Efficient exact recomputation of linear modifications of the constraints matrix in Linear Programming

EURO 2024

Guillaume Derval, Bardhyl Miftari, Damien Ernst, Quentin Louveaux
Montefiore Institute, University of Liège

LIÈGE université

Uncertainty

Linear programming

- In LP optimization
  - Formalize problem in terms of
    - Constraints
    - Objective function
  - Get one optimal solution

Parameters ← Hypothesis

Uncertainty

Linear programming

- In LP optimization
  - Formalize problem in terms of
    - Constraints
    - Objective function
  - Get one optimal solution

Parameters ← Hypothesis

Further assessment and analysis

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{array}{ll} \min & c^t x \\ \text{s.t} & Ax \le b \end{array}$$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{aligned} \min \quad & c^t x + \lambda c_\lambda^t x \\ \text{s.t} \quad & Ax \leq b \end{aligned}$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{aligned} \min \quad & c^t x + \lambda c_\lambda^t x \\ \text{s.t} \quad & Ax \leq b + \lambda b_\lambda \end{aligned}$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$
- Modification on the right-hand side $+\lambda b_\lambda$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{aligned} \min \quad & c^t x + \lambda c_\lambda^t x \\ \text{s.t} \quad & (A + \lambda D) x \leq b + \lambda b_\lambda \end{aligned}$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$
- Modification on the right-hand side $+\lambda b_\lambda$
- Modification on the left-hand side $+\lambda D$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{aligned} \min \quad & c^t x + \lambda c_\lambda^t x \\ \text{s.t} \quad & (A + \lambda D)x \leq b + \lambda b_\lambda \end{aligned}$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$
- Modification on the right-hand side $+\lambda b_\lambda$
- Modification on the left-hand side $+\lambda D$

} Discussed a LOT in the literature

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$\begin{aligned} \min \quad & c^t x + \lambda c_\lambda^t x \\ \text{s.t} \quad & (A + \lambda D)x \leq b + \lambda b_\lambda \end{aligned}$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$
- Modification on the right-hand side $+\lambda b_\lambda$

} Discussed a LOT in the literature

- Modification on the left-hand side $+\lambda D$ → Not much ?

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$
\begin{aligned}
f(\lambda) = \min \quad & c^t x + \lambda c_\lambda^t x \\
\text{s.t} \quad & (A + \lambda D)x \leq b + \lambda b_\lambda
\end{aligned}
$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$
- Modification on the right-hand side $+\lambda b_\lambda$
- Modification on the left-hand side $+\lambda D$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$f(\lambda) = \min \quad c^t x + \lambda c_\lambda^t x$$
$$\text{s.t} \quad (A + \lambda D)x \leq b + \lambda b_\lambda$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$ $f(\lambda)$ concave and piecewise linear!
- Modification on the right-hand side $+\lambda b_\lambda$
- Modification on the left-hand side $+\lambda D$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$f(\lambda) = \min \quad c^t x + \lambda c_\lambda^t x$$
$$\text{s.t} \quad (A + \lambda D)x \leq b + \lambda b_\lambda$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$ — $f(\lambda)$ concave and piecewise linear!
- Modification on the right-hand side $+\lambda b_\lambda$ — $f(\lambda)$ convex and piecewise linear!
- Modification on the left-hand side $+\lambda D$

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$f(\lambda) = \min \quad c^t x + \lambda c_\lambda^t x$$
$$\text{s.t} \quad (A + \lambda D)x \leq b + \lambda b_\lambda$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$ — $f(\lambda)$ concave and piecewise linear!
- Modification on the right-hand side $+\lambda b_\lambda$ — $f(\lambda)$ convex and piecewise linear!
- Modification on the left-hand side $+\lambda D$ — $f(\lambda)$ piecewise rational function...

Formalization

Parametric uncertainty

- Generic parametric linear optimization problem:

$$f(\lambda) = \min \quad c^t x + \lambda c_\lambda^t x$$
$$\text{s.t} \quad (A + \lambda D)x \leq b + \lambda b_\lambda$$

- Modification in the objective coefficients $+\lambda c_\lambda^t x$ — $f(\lambda)$ concave and piecewise linear!
- Modification on the right-hand side $+\lambda b_\lambda$ — $f(\lambda)$ convex and piecewise linear!
- Modification on the left-hand side $+\lambda D$ — $f(\lambda)$ piecewise rational function...

Note : The left-hand side modification $+\lambda D$ encapsulates the other modifications

Formalization

Parametric uncertainty

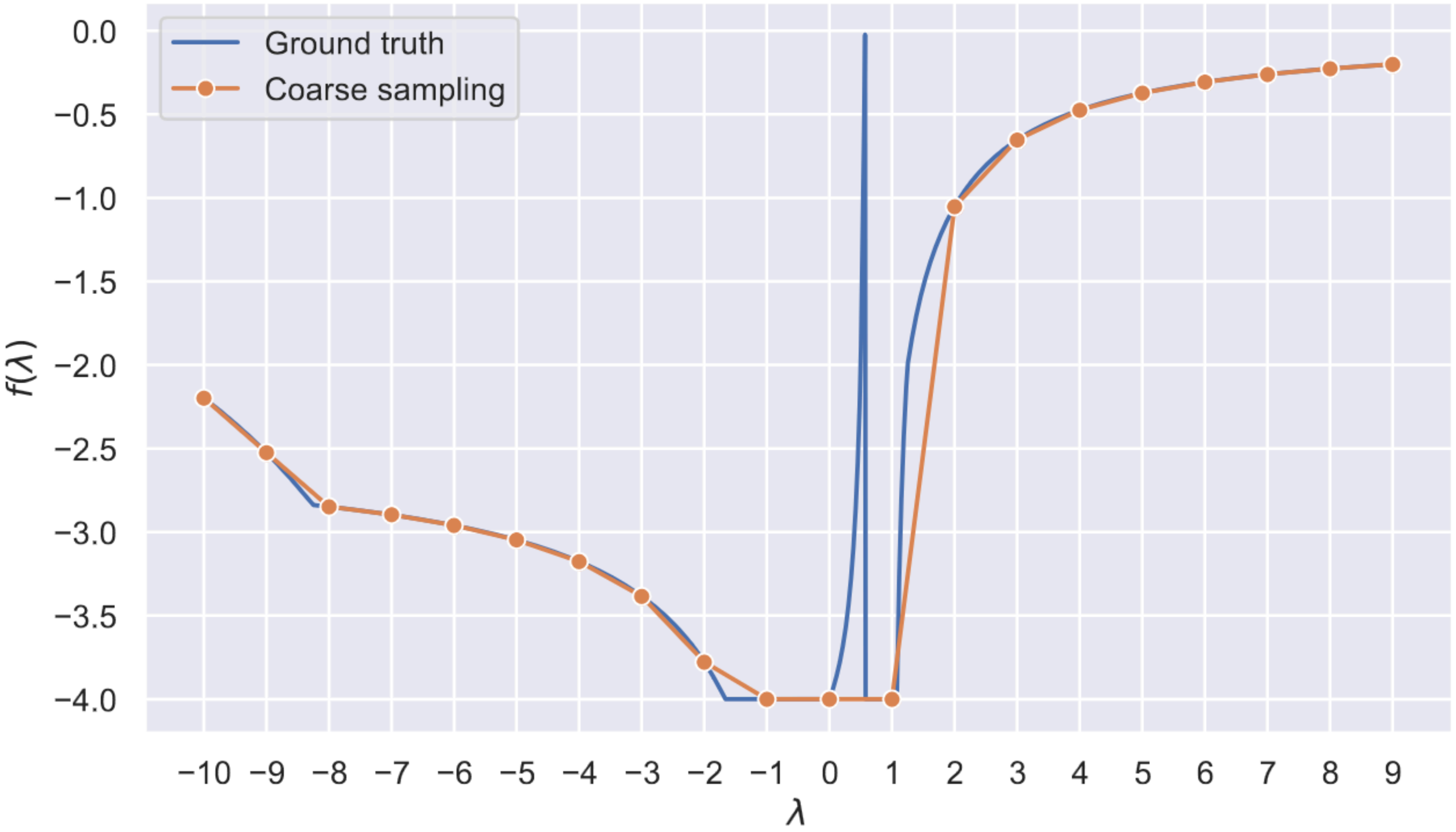
- We focus on this standard form:

$$
\begin{aligned}
f(\lambda) = \min \quad & c^t x \\
s.t. \quad & (A + \lambda D)x = b \\
& x \geq 0
\end{aligned}
$$

- In literature :
  - Usually rely on heavy computations,
  - approximations
  - and/or hypothesis on the matrix D

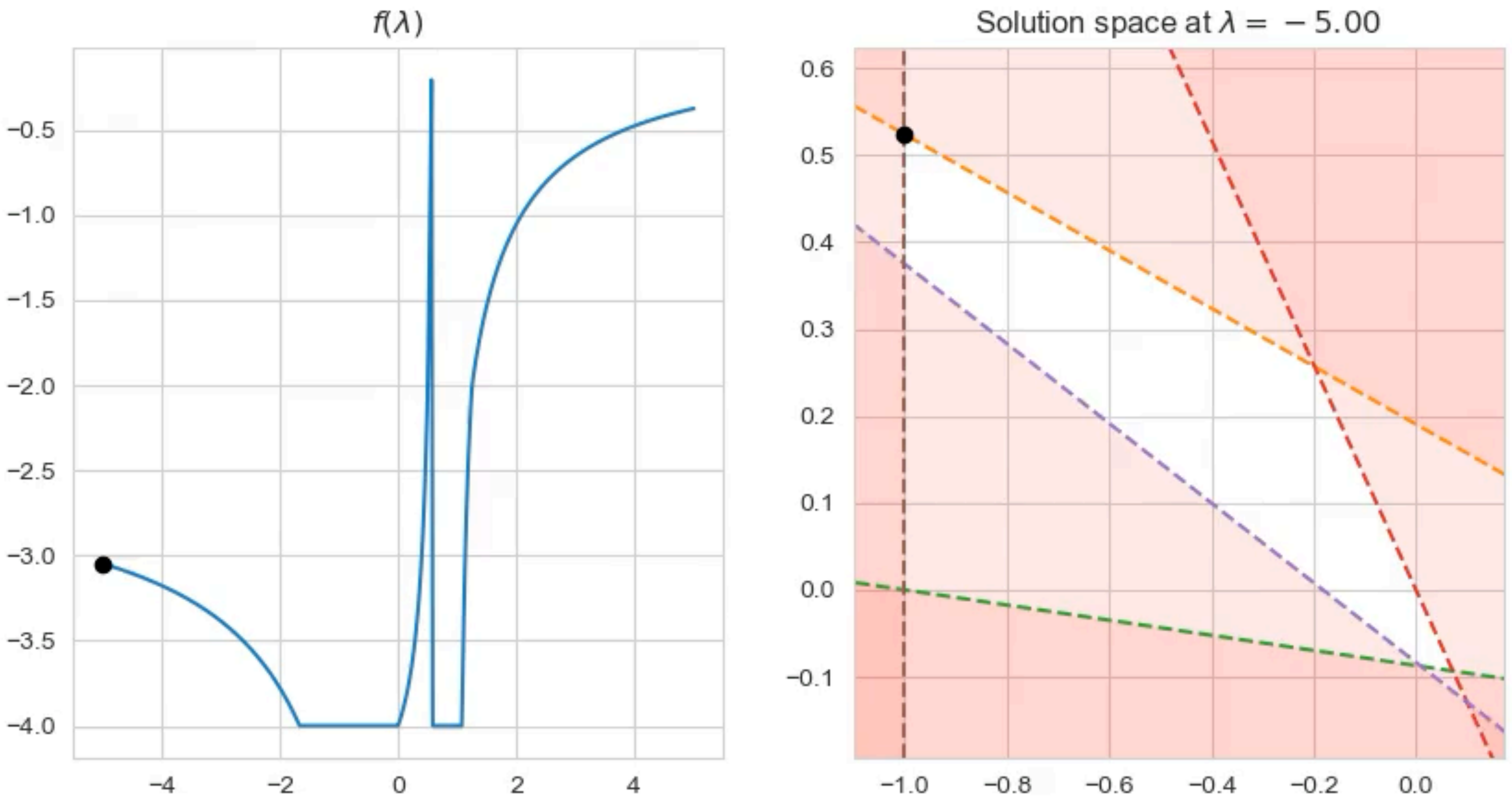
Naive solution

Heavy computations



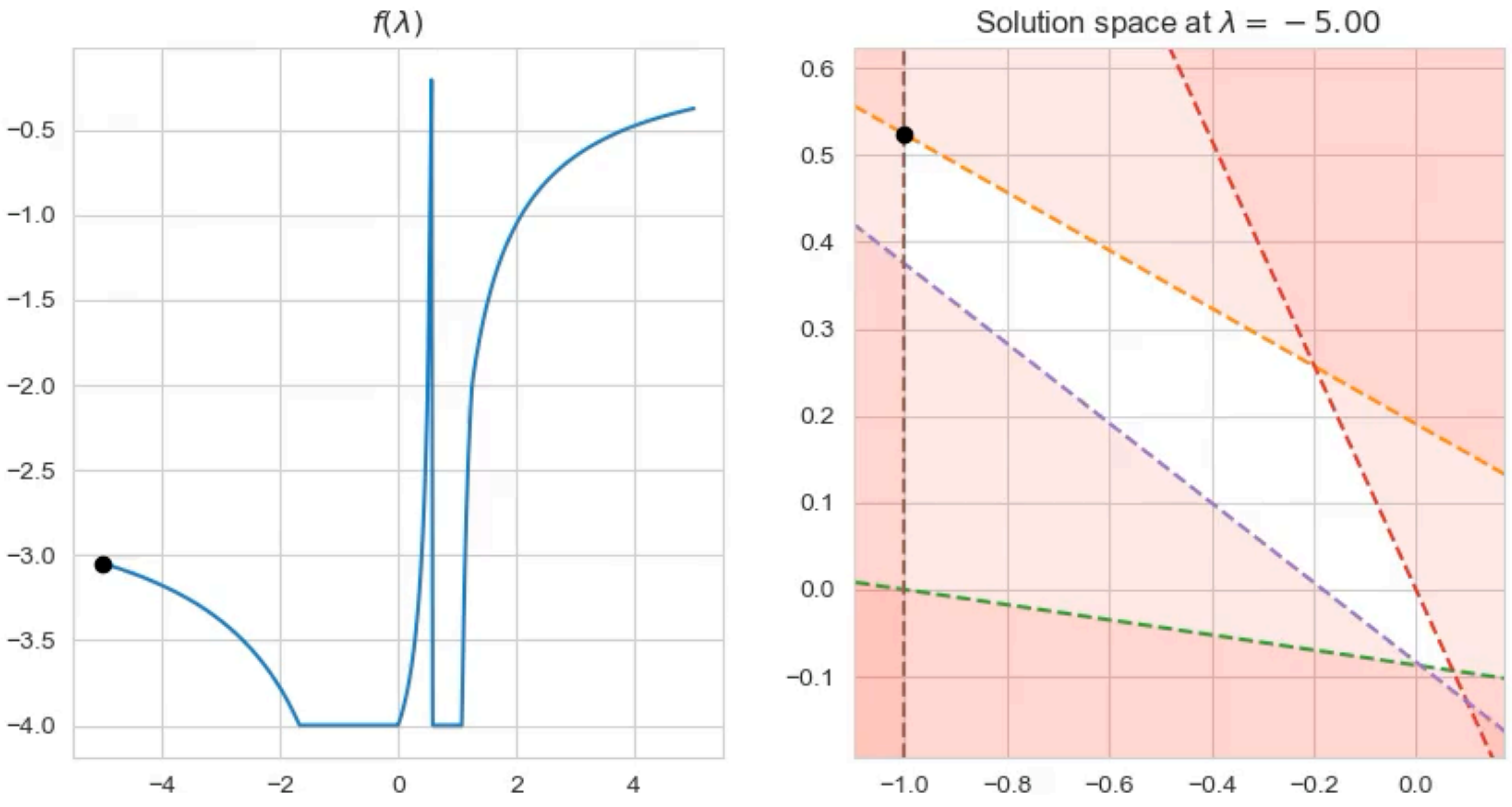
An observation

Piecewise component <=> optimal basis/face



An observation

Piecewise component <=> optimal basis/face



Given a basis...

- Assume we solved the problem for $\lambda = 0$ (without loss of generality)
- We obtain a basic optimal solution $x*$, an optimal objective $f(0) = o*$
- And a basis B, such that $x_B^* \geq 0$, $x_N^* = 0$
- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

Given a basis...

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- Conditions for remaining optimal:

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- Conditions for remaining optimal:
  - $A_B + \lambda D_B$ must be invertible/full-rank

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- Conditions for remaining optimal:
  - $A_B + \lambda D_B$ must be invertible/full-rank
  - $(A_B + \lambda D_B)^{-1} b \geq 0$

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- Conditions for remaining optimal:
  - $A_B + \lambda D_B$ must be invertible/full-rank
  - $(A_B + \lambda D_B)^{-1} b \geq 0$
  - Reduced costs ≥ 0

Given a basis...

- For any λ such that the basis remains optimal, we have:

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$

Everything depends on this matrix inversion!

- Conditions for remaining optimal:
  - $A_B + \lambda D_B$ must be invertible/full-rank
  - $(A_B + \lambda D_B)^{-1}b \geq 0$
  - Reduced costs ≥ 0

[Zuidwijk] to the rescue

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1\right)$$

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1\right)$$

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1\right)$$

- But (given n constraints):

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1\right)$$

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1 \right)$$

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1 \right)$$

- But (given n constraints):
    - Two eigenvalue problems to compute on an $n \times n$ matrix (ok?)

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1} b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1\right)$$

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1\right)$$

- But (given n constraints):
  - Two eigenvalue problems to compute on an $n \times n$ matrix (ok?)
  - Two n-degree polynomials to compute (numerical stability?)

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1} b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1\right)$$

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1\right)$$

- But (given n constraints):
    - Two eigenvalue problems to compute on an $n \times n$ matrix (ok?)
    - Two n-degree polynomials to compute (numerical stability?)
    - To assess if the basis is still optimal: $4n$ eigenvalue problems to compute on an $n \times n$ matrix!

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

[Zuidwijk] to the rescue

- In a research report lost on the internet, Zuidwijk tells us that there is a closed form solution!

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b = \frac{1}{\lambda}\left(\frac{\det(I + \lambda(A^{-1}D + bc^t))}{\det(I + \lambda A^{-1}D)} - 1\right)$$

$$= \frac{1}{\lambda}\left(\prod_i \frac{1 + \lambda\alpha_i}{1 + \lambda\beta_i} - 1\right)$$

- But (given n constraints):
  - Two eigenvalue problems to compute on an $n \times n$ matrix (ok?)
  - Two n-degree polynomials to compute (numerical stability?)
  - To assess if the basis is still optimal: $4n$ eigenvalue problems to compute on an $n \times n$ matrix!
    - (But gives the exact range of λ where the basis is optimal in exchange)

[Zuidwijk] R.A. Zuidwijk (2005) Linear Parametric Sensitivity Analysis of the Constraint Coefficient Matrix in Linear Programs. ERIM report series research in management

Today

- Another way of computing $f(\lambda)$
- Another way of computing the range of λs where the basis remains optimal
  - Less expensive...
  - ... but only gives a subset of the real range.

Simpler matrix inversion operations

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$

- We have A^{-1} from the solver:

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$
$$= c_B^t(A_B(I + \lambda A_B^{-1}D_B))^{-1}b \qquad \text{(Factorize } A_B\text{)}$$

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$

- We have A^{-1} from the solver:

$$f(\lambda) = c_B^t(A_B + \lambda D_B)^{-1}b$$

$$= c_B^t(A_B(I + \lambda A_B^{-1}D_B))^{-1}b \qquad \text{(Factorize } A_B\text{)}$$

$$= c_B^t(I + \lambda A_B^{-1}D_B)^{-1}A_B^{-1}b \qquad (XY)^{-1} = Y^{-1}X^{-1}$$

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- We have A^{-1} from the solver:

$$\begin{aligned} f(\lambda) &= c_B^t (A_B + \lambda D_B)^{-1} b \\ &= c_B^t (A_B (I + \lambda A_B^{-1} D_B))^{-1} b \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} A_B^{-1} b \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} x_B^* \end{aligned}$$

(Factorize A_B)

$(XY)^{-1} = Y^{-1} X^{-1}$

$A_B^{-1} b$ is the $\lambda = 0$ solution

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- We have A^{-1} from the solver:

$$\begin{aligned} f(\lambda) &= c_B^t (A_B + \lambda D_B)^{-1} b \\ &= c_B^t (A_B (I + \lambda A_B^{-1} D_B))^{-1} b && \text{(Factorize } A_B\text{)} \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} A_B^{-1} b && (XY)^{-1} = Y^{-1} X^{-1} \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} x_B^* && A_B^{-1} b \text{ is the } \lambda = 0 \text{ solution} \end{aligned}$$

- Let us write $E_B = A_B^{-1} D_B$

The idea: simpler inverse operation

- Let us assume we are on a basis B.

$$f(\lambda) = c_B^t (A_B + \lambda D_B)^{-1} b$$

- We have A^{-1} from the solver:

$$\begin{aligned} f(\lambda) &= c_B^t (A_B + \lambda D_B)^{-1} b \\ &= c_B^t (A_B (I + \lambda A_B^{-1} D_B))^{-1} b && \text{(Factorize } A_B\text{)} \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} A_B^{-1} b && (XY)^{-1} = Y^{-1} X^{-1} \\ &= c_B^t (I + \lambda A_B^{-1} D_B)^{-1} x_B^* && A_B^{-1} b \text{ is the } \lambda = 0 \text{ solution} \end{aligned}$$

- Let us write $E_B = A_B^{-1} D_B$

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

The idea: simpler inverse operation

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!

The idea: simpler inverse operation

$$f(\lambda) = c_B^t(I + \lambda E_B)^{-1}x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!
  - $E_B = Q\Lambda Q^{-1} \implies f(\lambda) = c_B^t(I + \lambda Q\Lambda Q^{-1})^{-1}x_B^*$

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
    - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!
    - $E_B = Q\Lambda Q^{-1} \implies f(\lambda) = c_B^t (I + \lambda Q\Lambda Q^{-1})^{-1} x_B^*$
      $= c_B^t (QQ^{-1} + \lambda Q\Lambda Q^{-1})^{-1} x_B^*$

The idea: simpler inverse operation

$$f(\lambda) = c_B^t(I + \lambda E_B)^{-1}x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!
  - $E_B = Q\Lambda Q^{-1} \implies f(\lambda) = c_B^t(I + \lambda Q\Lambda Q^{-1})^{-1}x_B^*$
    $= c_B^t(QQ^{-1} + \lambda Q\Lambda Q^{-1})^{-1}x_B^*$
    $= c_B^t Q(I + \lambda\Lambda)^{-1}Q^{-1}x_B^*$

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!
  - $E_B = Q\Lambda Q^{-1} \implies f(\lambda) = c_B^t (I + \lambda Q\Lambda Q^{-1})^{-1} x_B^*$
    $= c_B^t (QQ^{-1} + \lambda Q\Lambda Q^{-1})^{-1} x_B^*$
    $= c_B^t Q (I + \lambda \Lambda)^{-1} Q^{-1} x_B^*$
  - $I + \lambda\Lambda$ is diagonal: inversion trivial in $\mathcal{O}(n)$

The idea: simpler inverse operation

$$f(\lambda) = c_B^t (I + \lambda E_B)^{-1} x_B^*$$

- Still a non-trivial inversion/system solving to do for each λ
  - $\mathcal{O}(n^3)$ in practice... per λ
- Idea: decompose E_B to form a simpler to compute system
- First idea: eigendecomposition / diagonalization!
  - $E_B = Q \Lambda Q^{-1} \implies f(\lambda) = c_B^t (I + \lambda Q \Lambda Q^{-1})^{-1} x_B^*$

    $$\begin{aligned} &= c_B^t (QQ^{-1} + \lambda Q \Lambda Q^{-1})^{-1} x_B^* \\ &= c_B^t Q (I + \lambda \Lambda)^{-1} Q^{-1} x_B^* \end{aligned}$$

  - $I + \lambda \Lambda$ is diagonal: inversion trivial in $\mathcal{O}(n)$
  - Problem: most of the time, E_B is not diagonalizable

The idea: simpler inverse operation

The idea: simpler inverse operation

- Second idea: Schur's decomposition
  - $E_B = QUQ^H \implies f(\lambda) = c_B^t Q(I + \lambda U)^{-1} Q^H x_B^*$
  - $I + \lambda U$ is upper-triangular: system solving trivial in $\mathcal{O}(n^2)$

The idea: simpler inverse operation

- Second idea: Schur's decomposition
  - $E_B = QUQ^H \implies f(\lambda) = c_B^t Q(I + \lambda U)^{-1} Q^H x_B^*$
  - $I + \lambda U$ is upper-triangular: system solving trivial in $\mathcal{O}(n^2)$
- Potential problem: computing the decomposition

The idea: simpler inverse operation

- Second idea: Schur's decomposition
  - $E_B = QUQ^H \implies f(\lambda) = c_B^t Q(I + \lambda U)^{-1} Q^H x_B^*$
  - $I + \lambda U$ is upper-triangular: system solving trivial in $\mathcal{O}(n^2)$
- Potential problem: computing the decomposition
- Better numerical stability?

Ranges of bound validity

Optimality range

- We need three properties for the basis to stay optimal:
  - $A_B + \lambda D_B$ must be invertible/full-rank/non-singular
  - $(A_B + \lambda D_B)^{-1} b \geq 0$
  - Reduced costs ≥ 0

$A_B + \lambda D_B$ must be invertible/full-rank

- We saw that $(A_B + \lambda D_B) = A_B(I + \lambda E_B)$ with A_B already invertible
- Just need to check the eigenvalues α_i of E_B and ensure that $1 + \lambda\alpha_i \neq 0$
- That creates holes in the set of admissible λs
- (That was the easy part)

$$x_B^\lambda = (A_B + \lambda D_B)^{-1} b = (I + \lambda E_B)^{-1} x_B^* \geq 0$$

- Now we have to deal with the inversion
- Schur cannot help us here
- The idea: removing the inversion using Neumann series.

$$\|X\|_\infty < 1 \implies (I - X)^{-1} = \sum_{i=0}^{\infty} X^i$$

- If we apply this on $X = -\lambda E_B$,

$$\|\lambda E_B\|_\infty < 1 \implies x_B^\lambda = \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*$$

Now we have an infinite sum...

$$\|\lambda E_B\|_\infty < 1 \implies x_B^\lambda = \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*$$

- Typically what you say here is "let's neglect the terms for $i \geq 3$"
- But that's an approximation for which you lose all guarantees...
- Instead we use (sub-multiplicative) matrix norms!
- Let's say we focus on the two first terms of component j:

$$\begin{aligned} 0 \leq (x_B^\lambda)_j &= e_j \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^* \\ &= e_j (I - \lambda E_B + \sum_{i=2}^{\infty} (-\lambda E_B)^i) x_B^* \end{aligned}$$

Now we have an infinite sum...

$$\|\lambda E_B\|_\infty < 1 \implies x_B^\lambda = \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*$$

- Typically what you say here is "let's neglect the terms for $i \geq 3$"
- But that's an approximation, you lose all guarantees...
- Instead we use (sub-multiplicative) matrix norms!
- Let's say we focus on the two first terms of component j:

$$\begin{aligned} 0 \leq (x_B^\lambda)_j &= e_j \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^* \\ &= e_j (I - \lambda E_B + \sum_{i=2}^{\infty} (-\lambda E_B)^i) x_B^* \\ &= e_j x_B^* - \lambda e_j E_B x_B^* + \lambda^2 e_j E_B^2 \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^* \end{aligned}$$

Now we have an infinite sum...

$$0 \leq (x_B^\lambda)_j = e_j x_B^* - \lambda e_j E_B x_B^* + \lambda^2 e_j E_B^2 \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*$$

- The trick: take the absolute norm of the last part

$$(x_B^\lambda)_j \geq e_j x_B^* - \lambda e_j E_B x_B^* - \|\lambda^2 e_j E_B^2 \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*\|$$

- Now use sub-multiplicativity, triangular inequality, and geometric series closed-form on the last term:

$$\begin{aligned}\|\lambda^2 e_j E_B^2 \sum_{i=0}^{\infty} (-\lambda E_B)^i x_B^*\| &\leq \|\lambda^2 e_j E_B^2\| \cdot \sum_{i=0}^{\infty} \|(-\lambda E_B)^i\| \cdot \|x_B^*\| \\ &\leq \|\lambda^2 e_j E_B^2\| \cdot \frac{1}{1 - \|\lambda E_B\|} \cdot \|x_B^*\|\end{aligned}$$

Success

$$e_j x_B^* - \lambda e_j E_B x_B^* - \lambda^2 \frac{\|e_j E_B^2\| \cdot \|x_B^*\|}{1 - |\lambda| \cdot \|E_B\|} \geq 0 \implies (x_B^\lambda)_j \geq 0$$

(if $\|\lambda E_B\|_\infty, \|\lambda E_B\| < 1$)

- We thus have n polynomials of degree two for which we need the roots!
- Here we focused on the first 2 terms, giving degree 2 polynoms
  - if you keep the first v terms you get degree v polynoms.
- Far easier than n polynomials of degree n :-)
- But gives a subset of the true range of validity...
- The same tricks can be used to find ranges of optimality (using reduced costs)

What now?

What now?

- We have two "easy" way of computing $f(\lambda)$ given a known-optimal basis B
  - Zuidwijk's eigenvalues-based method
  - Schur decomposition
- We have now a scalable method to get the range of optimality of the basis B
  - At least a subset of it
- We lack a way to recompute the new matrix $E_B^\lambda = (A_B + \lambda D_B)^{-1} D_B$ for a λ a bit further than 0, to get a new range...
- For this, you can use one of the many $\mathcal{O}(n^\omega)$ techniques

Summary

	Eigenvalues	Schur+norms
Initial computation for each new basis	Compute eigenvalues	Compute Schur's decomposition
	$O(n^3)$	$O(n^3)$
For a new λ	Solve the polynomials	Solve the triangular system
	$O(n)$	$O(n^2)$
Validity/optimality range computation	Compute multiple eigenvalue problems	Compute norms and solve n degree-v polynomials
	$O(n^4)$	$O(vn^3+nv^2)$
(For the whole range)	Same: $O(n^4)$	An unknown number of time p, we need to recompute the basis matrix and the range
		$O(pvn^3+pnv^2)$

- Worth if p small.
- Schur decomposition can be updated in $O(n^3)$

Summary

	Eigenvalues	Schur+norms
Initial computation for each new basis	Compute eigenvalues	Compute Schur's decomposition
	$O(n^3)$	$O(n^3)$
For a new λ	Solve the polynomials	Solve the triangular system
	$O(n)$	$O(n^2)$
Validity/optimality range computation	Compute multiple eigenvalue problems	Compute norms and solve n degree-v polynomials
	$O(n^4)$	$O(vn^3+nv^2)$
(For the whole range)	Same: $O(n^4)$	An unknown number of time p, we need to recompute the basis matrix and the range
		$O(pvn^3+pnv^2)$

- Worth it if p small.
- Schur decomposition can be updated in $O(n^3)$
- You can mix the methods

Summary

	Eigenvalues	Schur+norms
Initial computation for each new basis	Compute eigenvalues	Compute Schur's decomposition
	$O(n^3)$	$O(n^3)$
For a new λ	Solve the polynomials	Solve the triangular system
	$O(n)$	$O(n^2)$
Validity/optimality range computation	Compute multiple eigenvalue problems	Compute norms and solve n degree-v polynomials
	$O(n^4)$	$O(vn^3+nv^2)$
(For the whole range)	Same: $O(n^4)$	An unknown number of time p, we need to recompute the basis matrix and the range
		$O(pvn^3+pnv^2)$

- Worth it if p small.
- Schur decomposition can be updated in $O(n^3)$
- You can mix the methods
- Paper not yet available

Summary

	Eigenvalues	Schur+norms
Initial computation for each new basis	Compute eigenvalues	Compute Schur's decomposition
	$O(n^3)$	$O(n^3)$
For a new λ	Solve the polynomials	Solve the triangular system
	$O(n)$	$O(n^2)$
Validity/optimality range computation	Compute multiple eigenvalue problems	Compute norms and solve n degree-v polynomials
	$O(n^4)$	$O(vn^3+nv^2)$
(For the whole range)	Same: $O(n^4)$	An unknown number of time p, we need to recompute the basis matrix and the range
		$O(pvn^3+pnv^2)$

- Worth it if p small.
- Schur decomposition can be updated in $O(n^3)$
- You can mix the methods
- Paper not yet available
- Other talk on this problem tomorrow!

Open questions

- How to characterize p?
- What is the physical meaning of $E_B = A_B^{-1} D_B$?
- What is the physical meaning of the closed-form of $f(\lambda)$?
- How well does all of this work in practice?