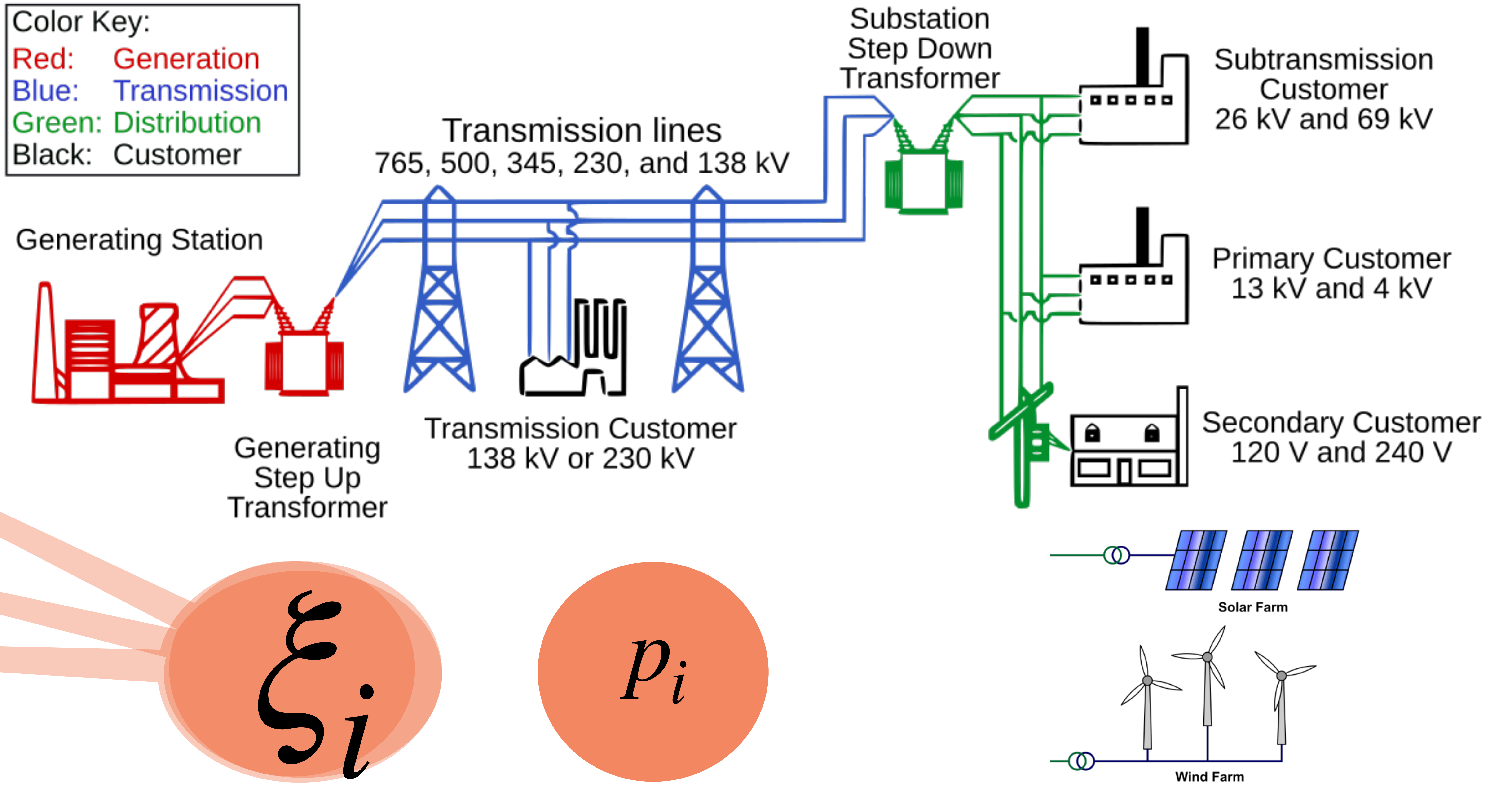
https://en.wikipedia.org/wiki/Electrical_grid

https://www.energy.ca.gov/programs-and-topics/topics/california-transmission-system/transmission-challenge

Application: risk-averse power grid dispatch



ξ_i p_i



https://en.wikipedia.org/wiki/Electrical_grid

https://www.energy.ca.gov/programs-and-topics/topics/california-transmission-system/transmission-challenge

Application: risk-averse dispatch

$$\min_{x \in X} f(x) + \mathsf{CVaR}_\alpha(G(x;\Xi)) : g_i(x) \leq 0,\ i = 1,\ldots,p$$

$$G_i(x;\xi_i) = \min_y b^\top y + \frac{1}{2} y^\top Q y : A_i y + b_i = C_i x,\ y \geq 0$$

- **Consider V_i as either DCOPF (convex) or ACOPF (nonconvex)**

Extension to weighted SAA

Extension to weighted SAA

- Nonuniform scenario probabilities: destroys isotonic/Z- structure of top-k-sum

Extension to weighted SAA

- Nonuniform scenario probabilities: destroys isotonic/Z- structure of top-k-sum
- Solve projection problem as a sequence of scalar proximal problems

$$\mathsf{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\Big\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}_\alpha^c(r^0)\Big\}$$

Extension to weighted SAA

- Nonuniform scenario probabilities: destroys isotonic/Z- structure of top-k-sum
- Solve projection problem as a sequence of scalar proximal problems

$$\mathrm{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}_\alpha^c(r^0)\right\}$$

- Smooth the CVaR (inf-convolution):

$$\mathsf{SCVaR}_\alpha^c(r) := \max_{q \in Q_\alpha^p} \langle q, r\rangle - \frac{1}{2c}\|q\|_2^2 \text{ and } Q_\alpha^p := \{q \in \mathbb{R}^m : 0 \le q \le \frac{p}{1-\alpha},\ \textstyle\sum_{i=1}^m q_i = 1\}$$

Extension to weighted SAA

- Nonuniform scenario probabilities: destroys isotonic/Z- structure of top-k-sum
- Solve projection problem as a sequence of scalar proximal problems

$$\mathrm{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}_\alpha^c(r^0)\right\}$$

- Smooth the CVaR (inf-convolution):

$$\mathsf{SCVaR}_\alpha^c(r) := \max_{q\in Q_\alpha^p}\langle q, r\rangle - \frac{1}{2c}\|q\|_2^2$$ and Q_α^p :

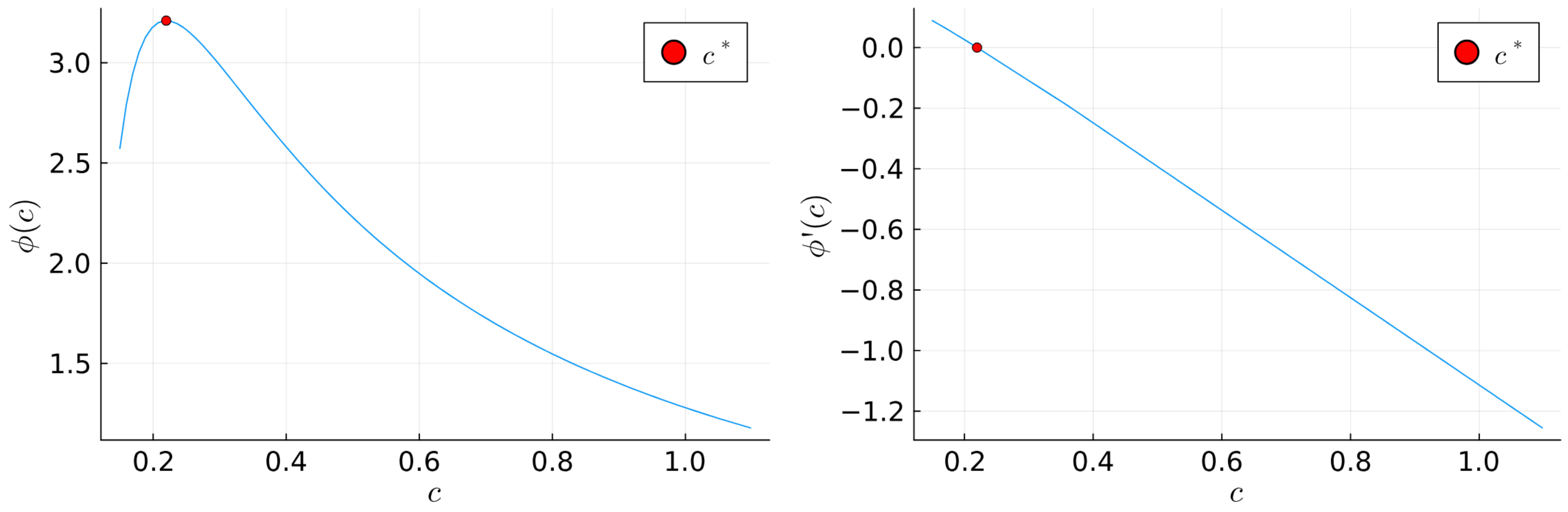


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}_\alpha^c(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

Extension to weighted SAA

Extension to weighted SAA

- Solve projection problem as a sequence of scalar proximal problems

$$\mathrm{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}^c_\alpha(r^0)\right\}$$

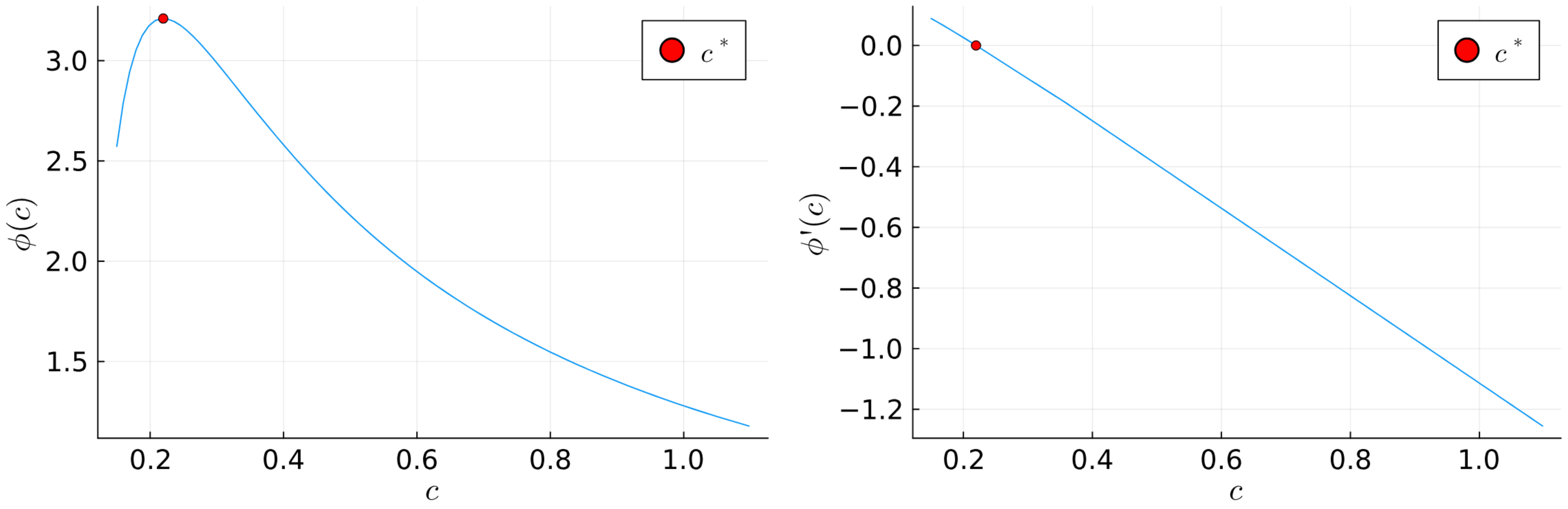


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}^c_\alpha(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

Extension to weighted SAA

- Solve projection problem as a sequence of scalar proximal problems

$$\mathrm{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}^c_\alpha(r^0)\right\}$$

- Interpret smooth CVaR as projection problem via its dual

$$\mathsf{SCVaR}^c_\alpha(r^0) = -\frac{1}{c}\min_{q\in Q_\alpha}\left\{\frac{1}{2}\|q - cr\|_2^2\right\} + \frac{c}{2}\|r\|_2^2$$

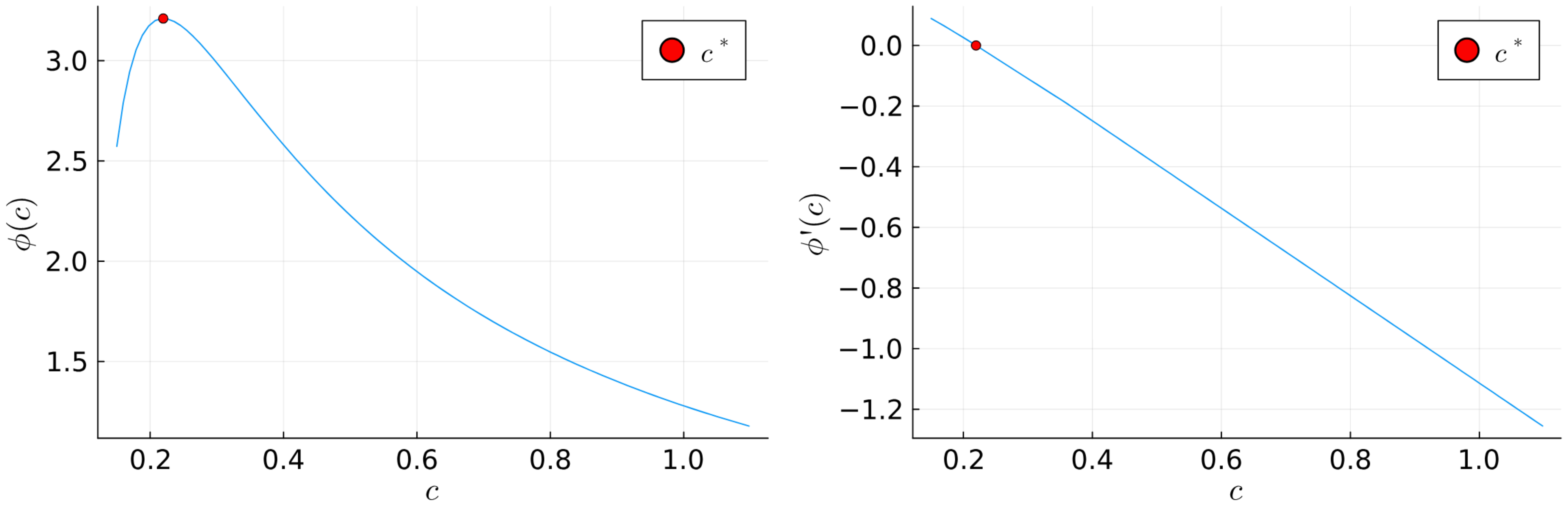


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}^c_\alpha(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

Extension to weighted SAA

- Solve projection problem as a sequence of scalar proximal problems

$$\text{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\text{SCVaR}^c_\alpha(r^0)\right\}$$

- Interpret smooth CVaR as projection problem via its dual

- Projection is pw-linear; again consider using (scalar) SSN to solve for c^*

$$\text{SCVaR}^c_\alpha(r^0) = -\frac{1}{c}\min_{q\in Q_\alpha}\left\{\frac{1}{2}\|q - cr\|_2^2\right\} + \frac{c}{2}\|r\|_2^2$$

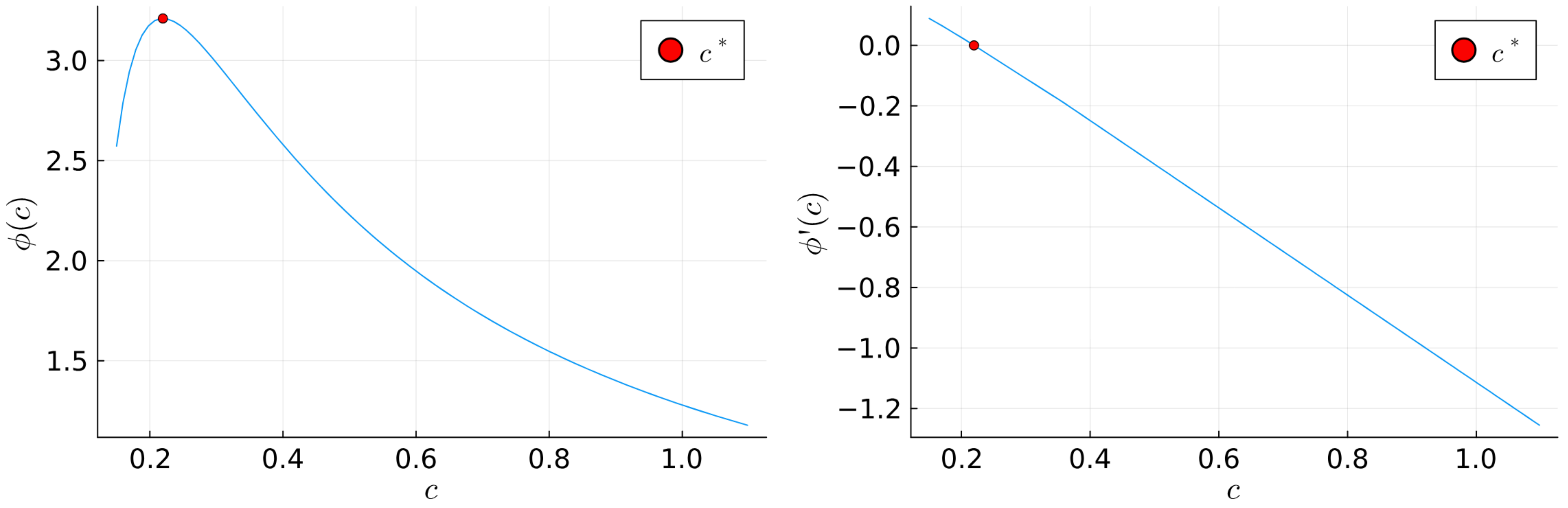


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}^c_\alpha(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

Extension to weighted SAA

- Solve projection problem as a sequence of scalar proximal problems

$$\text{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\text{SCVaR}^c_\alpha(r^0)\right\}$$

- Interpret smooth CVaR as projection problem via its dual
- Projection is pw-linear; again consider using (scalar) SSN to solve for c^*
- Each c-projection is a cts. quad. knapsack $O(m)$

$$\text{SCVaR}^c_\alpha(r^0) = -\frac{1}{c}\min_{q\in Q_\alpha}\left\{\frac{1}{2}\|q - cr\|_2^2\right\} + \frac{c}{2}\|r\|_2^2$$

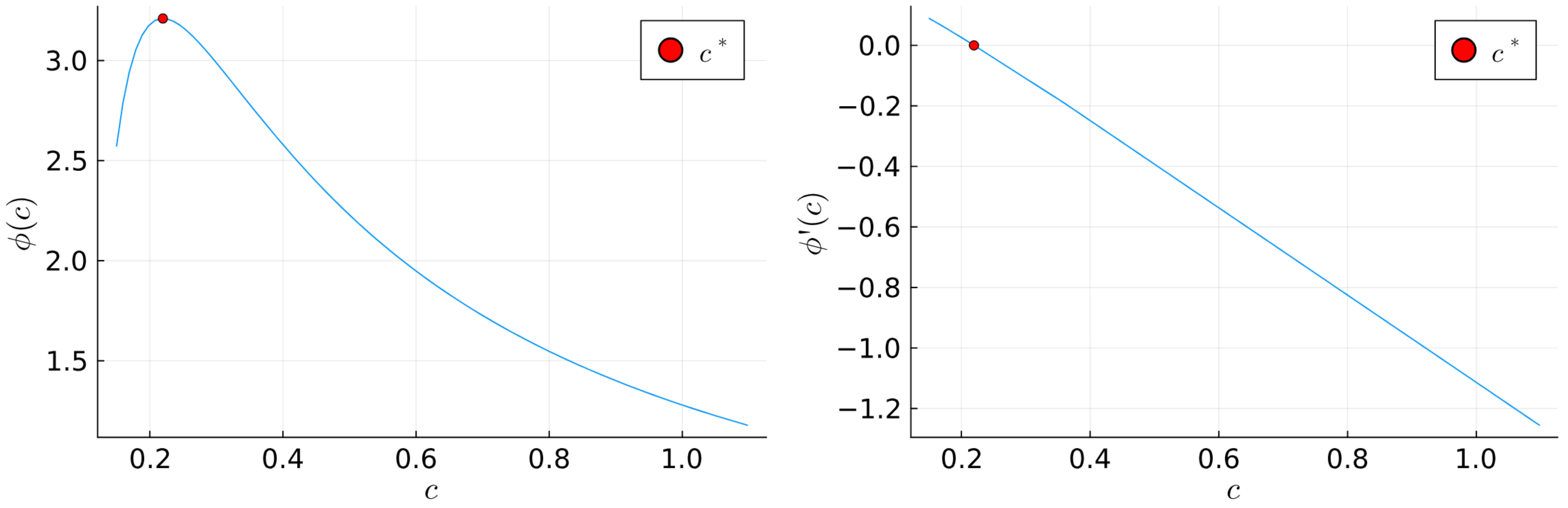


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}^c_\alpha(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

Extension to weighted SAA

- Solve projection problem as a sequence of scalar proximal problems

$$\mathrm{dist}(r^0, B_\alpha)^2 = \sup_{c>0}\left\{\phi(c) := \frac{1}{c}\mathsf{SCVaR}^c_\alpha(r^0)\right\}$$

- Interpret smooth CVaR as

$$\mathsf{SCVaR}^c_\alpha(r^0) = -\frac{1}{c}\min_{q\in Q_\alpha}\left\{\frac{1}{2}\|q - cr\|_2^2\right\} + \frac{c}{2}\|r\|_2^2$$

SSN + SSN (+ SSN)
(in progress)

- Each c-projection is a cts. quad. knapsack $O(m)$

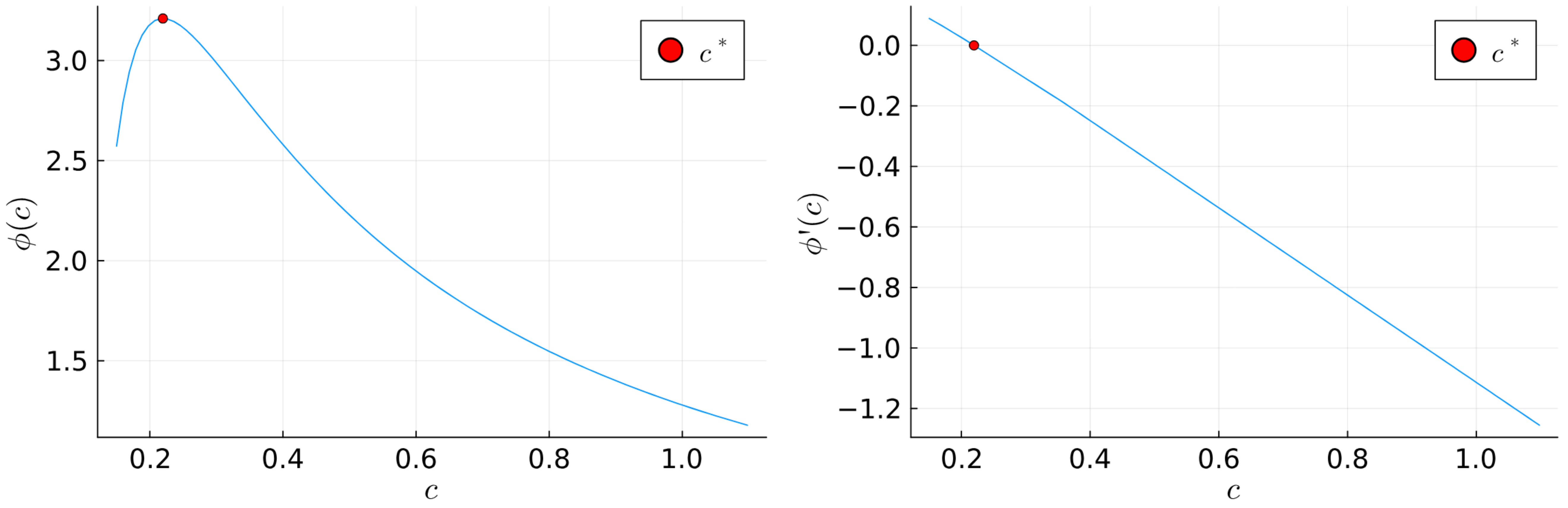


Figure 1: Smooth CVaR $\phi(c) := \frac{1}{c}\mathsf{C}^c_\alpha(r^0)$ (left) and $\phi'(c)$ (right) versus c for fixed α, p, r^0.

[Cominetti, Mascarenhas, and Silva: 2014 https://dx.doi.org/10.1007/s12532-014-0066-y]

Questions?

Thank you!!

Appendix

Extension to weighted SAA

- Nonuniform scenario probabilities: destroys isotonic/Z- structure of top-k-sum set
- Smooth the CVaR (inf-convolution):

$\mathsf{SCVaR}^c_\alpha(r) := \max_{q \in Q^p_\alpha} \langle q, r \rangle - \frac{1}{2c}\|q\|_2^2$ and $Q^{\boxed{p}}_\alpha := \{q \in \mathbb{R}^m : 0 \le q \le \boxed{\frac{p}{1-\alpha}}, \sum_{i=1}^m q_i = 1$

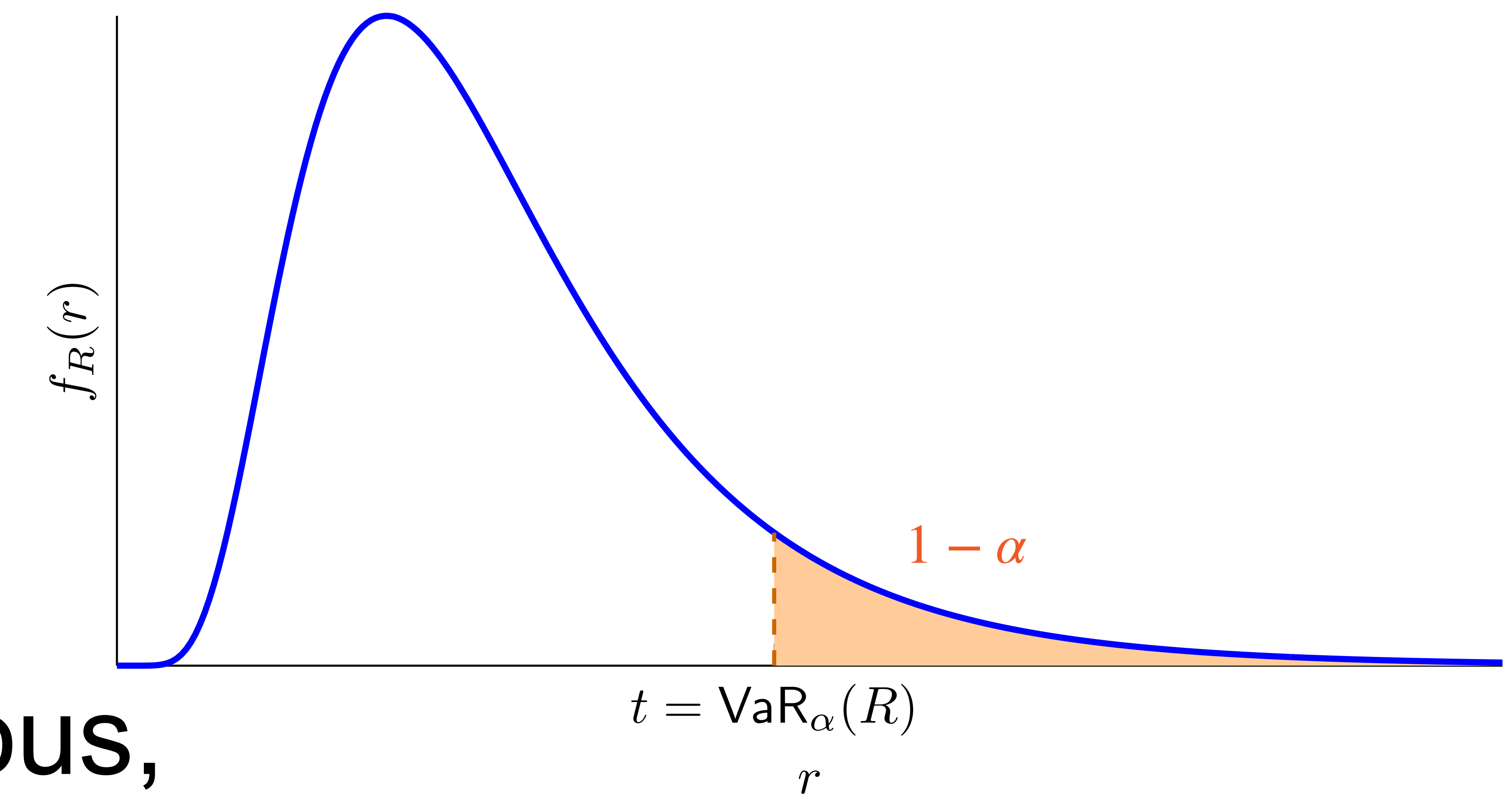
- In its "dual" form, it's just smoothing the $\max\{\cdot, 0\}$

$$c \cdot \mathsf{SCVaR}^c(r) = \max_{q \in Q^p_\alpha} \langle q, r \rangle - \frac{1}{2c}\|q\|_2^2 = \min_{z \ge 0, t} t + \frac{1}{1-\alpha}\langle p, z \rangle + \frac{c}{2}\|[r - t1 - z]_+\|_2^2$$

- For large c, $\mathsf{SCVaR}^c_\alpha(r) \uparrow \mathsf{CVaR}_\alpha(r)$ doesn't have curvature

$$
\begin{aligned}
\tfrac{1}{2}\mathsf{dist}(r^0, B_\alpha)^2 &= \tfrac{1}{2}\left\|r^0 - \mathsf{proj}_{B_\alpha}(r^0)\right\|_2^2 = \min_x \tfrac{1}{2}\|x - r^0\|_2^2 + \delta_{B_\alpha}(x) \\
&= \min_x \max_{\lambda \geq 0} \tfrac{1}{2}\|x - r^0\|_2^2 + \lambda \cdot \mathsf{CVaR}_\alpha(x) \\
&= \max_{\lambda \geq 0} \min_x \tfrac{1}{2}\|x - r^0\|_2^2 + \lambda \cdot \mathsf{CVaR}_\alpha(x) && \text{(strong duality)} \\
&= \max_{\lambda \geq 0} \min_x \max_{q \in Q_\alpha} \tfrac{1}{2}\|x - r^0\|_2^2 + \lambda \cdot q^\top x \\
&= \max_{\lambda \geq 0} \max_{q \in Q_\alpha} \min_x \tfrac{1}{2}\|x - r^0\|_2^2 + \lambda \cdot q^\top x && \text{(LP duality)} \\
&= \max_{\lambda \geq 0} \max_{q \in Q_\alpha} \lambda q^\top r^0 - \tfrac{\lambda^2}{2}\|q\|_2^2 && \text{(eliminate } x\text{)} \\
&= \sup_{c > 0} \tfrac{1}{c} \max_{q \in Q_\alpha} q^\top r^0 - \tfrac{1}{2c}\|q\|_2^2 && (\lambda := 1/c, \;\; r^0 \notin B_\alpha \Rightarrow \lambda > 0) \\
&= \sup_{c > 0} \tfrac{1}{c} \mathsf{C}_\alpha^c(r^0) \\
&= \sup_{v \in \mathsf{cone}\, Q_\alpha} v^\top r^0 - \tfrac{1}{2}\|v\|_2^2 && (v := c \cdot q).
\end{aligned}
$$

Superquantile



- Coherent risk measure: convex, positive-homogeneous, translation equivariant, i.e., $\mathsf{CVaR}_\alpha(R-\tau) = \mathsf{CVaR}_\alpha(R) - \tau$

- CVaR is polyhedral (value function of an LP): non smooth, multiple solutions

- Common reformulation (Rockafellar-Uryasev) linearizes max:

$$t + \frac{1}{1-\alpha}\sum_{i=1}^{m}\frac{1}{m}z_i \ \text{ with } \ z_i \geq R_i - t,\ z_i \geq 0,\ i = 1,\dots,m$$