Multistage stochastic optimization and polyhedral geometry

PhD Defense   Maël Forcier

advised by Stéphane Gaubert and Vincent Leclère,

supervised by Jean-Philippe Chancelier.

December 14th 2022

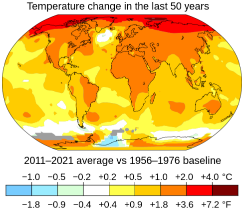
ÉCOLE POLYTECHNIQUE

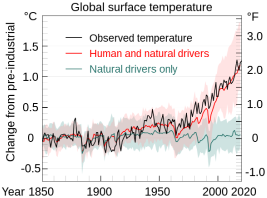
IP PARIS

École des Ponts ParisTech

Motivating example: hydroelectric energy management

- Need low-carbon energy to stop global warming
- Hydroelectricity is a controllable renewable energy
- $82,6\%$ of electricity is hydroelectric in Brazil, $17,1\%$ in France and 92% in Norway





Motivating example: hydroelectric energy management

- u water hustled
- d demand
- c cost of unmet demand
- x_0/x_1 water in the reservoir
- $\overline{x}$ capacity of the reservoir
- w rain and runoff

$$
\begin{aligned}
\min_{u, x_1} \; & c(d-u) \\
s.t. \; & 0 \leqslant u \leqslant d \\
& x_1 \leqslant x_0 - u + w \\
& 0 \leqslant x_1 \leqslant \overline{x} \\
& x_0 \text{ fixed}
\end{aligned}
$$

Motivating example: hydroelectric energy management

At step t

- u_t water hustled
- d_t demand
- c_t cost of unmet demand
- x_t water in the reservoir
- $\overline{x}$ capacity of the reservoir
- w_t rain and runoff

$$
\begin{aligned}
\min_{u_t, x_t} \quad & \sum_{t=1}^{T} c_t(d_t - u_t) & \\
s.t. \quad & 0 \leqslant u_t \leqslant d_t & , \forall t \in [T] \\
& x_{t+1} \leqslant x_t - u_t + w_t & , \forall t \in [T] \\
& 0 \leqslant x_t \leqslant \overline{x} & , \forall t \in [T] \\
& x_0 \text{ fixed} &
\end{aligned}
$$

Motivating example: hydroelectric energy management

At step t

- u_t water hustled
- d_t demand
- c_t cost of unmet demand
- x_t water in the reservoir
- $\overline{x}$ capacity of the reservoir
- w_t rain and runoff

$$\begin{aligned}
\min_{u_t, x_t} \quad & \sum_{t=1}^{T} c_t(d_t - u_t) & \\
s.t. \quad & 0 \leqslant u_t \leqslant d_t & , \forall t \in [T] \\
& x_{t+1} \leqslant x_t - u_t + w_t & , \forall t \in [T] \\
& 0 \leqslant x_t \leqslant \overline{x} & , \forall t \in [T] \\
& x_0 \text{ fixed} &
\end{aligned}$$

General form
$$\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & c^\top x \\
\text{s.t.} \quad & Ax \leqslant b
\end{aligned}$$

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ \\ \\ \\ \\ \\ \end{pmatrix}$$

$$x_1 + x_2 \leqslant 1 \qquad (1)$$

(2)

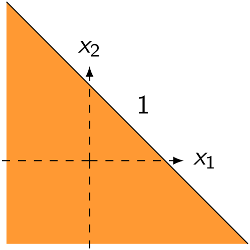
(3)

(4)

(5)

(6)

(7)



Linear Programming and polyhedra

$$\min_{x\in\mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

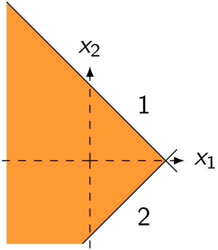
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ \\ \\ \\ \\ \\ \end{pmatrix}$$

$$x_1 + x_2 \leqslant 1 \qquad (1)$$
$$x_1 - x_2 \leqslant 1 \qquad (2)$$
$$(3)$$
$$(4)$$
$$(5)$$
$$(6)$$
$$(7)$$



Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

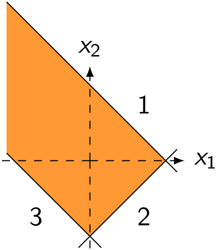
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \\ \\ \\ \\ \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 & (1) \\ x_1 - x_2 &\leqslant 1 & (2) \\ -x_1 - x_2 &\leqslant 1 & (3) \\ & & (4) \\ & & (5) \\ & & (6) \\ & & (7) \end{aligned}$$



Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

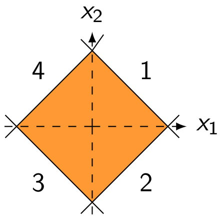
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ \\ \\ \\ \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \\ \\ \\ \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 && (1) \\ x_1 - x_2 &\leqslant 1 && (2) \\ -x_1 - x_2 &\leqslant 1 && (3) \\ -x_1 + x_2 &\leqslant 1 && (4) \\ & && (5) \\ & && (6) \\ & && (7) \end{aligned}$$



Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

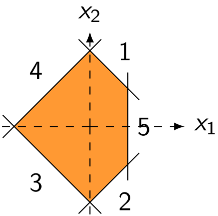
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ 1 & 0 \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ \\ \end{pmatrix}$$

$$\begin{array}{rl} x_1 + x_2 \leqslant 1 & (1) \\ x_1 - x_2 \leqslant 1 & (2) \\ -x_1 - x_2 \leqslant 1 & (3) \\ -x_1 + x_2 \leqslant 1 & (4) \\ x_1 \leqslant 0.5 & (5) \\ & (6) \\ & (7) \end{array}$$



Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

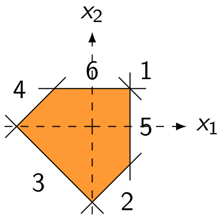
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 & (1)\\ x_1 - x_2 &\leqslant 1 & (2)\\ -x_1 - x_2 &\leqslant 1 & (3)\\ -x_1 + x_2 &\leqslant 1 & (4)\\ x_1 &\leqslant 0.5 & (5)\\ x_2 &\leqslant 0.5 & (6)\\ & & (7) \end{aligned}$$



Linear Programming and polyhedra

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leqslant b \end{aligned}$$

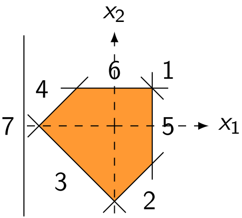
Definition

Polyhedron :
Intersection of finite number of halfspaces

The set $P = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b\}$ of admissible solutions is a polyhedron.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \\ -1.2 \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 && (1) \\ x_1 - x_2 &\leqslant 1 && (2) \\ -x_1 - x_2 &\leqslant 1 && (3) \\ -x_1 + x_2 &\leqslant 1 && (4) \\ x_1 &\leqslant 0.5 && (5) \\ x_2 &\leqslant 0.5 && (6) \\ x_1 &\geqslant -1.2 && (7) \end{aligned}$$



But renewables are inherently **stochastic** !

Rain, runoff, cost and demand are **random**.

At step t

- u_t water hustled
- d_t demand
- c_t cost of unmet demand
- x_t water in the reservoir
- $\overline{x}$ capacity of the reservoir
- w_t rain and runoff

$$
\begin{aligned}
\min_{u_t, x_t} \quad & \sum_{t=1}^{T} c_t(d_t - u_t) & \\
s.t.\ & 0 \leqslant u_t \leqslant d_t & , \forall t \in [T] \\
& x_{t+1} \leqslant x_t - u_t + w_t & , \forall t \in [T] \\
& 0 \leqslant x_t \leqslant \overline{x} & , \forall t \in [T] \\
& x_0 \text{ fixed} &
\end{aligned}
$$

But renewables are inherently **stochastic** !

Rain, runoff, cost and demand are **random**.

At step t

- $\boldsymbol{u}_t$ water hustled
- $\boldsymbol{d}_t$ demand
- $\boldsymbol{c}_t$ cost of unmet demand
- $\boldsymbol{x}_t$ water in the reservoir
- $\overline{x}$ capacity of the reservoir
- $\boldsymbol{w}_t$ rain and runoff

$$
\begin{aligned}
\min_{\boldsymbol{u}_t,\boldsymbol{x}_t} \ & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t(\boldsymbol{d}_t - \boldsymbol{u}_t)\Big] \\
s.t.\ & 0 \leqslant \boldsymbol{u}_t \leqslant \boldsymbol{d}_t && , \forall t \in [T] \\
& \boldsymbol{x}_{t+1} \leqslant \boldsymbol{x}_t - \boldsymbol{u}_t + \boldsymbol{w}_t && , \forall t \in [T] \\
& 0 \leqslant \boldsymbol{x}_t \leqslant \overline{x} && , \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} \\
& \underbrace{\begin{aligned}&\sigma(\boldsymbol{u}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{d}_\tau, \boldsymbol{w}_\tau)_{\tau \leqslant t} \\ &\sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{d}_\tau, \boldsymbol{w}_\tau)_{\tau \leqslant t}\end{aligned}}_{\text{Measurability constraints}} && \begin{aligned}&, \forall t \in [T] \\ &, \forall t \in [T]\end{aligned}
\end{aligned}
$$

Multistage stochastic linear programming (MSLP)

$$\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \xi_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \xi_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\right].$$

Multistage stochastic linear programming (MSLP)

$$\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \xi_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \xi_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \xi_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \xi_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\right].$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t\in[T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau\leqslant t} & \forall t\in[T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \xi_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \xi_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1:A_1x_1+B_1x_0\leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2:\boldsymbol{A}_2x_2+\boldsymbol{B}_2x_1\leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T:\boldsymbol{A}_Tx_T+\boldsymbol{B}_Tx_{T-1}\leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].$$

Multistage stochastic linear programming (MSLP)

$$\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$
x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow x_T
$$

Equivalent form

$$
\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].
$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \xi_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau\leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\Bigg].$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$
x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow x_T
$$

Equivalent form

$$
\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\right].
$$

Multistage stochastic linear programming (MSLP)

$$
\begin{aligned}
\min_{(\boldsymbol{x}_t)_{t\in[T]}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] & \\
\text{s.t.} \quad & \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{x}_{t-1} \leqslant \boldsymbol{b}_t & \forall t \in [T] \\
& \sigma(\boldsymbol{x}_t) \subset \sigma(\boldsymbol{c}_\tau, \boldsymbol{A}_\tau, \boldsymbol{B}_\tau, \boldsymbol{b}_\tau)_{\tau \leqslant t} & \forall t \in [T] \\
& \boldsymbol{x}_0 \equiv x_0 \text{ given} &
\end{aligned}
$$

$\boldsymbol{\xi}_t = (\boldsymbol{c}_t, \boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t)_{t\in[T]}$ is assumed to be stagewise independent.

At each time step: the present noise is revealed then we take a decision.

$$x_0 \rightsquigarrow \boldsymbol{\xi}_1 \rightsquigarrow x_1 \rightsquigarrow \boldsymbol{\xi}_2 \rightsquigarrow \cdots \rightsquigarrow x_{T-1} \rightsquigarrow \boldsymbol{\xi}_T \rightsquigarrow x_T$$

Equivalent form

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T\Big]\Big]\right].$$

Dynamic Programming (Bellman 1966)

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \mathbb{E}\big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T \big] \Big] \right]$$

$$\text{We set } V_{T+1} \equiv 0 \text{ and } V_t(x_{t-1}) := \mathbb{E}\begin{bmatrix} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{bmatrix}$$

Dynamic Programming (Bellman 1966)

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \mathbb{E}\Big[\cdots + \underbrace{\mathbb{E}\Big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T \Big]}_{V_T(x_{T-1})} \Big] \right]$$

$$\text{We set } V_{T+1} \equiv 0 \text{ and } V_t(x_{t-1}) := \mathbb{E}\begin{bmatrix} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{bmatrix}$$

Dynamic Programming (Bellman 1966)

$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \mathbb{E}\Bigg[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \underbrace{\mathbb{E}\Big[\cdots + \underbrace{\mathbb{E}\big[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T \big]}_{V_T(x_{T-1})} \Big]}_{V_3(x_2)} \Bigg]$$

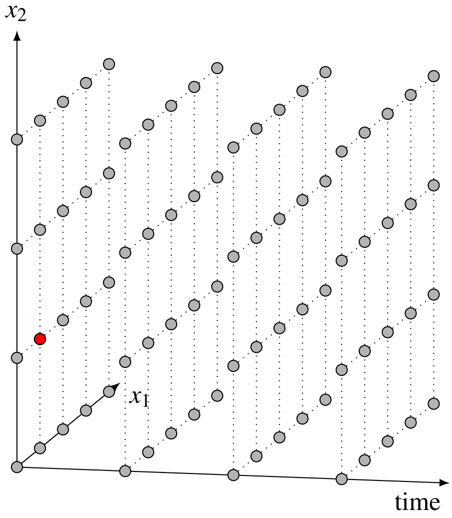
We set $V_{T+1} \equiv 0$ and $V_t(x_{t-1}) := \mathbb{E}\left[\begin{array}{ll} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{array}\right]$

Dynamic Programming (Bellman 1966)

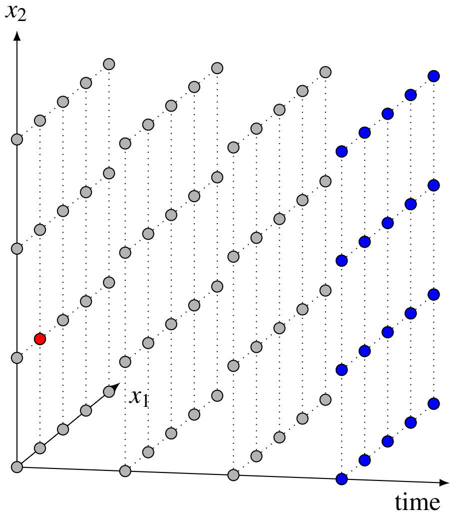
$$\min_{x_1 : A_1 x_1 + B_1 x_0 \leqslant b_1} c_1^\top x_1 + \underbrace{\mathbb{E}\left[\min_{x_2 : \boldsymbol{A}_2 x_2 + \boldsymbol{B}_2 x_1 \leqslant \boldsymbol{b}_2} \boldsymbol{c}_2^\top x_2 + \underbrace{\mathbb{E}\left[\cdots + \underbrace{\mathbb{E}\left[\min_{x_T : \boldsymbol{A}_T x_T + \boldsymbol{B}_T x_{T-1} \leqslant \boldsymbol{b}_T} \boldsymbol{c}_T^\top x_T \right]}_{V_T(x_{T-1})} \right]}_{V_3(x_2)} \right]}_{V_2(x_1)}$$

We set $V_{T+1} \equiv 0$ and $V_t(x_{t-1}) := \mathbb{E}\left[\begin{array}{ll} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{array} \right]$

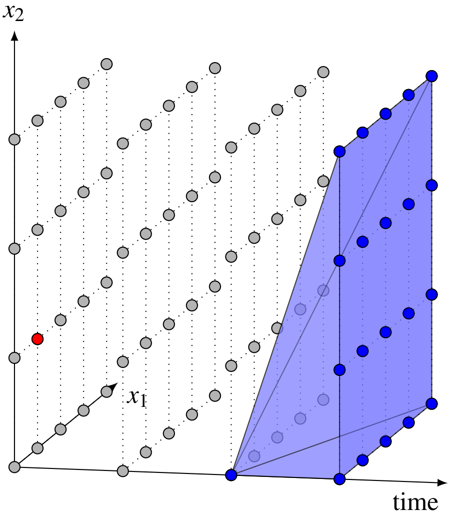
Dynamic programming : finite case



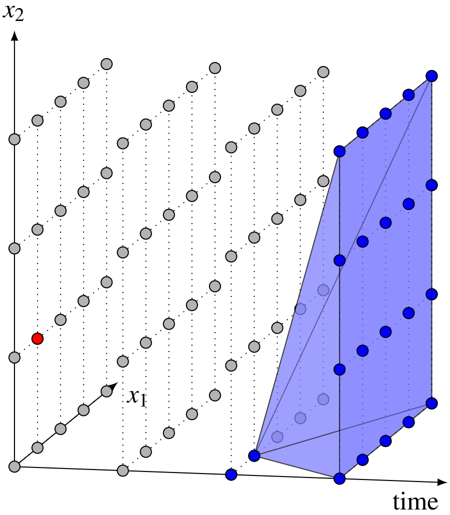
Dynamic programming : finite case



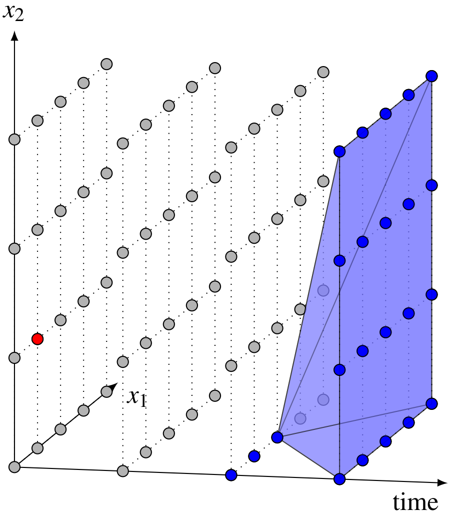
Dynamic programming : finite case



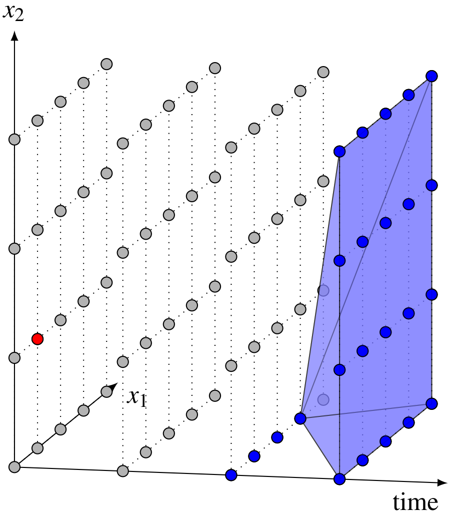
Dynamic programming : finite case



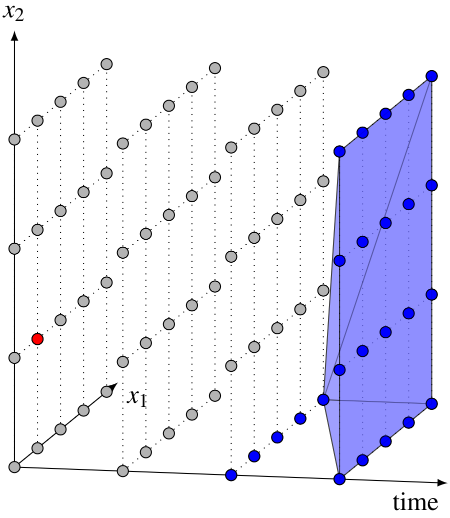
Dynamic programming : finite case



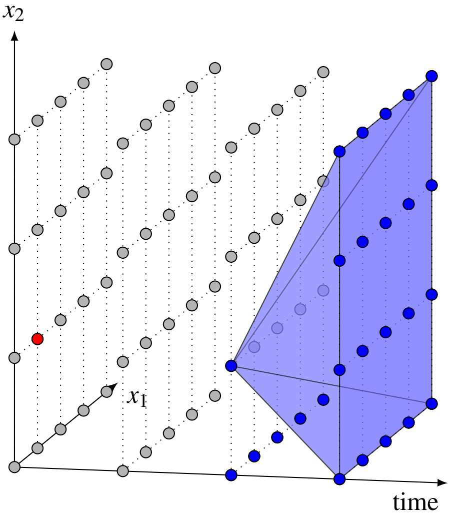
Dynamic programming : finite case



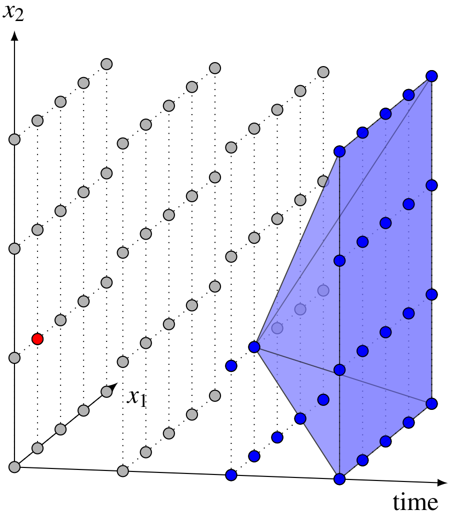
Dynamic programming : finite case



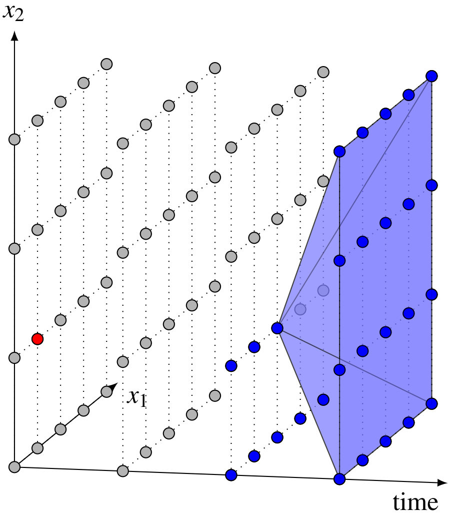
Dynamic programming : finite case



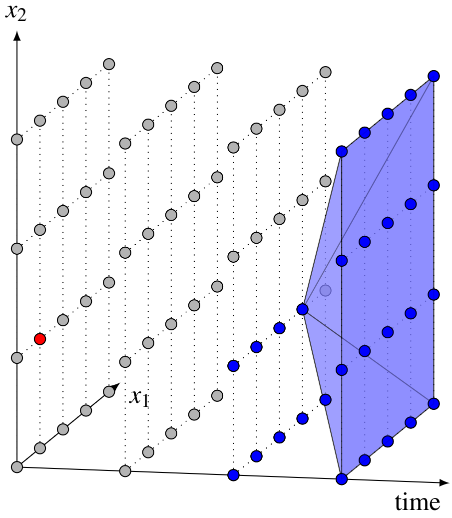
Dynamic programming : finite case



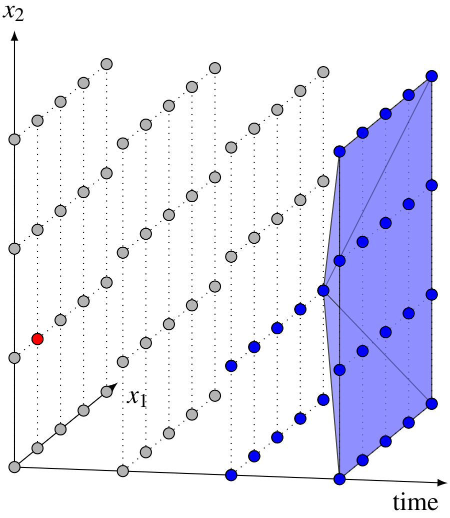
Dynamic programming : finite case



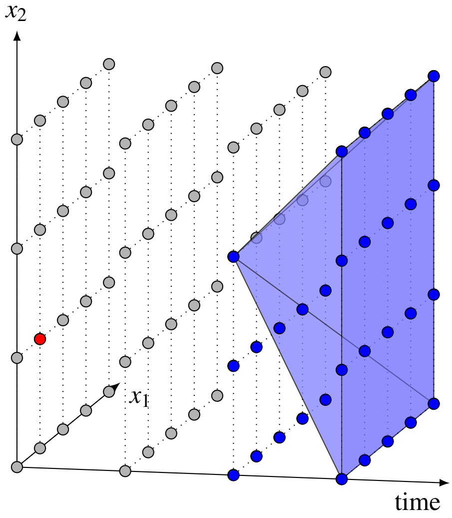
Dynamic programming : finite case



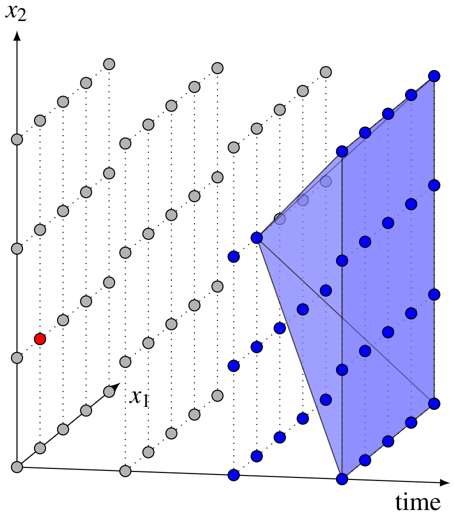
Dynamic programming : finite case



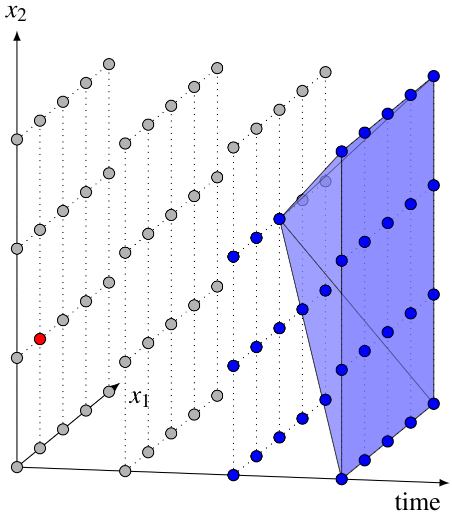
Dynamic programming : finite case



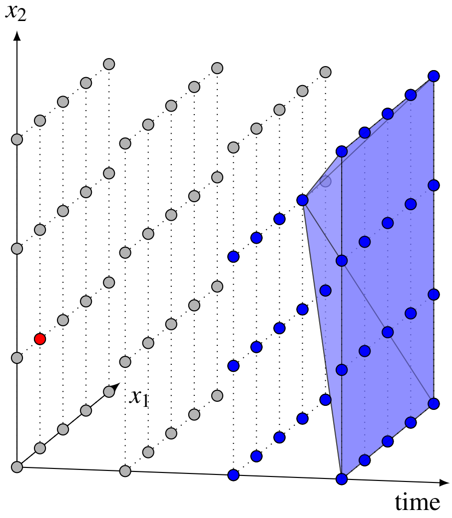
Dynamic programming : finite case



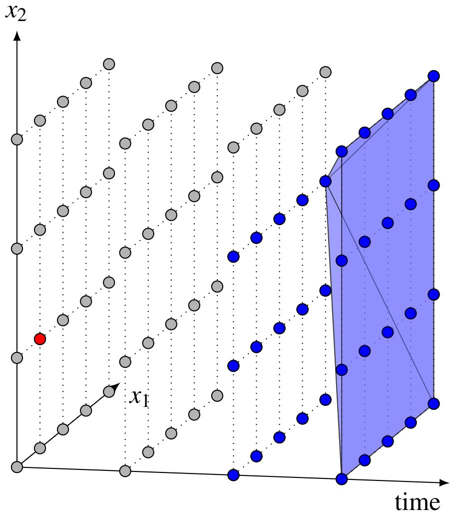
Dynamic programming : finite case



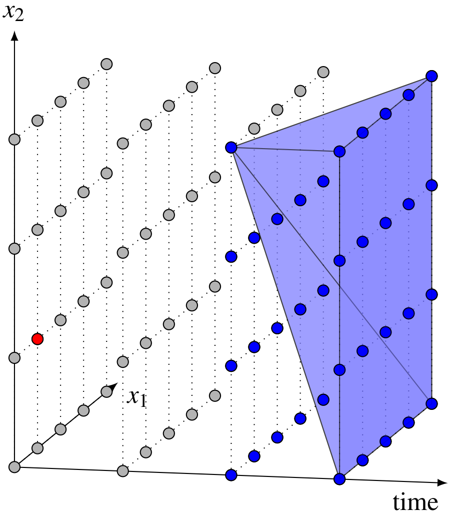
Dynamic programming : finite case



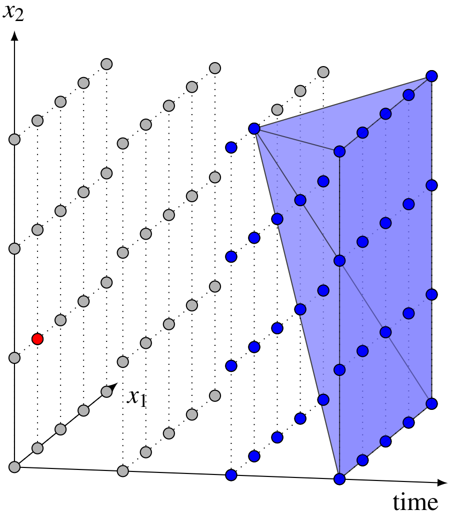
Dynamic programming : finite case



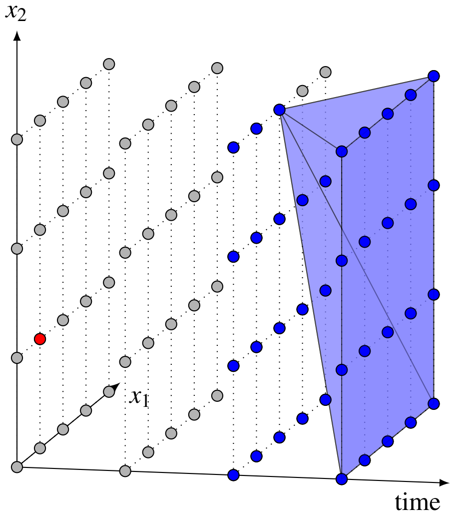
Dynamic programming : finite case



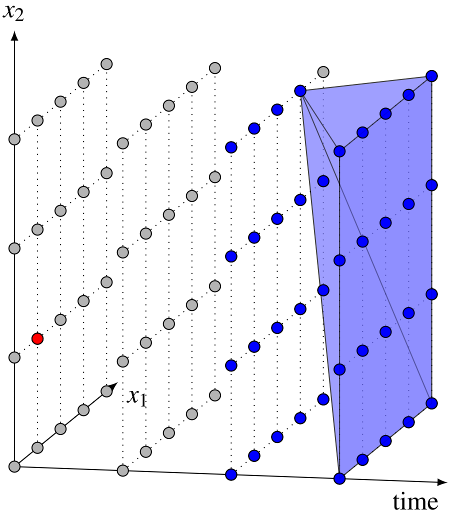
Dynamic programming : finite case



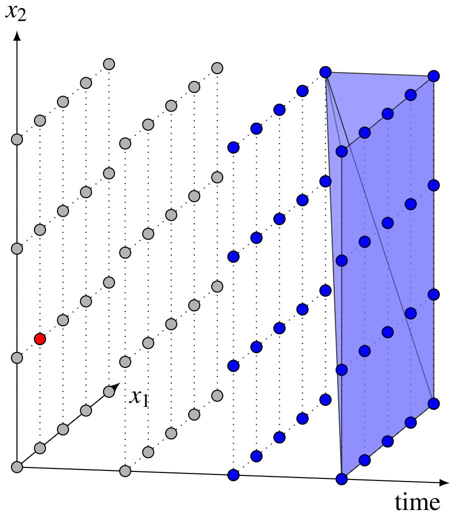
Dynamic programming : finite case



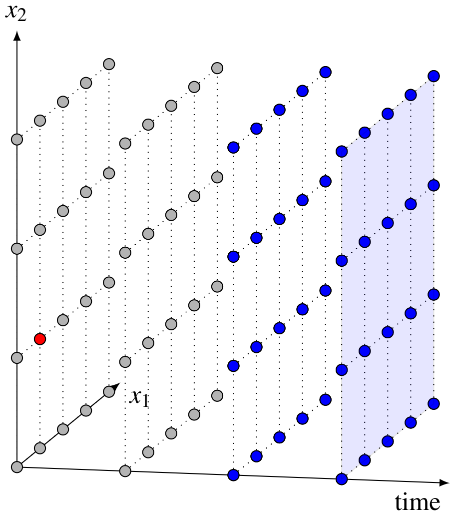
Dynamic programming : finite case



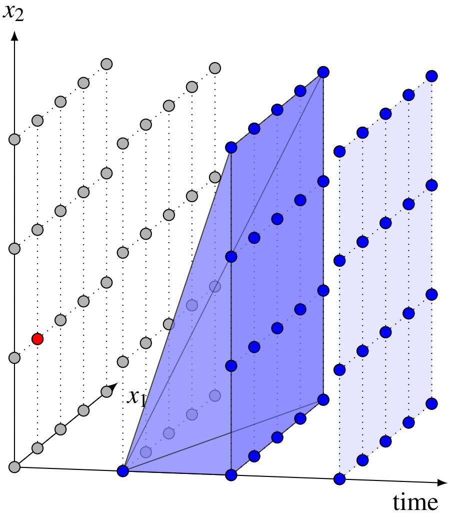
Dynamic programming : finite case



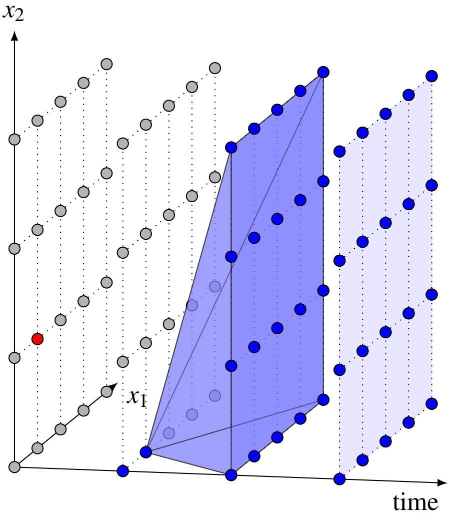
Dynamic programming : finite case



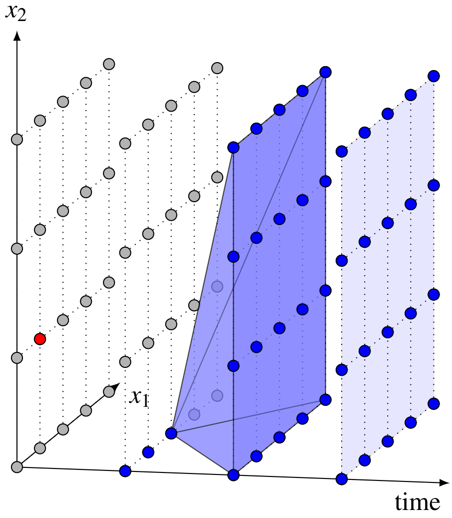
Dynamic programming : finite case



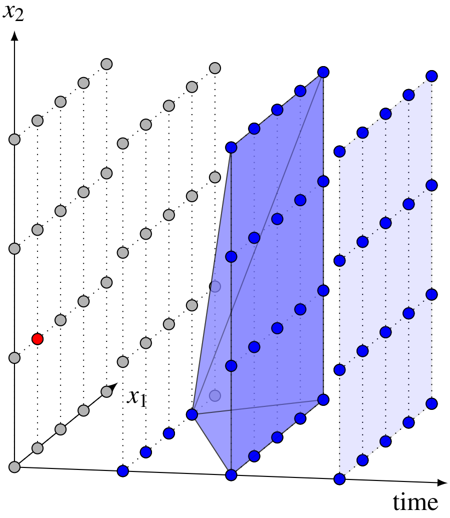
Dynamic programming : finite case



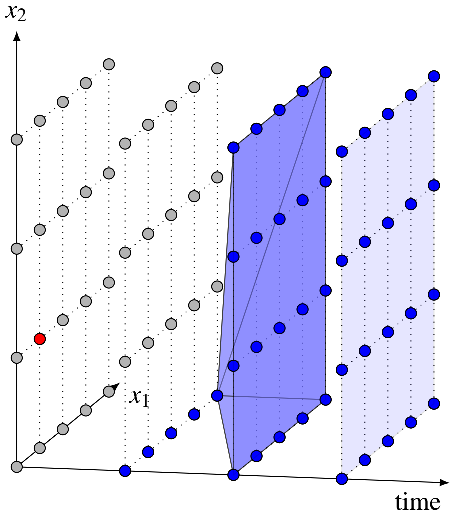
Dynamic programming : finite case



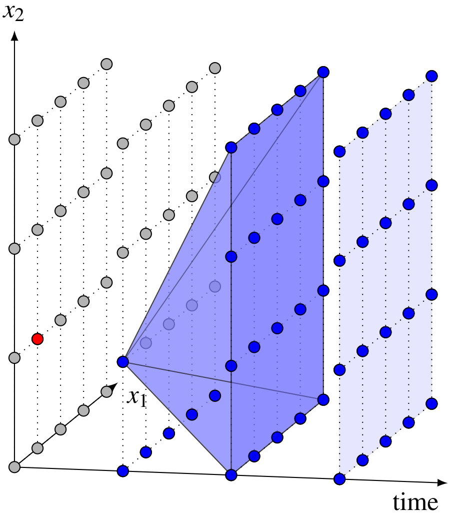
Dynamic programming : finite case



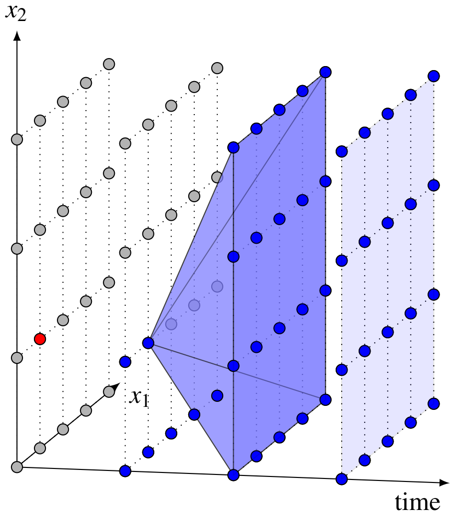
Dynamic programming : finite case



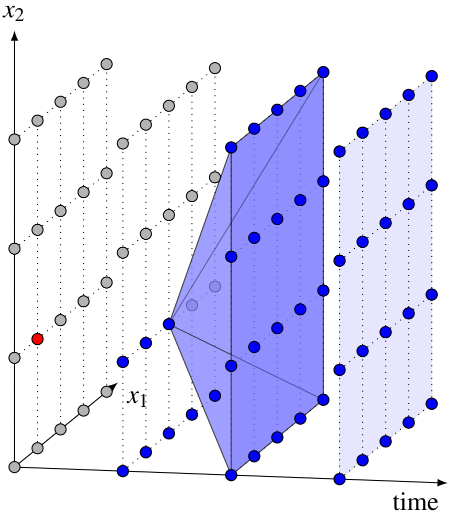
Dynamic programming : finite case



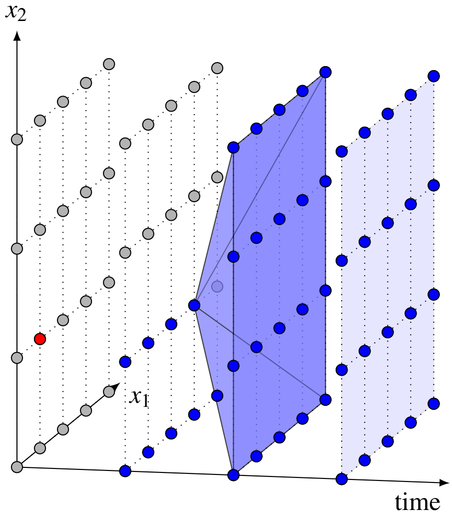
Dynamic programming : finite case



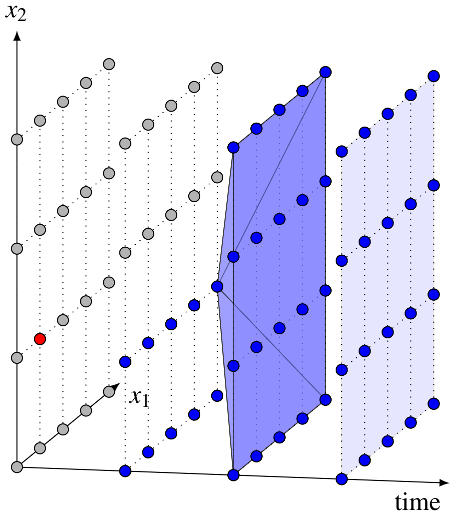
Dynamic programming : finite case



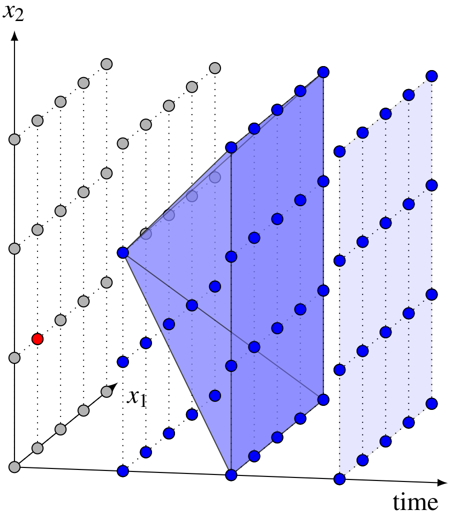
Dynamic programming : finite case



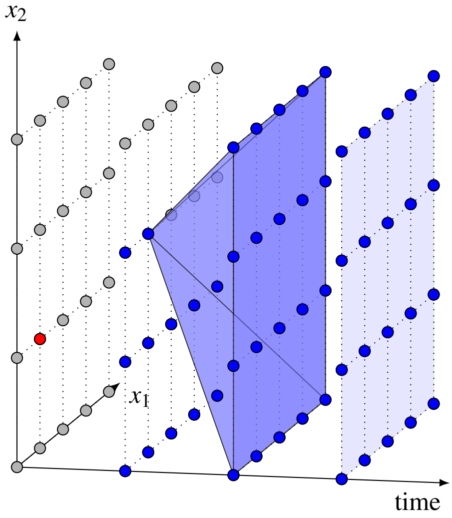
Dynamic programming : finite case



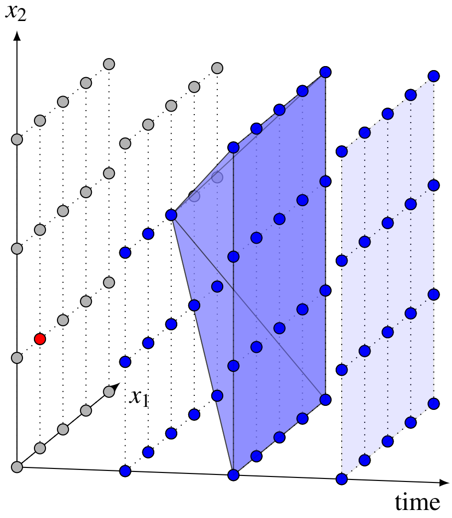
Dynamic programming : finite case



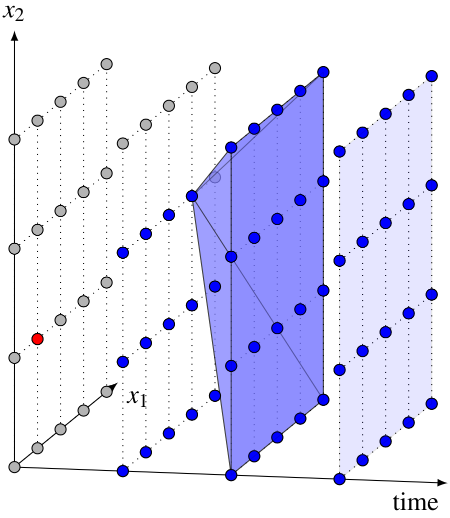
Dynamic programming : finite case



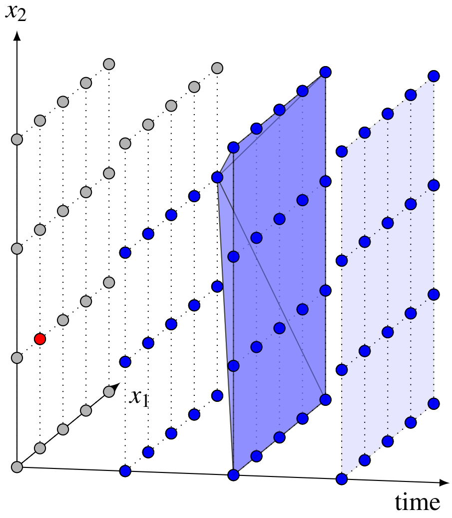
Dynamic programming : finite case



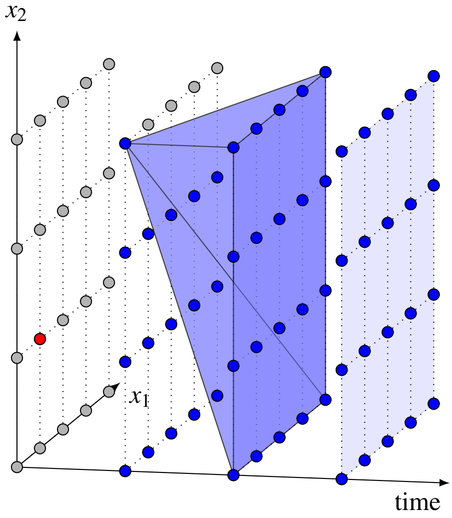
Dynamic programming : finite case



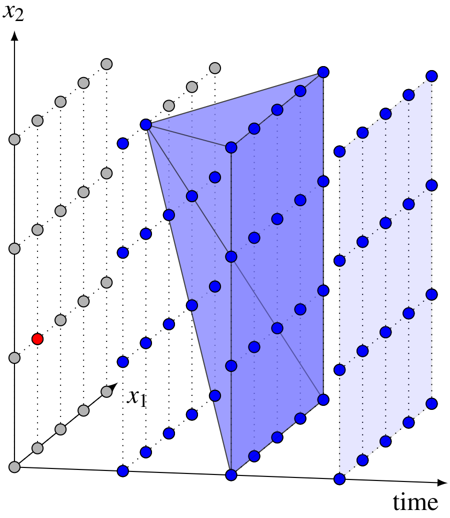
Dynamic programming : finite case



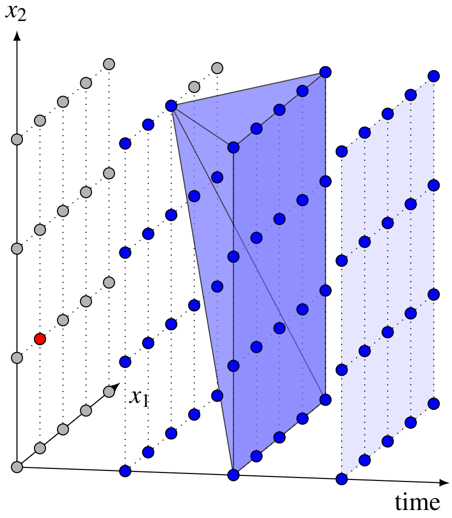
Dynamic programming : finite case



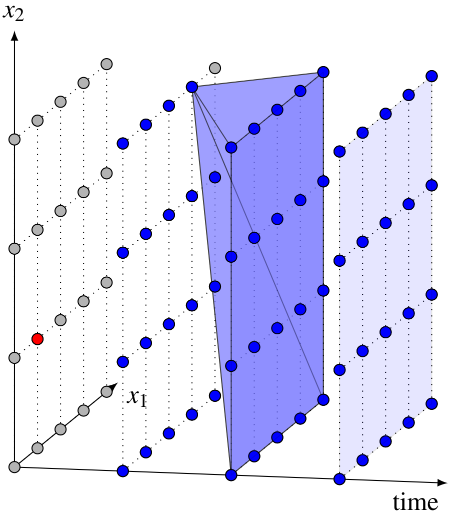
Dynamic programming : finite case



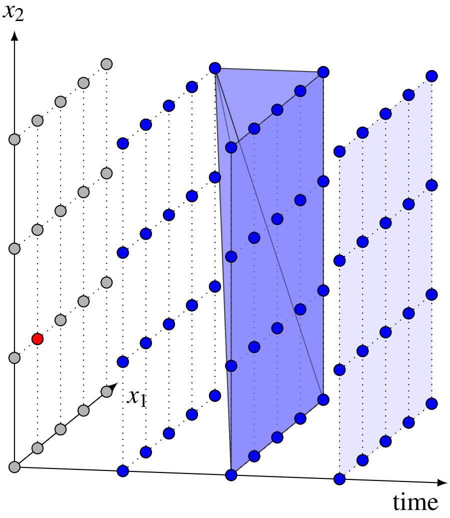
Dynamic programming : finite case



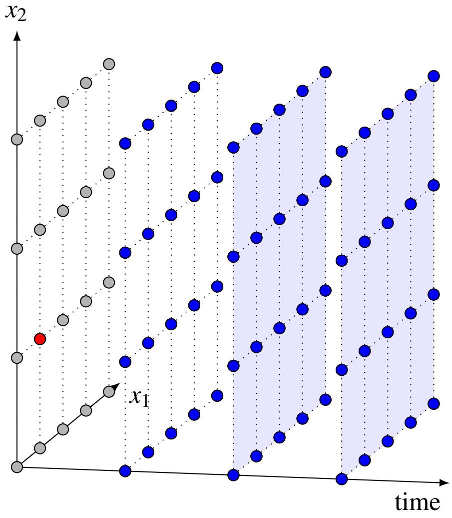
Dynamic programming : finite case



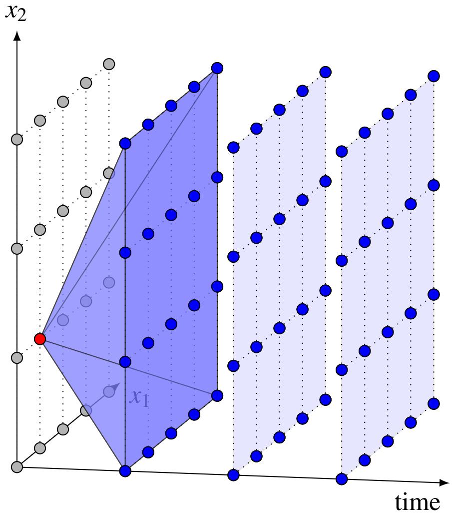
Dynamic programming : finite case



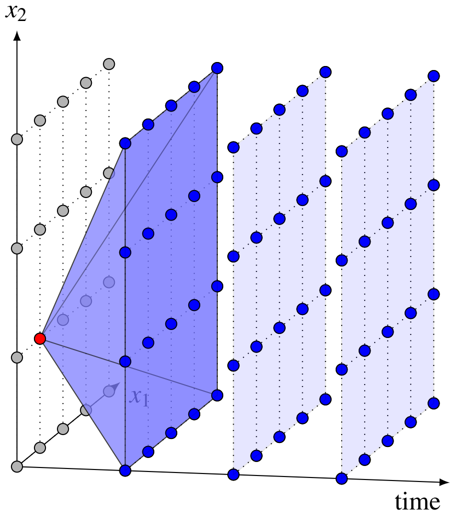
Dynamic programming : finite case



Dynamic programming : finite case

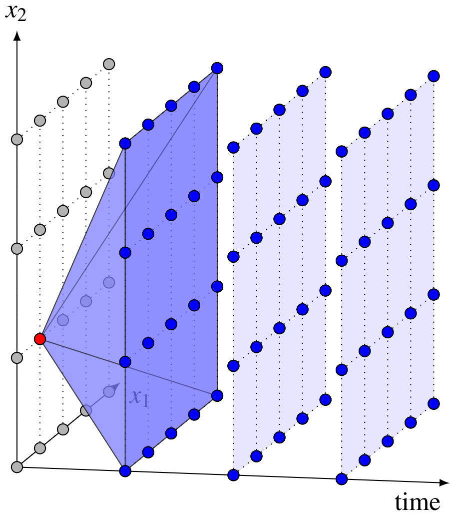


Dynamic programming : finite case



➥ Continuous space : algorithms such as SDDP later discussed.

Dynamic programming : finite case

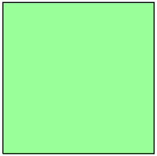


- ➥ Continuous space : algorithms such as SDDP later discussed.
- ➥ How to deal with continuous distributions ?

Quantization of a MSLP

Real problem

$$V_t(x) = \mathbb{E}\left[\hat{V}_t\left(x, \boldsymbol{\xi}_t\right)\right] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^{n_t}} & \boldsymbol{c_t}^\top y + V_{t+1}(y) \\ \text{s.t.} & \boldsymbol{A}_t y + \boldsymbol{B}_t x \leqslant \boldsymbol{b}_t \end{bmatrix}$$

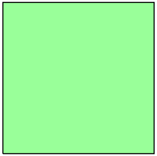


$\boldsymbol{\xi}_t$ continuous

Quantization of a MSLP

Real problem

$$V_t(x) = \mathbb{E}\big[\hat{V}_t\big(x, \boldsymbol{\xi}_t\big)\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^{n_t}} & \boldsymbol{c_t}^\top y + V_{t+1}(y) \\ \text{s.t.} & \boldsymbol{A}_t y + \boldsymbol{B}_t x \leqslant \boldsymbol{b}_t \end{bmatrix}$$

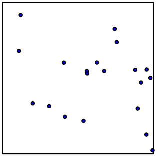


$\boldsymbol{\xi}_t$ continuous

Sample Average Approximation (SAA)

$$V^{SAA}_{t,N}(x) := \frac{1}{N}\sum_{k=1}^{N} \hat{V}_t\big(x, \xi^k\big)$$

$\xi^1, \cdots, \xi^N$ drawn by Monte Carlo (ex Shapiro 2011)

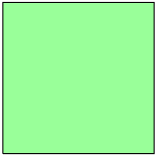


SAA $N = 20$

Quantization of a MSLP

Real problem

$$V_t(x) = \mathbb{E}\big[\hat{V}_t(x, \boldsymbol{\xi}_t)\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^{n_t}} & \boldsymbol{c_t}^\top y + V_{t+1}(y) \\ \text{s.t.} & \boldsymbol{A_t} y + \boldsymbol{B_t} x \leqslant \boldsymbol{b_t} \end{bmatrix}$$

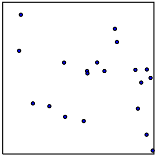


$\boldsymbol{\xi}_t$ continuous

Sample Average Approximation (SAA)

$$V_{t,N}^{SAA}(x) := \frac{1}{N} \sum_{k=1}^{N} \hat{V}_t(x, \xi^k)$$

$\xi^1, \cdots, \xi^N$ drawn by Monte Carlo (ex Shapiro 2011)

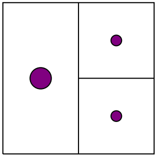


SAA $N = 20$

Partition-based

$$V_{t,\mathcal{P}}(x) := \sum_{P \in \mathcal{P}} \check{p}_{t,P} \hat{V}_t(x, \check{\xi}_{t,P})$$

with $\check{p}_{t,P} := \mathbb{P}\big[\boldsymbol{\xi}_t \in P\big]$ and $\check{\xi}_{t,P} := \mathbb{E}\big[\boldsymbol{\xi}_t \,|\, \boldsymbol{\xi}_t \in P\big]$

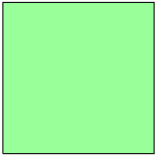


Partition-based

Quantization of a MSLP

Real problem

$$V_t(x) = \mathbb{E}\big[\hat{V}_t\big(x, \boldsymbol{\xi}_t\big)\big] = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y \in \mathbb{R}^{n_t}} & \boldsymbol{c_t}^\top y + V_{t+1}(y) \\ \text{s.t.} & \boldsymbol{A}_t y + \boldsymbol{B}_t x \leqslant \boldsymbol{b}_t \end{array}\right]$$

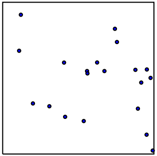


$\boldsymbol{\xi}_t$ continuous

Sample Average Approximation (SAA)

$$V^{SAA}_{t,N}(x) := \frac{1}{N}\sum_{k=1}^{N} \hat{V}_t\big(x, \xi^k\big)$$

$\xi^1, \cdots, \xi^N$ drawn by Monte Carlo (ex Shapiro 2011)



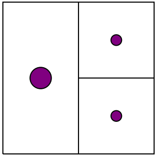
SAA $N = 20$

Partition-based

$$V_{t,\mathcal{P}}(x) := \sum_{P \in \mathcal{P}} \check{p}_{t,P}\, \hat{V}_t(x, \check{\xi}_{t,P})$$

with $\check{p}_{t,P} := \mathbb{P}\big[\boldsymbol{\xi}_t \in P\big]$ and $\check{\xi}_{t,P} := \mathbb{E}\big[\boldsymbol{\xi}_t \,|\, \boldsymbol{\xi}_t \in P\big]$

If $\xi \mapsto \hat{V}(x, \xi)$ is convex, $V_{t,\mathcal{P}}(x) \leqslant V_t(x)$ (Jensen, Kuhn)



Partition-based

Exact quantization

Definition

A MSLP admits a local exact quantization at time t on x if there exists a finitely supported $(\check{\xi}_t)_{t\in[T]}$ such that

$$V_t(x) = \mathbb{E}\big[\hat{V}_t(x,\xi_t)\big] = \mathbb{E}\big[\hat{V}_t(x,\check{\xi}_t)\big].$$

We call an exact quantization

- uniform if it is locally exact at all $x \in \mathbb{R}^{n_t}$, and all $t \in [T]$.
- universal if there exists a partition $\mathcal{P}_{t,x}$ such that the induced quantization is exact at time t on x, for all distributions of $(\xi_\tau)_{\tau\in[T]}$.

Questions:

1. Under which condition does there exist an exact quantization ?
2. Can we construct a uniform and universal exact quantization ?

Exact quantization

Definition

A MSLP admits a local exact quantization at time t on x if there exists a finitely supported $(\check{\xi}_t)_{t\in[T]}$ such that

$$V_t(x) = \mathbb{E}\big[\hat{V}_t(x,\xi_t)\big] = \mathbb{E}\big[\hat{V}_t(x,\check{\xi}_t)\big].$$

We call an exact quantization

- uniform if it is locally exact at all $x \in \mathbb{R}^{n_t}$, and all $t \in [T]$.
- universal if there exists a partition $\mathcal{P}_{t,x}$ such that the induced quantization is exact at time t on x, for all distributions of $(\xi_\tau)_{\tau\in[T]}$.

Questions:

1. Under which condition does there exist an exact quantization ?
2. Can we construct a uniform and universal exact quantization ?

Conditions for the existence of an exact quantization ?

Assume $V_{t+1} \equiv 0$ and denote $V := V_t$, $\hat{V} := \hat{V}_t$ and $\boldsymbol{\xi} := \boldsymbol{\xi}_t$ for now.

$$V(x) = \mathbb{E}\big[\hat{V}\big(x, \boldsymbol{\xi}\big)\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^n} & \boldsymbol{c}^\top y \\ \text{s.t.} & \boldsymbol{A} y + \boldsymbol{B} x \leqslant \boldsymbol{b} \end{bmatrix}$$

We have an exact quantization if and only if there exists a finitely supported noise $\check{\boldsymbol{\xi}}$ such that

$$\mathbb{E}\big[\hat{V}(x, \boldsymbol{\xi})\big] = \mathbb{E}\big[\hat{V}(x, \check{\boldsymbol{\xi}})\big].$$

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	?	?	?
Uniform	?	?	?

A first counter example

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	?	?	?
Uniform	?	?	?

Let $\boldsymbol{A} = (-\boldsymbol{u})$, $\boldsymbol{B} \equiv (0)$, $\boldsymbol{b} \equiv (-1)$ where $\boldsymbol{u} \sim \mathcal{U}\big([1,2]\big)$.

$$\hat{V}(x, \xi) = \begin{array}{ll} \min\limits_{y \in \mathbb{R}} & y \\ \text{s.t.} & uy \geqslant 1 \end{array} = \frac{1}{u}$$

By strict convexity, for all partition $\mathcal{P}$

$$\sum_{P \in \mathcal{P}} \check{p}_P \hat{V}\big(x, \check{\xi}_P\big) < V(x) = \mathbb{E}\left[\frac{1}{\boldsymbol{u}}\right]$$

with $\check{p}_P = \mathbb{P}\big[\boldsymbol{\xi} \in P\big]$, $\check{\xi}_P = \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big]$.

- ➥ There is no partition-based (local, uniform or universal) exact quantization result for $\boldsymbol{A}$ non-finitely supported.
- ➥ For now on, A is deterministic: fixed recourse.

A first counter example

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	?	?	?
Uniform	?	?	?

Let $\boldsymbol{A} = (-\boldsymbol{u})$, $\boldsymbol{B} \equiv (0)$, $\boldsymbol{b} \equiv (-1)$ where $\boldsymbol{u} \sim \mathcal{U}([1, 2])$.

$$\hat{V}(x, \xi) = \begin{array}{ll} \min\limits_{y \in \mathbb{R}} & y \\ \text{s.t.} & uy \geqslant 1 \end{array} = \frac{1}{u}$$

By strict convexity, for all partition $\mathcal{P}$

$$\sum_{P \in \mathcal{P}} \check{p}_P \hat{V}\big(x, \check{\xi}_P\big) < V(x) = \mathbb{E}\left[\frac{1}{\boldsymbol{u}}\right]$$

with $\check{p}_P = \mathbb{P}\big[\boldsymbol{\xi} \in P\big]$, $\check{\xi}_P = \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big]$.

➥ There is no partition-based (local, uniform or universal) exact quantization result for $\boldsymbol{A}$ non-finitely supported.

➥ For now on, A is deterministic: fixed recourse.

A first counter example

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	?	?	?
Uniform	?	?	?

Let $\boldsymbol{A} = (-\boldsymbol{u})$, $\boldsymbol{B} \equiv (0)$, $\boldsymbol{b} \equiv (-1)$ where $\boldsymbol{u} \sim \mathcal{U}\big([1, 2]\big)$.

$$\hat{V}(x, \xi) = \begin{array}{ll} \min\limits_{y \in \mathbb{R}} & y \\ \text{s.t.} & uy \geqslant 1 \end{array} = \frac{1}{u}$$

By strict convexity, for all partition $\mathcal{P}$

$$\sum_{P \in \mathcal{P}} \check{p}_P \hat{V}\big(x, \check{\xi}_P\big) < V(x) = \mathbb{E}\left[\frac{1}{\boldsymbol{u}}\right]$$

with $\check{p}_P = \mathbb{P}\big[\boldsymbol{\xi} \in P\big]$, $\check{\xi}_P = \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big]$.

- ➡ There is no partition-based (local, uniform or universal) exact quantization result for $\boldsymbol{A}$ non-finitely supported.
- ➡ For now on, A is deterministic: fixed recourse.

A first counter example

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	?
Uniform	$\times$	?	?

Let $\boldsymbol{A} = (-\boldsymbol{u})$, $\boldsymbol{B} \equiv (0)$, $\boldsymbol{b} \equiv (-1)$ where $\boldsymbol{u} \sim \mathcal{U}\big([1,2]\big)$.

$$\hat{V}(x,\xi) = \begin{array}{ll} \min\limits_{y \in \mathbb{R}} & y \\ \text{s.t.} & uy \geqslant 1 \end{array} = \frac{1}{u}$$

By strict convexity, for all partition $\mathcal{P}$

$$\sum_{P \in \mathcal{P}} \check{p}_P \hat{V}\big(x, \check{\xi}_P\big) < V(x) = \mathbb{E}\left[\frac{1}{\boldsymbol{u}}\right]$$

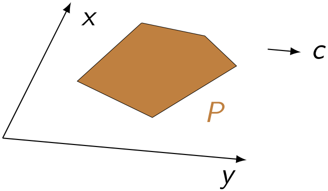
with $\check{p}_P = \mathbb{P}\big[\boldsymbol{\xi} \in P\big]$, $\check{\xi}_P = \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in P\big]$.

➥ There is no partition-based (local, uniform or universal) exact quantization result for $\boldsymbol{A}$ non-finitely supported.

➥ For now on, A is deterministic: fixed recourse.

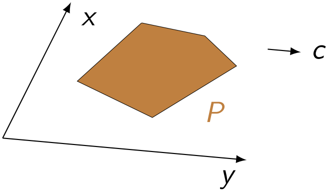
Uniform exact quantization and polyhedrality

$$
\begin{aligned}
\hat{V}(x,\xi) = \min_{y \in \mathbb{R}^m} \ & c^\top y \\
\text{s.t.} \ & Ay + Bx \leqslant b
\end{aligned}
$$



Uniform exact quantization and polyhedrality

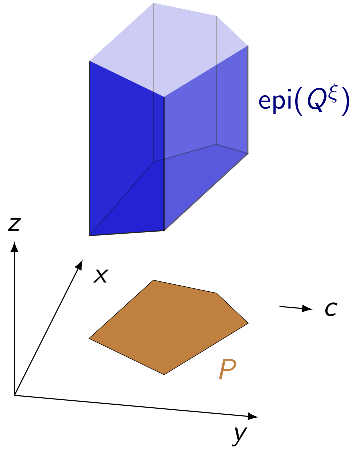
$$\hat{V}(x,\xi) = \min_{y\in\mathbb{R}^m} \; c^\top y$$
$$\text{s.t.} \quad (x,y) \in P$$



Uniform exact quantization and polyhedrality

$$\hat{V}(x,\xi) = \min_{y \in \mathbb{R}^m} \; c^\top y \quad \text{s.t.} \quad (x,y) \in P$$
$$= \min_{y \in \mathbb{R}^m} Q^\xi(x,y)$$

with $Q^\xi(x,y) := c^\top y + \mathbb{I}_{(x,y)\in P}$.



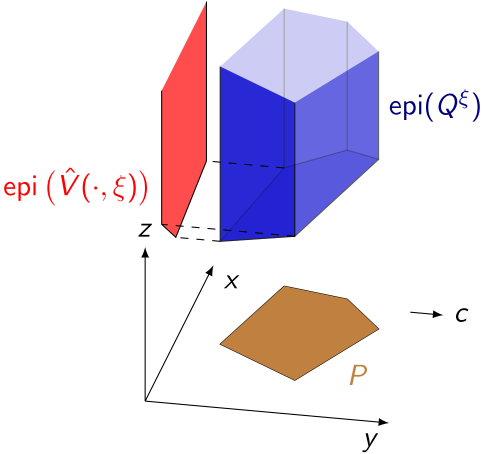
Uniform exact quantization and polyhedrality

$$\hat{V}(x,\xi) = \min_{y \in \mathbb{R}^m} \; c^\top y \quad \text{s.t.} \quad (x,y) \in P$$

$$= \min_{y \in \mathbb{R}^m} Q^\xi(x,y)$$

with $Q^\xi(x,y) := c^\top y + \mathbb{I}_{(x,y)\in P}$.

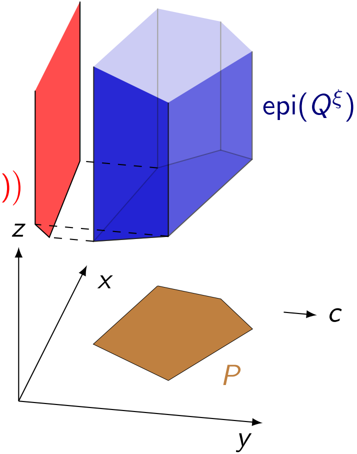
$\hat{V}(\cdot,\xi)$ is polyhedral because epi $\big(\hat{V}(\cdot,\xi)\big)$ is the projection of epi(Q^ξ).



Uniform exact quantization and polyhedrality

$$\hat{V}(x,\xi) = \min_{y\in\mathbb{R}^m} \; c^\top y \quad \text{s.t.} \quad (x,y) \in P$$

$$= \min_{y\in\mathbb{R}^m} Q^\xi(x,y)$$



with $Q^\xi(x,y) := c^\top y + \mathbb{I}_{(x,y)\in P}$.

$\hat{V}(\cdot,\xi)$ is polyhedral because epi $\left(\hat{V}(\cdot,\xi)\right)$ is the projection of epi(Q^ξ).

$$V(x) = \mathbb{E}\left[\hat{V}(x,\boldsymbol{\xi})\right] = \sum_{\xi\in\text{supp}(\check{\boldsymbol{\xi}})} p_\xi \hat{V}(x,\xi)$$

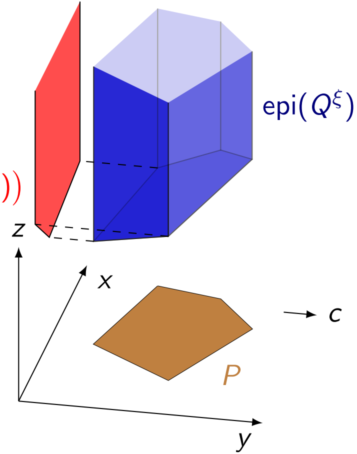
➥ If the noise is finitely supported, then V is polyhedral

Uniform exact quantization and polyhedrality

$$\hat{V}(x,\xi) = \min_{y\in\mathbb{R}^m} \ c^\top y \quad \text{s.t.} \quad (x,y)\in P$$

$$= \min_{y\in\mathbb{R}^m} Q^\xi(x,y)$$

with $Q^\xi(x,y) := c^\top y + \mathbb{I}_{(x,y)\in P}$.



$\hat{V}(\cdot,\xi)$ is polyhedral because epi $\left(\hat{V}(\cdot,\xi)\right)$ is the projection of epi(Q^ξ).

$$V(x) = \mathbb{E}\left[\hat{V}(x,\boldsymbol{\xi})\right] = \sum_{\xi\in\mathrm{supp}(\check{\boldsymbol{\xi}})} p_\xi \hat{V}(x,\xi)$$

- ➥ If the noise is finitely supported, then V is polyhedral
- ➥ Existence of uniform exact quantization implies polyhedrality of V.

Counter examples with stochastic constraints

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	?
Uniform	$\times$	?	?

$\boldsymbol{u}$ is uniform on $[0, 1]$

Counter examples with stochastic constraints

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	?
Uniform	$\times$	?	?

Stochastic $\boldsymbol{B}$

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & \boldsymbol{u}x - y \leqslant 0 \\ & y \geqslant 1 \end{bmatrix}$$

$$= \mathbb{E}\big[\max(\boldsymbol{u}x, 1)\big]$$

$$= \begin{cases} 1 & \text{if } x \leqslant 1 \\ \frac{x}{2} + \frac{1}{2x} & \text{if } x \geqslant 1 \end{cases}$$

$\boldsymbol{u}$ is uniform on $[0, 1]$

Counter examples with stochastic constraints

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	?
Uniform	$\times$	?	?

Stochastic $\boldsymbol{B}$

$$V(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & \boldsymbol{u}x - y \leqslant 0 \\ & y \geqslant 1 \end{array}\right]$$

$$= \mathbb{E}\left[\max(\boldsymbol{u}x, 1)\right]$$

$$= \begin{cases} 1 & \text{if } x \leqslant 1 \\ \frac{x}{2} + \frac{1}{2x} & \text{if } x \geqslant 1 \end{cases}$$

Stochastic $\boldsymbol{b}$

$$V(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & y \geqslant \boldsymbol{u} \\ & x - y \leqslant 0 \end{array}\right]$$

$$= \mathbb{E}\left[\max(x, \boldsymbol{u})\right]$$

$$= \begin{cases} \frac{1}{2} & \text{if } x \leqslant 0 \\ \frac{x^2+1}{2} & \text{if } x \in [0,1] \\ x & \text{if } x \geqslant 1 \end{cases}$$

$\boldsymbol{u}$ is uniform on $[0, 1]$

Counter examples with stochastic constraints

	A	(B, b)	c
Local	×	?	?
Uniform	×	?	?

Stochastic B

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & ux - y \leqslant 0 \\ & y \geqslant 1 \end{bmatrix} = \mathbb{E}\big[\max(ux, 1)\big] = \begin{cases} 1 & \text{if } x \leqslant 1 \\ \frac{x}{2} + \frac{1}{2x} & \text{if } x \geqslant 1 \end{cases}$$

Stochastic b

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & y \geqslant u \\ & x - y \leqslant 0 \end{bmatrix} = \mathbb{E}\big[\max(x, u)\big] = \begin{cases} \frac{1}{2} & \text{if } x \leqslant 0 \\ \frac{x^2+1}{2} & \text{if } x \in [0, 1] \\ x & \text{if } x \geqslant 1 \end{cases}$$

➥ V is not polyhedral $\Rightarrow$ No uniform exact quantization for non-finitely supported B and b.

---

u is uniform on $[0, 1]$

Counter examples with stochastic constraints

	A	(B, b)	c
Local	×	?	?
Uniform	×	×	?

Stochastic B

$$V(x) = \mathbb{E}\left[\begin{array}{ll}\min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & \boldsymbol{u}x - y \leqslant 0 \\ & y \geqslant 1\end{array}\right]$$

$$= \mathbb{E}\big[\max(\boldsymbol{u}x, 1)\big]$$

$$= \begin{cases} 1 & \text{if } x \leqslant 1 \\ \frac{x}{2} + \frac{1}{2x} & \text{if } x \geqslant 1 \end{cases}$$

Stochastic b

$$V(x) = \mathbb{E}\left[\begin{array}{ll}\min\limits_{y \in \mathbb{R}^m} & y \\ \text{s.t.} & y \geqslant \boldsymbol{u} \\ & x - y \leqslant 0\end{array}\right]$$

$$= \mathbb{E}\big[\max(x, \boldsymbol{u})\big]$$

$$= \begin{cases} \frac{1}{2} & \text{if } x \leqslant 0 \\ \frac{x^2+1}{2} & \text{if } x \in [0, 1] \\ x & \text{if } x \geqslant 1 \end{cases}$$

➥ V is not polyhedral $\Rightarrow$ No uniform exact quantization for non-finitely supported $\boldsymbol{B}$ and $\boldsymbol{b}$.

$\boldsymbol{u}$ is uniform on $[0, 1]$

Remaining cases

$$V(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y \in \mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & \boldsymbol{A}y + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{array}\right]$$

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	?	?
Uniform	×	×	?

Remaining cases

$$V(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y\in\mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & \boldsymbol{A}y + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{array}\right]$$

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	?	✓
Uniform	×	×	✓

Theorem (FGL 2021)

If A, B and b are deterministic,
then there exists a universal and uniform exact quantization.

Remaining cases

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y\in\mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & \boldsymbol{A}y + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix}$$

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	?	✓
Uniform	×	×	✓

Theorem (FGL 2021)

If A, B and b are deterministic,
then there exists a universal and uniform exact quantization.

Theorem (FL 2022)

If A is deterministic,
then there exists a universal and local exact quantization.

Remaining cases

$$V(x) = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & \boldsymbol{A}y + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix}$$

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	✓	✓
Uniform	×	×	✓

Theorem (FGL 2021)

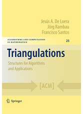
If A, B and b are deterministic,
then there exists a universal and uniform exact quantization.

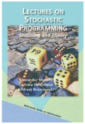
Theorem (FL 2022)

If A is deterministic,
then there exists a universal and local exact quantization.

Contents of the manuscript and articles

Chapter 3:





Chapter 4:

M. Forcier, S. Gaubert, V. Leclère
Exact quantization of multistage stochastic linear problems,
arXiv preprint arXiv:2107.09566 (2021),
Best student paper, ECSO-CMS 2022, Venice.

Chapter 5:

M. Forcier, V. Leclère
Generalized adaptive partition-based method for two-stage stochastic linear programs: convergence and generalization,
Operation Research Letters, to appear (2022).

Chapter 6:

M. Forcier, V. Leclère
Convergence of Trajectory Following Dynamic Programming algorithms for multistage stochastic problems without finite support assumptions,
HAL Id : hal-03683697 (2022).

Contents

1 Universal Exact Quantization for cost
- Local in 2-stage
- Uniform in 2-stage
- Uniform in multistage
- Complexity results

2 Local and universal exact Quantization for constraints
- Adapted partitions
- Adaptive Partition-based Methods
- Convergence, complexity and numerical results

3 Trajectory Following Dynamic Programming

4 Conclusion and perspectives

Contents

1 Universal Exact Quantization for cost
- Local in 2-stage
- Uniform in 2-stage
- Uniform in multistage
- Complexity results

2 Local and universal exact Quantization for constraints
- Adapted partitions
- Adaptive Partition-based Methods
- Convergence, complexity and numerical results

3 Trajectory Following Dynamic Programming

4 Conclusion and perspectives

Reformulation of $V(x)$ highlighting the role of the fiber P_x

For a given x, (we still assume $V_{t+1} \equiv 0$)

$$V(x) := \mathbb{E}\begin{bmatrix} \min_{y \in \mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & Ay + Bx \leqslant b \end{bmatrix}$$

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big] \quad \text{where} \quad P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

Illustrative running example:

$$P_x := \{y \in \mathbb{R}^m \mid \|y\|_1 \leqslant 1, \\ y_1 \leqslant x,\ y_2 \leqslant x\}$$

Reformulation of $V(x)$ highlighting the role of the fiber P_x

For a given x, (we still assume $V_{t+1} \equiv 0$)

$$V(x) := \mathbb{E}\begin{bmatrix} \min\limits_{y\in\mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & Ay + Bx \leqslant b \end{bmatrix}$$

$$V(x) = \mathbb{E}\big[\min_{y\in P_x} \boldsymbol{c}^\top y\big] \quad \text{where} \quad P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

Illustrative running example:

$$P_x := \{y \in \mathbb{R}^m \mid \|y\|_1 \leqslant 1, \; y_1 \leqslant x, \; y_2 \leqslant x\}$$

Reformulation of $V(x)$ highlighting the role of the fiber P_x

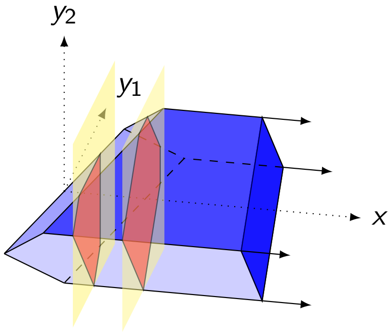
For a given x, (we still assume $V_{t+1} \equiv 0$)

$$V(x) := \mathbb{E}\begin{bmatrix} \min_{y \in \mathbb{R}^m} & \boldsymbol{c}^\top y \\ \text{s.t.} & Ay + Bx \leqslant b \end{bmatrix}$$

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big] \quad \text{where} \quad P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

Illustrative running example:

$$P_x := \{y \in \mathbb{R}^m \mid \|y\|_1 \leqslant 1, \\ y_1 \leqslant x,\ y_2 \leqslant x\}$$



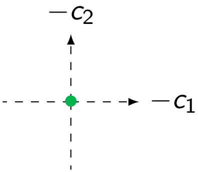
Normal fan $\mathcal{N}(P_x)$

Definition

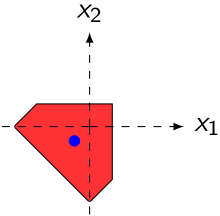
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

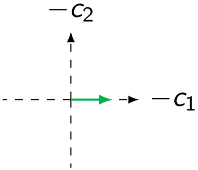
Normal fan $\mathcal{N}(P_x)$

Definition

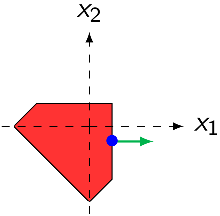
The normal fan of the fiber P_x *is*

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

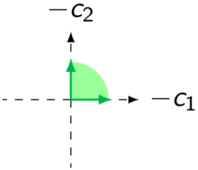
Normal fan $\mathcal{N}(P_x)$

Definition

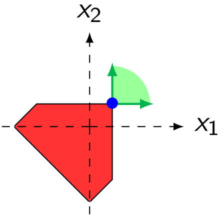
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

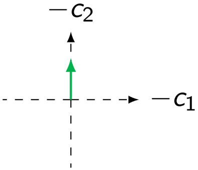
Normal fan $\mathcal{N}(P_x)$

Definition

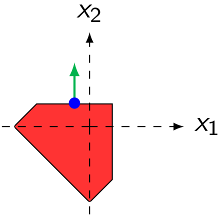
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top (y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y*.*



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

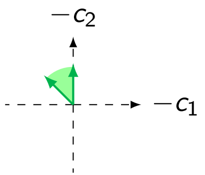
Normal fan $\mathcal{N}(P_x)$

Definition

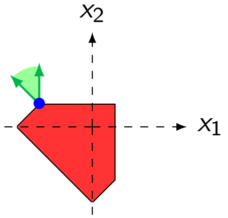
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of P_x at y.*



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

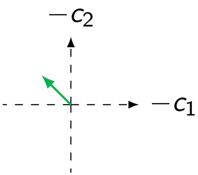
Normal fan $\mathcal{N}(P_x)$

Definition

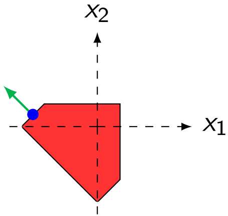
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of P_x at y.*



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

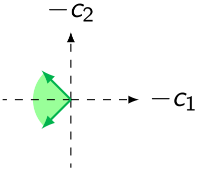
Normal fan $\mathcal{N}(P_x)$

Definition

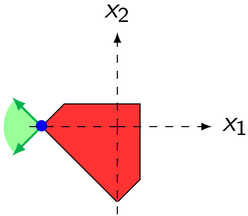
The normal fan of the fiber P_x *is*

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

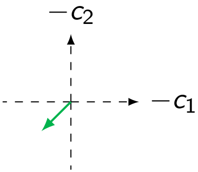
Normal fan $\mathcal{N}(P_x)$

Definition

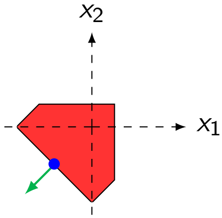
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

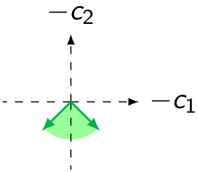
Normal fan $\mathcal{N}(P_x)$

Definition

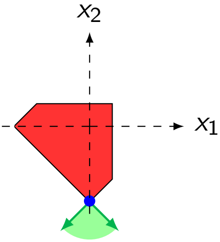
The normal fan of the fiber P_x *is*

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

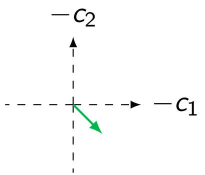
Normal fan $\mathcal{N}(P_x)$

Definition

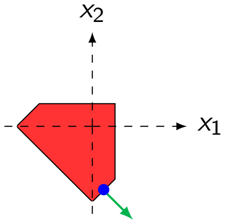
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

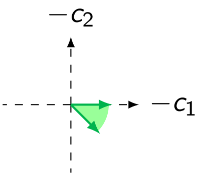
Normal fan $\mathcal{N}(P_x)$

Definition

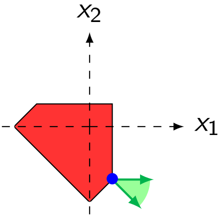
The normal fan of the fiber P_x *is*

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top (y' - y) \leqslant 0\}$ *the normal cone of* P_x *at* y.



$N_{P_x}(y)$ for $x = 0.3$



P_x, y and $N_{P_x}(y)$ for $x = 0.3$

Normal fan $\mathcal{N}(P_x)$

Definition

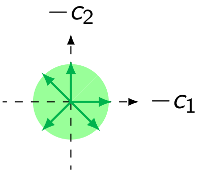
The normal fan of the fiber P_x is

$$\mathcal{N}(P_x) := \{N_{P_x}(y) \mid y \in P_x\}$$

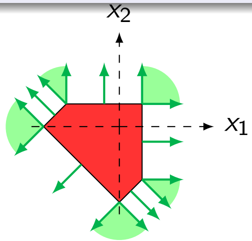
with $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top(y' - y) \leqslant 0\}$ *the normal cone of P_x at y.*

Proposition

If P_x is bounded, $\{\mathrm{ri}(N) \mid N \in \mathcal{N}(P_x)\}$ is a partition of $\mathbb{R}^m$.



$\mathcal{N}(P_x)$ for $x = 0.3$

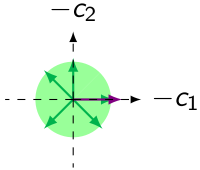


P_x and $\mathcal{N}(P_x)$ for $x = 0.3$

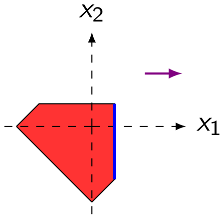
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

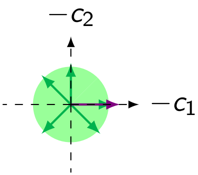


P_x for $x = 0.3$

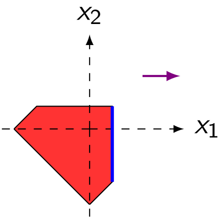
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

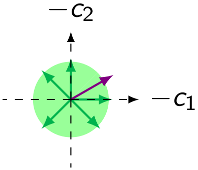


P_x for $x = 0.3$

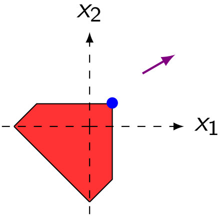
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\left[\min_{y \in P_x} \boldsymbol{c}^\top y\right]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

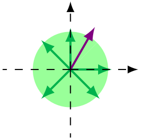


P_x for $x = 0.3$

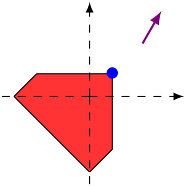
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

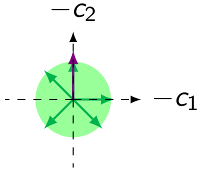


P_x for $x = 0.3$

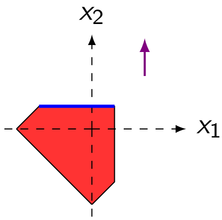
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

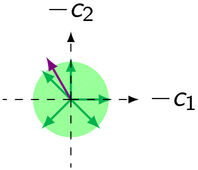


P_x for $x = 0.3$

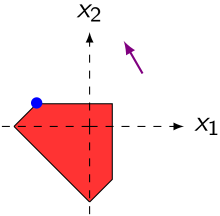
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

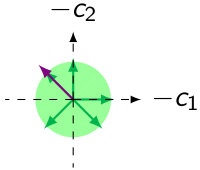


P_x for $x = 0.3$

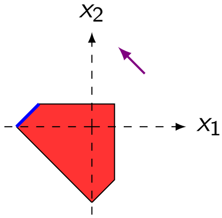
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

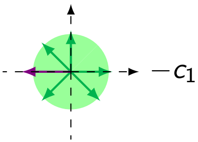


P_x for $x = 0.3$

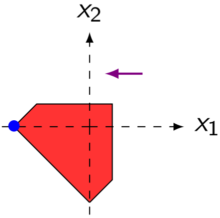
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

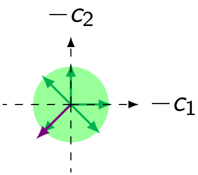


P_x for $x = 0.3$

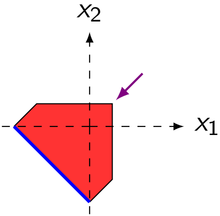
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

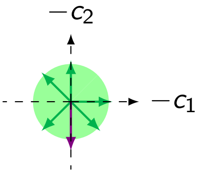


P_x for $x = 0.3$

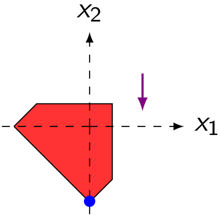
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

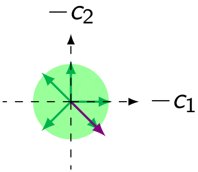


P_x for $x = 0.3$

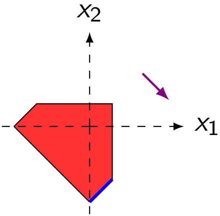
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

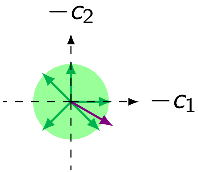


P_x for $x = 0.3$

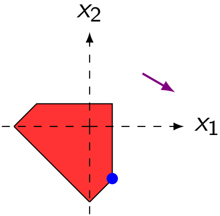
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

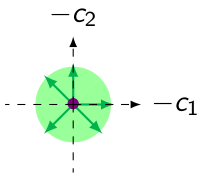


P_x for $x = 0.3$

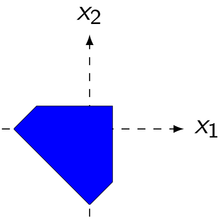
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$

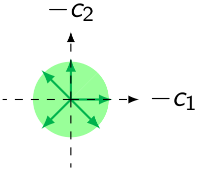


P_x for $x = 0.3$

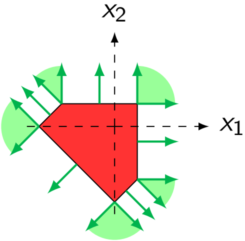
$\mathcal{N}(P_x)$: partition of cost coherent with the min

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

For any $N \in \mathcal{N}(P_x)$, $-c \mapsto \arg\min_{y \in P_x} c^\top y$ is constant for all $-c \in \mathrm{ri}(N)$.



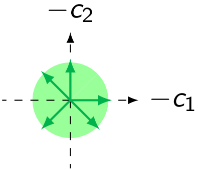
Cost $-c$ and $\mathcal{N}(P_x)$ for $x = 0.3$



P_x for $x = 0.3$

Local and universal exact quantization for $\boldsymbol{c}$

$$V(x) = \mathbb{E}\Big[\min_{y \in P_x} \boldsymbol{c}^\top y\Big]$$
$$= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\Big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \min_{y \in P_x} \boldsymbol{c}^\top y\Big]$$



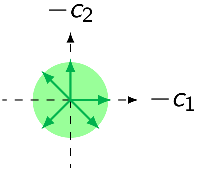
$\mathcal{N}(P_x)$ for $x = 0.3$

Local and universal exact quantization for $\boldsymbol{c}$

$$V(x) = \mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big]$$

$$= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \min_{y \in P_x} \boldsymbol{c}^\top y\big] \quad \text{where } y_N(x) \in \arg\min_{y \in P_x} \underbrace{c^\top}_{\in -\operatorname{ri} N} y.$$

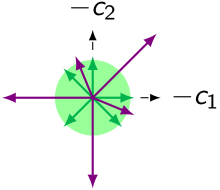
$$= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \boldsymbol{c}^\top\big] y_N(x)$$



$\mathcal{N}(P_x)$ for $x = 0.3$

Local and universal exact quantization for $\boldsymbol{c}$

$$
\begin{aligned}
V(x) &= \mathbb{E}\Big[\min_{y\in P_x} \boldsymbol{c}^\top y\Big] \\
&= \sum_{N\in\mathcal{N}(P_x)} \mathbb{E}\Big[\mathbb{1}_{\boldsymbol{c}\in -\operatorname{ri} N} \min_{y\in P_x} \boldsymbol{c}^\top y\Big] \quad \text{where } y_N(x) \in \arg\min_{y\in P_x} \underbrace{c^\top}_{\in -\operatorname{ri} N} y. \\
&= \sum_{N\in\mathcal{N}(P_x)} \mathbb{E}\Big[\mathbb{1}_{\boldsymbol{c}\in -\operatorname{ri} N} \boldsymbol{c}^\top\Big] y_N(x) \\
&= \sum_{N\in\mathcal{N}(P_x)} p_N \check{c}_N{}^\top y_N(x)
\end{aligned}
$$



$\mathcal{N}(P_x)$ and $p_N \check{c}_N$ for $x = 0.3$

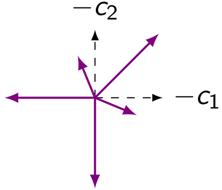
For $N \in \mathcal{N}(P_x)$,

$$
\begin{aligned}
p_N &:= \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big] \\
\check{c}_N &:= \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]
\end{aligned}
$$

We replace the continuous cost $\boldsymbol{c}$, by the discrete cost $\check{\boldsymbol{c}}$.

Local and universal exact quantization for $\boldsymbol{c}$

$$
\begin{aligned}
V(x) &= \mathbb{E}\Big[\min_{y \in P_x} \boldsymbol{c}^\top y\Big] \\
&= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\Big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \min_{y \in P_x} \boldsymbol{c}^\top y\Big] \quad \text{where } y_N(x) \in \arg\min_{y \in P_x} \underbrace{c^\top}_{\in -\operatorname{ri} N} y. \\
&= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\Big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \boldsymbol{c}^\top\Big] y_N(x) \\
&= \sum_{N \in \mathcal{N}(P_x)} p_N \check{c}_N^\top y_N(x) \\
&= \sum_{N \in \mathcal{N}(P_x)} p_N \min_{y \in P_x} \check{c}_N^\top y
\end{aligned}
$$



$p_N \check{c}_N$ for $x = 0.3$

For $N \in \mathcal{N}(P_x)$,

$$
\begin{aligned}
p_N &:= \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big] \\
\check{c}_N &:= \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]
\end{aligned}
$$

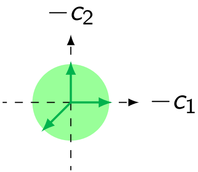
We replace the continuous cost $\boldsymbol{c}$, by the discrete cost $\boldsymbol{\check{c}}$.

Contents

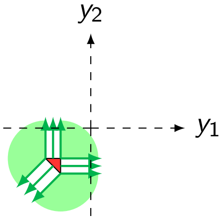
1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

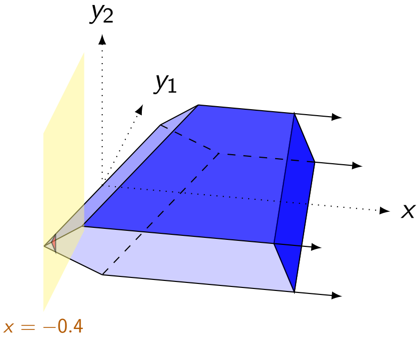
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



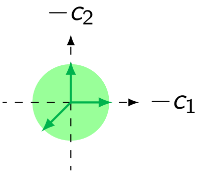
P_x and $\mathcal{N}(P_x)$



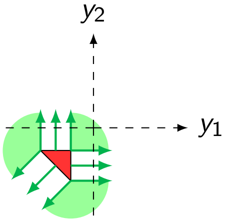
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

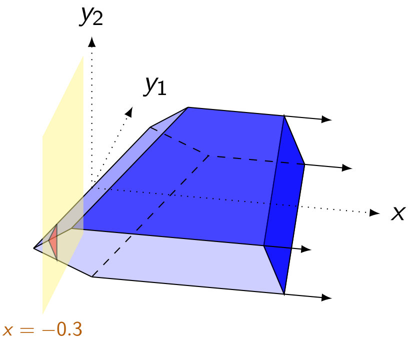
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



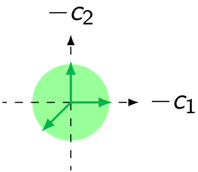
P_x and $\mathcal{N}(P_x)$



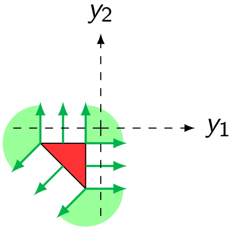
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

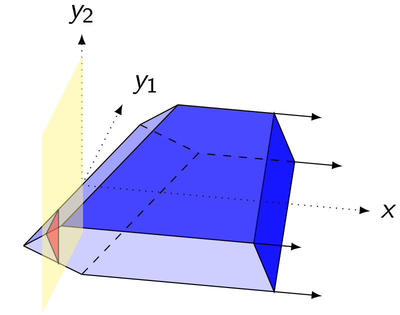
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



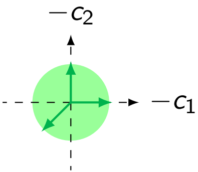
P_x and $\mathcal{N}(P_x)$



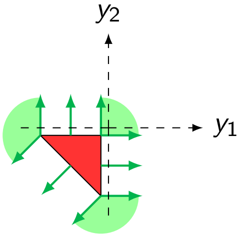
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

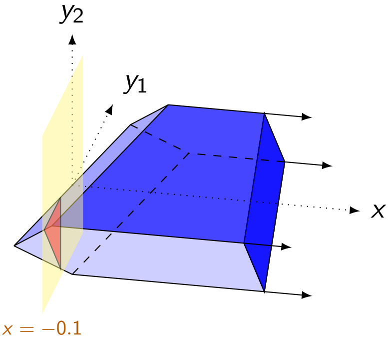
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



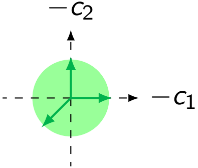
P_x and $\mathcal{N}(P_x)$



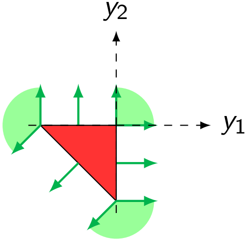
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

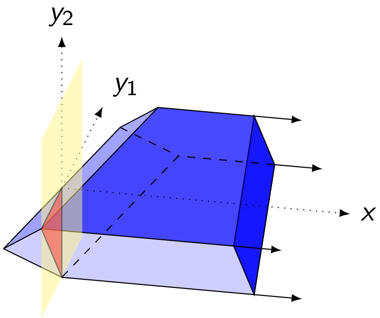
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



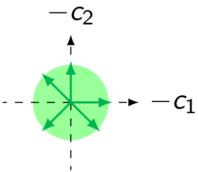
P_x and $\mathcal{N}(P_x)$



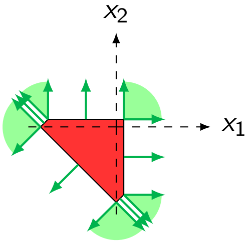
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

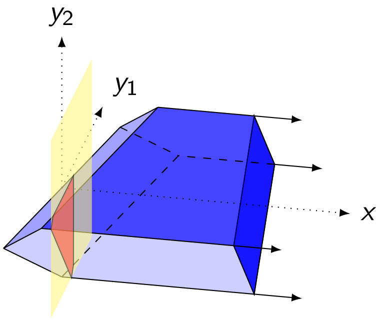
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



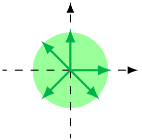
P_x and $\mathcal{N}(P_x)$



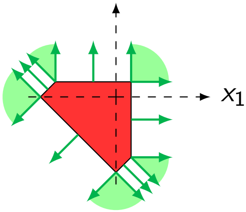
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

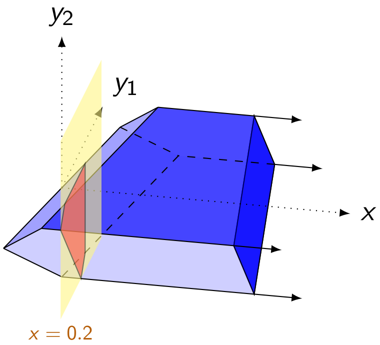
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



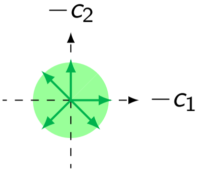
P_x and $\mathcal{N}(P_x)$



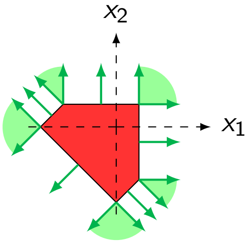
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

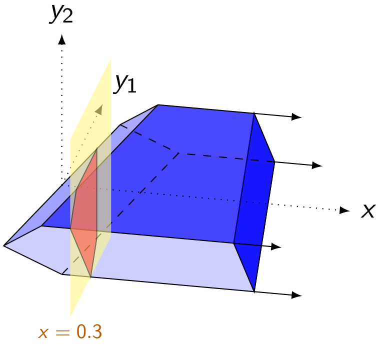
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



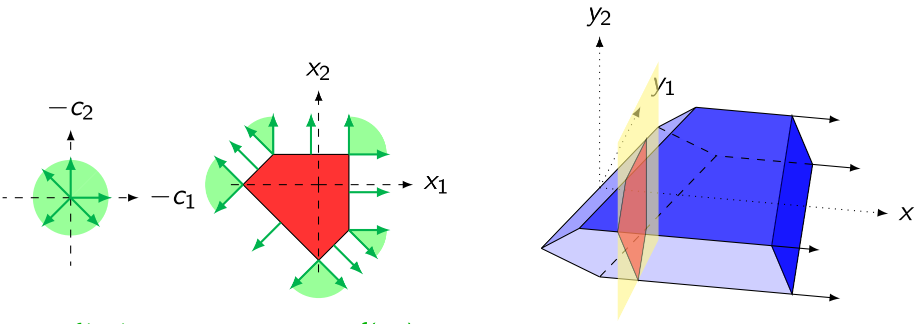
P_x and $\mathcal{N}(P_x)$



P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

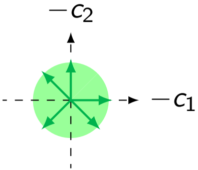
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



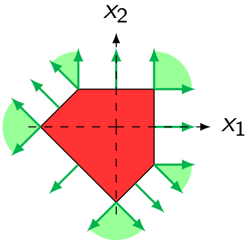
$\mathcal{N}(P_x)$ $\quad$ P_x and $\mathcal{N}(P_x)$ $\quad$ P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

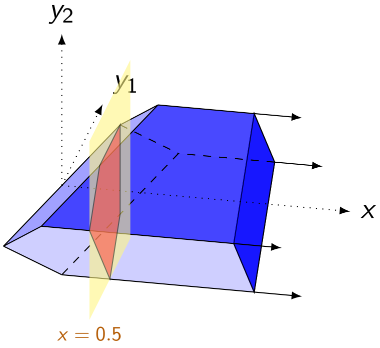
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



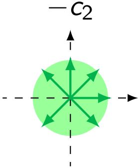
P_x and $\mathcal{N}(P_x)$



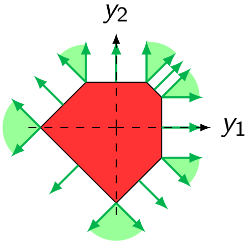
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

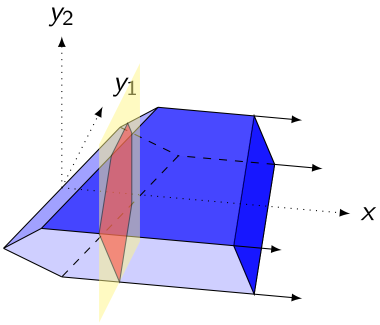
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



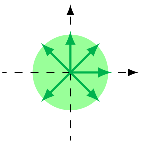
P_x and $\mathcal{N}(P_x)$



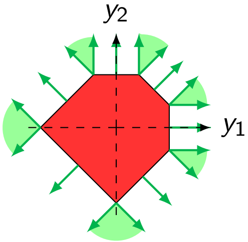
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

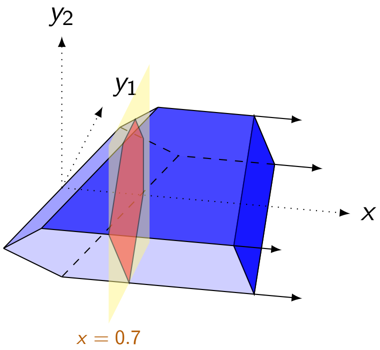
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



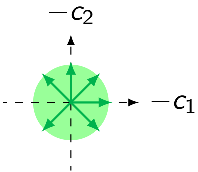
P_x and $\mathcal{N}(P_x)$



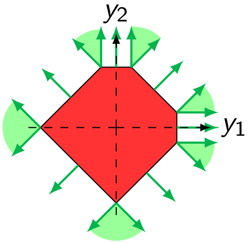
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

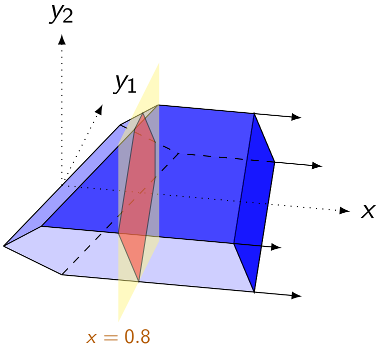
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



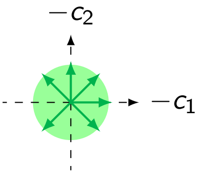
P_x and $\mathcal{N}(P_x)$



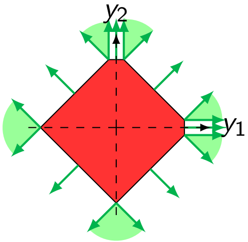
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

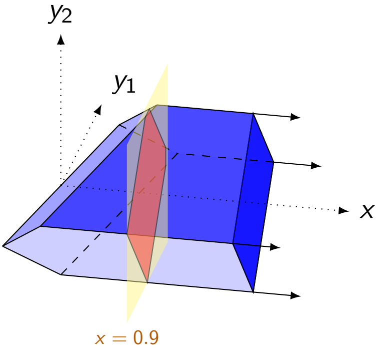
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



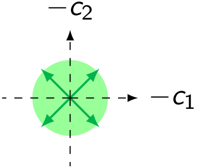
P_x and $\mathcal{N}(P_x)$



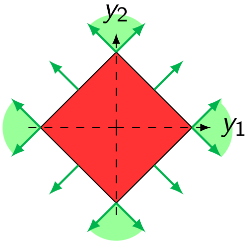
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

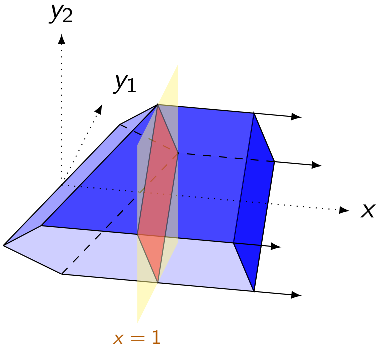
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



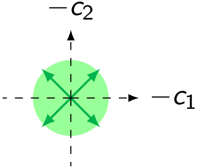
P_x and $\mathcal{N}(P_x)$



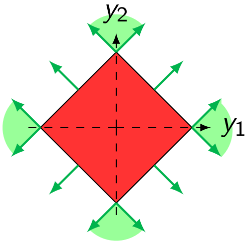
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

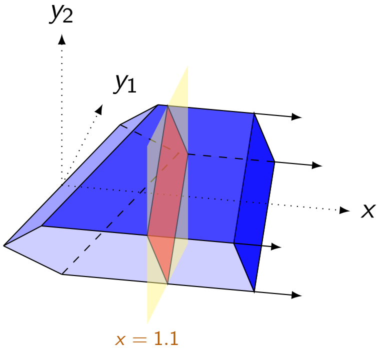
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



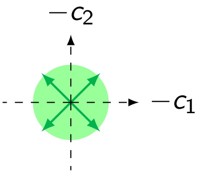
P_x and $\mathcal{N}(P_x)$



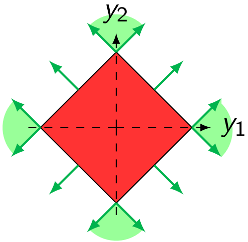
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

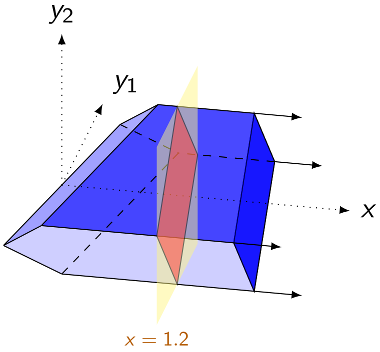
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



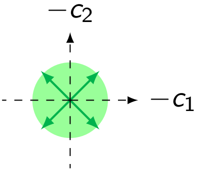
P_x and $\mathcal{N}(P_x)$



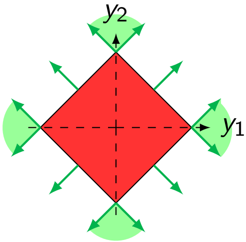
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

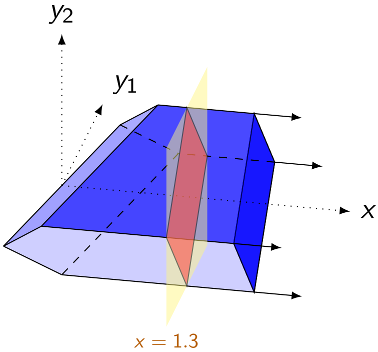
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



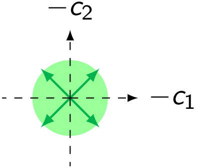
P_x and $\mathcal{N}(P_x)$



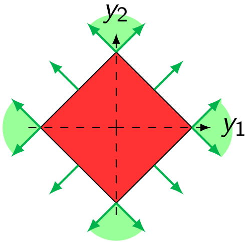
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

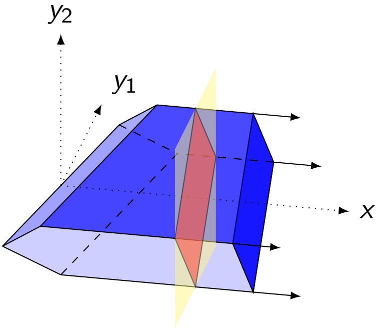
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



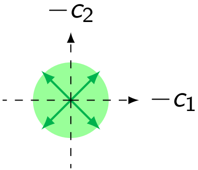
P_x and $\mathcal{N}(P_x)$



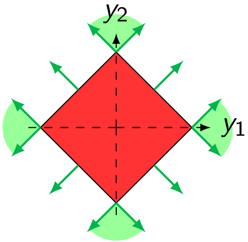
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

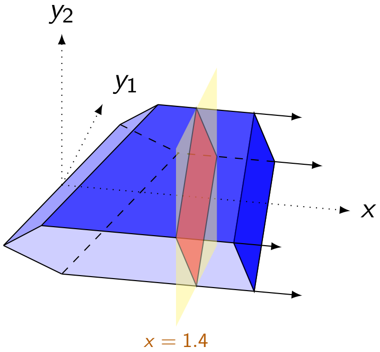
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



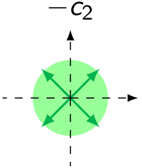
P_x and $\mathcal{N}(P_x)$



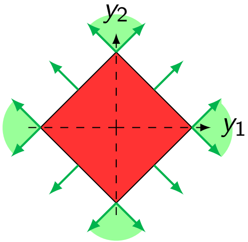
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

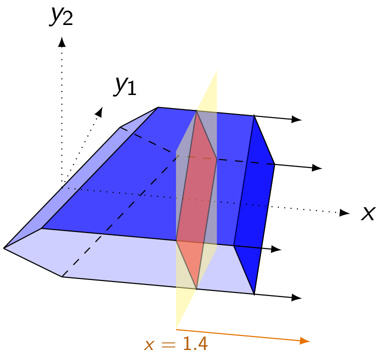
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



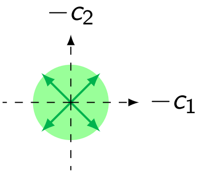
P_x and $\mathcal{N}(P_x)$



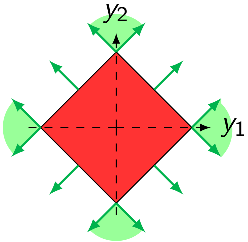
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

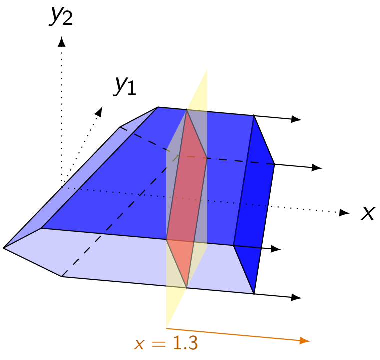
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



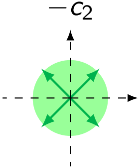
P_x and $\mathcal{N}(P_x)$



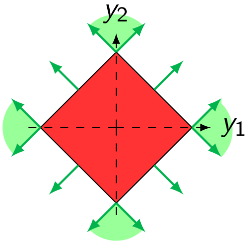
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

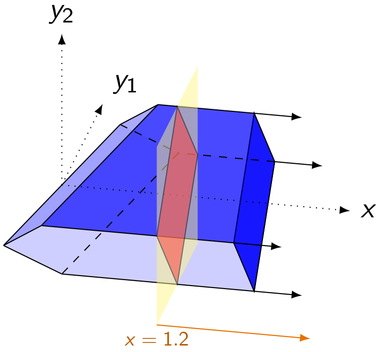
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



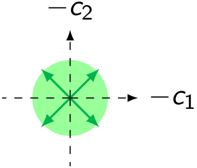
P_x and $\mathcal{N}(P_x)$



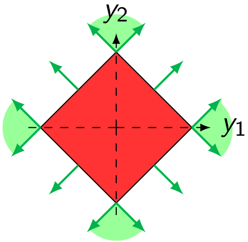
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

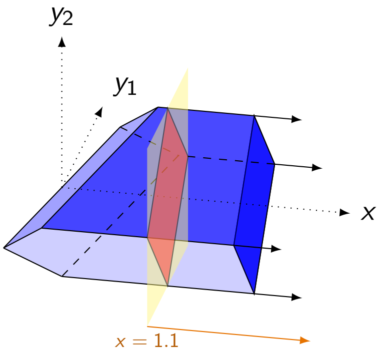
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



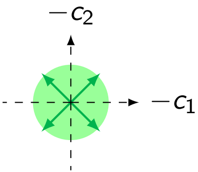
P_x and $\mathcal{N}(P_x)$



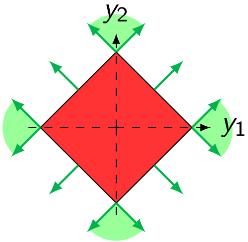
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

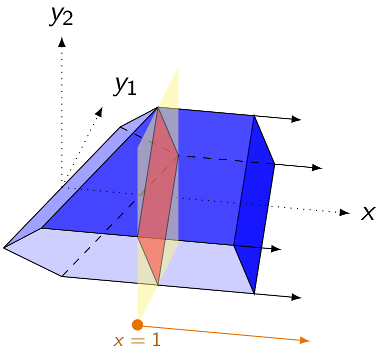
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



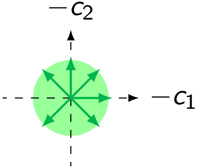
P_x and $\mathcal{N}(P_x)$



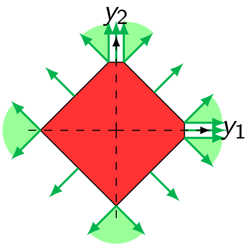
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

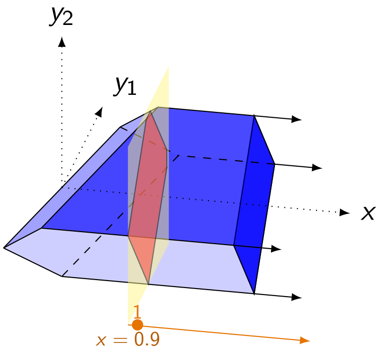
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



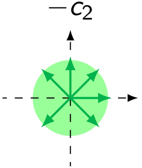
P_x and $\mathcal{N}(P_x)$



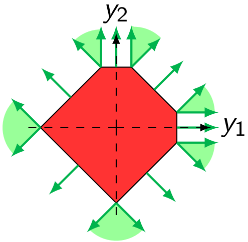
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

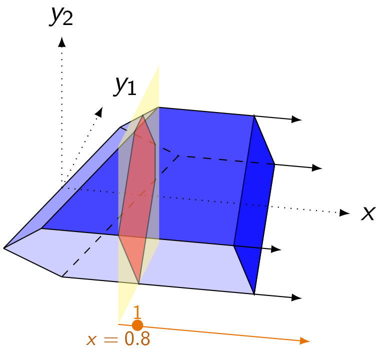
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



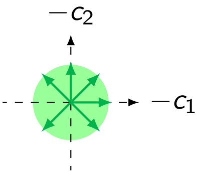
P_x and $\mathcal{N}(P_x)$



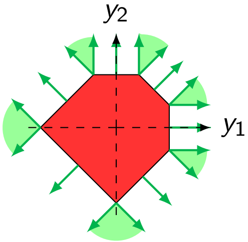
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

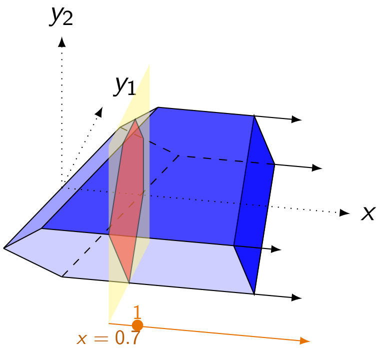
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



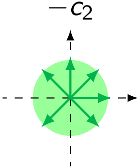
P_x and $\mathcal{N}(P_x)$



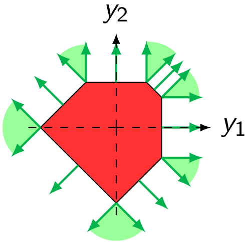
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

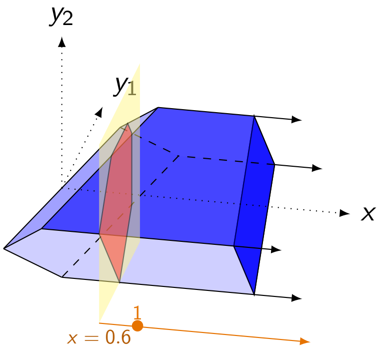
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



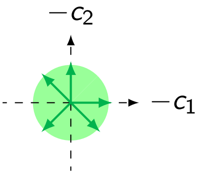
P_x and $\mathcal{N}(P_x)$



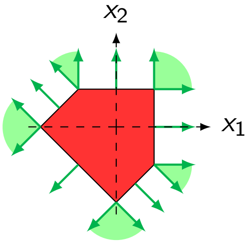
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

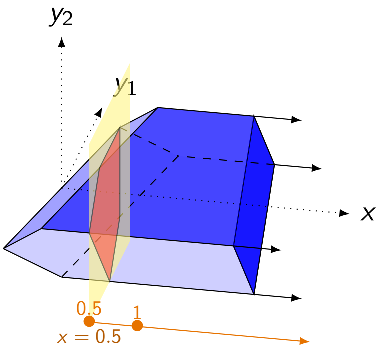
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



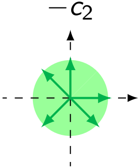
P_x and $\mathcal{N}(P_x)$



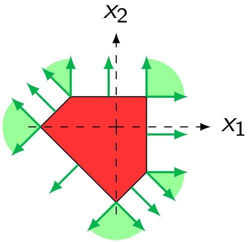
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

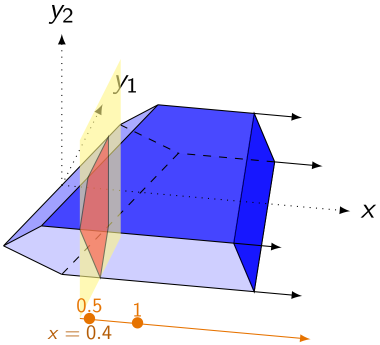
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



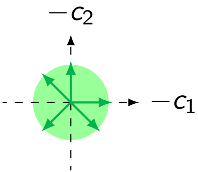
P_x and $\mathcal{N}(P_x)$



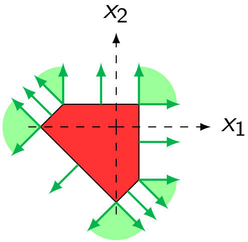
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

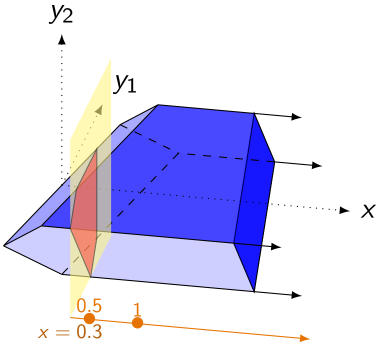
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



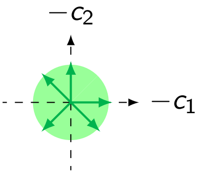
P_x and $\mathcal{N}(P_x)$



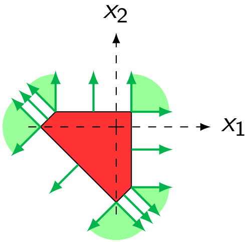
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

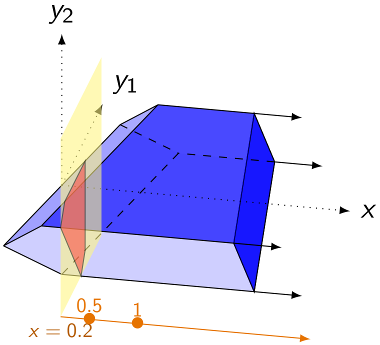
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



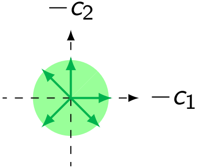
P_x and $\mathcal{N}(P_x)$



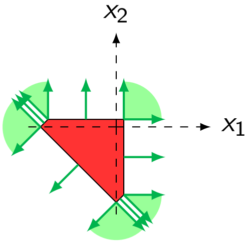
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

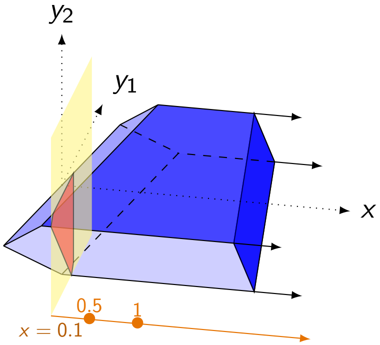
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



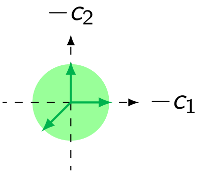
P_x and $\mathcal{N}(P_x)$



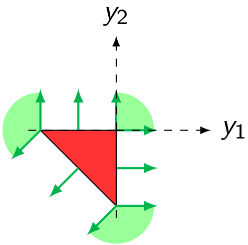
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

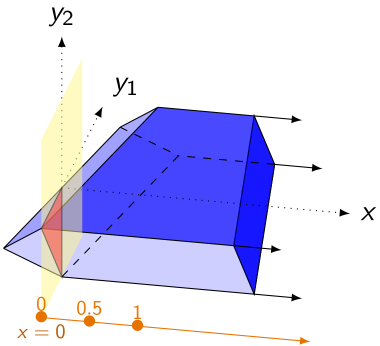
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



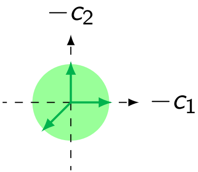
P_x and $\mathcal{N}(P_x)$



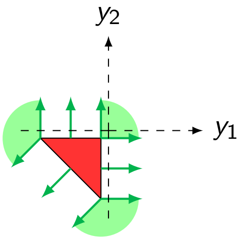
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

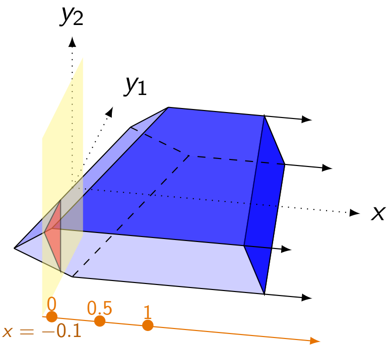
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



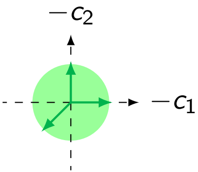
P_x and $\mathcal{N}(P_x)$



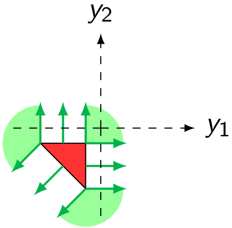
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

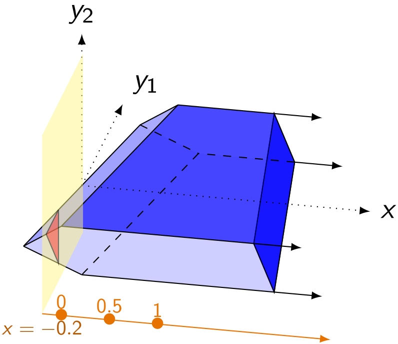
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



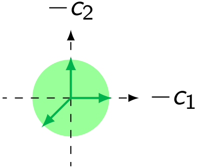
P_x and $\mathcal{N}(P_x)$



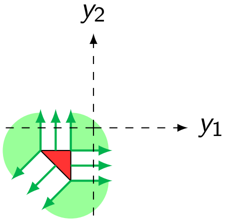
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

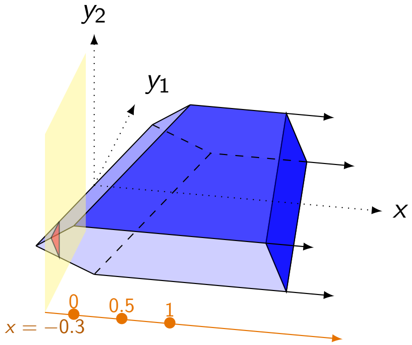
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



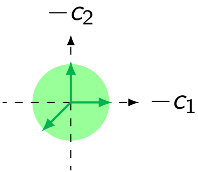
P_x and $\mathcal{N}(P_x)$



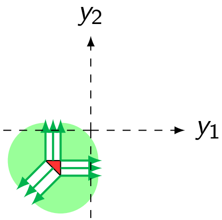
P and P_x

x is no longer fixed but $x \mapsto \mathcal{N}(P_x)$ is piecewise constant.

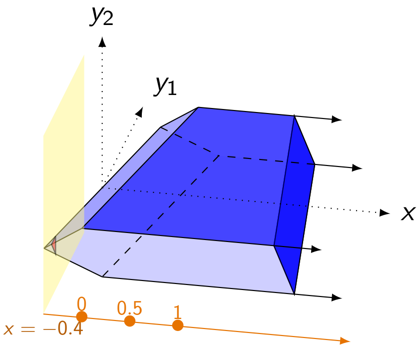
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



$\mathcal{N}(P_x)$



P_x and $\mathcal{N}(P_x)$



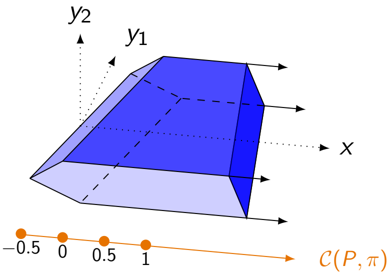
P and P_x

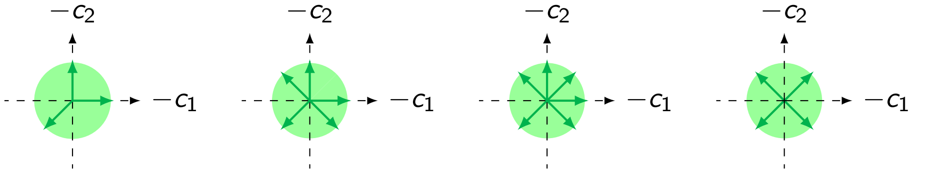
What are the constant regions of $x \mapsto \mathcal{N}(P_x)$?

Proposition

There exists a collection $\mathcal{C}(P,\pi)$ called the **chamber complex** *whose relative interior of cells are the constant regions of $x \mapsto \mathcal{N}(P_x)$.*

I.e, for $\sigma \in \mathcal{C}(P,\pi)$ and $x, x' \in \mathrm{ri}(\sigma)$, we have $\mathcal{N}(P_x) = \mathcal{N}(P_{x'}) =: \mathcal{N}_\sigma$





$\mathcal{N}_\sigma$ for $\sigma = [-0.5, 0]$ $\quad$ $\mathcal{N}_\sigma$ for $\sigma = [0, 0.5]$ $\quad$ $\mathcal{N}_\sigma$ for $\sigma = [0.5, 1]$ $\quad$ $\mathcal{N}_\sigma$ for $\sigma = [1, +\infty)$

Chamber complex

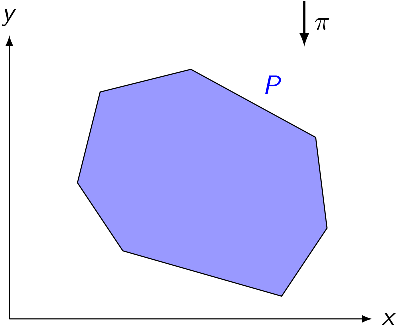
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P,\pi)$ of P along π is

$$\mathcal{C}(P,\pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F\in\mathcal{F}(P) \mid x\in\pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x,y) \mapsto x$.

Chamber complex

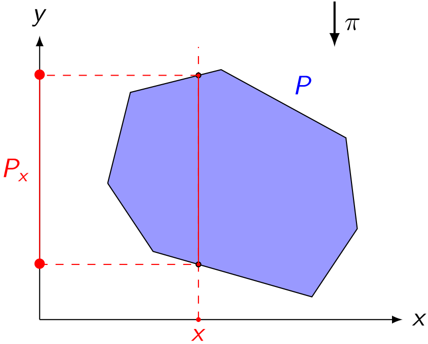
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P,\pi)$ of P along π is

$$\mathcal{C}(P,\pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F\in\mathcal{F}(P)\,|\,x\in\pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x,y) \mapsto x$.

Chamber complex

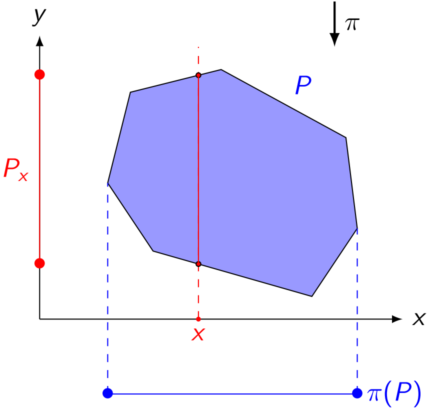
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P,\pi)$ of P along π is

$$\mathcal{C}(P,\pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F \in \mathcal{F}(P) \,|\, x \in \pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x,y) \mapsto x$.

Chamber complex

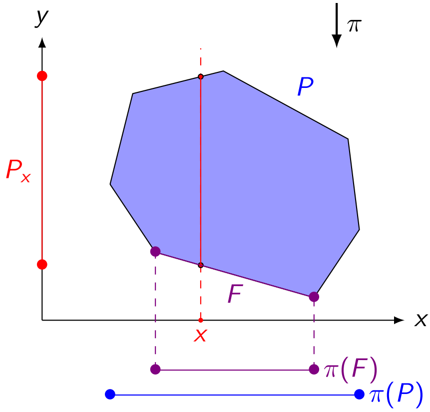
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P, \pi)$ of P along π is

$$\mathcal{C}(P, \pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F \in \mathcal{F}(P) \,|\, x \in \pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x, y) \mapsto x$.

Chamber complex

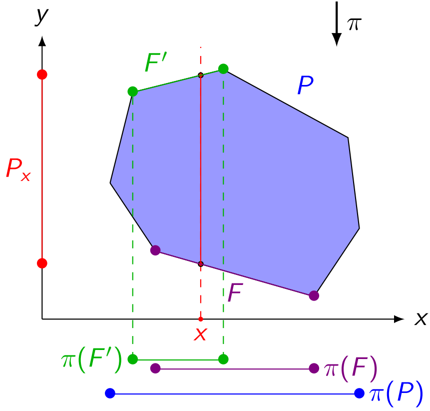
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P,\pi)$ of P along π is

$$\mathcal{C}(P,\pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F\in\mathcal{F}(P)\,|\,x\in\pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x,y)\mapsto x$.

Chamber complex

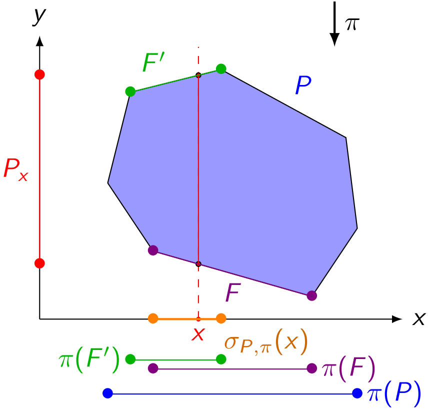
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P, \pi)$ of P along π is

$$\mathcal{C}(P, \pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

$$\sigma_{P,\pi}(x) := \bigcap_{F \in \mathcal{F}(P) \,|\, x \in \pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x, y) \mapsto x$.

Chamber complex

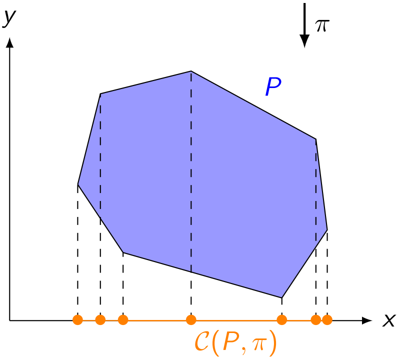
Definition (Billera, Sturmfels 92)

The *chamber complex* $\mathcal{C}(P,\pi)$ of P along π is

$$\mathcal{C}(P,\pi) := \{\sigma_{P,\pi}(x) \mid x \in \pi(P)\}$$

where

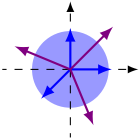
$$\sigma_{P,\pi}(x) := \bigcap_{F \in \mathcal{F}(P) \,|\, x \in \pi(F)} \pi(F)$$



where $\mathcal{F}(P)$ is the set of faces of P and π is the projection $(x,y) \mapsto x$.

Common Refinement of Normal Fans

We can quantize $\boldsymbol{c}$ on each chamber.



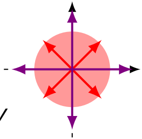
$\mathcal{N}_\sigma$ and $\check{\boldsymbol{c}}$

For all $x \in \mathrm{ri}(\sigma)$,

$$V(x) = \sum_{N \in \mathcal{N}_\sigma} p_N \min_{y \in P_x} \check{c}_N^\top y$$

For all $x' \in \mathrm{ri}(\tau)$,

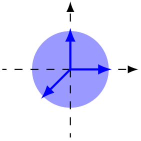
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



$\mathcal{N}_\tau$ and $\check{\boldsymbol{c}}$

Common Refinement of Normal Fans

We can quantize $\boldsymbol{c}$ on each chamber.



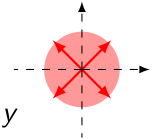
$\mathcal{N}_\sigma$

For all $x \in \mathrm{ri}(\sigma)$,

$$V(x) = \sum_{N \in \mathcal{N}_\sigma} p_N \min_{y \in P_x} \check{c}_N^\top y$$

For all $x' \in \mathrm{ri}(\tau)$,

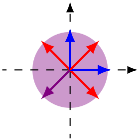
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



$\mathcal{N}_\tau$

We take the *common refinement*:

$$\mathcal{R} := \mathcal{N}_\sigma \wedge \mathcal{N}_\tau = \{N \cap N' \mid N \in \mathcal{N}_\sigma, N' \in \mathcal{N}_\tau\}$$



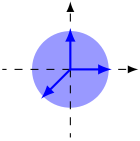
$\mathcal{N}_\sigma \wedge \mathcal{N}_\tau$

For all $x \in \mathrm{ri}(\sigma) \cup \mathrm{ri}(\tau)$,

$$V(x) = \sum_{N \in \mathcal{N}_\sigma \wedge \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$

Common Refinement of Normal Fans

We can quantize $\boldsymbol{c}$ on each chamber.



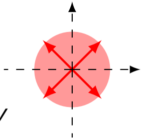
$\mathcal{N}_\sigma$

For all $x \in \mathrm{ri}(\sigma)$,

$$V(x) = \sum_{N \in \mathcal{N}_\sigma} p_N \min_{y \in P_x} \check{c}_N^\top y$$

For all $x' \in \mathrm{ri}(\tau)$,

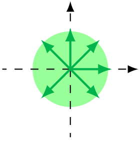
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



$\mathcal{N}_\tau$

We take the *common refinement*:

$$\mathcal{R} := \mathcal{N}_\sigma \wedge \mathcal{N}_\tau = \{N \cap N' \mid N \in \mathcal{N}_\sigma, N' \in \mathcal{N}_\tau\}$$



$\mathcal{R}$

For all $x \in \mathrm{ri}(\sigma) \cup \mathrm{ri}(\tau)$,

$$V(x) = \sum_{N \in \mathcal{R}} p_N \min_{y \in P_x} \check{c}_N^\top y$$

Uniform exact quantization for c

Let's sum up:

- local exact quantization at x induced by $\mathcal{N}(P_x)$,
- $x \mapsto \mathcal{N}(P_x)$ is constant on each $\sigma \in \mathcal{C}(P, \pi)$,
- local exact quantization at $\mathrm{ri}(\sigma)$ induced by $\mathcal{N}_\sigma$,
- local exact quantization at $\mathrm{ri}(\sigma) \cup \mathrm{ri}(\tau)$ induced by $\mathcal{N}_\sigma \wedge \mathcal{N}_\tau$.

Uniform exact quantization for $\boldsymbol{c}$

Let's sum up:

- local exact quantization at x induced by $\mathcal{N}(P_x)$,
- $x \mapsto \mathcal{N}(P_x)$ is constant on each $\sigma \in \mathcal{C}(P,\pi)$,
- local exact quantization at $\mathrm{ri}(\sigma)$ induced by $\mathcal{N}_\sigma$,
- local exact quantization at $\mathrm{ri}(\sigma) \cup \mathrm{ri}(\tau)$ induced by $\mathcal{N}_\sigma \wedge \mathcal{N}_\tau$.

Theorem (FGL21, Uniform and universal quantization of the cost)

Let $\mathcal{R} = \bigwedge_{\sigma \in \mathcal{C}(P,\pi)} -\mathcal{N}_\sigma$, *then* **for all** $x \in \mathbb{R}^n$

$$V(x) = \sum_{R \in \mathcal{R}} \check{p}_R \min_{y \in P_x} \check{c}_R^\top y$$

where $\check{p}_R := \mathbb{P}\big[\boldsymbol{c} \in \mathrm{ri}(R)\big]$ *and* $\check{c}_R := \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in \mathrm{ri}(R)\big]$

Polyhedral characterization of V

Theorem (FGL 2021)

For all distributions of $\boldsymbol{c}$, V is affine on each cell of $\mathcal{C}(P, \pi)$.

Polyhedral characterization of V

Theorem (FGL 2021)

For all distributions of $\boldsymbol{c}$, V is affine on each cell of $\mathcal{C}(P, \pi)$.

Theorem (FGL 2021)

Under an affine change of variable, V is the support function of E

$$V(x) = \sigma_E\big(b - Bx\big) = \sup_{\lambda \in E} (b - Bx)^\top \lambda$$

Polyhedral characterization of V

Theorem (FGL 2021)

For all distributions of $\boldsymbol{c}$, V is affine on each cell of $\mathcal{C}(P, \pi)$.

Theorem (FGL 2021)

Under an affine change of variable, V is the support function of E

$$V(x) = \sigma_E\big(b - Bx\big) = \sup_{\lambda \in E}(b - Bx)^\top \lambda$$

where $E := \mathbb{E}\big[D_{\boldsymbol{c}}\big] = \int D_c \mathbb{P}(dc)$ is the weighted fiber polyhedron and $D_c := \big\{\lambda \,|\, A^\top \lambda + c = 0\big\}$ the dual admissible set.

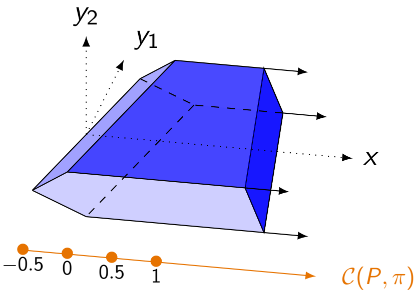
The weighted fiber polyhedron is a Minkowski integral with respect to the distribution $d\mathbb{P}(c)$

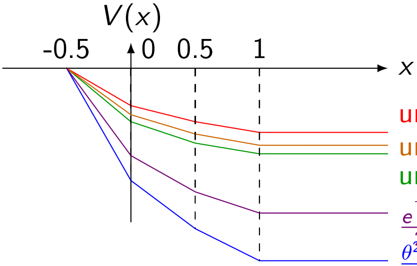
$\rightsquigarrow$ extension of fiber polytope (uniform distribution) of

L. Billera, B. Sturmfels, Fiber polytopes, *Annals of Mathematics*, p527–549, 1992.

Explicit computation of the example

$$V(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{y\in\mathbb{R}^2} & \boldsymbol{c}^\top y \\ \quad\text{s.t.} & \|y\|_1 \leqslant 1 \\ & y_1 \leqslant x \\ & y_2 \leqslant x \end{array}\right]$$





Different distributions of $\boldsymbol{c}$:

uniform on norm 1 ball

uniform on norm 2 ball

uniform on norm ∞ ball

$\frac{e^{-\frac{\|c\|_2^2}{2\gamma^2}}}{2\pi\gamma^2}dc$

$\frac{\theta^2 e^{-\theta\|c\|_1}}{4}dc$

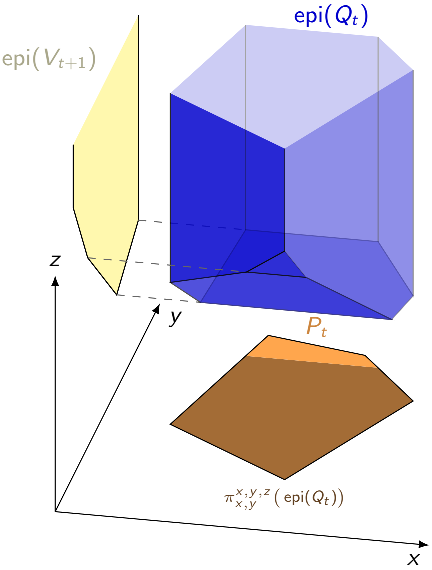
Contents

1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll}\min\limits_{y \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top y + V_{t+1}(y) \\ \\ \text{s.t.} & (x,y) \in P_t\end{array}\right]$$

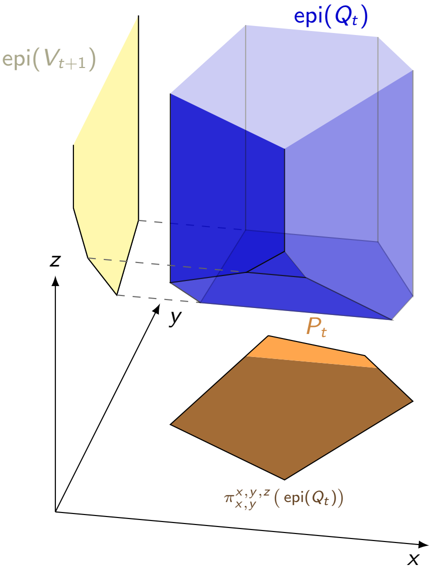
with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y) \in P_t}$.



Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y \in \mathbb{R}^{n_t} \\ z \in \mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ \\ \text{s.t.} & (x, y, z) \in \operatorname{epi}(Q_t) \end{array}\right]$$

with $Q_t(x, y) := V_{t+1}(y) + \mathbb{I}_{(x,y) \in P_t}$.

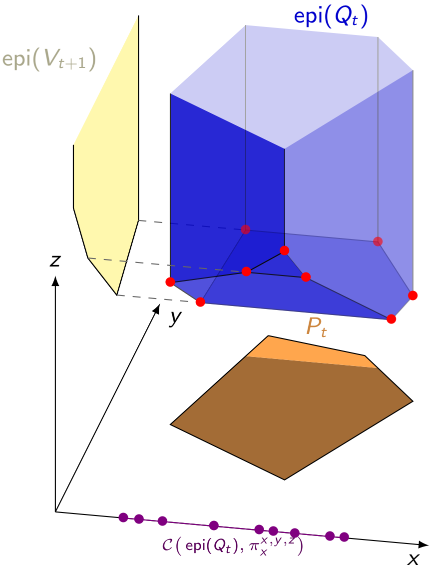


Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ \text{s.t.} & (x,y,z)\in \mathrm{epi}(Q_t) \end{array}\right]$$

with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➥ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$



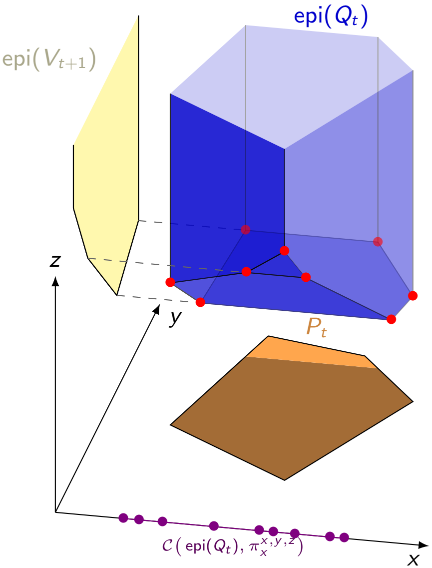
Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ \text{s.t.} & (x,y,z) \in \mathrm{epi}(Q_t) \end{array}\right]$$

with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➥ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

⚠ epi(Q_t) appears in the constraint and depends on $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$!



Multistage uniform and universal exact quantization

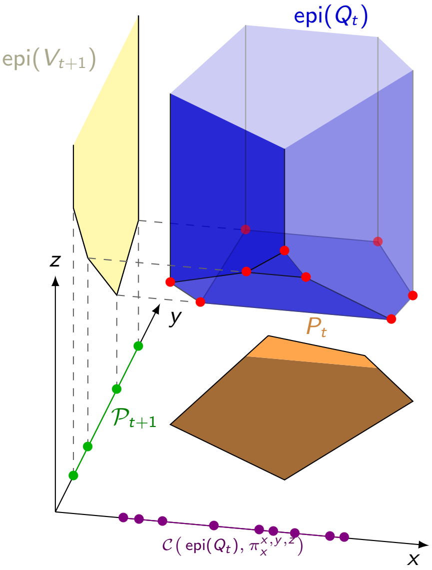
$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ & \\ \text{s.t.} & (x,y,z) \in \text{epi}(Q_t) \end{array}\right]$$

with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➥ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\text{epi}(Q_t), \pi_x^{x,y,z})$

⚠ epi(Q_t) appears in the constraint and depends on $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$!

V_{t+1} affine on $\mathcal{P}_{t+1}$ (by assumption)



Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ & \\ \text{s.t.} & (x,y,z) \in \mathrm{epi}(Q_t) \end{array}\right]$$

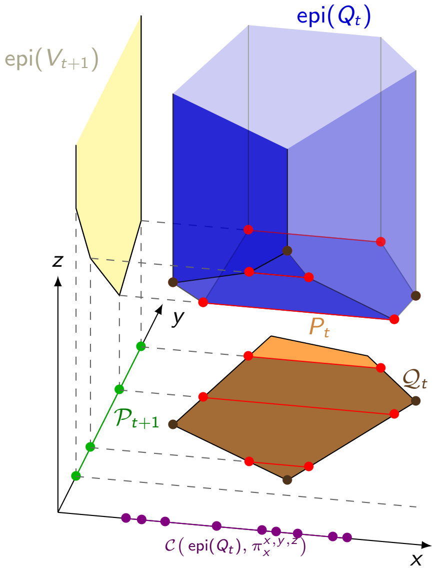
with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➥ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

⚠ epi(Q_t) appears in the constraint and depends on $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$!

V_{t+1} affine on $\mathcal{P}_{t+1}$ (by assumption)

$\mathcal{Q}_t := (\mathbb{R}^{n_t} \times \mathcal{P}_{t+1}) \wedge \mathcal{F}(P_t)$



Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ \text{s.t.} & (x,y,z)\in \mathrm{epi}(Q_t)\end{array}\right]$$

with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

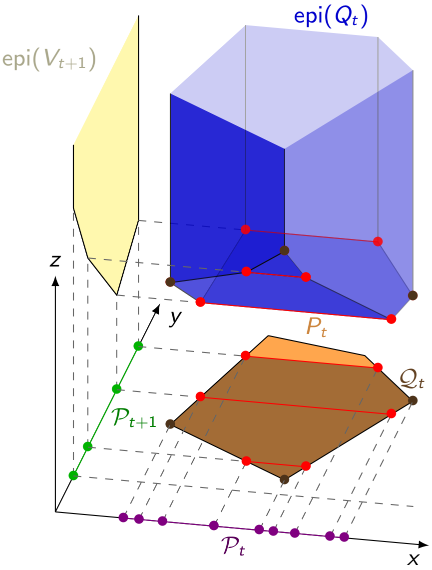
➥ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

⚠ epi(Q_t) appears in the constraint and depends on $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$!

V_{t+1} affine on $\mathcal{P}_{t+1}$ (by assumption)

$\mathcal{Q}_t := (\mathbb{R}^{n_t} \times \mathcal{P}_{t+1}) \wedge \mathcal{F}(P_t)$

$\mathcal{P}_t := \mathcal{C}(\mathcal{Q}_t, \pi_x^{x,y})$



Multistage uniform and universal exact quantization

$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ \text{s.t.} & (x,y,z) \in \mathrm{epi}(Q_t) \end{array}\right]$$

with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➡ V_t affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

⚠ epi(Q_t) appears in the constraint and depends on $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$!

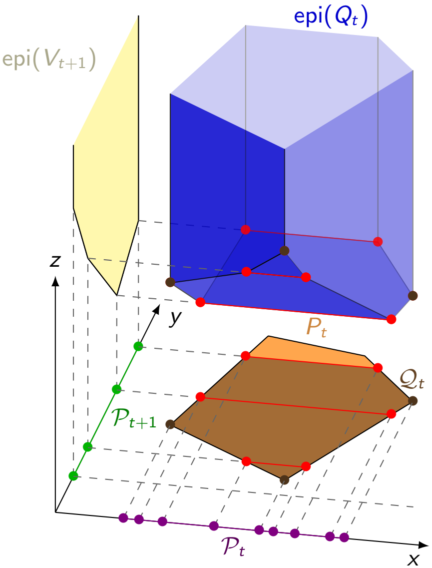
V_{t+1} affine on $\mathcal{P}_{t+1}$ (by assumption)

$\mathcal{Q}_t := (\mathbb{R}^{n_t} \times \mathcal{P}_{t+1}) \wedge \mathcal{F}(P_t)$

$\mathcal{P}_t := \mathcal{C}(\mathcal{Q}_t, \pi_x^{x,y})$

[FGL21, Lem. 4.1]: $\mathcal{P}_t \preccurlyeq \mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

➡ V_t affine on $\mathcal{P}_t$, $\mathcal{N}(P_x)$ constant on $\mathcal{P}_t$



Extension to multistage and stochastic constraints

Iterated chamber complexes by backward induction

$$\mathcal{P}_{t,\xi} := \mathcal{C}\Big((\mathbb{R}^{n_t} \times \mathcal{P}_{t+1}) \wedge \mathcal{F}\big(P_t(\xi)\big), \pi_{x_{t-1}}^{x_{t-1},x_t} \Big)$$

$$\mathcal{P}_t := \bigwedge_{\xi_t \in \operatorname{supp} \boldsymbol{\xi}_t} \mathcal{P}_{t,\xi}$$

Extension to multistage and stochastic constraints

Iterated chamber complexes by backward induction

$$\mathcal{P}_{t,\xi} := \mathcal{C}\Big((\mathbb{R}^{n_t} \times \mathcal{P}_{t+1}) \wedge \mathcal{F}\big(P_t(\xi)\big), \pi_{x_{t-1}}^{x_{t-1},x_t} \Big)$$
$$\mathcal{P}_t := \bigwedge_{\xi_t \in \operatorname{supp} \boldsymbol{\xi}_t} \mathcal{P}_{t,\xi}$$

Theorem (FGL 21)

All results generalizes to MSLP with finitely supported stochastic constraints.

- $(V_t)_t$ *are affine on universal chamber complexes,* *i.e. independent of the law of* $(\boldsymbol{c}_t)_t$
- *We have an uniform and universal exact quantization.*

Contents

1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

Earlier and new complexity results

Volume of a polytope

$$\mathrm{Vol}\left(\{z \in \mathbb{R}^d \mid Az \leqslant b\}\right) \text{ or}$$
$$\mathrm{Vol}\left(\,\mathrm{Conv}(v_1, \cdots, v_n)\right)$$

- $\sharp P$-complete:
  Dyer and Frieze (1988)
- Polynomial for fixed dimension
  d: Lawrence (1991)

Earlier and new complexity results

Volume of a polytope

$$\mathrm{Vol}\left(\{z \in \mathbb{R}^d \mid Az \leqslant b\}\right) \text{ or}$$
$$\mathrm{Vol}\left(\mathrm{Conv}(v_1, \cdots, v_n)\right)$$

- $\sharp P$-complete: Dyer and Frieze (1988)
- Polynomial for fixed dimension d: Lawrence (1991)

2-stage linear problem

$$\min_{x \in \mathbb{R}^n} c^\top x + \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^m} \boldsymbol{q}^\top y \\ \text{s.t. } Tx + Wy \leqslant h \end{bmatrix}$$
$$\text{s.t. } Ax \leqslant b$$

- $\sharp P$-hard: Hanasusanto, Kuhn and Wiesemann (2016)
- Polynomial for fixed m ?

Earlier and new complexity results

Volume of a polytope

$$\mathrm{Vol}\left(\{z \in \mathbb{R}^d \mid Az \leqslant b\}\right) \text{ or}$$
$$\mathrm{Vol}\left(\mathrm{Conv}(v_1, \cdots, v_n)\right)$$

- $\sharp P$-complete: Dyer and Frieze (1988)
- Polynomial for fixed dimension d: Lawrence (1991)

2-stage linear problem

$$\min_{x \in \mathbb{R}^n} c^\top x + \mathbb{E}\left[\begin{array}{l} \min_{y \in \mathbb{R}^{m}} \boldsymbol{q}^\top y \\ \text{s.t. } Tx + Wy \leqslant h \end{array}\right]$$
$$\text{s.t. } Ax \leqslant b$$

- $\sharp P$-hard: Hanasusanto, Kuhn and Wiesemann (2016)
- Polynomial for fixed m: FGL (2021)

Earlier and new complexity results

Volume of a polytope

$$\mathrm{Vol}\left(\{z \in \mathbb{R}^d \mid Az \leqslant b\}\right) \text{ or}$$
$$\mathrm{Vol}\left(\mathrm{Conv}(v_1, \cdots, v_n)\right)$$

- $\sharp P$-complete: Dyer and Frieze (1988)
- Polynomial for fixed dimension d: Lawrence (1991)

2-stage linear problem

$$\min_{x \in \mathbb{R}^n} c^\top x + \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^{m}} \boldsymbol{q}^\top y \\ \text{s.t. } \; Tx + Wy \leqslant h \end{bmatrix}$$
$$\text{s.t. } \; Ax \leqslant b$$

- $\sharp P$-hard: Hanasusanto, Kuhn and Wiesemann (2016)
- Polynomial for fixed m: FGL (2021)
  $\rightsquigarrow$ Exact case
  $\rightsquigarrow$ Approximated case

Complexity result multistage

Theorem (FGL21: MSLP is polynomial for fixed dimensions)

Assume that T, $n_2, \cdots, n_T$, are fixed.[1]
Assume that $\boldsymbol{c}$ admits a density function with a bounded total variation.

Then, there exists an algorithm that finds an ε-solution[2] *in polynomial time in $\log(\frac{1}{\varepsilon})$ with probability 1.*

[1] No requirement for the first decision.
[2] Or asserts that MSLP is unfeasible.

Complexity result multistage

Theorem (FGL21: MSLP is polynomial for fixed dimensions)

Assume that T, $n_2, \cdots, n_T$, are fixed.[1]
Assume that $\boldsymbol{c}$ admits a density function with a bounded total variation.

Then, there exists an algorithm that finds an ε-solution[2] in polynomial time in $\log(\frac{1}{\varepsilon})$ with probability 1.

➥ Can be adapted to exact complexity when we can compute exactly $\mathbb{E}\big[\boldsymbol{c}|\boldsymbol{c} \in C, (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$ and $\mathbb{P}\big[\boldsymbol{c} \in C|(\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$.

[1] No requirement for the first decision.
[2] Or asserts that MSLP is unfeasible.

Complexity result multistage

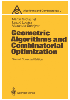
Theorem (FGL21: MSLP is polynomial for fixed dimensions)

Assume that T, $n_2, \cdots, n_T$, are fixed.[1]
Assume that $\boldsymbol{c}$ admits a density function with a bounded total variation.

Then, there exists an algorithm that finds an ε-solution[2] *in polynomial time in $\log(\frac{1}{\varepsilon})$ with probability 1.*

➥ Can be adapted to exact complexity when we can compute exactly $\mathbb{E}\big[\boldsymbol{c}|\boldsymbol{c} \in C, (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$ and $\mathbb{P}\big[\boldsymbol{c} \in C|(\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$.

Proof based on ellipsoid (Gröstchel, Lovász, Schrijver)
and upper bound theorems (McMullen, Stanley)



[1] No requirement for the first decision.
[2] Or asserts that MSLP is unfeasible.

Complexity result multistage

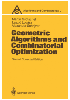
Theorem (FGL21: MSLP is polynomial for fixed dimensions)

Assume that T, $n_2, \cdots, n_T$, are fixed.[1]
Assume that $\boldsymbol{c}$ admits a density function with a bounded total variation.

Then, there exists an algorithm that finds an ε-solution[2] in polynomial time in $\log(\frac{1}{\varepsilon})$ with probability 1.

➥ Can be adapted to exact complexity when we can compute exactly $\mathbb{E}\big[\boldsymbol{c}|\boldsymbol{c} \in C, (\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$ and $\mathbb{P}\big[\boldsymbol{c} \in C|(\boldsymbol{A}_t, \boldsymbol{B}_t, \boldsymbol{b}_t) = (A, B, b)\big]$.

Proof based on ellipsoid (Gröstchel, Lovász, Schrijver) and upper bound theorems (McMullen, Stanley)



By SAA, we can solve MSLP, up to precision ε, in pseudo-polynomial time, i.e. polynomial in $\frac{1}{\varepsilon}$, with probability $1 - \alpha$, when T, $n_1, \cdots, n_T$ are fixed.

[1] No requirement for the first decision.
[2] Or asserts that MSLP is unfeasible.

Contents

1 Universal Exact Quantization for cost
- Local in 2-stage
- Uniform in 2-stage
- Uniform in multistage
- Complexity results

2 Local and universal exact Quantization for constraints
- Adapted partitions
- Adaptive Partition-based Methods
- Convergence, complexity and numerical results

3 Trajectory Following Dynamic Programming

4 Conclusion and perspectives

Local exact quantization for constraints ?

Back to the 2-stage problem

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	?	✓
Uniform	×	×	✓

Duality result

$$V(x) = \mathbb{E}\big[V(x,\xi)\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^n} & c^\top y \\ \text{s.t.} & Ay + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix} = \mathbb{E}\begin{bmatrix} \max\limits_{\lambda \in \mathbb{R}^\ell} & (\boldsymbol{B}x - \boldsymbol{b})^\top \lambda \\ \text{s.t.} & A^\top \lambda + c = 0 \end{bmatrix}$$

➥ Back to the case with random cost

⚠ The new cost depends on x: only local exact quantization.

Local exact quantization for constraints ?

Back to the 2-stage problem

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	✓
Uniform	$\times$	$\times$	✓

Duality result

$$V(x) = \mathbb{E}\big[V(x, \boldsymbol{\xi})\big] = \mathbb{E}\begin{bmatrix} \min_{y \in \mathbb{R}^n} & c^\top y \\ \text{s.t.} & Ay + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix} = \mathbb{E}\begin{bmatrix} \max_{\lambda \in \mathbb{R}^\ell} & (\boldsymbol{B}x - \boldsymbol{b})^\top \lambda \\ \text{s.t.} & A^\top \lambda + c = 0 \end{bmatrix}$$

➥ Back to the case with random cost

⚠ The new cost depends on x: only local exact quantization.

Local exact quantization for constraints ?

Back to the 2-stage problem

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	?	✓
Uniform	×	×	✓

Duality result

$$V(x) = \mathbb{E}\big[V(x, \boldsymbol{\xi})\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^n} & c^\top y \\ \text{s.t.} & Ay + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix} = \mathbb{E}\begin{bmatrix} \max\limits_{\lambda \in \mathbb{R}^\ell} & (\boldsymbol{B}x - \boldsymbol{b})^\top \lambda \\ \text{s.t.} & A^\top \lambda + c = 0 \end{bmatrix}$$

➥ Back to the case with random cost

⚠ The new cost depends on x: only local exact quantization.

Local exact quantization for constraints ?

Back to the 2-stage problem

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	?	$\checkmark$
Uniform	$\times$	$\times$	$\checkmark$

Duality result

$$V(x) = \mathbb{E}\big[V(x, \boldsymbol{\xi})\big] = \mathbb{E}\begin{bmatrix} \min\limits_{y \in \mathbb{R}^n} & c^\top y \\ \text{s.t.} & Ay + \boldsymbol{B}x \leqslant \boldsymbol{b} \end{bmatrix} = \mathbb{E}\begin{bmatrix} \max\limits_{\lambda \in \mathbb{R}^\ell} & (\boldsymbol{B}x - \boldsymbol{b})^\top \lambda \\ \text{s.t.} & A^\top \lambda + c = 0 \end{bmatrix}$$

➥ Back to the case with random cost

⚠ The new cost depends on x: only local exact quantization.

Local exact quantization for constraints

random cost

Recall that for a fixed x,

$$V(x) = \mathbb{E}\Big[\min_{y \in P_x} \boldsymbol{c}^\top y\Big] = \sum_{N \in \mathcal{N}(P_x)} p_N \min_{y \in P_x} \check{c}_N{}^\top y$$

where,

$$p_N := \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big]$$
$$\check{c}_N := \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]$$

$$P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

random constraints

Similarly, for a given c and x,

$$V(x) = \mathbb{E}\Big[\max_{\lambda \in D_c} (\boldsymbol{b} - \boldsymbol{B}x)^\top \lambda\Big] = \sum_{N \in \mathcal{N}(D_c)} p_{N,x} \max_{\lambda \in D_c} \psi_{N,x}{}^\top \lambda$$

where,

$$p_{N,x} := \mathbb{P}\big[\boldsymbol{b} - \boldsymbol{B}x \in \operatorname{ri} N\big]$$
$$\psi_{N,x} := \mathbb{E}\big[\boldsymbol{b} - \boldsymbol{B}x \mid \boldsymbol{b} - \boldsymbol{B}x \in \operatorname{ri} N\big]$$

$$D_c := \{\lambda \in \mathbb{R}^l \mid A^\top \lambda + c = 0\}$$

Local exact quantization for constraints

random cost

Recall that for a fixed x,

$$V(x) = \mathbb{E}\Big[\min_{y \in P_x} \boldsymbol{c}^\top y\Big] = \sum_{N \in \mathcal{N}(P_x)} p_N \min_{y \in P_x} \check{c}_N{}^\top y$$

where,

$$p_N := \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big]$$
$$\check{c}_N := \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]$$

$$P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

random constraints

Similarly, for a given c and x,

$$V(x) = \mathbb{E}\Big[\max_{\lambda \in D_c} (\boldsymbol{b} - \boldsymbol{B}x)^\top \lambda\Big] = \sum_{N \in \mathcal{N}(D_c)} p_{N,x} \max_{\lambda \in D_c} \psi_{N,x}{}^\top \lambda$$

where,

$$p_{N,x} := \mathbb{P}\big[\boldsymbol{b} - \boldsymbol{B}x \in \operatorname{ri} N\big]$$
$$\psi_{N,x} := \mathbb{E}\big[\boldsymbol{b} - \boldsymbol{B}x \mid \boldsymbol{b} - \boldsymbol{B}x \in \operatorname{ri} N\big]$$

$$D_c := \{\lambda \in \mathbb{R}^l \mid A^\top \lambda + c = 0\}$$

Contents

1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

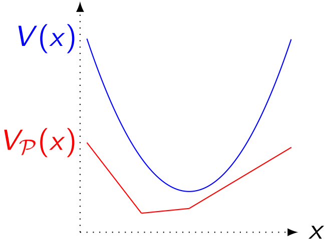
Partitioned cost-to-go functions

Recall that

$$V(x) = \quad \mathbb{E}\Big[\hat{V}(x, \boldsymbol{\xi})\Big]$$

$$V_{\mathcal{P}}(x) = \quad \sum_{P \in \mathcal{P}} \mathbb{P}\big[P\big]\, \hat{V}\big(x, \mathbb{E}\big[\boldsymbol{\xi} | P\big]\big)$$

- $\hat{V}(x, \cdot)$ is convex $\rightsquigarrow$ $V_{\mathcal{P}} \leqslant V$.
- $\hat{V}\big(\cdot, \mathbb{E}\big[\boldsymbol{\xi} | P\big]\big)$ is polyhedral $\rightsquigarrow$ $V_{\mathcal{P}}$ is polyhedral.



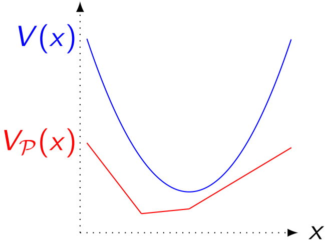
Partitioned cost-to-go functions

Recall that

$$V(x) = \quad \mathbb{E}\left[\hat{V}(x, \boldsymbol{\xi})\right]$$

$$V_{\mathcal{P}}(x) = \quad \sum_{P \in \mathcal{P}} \mathbb{P}\left[P\right] \hat{V}\left(x, \mathbb{E}\left[\boldsymbol{\xi} | P\right]\right)$$

- $\hat{V}(x, \cdot)$ is convex $\rightsquigarrow$ $V_{\mathcal{P}} \leqslant V$.
- $\hat{V}\left(\cdot, \mathbb{E}\left[\boldsymbol{\xi} | P\right]\right)$ is polyhedral $\rightsquigarrow$ $V_{\mathcal{P}}$ is polyhedral.

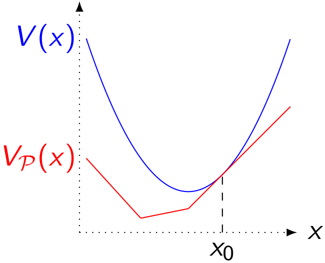


Adapted partition

Definition

We say that a partition $\mathcal{P}$ is adapted to x_0 if

$$V_{\mathcal{P}}(x_0) = V(x_0) := \mathbb{E}\big[Q(x_0, \boldsymbol{\xi})\big]$$

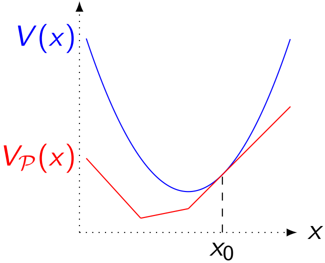


Adapted partition

Definition

We say that a partition $\mathcal{P}$ is adapted to x_0 if

$$V_{\mathcal{P}}(x_0) = V(x_0) := \mathbb{E}\big[Q(x_0, \boldsymbol{\xi})\big]$$



Consider $x \in \mathbb{R}^n$ and $N \in \mathcal{N}(D_q)$ a normal cone of D_q. We define

$$E_{N,x} := \{\xi \in \Xi \mid b - Bx \in \operatorname{ri} N\}$$

Theorem (FL 2021)

$\mathcal{R}_x := \big\{E_{N,x} \mid N \in \mathcal{N}(D_q)\big\}$ *is adapted to x i.e.* $V_{\mathcal{R}_x}(x) = V(x)$
In particular: if only $\boldsymbol{B}$ *and* $\boldsymbol{b}$ *are stochastic,*
then there exists a universal and local exact quantization.
Bonus: necessary and sufficient condition for a partition to be adapted

Contents

1 Universal Exact Quantization for cost
- Local in 2-stage
- Uniform in 2-stage
- Uniform in multistage
- Complexity results

2 Local and universal exact Quantization for constraints
- Adapted partitions
- Adaptive Partition-based Methods
- Convergence, complexity and numerical results

3 Trajectory Following Dynamic Programming

4 Conclusion and perspectives

General framework for Adaptive Partition-based Methods

$\mathcal{P}^0 \leftarrow \{\Xi\}$;
for $k = 1 \cdots \infty$ **do**
  Let x^k be an optimal solution $\min_{x \in X} c^\top x + V_{\mathcal{P}^{k-1}}(x)$;
  Let $\mathcal{P}_{x^k}$ a partition adapted to x^k ;
  $\mathcal{P}^k \leftarrow \mathcal{P}^{k-1} \wedge \mathcal{P}_{x^k}$;
end

$$\min_{x \in X} c^\top x + V_{\mathcal{P}}(x)$$

is equivalent to

$$\min_{x \in X, (y_P)_{P \in \mathcal{P}}} \quad c^\top x + \sum_{P \in \mathcal{P}} \mathbb{P}[P] q^\top y_P$$

$$\mathbb{E}\big[\boldsymbol{B} | P\big] x + A y_P \leqslant \mathbb{E}\big[\boldsymbol{b} | P\big] \qquad \forall P \in \mathcal{P}$$

General framework for Adaptive Partition-based Methods

$\mathcal{P}^0 \leftarrow \{\Xi\}$;
for $k = 1 \cdots \infty$ **do**
  Let x^k be an optimal solution $\min_{x \in X} c^\top x + V_{\mathcal{P}^{k-1}}(x)$;
  Let $\mathcal{P}_{x^k}$ a partition adapted to x^k ;
  $\mathcal{P}^k \leftarrow \mathcal{P}^{k-1} \wedge \mathcal{P}_{x^k}$;
end

$$\min_{x \in X} c^\top x + V_{\mathcal{P}}(x)$$

is equivalent to

$$\begin{aligned} \min_{x \in X, (y_P)_{P \in \mathcal{P}}} \quad & c^\top x + \sum_{P \in \mathcal{P}} \mathbb{P}\big[P\big] q^\top y_P \\ & \mathbb{E}\big[\boldsymbol{B} \big| P\big] x + A y_P \leqslant \mathbb{E}\big[\boldsymbol{b} \big| P\big] \qquad \forall P \in \mathcal{P} \end{aligned}$$

A (partial) comparison between partition based results

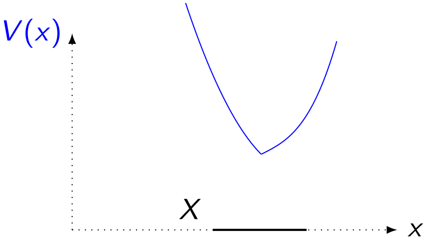
Paper	Song, Luedtke (2015)	Ramirez-Pico, Moreno (2020)	FL (2021)
Non-finite supp($\boldsymbol{\xi}$)	×	✓	✓
Explicit oracle	✓	×	✓
Proof of convergence	✓	×	✓
Complexity result	×	×	✓
Fast iteration	✓	×	×

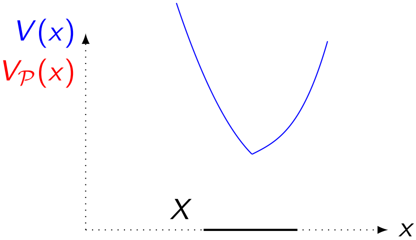
Contents

1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

Link with Benders decomposition and L-shaped

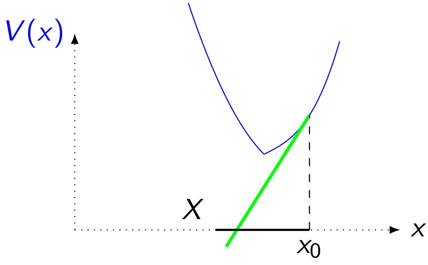
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.





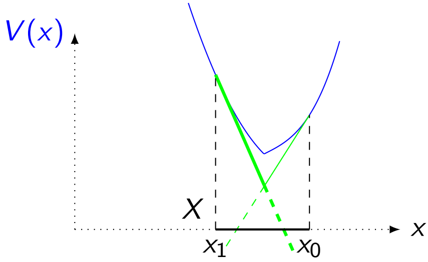
Link with Benders decomposition and L-shaped

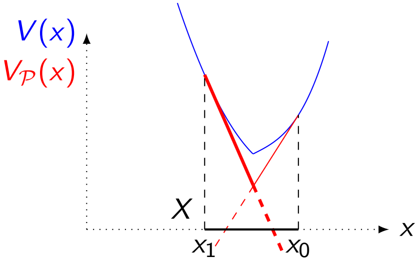
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.



Link with Benders decomposition and L-shaped

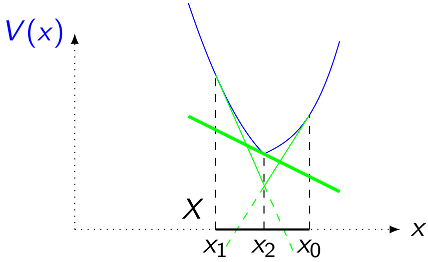
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

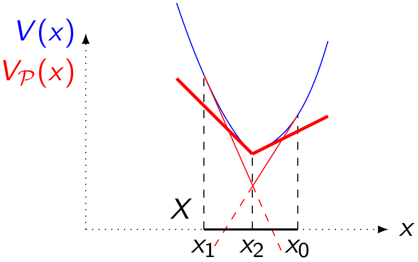




Link with Benders decomposition and L-shaped

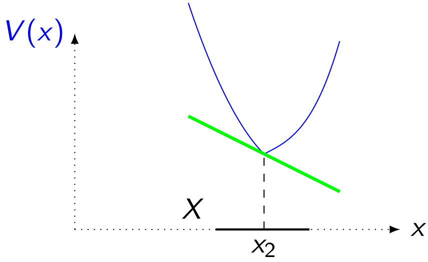
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

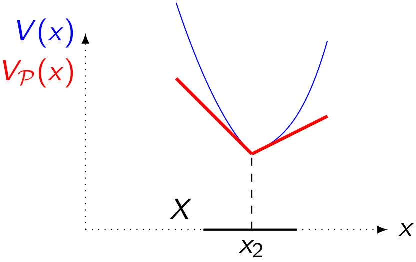




Link with Benders decomposition and L-shaped

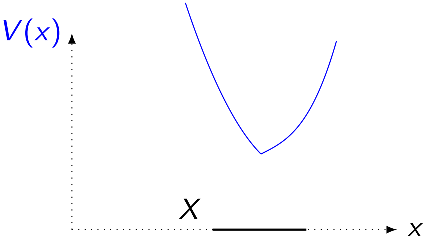
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.





Link with Benders decomposition and L-shaped

Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.



Theorem (Convergence and complexity results)

If $X \cap \mathrm{dom}(V) \subset \mathbb{R}^+$ is contained in a ball of diameter $M \in \mathbb{R}^+$ and $x \to c^\top x + V(x)$ is Lipschitz with constant L then the partition based method finds an ε-solution in at most $\left(\frac{LM}{\varepsilon} + 1\right)^n$ iterations.

Numerical Results - ProdMix

k	x_k	z_L^k	z_U^k	Gap	$\lvert\mathcal{P}_k^{\max}\rvert$
1	$(1333.33, 66.67)$	-18666.67	-16939.71	9.3%	4
2	$(1441.41, 59.57)$	-17873.01	-17383.73	2.7%	9
3	$(1399.05, 57.91)$	-17789.88	-17659.19	0.74%	16
4	$(1379.98, 56.64)$	-17744.67	-17708.00	0.20%	25
5	$(1371.36, 55.71)$	-17718.96	-17709.05	0.056%	36
6	$(1375.55, 56.21)$	-17713.74	-17711.37	0.013%	49

Table: Results for problem Prod-Mix

To compare our approach with SAA, we solved the same problem 100 times, each with 10 000 scenarios randomly drawn, yielding a 95% confidence interval centered in -17711, with radius 2.2.

Contents

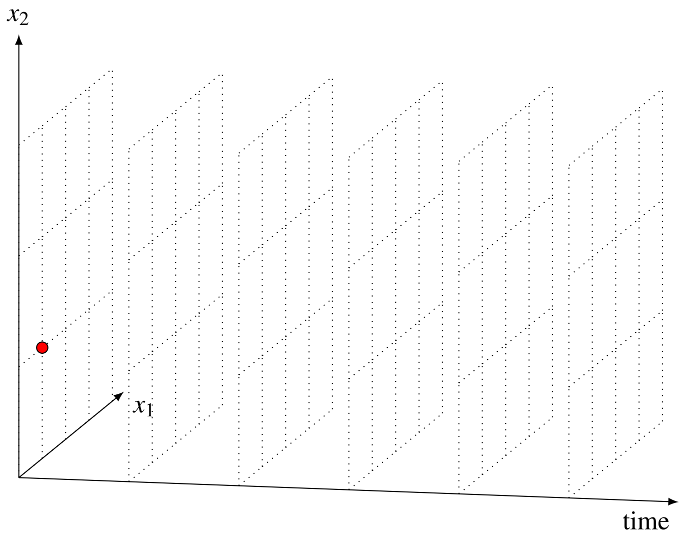
1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

History of stochastic dual dynamic programming (SDDP)

- Designed by Pereira and Pinto in 1991, used to manage brazilian hydroelectricity network
- Proof of asymptotic convergence in the linear case (Philpott and Guan 2008) and in the convex case (Girardeau, Leclère, Philpott 2015)
- Complexity proof (Lan 2020, Zhang and Sun 2022)
- Plenty of variants: trajectory following dynamic programming algorithms

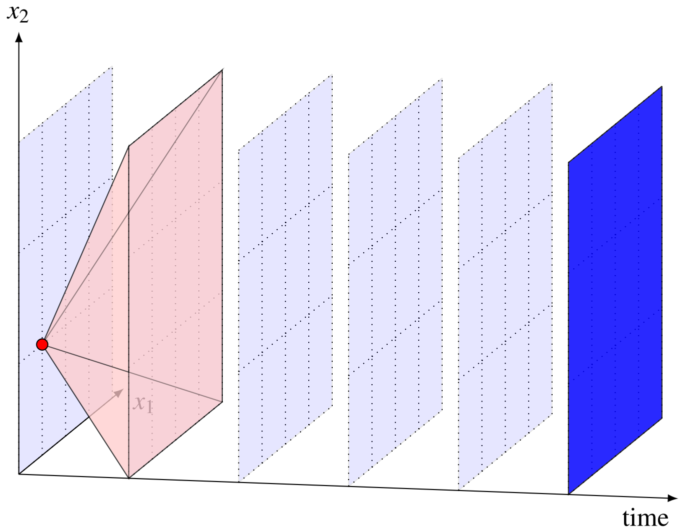
➥ All with finitely supported distribution

Trajectory Following Dynamic Programming



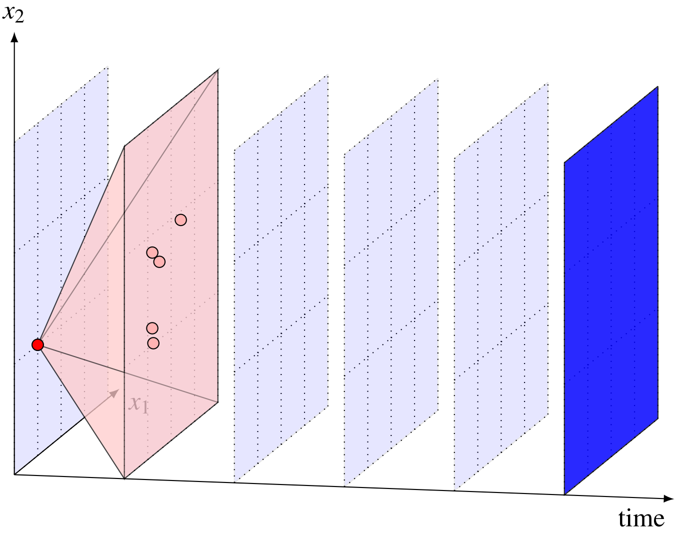
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



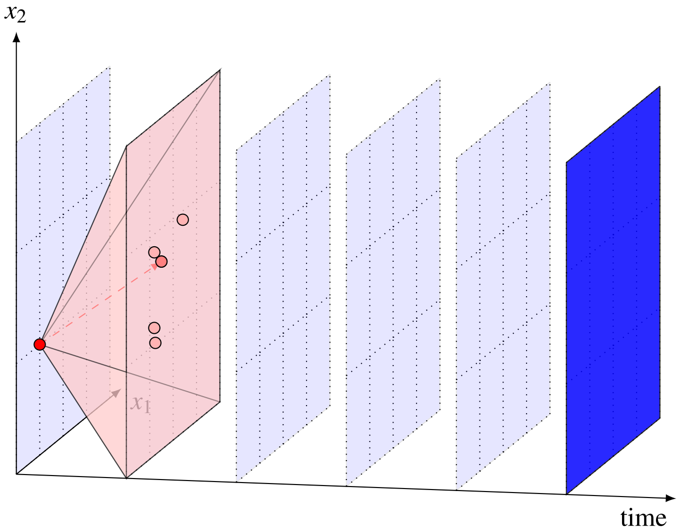
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



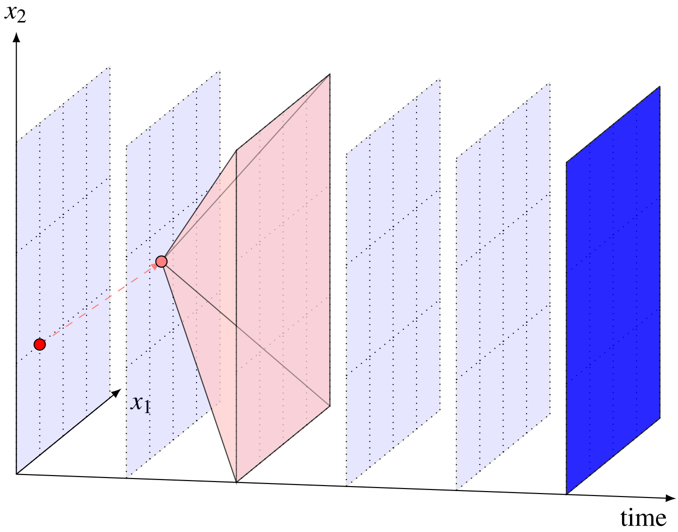
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



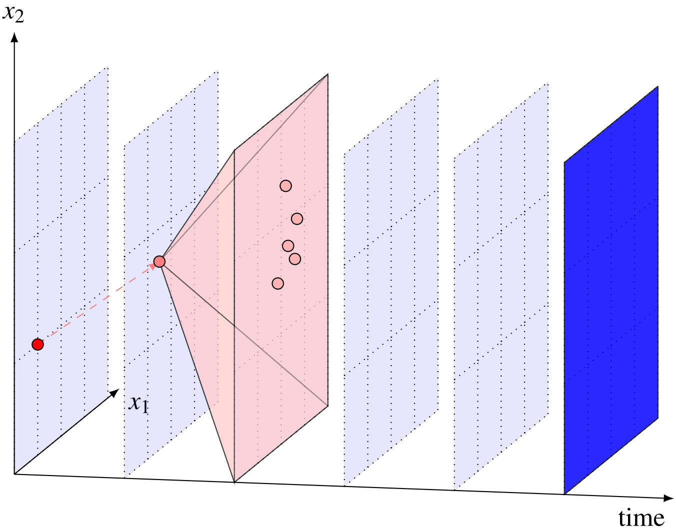
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



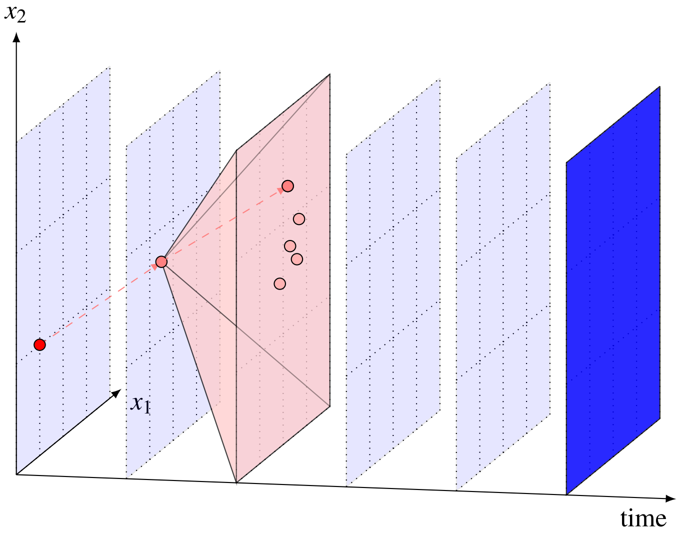
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



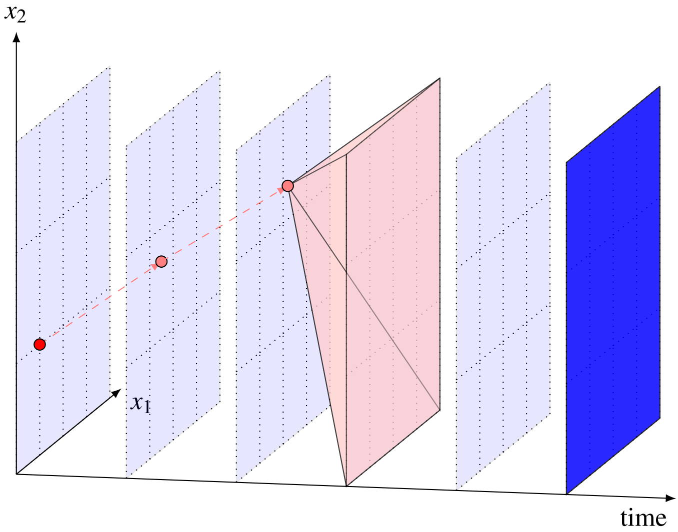
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



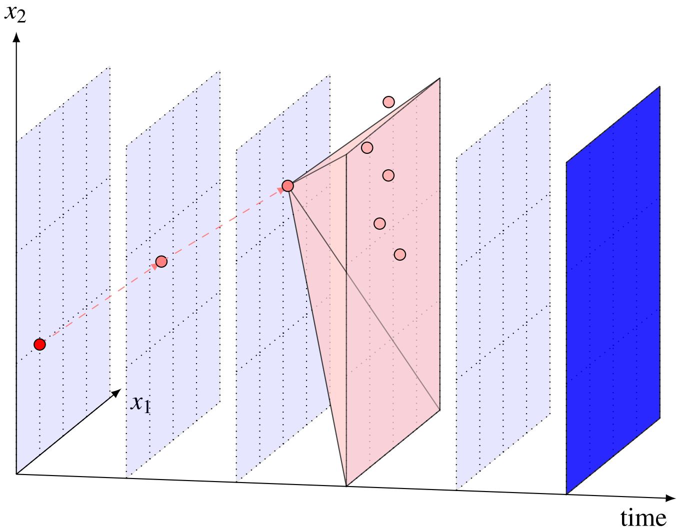
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



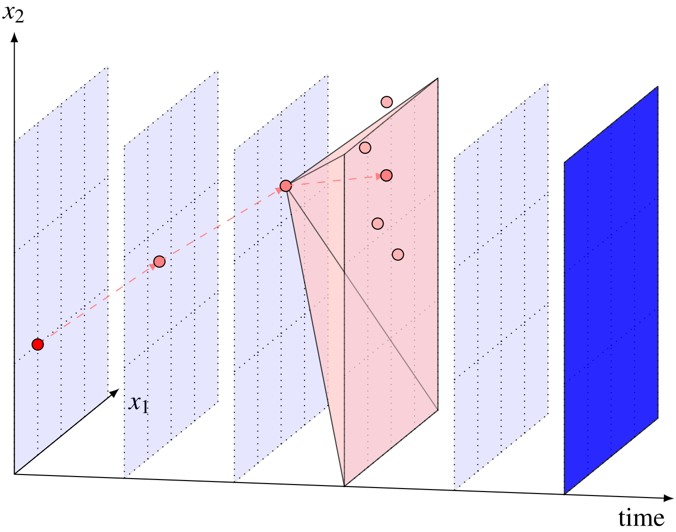
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



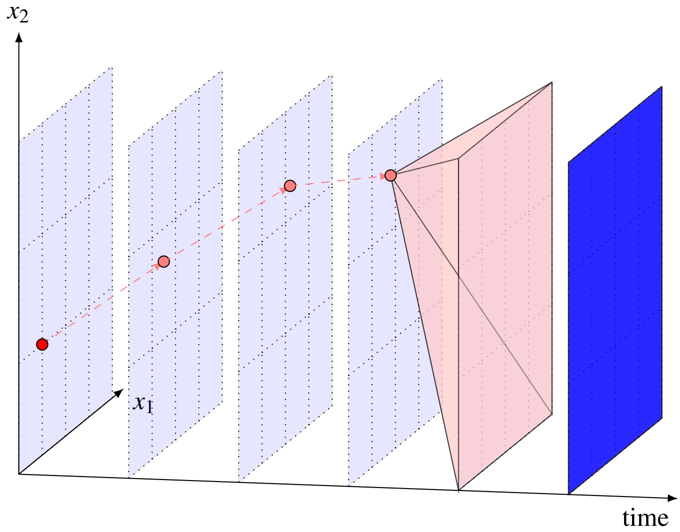
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



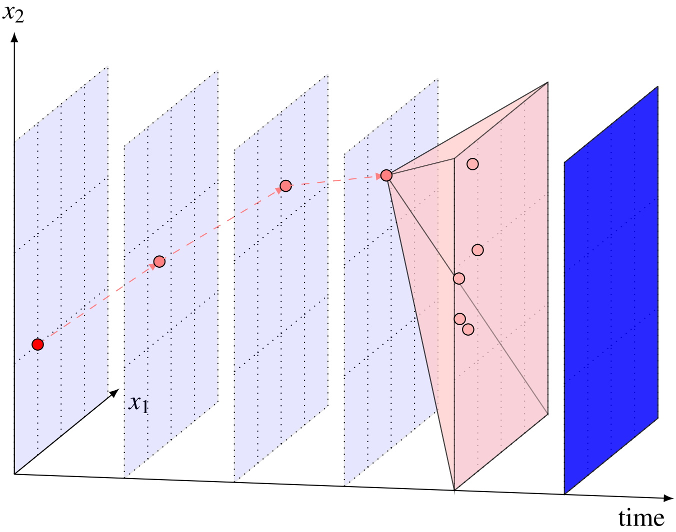
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



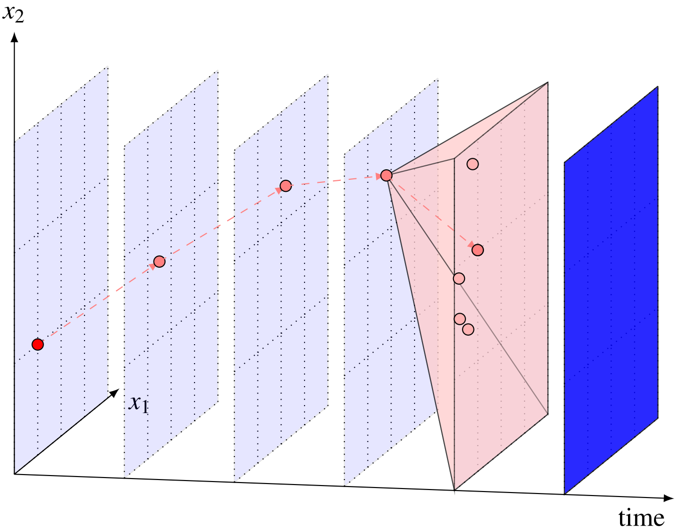
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



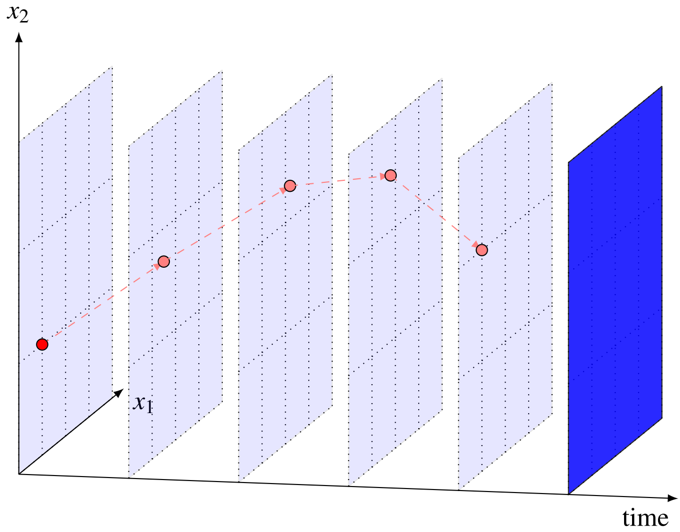
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



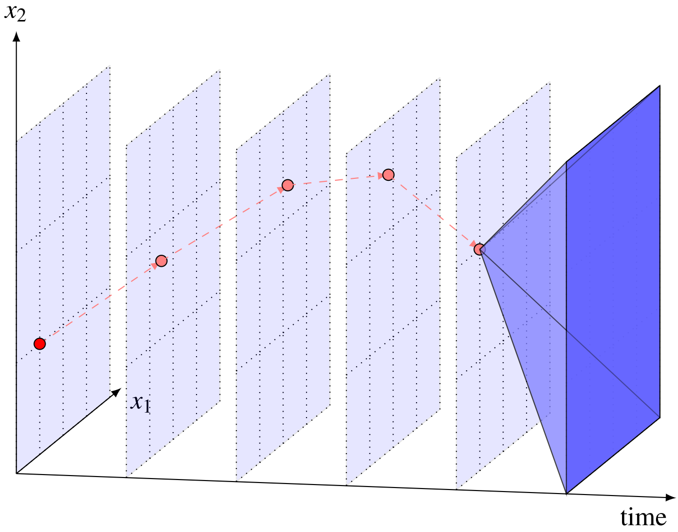
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



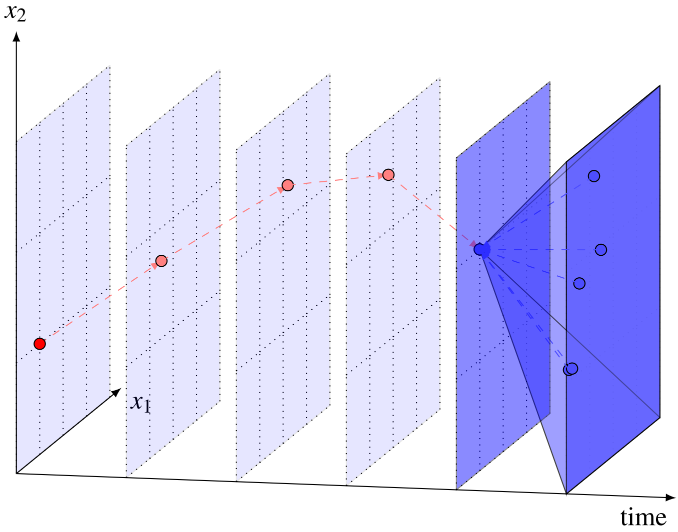
First forward pass : computing trajectory

Trajectory Following Dynamic Programming



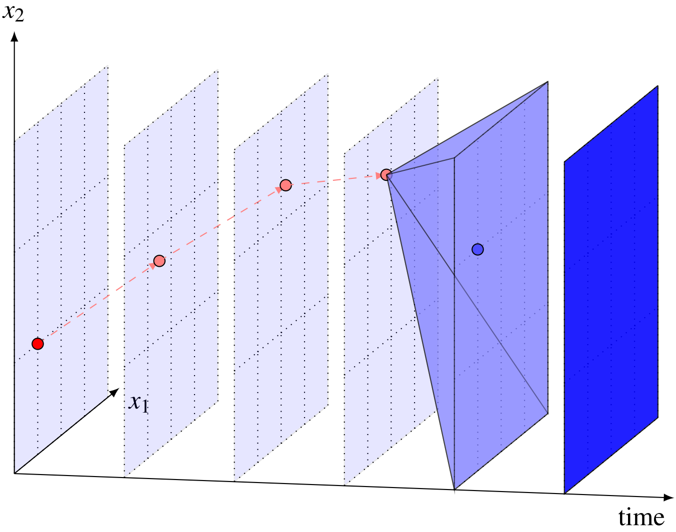
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



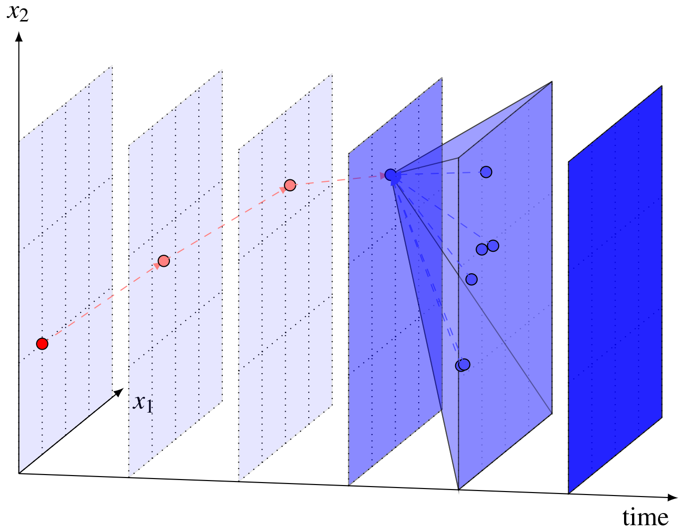
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



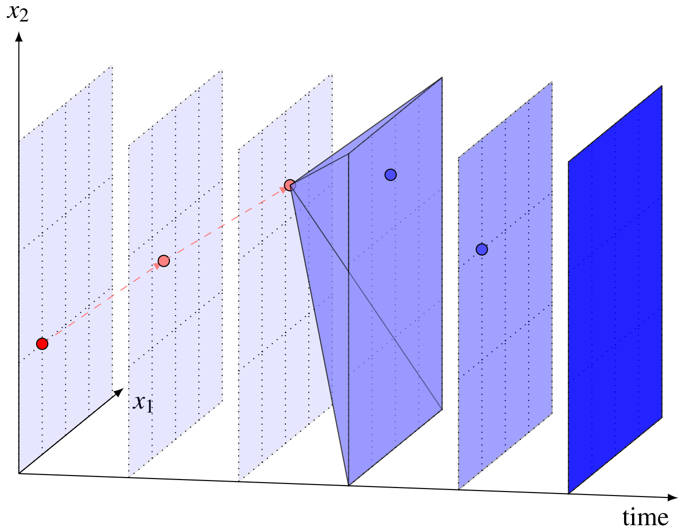
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



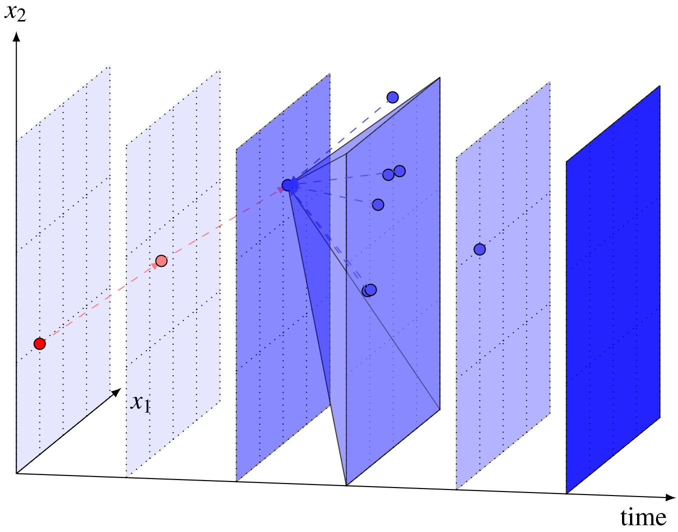
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



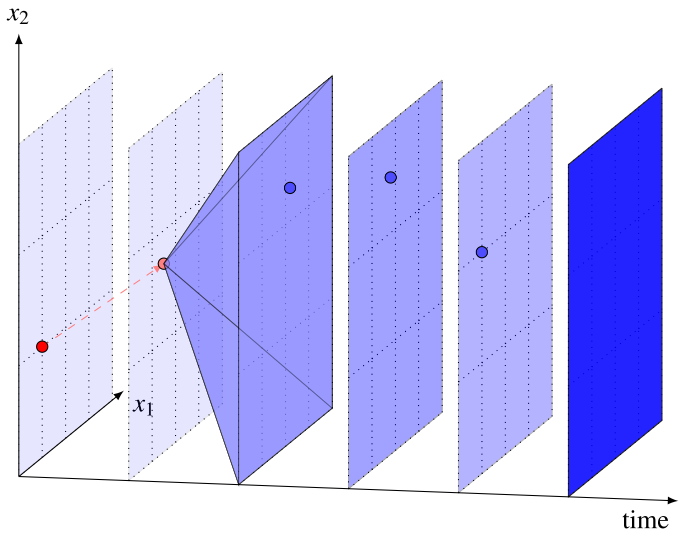
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



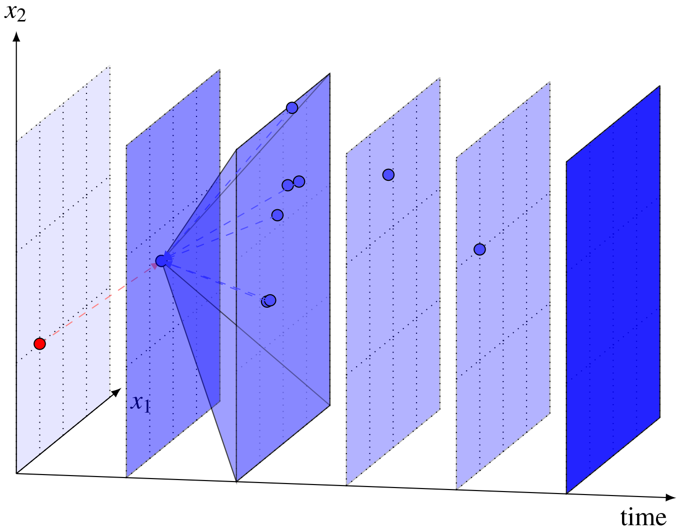
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



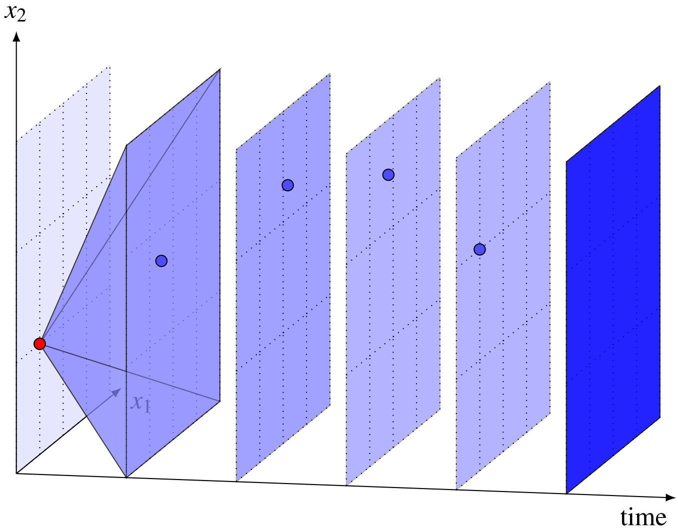
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



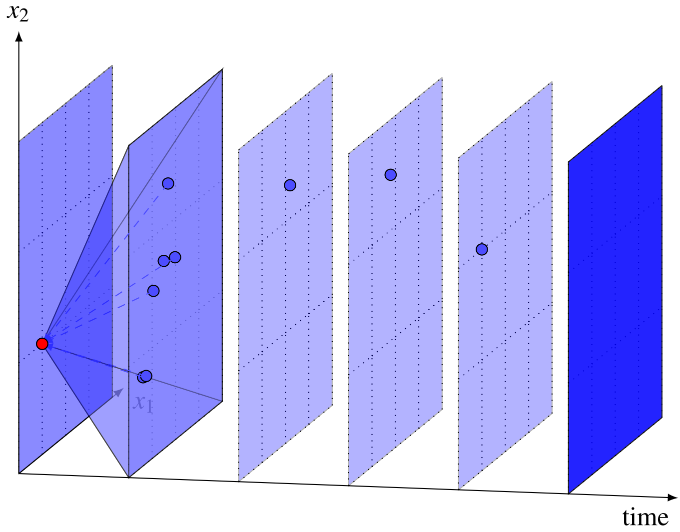
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



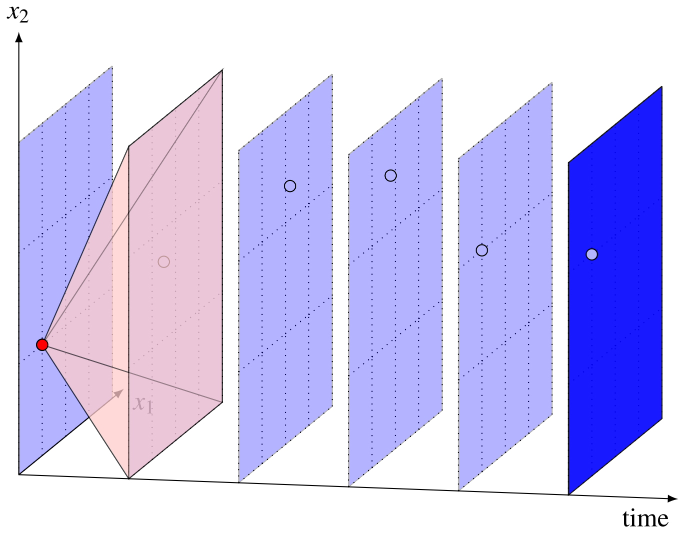
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



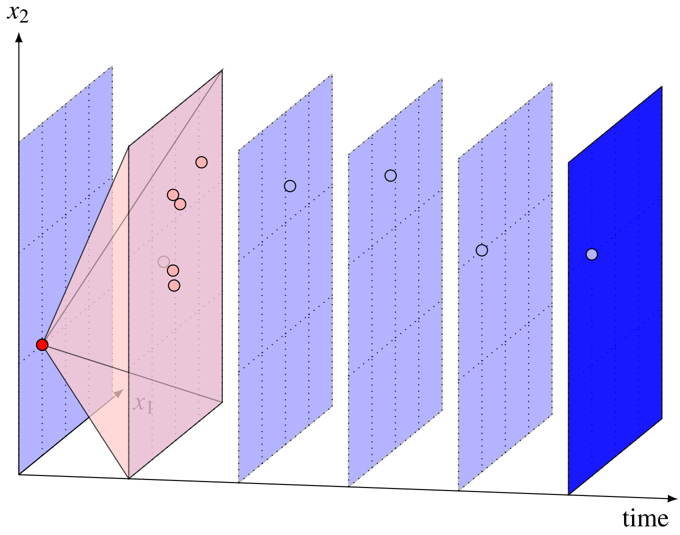
First backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



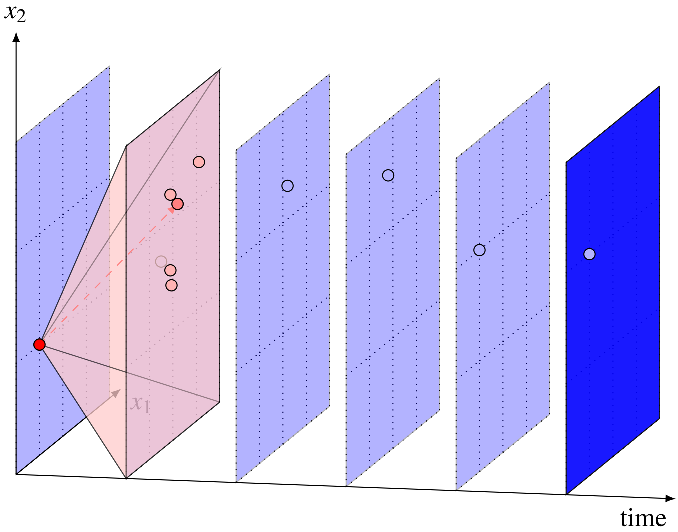
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



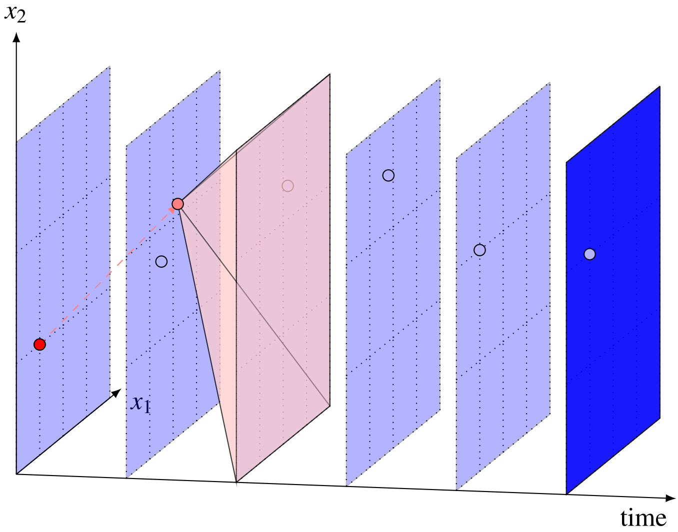
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



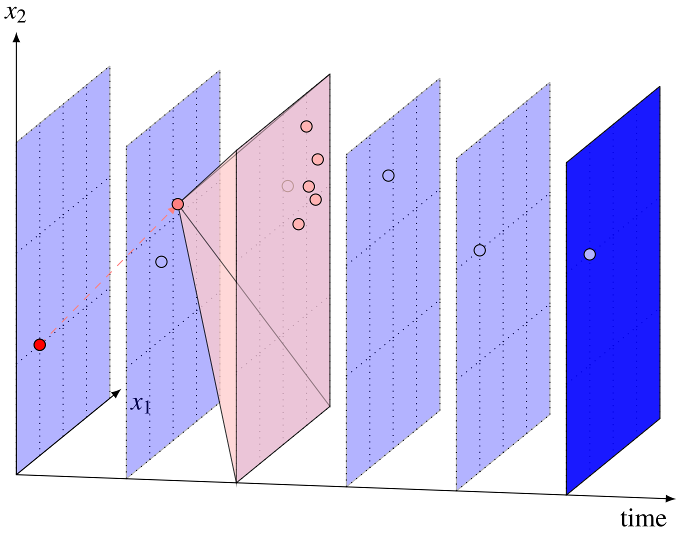
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



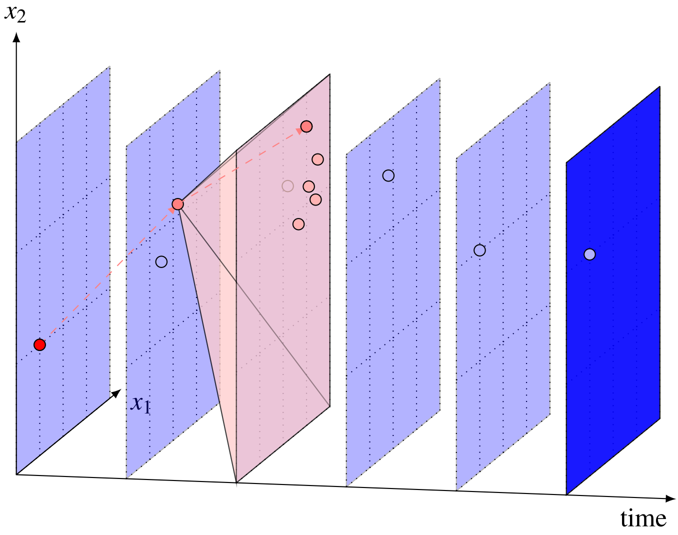
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



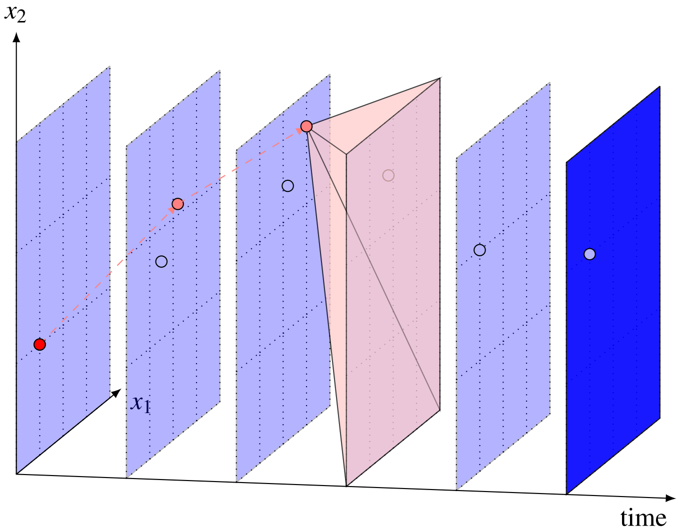
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



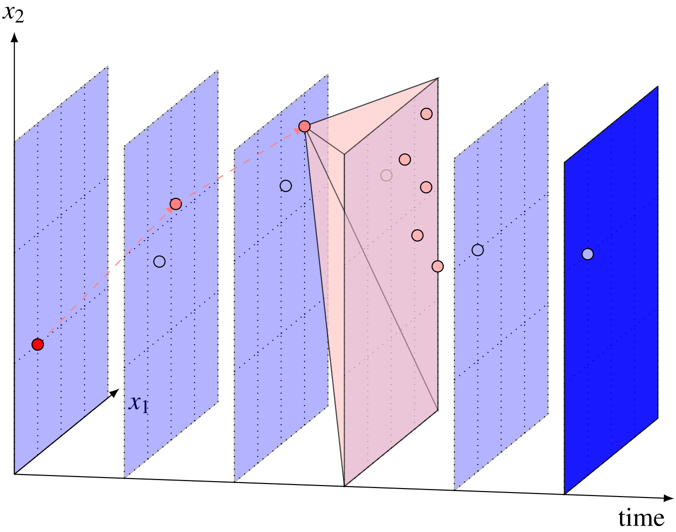
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



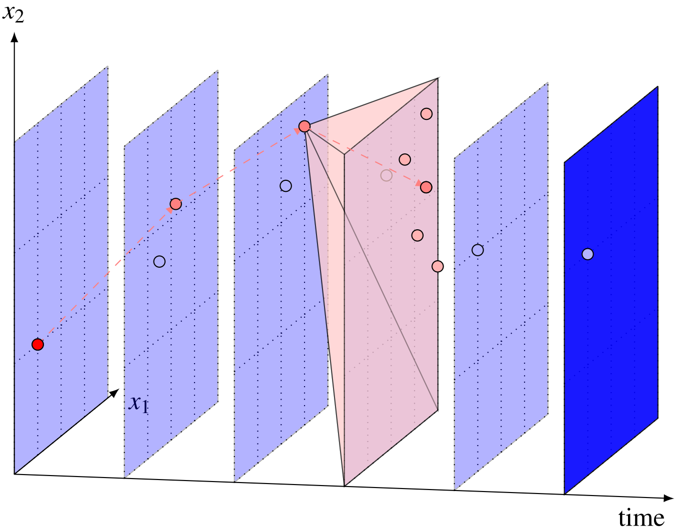
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



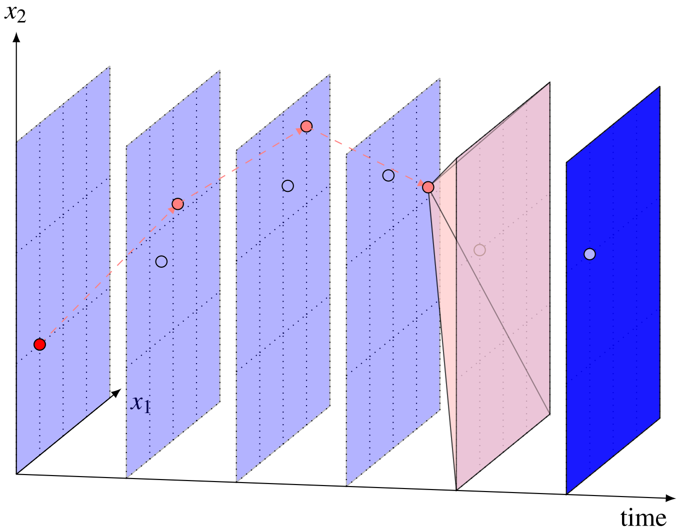
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



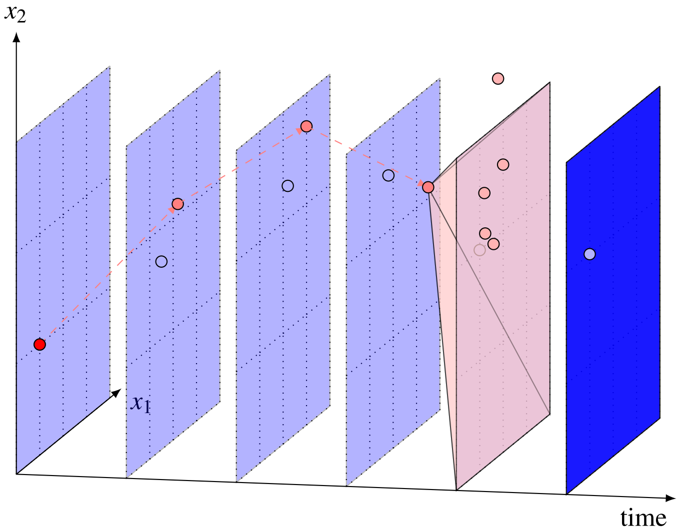
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



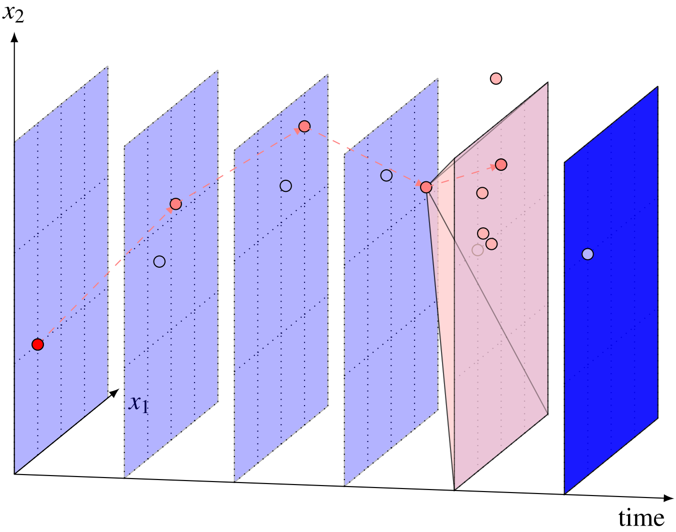
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



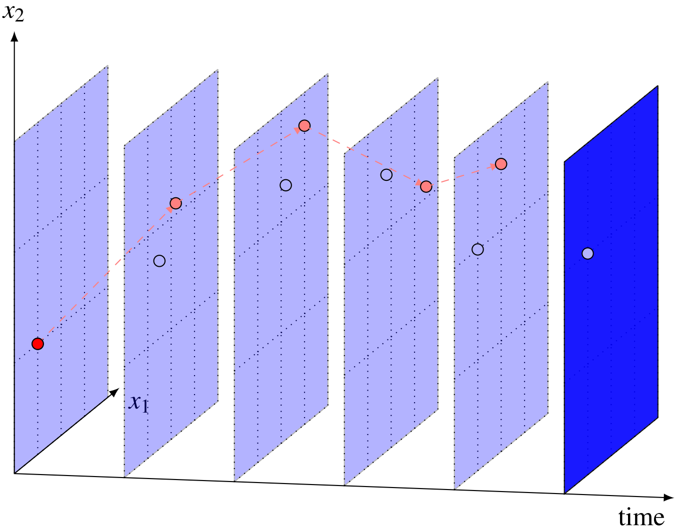
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



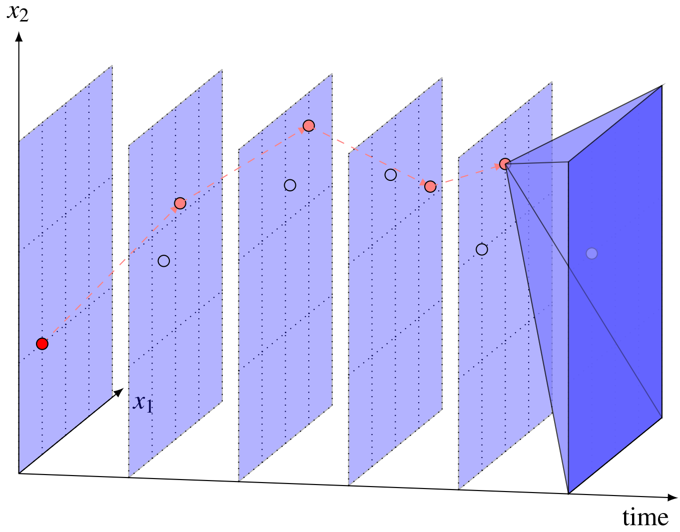
second forward pass : computing trajectory

Trajectory Following Dynamic Programming



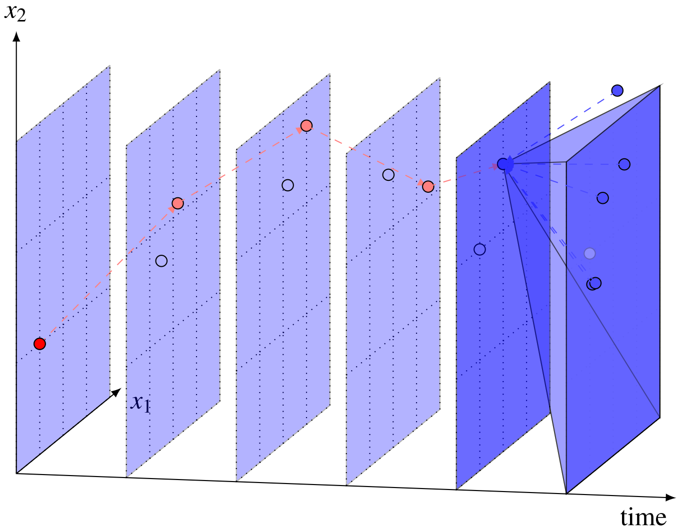
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



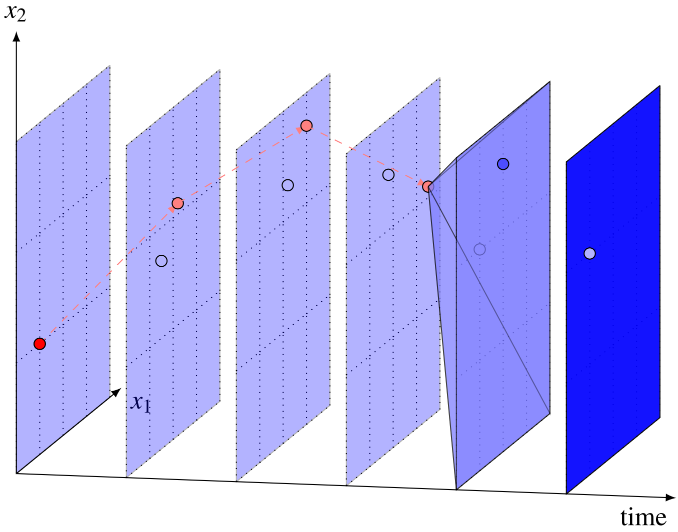
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



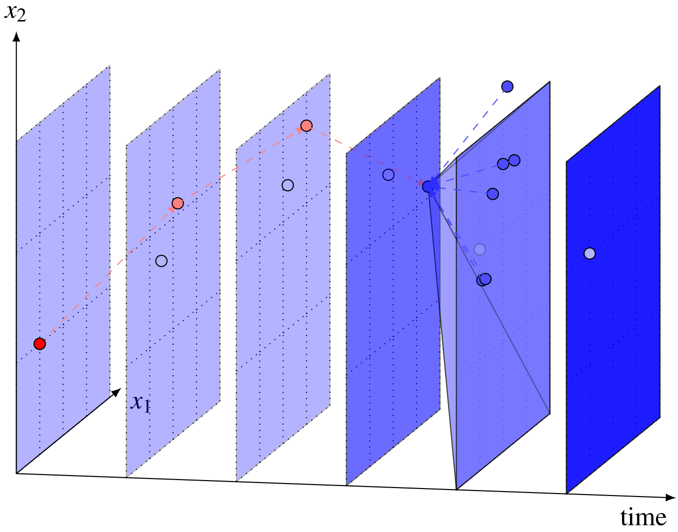
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



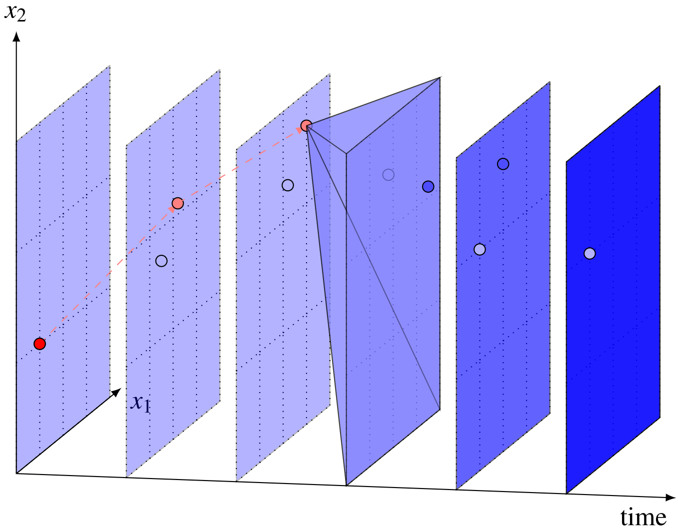
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



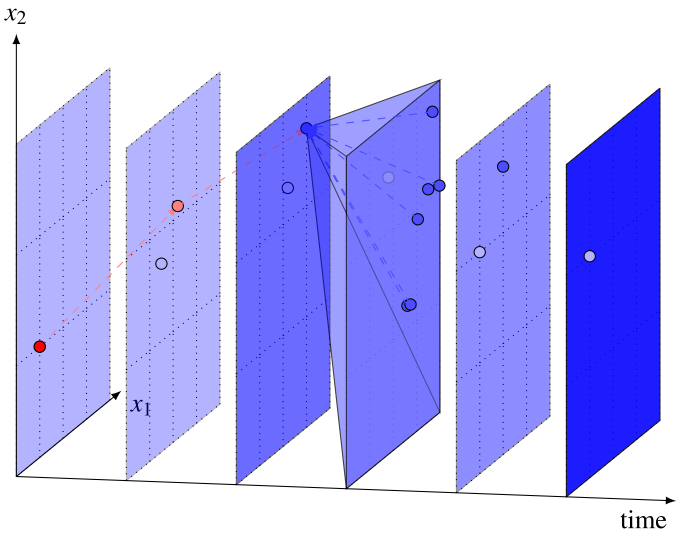
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



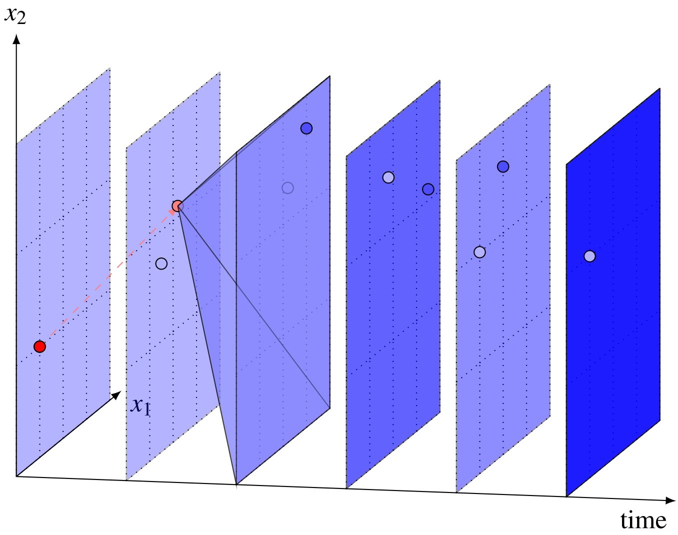
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



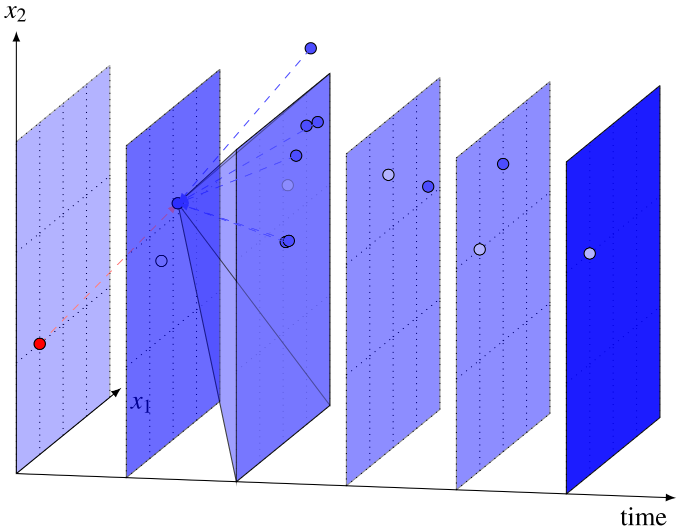
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



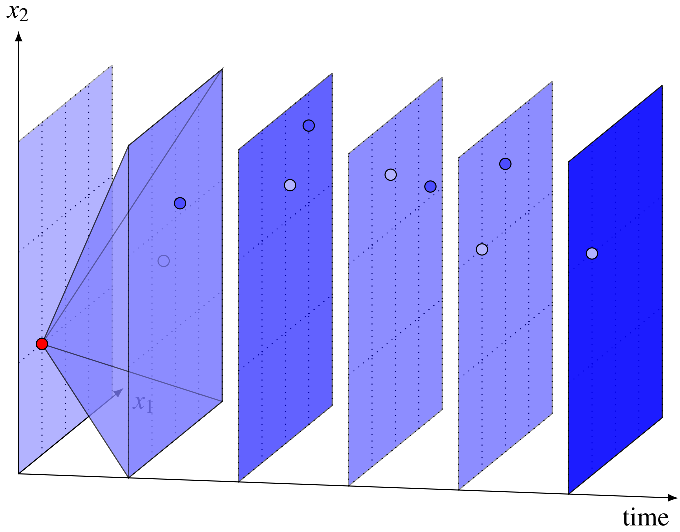
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



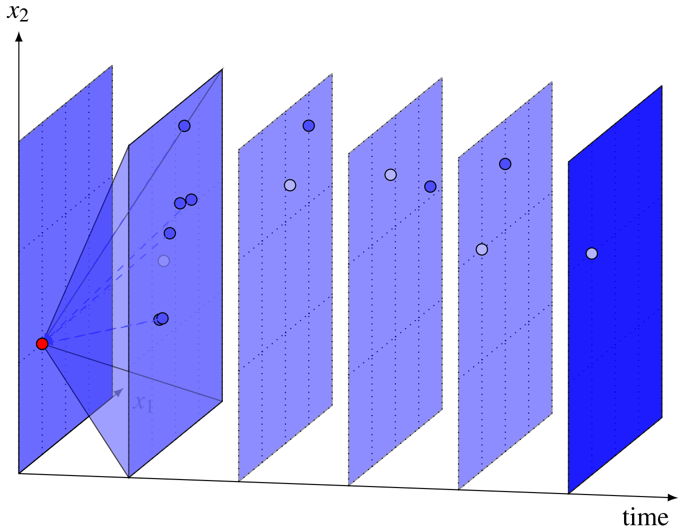
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



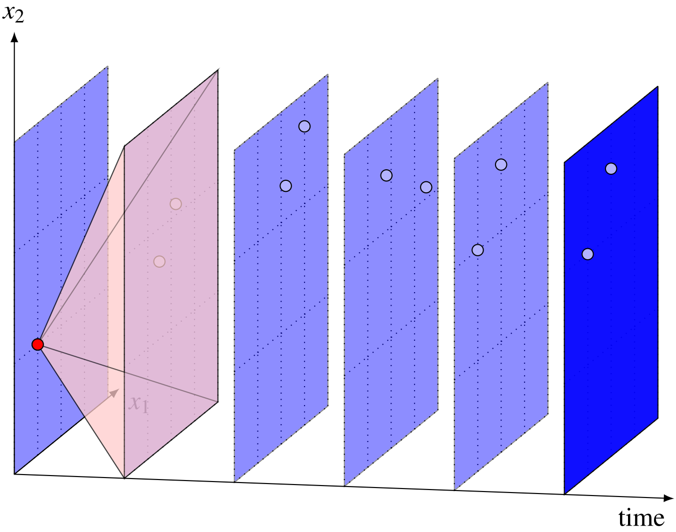
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



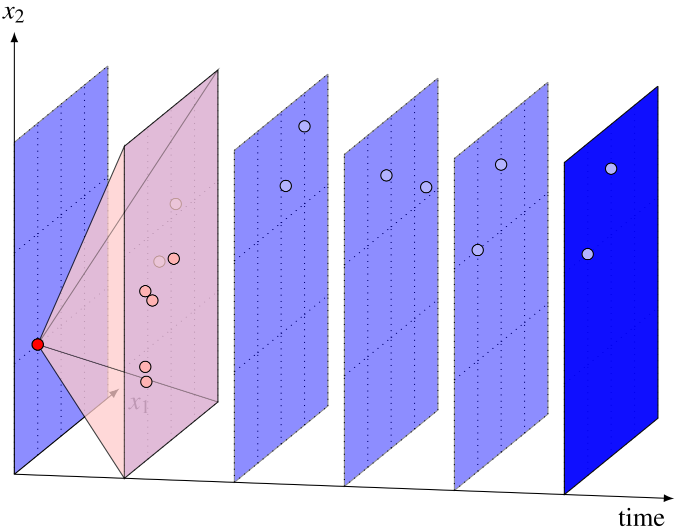
second backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



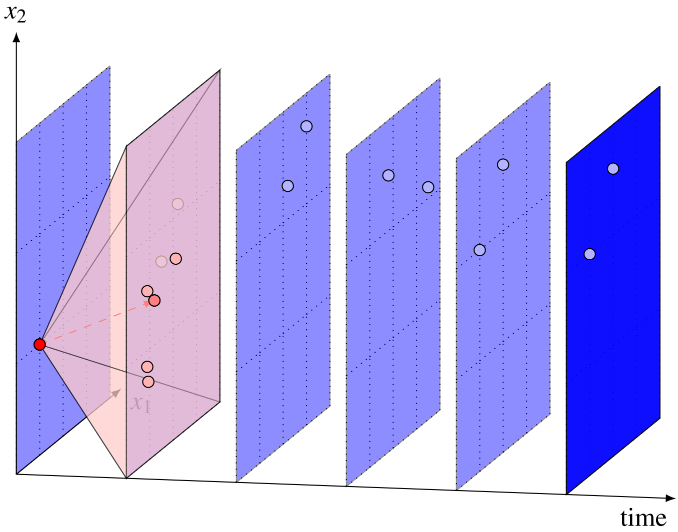
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



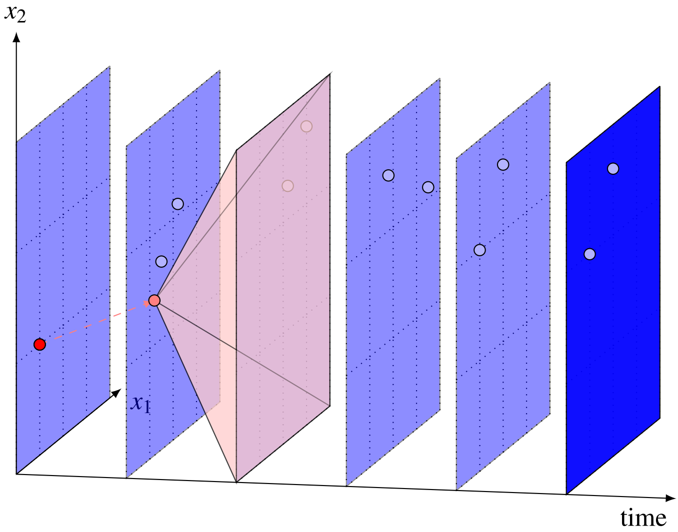
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



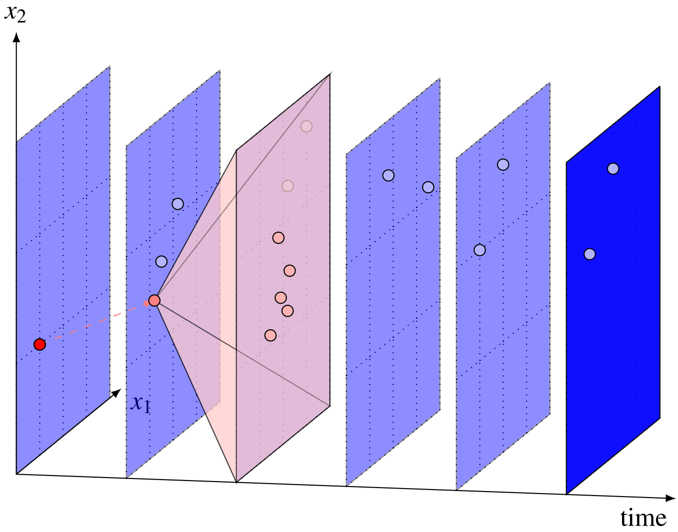
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



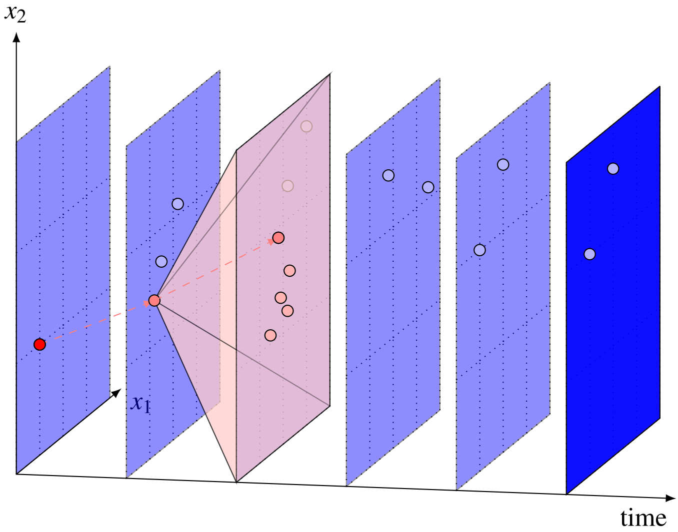
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



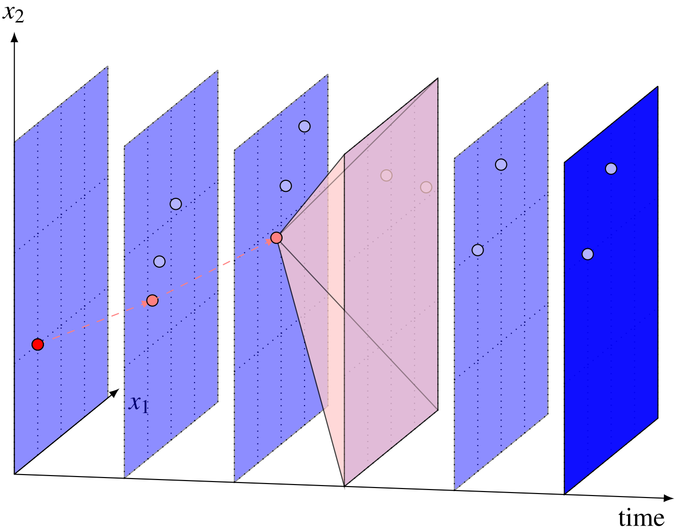
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



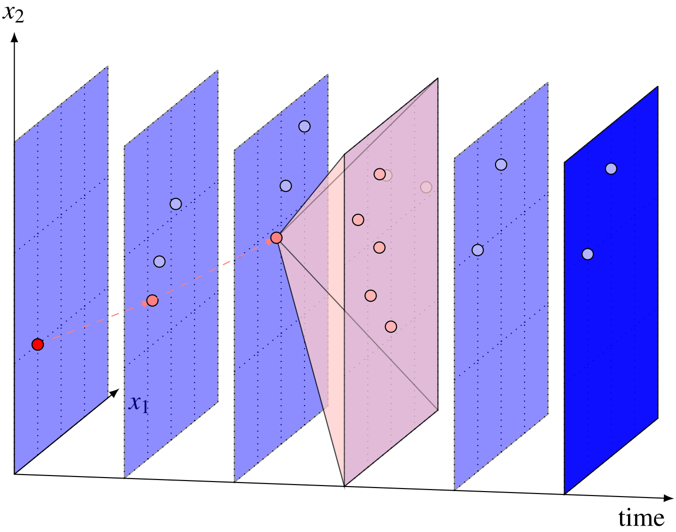
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



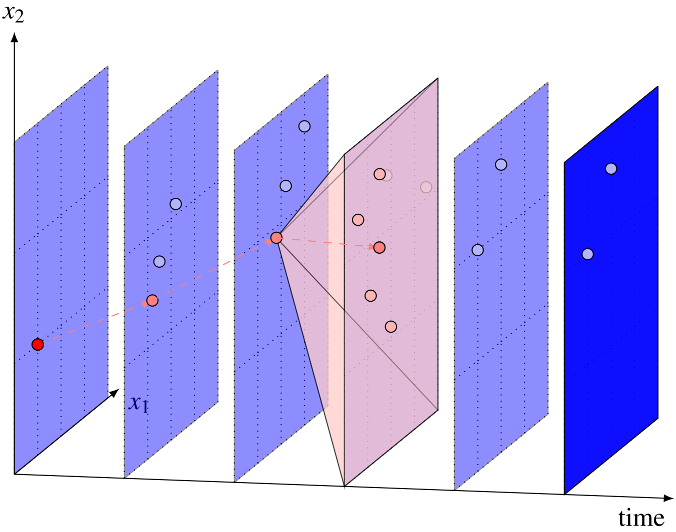
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



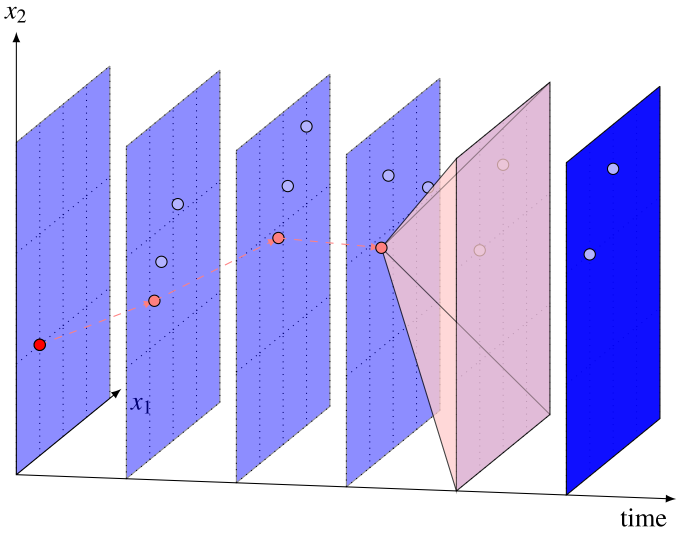
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



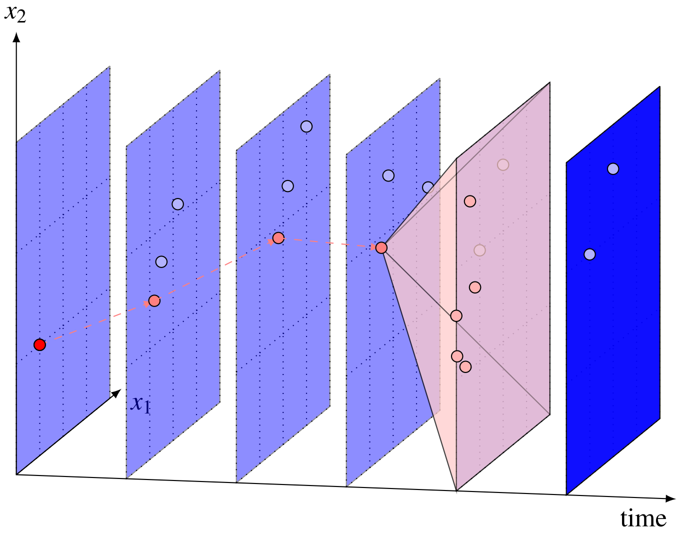
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



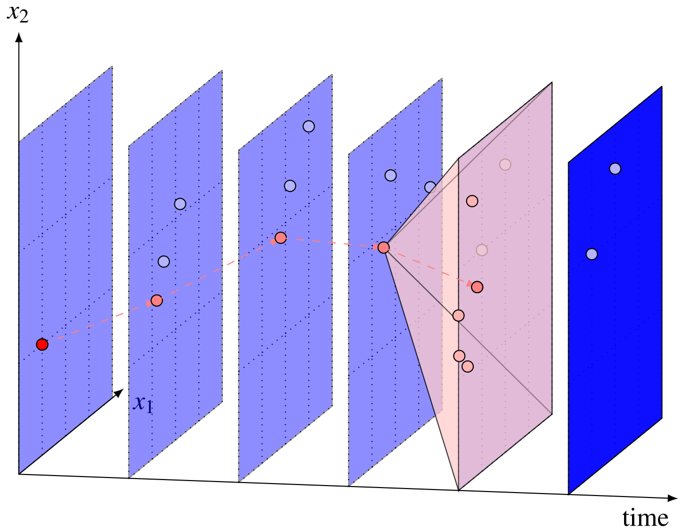
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



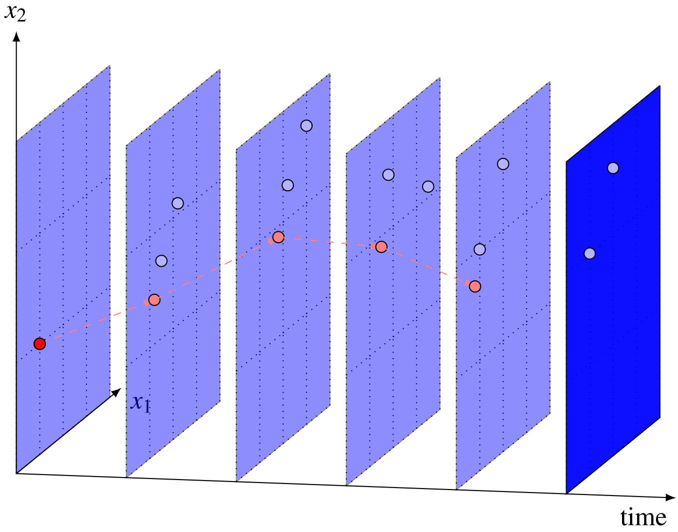
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



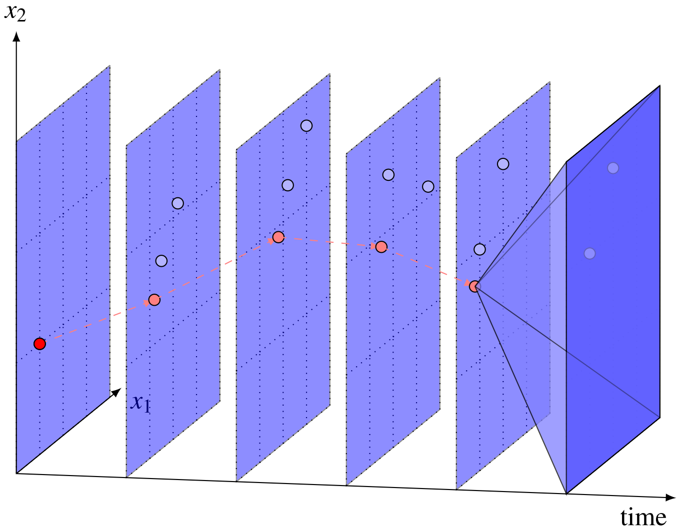
third forward pass : computing trajectory

Trajectory Following Dynamic Programming



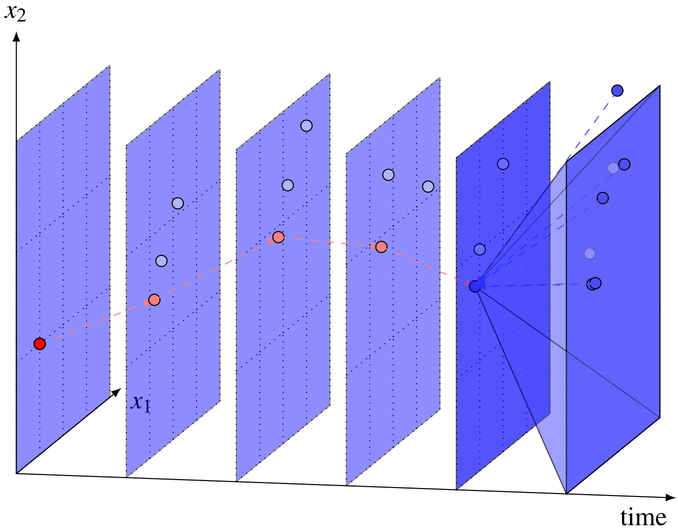
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



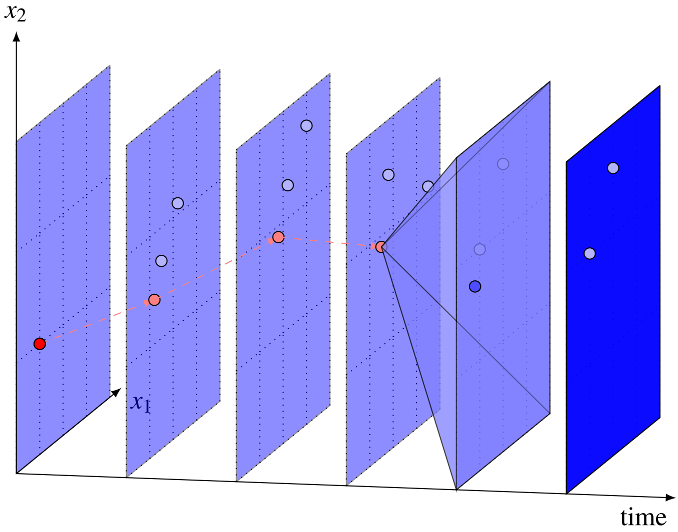
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



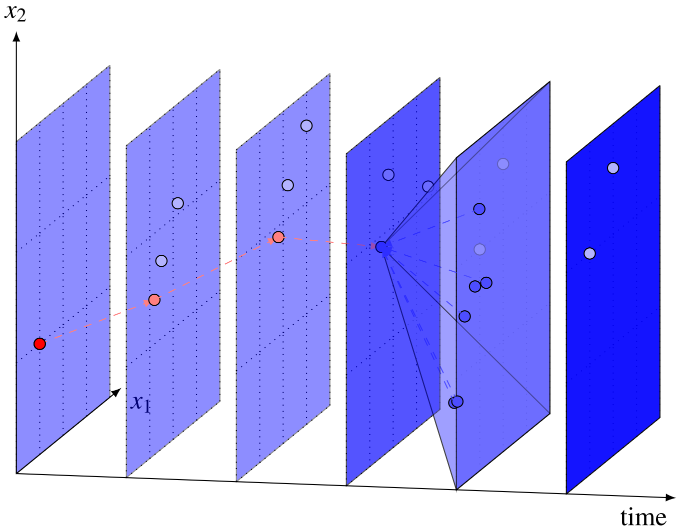
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



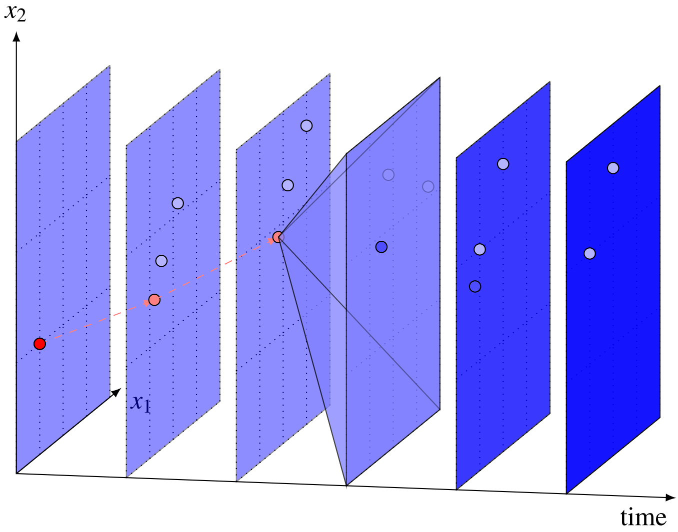
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



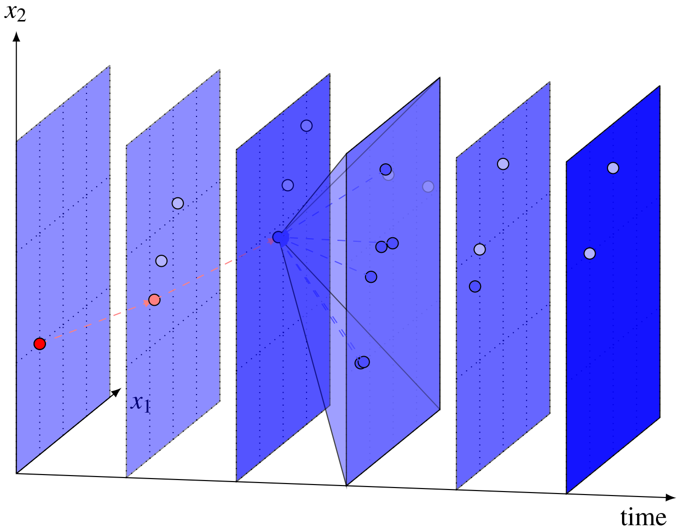
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



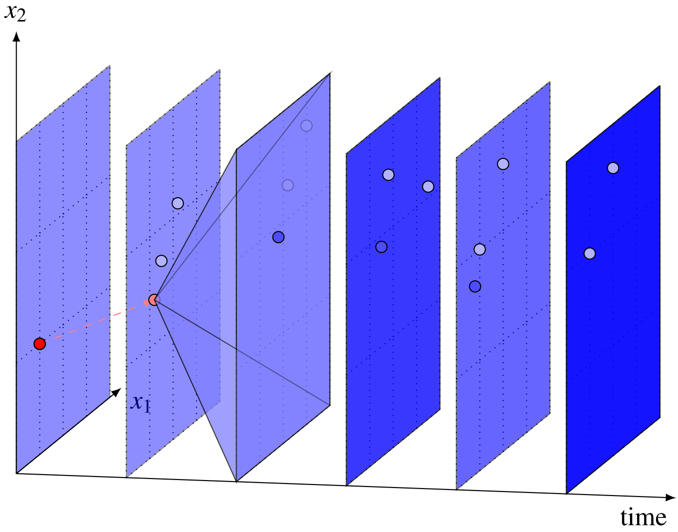
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



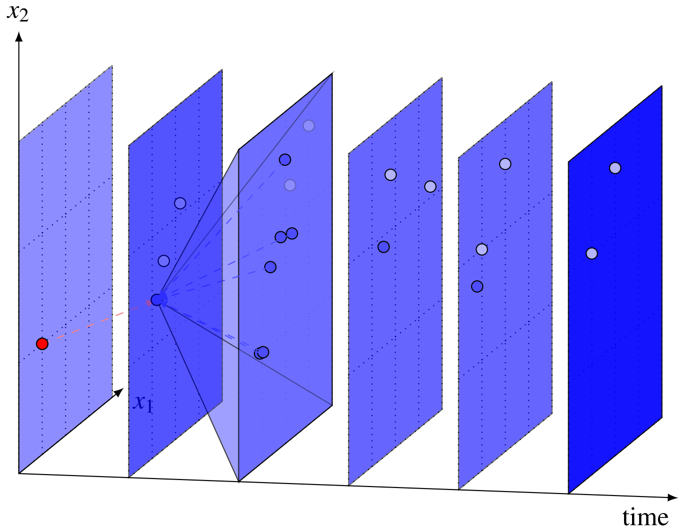
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



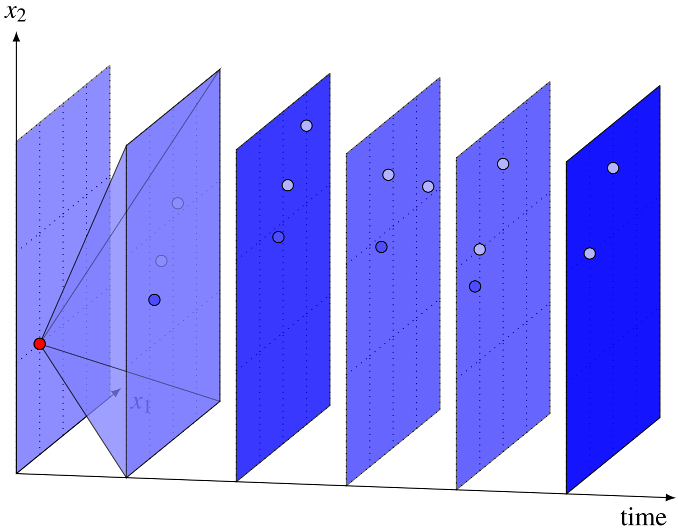
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



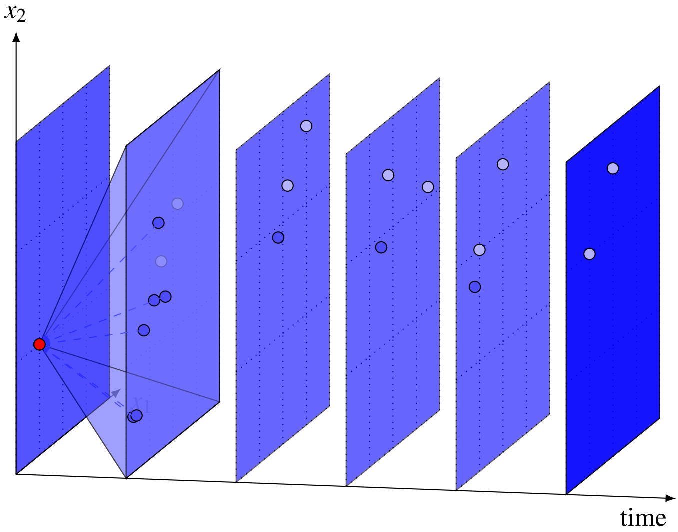
third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



third backward pass : refining approximation (adding cuts)

Trajectory Following Dynamic Programming



And so on...

Contributions on SDDP and its variants

- New framework called Trajectory Following Dynamic Programming (TFDP) encompassing at least 14 variants of SDDP
- Complexity proofs, new for most of those variants
- Do not require finite support assumption
- Allow approximation error
- Adapt to robust and risk averse cases

Some TFDP algorithms

Algorithm's name	Node selection: Choice ξ_t^k	$\mathcal{F}_t$	$\underline{V}_t^k$	$\overline{V}_t^k$	Hypothesis	Complexity known
SDDP	Random sampling	Exact	Benders cuts	V_t	Convex	✔
EDDP	Explorative	Exact	Benders cuts	V_t	Convex	✔
APSDDP	Random sampling	Exact	Adaptive partition	V_t	Linear	✖
SDDiP	Random sampling	Exact	Lagrangian or integer cuts	V_t	Mixed Integer Linear	✖
MIDAS	Random sampling	Exact	Step cuts	V_t	Monotonic Mixed Integer	✖
SLDP	Random sampling	Exact	Reverse norm cuts	V_t	Non-Convex	✖
BDZ17	Problem child	Exact	Benders cuts	Epigraph as convex hull	Convex	✖
BDZ18	Problem child	Exact	Benders $\times$ Epigraph	Hypograph $\times$ Benders	Convex-Concave	✖
RDDP	Deterministic	Exact	Benders cuts	Epigraph as convex hull	Robust	✖
ISDDP	Random sampling	Inexact	Inexact Lagrangian cuts	V_t	Convex	✖
TDP	Problem child	Exact	Benders cuts	Min of quadratic	Convex	✖
ZS19	Random or Problem	Regularized	Generalized conjugacy cuts	Norm cuts	Mixed Integer Convex	✔
NDDP	Random or Problem	Regularized	Benders cuts	Norm cuts	Distributionally Robust	✔
DSDDP	Random sampling	Exact	Benders cuts	Fenchel transform	Linear	✖

Contents

1. Universal Exact Quantization for cost
   - Local in 2-stage
   - Uniform in 2-stage
   - Uniform in multistage
   - Complexity results
2. Local and universal exact Quantization for constraints
   - Adapted partitions
   - Adaptive Partition-based Methods
   - Convergence, complexity and numerical results
3. Trajectory Following Dynamic Programming
4. Conclusion and perspectives

Conclusion

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	✓	✓
Uniform	×	×	✓

- Links with fundamental polyhedral geometry, regular subdivisions and fiber polytope (Chap. 3 and 4).
- *Uniform and universal* exact quantization for $\boldsymbol{c}$ in MSLP (Chap.4).
  - ➥ Polynomial time complexity results.
- *Local* exact quantization for $\boldsymbol{B}$ and $\boldsymbol{b}$.
  - ➥ Adaptive Partition-based Methods (APM) for general distribution: solves small 2SLP with high precision (Chap. 5).
- Extension of Stochastic Dual Dynamic Programming algorithms and more generally all Trajectory Following Dynamic Programming algorithm for non finitely supported distribution (Chap. 6).

Conclusion

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	✓	✓
Uniform	×	×	✓

- Links with fundamental polyhedral geometry, regular subdivisions and fiber polytope (Chap. 3 and 4).
- *Uniform and universal* exact quantization for $\boldsymbol{c}$ in MSLP (Chap.4).
  - ➥ Polynomial time complexity results.
- *Local* exact quantization for $\boldsymbol{B}$ and $\boldsymbol{b}$.
  - ➥ Adaptive Partition-based Methods (APM) for general distribution: solves small 2SLP with high precision (Chap. 5).
- Extension of Stochastic Dual Dynamic Programming algorithms and more generally all Trajectory Following Dynamic Programming algorithm for non finitely supported distribution (Chap. 6).

Conclusion

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	$\times$	$\checkmark$	$\checkmark$
Uniform	$\times$	$\times$	$\checkmark$

- Links with fundamental polyhedral geometry, regular subdivisions and fiber polytope (Chap. 3 and 4).
- *Uniform and universal* exact quantization for $\boldsymbol{c}$ in MSLP (Chap.4).
  - ➥ Polynomial time complexity results.
- *Local* exact quantization for $\boldsymbol{B}$ and $\boldsymbol{b}$.
  - ➥ Adaptive Partition-based Methods (APM) for general distribution: solves small 2SLP with high precision (Chap. 5).
- Extension of Stochastic Dual Dynamic Programming algorithms and more generally all Trajectory Following Dynamic Programming algorithm for non finitely supported distribution (Chap. 6).

Conclusion

	$\boldsymbol{A}$	$(\boldsymbol{B}, \boldsymbol{b})$	$\boldsymbol{c}$
Local	×	✓	✓
Uniform	×	×	✓

- Links with fundamental polyhedral geometry, regular subdivisions and fiber polytope (Chap. 3 and 4).
- *Uniform and universal* exact quantization for $\boldsymbol{c}$ in MSLP (Chap.4).
  - ➥ Polynomial time complexity results.
- *Local* exact quantization for $\boldsymbol{B}$ and $\boldsymbol{b}$.
  - ➥ Adaptive Partition-based Methods (APM) for general distribution: solves small 2SLP with high precision (Chap. 5).
- Extension of Stochastic Dual Dynamic Programming algorithms and more generally all Trajectory Following Dynamic Programming algorithm for non finitely supported distribution (Chap. 6).

Perspectives (Chap. 7)

- Higher order simplex algorithm on the chamber complex for 2SLP.
- 2-time scale MSLP, nested fiber polyhedra and convex bodies.
- Reintroduce approximation or sampling.
- Exact quantization for stochastic integer linear problems
- Understanding the complexity of MSLP.

Perspectives (Chap. 7)

- Higher order simplex algorithm on the chamber complex for 2SLP.
- 2-time scale MSLP, nested fiber polyhedra and convex bodies.
- Reintroduce approximation or sampling.
- Exact quantization for stochastic integer linear problems
- Understanding the complexity of MSLP.

Perspectives (Chap. 7)

- Higher order simplex algorithm on the chamber complex for 2SLP.
- 2-time scale MSLP, nested fiber polyhedra and convex bodies.
- Reintroduce approximation or sampling.
- Exact quantization for stochastic integer linear problems
- Understanding the complexity of MSLP.

Perspectives (Chap. 7)

- Higher order simplex algorithm on the chamber complex for 2SLP.
- 2-time scale MSLP, nested fiber polyhedra and convex bodies.
- Reintroduce approximation or sampling.
- Exact quantization for stochastic integer linear problems
- Understanding the complexity of MSLP.

Perspectives (Chap. 7)

- Higher order simplex algorithm on the chamber complex for 2SLP.
- 2-time scale MSLP, nested fiber polyhedra and convex bodies.
- Reintroduce approximation or sampling.
- Exact quantization for stochastic integer linear problems
- Understanding the complexity of MSLP.

Thank you for listening ! Any question ?

M. Forcier, S. Gaubert, V. Leclère
Exact quantization of multistage stochastic linear problems.
arXiv preprint arXiv:2107.09566 (2021).

M. Forcier, V. Leclère
Generalized adaptive partition-based method for two-stage stochastic linear programs: convergence and generalization.
Operation Research Letters, to appear (2022).

M. Forcier, V. Leclère
Convergence of Trajectory Following Dynamic Programming algorithms for multistage stochastic problems without finite support assumptions
HAL Id : hal-03683697 (2022).

Contents

5. Explicit formulas for general distributions

6. Details on GAPM
   - Recalls on APM
   - A novel APM algorithm
   - Extension of GAPM to general costs

7. Nested fiber polyhedra

8. Polyhedral toolbox for stochastic optimizers
   - Active constraints
   - Link with regular subdivisions
   - Correspondences for parametric linear programming
   - Correspondences for 2SLP

Explicit formulas for usual distributions

in the exact case, we need to compute the quantized probalities $\check{p}_S = \mathbb{P}\left[\boldsymbol{c} \in S\right]$ and the quantized cost $\check{c}_S = \mathbb{E}\left[\boldsymbol{\xi} \mid \boldsymbol{c} \in S\right]$ when S is a polyhedron.

Explicit formulas, valid for S full dimensional simplex or simplicial cone, which can be used through triangulation.

Distribution	Uniform on polytope	Exponential	Gaussian
$d\mathbb{P}(\xi)$	$\frac{\mathbb{1}_{\xi \in Q}}{\mathrm{Vol}_d(Q)} \mathcal{L}_{\mathrm{Aff}(Q)}(d\xi)$	$\frac{e^{\theta^\top \xi} \mathbb{1}_{\xi \in K}}{\Phi_K(\theta)} \mathcal{L}_{\mathrm{Aff}(K)}(d\xi)$	$\frac{e^{-\frac{1}{2}\xi^\top M^{-2} \xi}}{(2\pi)^{\frac{m}{2}} \det M} d\xi$
Support	Polytope : Q	Cone : K	$\mathbb{R}^m$
$\check{p}_S$	$\dfrac{\mathrm{Vol}_d(S)}{\mathrm{Vol}_d(Q)}$	$\dfrac{\lvert \det(\mathrm{Ray}(S)) \rvert}{\Phi_K(\theta)} \prod_{r \in \mathrm{Ray}(S)} \dfrac{1}{-r^\top \theta}$	$\mathrm{Ang}\left(M^{-1} S\right)$
$\check{c}_S$	$\frac{1}{d} \sum_{v \in \mathrm{Vert}(S)} v$	$\left(\sum_{r \in \mathrm{Ray}(S)} \frac{-r_i}{r^\top \theta} \right)_{i \in [m]}$	$\frac{\sqrt{2}\Gamma(\frac{m+1}{2})}{\Gamma(\frac{m}{2})} M \,\mathrm{Ctr}\left(S \cap \mathbb{S}_{m-1}\right)$

Explicit formulas for usual distributions

in the exact case, we need to compute the quantized probalities $\check{p}_S = \mathbb{P}\big[\boldsymbol{c} \in S\big]$ and the quantized cost $\check{c}_S = \mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{c} \in S\big]$ when S is a polyhedron.

Explicit formulas, valid for S full dimensional simplex or simplicial cone, which can be used through triangulation.

Distribution	Uniform on polytope	Exponential	Gaussian
$d\mathbb{P}(\xi)$	$\frac{\mathbb{1}_{\xi \in Q}}{\mathrm{Vol}_d(Q)} \mathcal{L}_{\mathrm{Aff}(Q)}(d\xi)$	$\frac{e^{\theta^\top \xi} \mathbb{1}_{\xi \in K}}{\Phi_K(\theta)} \mathcal{L}_{\mathrm{Aff}(K)}(d\xi)$	$\frac{e^{-\frac{1}{2}\xi^\top M^{-2} \xi}}{(2\pi)^{\frac{m}{2}} \det M} d\xi$
Support	Polytope : Q	Cone : K	$\mathbb{R}^m$
$\check{p}_S$	$\dfrac{\mathrm{Vol}_d(S)}{\mathrm{Vol}_d(Q)}$	$\dfrac{\lvert\det(\mathrm{Ray}(S))\rvert}{\Phi_K(\theta)} \displaystyle\prod_{r \in \mathrm{Ray}(S)} \dfrac{1}{-r^\top \theta}$	$\mathrm{Ang}\left(M^{-1} S\right)$
$\check{c}_S$	$\frac{1}{d} \sum_{v \in \mathrm{Vert}(S)} v$	$\left(\sum_{r \in \mathrm{Ray}(S)} \frac{-r_i}{r^\top \theta}\right)_{i \in [m]}$	$\frac{\sqrt{2}\Gamma(\frac{m+1}{2})}{\Gamma(\frac{m}{2})} M \,\mathrm{Ctr}\left(S \cap \mathbb{S}_{m-1}\right)$

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

2 stage stochastic linear programming (2SLP)

$$\min_{x \in \mathbb{R}^n_+} \quad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$
$$\text{s.t.} \quad Ax = b$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \qquad = \max_{\lambda \in \mathbb{R}^n} (h - Tx)^\top \lambda$$
$$\text{s.t.} \quad Tx + Wy = h \qquad \text{s.t.} \quad W^\top \lambda \leqslant q$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\min_{x \in \mathbb{R}^n_+} \quad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$
$$\text{s.t.} \quad Ax = b$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \quad \text{s.t.} \quad Tx + Wy = h \qquad = \max_{\lambda \in \mathbb{R}^n} (h - Tx)^\top \lambda \quad \text{s.t.} \quad W^\top \lambda \leqslant q$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\begin{aligned} \min_{x \in \mathbb{R}^n_+} \quad & c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big] \\ \text{s.t.} \quad & Ax = b \end{aligned}$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \quad \text{s.t.} \quad Tx + Wy = h \qquad = \max_{\lambda \in \mathbb{R}^n} (h - Tx)^\top \lambda \quad \text{s.t.} \quad W^\top \lambda \leqslant q$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\min_{x \in \mathbb{R}^n_+} \quad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$
$$\text{s.t.} \quad Ax = b$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \quad \text{s.t.} \quad Tx + Wy = h \qquad = \max_{\lambda \in \mathbb{R}^n} (h - Tx)^\top \lambda \quad \text{s.t.} \quad W^\top \lambda \leqslant q$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\min_{x \in X} \qquad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \qquad = \max_{\lambda \in D} (h - Tx)^\top \lambda$$

$$\text{s.t.} \quad Tx + Wy = h$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\min_{x \in X} \qquad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \qquad = \max_{\lambda \in D} (h - Tx)^\top \lambda$$
$$\text{s.t.} \quad Tx + Wy = h$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

No direct formula to compute $V(x) := \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$ even for fixed x.

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

2 stage stochastic linear programming (2SLP)

$$\min_{x \in X} \qquad c^\top x + \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$$

where $\boldsymbol{\xi} = (\boldsymbol{T}, \boldsymbol{h})$ is random whereas q and W are deterministic[1]

$$Q(x, \xi) := \min_{y \in \mathbb{R}^m_+} q^\top y \qquad\qquad = \max_{\lambda \in D} (h - Tx)^\top \lambda$$
$$\text{s.t.} \quad Tx + Wy = h$$

We define

$$X := \{x \in \mathbb{R}^n_+ \mid Ax = b\} \qquad\qquad D := \{\lambda \in \mathbb{R}^\ell \mid W^\top \lambda \leqslant q\}$$

No direct formula to compute $V(x) := \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big]$ even for fixed x.
$\rightsquigarrow$ need to discretize $\boldsymbol{\xi}$

[1] Can be extended to generic random $\boldsymbol{q}$, and finitely supported $\boldsymbol{W}$

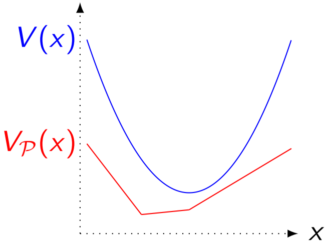
Properties of partitioned cost-to-go

Recall that

$$V(x) = \mathbb{E}\Big[Q(x,\boldsymbol{\xi})\Big]$$

$$V_{\mathcal{P}}(x) = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\,Q\Big(x, \mathbb{E}\big[\boldsymbol{\xi}|P\big]\Big)$$

- $Q(x,\cdot)$ is convex $\rightsquigarrow V_{\mathcal{P}} \leqslant V$.
- $Q\big(\cdot, \mathbb{E}\big[\boldsymbol{\xi}|P\big]\big)$ is polyhedral $\rightsquigarrow V_{\mathcal{P}}$ is polyhedral.



Finally,

$$\min_{x\in X} c^\top x + V_{\mathcal{P}}(x) \qquad (2SLP_{\mathcal{P}})$$

is equivalent to

$$\min_{x\in X,(y_P)_{P\in\mathcal{P}}} \quad c^\top x + \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] q^\top y_P$$

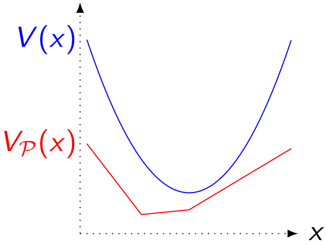
$$\mathbb{E}\big[\boldsymbol{T}|P\big]x + Wy_P \leqslant \mathbb{E}\big[\boldsymbol{h}|P\big] \qquad \forall P\in\mathcal{P}$$

Properties of partitioned cost-to-go

Recall that

$$V(x) = \mathbb{E}\Big[Q(x,\boldsymbol{\xi})\Big]$$
$$V_{\mathcal{P}}(x) = \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big]\, Q\big(x, \mathbb{E}\big[\boldsymbol{\xi}|P\big]\big)$$

- $Q(x,\cdot)$ is convex $\rightsquigarrow V_{\mathcal{P}} \leqslant V$.
- $Q\big(\cdot,\mathbb{E}\big[\boldsymbol{\xi}|P\big]\big)$ is polyhedral $\rightsquigarrow V_{\mathcal{P}}$ is polyhedral.



Finally,

$$\min_{x\in X} c^\top x + V_{\mathcal{P}}(x) \tag{$2SLP_{\mathcal{P}}$}$$

is equivalent to

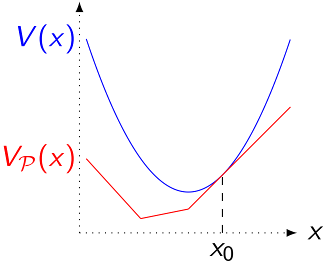
$$\begin{aligned} \min_{x\in X,(y_P)_{P\in\mathcal{P}}} \quad & c^\top x + \sum_{P\in\mathcal{P}} \mathbb{P}\big[P\big] q^\top y_P \\ & \mathbb{E}\big[\boldsymbol{T}|P\big]x + W y_P \leqslant \mathbb{E}\big[\boldsymbol{h}|P\big] \qquad \forall P \in \mathcal{P} \end{aligned}$$

Adapted partition

Definition

We say that a partition $\mathcal{P}$ is adapted to x_0 if

$$V_{\mathcal{P}}(x_0) = V(x_0) := \mathbb{E}\big[Q(x_0, \boldsymbol{\xi})\big]$$

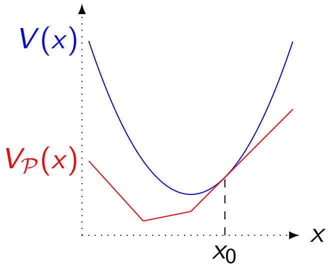


Adapted partition

Definition

We say that a partition $\mathcal{P}$ is adapted to x_0 if

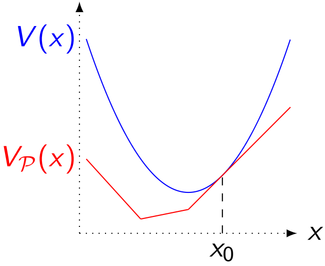
$$V_{\mathcal{P}}(x_0) = V(x_0) := \mathbb{E}\left[Q(x_0, \xi)\right]$$



Definition

An partition oracle is a function taking a first stage decision x^k as argument and returning an partition of Ξ.

Adapted partition



Definition

We say that a partition $\mathcal{P}$ is adapted to x_0 if

$$V_{\mathcal{P}}(x_0) = V(x_0) := \mathbb{E}\left[Q(x_0, \xi)\right]$$

Definition

An partition oracle is a function taking a first stage decision x^k as argument and returning an partition of Ξ.

Definition

An adapted partition oracle is a function taking a first stage decision x^k as argument and returning an adapted to x^k partition of Ξ.

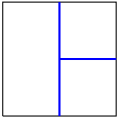
Refinement

$\mathcal{R}$ refines $\mathcal{P}$ ($\mathcal{R} \preccurlyeq \mathcal{P}$) if

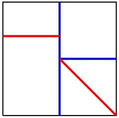
$$\forall R \in \mathcal{R}, \exists P \in P, R \subset P$$

[$\mathcal{R} \preccurlyeq_{\mathbb{P}} \mathcal{P}$ if $\mathcal{R}$ refines $\mathcal{P}$ up to $\mathbb{P}$-null sets.]

Then, $\quad \mathcal{R} \preccurlyeq_{\mathbb{P}} \mathcal{P} \Rightarrow V_{\mathcal{R}} \geqslant V_{\mathcal{P}}$



$\mathcal{P}$



$\mathcal{R}$

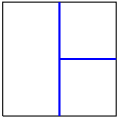
Refinement

$\mathcal{R}$ refines $\mathcal{P}$ ($\mathcal{R} \preccurlyeq \mathcal{P}$) if

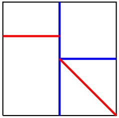
$$\forall R \in \mathcal{R}, \exists P \in P, R \subset P$$

[$\mathcal{R} \preccurlyeq_{\mathbb{P}} \mathcal{P}$ if $\mathcal{R}$ refines $\mathcal{P}$ up to $\mathbb{P}$-null sets.]

Then, $\quad \mathcal{R} \preccurlyeq_{\mathbb{P}} \mathcal{P} \Rightarrow V_{\mathcal{R}} \geqslant V_{\mathcal{P}}$



$\mathcal{P}$



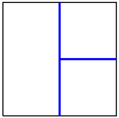
$\mathcal{R}$

The common refinement of $\mathcal{P}$ and $\mathcal{P}'$ is

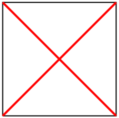
$$\mathcal{P} \wedge \mathcal{P}' := \{P \cap P' \mid P \in \mathcal{P}, P' \in \mathcal{P}'\}$$

Since $\mathcal{P} \wedge \mathcal{P}'$ refines $\mathcal{P}$ and $\mathcal{P}'$

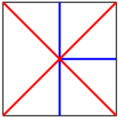
$$\max(V_{\mathcal{P}}, V_{\mathcal{P}'}) \leqslant V_{\mathcal{P} \wedge \mathcal{P}'}$$



$\mathcal{P}$



$\mathcal{P}'$



$\mathcal{P} \wedge \mathcal{P}'$

General framework for APM

$k \leftarrow 0$, $z_U^0 \leftarrow +\infty$, $z_L^0 \leftarrow -\infty$, $\mathcal{P}^0 \leftarrow \{\Xi\}$;
while $z_U^k - z_L^k > \varepsilon$ **do**
  $k \leftarrow k+1$;
  Solve $z_L^k \leftarrow \min_{x \in X} c^\top x + V_{\mathcal{P}^{k-1}}(x)$;
  and let x^k be an optimal solution ;
  Call an adapted partition oracle on x^k yielding $\mathcal{P}_{x^k}$;
  $\mathcal{P}^k \leftarrow \mathcal{P}^{k-1} \wedge \mathcal{P}_{x^k}$;
  $z_U^k \leftarrow \min\left(z_U^{k-1}, c^\top x^k + V_{\mathcal{P}^k}(x^k)\right)$;
end

Algorithm 1: Generic framework for APM.

Previous APM methods

Lemma (Song & Luedtke, 2015)

Let $\mathcal{P}$ a partition of Ξ. $\mathcal{P}$ is adapted at x iff for all set of scenarios $P \in \mathcal{P}$, there exists a common optimal multiplier λ_P, i.e.

$$\forall P \in \mathcal{P}, \quad \exists \lambda_P \in D, \quad \forall \xi_k \in P, \qquad \lambda_P \in \operatorname*{argmax}_{\lambda \in D} (h^k - T^k x)^\top \lambda$$

Previous APM methods

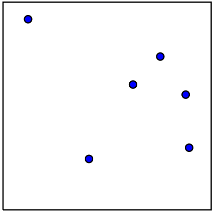
Lemma (Song & Luedtke, 2015)

Let $\mathcal{P}$ a partition of Ξ. $\mathcal{P}$ is adapted at x iff for all set of scenarios $P \in \mathcal{P}$, there exists a common optimal multiplier λ_P, i.e.

$$\forall P \in \mathcal{P}, \quad \exists \lambda_P \in D, \quad \forall \xi_k \in P, \qquad \lambda_P \in \underset{\lambda \in D}{\operatorname{argmax}} (h^k - T^k x)^\top \lambda$$

Idea

- Sample a large number of scenario
- without loss of precision aggregate scenarios



Previous APM methods

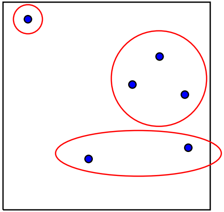
Lemma (Song & Luedtke, 2015)

Let $\mathcal{P}$ a partition of Ξ. $\mathcal{P}$ is adapted at x iff for all set of scenarios $P \in \mathcal{P}$, there exists a common optimal multiplier λ_P, i.e.

$$\forall P \in \mathcal{P}, \quad \exists \lambda_P \in D, \quad \forall \xi_k \in P, \qquad \lambda_P \in \underset{\lambda \in D}{\operatorname{argmax}} (h^k - T^k x)^\top \lambda$$

Idea

- Sample a large number of scenario
- without loss of precision aggregate scenarios



Previous APM methods

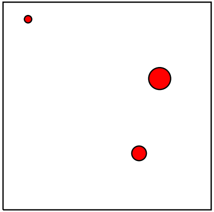
Lemma (Song & Luedtke, 2015)

Let $\mathcal{P}$ a partition of Ξ. $\mathcal{P}$ is adapted at x iff for all set of scenarios $P \in \mathcal{P}$, there exists a common optimal multiplier λ_P, i.e.

$$\forall P \in \mathcal{P}, \quad \exists \lambda_P \in D, \quad \forall \xi_k \in P, \qquad \lambda_P \in \underset{\lambda \in D}{\mathrm{argmax}} (h^k - T^k x)^\top \lambda$$

Idea

- Sample a large number of scenario
- without loss of precision aggregate scenarios



Previous APM methods

Lemma (Song & Luedtke, 2015)

Let $\mathcal{P}$ a partition of Ξ. $\mathcal{P}$ is adapted at x iff for all set of scenarios $P \in \mathcal{P}$, there exists a common optimal multiplier λ_P, i.e.

$$\forall P \in \mathcal{P}, \quad \exists \lambda_P \in D, \quad \forall \xi_k \in P, \qquad \lambda_P \in \underset{\lambda \in D}{\operatorname{argmax}} (h^k - T^k x)^\top \lambda$$

Lemma (Ramirez-Pico & Moreno, 2020)

Let $\mathcal{P}$ a partition of Ξ. If there exists $\lambda(\boldsymbol{\xi})$ such that, for all $P \in \mathcal{P}$,

$$\mathbb{E}\left[\boldsymbol{h}|P\right]^\top \mathbb{E}\left[\lambda(\boldsymbol{\xi})|P\right] = \mathbb{E}\left[\boldsymbol{h}^\top \lambda(\boldsymbol{\xi})|P\right]$$
$$x^\top \mathbb{E}\left[\boldsymbol{T}|P\right]^\top \mathbb{E}\left[\lambda(\boldsymbol{\xi})|P\right] = x^\top \mathbb{E}\left[\boldsymbol{T}^\top \lambda(\boldsymbol{\xi})|P\right]$$

then $\mathcal{P}$ is an adapted partition.

A (partial) comparison between partition based results

Paper	Song, Luedtke (2015)	Ramirez-Pico, Moreno (2020)	F., Leclère (2021)
Non-finite supp(ξ)	×	✓	✓
Explicit oracle	✓	×	✓
Proof of convergence	✓	×	✓
Complexity result	×	×	✓
Fast iteration	✓	×	×

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Local exact quantization and adapted partition

Local exact quantization

random cost

Recall that for a fixed x,

$$\mathbb{E}\big[\min_{y \in P_x} \boldsymbol{c}^\top y\big] = \sum_{N \in \mathcal{N}(P_x)} p_N \min_{y \in P_x} \check{c}_N{}^\top y$$

where,

$$p_N := \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big]$$
$$\check{c}_N := \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]$$

$$P_x := \{y \in \mathbb{R}^m \mid Ay + Bx \leqslant b\}$$

GAPM

random constraints

Similarly, for a given q, and all x,

$$\begin{aligned} V(x) &:= \mathbb{E}\big[Q(x, \boldsymbol{\xi})\big] \\ &= \mathbb{E}\big[\max_{\lambda \in D_q} (\boldsymbol{h} - \boldsymbol{T}x)^\top \lambda\big] \\ &= \sum_{N \in \mathcal{N}(D_q)} p_N \max_{\lambda \in D_q} \psi_{N,x}{}^\top \lambda \end{aligned}$$

where,

$$p_N := \mathbb{P}\big[\boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\big]$$
$$\psi_{N,x} := \mathbb{E}\big[\boldsymbol{h} - \boldsymbol{T}x \mid \boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\big]$$

$$D_q := \{\lambda \in \mathbb{R}^l \mid W^\top \lambda \leqslant q\}$$

An explicit adapted partition

Consider $x \in \mathbb{R}^n$ and $N \in \mathcal{N}(D_q)$ a normal cone of D_q. We define

$$E_{N,x} := \{\xi \in \Xi \mid h - Tx \in \operatorname{ri} N\}$$

Theorem (FL 2021)

$\mathcal{R}_x := \left\{E_{N,x} \mid N \in \mathcal{N}(D_q)\right\}$ *is an adapted partition to* x
i.e. $V_{\mathcal{R}_x}(x) = V(x)$

Proof:

$$\begin{aligned} V(x) &:= \mathbb{E}\left[Q(x,\xi)\right] \\ &= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\left[\boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\right] \min_{\lambda \in D} \mathbb{E}\left[\boldsymbol{h} - \boldsymbol{T}x \,\middle|\, \boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\right]^\top \lambda \\ &= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\left[\boldsymbol{\xi} \in E_{N,x}\right] Q\Big(\mathbb{E}\left[\boldsymbol{\xi} \,\middle|\, \boldsymbol{\xi} \in E_{N,x}\right], x\Big) = V_{\mathcal{R}_x}(x) \end{aligned}$$

An explicit adapted partition

Consider $x \in \mathbb{R}^n$ and $N \in \mathcal{N}(D_q)$ a normal cone of D_q. We define

$$E_{N,x} := \{\xi \in \Xi \mid h - Tx \in \operatorname{ri} N\}$$

Theorem (FL 2021)

$\mathcal{R}_x := \big\{E_{N,x} \mid N \in \mathcal{N}(D_q)\big\}$ *is an adapted partition to* x
i.e. $V_{\mathcal{R}_x}(x) = V(x)$

Proof:

$$\begin{aligned} V(x) &:= \mathbb{E}\big[Q(x,\boldsymbol{\xi})\big] \\ &= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\big[\boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\big] \min_{\lambda \in D} \mathbb{E}\big[\boldsymbol{h} - \boldsymbol{T}x \,|\, \boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\big]^\top \lambda \\ &= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\big[\boldsymbol{\xi} \in E_{N,x}\big] Q\Big(\mathbb{E}\big[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in E_{N,x}\big], x\Big) = V_{\mathcal{R}_x}(x) \end{aligned}$$

An explicit adapted partition

Consider $x \in \mathbb{R}^n$ and $N \in \mathcal{N}(D_q)$ a normal cone of D_q. We define

$$E_{N,x} := \{\xi \in \Xi \mid h - Tx \in \operatorname{ri} N\}$$

Theorem (FL 2021)

$\mathcal{R}_x := \left\{E_{N,x} \mid N \in \mathcal{N}(D_q)\right\}$ *is an adapted partition to* x
i.e. $V_{\mathcal{R}_x}(x) = V(x)$

Proof:

$$\begin{aligned}
V(x) &:= \mathbb{E}\left[Q(x, \boldsymbol{\xi})\right] \\
&= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\left[\boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\right] \min_{\lambda \in D} \mathbb{E}\left[\boldsymbol{h} - \boldsymbol{T}x \,|\, \boldsymbol{h} - \boldsymbol{T}x \in \operatorname{ri} N\right]^\top \lambda \\
&= \sum_{N \in \mathcal{N}(D)} \mathbb{P}\left[\boldsymbol{\xi} \in E_{N,x}\right] Q\Big(\mathbb{E}\left[\boldsymbol{\xi} \,|\, \boldsymbol{\xi} \in E_{N,x}\right], x\Big) = V_{\mathcal{R}_x}(x)
\end{aligned}$$

➥ Is it the coarsest one ?

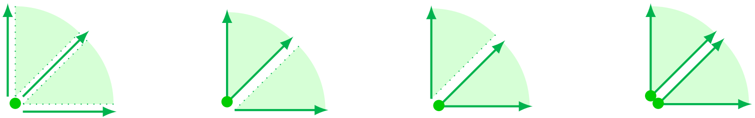
Conditions for a partition to be adapted

Theorem (FL 2021)

For $x \in \mathbb{R}^n$ and $\mathcal{P}$ a partition of Ξ, there exists $\overline{\mathcal{R}}_x \succcurlyeq_{\mathbb{P}} \mathcal{R}_x$ such that

$$\mathcal{P} \preccurlyeq_{\mathbb{P}} \overline{\mathcal{R}}_x \iff V_{\mathcal{P}}(x) = V(x).$$

- If ξ admits a density, $\mathcal{R}_x =_{\mathbb{P}} \overline{\mathcal{R}}_x$.
- An oracle is adapted if and only if it returns a partition $\mathcal{P}$ refining $\overline{\mathcal{R}}_x$.



$\mathcal{R}_x$ $\mathcal{P}$ $\mathcal{P}'$ $\overline{\mathcal{R}}_x$

$$E_{N,x} := \{\xi \in \Xi \mid h - Tx \in \mathrm{ri}(N)\}$$
$$\mathcal{R}_x := \left\{E_{N,x} \mid N \in \mathcal{N}(D_q)\right\}$$

$$\overline{E}_{N,x} := \{\xi \in \Xi \mid h - Tx \in N\}$$
$$\overline{\mathcal{R}}_x := \left\{E_{N,x} \mid N \in \mathcal{N}(D_q)^{\max}\right\}.$$

Subgradient of partition function

Recall that if $\mathcal{P} \preccurlyeq_{\mathbb{P}} \mathcal{R}_x$ then

$$V_{\mathcal{R}_x}(x) = V_{\mathcal{P}}(x) = V(x)$$

$$V_{\mathcal{R}_x}(\cdot) \leqslant V_{\mathcal{P}}(\cdot) \leqslant V(\cdot)$$

Lemma

Let $x \in \mathrm{dom}(V)$ *and* $\mathcal{P}$ *be a refinement of* $\mathcal{R}_x$*, i.e.* $\mathcal{P} \preccurlyeq \mathcal{R}_x$*, then*

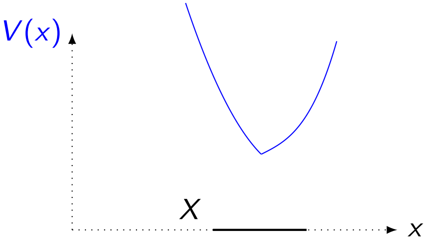
$$\partial V_{\mathcal{R}_x}(x) \subset \partial V_{\mathcal{P}}(x) \subset \partial V(x)$$

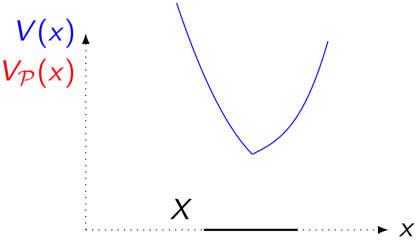
Furthermore, if $x \in \mathrm{ri}\,\mathrm{dom}(V)$*,*

$$\partial V_{\mathcal{R}_x}(x) = \partial V_{\mathcal{P}}(x) = \partial V(x)$$

Link with Benders decomposition and L-shaped

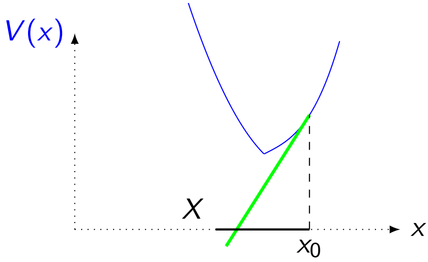
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

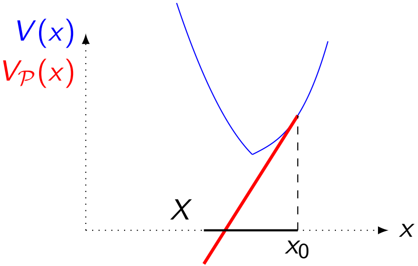




Link with Benders decomposition and L-shaped

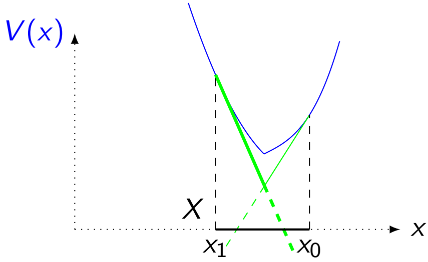
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

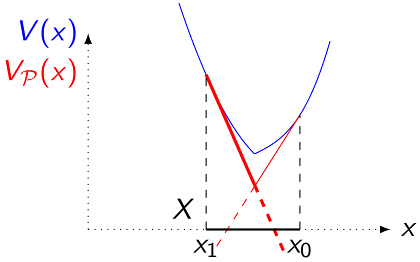




Link with Benders decomposition and L-shaped

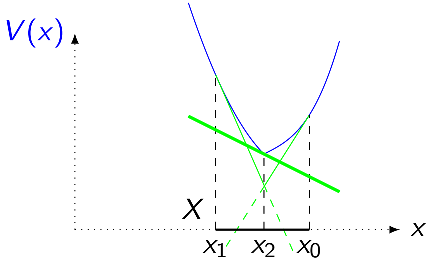
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

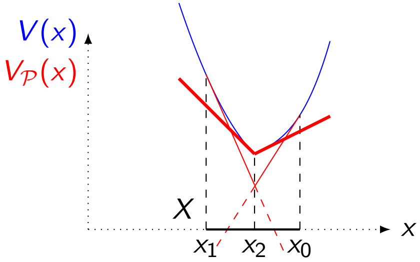




Link with Benders decomposition and L-shaped

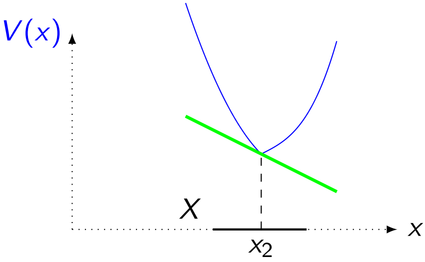
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

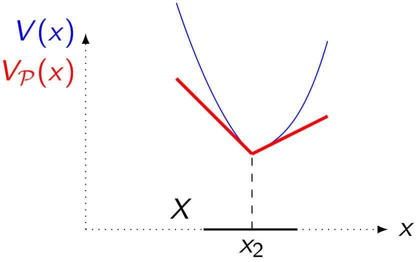




Link with Benders decomposition and L-shaped

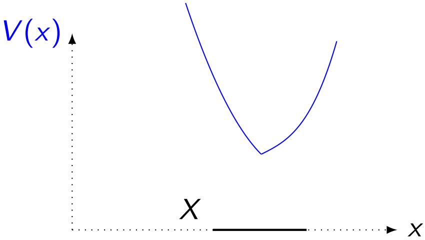
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

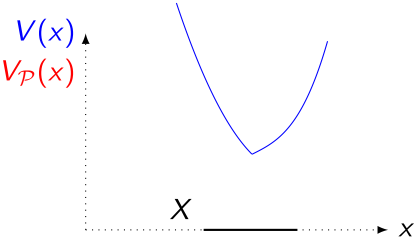




Link with Benders decomposition and L-shaped

Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

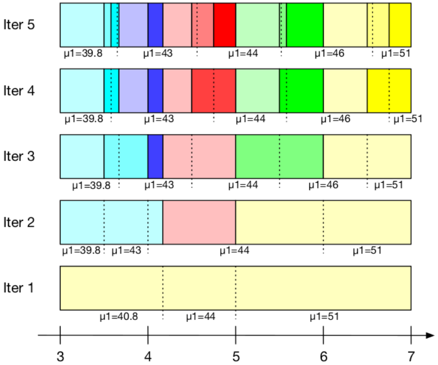




Theorem (Convergence and complexity results)

If $X \cap \mathrm{dom}(V) \subset \mathbb{R}^+$ is contained in a ball of diameter $M \in \mathbb{R}^+$ and $x \to c^\top x + V(x)$ is Lipschitz with constant L then the partition based method finds an ε-solution in at most $\left(\frac{LM}{\varepsilon} + 1\right)^n$ iterations.

Numerical Results - LandS



Iter	x_1	x_2	x_3	x_4
1	0.833	3.000	4.167	4.000
2	2.500	3.000	3.500	3.000
3	1.833	4.000	3.667	2.500
4	2.000	4.167	3.583	2.250
5	1.917	4.083	3.625	2.375
6	1.875	4.042	3.646	2.438

Iter	LB	UB	Gap
1	378.667	382.711	1.0567%
2	380.122	381.100	0.2567%
3	380.601	380.844	0.0640%
4	380.842	380.893	0.0007%
5	380.843	380.856	0.0004%
6	380.844	380.847	0.0002%

Results given by GAPM for LandS problem[2]

[2] illustration from Ramirez-Pico and Moreno

Numerical Results - ProdMix

k	x_k	z_L^k	z_U^k	Gap	$\lvert\mathcal{P}_k^{\max}\rvert$
1	$(1333.33, 66.67)$	-18666.67	-16939.71	9.3%	4
2	$(1441.41, 59.57)$	-17873.01	-17383.73	2.7%	9
3	$(1399.05, 57.91)$	-17789.88	-17659.19	0.74%	16
4	$(1379.98, 56.64)$	-17744.67	-17708.00	0.20%	25
5	$(1371.36, 55.71)$	-17718.96	-17709.05	0.056%	36
6	$(1375.55, 56.21)$	-17713.74	-17711.37	0.013%	49

Table: Results for problem Prod-Mix

To compare our approach with SAA, we solved the same problem 100 times, each with 10 000 scenarios randomly drawn, yielding a 95% confidence interval centered in -17711, with radius 2.2.

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Synthesis of local and uniform quantization results

	$\boldsymbol{W}$	$(\boldsymbol{T}, \boldsymbol{h})$	$\boldsymbol{q}$
Local	$\emptyset$	$\mathcal{R}_x$	$\mathcal{N}(P_x)$
Uniform	$\emptyset$	$\emptyset$	$\bigwedge_{\sigma \in \mathcal{C}(P,\pi)} \mathcal{N}_\sigma$

Stochastic cost and recourse

- We have shown a local exact quantization result for random $\boldsymbol{T}, \boldsymbol{h}$, and deterministic q, W.
- If $\boldsymbol{q}$ and $\boldsymbol{W}$ are finitely supported random variable:
  1. compute an exact quantization $\mathcal{N}_\xi$ for every element of the support;
  2. take the common refinement.

We have seen that we can deal with non-finitely supported $\boldsymbol{q}$ through the chamber complexes.

➥ Can we do the same here ?

Stochastic cost and recourse

- We have shown a local exact quantization result for random $\boldsymbol{T}, \boldsymbol{h}$, and deterministic q, W.
- If $\boldsymbol{q}$ and $\boldsymbol{W}$ are finitely supported random variable:
  1. compute an exact quantization $\mathcal{N}_\xi$ for every element of the support;
  2. take the common refinement.

We have seen that we can deal with non-finitely supported $\boldsymbol{q}$ through the chamber complexes.

➥ Can we do the same here ?

Adapted partition for general q

We define coupling constraint and fiber for the dual.

$$D_q := \left\{ \lambda \in \mathbb{R}^\ell \quad \middle| \quad W^\top \lambda \leqslant q \right\}$$
$$\Delta := \left\{ (\lambda, q) \in \mathbb{R}^\ell \times \mathbb{R}^m \quad \middle| \quad W^\top \lambda \leqslant q \right\}$$
$$\mathcal{R}_{x,q} := \left\{ E_{N,x} \quad \middle| \quad N \in \mathcal{N}(D_q) \right\}$$

Recall that $q \mapsto \mathcal{N}(D_q)$ is piecewise constant on $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q})$ and so is $\mathcal{R}_{x,q}$.
➡ we can take the common refinement of a finite number of $\mathcal{R}_{x,q}$!!

More precisely:

- The chamber complex $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q}) = \Sigma\text{-fan}(W)$[3].
- For $S \in \Sigma\text{-fan}(W)$ define $\mathcal{R}_{x,S} := \mathcal{R}_{x,q}$ for any $q \in \mathrm{ri}(S)$.

➡ $\left\{ \mathrm{ri}(S) \times R \,\middle|\, S \in \Sigma\text{-fan}(W), R \in \mathcal{R}_{x,S} \right\}$ is an adapted partition to x.

[3] The well studied secondary fan of W

Adapted partition for general q

We define coupling constraint and fiber for the dual.

$$
\begin{aligned}
D_q &:= \{\lambda \in \mathbb{R}^\ell \quad | \quad W^\top \lambda \leqslant q\} \\
\Delta &:= \{(\lambda, q) \in \mathbb{R}^\ell \times \mathbb{R}^m \quad | \quad W^\top \lambda \leqslant q\} \\
\mathcal{R}_{x,q} &:= \{E_{N,x} \quad | \quad N \in \mathcal{N}(D_q)\}
\end{aligned}
$$

Recall that $q \mapsto \mathcal{N}(D_q)$ is piecewise constant on $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q})$ and so is $\mathcal{R}_{x,q}$.
➥ we can take the common refinement of a finite number of $\mathcal{R}_{x,q}$!!

More precisely:

- The chamber complex $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q}) = \Sigma\text{-fan}(W)$[3].
- For $S \in \Sigma\text{-fan}(W)$ define $\mathcal{R}_{x,S} := \mathcal{R}_{x,q}$ for any $q \in \mathrm{ri}(S)$.

➥ $\{\mathrm{ri}(S) \times R \,|\, S \in \Sigma\text{-fan}(W), R \in \mathcal{R}_{x,S}\}$ is an adapted partition to x.

[3] The well studied secondary fan of W

Adapted partition for general q

We define coupling constraint and fiber for the dual.

$$\begin{aligned} D_q &:= \{\lambda \in \mathbb{R}^\ell \quad | \quad W^\top \lambda \leqslant q\} \\ \Delta &:= \{(\lambda, q) \in \mathbb{R}^\ell \times \mathbb{R}^m \quad | \quad W^\top \lambda \leqslant q\} \\ \mathcal{R}_{x,q} &:= \{E_{N,x} \quad | \quad N \in \mathcal{N}(D_q)\} \end{aligned}$$

Recall that $q \mapsto \mathcal{N}(D_q)$ is piecewise constant on $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q})$ and so is $\mathcal{R}_{x,q}$.
➡ we can take the common refinement of a finite number of $\mathcal{R}_{x,q}$!!

More precisely:

- The chamber complex $\mathcal{C}(\Delta, \pi_\lambda^{\lambda,q}) = \Sigma\text{-fan}(W)$[3].
- For $S \in \Sigma\text{-fan}(W)$ define $\mathcal{R}_{x,S} := \mathcal{R}_{x,q}$ for any $q \in \mathrm{ri}(S)$.

➡ $\{\mathrm{ri}(S) \times R \mid S \in \Sigma\text{-fan}(W), R \in \mathcal{R}_{x,S}\}$ is an adapted partition to x.

[3] The well studied secondary fan of W

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Dual problem

$$V(x) := \mathbb{E}\begin{bmatrix} \inf_{y} & \boldsymbol{c}^\top y \\ \text{s.t.} & Ax + By \leqslant b \end{bmatrix} = \mathbb{E}\big[\inf_{y \in P_x} \boldsymbol{c}^\top y\big]$$

where $P_x = \{x \mid Ax + By \leqslant b\}$

$$V(x) := \mathbb{E}\begin{bmatrix} \sup_{\mu} & (Ax - b)^\top \mu \\ \text{s.t.} & B^\top \mu + \boldsymbol{c} = 0 \\ & \mu \geqslant 0 \end{bmatrix} = \mathbb{E}\big[\sup_{\mu \in D_{\boldsymbol{c}}} (Ax - b)^\top \mu\big]$$

where $D_c = \{\mu \mid B^\top \mu + c = 0, \mu \geqslant 0\}$

Fiber Polyhedron

Minkowski sum :

$$E + F = \{x + x' \,|x \in E, x' \in F\}$$

Definition

The *fiber polyhedron* E of the bundle $(D_c)_{c \in \mathrm{supp}(\boldsymbol{c})}$ is the Minkowsky integral of all the fiber at c when c varies according to its probability distribution:

$$E := \int D_c \mathbb{P}(dc) = \Big\{ \int \mu(c)\mathbb{P}(dc) \,|\, \mu(c) \in D_c \quad a.s.,\ \mu \in L_\infty(\mathbb{R}^m, \mathbb{R}^l) \Big\}$$

$$V(x) = \mathbb{E}\big[\sup_{\mu \in D_{\boldsymbol{c}}} (Ax - b)^\top \mu \big]$$

$$= \begin{cases} \sup_{\mu(\cdot)} & (Ax-b)^\top \mathbb{E}\big[\mu(\boldsymbol{c})\big] \\ \text{s.t.} & \mu(\boldsymbol{c}) \in D_{\boldsymbol{c}} \ \text{ a.s.} \end{cases}$$

Fiber Polyhedron

Minkowski sum :

$$E + F = \{x + x' \,|x \in E, x' \in F\}$$

Definition

The *fiber polyhedron* E of the bundle $(D_c)_{c \in \mathrm{supp}(\boldsymbol{c})}$ is the Minkowsky integral of all the fiber at c when c varies according to its probability distribution:

$$E := \int D_c \mathbb{P}(dc) = \Big\{ \int \mu(c)\mathbb{P}(dc) \,|\, \mu(c) \in D_c \quad a.s.,\ \mu \in L_\infty(\mathbb{R}^m, \mathbb{R}^l) \Big\}$$

$$V(x) = \mathbb{E}\big[\sup_{\mu \in D_{\boldsymbol{c}}} (Ax - b)^\top \mu \big]$$

$$= \begin{cases} \sup_{\mu(\cdot)} & (Ax - b)^\top \mathbb{E}\big[\mu(\boldsymbol{c})\big] \\ \text{s.t.} & \mu(\boldsymbol{c}) \in D_{\boldsymbol{c}} \ \text{ a.s.} \end{cases}$$

Fiber Polyhedron

Minkowski sum :

$$E + F = \{x + x' \,|x \in E, x' \in F\}$$

Definition

The *fiber polyhedron* E of the bundle $(D_c)_{c \in \mathrm{supp}(\boldsymbol{c})}$ is the Minkowsky integral of all the fiber at c when c varies according to its probability distribution:

$$E := \int D_c \mathbb{P}(dc) = \left\{ \int \mu(c)\mathbb{P}(dc) \,|\, \mu(c) \in D_c \quad a.s., \ \mu \in L_\infty(\mathbb{R}^m, \mathbb{R}^l) \right\}$$

$$V(x) = \mathbb{E}\Big[\sup_{\mu \in D_{\boldsymbol{c}}} (Ax - b)^\top \mu \Big]$$

$$= \begin{cases} \sup_{\mu(\cdot)} & (Ax - b)^\top \mathbb{E}\big[\mu(\boldsymbol{c})\big] \\ \text{s.t.} & \mathbb{E}\big[\mu(\boldsymbol{c})\big] \in E \end{cases}$$

Fiber Polyhedron

Minkowski sum :

$$E + F = \{x + x' \,|x \in E, x' \in F\}$$

Definition

The *fiber polyhedron* E of the bundle $(D_c)_{c \in \mathrm{supp}(\boldsymbol{c})}$ is the Minkowsky integral of all the fiber at c when c varies according to its probability distribution:

$$E := \int D_c \mathbb{P}(dc) = \Big\{ \int \mu(c)\mathbb{P}(dc) \,|\, \mu(c) \in D_c \quad a.s.,\ \mu \in L_\infty(\mathbb{R}^m, \mathbb{R}^l) \Big\}$$

$$\begin{aligned} V(x) &= \mathbb{E}\big[\sup_{\mu \in D_{\boldsymbol{c}}} (Ax - b)^\top \mu\big] \\ &= \begin{cases} \sup_{\mu(\cdot)} & (Ax - b)^\top \mathbb{E}\big[\mu(\boldsymbol{c})\big] \\ \text{s.t.} & \mathbb{E}\big[\mu(\boldsymbol{c})\big] \in E \end{cases} \\ &= \sup (Ax - b)^\top \lambda \end{aligned}$$

The Fiber Polyhedron is a finite Minkowski sum

Theorem

There exists a chamber complex $\mathcal{R}$ depending on A such that

$$E = \int D_c \mathbb{P}(dc) = \sum_{R \in \mathcal{R}} \check{p}_R D_{\check{c}_R}$$

where $\check{p}_R := \mathbb{P}\big[\boldsymbol{c} \in \mathrm{ri}(R)\big]$ *and* $\check{c}_R := \mathbb{E}\big[\boldsymbol{c} \,|\, \boldsymbol{c} \in \mathrm{ri}(R)\big]$.

Alternative proof of the quantization result

$$V(x) = \sigma_E(Ax - b) = \sum_{R \in \mathcal{R}} \check{p}_R \sigma_{D_{\check{c}_R}}(Ax - b) = \sum_{R \in \mathcal{R}} \check{p}_R \min_{y \in P_x} \check{c}_R^\top y$$

The Fiber Polyhedron is a finite Minkowski sum

Theorem

There exists a chamber complex $\mathcal{R}$ depending on A such that

$$E = \int D_c \mathbb{P}(dc) = \sum_{R \in \mathcal{R}} \check{p}_R D_{\check{c}_R}$$

where $\check{p}_R := \mathbb{P}\big[\boldsymbol{c} \in \mathrm{ri}(R)\big]$ and $\check{c}_R := \mathbb{E}\big[\boldsymbol{c} \,|\, \boldsymbol{c} \in \mathrm{ri}(R)\big]$.

Alternative proof of the quantization result

$$V(x) = \sigma_E(Ax - b) = \sum_{R \in \mathcal{R}} \check{p}_R \sigma_{D_{\check{c}_R}}(Ax - b) = \sum_{R \in \mathcal{R}} \check{p}_R \min_{y \in P_x} \check{c}_R^\top y$$

Nested Fiber Polyhedra for Multistage

$$V_t(x_{t-1}) = \mathbb{E}\left[\begin{array}{l}\min\limits_{x_t \in \mathbb{R}^{n_t}} \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t. } A_t x_t + B_t x_{t-1} \leqslant b_t\end{array}\right]$$

Definition

We define by induction the following nested fiber polyhedra

$$\begin{aligned}
D_{t,c_t} &:= \{\mu_t | \mu_t \geqslant 0, A_t^\top \mu_t + c_t = 0\} && \forall t \in [T] \\
F_{T,c_T} &:= D_{T,c_T} \\
E_t &:= \mathbb{E}\left[F_{t,\boldsymbol{c}_t}\right] && \forall t \in [T] \\
F_{t,c_t} &:= \{(\mu_t, \lambda_{[t+1:T]}) \,|\, \mu_t \in D_{t,c_t + B_{t+1}^\top \lambda_{t+1}},\ \lambda_{[t+1:T]} \in E_{t+1}\} && \forall t \in [T-1]
\end{aligned}$$

$$V_t(x_{t-1}) = \sigma_{E_t}(B_t x_{t-1} - b_t, -b_{[t+1:T]})$$

Nested Fiber Polyhedra for Multistage

$$V_t(x_{t-1}) = \mathbb{E}\left[\begin{array}{l} \min\limits_{x_t \in \mathbb{R}^{n_t}} \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{ s.t. } A_t x_t + B_t x_{t-1} \leqslant b_t \end{array}\right]$$

Definition

We define by induction the following nested fiber polyhedra

$$D_{t,c_t} := \{\mu_t | \mu_t \geqslant 0, A_t^\top \mu_t + c_t = 0\} \qquad \forall t \in [T]$$

$$F_{T,c_T} := D_{T,c_T}$$

$$E_t := \mathbb{E}\left[F_{t,\boldsymbol{c}_t}\right] \qquad \forall t \in [T]$$

$$F_{t,c_t} := \{(\mu_t, \lambda_{[t+1:T]}) \,|\, \mu_t \in D_{t,c_t + B_{t+1}^\top \lambda_{t+1}},\ \lambda_{[t+1:T]} \in E_{t+1}\} \quad \forall t \in [T-1]$$

$$V_t(x_{t-1}) = \sigma_{E_t}(B_t x_{t-1} - b_t, -b_{[t+1:T]})$$

2-time scale MSLP reduced to Quadratic Problem

We consider a MSLP where the dynamic depends on parameters p we have to optimize

$$\begin{aligned}
\min_{p\in\mathbb{R}^m,(\boldsymbol{x}_t)\in\mathbb{R}^{n_t}} \quad & q^\top p + \mathbb{E}\Big[\sum_{t=1}^{T} \boldsymbol{c}_t^\top \boldsymbol{x}_t\Big] \\
\text{s.t.} \quad & Dp \leqslant d \\
& A_t\boldsymbol{x}_t + B_t\boldsymbol{x}_{t-1} + C_t p \leqslant h_t \quad \text{a.s.} \qquad \forall t\in[T] \\
& \boldsymbol{x}_t \preceq \sigma(\boldsymbol{c}_1,\cdots,\boldsymbol{c}_t) \qquad \forall t\in[T]
\end{aligned}$$

If we know the fiber polyhedron, it reduces to a finite dimensional quadratic problem

$$\begin{aligned}
& \min_{p\in\mathbb{R}^m} q^\top p + \sup_{(\lambda_t)_{t\in[T]}} \sum_{t=1}^{T} (C_t p - h_t)^\top \lambda_t \\
& \text{s.t. } Dp \leqslant d \\
& \qquad (\lambda_1,\cdots,\lambda_T) \in E_1
\end{aligned}$$

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

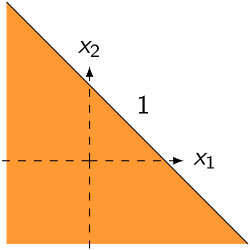
Linear Programming and polyhedra

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leqslant b \end{aligned}$$

Example: $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ \\ \\ \end{pmatrix}$$

$x_1 + x_2 \leqslant 1$



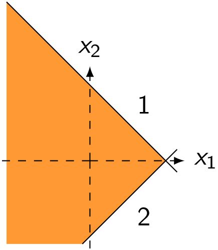
Linear Programming and polyhedra

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leqslant b \end{aligned}$$

Example: $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ \\ \\ \\ \\ \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ \\ \\ \\ \end{pmatrix}$$

$$x_1 + x_2 \leqslant 1$$
$$x_1 - x_2 \leqslant 1$$

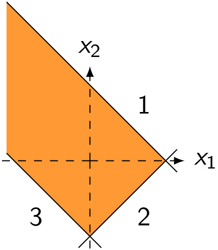


Linear Programming and polyhedra

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & c^\top x \\ \text{s.t.} & Ax \leqslant b \end{array}$$

Example: $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \\ \\ \\ \end{pmatrix} \quad \begin{aligned} x_1 + x_2 &\leqslant 1 \\ x_1 - x_2 &\leqslant 1 \\ -x_1 - x_2 &\leqslant 1 \end{aligned}$$



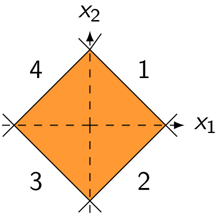
Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

Example: $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \\ \\ \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 \\ x_1 - x_2 &\leqslant 1 \\ -x_1 - x_2 &\leqslant 1 \\ -x_1 + x_2 &\leqslant 1 \end{aligned}$$



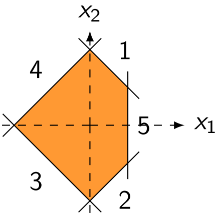
Linear Programming and polyhedra

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leqslant b \end{aligned}$$

Example: $P = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ \\ \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 \\ x_1 - x_2 &\leqslant 1 \\ -x_1 - x_2 &\leqslant 1 \\ -x_1 + x_2 &\leqslant 1 \\ x_1 &\leqslant 0.5 \end{aligned}$$



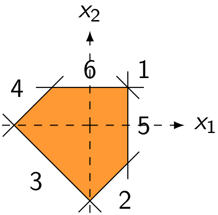
Linear Programming and polyhedra

$$\min_{x \in \mathbb{R}^n} \quad c^\top x$$
$$\text{s.t.} \quad Ax \leqslant b$$

Example: $P = \{x \in \mathbb{R}^n \,|\, Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 \\ x_1 - x_2 &\leqslant 1 \\ -x_1 - x_2 &\leqslant 1 \\ -x_1 + x_2 &\leqslant 1 \\ x_1 &\leqslant 0.5 \\ x_2 &\leqslant 0.5 \end{aligned}$$



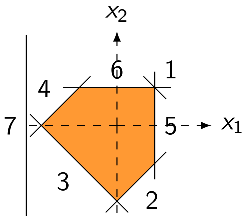
Linear Programming and polyhedra

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leqslant b \end{aligned}$$

Example: $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$$A = \begin{pmatrix} 1 & 1 \\ & \\ & \\ & \\ & \\ & \\ & \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \\ -1.2 \end{pmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leqslant 1 \\ x_1 - x_2 &\leqslant 1 \\ -x_1 - x_2 &\leqslant 1 \\ -x_1 + x_2 &\leqslant 1 \\ x_1 &\leqslant 0.5 \\ x_2 &\leqslant 0.5 \\ x_1 &\geqslant -1.2 \end{aligned}$$



Contents

5. Explicit formulas for general distributions

6. Details on GAPM
   - Recalls on APM
   - A novel APM algorithm
   - Extension of GAPM to general costs

7. Nested fiber polyhedra

8. Polyhedral toolbox for stochastic optimizers
   - Active constraints
   - Link with regular subdivisions
   - Correspondences for parametric linear programming
   - Correspondences for 2SLP

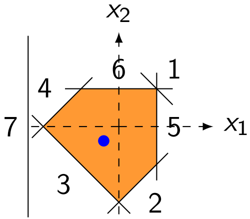
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, *the collection of sets of active constraints as :*

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \emptyset$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, \qquad \}$$

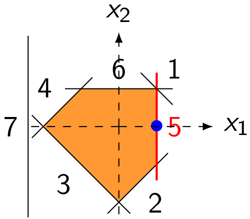
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{5\}$

To ease the notation, we write:

$\mathcal{I}(A, b) = \{\emptyset, 5, \qquad \}$

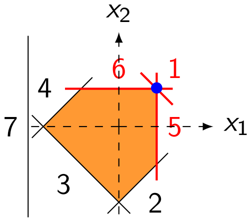
Active constraints

Definition

We denote by $\mathcal{I}(A,b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A,b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{1, 5, 6\}$

To ease the notation, we write:

$\mathcal{I}(A,b) = \{\emptyset, 5, 156, \qquad\qquad\qquad\qquad \}$

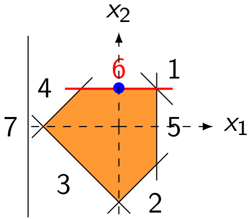
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{6\}$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, \qquad\qquad\qquad \}$$

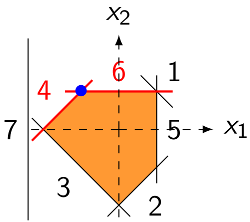
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{4, 6\}$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, 46, \qquad \}$$

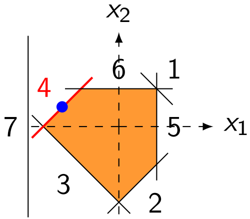
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{4\}$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, 46, 4, \qquad \}$$

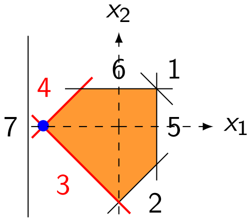
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{3, 4\}$

To ease the notation, we write:

$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, 46, 4, 34, \quad \}$

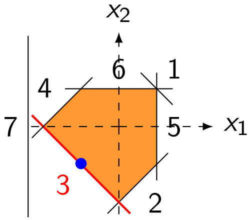
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{3\}$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, 46, 4, 34, 3, \qquad \}$$

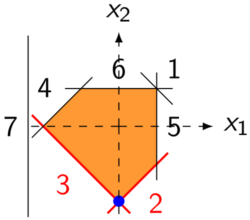
Active constraints

Definition

We denote by $\mathcal{I}(A, b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A, b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{2, 3\}$

To ease the notation, we write:

$$\mathcal{I}(A, b) = \{\emptyset, 5, 156, 6, 46, 4, 34, 3, 23, \qquad \}$$

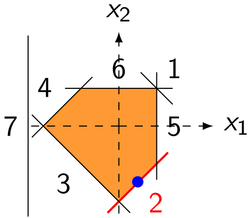
Active constraints

Definition

We denote by $\mathcal{I}(A,b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A,b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{2\}$

To ease the notation, we write:

$$\mathcal{I}(A,b) = \{\emptyset, 5, 156, 6, 46, 4, 34, 3, 23, 2, \quad \}$$

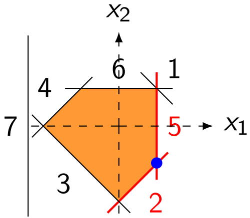
Active constraints

Definition

We denote by $\mathcal{I}(A,b)$, the collection of sets of active constraints as :

$$\mathcal{I}(A,b) = \{I_{A,b}(x) \mid Ax \leqslant b\}$$

with $I_{A,b}(x) := \{i \in [q] \mid A_i x = b_i\}$



$P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$

$I_{A,b}(x) = \{2, 5\}$

To ease the notation, we write:

$$\mathcal{I}(A,b) = \left\{\emptyset, 5, 156, 6, 46, 4, 34, 3, 23, 2, 25\right\}$$

Faces

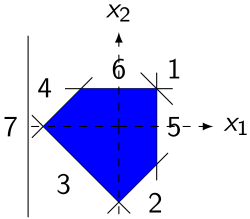
Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$

Example for $I = \emptyset$



P, P^I

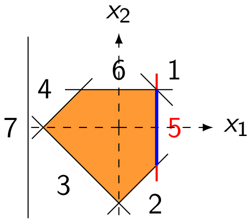
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{5\}$



P, P^I

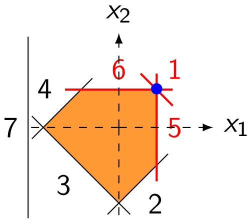
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{1, 5, 6\}$



P, P^I

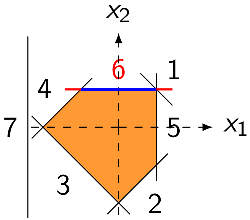
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{6\}$



P, P^I

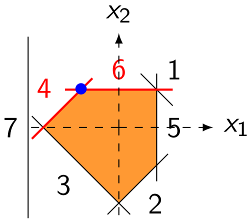
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \,|\, A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{4, 6\}$



P, P^I

Faces

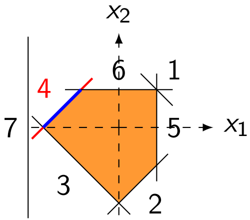
Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \,|\, A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$

Example for $I = \{4\}$



P, P^I

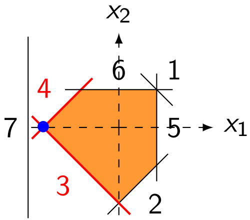
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{3, 4\}$



P, P^I

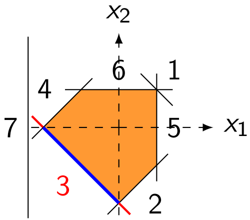
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{3\}$



P, P^I

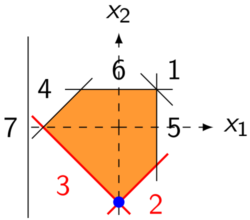
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{2, 3\}$



P, P^I

Faces

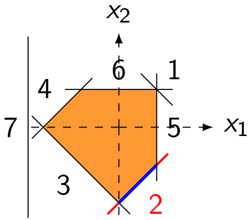
Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$

Example for $I = \{2\}$



P, P^I

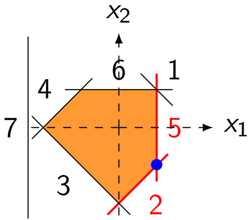
Faces

Definition

Let $I \in \mathcal{I}(A, b)$, we denote by P^I the face of P such that:

$$P^I = \{x \in P \mid A_I x = b_I\}$$

We have $\dim(P^I) = n - \mathrm{rg}(A_I)$
Example for $I = \{2, 5\}$

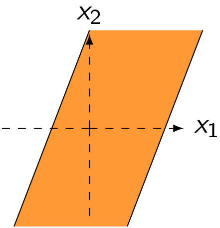


P, P^I

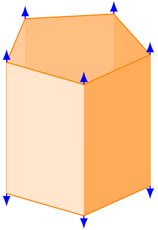
Lineality space, vertices and bases

Definition (Lineality space)

$\mathrm{Lin}(C) := \{u \in C \mid \forall t \in \mathbb{R},\ \forall x \in c,\ x + tu \in C\}.$



If
$P = \{x \in \mathbb{R}^n | Ax \leqslant b\}$,
then $\mathrm{Lin}(P) = \mathrm{Ker}(A)$



Definition (Bases and vertices)

A basis B is a subset of $[p]$ such that $A_B = (A_{i,j})_{i \in B, 1 \leqslant j \leqslant n}$ is invertible.
A vertex of P is a face of dimension 0. $\mathrm{Vert}(P)$ is the set of vertices.

$\mathrm{Vert}(P) \neq \emptyset \Leftrightarrow A$ admits at least one basis $\Leftrightarrow \mathrm{rg}(A) = n \Leftrightarrow \mathrm{Lin}(P) = \{0\}$

We make this assumption without loss of generality.

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Link with regular subdivisions

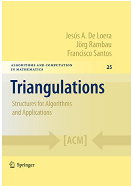
Definition (DLRS10)

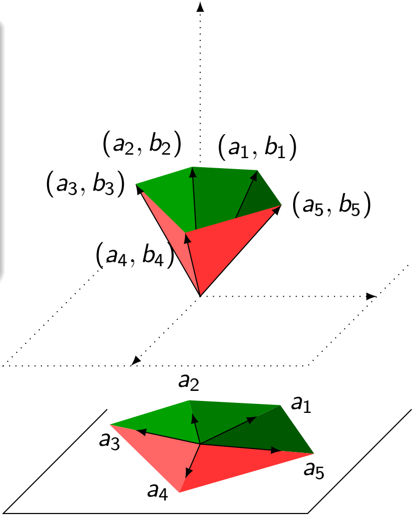
$$\mathcal{S}(A^\top, b) := \{I_F \mid F \in \mathcal{F}_{\text{low}}\left(LC_{A^\top,b}\right)\}$$

$$LC_{A^\top,b} := \operatorname{Cone}\left(\begin{pmatrix} a_i \\ b_i \end{pmatrix}_{i\in[q]}\right)$$

$$I_F := \{i \in [q] \mid (a_i, b_i) \in F\}.$$

$$\mathcal{S}(A^\top, b) = \mathcal{I}(A, b)$$





$$\mathcal{I}(W^\top q) = \mathcal{I}_{com} \cup \{\{5\}, \{4,5\}, \{1,5\}\}$$

Link with regular subdivisions

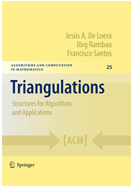
Definition (DLRS10)

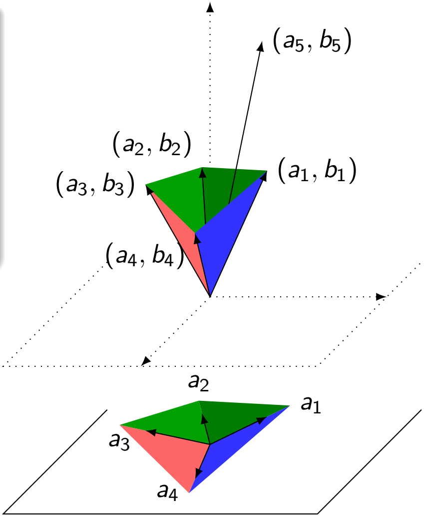
$$\mathcal{S}(A^\top, b) := \{I_F \mid F \in \mathcal{F}_{\mathrm{low}}(LC_{A^\top,b})\}$$

$$LC_{A^\top,b} := \mathrm{Cone}\left(\begin{pmatrix} a_i \\ b_i \end{pmatrix}_{i\in[q]}\right)$$

$$I_F := \{i \in [q] \mid (a_i, b_i) \in F\}.$$

$$\mathcal{S}(A^\top, b) = \mathcal{I}(A, b)$$





$$\mathcal{I}(W^\top q) = \mathcal{I}_{com} \cup \{\{1, 4\}\}$$

Link with regular subdivisions

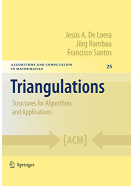
Definition (DLRS10)

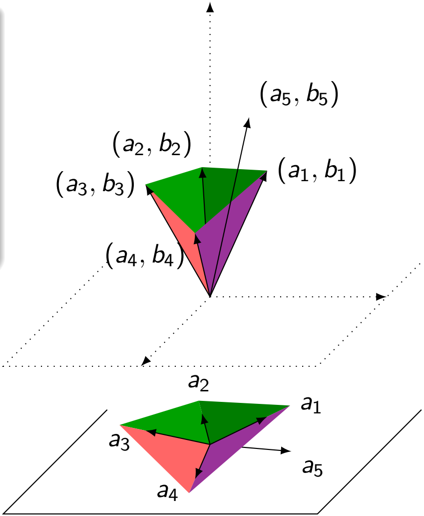
$$\mathcal{S}(A^\top, b) := \{ I_F \mid F \in \mathcal{F}_{\text{low}}(LC_{A^\top,b}) \}$$

$$LC_{A^\top,b} := \mathrm{Cone}\left(\begin{pmatrix} a_i \\ b_i \end{pmatrix}_{i \in [q]} \right)$$

$$I_F := \{ i \in [q] \mid (a_i, b_i) \in F \}.$$

$$\mathcal{S}(A^\top, b) = \mathcal{I}(A, b)$$





$$\mathcal{I}(W^\top, q) = \mathcal{I}_{com} \cup \{\{1, 4, 5\}\}$$

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

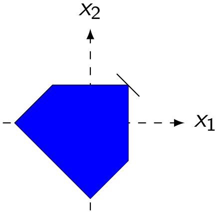
8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Correspondences

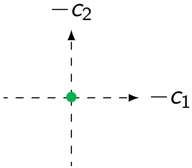
$$\begin{tikzcd}
& F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in \mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in \mathrm{ri}(F)"] & \\
I \in \mathcal{I}(A,b), \subset \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & N \in \mathcal{N}(P), \subset \arrow[ul, bend right, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \emptyset$



F



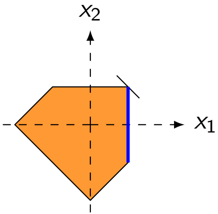
N

Correspondences

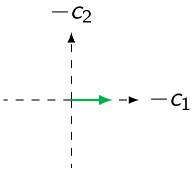
$$\begin{tikzcd}[column sep=small]
& \boxed{F \in \mathcal{F}(P)\backslash\{\emptyset\}, \supset} \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"'] & \\
\boxed{I \in \mathcal{I}(A,b), \subset} \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \mathcal{N}(P), \subset} \arrow[ul, bend left, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{5\}$



F



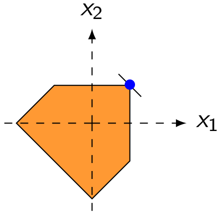
N

Correspondences

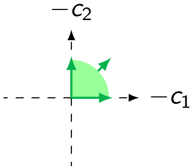
$$\begin{tikzcd}[column sep=small]
 & {\color{blue}F \in \mathcal{F}(P)}\setminus\{\emptyset\}, \supset \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"] & \\
{\color{red}I \in \mathcal{I}(A, b)}, \subset \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & {\color{green}N \in \mathcal{N}(P)}, \subset \arrow[ul, bend right, "\arg\min_{x\in P} c^\top x \text{ for } c\in-\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top\in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{1, 5, 6\}$



F



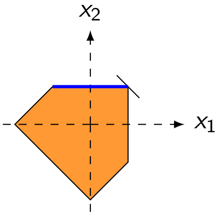
N

Correspondences

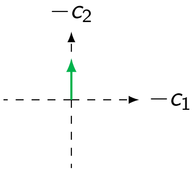
$$\begin{tikzcd}[column sep=small]
 & F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset \arrow[dl, shift right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, shift left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"] & \\
I \in \mathcal{I}(A,b), \subset \arrow[ur, shift right, "P^I"'] \arrow[rr, shift left, "\mathrm{Cone}(A_I^\top)"] & & N \in \mathcal{N}(P), \subset \arrow[ul, shift left, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, shift left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{6\}$



F



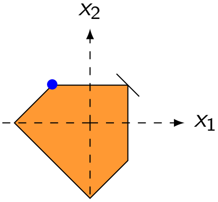
N

Correspondences

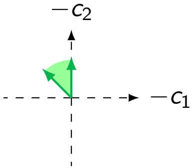
$$
\begin{tikzcd}
 & \color{blue}{F \in \mathcal{F}(P)}\setminus\{\emptyset\}, \supset \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x \in \mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x \in \mathrm{ri}(F)"'] & \\
\color{red}{I \in \mathcal{I}(A,b)}, \subset \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \color{green}{N \in \mathcal{N}(P)}, \subset \arrow[ll, bend left, "\{i \in \mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"] \arrow[ul, bend right, "\arg\min_{x \in P} c^\top x \text{ for } c \in -\mathrm{ri}(N)"']
\end{tikzcd}
$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{4, 6\}$



F



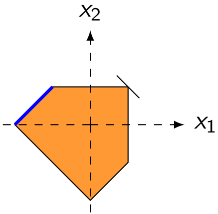
N

Correspondences

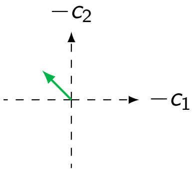
$$
\begin{tikzcd}[column sep=small]
 & {\color{blue}F \in \mathcal{F}(P)}\setminus\{\emptyset\}, \supset \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"'] & \\
{\color{red}I \in \mathcal{I}(A,b)}, \subset \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & {\color{green}N \in \mathcal{N}(P)}, \subset \arrow[ul, bend left, "\arg\min_{x\in P} c^\top x \text{ for } c\in-\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top\in N\}"]
\end{tikzcd}
$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{4\}$



F



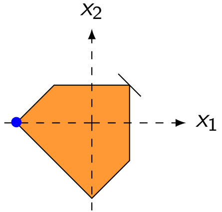
N

Correspondences

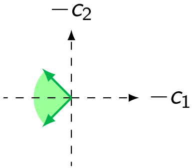
$$\begin{tikzcd}
& \boxed{F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset} \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in \mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in \mathrm{ri}(F)"] & \\
\boxed{I \in \mathcal{I}(A,b), \subset} \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \mathcal{N}(P), \subset} \arrow[ul, bend right, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{3, 4\}$



F



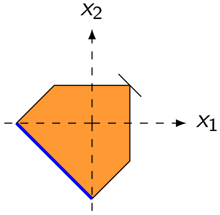
N

Correspondences

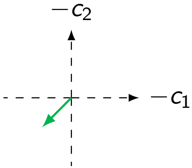
$$
\begin{tikzcd}
& F \in \mathcal{F}(P)\backslash\{\emptyset\}, \supset \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"'] & \\
I \in \mathcal{I}(A,b), \subset \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & N \in \mathcal{N}(P), \subset \arrow[ul, bend left, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top\in N\}"]
\end{tikzcd}
$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{3\}$



F



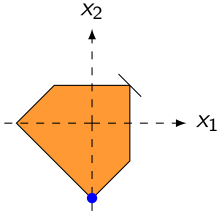
N

Correspondences

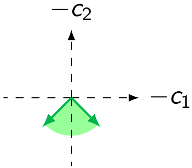
$$\begin{tikzcd}
& \boxed{F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset} \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in \mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in \mathrm{ri}(F)"] & \\
\boxed{I \in \mathcal{I}(A,b), \subset} \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \mathcal{N}(P), \subset} \arrow[ul, bend right, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{2, 3\}$



F



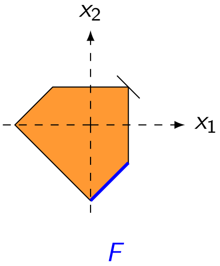
N

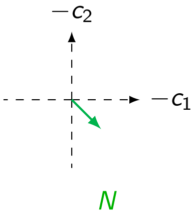
Correspondences

$$\begin{tikzcd}
 & \boxed{F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset} \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in \mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in \mathrm{ri}(F)"] & \\
\boxed{I \in \mathcal{I}(A,b), \subset} \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \mathcal{N}(P), \subset} \arrow[ul, bend right, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \operatorname{supp}\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{2\}$



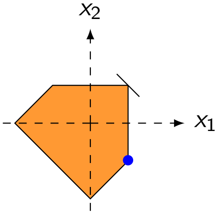


Correspondences

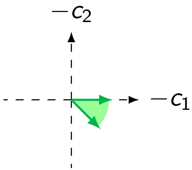
$$\begin{tikzcd}[column sep=small]
 & \boxed{F \in \mathcal{F}(P)\setminus\{\emptyset\}, \supset} \arrow[dl, bend right, "I_{A,b}(x) \text{ for } x\in\mathrm{ri}(F)"'] \arrow[dr, bend left, "N_P(x) \text{ for } x\in\mathrm{ri}(F)"] & \\
\boxed{I \in \mathcal{I}(A,b), \subset} \arrow[ur, bend right, "P^I"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \mathcal{N}(P), \subset} \arrow[ul, bend left, "\arg\min_{x\in P} c^\top x \text{ for } c\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$I = \{2, 5\}$

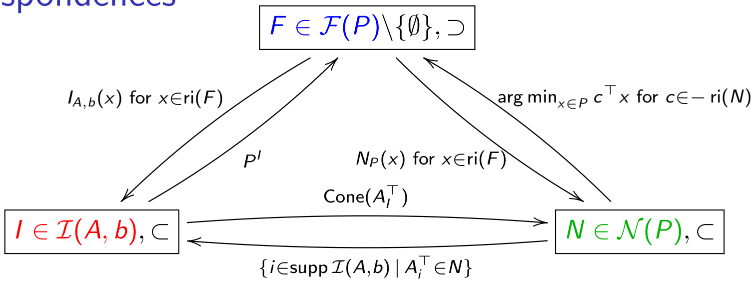


F



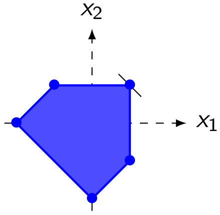
N

Correspondences

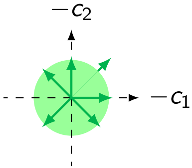


$$\mathrm{rg}(A_I) = n - \dim(F) = \dim(N)$$

$\mathcal{I}(A, b) =$
$\{\emptyset, 5, 156, 6, 46, 4,$
$34, 3, 23, 2, 25\}$



$\mathcal{F}(P)$

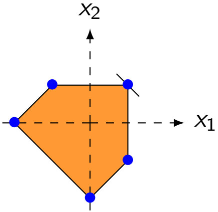


$\mathcal{N}(P)$

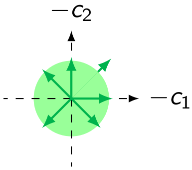
Correspondences

$$\begin{tikzcd}
 & \boxed{v \in \mathrm{Vert}(P)} \arrow[dl, bend right, "I_{A,b}(v)"'] \arrow[dr, bend left, "N_P(v)"] & \\
\boxed{I \in \overline{\mathcal{I}(A,b)}} \arrow[ur, bend right, "A_{B_I}^{-1} b_{B_I}"'] \arrow[rr, bend left, "\mathrm{Cone}(A_I^\top)"] & & \boxed{N \in \overline{\mathcal{N}(P)}} \arrow[ul, bend right, "\arg\min_{x \in P} c^\top x \text{ for } c \in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i \in \mathrm{supp}\,\mathcal{I}(A,b) \mid A_i^\top \in N\}"]
\end{tikzcd}$$

$\overline{\mathcal{I}(A,b)} =$
$\{156, 46, 34, 23, 25\}$



$\mathrm{Vert}(P)$



$\overline{\mathcal{N}(P)}$

Contents

5 Explicit formulas for general distributions

6 Details on GAPM
- Recalls on APM
- A novel APM algorithm
- Extension of GAPM to general costs

7 Nested fiber polyhedra

8 Polyhedral toolbox for stochastic optimizers
- Active constraints
- Link with regular subdivisions
- Correspondences for parametric linear programming
- Correspondences for 2SLP

Proof of normal equivalence

$$\mathcal{G}_x := \left\{ G \in \mathcal{F}(P) \,|\, x \in \operatorname{ri}\big(\pi(G)\big) \right\}$$

Let $\sigma \in \mathcal{C}(P,\pi)$, for all $x, x' \in \operatorname{ri}(\sigma)$, we have

$$\mathcal{G}_\sigma := \mathcal{G}_x = \mathcal{G}_{x'}$$

$$\begin{tikzcd}
\boxed{G \in \mathcal{G}_x} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend left, "I_{(T,W),h}(G)"] & & \boxed{F \in \mathcal{F}(P_x)\setminus\{\emptyset\}} \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} \{x\}\times F\subset G"'] \arrow[dl, bend right, "I_{W,h-Tx}(F)"'] \arrow[dr, bend right, "N_{P_x}(F)"'] & \\
& \boxed{I \in \mathcal{I}(W, h - Tx)} \arrow[ul, bend left, "P^I"] \arrow[ur, bend right, "P^I_x"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \mathcal{N}(P_x)} \arrow[ul, bend right, "\cap_{q\in -N} \arg\min_{y\in P_x} q^\top y"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}$$

By the correspondences,

$$\mathcal{I}_\sigma := \mathcal{I}(W, h - Tx) = \mathcal{I}(W, h - Tx') \qquad \mathcal{N}_\sigma := \mathcal{N}(P_x) = \mathcal{N}(P_{x'})$$

Proof of normal equivalence

$$\mathcal{G}_x := \left\{ G \in \mathcal{F}(P) \,\middle|\, x \in \mathrm{ri}\left(\pi(G)\right) \right\}$$

Let $\sigma \in \mathcal{C}(P, \pi)$, for all $x, x' \in \mathrm{ri}(\sigma)$, we have

$$\mathcal{G}_\sigma := \mathcal{G}_x = \mathcal{G}_{x'}$$

$$\begin{tikzcd}
\boxed{G \in \mathcal{G}_x} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend left, "I_{(T,W),h}(G)"] & & \boxed{F \in \mathcal{F}(P_x) \setminus \{\emptyset\}} \arrow[ll, bend right, "G \in \mathcal{F}(P) \text{ minimal s.t. } \{x\} \times F \subset G"'] \arrow[dl, bend right, "I_{W,h-Tx}(F)"'] \arrow[dr, bend right, "N_{P_x}(F)"'] & \\
& \boxed{I \in \mathcal{I}(W, h - Tx)} \arrow[ul, bend left, "P^I"] \arrow[ur, bend right, "P^I_x"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \mathcal{N}(P_x)} \arrow[ul, bend right, "\cap_{q \in -N} \arg\min_{y \in P_x} q^\top y"'] \arrow[ll, bend left, "\{i \in \mathrm{supp}\, \mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}$$

By the correspondences,

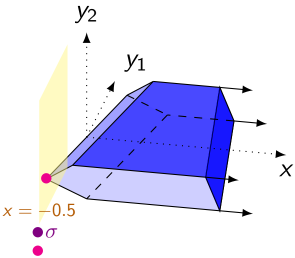
$$\mathcal{I}_\sigma := \mathcal{I}(W, h - Tx) = \mathcal{I}(W, h - Tx') \qquad \mathcal{N}_\sigma := \mathcal{N}(P_x) = \mathcal{N}(P_{x'})$$

Correspondences

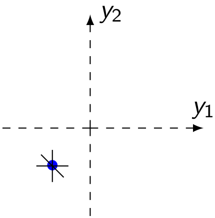
$$
\begin{tikzcd}[column sep=large, row sep=huge]
\boxed{\color{magenta} G \in \mathcal{G}_\sigma, \supset} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend left, "I_{(T,W),h}(G)"] & & \boxed{\color{blue} F \in \mathcal{F}(P_x) \setminus \{\emptyset\}, \supset} \arrow[ll, bend right, "G \in \mathcal{F}(P) \text{ minimal s.t.} \{x\} \times F \subset G"'] \arrow[dl, bend right, "I_{W,h-Tx}(F)"'] \arrow[dr, bend right, "N_{P_x}(F)"'] & \\
& \boxed{\color{red} I \in \mathcal{I}_\sigma, \subset} \arrow[ul, bend left, "P^I"] \arrow[ur, bend right, "P^I_x"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{\color{green} N \in \mathcal{N}_\sigma, \subset} \arrow[ul, bend right, "\arg\min_{y \in P_x} q^\top x \text{ for } q \in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i \in \mathrm{supp}\, \mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}
$$

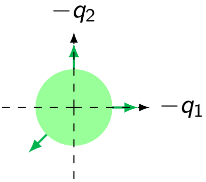
Correspondences

$$\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in - \mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}$$



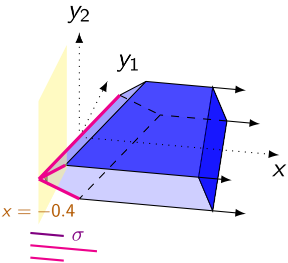
$\overline{\mathcal{I}_\sigma} = \{356\}$



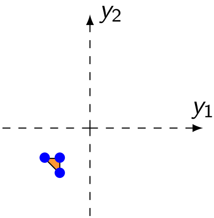


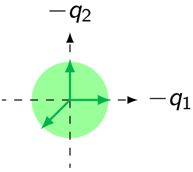
Correspondences

$$\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}$$



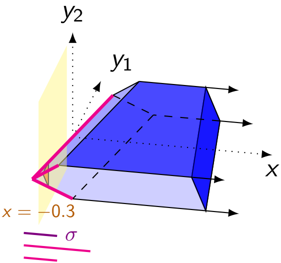
$\overline{\mathcal{I}_\sigma} = \{53, 36, 65\}$



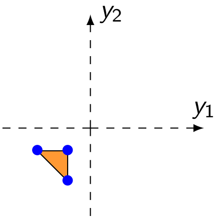


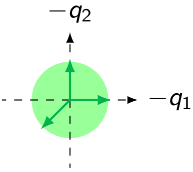
Correspondences

$$
\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P)\text{ minimal s.t.}(x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\operatorname{supp}\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}
$$



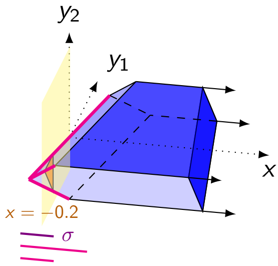
$\overline{\mathcal{I}_\sigma} =$
$\{53, 36, 65\}$



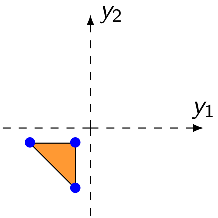


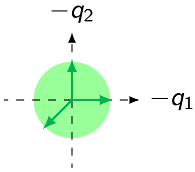
Correspondences

$$\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"']
\end{tikzcd}$$



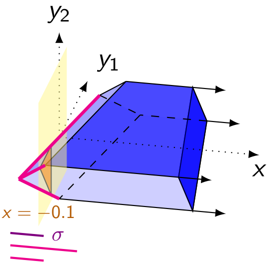
$\overline{\mathcal{I}_\sigma} = \{53, 36, 65\}$



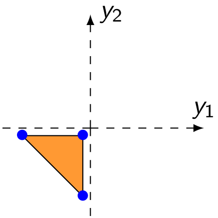


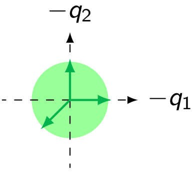
Correspondences

$$
\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend left, "N_{P_x}(v)"] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend left, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in - \mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}
$$

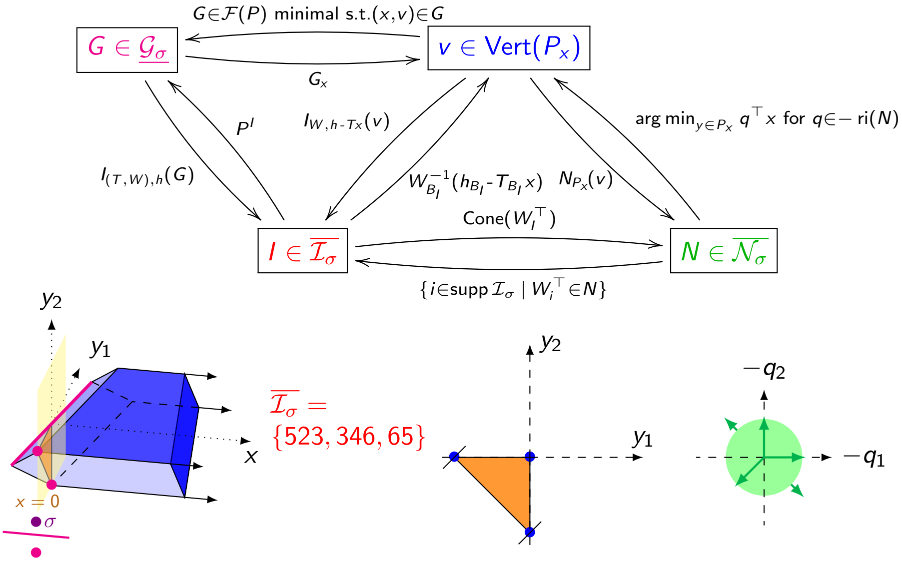


$\overline{\mathcal{I}_\sigma} = \{53, 36, 65\}$

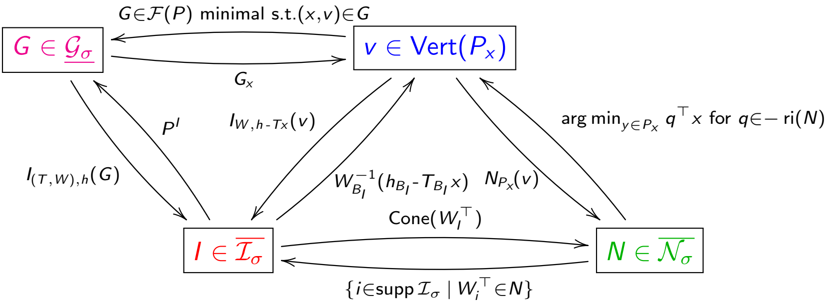


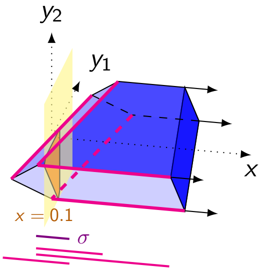


Correspondences

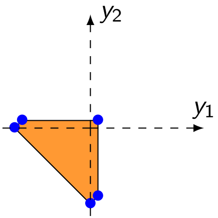


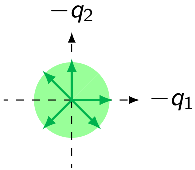
Correspondences





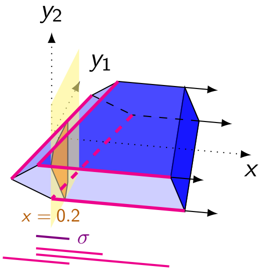
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 65\}$



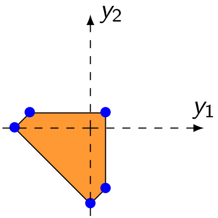


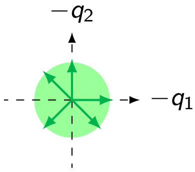
Correspondences

$$
\begin{tikzcd}[column sep=small]
\boxed{G \in \mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & \boxed{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right, "G\in\mathcal{F}(P)\text{ minimal s.t.}(x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend left, "N_{P_x}(v)"] & \\
& \boxed{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \overline{\mathcal{N}_\sigma}} \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, bend left, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"]
\end{tikzcd}
$$

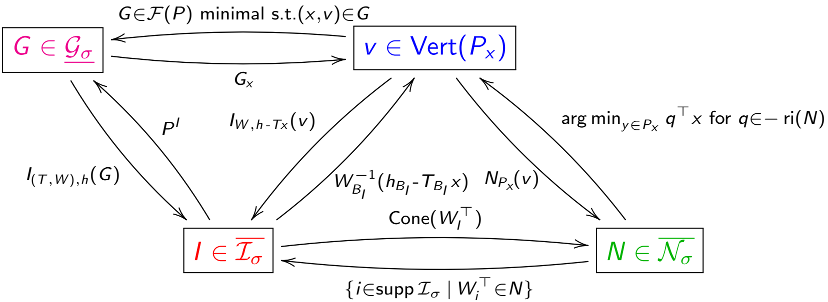


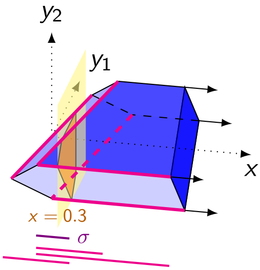
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 65\}$



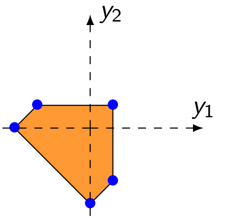


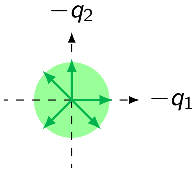
Correspondences





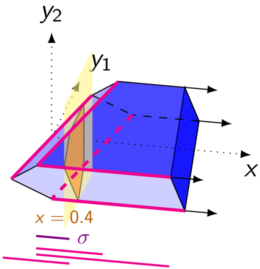
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 65\}$



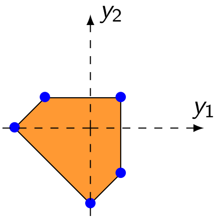


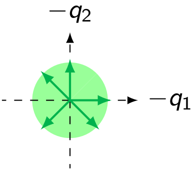
Correspondences

$$\begin{tikzcd}[column sep=large, row sep=huge]
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in-\mathrm{ri}(N)"']
\end{tikzcd}$$



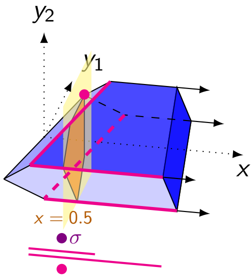
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 65\}$



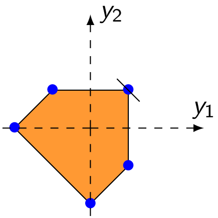


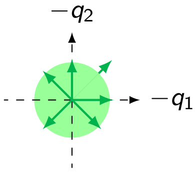
Correspondences

$$\begin{tikzcd}[column sep=large, row sep=huge]
\boxed{G \in \underline{\mathcal{G}_\sigma}} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & \boxed{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& \boxed{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \overline{\mathcal{N}_\sigma}} \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}$$



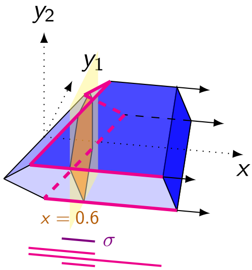
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 615\}$



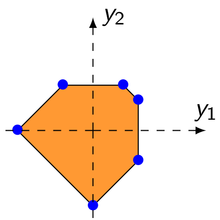


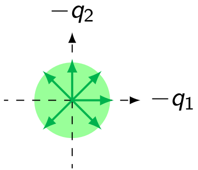
Correspondences

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\boxed{G \in \underline{\mathcal{G}_\sigma}} \arrow[rr, bend right=10, "G_x"'] \arrow[dr, bend right=10, "I_{(T,W),h}(G)"'] & & \boxed{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right=10, "G\in\mathcal{F}(P)\text{ minimal s.t.}(x,v)\in G"'] \arrow[dl, bend right=10, "I_{W,h-Tx}(v)"'] \arrow[dr, bend right=10, "N_{P_x}(v)"'] & \\
& \boxed{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, bend right=10, "P^I"'] \arrow[ur, bend right=10, "W_{B_I}^{-1}(h_{B_I}-T_{B_I}x)"'] \arrow[rr, bend left=5, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \overline{\mathcal{N}_\sigma}} \arrow[ll, bend left=5, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, bend right=10, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in-\mathrm{ri}(N)"']
\end{tikzcd}
$$



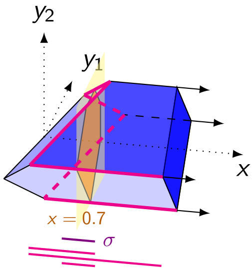
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 61, 15\}$



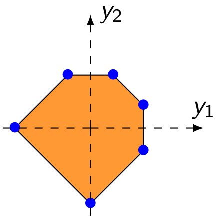


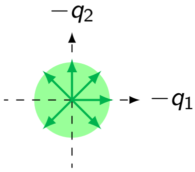
Correspondences

$$
\begin{tikzcd}[column sep=large, row sep=huge]
G \in \mathcal{G}_\sigma \arrow[rr, bend right=10, "G_x"'] \arrow[dr, bend right=10, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right=10, "G \in \mathcal{F}(P) \text{ minimal s.t.} (x,v) \in G"'] \arrow[dl, bend right=10, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right=10, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right=10, "P^I"'] \arrow[ur, bend right=10, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left=5, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right=10, "\arg\min_{y \in P_x} q^\top x \text{ for } q \in -\mathrm{ri}(N)"'] \arrow[ll, bend left=5, "\{i \in \mathrm{supp}\, \mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}
$$



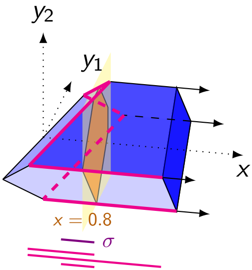
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 61, 15\}$



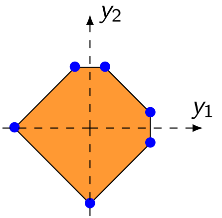


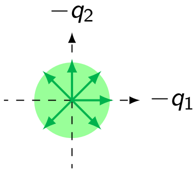
Correspondences

$$\begin{tikzcd}[column sep=large, row sep=huge]
\color{magenta}{G \in \underline{\mathcal{G}_\sigma}} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & \color{blue}{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h-Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& \color{red}{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I} - T_{B_I} x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \color{green}{N \in \overline{\mathcal{N}_\sigma}} \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in \mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}$$



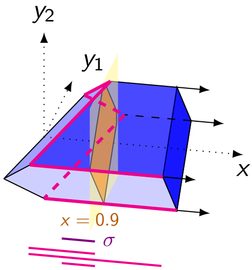
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 61, 15\}$



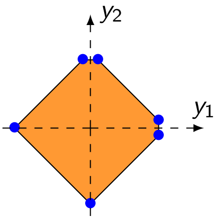


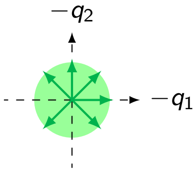
Correspondences

$$\begin{tikzcd}
\color{magenta}{G \in \underline{\mathcal{G}_\sigma}} \arrow[rr, bend right, "G_x"'] \arrow[dr, "I_{(T,W),h}(G)"'] & & \color{blue}{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, "N_{P_x}(v)"] & \\
& \color{red}{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, "P^I"'] \arrow[ur, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \color{green}{N \in \overline{\mathcal{N}_\sigma}} \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"']
\end{tikzcd}$$



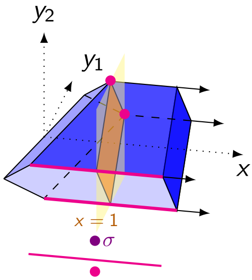
$\overline{\mathcal{I}_\sigma} = \{52, 23, 34, 46, 61, 15\}$



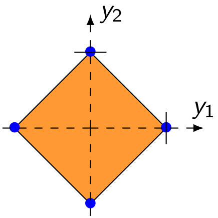


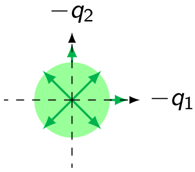
Correspondences

$$\begin{tikzcd}[column sep=large, row sep=huge]
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G \in \mathcal{F}(P) \text{ minimal s.t.} (x,v) \in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right, "\arg\min_{y \in P_x} q^\top x \text{ for } q \in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i \in \mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}$$



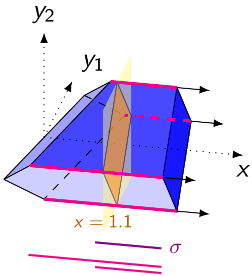
$\overline{\mathcal{I}_\sigma} = \{152, 23, 34, 461\}$



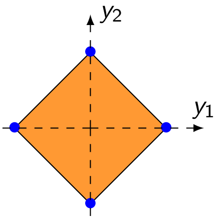


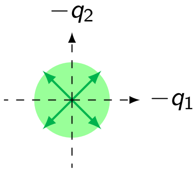
Correspondences

$$
\begin{tikzcd}
G \in \underline{\mathcal{G}_\sigma} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h-Tx}(v)"'] \arrow[dr, bend left, "N_{P_x}(v)"] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}-T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend left, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}
$$



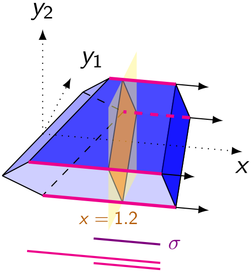
$\overline{\mathcal{I}_\sigma} =$
$\{12, 23, 34, 41\}$



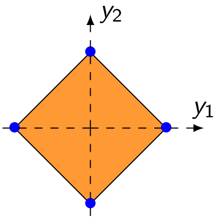


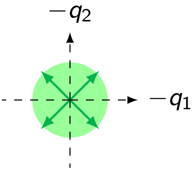
Correspondences

$$
\begin{tikzcd}[column sep=large, row sep=huge]
\boxed{G \in \underline{\mathcal{G}_\sigma}} \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & \boxed{v \in \mathrm{Vert}(P_x)} \arrow[ll, bend right, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& \boxed{I \in \overline{\mathcal{I}_\sigma}} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & \boxed{N \in \overline{\mathcal{N}_\sigma}} \arrow[ll, bend left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"] \arrow[ul, bend right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"']
\end{tikzcd}
$$



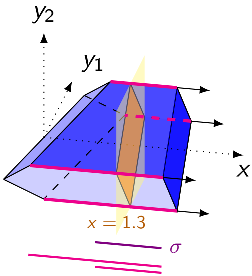
$\overline{\mathcal{I}_\sigma} =$
$\{12, 23, 34, 41\}$



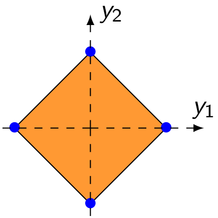


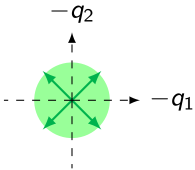
Correspondences

$$
\begin{tikzcd}
G \in \mathcal{G}_\sigma \arrow[rr, bend right, "G_x"'] \arrow[dr, bend right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, bend right, "G \in \mathcal{F}(P) \text{ minimal s.t.} (x,v) \in G"'] \arrow[dl, bend right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, bend right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, bend right, "P^I"'] \arrow[ur, bend right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, bend left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, bend right, "\arg\min_{y \in P_x} q^\top x \text{ for } q \in -\mathrm{ri}(N)"'] \arrow[ll, bend left, "\{i \in \mathrm{supp}\, \mathcal{I}_\sigma \mid W_i^\top \in N\}"]
\end{tikzcd}
$$



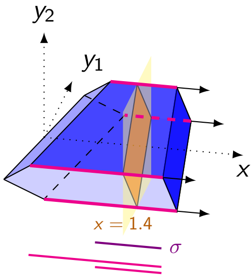
$\overline{\mathcal{I}_\sigma} =$
$\{12, 23, 34, 41\}$



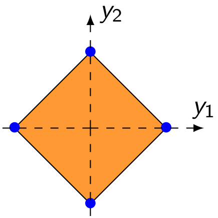


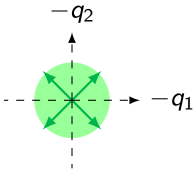
Correspondences

$$\begin{tikzcd}
G \in \mathcal{G}_\sigma \arrow[rr, shift left, "G_x"'] \arrow[dr, shift right, "I_{(T,W),h}(G)"'] & & v \in \mathrm{Vert}(P_x) \arrow[ll, shift left, "G\in\mathcal{F}(P) \text{ minimal s.t.} (x,v)\in G"'] \arrow[dl, shift right, "I_{W,h\text{-}Tx}(v)"'] \arrow[dr, shift right, "N_{P_x}(v)"'] & \\
& I \in \overline{\mathcal{I}_\sigma} \arrow[ul, shift right, "P^I"'] \arrow[ur, shift right, "W_{B_I}^{-1}(h_{B_I}\text{-}T_{B_I}x)"'] \arrow[rr, shift left, "\mathrm{Cone}(W_I^\top)"] & & N \in \overline{\mathcal{N}_\sigma} \arrow[ul, shift right, "\arg\min_{y\in P_x} q^\top x \text{ for } q\in -\mathrm{ri}(N)"'] \arrow[ll, shift left, "\{i\in\mathrm{supp}\,\mathcal{I}_\sigma \mid W_i^\top\in N\}"]
\end{tikzcd}$$



$\overline{\mathcal{I}_\sigma} =$
$\{12, 23, 34, 41\}$





H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices

$$x \in \pi(P^I) \iff \begin{cases} \exists y \in \mathbb{R}^m, \qquad (x, y) \in P^I \\ \end{cases}$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices

$$x \in \pi(P^I) \iff \begin{cases} \exists y \in \mathbb{R}^m, & T_I x + W_I y = h_I \\ \quad \forall j \in [q] \backslash I, & T_j x + W_j y \leqslant h_j \end{cases} \Longleftarrow I \in \mathcal{I}(W, h - Tx)$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices

$$x \in \operatorname{ri} \pi(P^I) \iff \begin{cases} \exists y \in \mathbb{R}^m, & T_I x + W_I y = h_I \\ \quad \forall j \in [q] \backslash I, & T_j x + W_j y < h_j \end{cases} \iff I \in \mathcal{I}(W, h - Tx)$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices from which we can extract a basis (i.e. $\mathrm{rg}(W_I^\top) = m$) and let B such a basis

$$x \in \operatorname{ri} \pi(P^I) \iff \begin{cases} \exists y \in \mathbb{R}^m, & T_B x + W_B y = h_B \\ \forall i \in I \backslash B, & T_i x + W_i y = h_i \\ \forall j \in [q] \backslash I, & T_j x + W_j y < h_j \end{cases} \iff I \in \mathcal{I}(W, h - Tx)$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices from which we can extract a basis (i.e. $\mathrm{rg}(W_I^\top) = m$) and let B such a basis

$$x \in \operatorname{ri} \pi(P^I) \iff \begin{cases} \exists y \in \mathbb{R}^m, & y = W_B^{-1}(h_B - T_B x) \\ \forall i \in I \backslash B, & T_i x + W_i y = h_i \\ \forall j \in [q] \backslash I, & T_j x + W_j y < h_j \end{cases} \iff I \in \mathcal{I}(W, h - Tx)$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices from which we can extract a basis (i.e. $\mathrm{rg}(W_I^\top) = m$) and let B such a basis

$$x \in \mathrm{ri}\, \pi(P^I) \iff \begin{cases} \forall i \in I \backslash B, & T_i x + W_i W_B^{-1}(h_B - T_B x) = h_i \\ \forall j \in [q] \backslash I, & T_j x + W_j W_B^{-1}(h_B - T_B x) < h_j \end{cases}$$

H-representation of projection of faces

Let $I \in \mathcal{I}\big((T, W), h\big)$ be a set of indices from which we can extract a basis (i.e. $\mathrm{rg}(W_I^\top) = m$) and let B such a basis

$$x \in \mathrm{ri}\left(\pi(P^I)\right) \iff \begin{cases} \forall i \in I \backslash B, & (v_i^B)^\top x = u_i^B \\ \forall j \in [q] \backslash I, & (v_j^B)^\top x < u_j^B \end{cases} \iff I \in \mathcal{I}(W, h - Tx)$$

where

$$\begin{aligned} v_i^B &:= (T_i - W_i W_B^{-1} T_B)^\top \\ u_i^B &:= h_i - W_i W_B^{-1} h_B \end{aligned}$$

H-representation of chambers

Let $\sigma \in \mathcal{C}(P, \pi)$

$$x \in \bigcap_{I \in \overline{\mathcal{I}_\sigma}} \mathrm{ri}\left(\pi(P^I)\right) \iff \begin{cases} \forall I \in \overline{\mathcal{I}_\sigma}, \\ \forall i \in I \backslash B_I, \quad (v_i^{B_I})^\top x = u_i^{B_I} \\ \forall j \in [q] \backslash I, \quad (v_j^{B_I})^\top x < u_j^{B_I} \end{cases} \iff \mathcal{I}(W, h - Tx) = \mathcal{I}_\sigma$$

where

$$\begin{aligned} v_i^B &:= (T_i - W_i W_B^{-1} T_B)^\top \\ u_i^B &:= h_i - W_i W_B^{-1} h_B \end{aligned}$$

with B_I basis $\subset I$ and

$$\begin{aligned} \mathcal{G}_\sigma &:= \{F \in \mathcal{F}(P) \mid \mathrm{ri}(\sigma) \subset \mathrm{ri}\left(\pi(F)\right)\} \\ \mathcal{I}_\sigma &:= \{I \in \mathcal{I}\big((T, W), h\big) \mid \mathrm{ri}(\sigma) \subset \mathrm{ri}\left(\pi(P^I)\right)\} \end{aligned}$$

We have $\sigma = \cap_{G \in \mathcal{G}_\sigma} \pi(G) = \cap_{I \in \mathcal{I}_\sigma} \pi(P^I)$

H-representation of chambers

Let $\sigma \in \mathcal{C}(P, \pi)$

$$x \in \mathrm{ri}(\sigma) \iff \begin{cases} \forall I \in \overline{\mathcal{I}_\sigma}, \\ \forall i \in I \backslash B_I, \quad (v_i^{B_I})^\top x = u_i^{B_I} \\ \forall j \in [q] \backslash I, \quad (v_j^{B_I})^\top x < u_j^{B_I} \end{cases} \iff \mathcal{I}(W, h - Tx) = \mathcal{I}_\sigma$$

where

$$\begin{aligned} v_i^B &:= (T_i - W_i W_B^{-1} T_B)^\top \\ u_i^B &:= h_i - W_i W_B^{-1} h_B \end{aligned}$$

with B_I basis $\subset I$ and

$$\begin{aligned} \mathcal{G}_\sigma &:= \{F \in \mathcal{F}(P) \mid \mathrm{ri}(\sigma) \subset \mathrm{ri}\left(\pi(F)\right)\} \\ \mathcal{I}_\sigma &:= \{I \in \mathcal{I}\big((T, W), h\big) \mid \mathrm{ri}(\sigma) \subset \mathrm{ri}\left(\pi(P^I)\right)\} \end{aligned}$$

We have $\sigma = \cap_{G \in \underline{\mathcal{G}_\sigma}} \pi(G) = \cap_{I \in \overline{\mathcal{I}_\sigma}} \pi(P^I)$