# Multistage stochastic optimization and polyhedral geometry

PhD Defense    Maël Forcier

advised by Stéphane Gaubert and Vincent Leclère,

supervised by Jean-Philippe Chancelier.

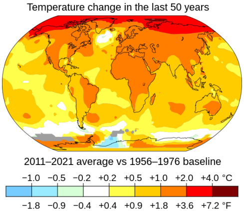
December 14th 2022

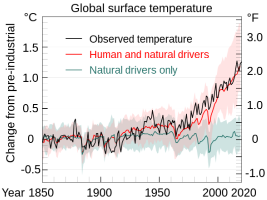
ÉCOLE POLYTECHNIQUE
IP PARIS

École des Ponts
ParisTech

# Motivating example: hydroelectric energy management

- Need low-carbon energy to stop global warming
- Hydroelectricity is a controllable renewable energy
- 83% of electricity is hydroelectric in Brazil, 17% 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

$$\min_{u_t, x_t} \sum_{t=1}^{T} c_t(d_t - u_t)$$

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

# 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

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

# 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.

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

$x_1 + x_2 \leqslant 1$ (1)

(2)

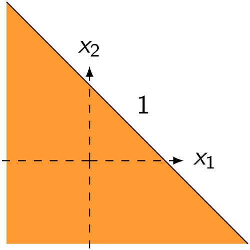
(3)

(4)

(5)

(6)

(7)



# Linear Programming and polyhedra

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

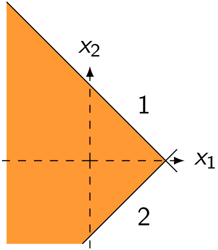
**Definition**

*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

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

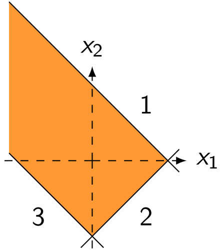
**Definition**

*Polyhedron:*
*Intersection of finite number of halfspaces*

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ 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

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

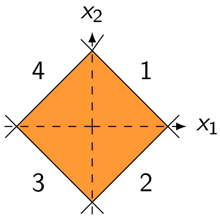
**Definition**

*Polyhedron:*
*Intersection of finite number of halfspaces*

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ 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

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

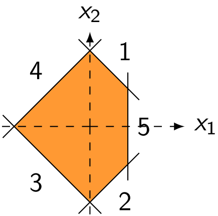
**Definition**

*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.

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

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



# Linear Programming and polyhedra

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

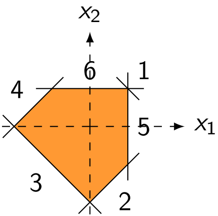
**Definition**

*Polyhedron:*
*Intersection of finite number of halfspaces*

The set $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ 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

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

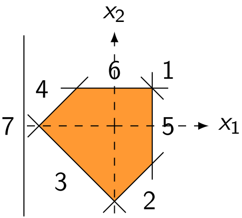
**Definition**

*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.

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

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



# 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

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

# 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

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

# 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.

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

Equivalent form

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

# 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.

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

Equivalent form

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

# 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

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

# 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.

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

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.

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

Equivalent form

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

# 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

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

# 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

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

# 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

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

# Dynamic Programming (Bellman 1966)

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

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

# Dynamic Programming (Bellman 1966)

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

We set $V_{T+1} \equiv 0$ 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)

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

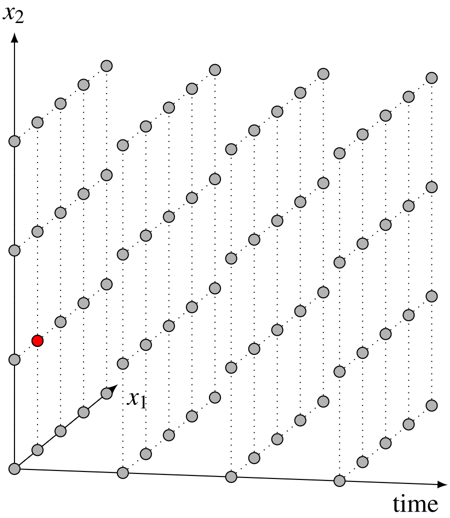
We set $V_{T+1} \equiv 0$ and $V_t(x_{t-1}) := \mathbb{E}\left[ \begin{array}{ll} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{array} \right]$

# Dynamic Programming (Bellman 1966)

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

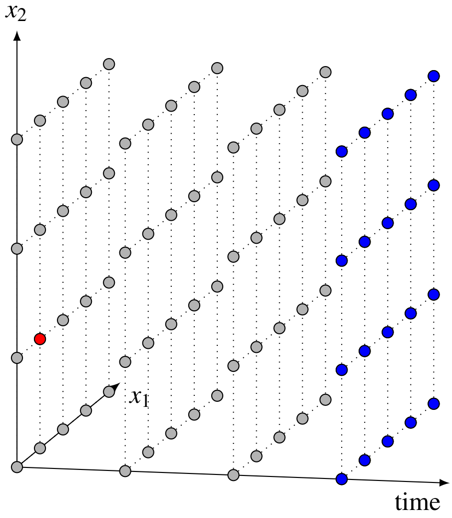
We set $V_{T+1} \equiv 0$ and $V_t(x_{t-1}) := \mathbb{E}\left[ \begin{array}{ll} \min\limits_{x_t \in \mathbb{R}^{n_t}} & \boldsymbol{c}_t^\top x_t + V_{t+1}(x_t) \\ \text{s.t.} & \boldsymbol{A}_t x_t + \boldsymbol{B}_t x_{t-1} \leqslant \boldsymbol{b}_t \end{array} \right]$

# Dynamic programming: finite case

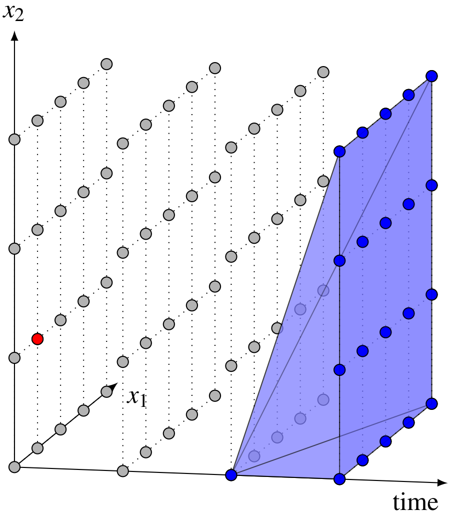



Thank you Vincent for this animation.

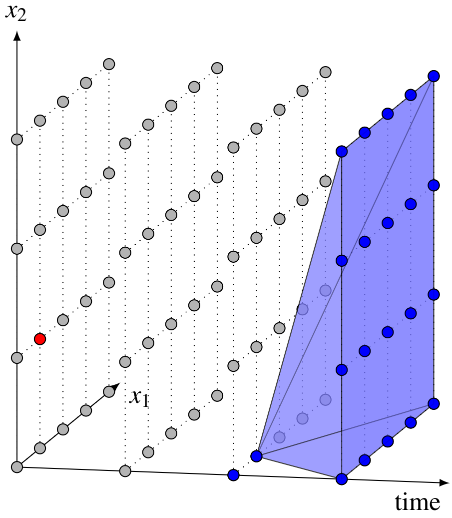
# Dynamic programming: finite case



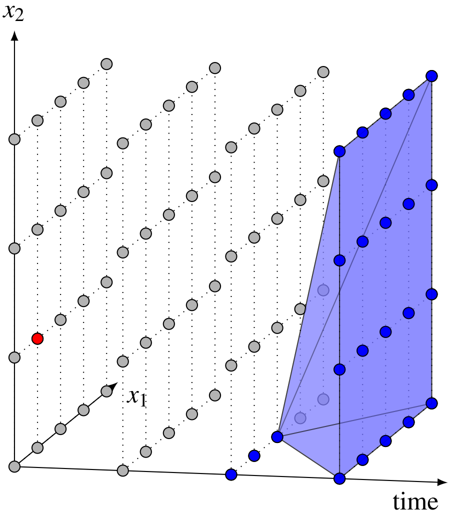
# Dynamic programming: finite case



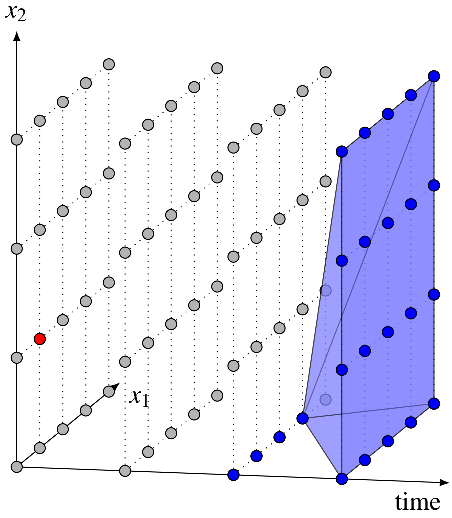
# Dynamic programming: finite case



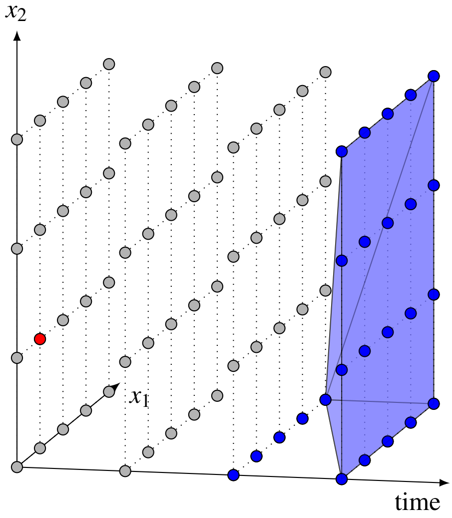
# Dynamic programming: finite case



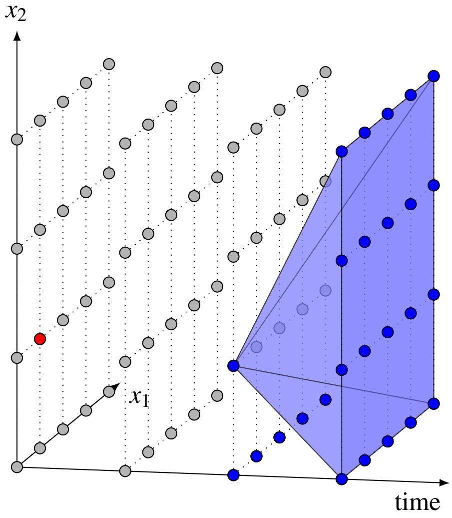
# Dynamic programming: finite case



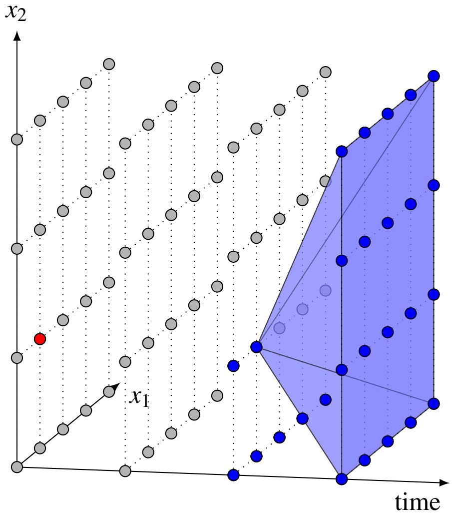
# Dynamic programming: finite case



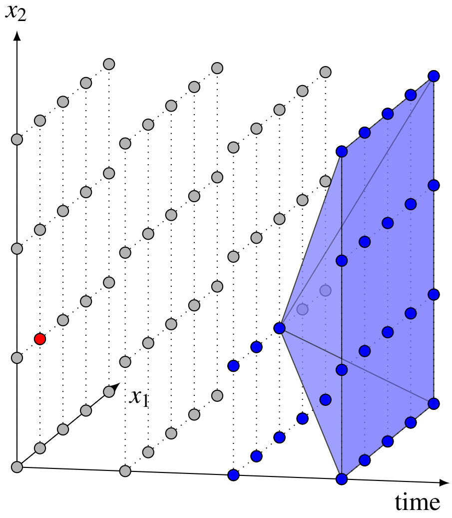
# Dynamic programming: finite case



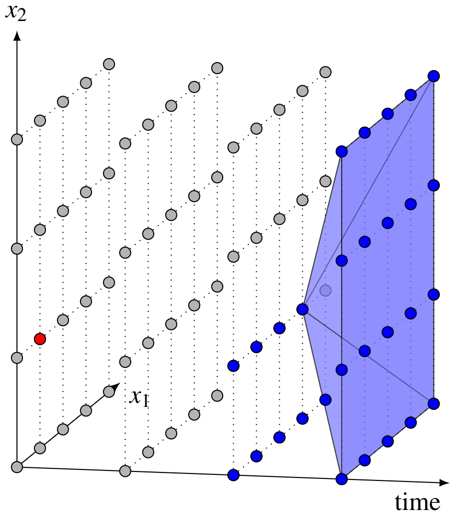
# Dynamic programming: finite case



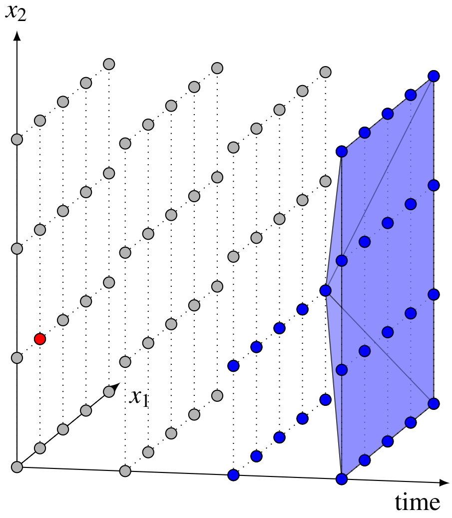
# Dynamic programming: finite case



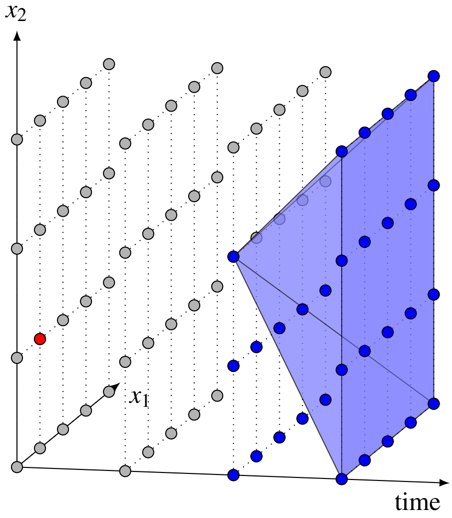
# Dynamic programming: finite case



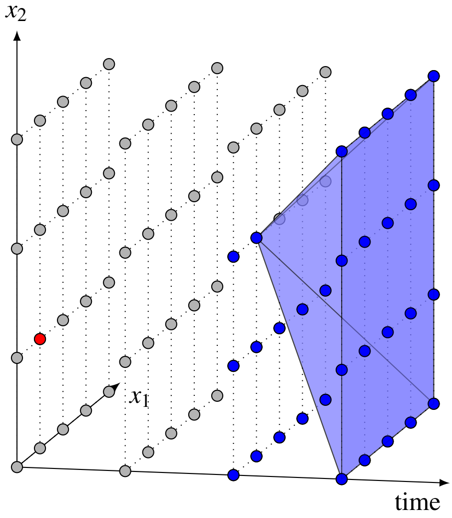
# Dynamic programming: finite case



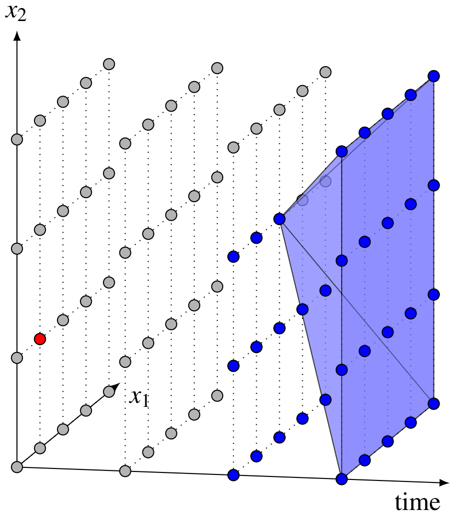
# Dynamic programming: finite case



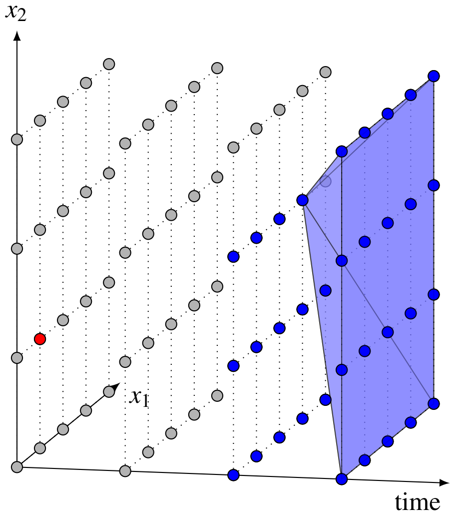
# Dynamic programming: finite case



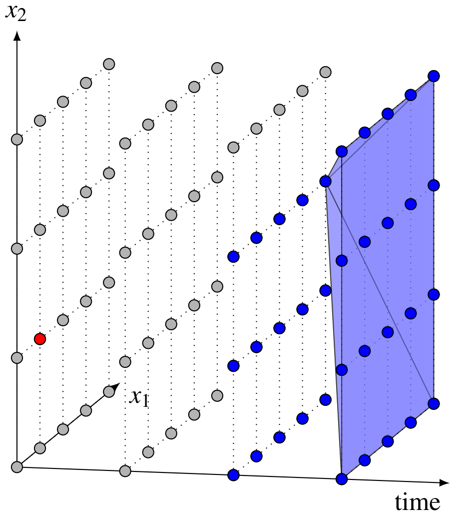
# Dynamic programming: finite case



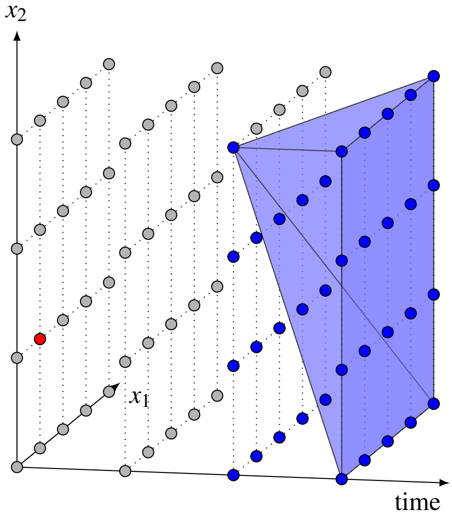
# Dynamic programming: finite case



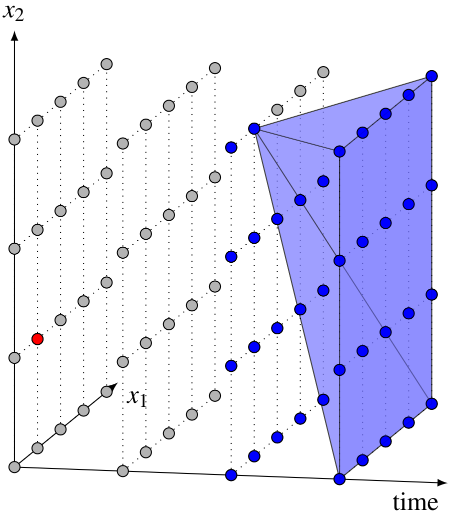
# Dynamic programming: finite case



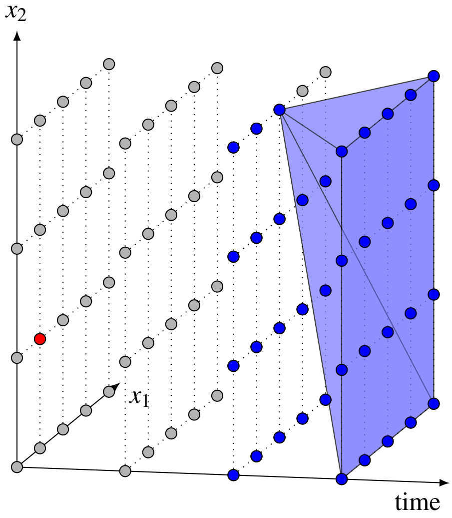
# Dynamic programming: finite case



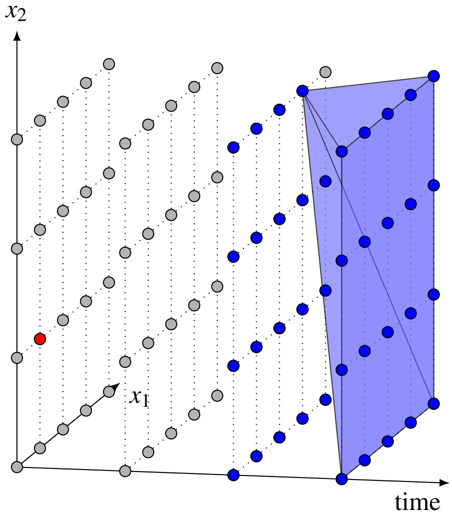
# Dynamic programming: finite case



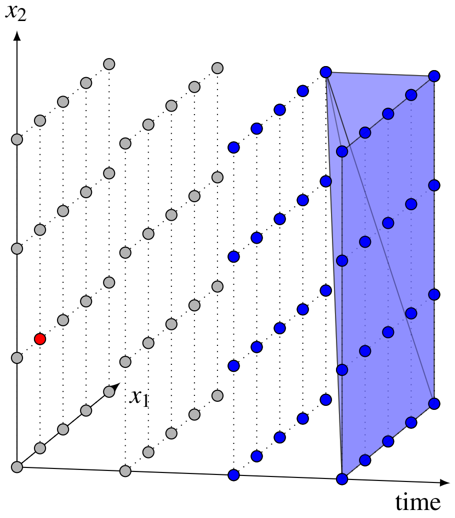
# Dynamic programming: finite case



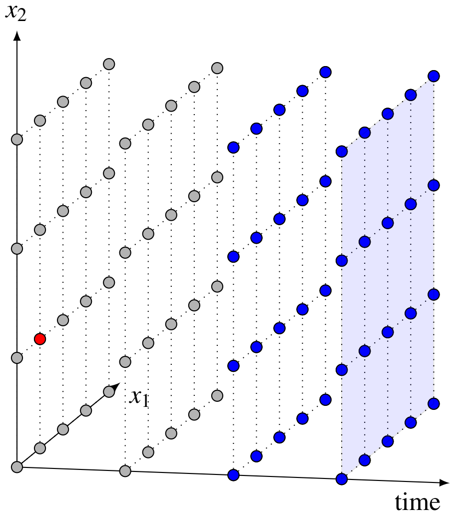
# Dynamic programming: finite case



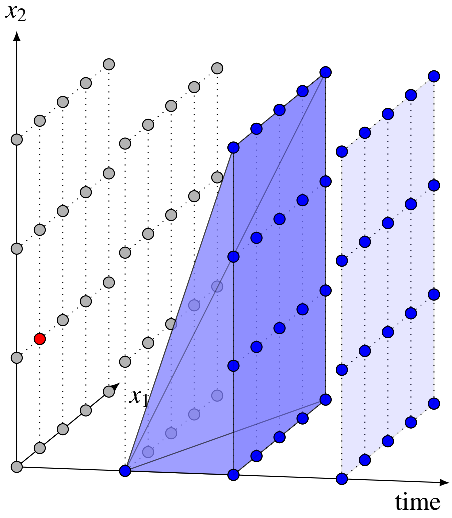
# Dynamic programming: finite case



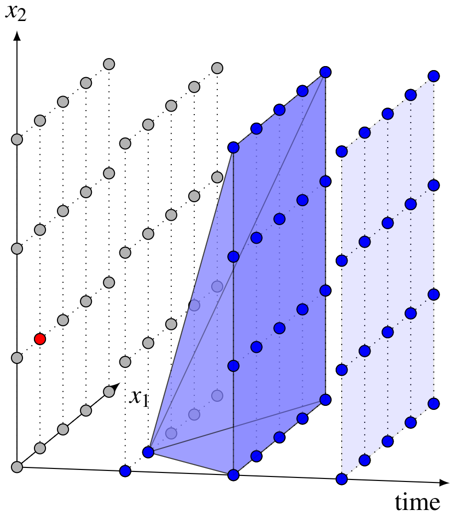
# Dynamic programming: finite case



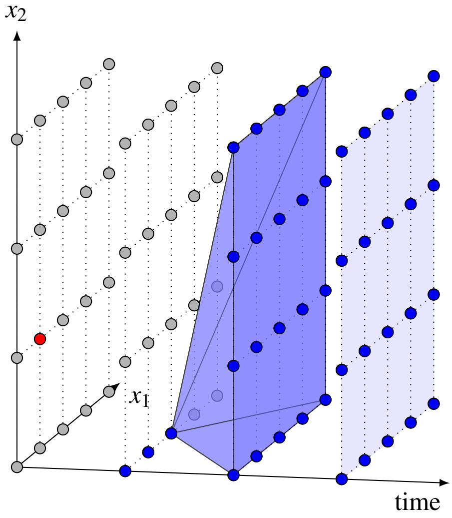
# Dynamic programming: finite case



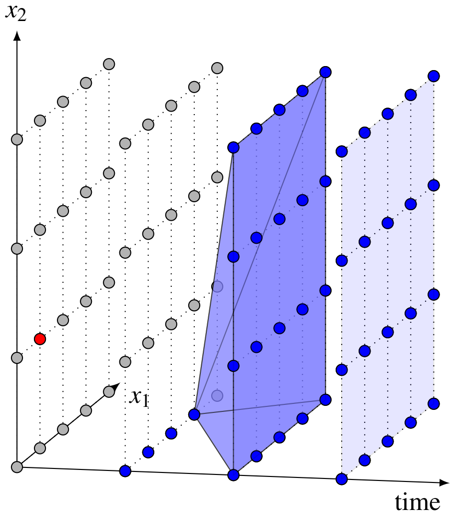
# Dynamic programming: finite case



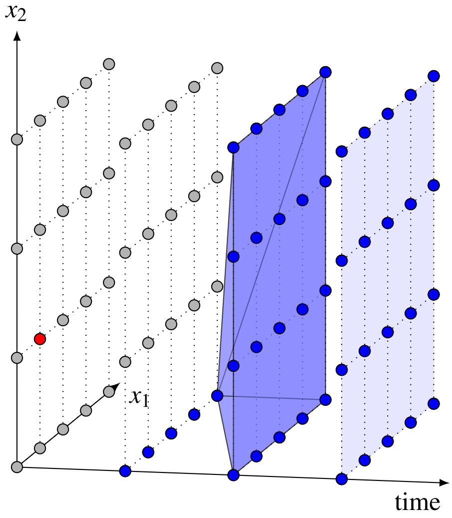
# Dynamic programming: finite case



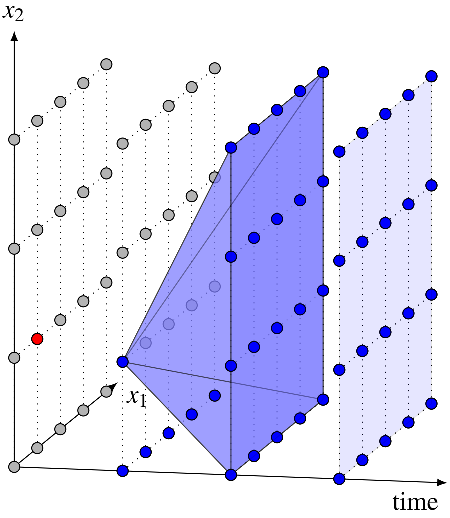
# Dynamic programming: finite case



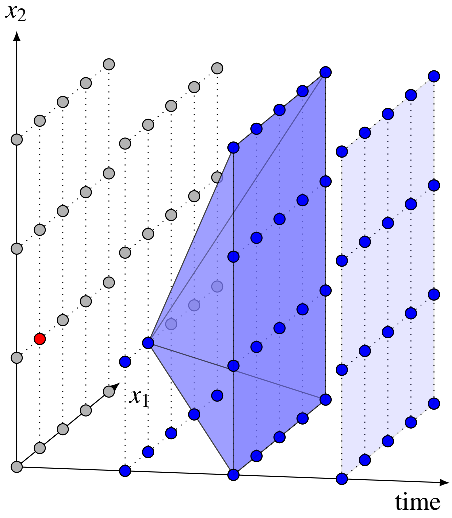
# Dynamic programming: finite case



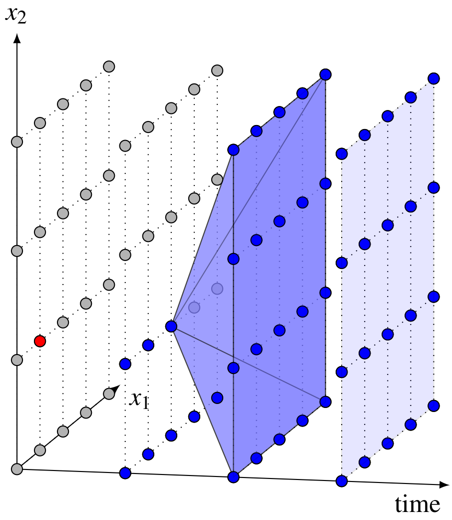
# Dynamic programming: finite case



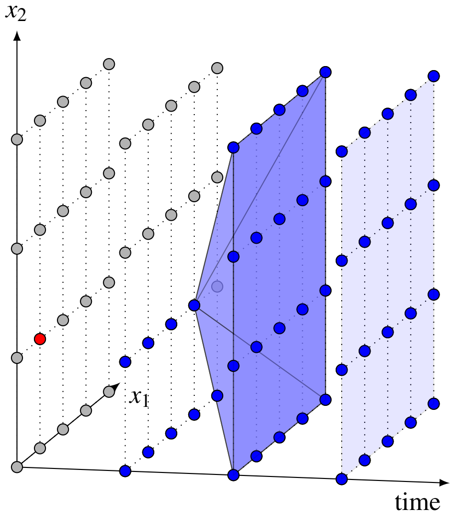
# Dynamic programming: finite case



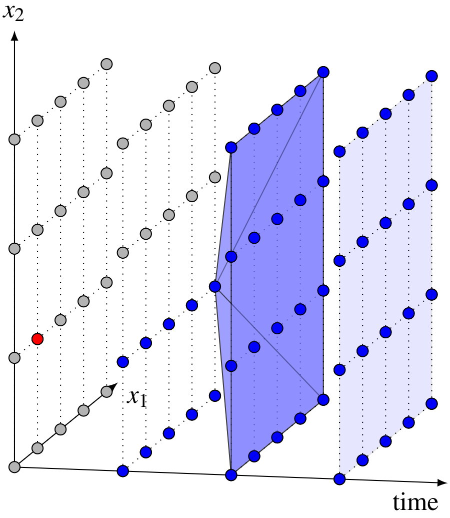
# Dynamic programming: finite case



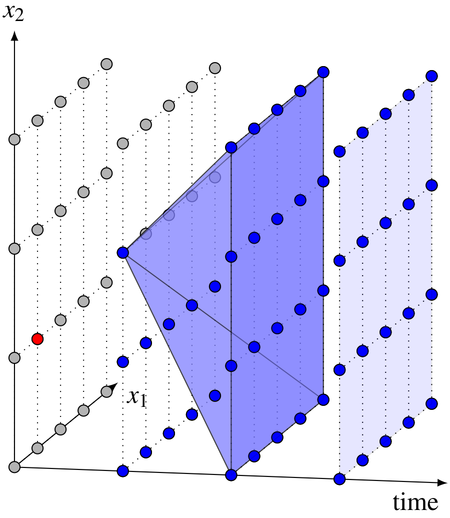
# Dynamic programming: finite case



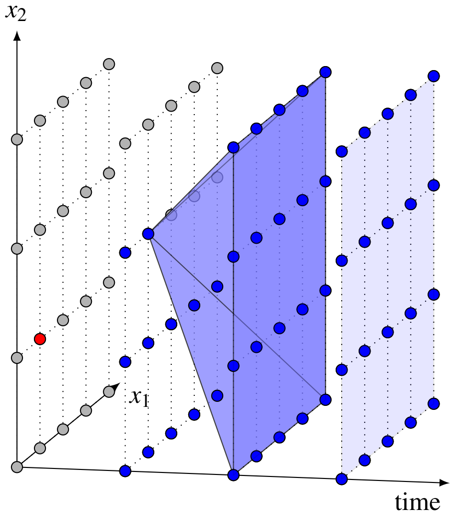
# Dynamic programming: finite case



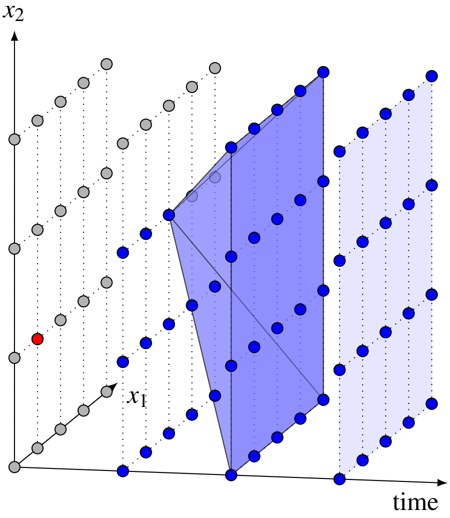
# Dynamic programming: finite case



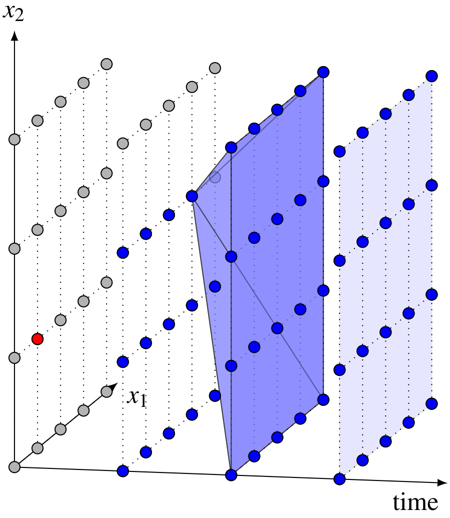
# Dynamic programming: finite case



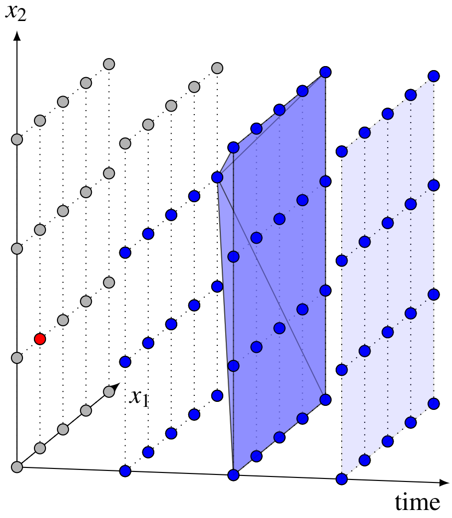
# Dynamic programming: finite case



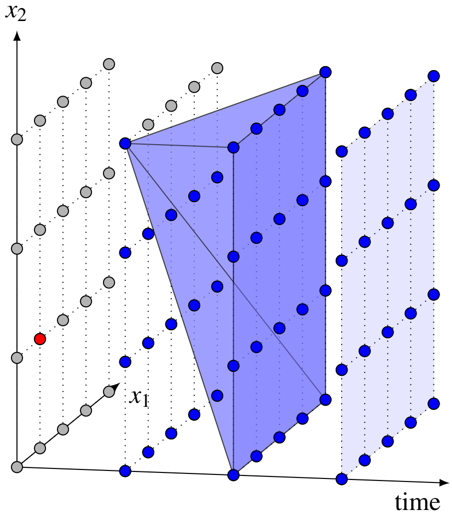
# Dynamic programming: finite case



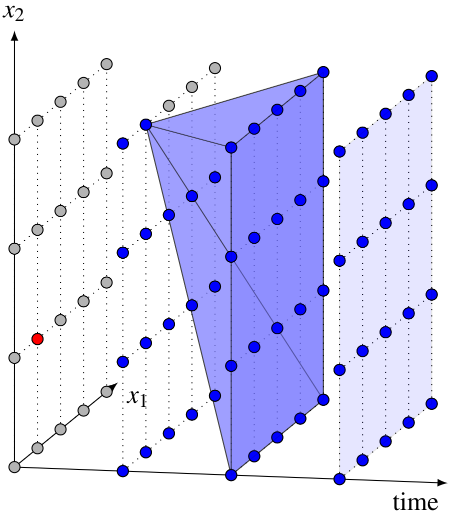
# Dynamic programming: finite case



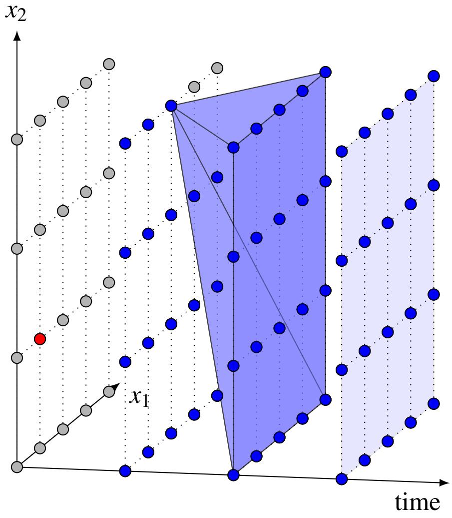
# Dynamic programming: finite case



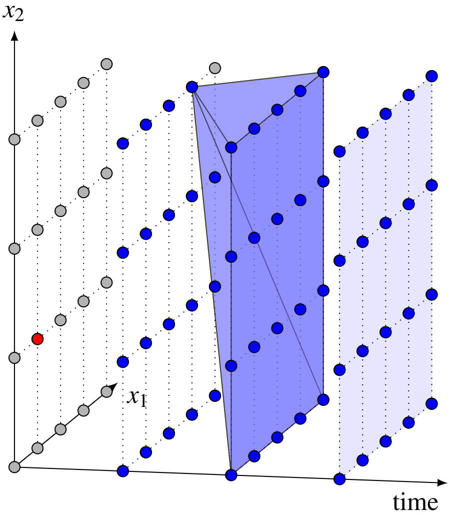
# Dynamic programming: finite case



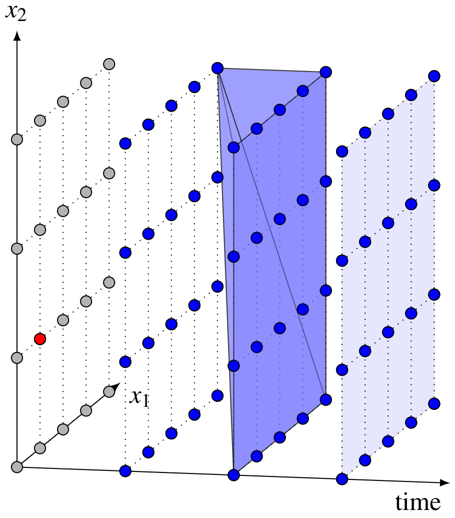
# Dynamic programming: finite case



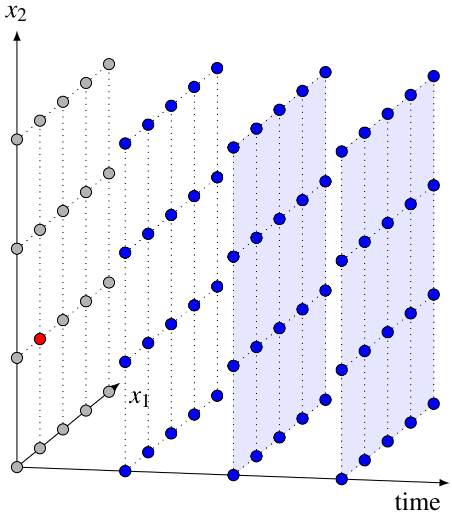
# Dynamic programming: finite case



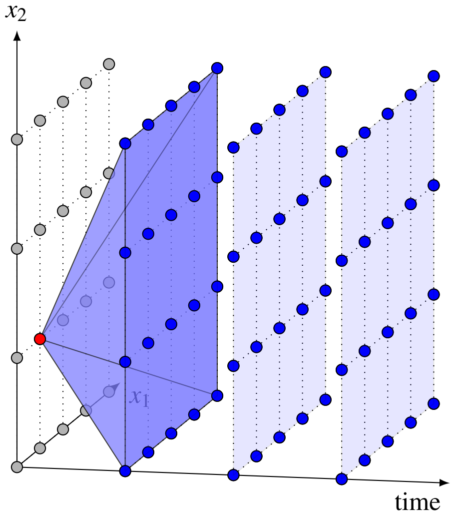
# Dynamic programming: finite case



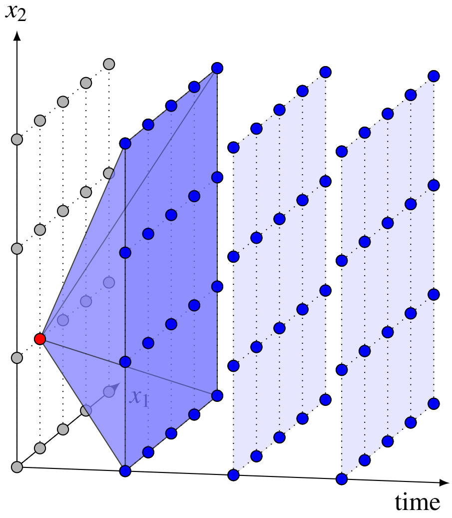
# Dynamic programming: finite case



# Dynamic programming: finite case

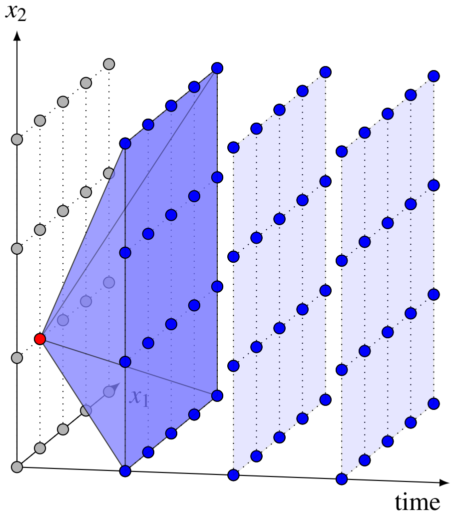


# Dynamic programming: finite case




➥ Continuous space: algorithms such as SDDP (discussed later).

# Dynamic programming: finite case

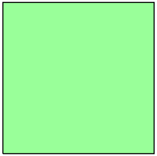



- ➡ Continuous space: algorithms such as SDDP (discussed later).
- ➡ How to deal with continuous distributions ?

# Quantization of a MSLP

**Real problem**

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

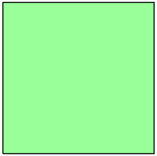


$\xi_t$ continuous

# Quantization of a MSLP

**Real problem**

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

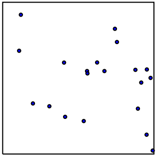


$\boldsymbol{\xi}_t$ continuous

**Sample Average Approximation (SAA)**

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

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

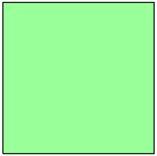


SAA $N = 20$

# Quantization of a MSLP

**Real problem**

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

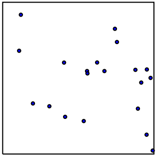


$\xi_t$ continuous

**Sample Average Approximation (SAA)**

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

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

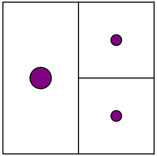


SAA $N = 20$

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

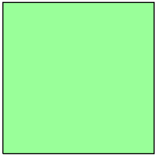


Partition-based

# Quantization of a MSLP

**Real problem**

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

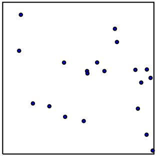


$\boldsymbol{\xi}_t$ continuous

**Sample Average Approximation (SAA)**

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

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



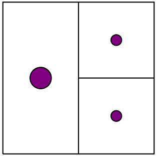
SAA $N = 20$

**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 $\check{\xi}_{t,P} := \mathbb{E}\big[\boldsymbol{\xi}_t \mid \boldsymbol{\xi}_t \in P\big]$

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



Partition-based

# Exact quantization

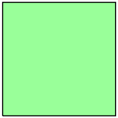
## Definition

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

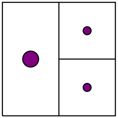
$$V_t(x) = \mathbb{E}\left[\hat{V}_t(x, \xi_t)\right] = \mathbb{E}\left[\hat{V}_t(x, \check{\xi}_t)\right].$$

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 $(\xi_\tau)_{\tau\in[T]}$.



$\xi_t$ continuous



$\check{\xi}_t$ quantized

# 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 $\boldsymbol{u} \sim \mathcal{U}\big([1,2]\big)$.

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

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.
- ➥ From 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 $\mathcal{P}$

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

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.
- From 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.

➥ From 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 $\mathcal{P}$

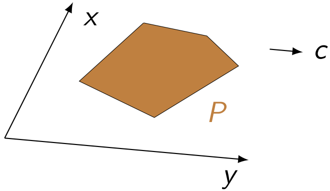
$$\sum_{P \in \mathcal{P}} \check{p}_P \hat{V}\big(x, \check{\xi}_P\big) < V(x) = \mathbb{E}\left[\frac{1}{\boldsymbol{u}}\right]$$

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.
- ➥ From now on, $A$ is deterministic: fixed recourse.

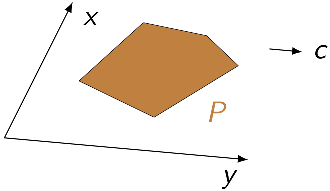
# Uniform exact quantization and polyhedrality

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



# Uniform exact quantization and polyhedrality

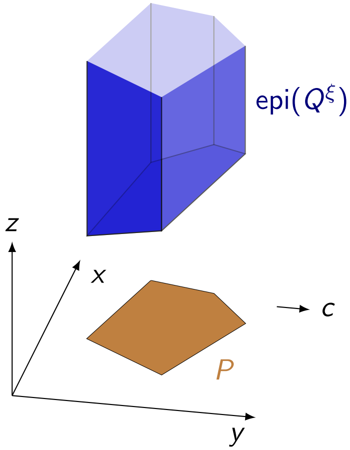
$$\hat{V}(x,\xi) = \min_{y \in \mathbb{R}^m} \; c^\top y \quad \text{s.t.} \quad (x,y) \in P$$



# Uniform exact quantization and polyhedrality

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

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

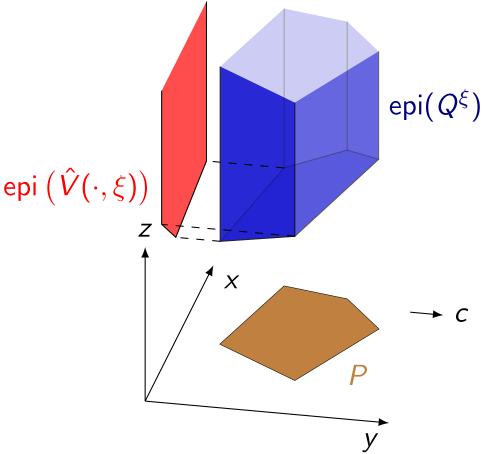


# Uniform exact quantization and polyhedrality

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

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

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

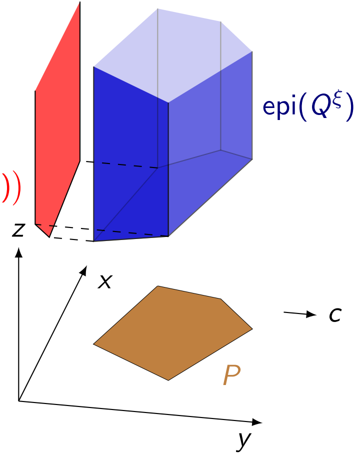


# Uniform exact quantization and polyhedrality

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

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

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




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

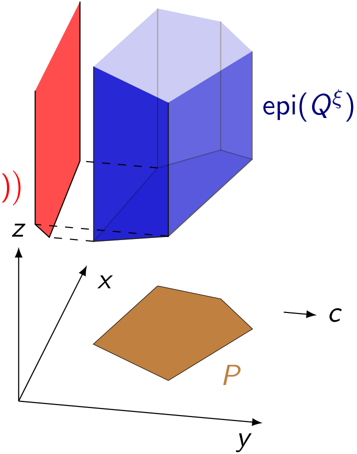
➥ If the noise is finitely supported, then $V$ is polyhedral

# 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$(Q^\xi)$.




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

- ➥ 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 | × | ? | ? |
| Uniform | × | **?** | ? |

Stochastic $\boldsymbol{B}$

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

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

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

---

$\boldsymbol{u}$ 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

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

Stochastic $\boldsymbol{B}$

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

Stochastic $\boldsymbol{b}$

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

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

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

# Counter examples with stochastic constraints

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

Stochastic $\boldsymbol{B}$

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

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

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

Stochastic $\boldsymbol{b}$

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

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

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

➥ $V$ 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

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

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

**Theorem (FGL 2021)**

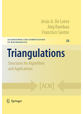
*If $A$, $B$ and $b$ are deterministic,*
*then there exists a universal and uniform exact quantization.*

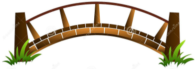
**Theorem (FL 2022)**

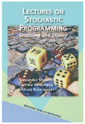
*If $A$ is deterministic,*
*then there exists a universal and local exact quantization.*

# 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

# Contents

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

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

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

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

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 $V_{t+1} \equiv 0$)

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

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

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 $P_x$

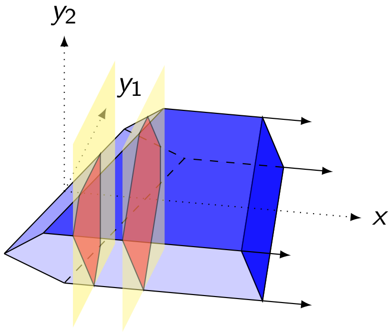
For a given $x$, (we still assume $V_{t+1} \equiv 0$)

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

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

Illustrative running example:

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



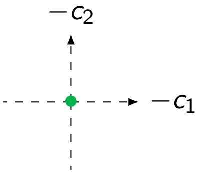
# Normal fan $\mathcal{N}(P_x)$

## Definition

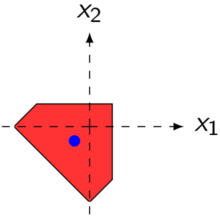
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

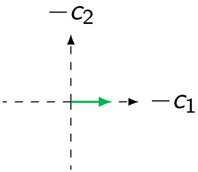
# Normal fan $\mathcal{N}(P_x)$

## Definition

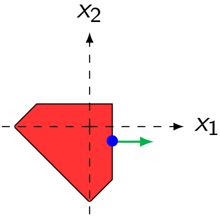
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

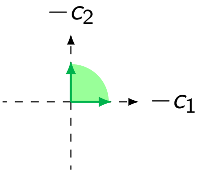
# Normal fan $\mathcal{N}(P_x)$

**Definition**

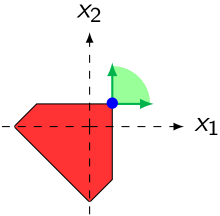
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

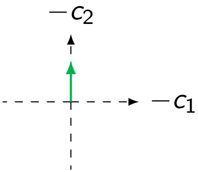
# Normal fan $\mathcal{N}(P_x)$

**Definition**

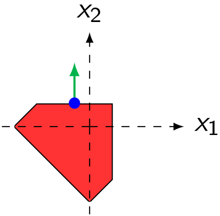
*The normal fan of the fiber* $P_x$ *is*

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

*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 $x = 0.3$

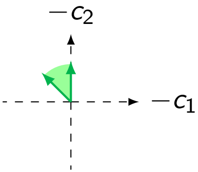
# Normal fan $\mathcal{N}(P_x)$

## Definition

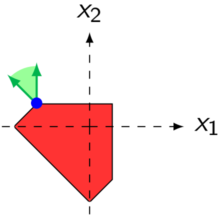
*The normal fan of the fiber* $P_x$ *is*

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

*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 $x = 0.3$

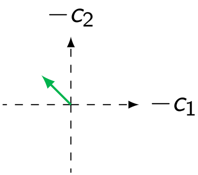
# Normal fan $\mathcal{N}(P_x)$

**Definition**

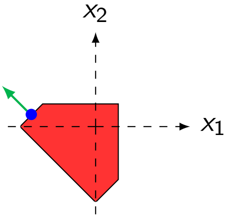
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

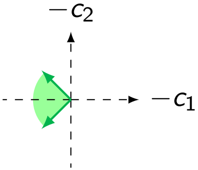
# Normal fan $\mathcal{N}(P_x)$

**Definition**

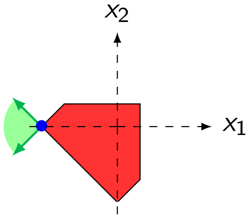
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

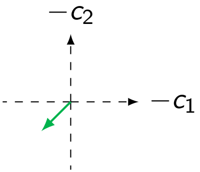
# Normal fan $\mathcal{N}(P_x)$

**Definition**

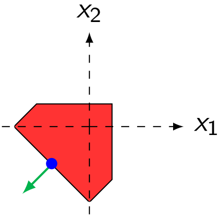
*The normal fan of the fiber* $P_x$ *is*

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

*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 $x = 0.3$

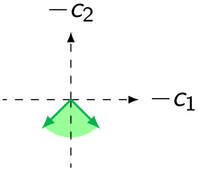
# Normal fan $\mathcal{N}(P_x)$

**Definition**

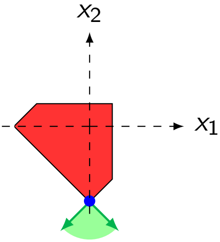
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

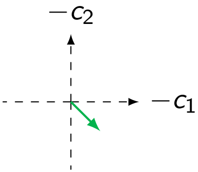
# Normal fan $\mathcal{N}(P_x)$

**Definition**

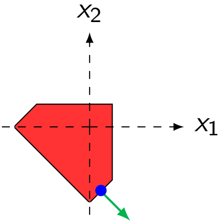
*The normal fan of the fiber $P_x$ is*

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

*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 $x = 0.3$

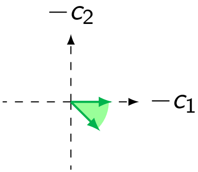
# Normal fan $\mathcal{N}(P_x)$

**Definition**

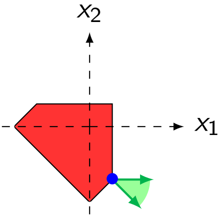
*The normal fan of the fiber $P_x$ is*

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

*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

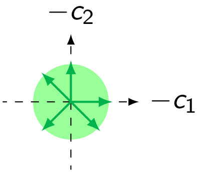
*The normal fan of the fiber $P_x$ is*

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

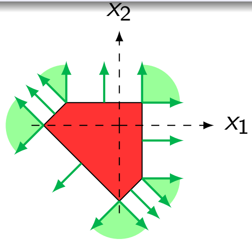
*with* $N_{P_x}(y) = \{c \mid \forall y' \in P_x,\ c^\top (y' - y) \leqslant 0\}$ *the normal cone of $P_x$ at $y$.*

## 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 $x = 0.3$

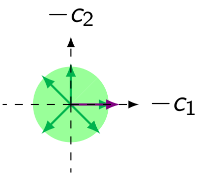



$P_x$ and $\mathcal{N}(P_x)$ for $x = 0.3$

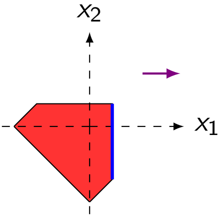
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

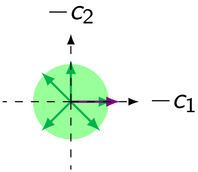



$P_x$ for $x = 0.3$

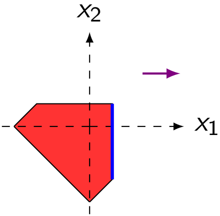
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

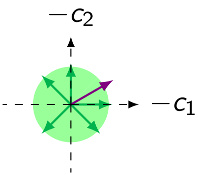



$P_x$ for $x = 0.3$

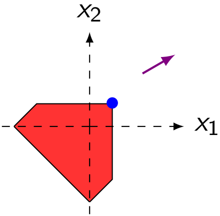
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

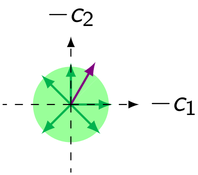



$P_x$ for $x = 0.3$

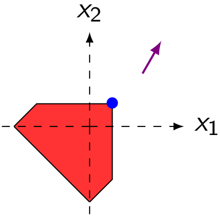
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

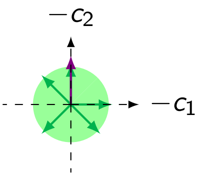



$P_x$ for $x = 0.3$

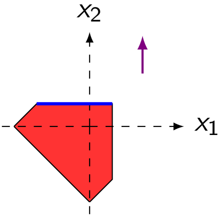
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

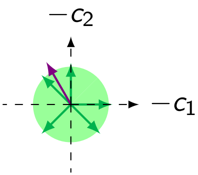



$P_x$ for $x = 0.3$

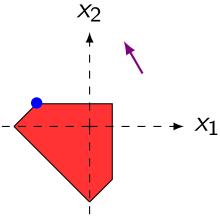
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

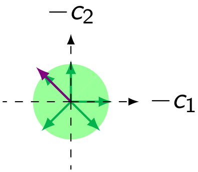



$P_x$ for $x = 0.3$

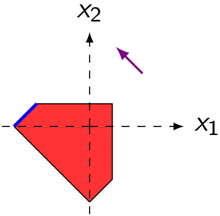
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

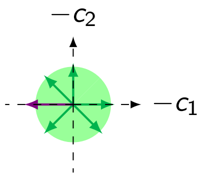



$P_x$ for $x = 0.3$

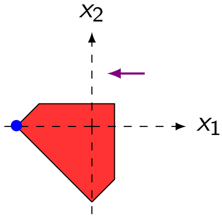
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

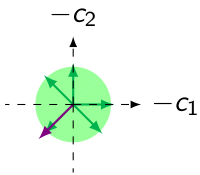



$P_x$ for $x = 0.3$

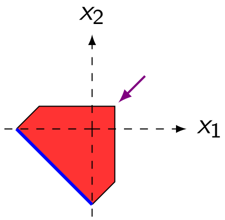
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

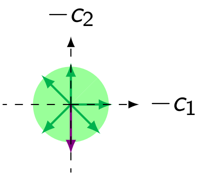



$P_x$ for $x = 0.3$

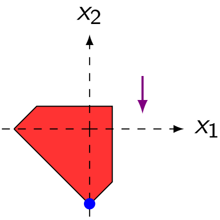
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

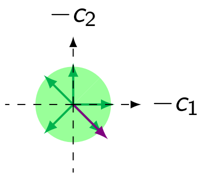



$P_x$ for $x = 0.3$

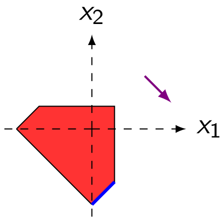
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

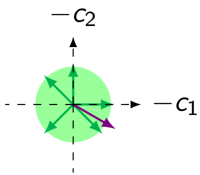



$P_x$ for $x = 0.3$

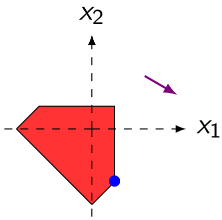
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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




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

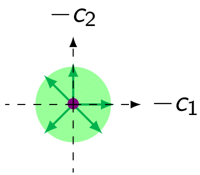



$P_x$ for $x = 0.3$

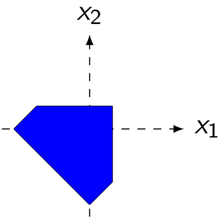
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

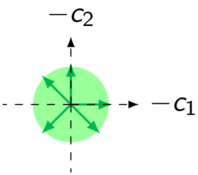



$P_x$ for $x = 0.3$

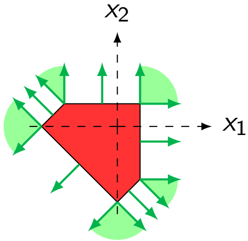
# $\mathcal{N}(P_x)$: partition of cost coherent with the min

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

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 $x = 0.3$

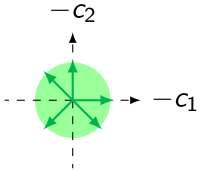



$P_x$ for $x = 0.3$

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

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

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



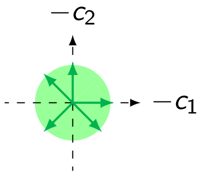

$\mathcal{N}(P_x)$ for $x = 0.3$

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

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

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

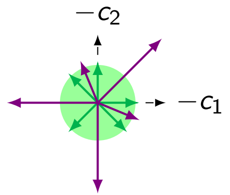
$$= \sum_{N \in \mathcal{N}(P_x)} \mathbb{E}\big[\mathbb{1}_{\boldsymbol{c} \in -\operatorname{ri} N} \boldsymbol{c}^\top\big] y_N(x)$$




$\mathcal{N}(P_x)$ 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 $\boldsymbol{c}$

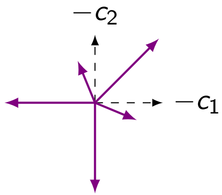
$$V(x) = \mathbb{E}\Big[\min_{y \in \color{red}{P_x}} \boldsymbol{c}^\top y\Big]$$

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

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

$$= \sum_{\color{green}{N \in \mathcal{N}(P_x)}} \color{purple}{p_N \check{c}_N}^\top \color{blue}{y_N(x)}$$

$$= \sum_{\color{green}{N \in \mathcal{N}(P_x)}} \color{purple}{p_N} \min_{y \in \color{red}{P_x}} \color{purple}{\check{c}_N}^\top y$$




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

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

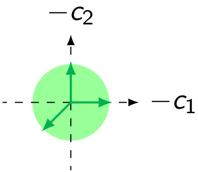
$$\color{purple}{p_N} := \mathbb{P}\big[\boldsymbol{c} \in -\operatorname{ri} N\big]$$
$$\color{purple}{\check{c}_N} := \mathbb{E}\big[\boldsymbol{c} \mid \boldsymbol{c} \in -\operatorname{ri} N\big]$$

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

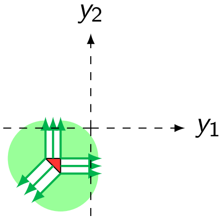
# Contents

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

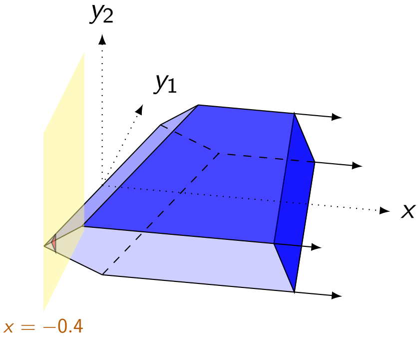
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



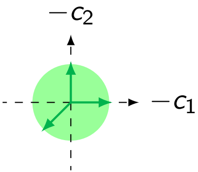

$P_x$ and $\mathcal{N}(P_x)$



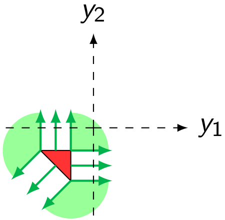

$P$ and $P_x$

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

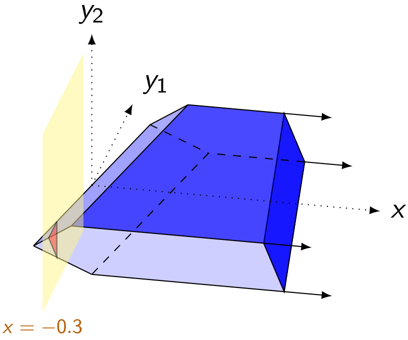
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



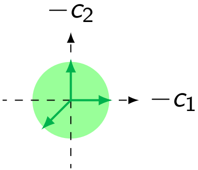

$P_x$ and $\mathcal{N}(P_x)$



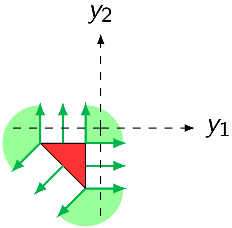

$P$ and $P_x$

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

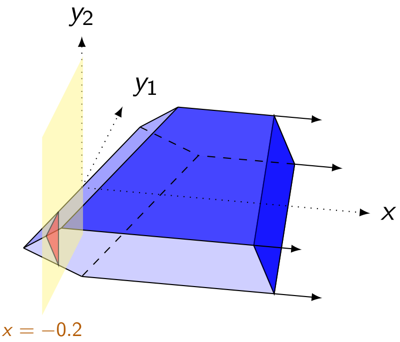
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



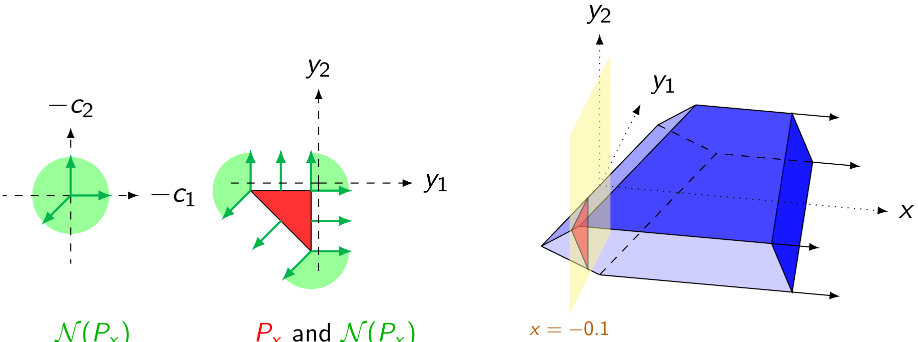

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

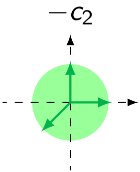
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$



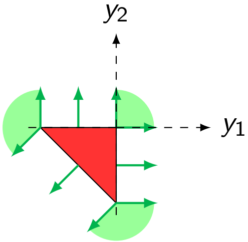

$\mathcal{N}(P_x)$ $P_x$ and $\mathcal{N}(P_x)$ $P$ and $P_x$

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

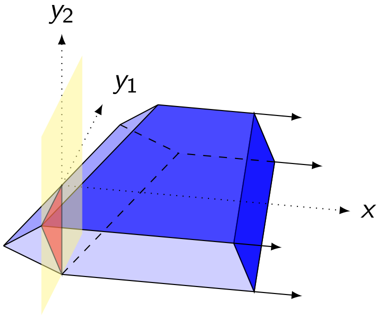
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



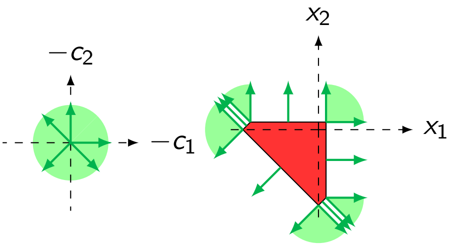

$P_x$ and $\mathcal{N}(P_x)$



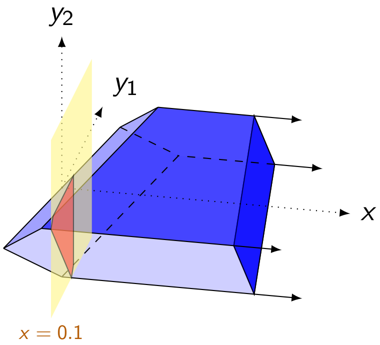

$P$ and $P_x$

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

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



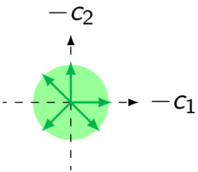

$\mathcal{N}(P_x)$ $\qquad$ $P_x$ and $\mathcal{N}(P_x)$



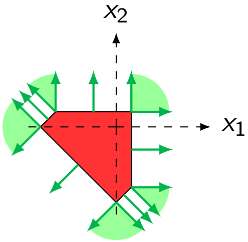

$P$ and $P_x$

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

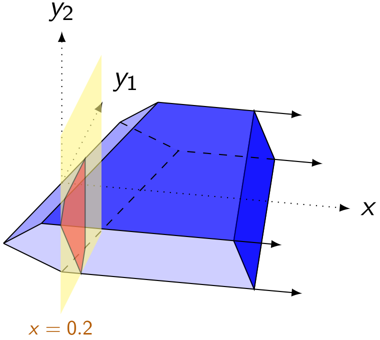
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



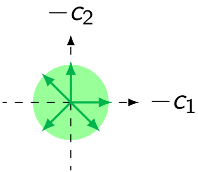

$P_x$ and $\mathcal{N}(P_x)$



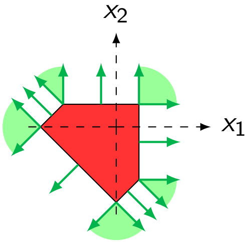

$P$ and $P_x$

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

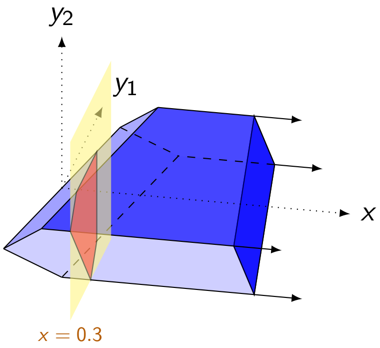
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



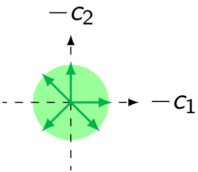

$P_x$ and $\mathcal{N}(P_x)$



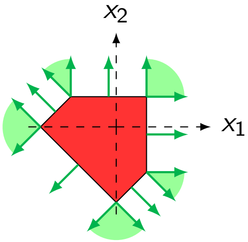

$P$ and $P_x$

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

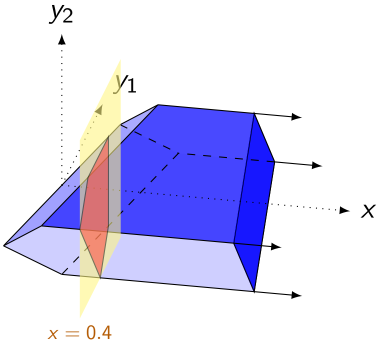
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



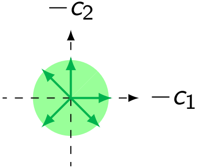

$P_x$ and $\mathcal{N}(P_x)$



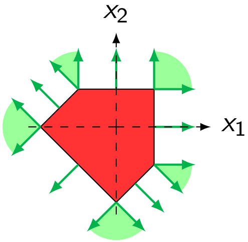

$P$ and $P_x$

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

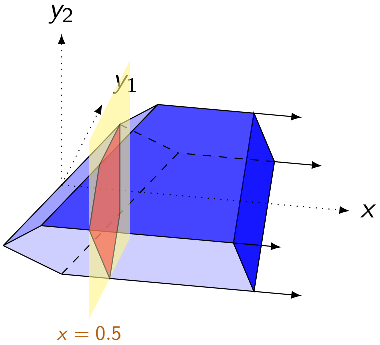
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



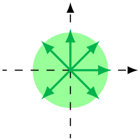

$P_x$ and $\mathcal{N}(P_x)$



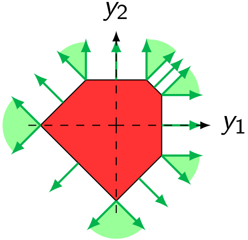

$P$ and $P_x$

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

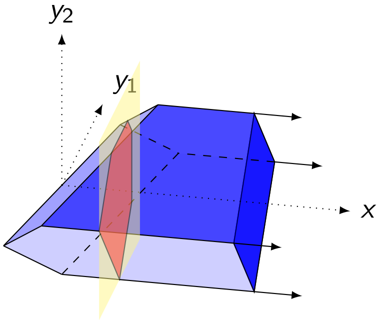
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



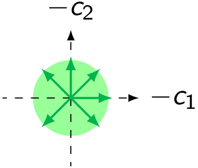

$P_x$ and $\mathcal{N}(P_x)$



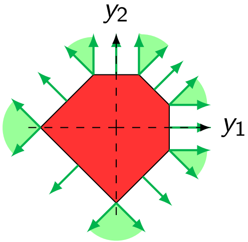

$P$ and $P_x$

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

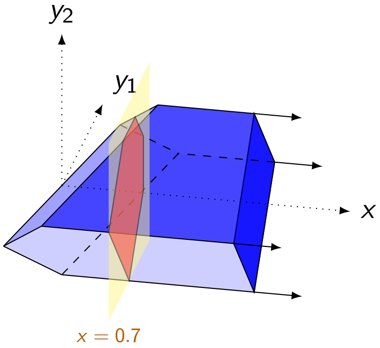
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



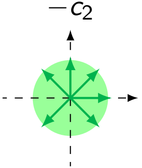

$P_x$ and $\mathcal{N}(P_x)$



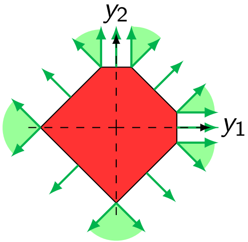

$P$ and $P_x$

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

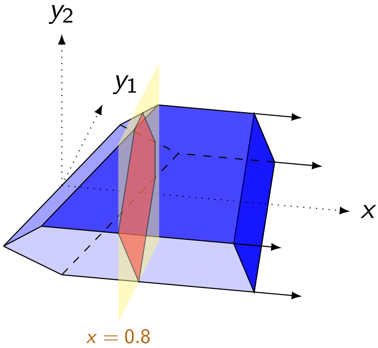
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



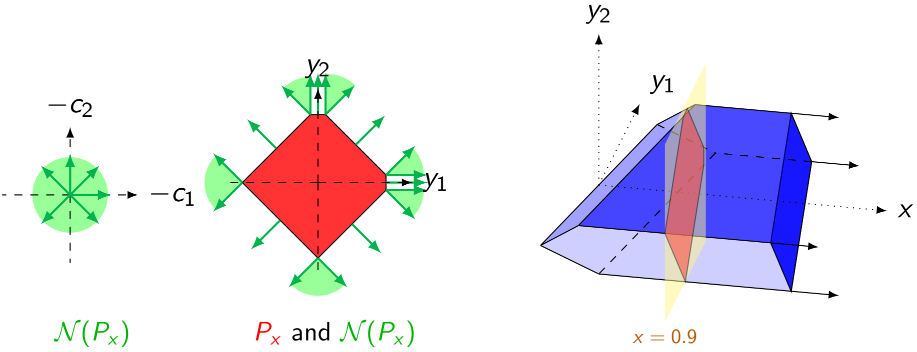

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

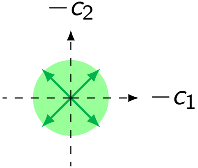
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



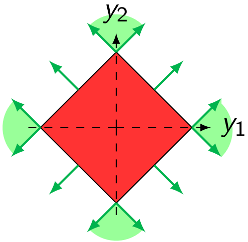

$\mathcal{N}(P_x)$ $\quad$ $P_x$ and $\mathcal{N}(P_x)$ $\quad$ $P$ and $P_x$

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

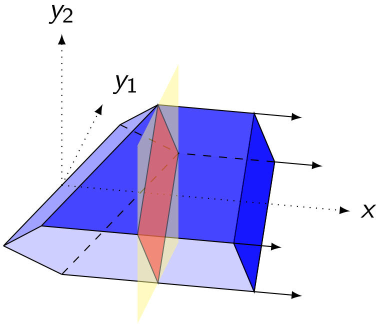
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



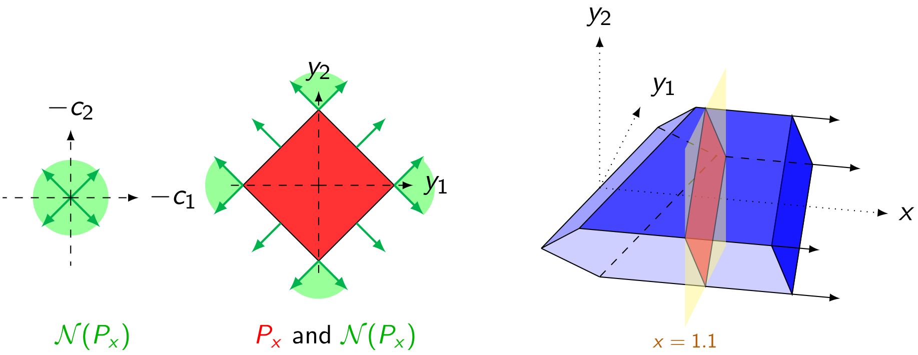

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

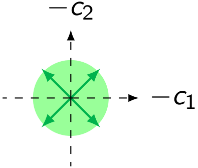
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



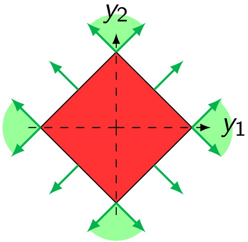

$\mathcal{N}(P_x)$ $\qquad$ $P_x$ and $\mathcal{N}(P_x)$ $\qquad$ $P$ and $P_x$

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

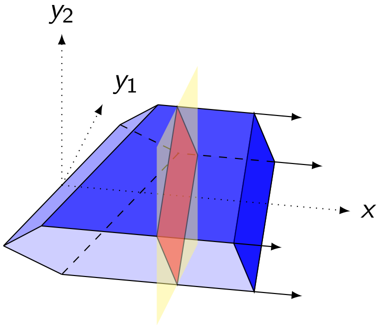
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



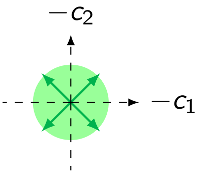

$P_x$ and $\mathcal{N}(P_x)$



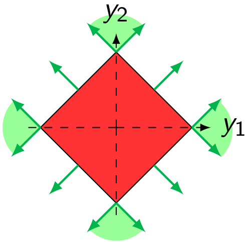

$P$ and $P_x$

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

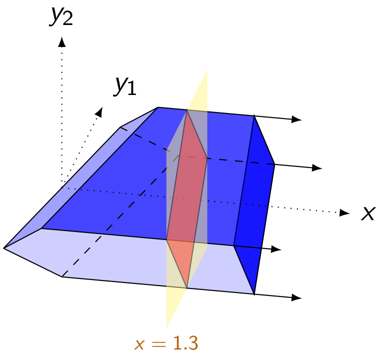
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



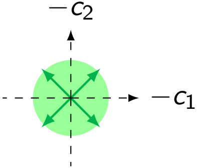

$P_x$ and $\mathcal{N}(P_x)$



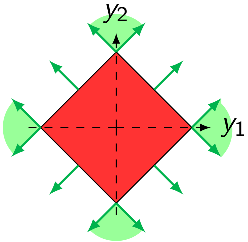

$P$ and $P_x$

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

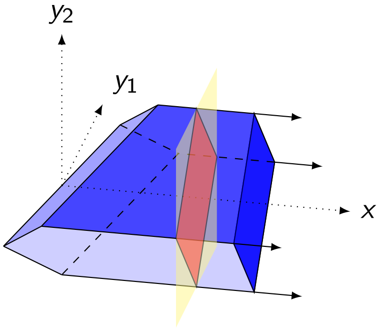
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



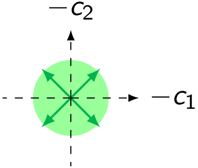

$P_x$ and $\mathcal{N}(P_x)$



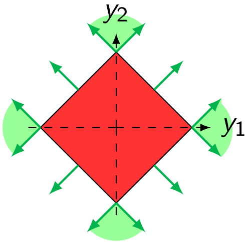

$P$ and $P_x$

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

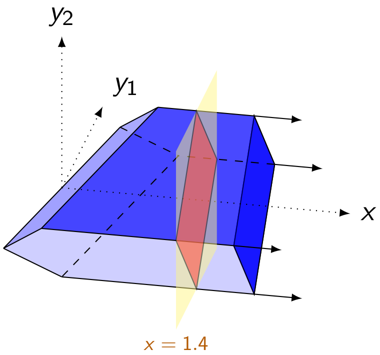
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



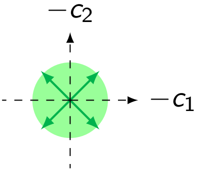

$P_x$ and $\mathcal{N}(P_x)$



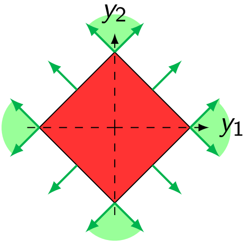

$P$ and $P_x$

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

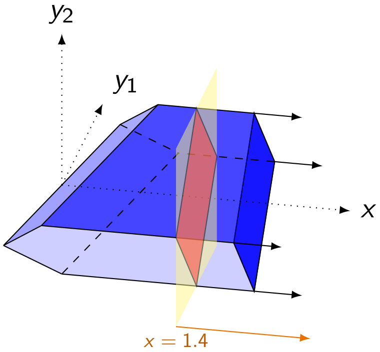
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



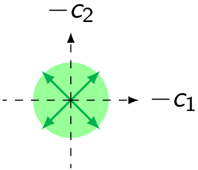

$P_x$ and $\mathcal{N}(P_x)$



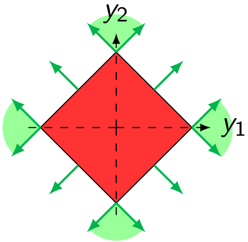

$P$ and $P_x$

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

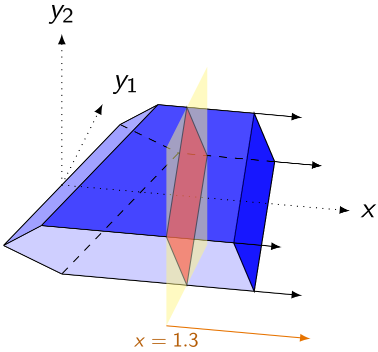
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



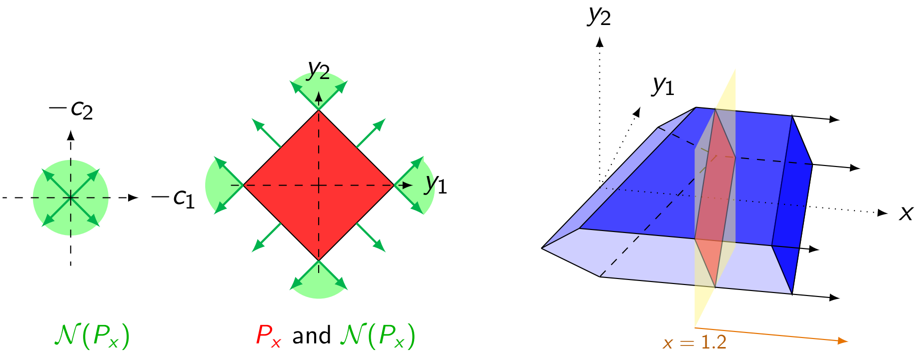

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

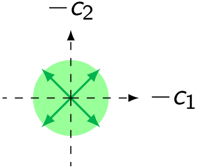
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



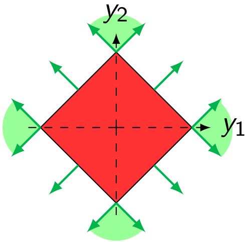

$\mathcal{N}(P_x)$ $\qquad$ $P_x$ and $\mathcal{N}(P_x)$ $\qquad$ $P$ and $P_x$

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

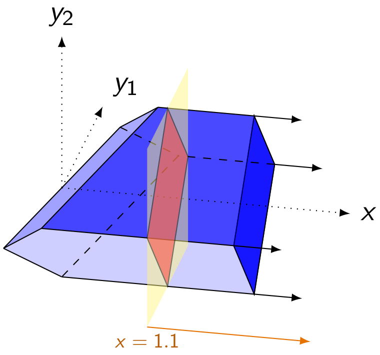
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



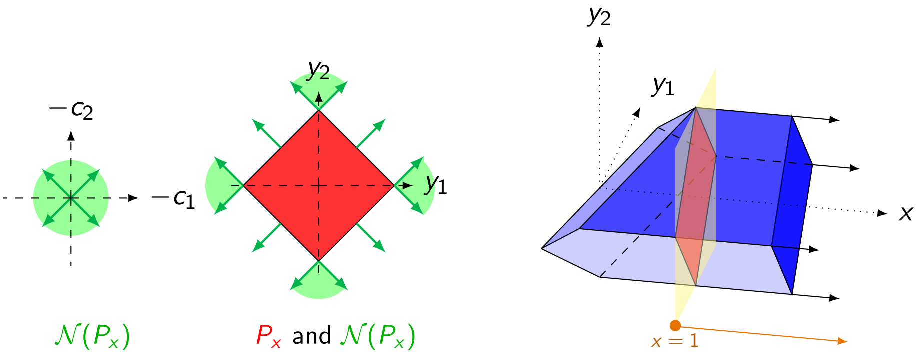

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

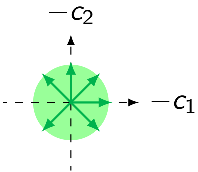
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



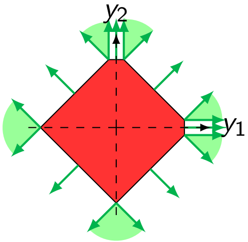

$\mathcal{N}(P_x)$ — $P_x$ and $\mathcal{N}(P_x)$ — $P$ and $P_x$

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

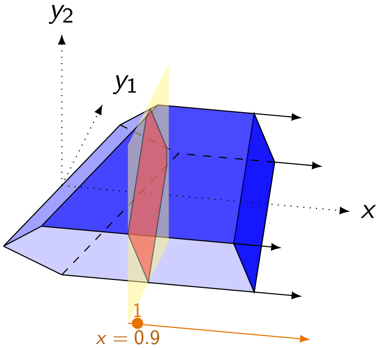
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



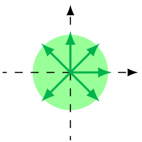

$P_x$ and $\mathcal{N}(P_x)$



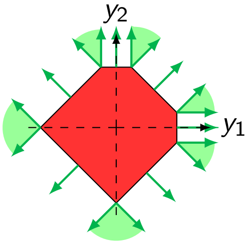

$P$ and $P_x$

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

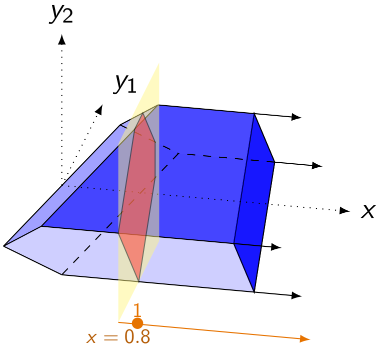
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



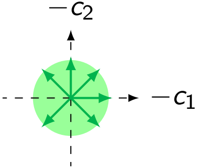

$P_x$ and $\mathcal{N}(P_x)$



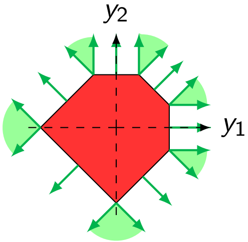

$P$ and $P_x$

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

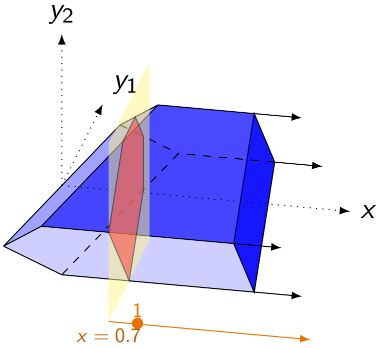
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



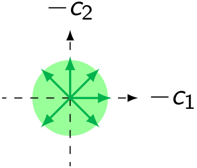

$P_x$ and $\mathcal{N}(P_x)$



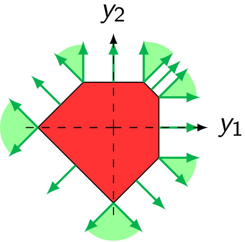

$P$ and $P_x$

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

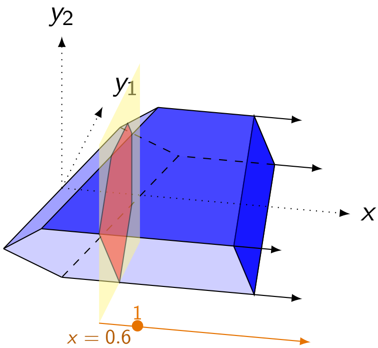
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



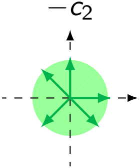

$P_x$ and $\mathcal{N}(P_x)$



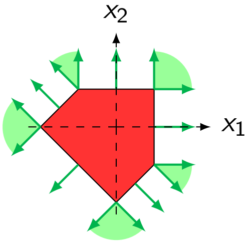

$P$ and $P_x$

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

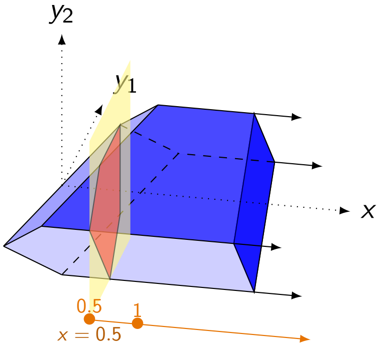
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



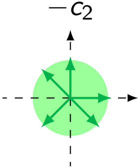

$P_x$ and $\mathcal{N}(P_x)$



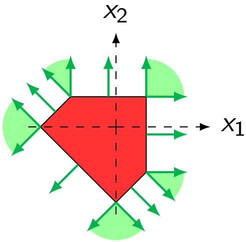

$P$ and $P_x$

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

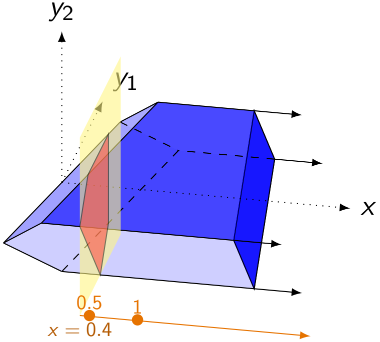
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



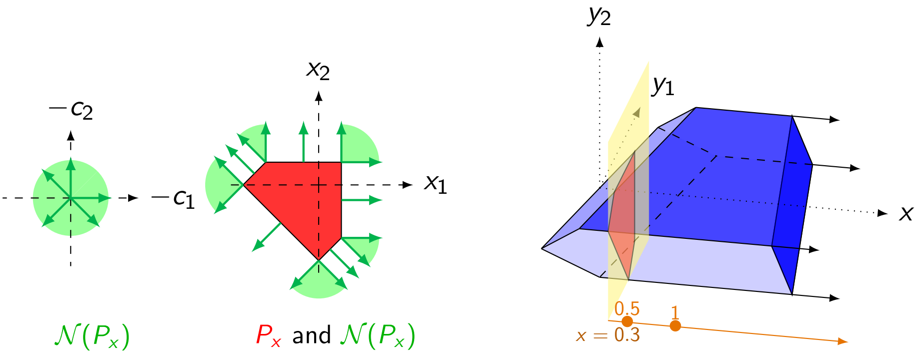

$P_x$ and $\mathcal{N}(P_x)$




$P$ and $P_x$

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

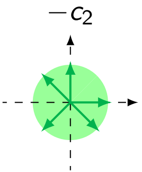
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$



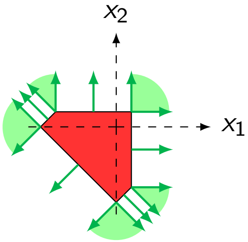

$\mathcal{N}(P_x)$ $\quad$ $P_x$ and $\mathcal{N}(P_x)$ $\quad$ $P$ and $P_x$

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

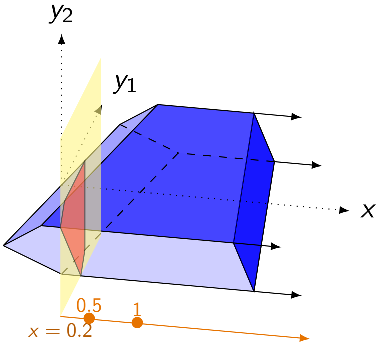
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



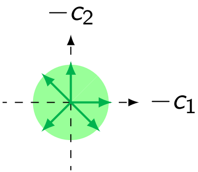

$P_x$ and $\mathcal{N}(P_x)$



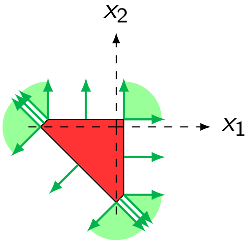

$P$ and $P_x$

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

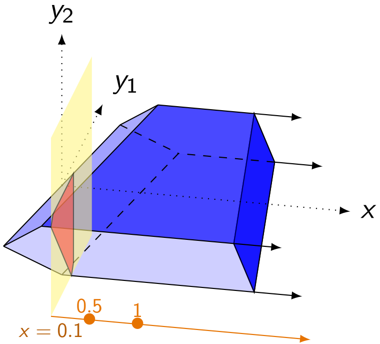
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



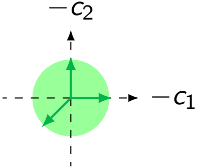

$P_x$ and $\mathcal{N}(P_x)$



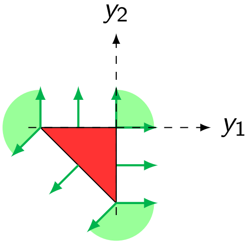

$P$ and $P_x$

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

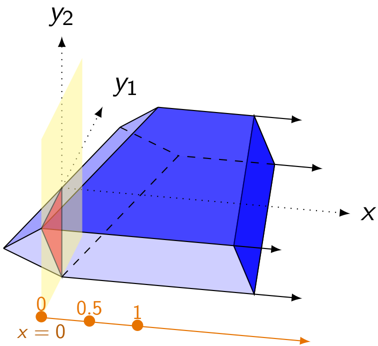
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



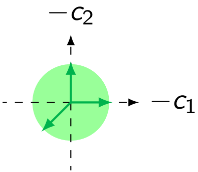

$P_x$ and $\mathcal{N}(P_x)$



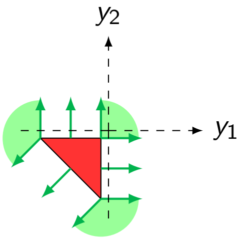

$P$ and $P_x$

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

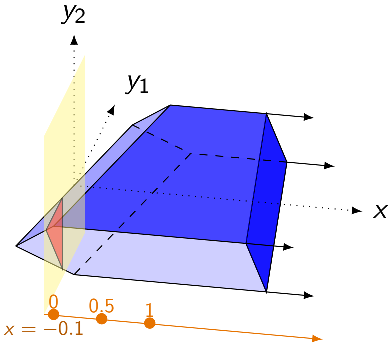
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x, y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



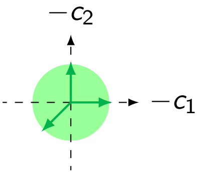

$P_x$ and $\mathcal{N}(P_x)$



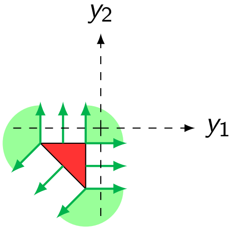

$P$ and $P_x$

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

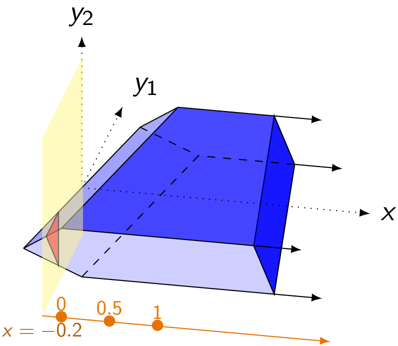
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



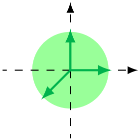

$P_x$ and $\mathcal{N}(P_x)$



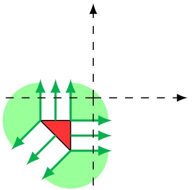

$P$ and $P_x$

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

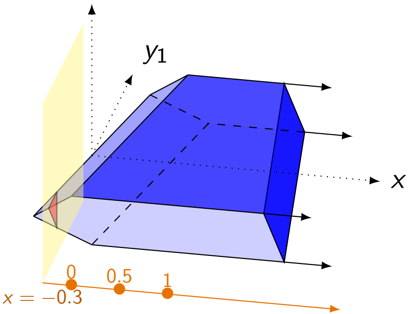
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$



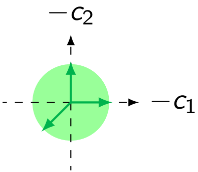

$P_x$ and $\mathcal{N}(P_x)$



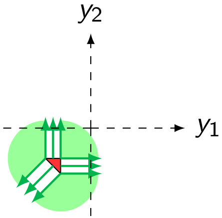

$P$ and $P_x$

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

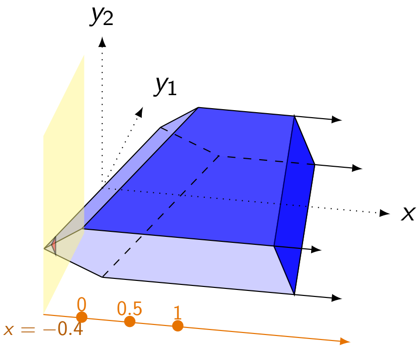
$$P_x := \{y \mid Ay + Bx \leqslant b\} \quad \text{and} \quad P := \{(x,y) \mid Ay + Bx \leqslant b\}$$




$\mathcal{N}(P_x)$




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



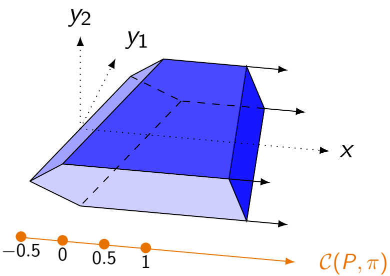

$P$ and $P_x$

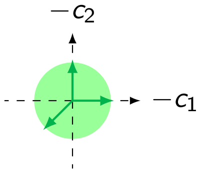
# What are the constant regions of $x \mapsto \mathcal{N}(P_x)$ ?

**Proposition**

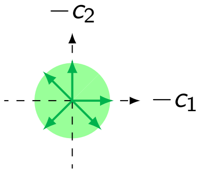
*There exists a collection* $\mathcal{C}(P, \pi)$ *called the* **chamber complex** *whose relative interior of cells are the constant regions of* $x \mapsto \mathcal{N}(P_x)$*.*

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

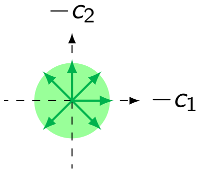


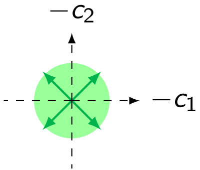

$\mathcal{N}_\sigma$ for $\sigma = [-0.5, 0]$




$\mathcal{N}_\sigma$ for $\sigma = [0, 0.5]$




$\mathcal{N}_\sigma$ for $\sigma = [0.5, 1]$




$\mathcal{N}_\sigma$ for $\sigma = [1, +\infty)$

# Chamber complex

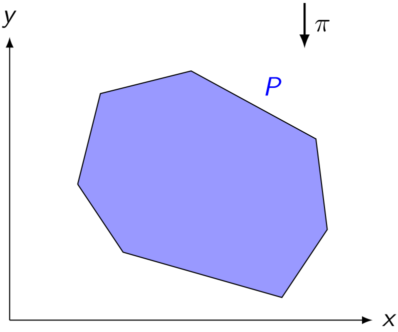
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x,y) \mapsto x$.

# Chamber complex

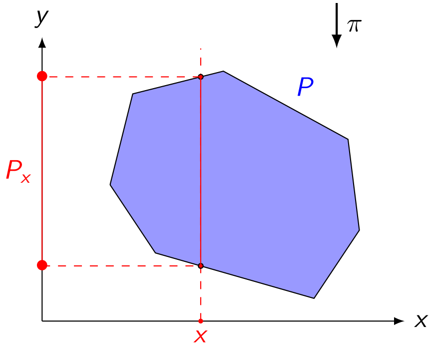
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x,y) \mapsto x$.

# Chamber complex

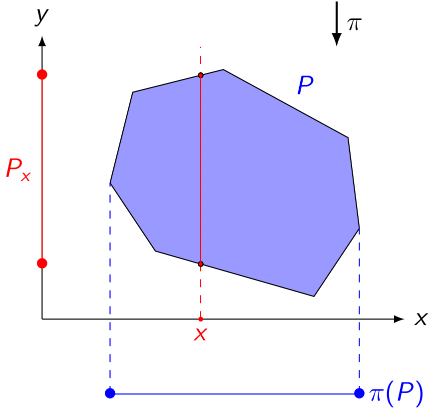
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x,y) \mapsto x$.

# Chamber complex

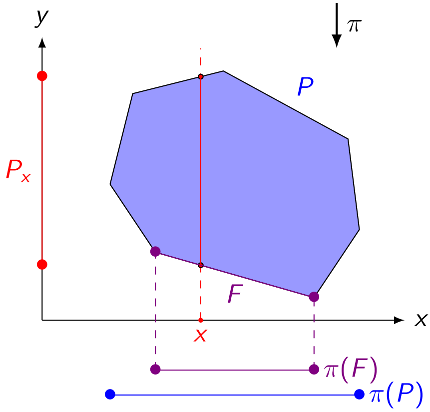
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x,y) \mapsto x$.

# Chamber complex

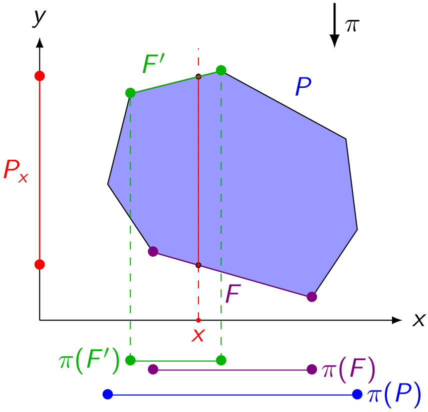
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x,y) \mapsto x$.

# Chamber complex

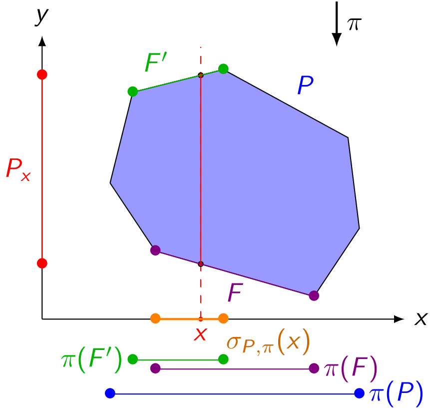
**Definition (Billera, Sturmfels 92)**

The *chamber complex* $\mathcal{C}(P,\pi)$ of $P$ along $\pi$ 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 $\pi$ is the projection $(x, y) \mapsto x$.

# Chamber complex

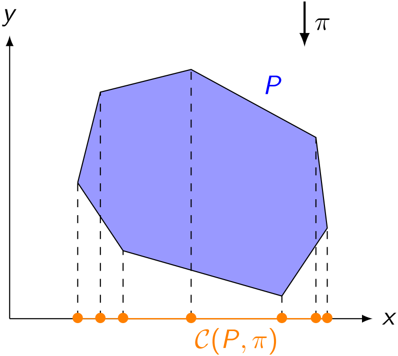
**Definition (Billera, Sturmfels 92)**

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

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

where

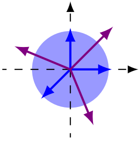
$$\sigma_{P,\pi}(x) := \bigcap_{F\in\mathcal{F}(P)\,\mid\, x\in\pi(F)} \pi(F)$$




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

# Common Refinement of Normal Fans

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



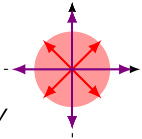
$\mathcal{N}_\sigma$ and $\check{\boldsymbol{c}}$

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

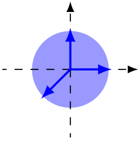
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



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

# Common Refinement of Normal Fans

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



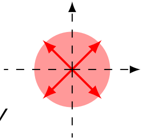
$\mathcal{N}_\sigma$

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

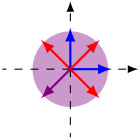
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



$\mathcal{N}_\tau$

We take the *common refinement*:

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



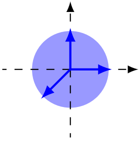
$\mathcal{N}_\sigma \wedge \mathcal{N}_\tau$

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.



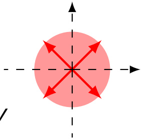
$\mathcal{N}_\sigma$

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

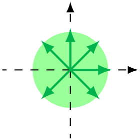
$$V(x') = \sum_{N \in \mathcal{N}_\tau} p_N \min_{y \in P_x} \check{c}_N^\top y$$



$\mathcal{N}_\tau$

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 $E$*

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

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

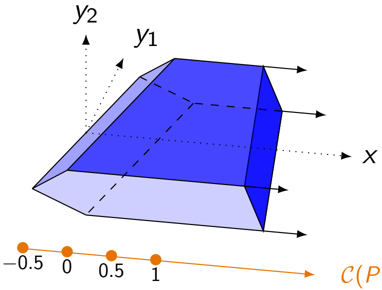
The weighted fiber polyhedron is a Minkowski integral with respect to the distribution $d\mathbb{P}(c)$

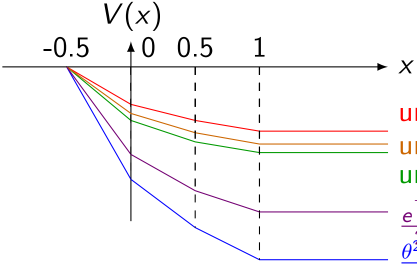
$\rightsquigarrow$ extension of fiber polytope (uniform distribution) of

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

# Explicit computation of the example

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




Different distributions of $\boldsymbol{c}$:

uniform on norm 1 ball

uniform on norm 2 ball

uniform on norm $\infty$ ball

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

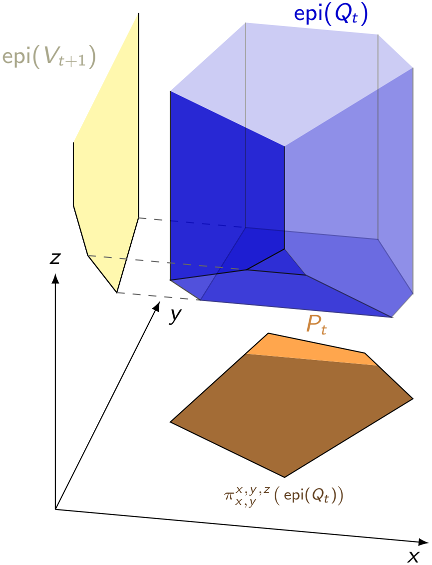
$\dfrac{\theta^2 e^{-\theta\|c\|_1}}{4} dc$

# Contents

# Multistage uniform and universal exact quantization

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

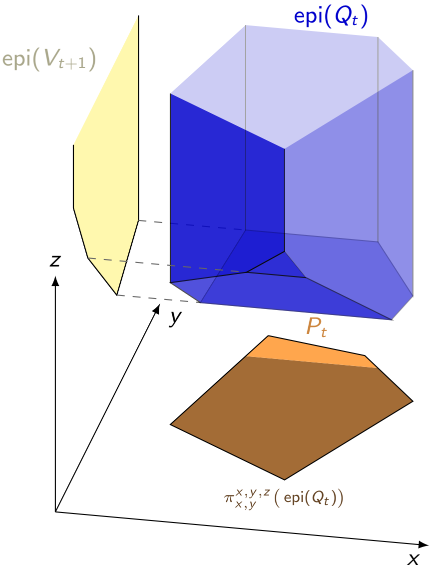
with $Q_t(x, y) := V_{t+1}(y) + \mathbb{I}_{(x,y) \in P_t}$.



# Multistage uniform and universal exact quantization

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

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

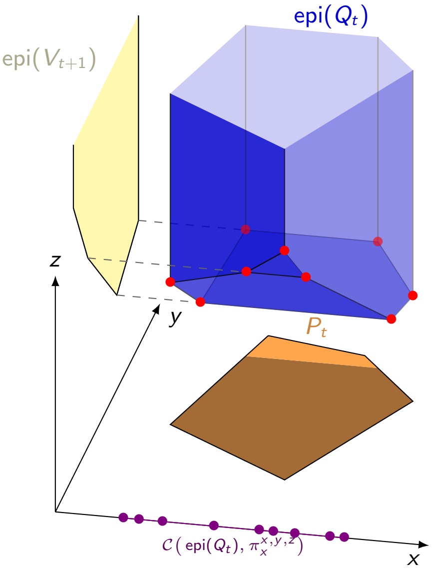


# Multistage uniform and universal exact quantization

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

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 $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$



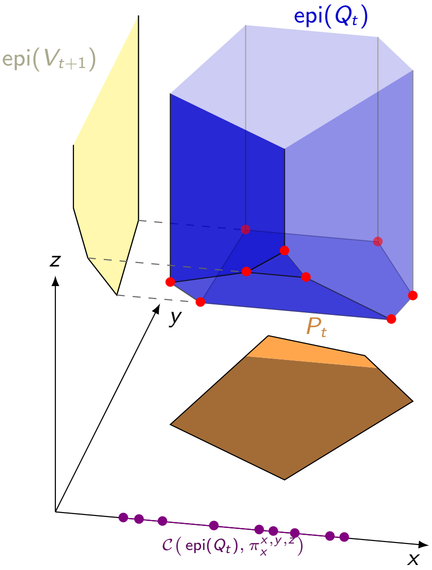
# Multistage uniform and universal exact quantization

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

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

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

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



# Multistage uniform and universal exact quantization

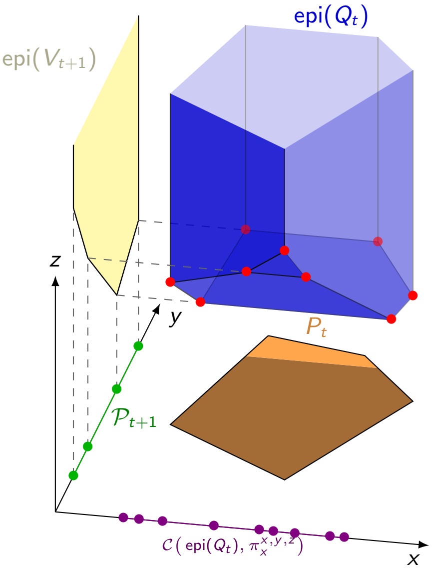
$$V_t(x) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{\substack{y\in\mathbb{R}^{n_t}\\ z\in\mathbb{R}}} & \boldsymbol{c}_t^\top y + z \\ & \\ \text{s.t.} & (x,y,z) \in \mathrm{epi}(Q_t) \end{array}\right]$$

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)



# Multistage uniform and universal exact quantization

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

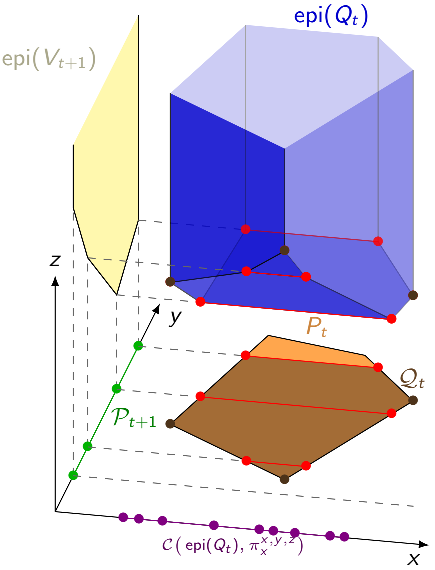
with $Q_t(x,y) := V_{t+1}(y) + \mathbb{I}_{(x,y)\in P_t}$.

➥ $V_t$ 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 $c_{t+1}, \cdots, c_T$ !

$V_{t+1}$ 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

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

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

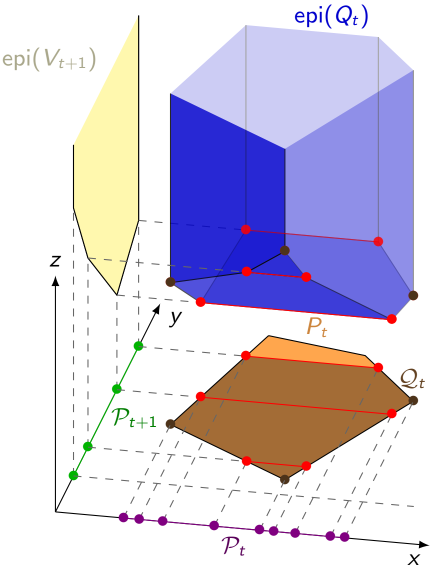
➥ $V_t$ affine, $x \mapsto \mathcal{N}(P_x)$ constant on $\mathcal{C}(\mathrm{epi}(Q_t), \pi_x^{x,y,z})$

⚠ 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)

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

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



# 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 $\boldsymbol{c}_{t+1}, \cdots, \boldsymbol{c}_T$ !

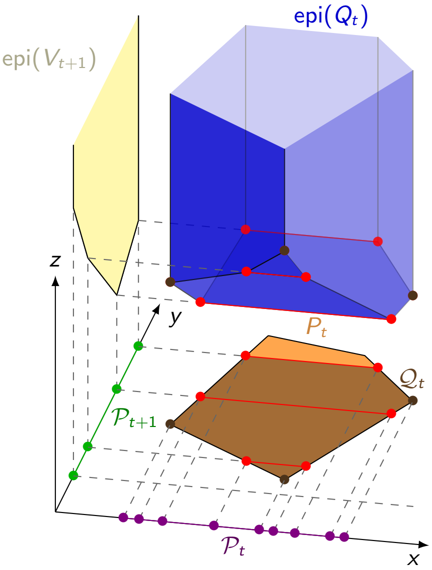
$V_{t+1}$ affine on $\mathcal{P}_{t+1}$ (by assumption)

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

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

[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

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

**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

# 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

$$\min_{x \in \mathbb{R}^n} c_1^\top x + \mathbb{E}\left[\begin{array}{l} \min\limits_{y \in \mathbb{R}^m} \boldsymbol{c}_2^\top y \\ \text{s.t. } A_2 y + B_2 x \leqslant b_2 \end{array}\right]$$
$$\text{s.t. } A_1 x \leqslant b_1$$

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

# Earlier and new complexity results

Volume of a polytope

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

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

2-stage linear problem

$$\min_{x \in \mathbb{R}^n} c_1^\top x + \mathbb{E}\left[\begin{array}{l} \min_{y \in \mathbb{R}^{m}} \boldsymbol{c}_2^\top y \\ \text{s.t. } A_2 y + B_2 x \leqslant b_2 \end{array}\right]$$
$$\text{s.t. } A_1 x \leqslant b_1$$

- $\sharp P$-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

$$\min_{x \in \mathbb{R}^n} c_1^\top x + \mathbb{E}\begin{bmatrix} \min_{y \in \mathbb{R}^m} \boldsymbol{c}_2^\top y \\ \text{s.t. } A_2 y + B_2 x \leqslant b_2 \end{bmatrix}$$
$$\text{s.t. } A_1 x \leqslant b_1$$

- $\sharp P$-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 $\varepsilon$-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 $\varepsilon$-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

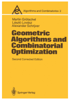
**Theorem (FGL21: MSLP is polynomial for fixed dimensions)**

*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 $\varepsilon$-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

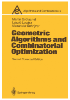
**Theorem (FGL21: MSLP is polynomial for fixed dimensions)**

*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 $\varepsilon$-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 $\varepsilon$, 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

# 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 | × | ? | ✓ |
| 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

| | $A$ | $(B, b)$ | $c$ |
|---|---|---|---|
| Local | × | ? | ✓ |
| Uniform | × | × | ✓ |

Duality result

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

➥ 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

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

➥ 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 $x$,

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

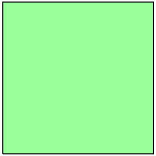
where,

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

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

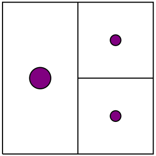
# Contents

# Partitioned cost-to-go functions (recalls)



$\xi_t$ continuous

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

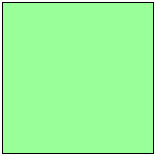


$\check{\xi}_t$ partitioned

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

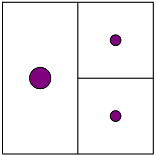
- $\hat{V}(x, \cdot)$ is convex
  ➥ $V_{\mathcal{P}} \leqslant V$.
- $\hat{V}\big(\cdot, \mathbb{E}[\xi | P]\big)$ is polyhedral
  ➥ $V_{\mathcal{P}}$ is polyhedral.

# Partitioned cost-to-go functions (recalls)



$\xi_t$ continuous

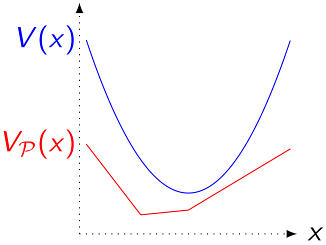
$$V(x) = \mathbb{E}\Big[\hat{V}(x, \xi)\Big]$$



$\check{\xi}_t$ partitioned

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

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

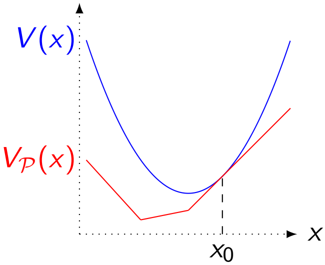


# Adapted partition

**Definition**

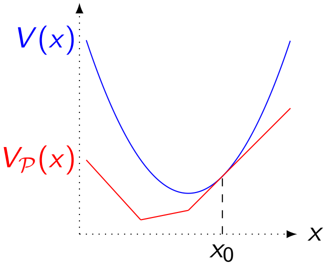
*A partition $\mathcal{P}$ is adapted to $x_0$ if*

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




[1] Can be extended to generic random $\boldsymbol{c}$ and finitely supported $\boldsymbol{A}$

# Adapted partition




**Definition**

*A partition $\mathcal{P}$ is adapted to $x_0$ if*

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

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)**

$\mathcal{R}_x := \left\{E_{N,x} \mid N \in \mathcal{N}(D_q)\right\}$ *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*[1].
*Bonus: necessary and sufficient condition for a partition to be adapted*

[1] Can be extended to generic random $\boldsymbol{c}$ and finitely supported $\boldsymbol{A}$

# Contents

# 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 $\min_{x \in X} c_1^\top x + V_{\mathcal{P}^{k-1}}(x)$ ;
  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**

**Algorithm 1:** General framework for APM.

$$\min_{x \in X} c_1^\top x + V_{\mathcal{P}}(x)$$

is equivalent to

$$\begin{aligned} \min_{x \in X, (y_P)_{P \in \mathcal{P}}} \quad & c_1^\top x + \sum_{P \in \mathcal{P}} \mathbb{P}[P] c_2^\top y_P \\ & A y_P + \mathbb{E}\left[\boldsymbol{B} \middle| P\right] x \leqslant \mathbb{E}\left[\boldsymbol{b} \middle| P\right] \qquad , \forall P \in \mathcal{P} \end{aligned}$$

# 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 $\min_{x \in X} c_1^\top x + V_{\mathcal{P}^{k-1}}(x)$ ;
  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**

**Algorithm 1:** General framework for APM.

$$\min_{x \in X} c_1^\top x + V_{\mathcal{P}}(x)$$

is equivalent to

$$\begin{aligned} \min_{x \in X, (y_P)_{P \in \mathcal{P}}} \quad & c_1^\top x + \sum_{P \in \mathcal{P}} \mathbb{P}\big[P\big] c_2^\top y_P \\ & A y_P + \mathbb{E}\big[\boldsymbol{B} | P\big] x \leqslant \mathbb{E}\big[\boldsymbol{b} | P\big] \qquad , \forall P \in \mathcal{P} \end{aligned}$$

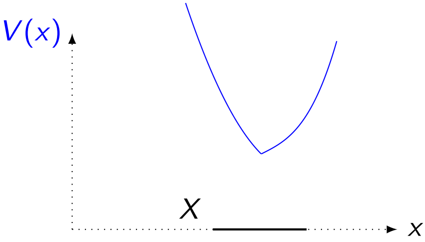
# A (partial) comparison between partition based results

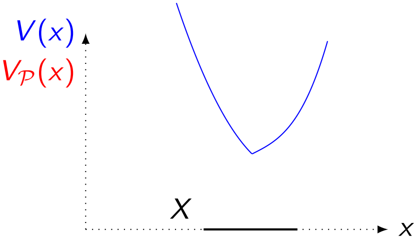
| Paper | Song, Luedtke (2015) | Ramirez-Pico, Moreno (2020) | FL (2021) |
|---|---|---|---|
| Non-finite supp($\xi$) | $\times$ | $\checkmark$ | $\checkmark$ |
| Explicit oracle | $\checkmark$ | $\times$ | $\checkmark$ |
| Proof of convergence | $\checkmark$ | $\times$ | $\checkmark$ |
| Complexity result | $\times$ | $\times$ | $\checkmark$ |
| Fast iteration | $\checkmark$ | $\times$ | $\times$ |

# Contents

# Link with Benders decomposition and L-shaped

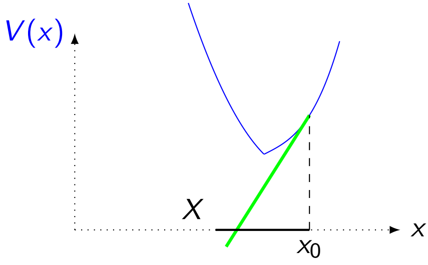
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.



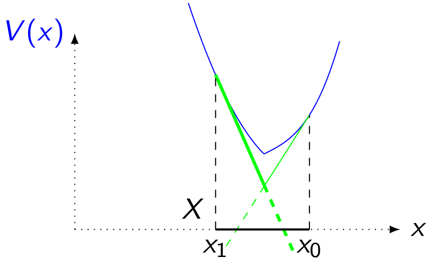
# Link with Benders decomposition and L-shaped

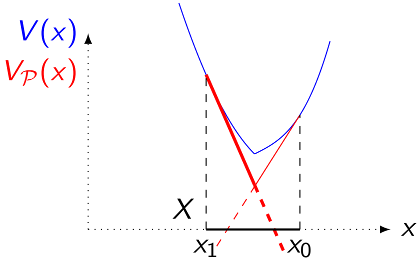
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.



# Link with Benders decomposition and L-shaped

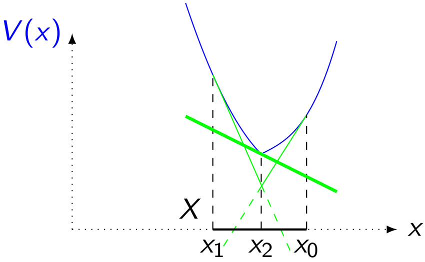
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

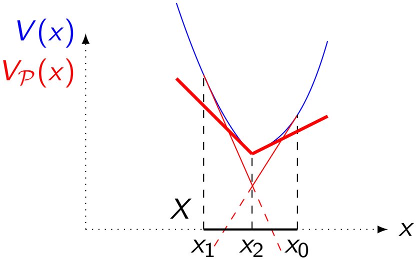


# Link with Benders decomposition and L-shaped

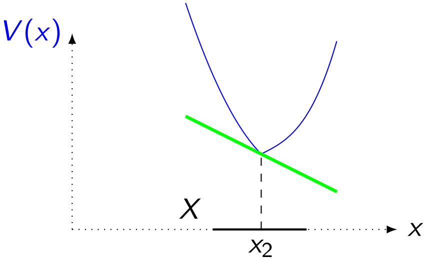
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.

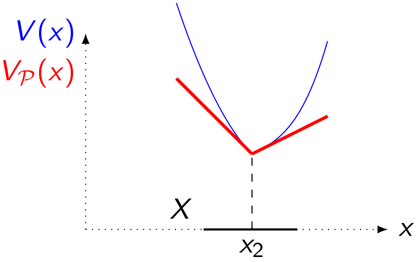


# Link with Benders decomposition and L-shaped

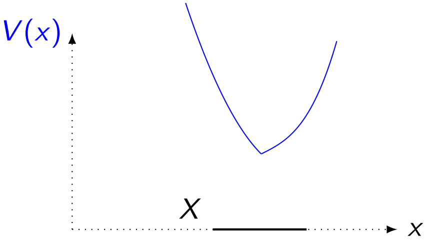
Partition based method can be seen as a tangent cone method: a cutting plane method where we add all active cuts instead of a single one.



# 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* $x \to c_1^\top x + V(x)$ *is Lipschitz with constant* $L$ *then the partition based method finds an* $\varepsilon$*-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.

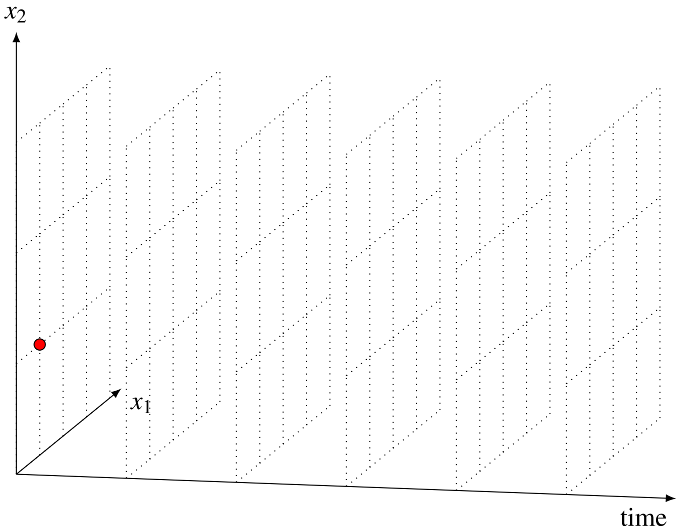
# Contents

# 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

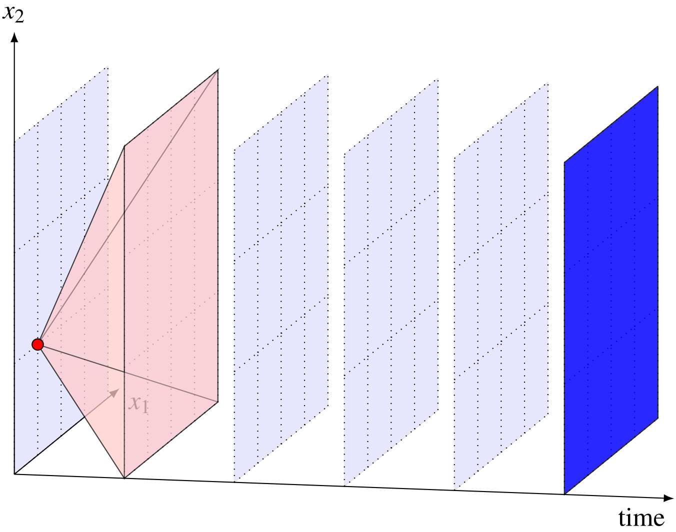
➥ All with finitely supported distribution

# Trajectory Following Dynamic Programming



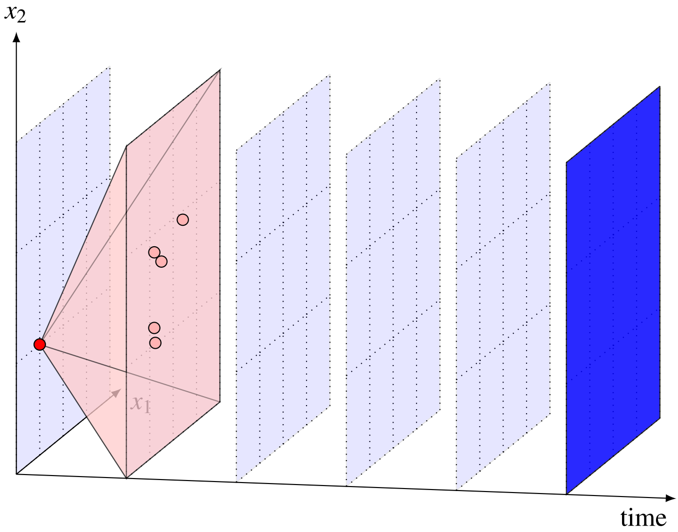

Thanks again Vincent !

# Trajectory Following Dynamic Programming



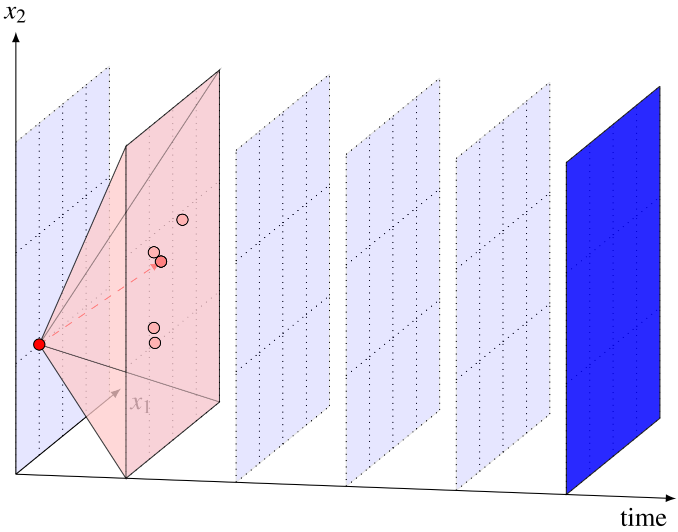

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



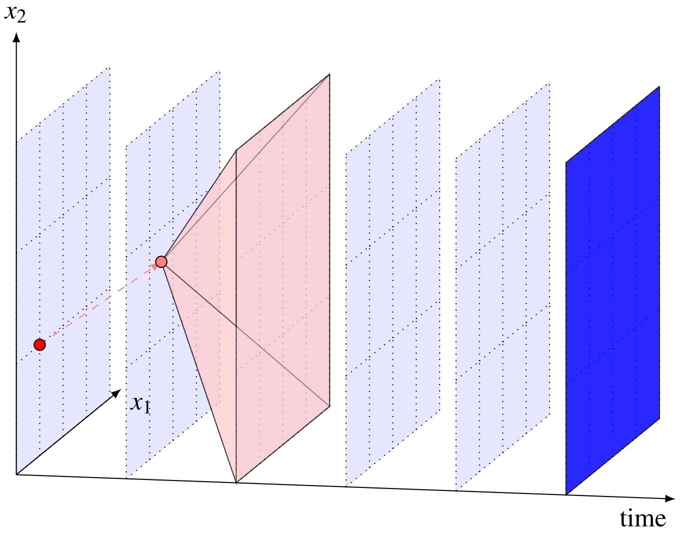

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



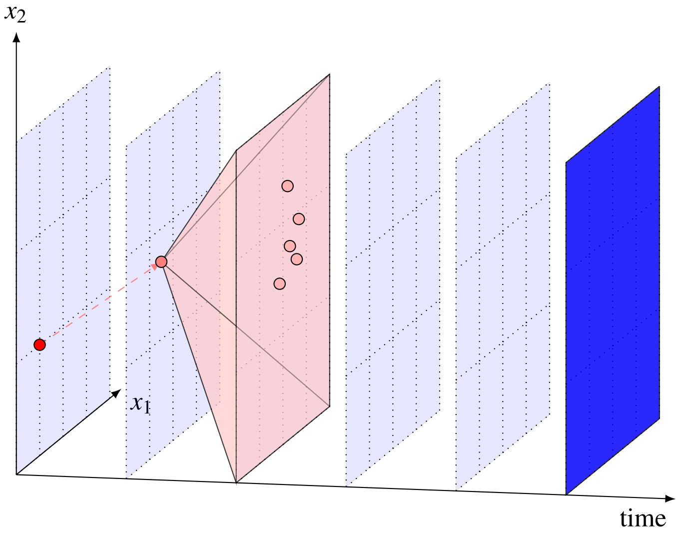

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



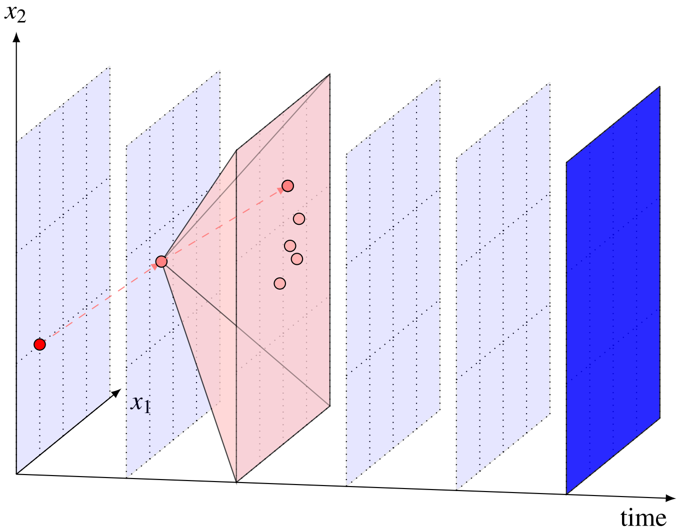

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



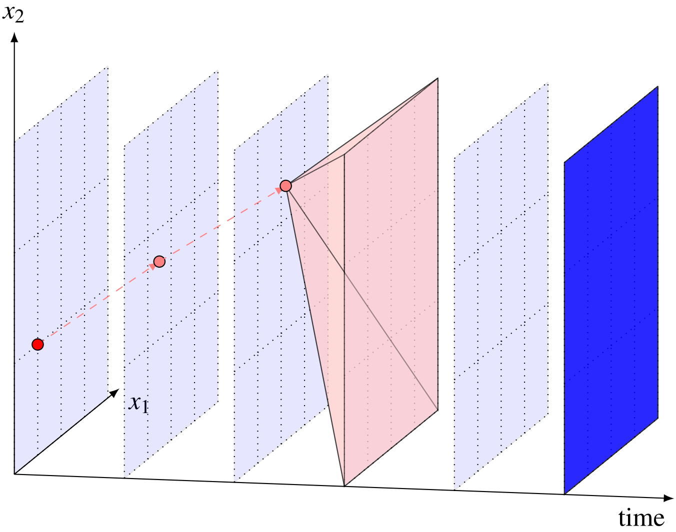

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



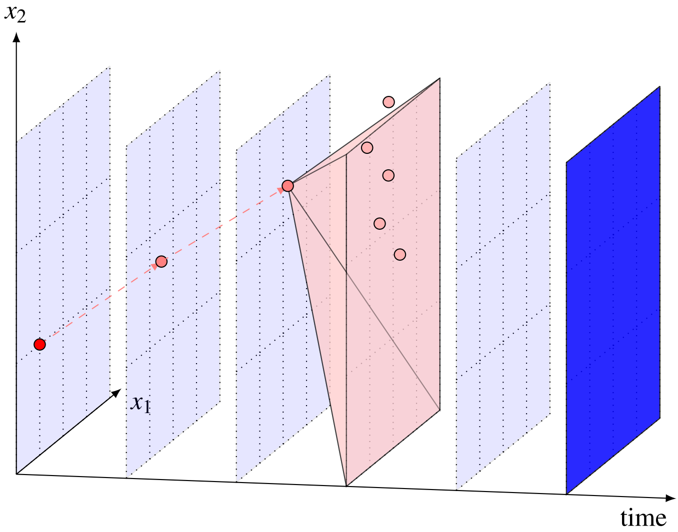

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



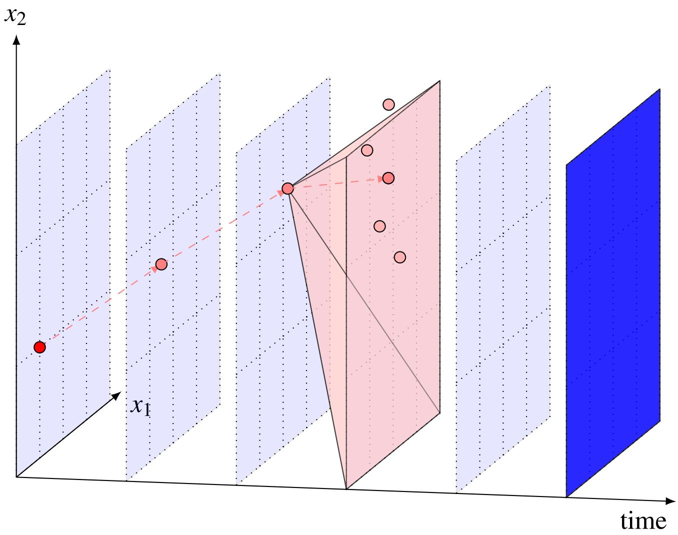

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



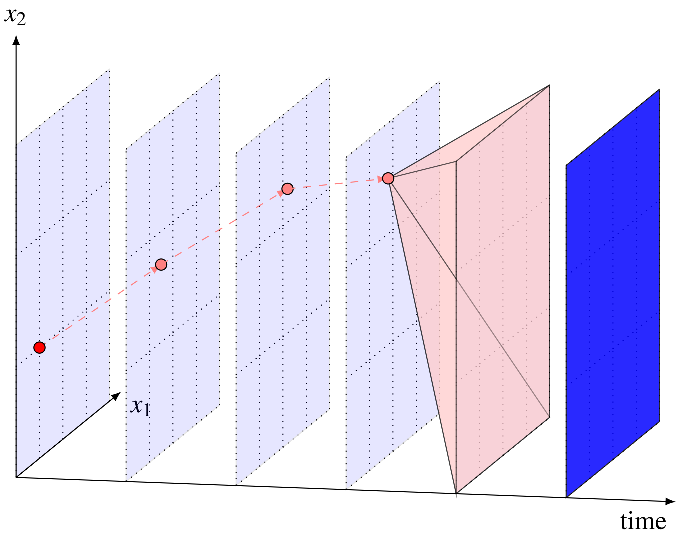

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



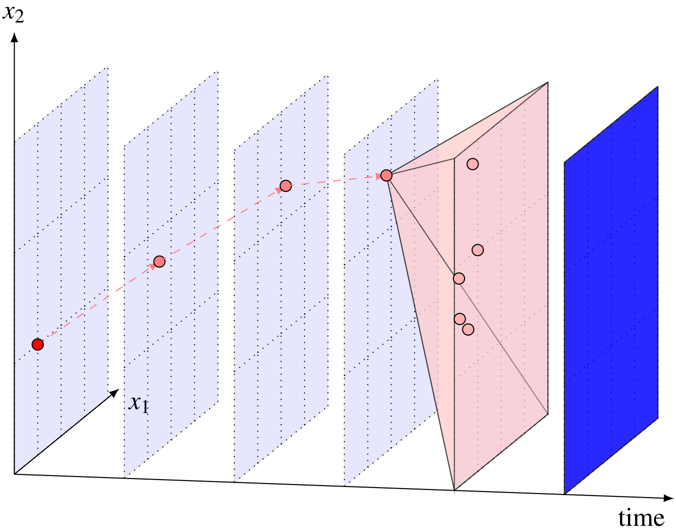

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



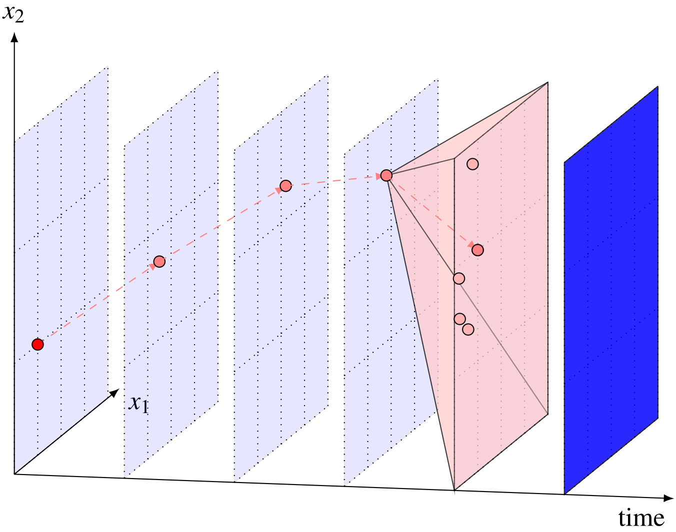

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



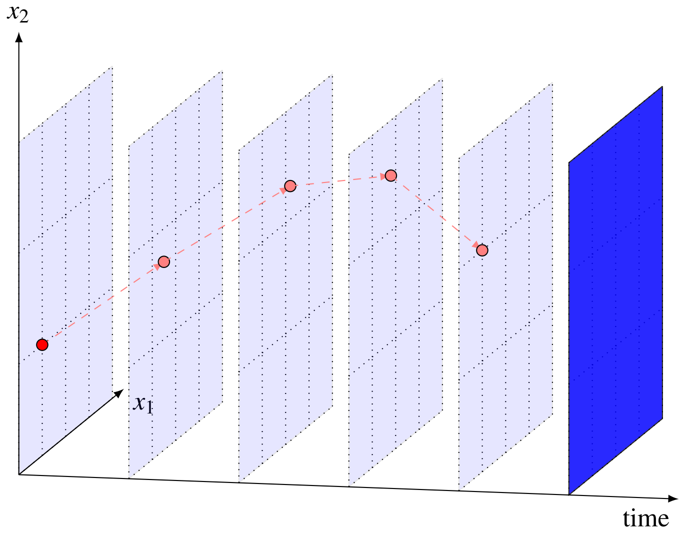

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



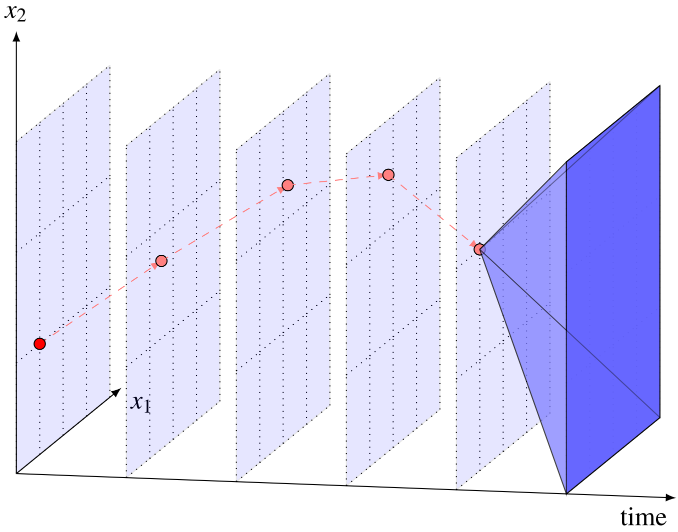

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



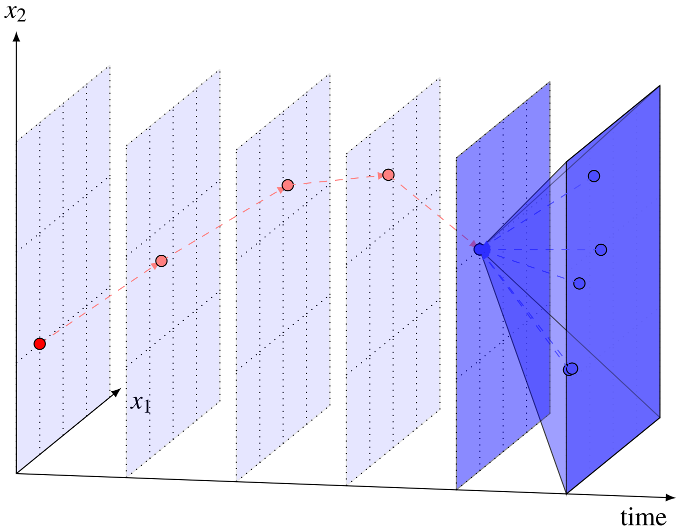

First forward pass : computing trajectory

# Trajectory Following Dynamic Programming



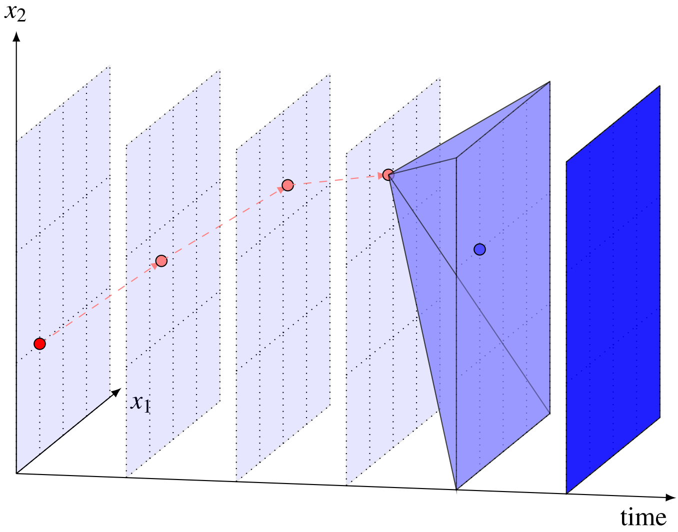

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



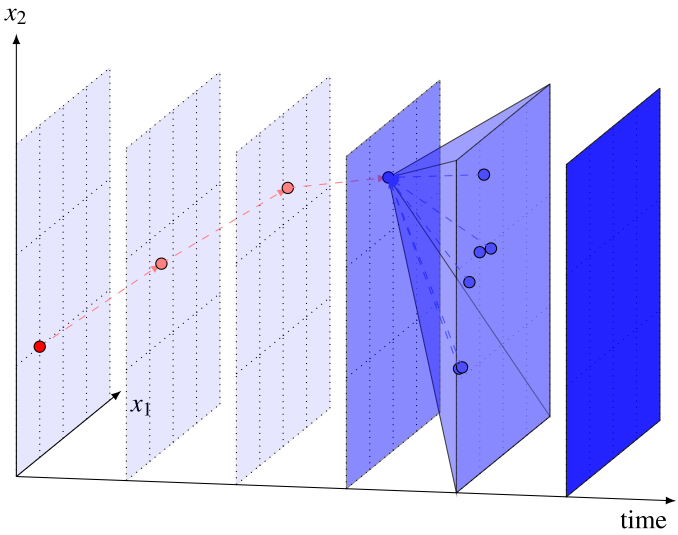

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



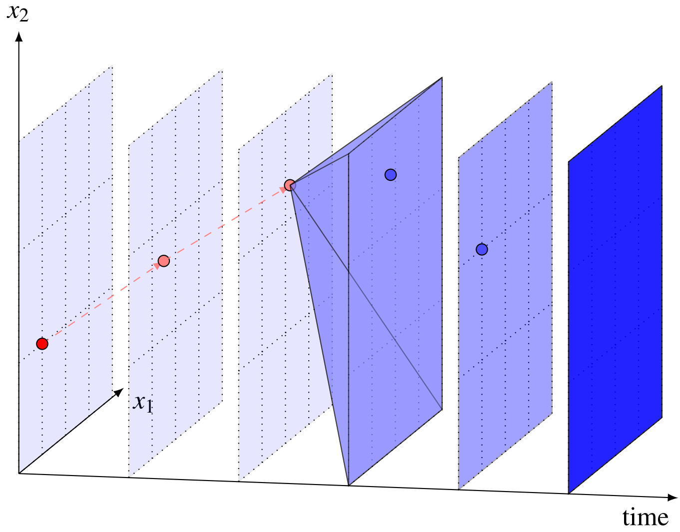

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



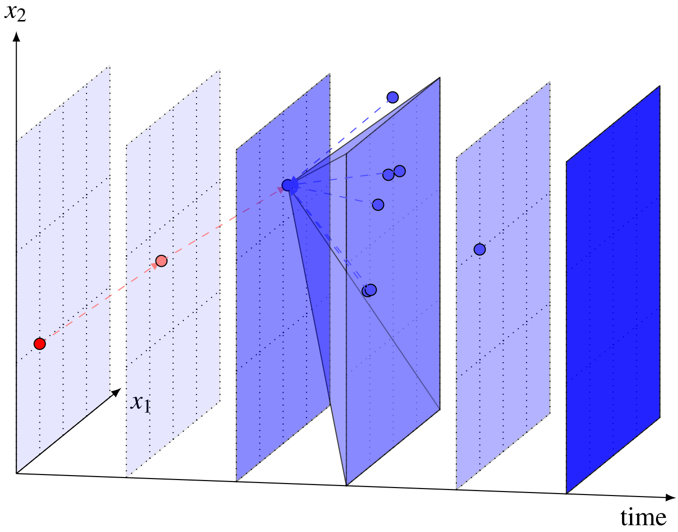

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



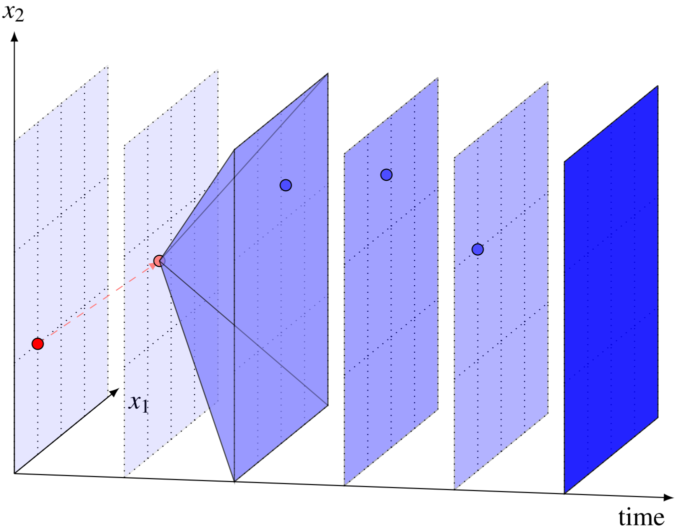

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



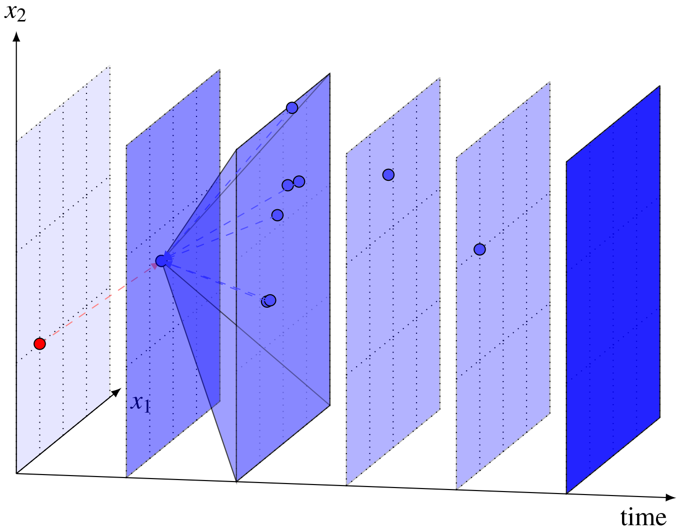

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



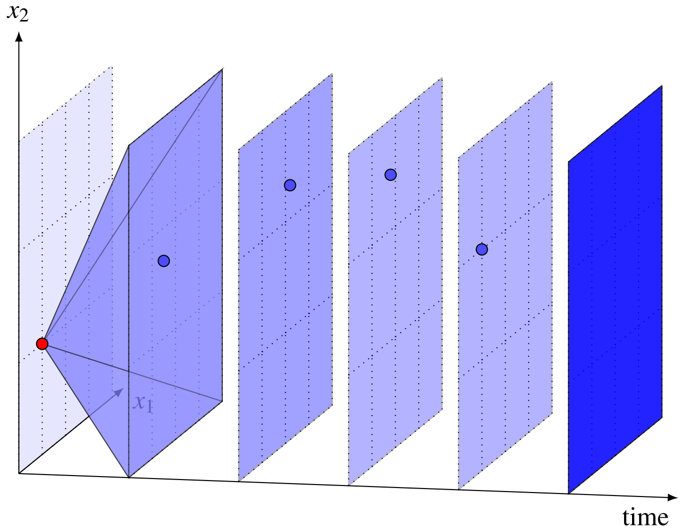

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



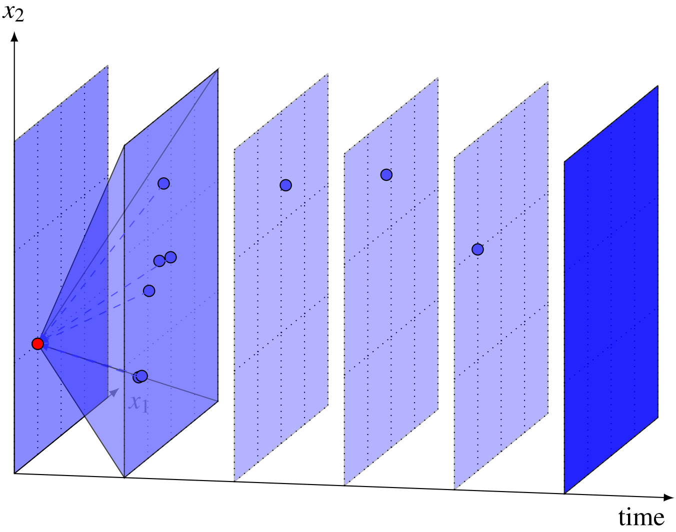

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



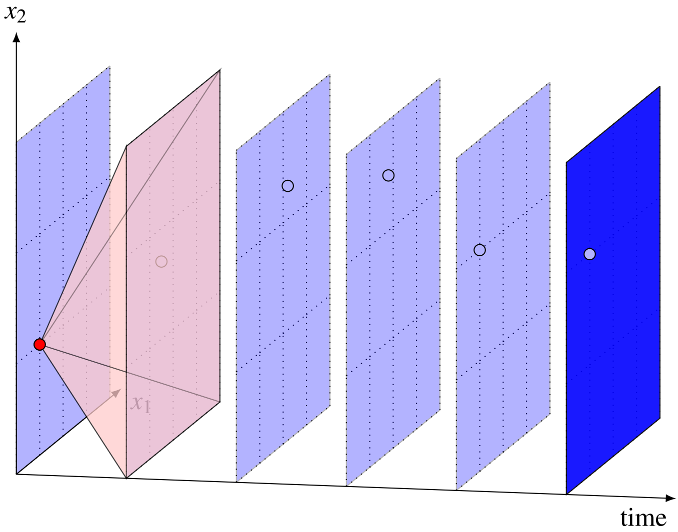

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



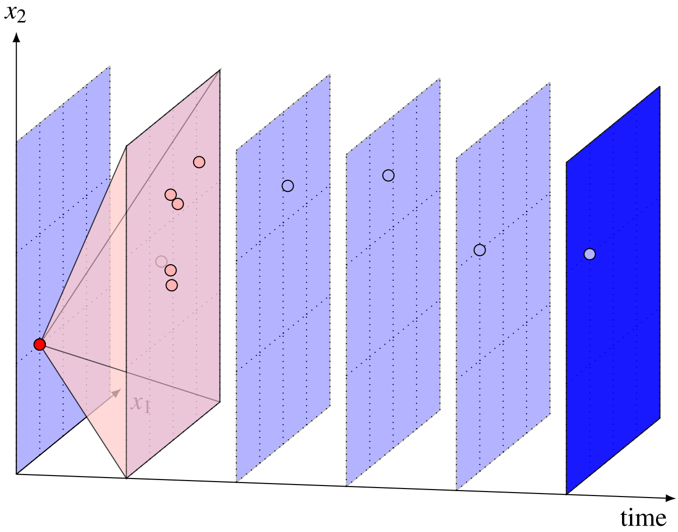

First backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



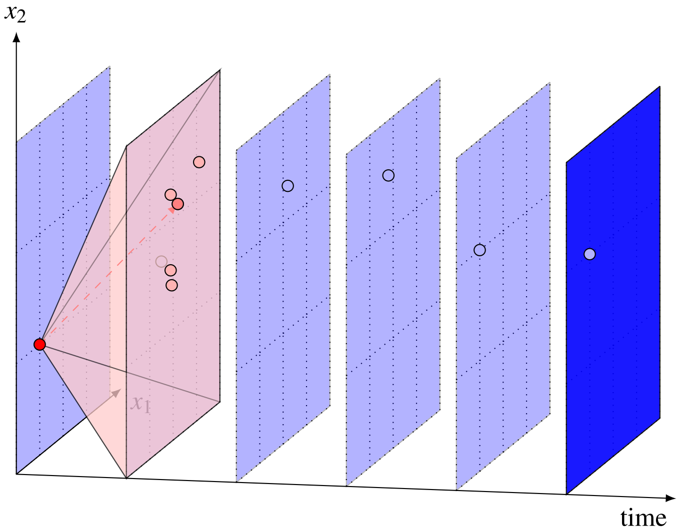

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



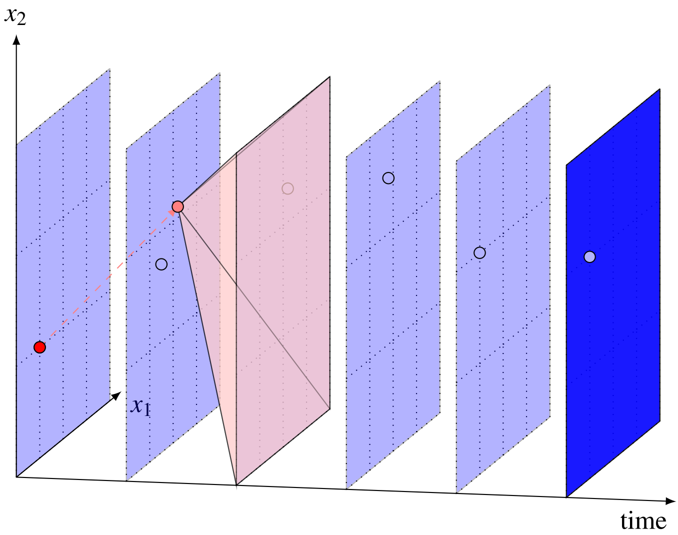

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



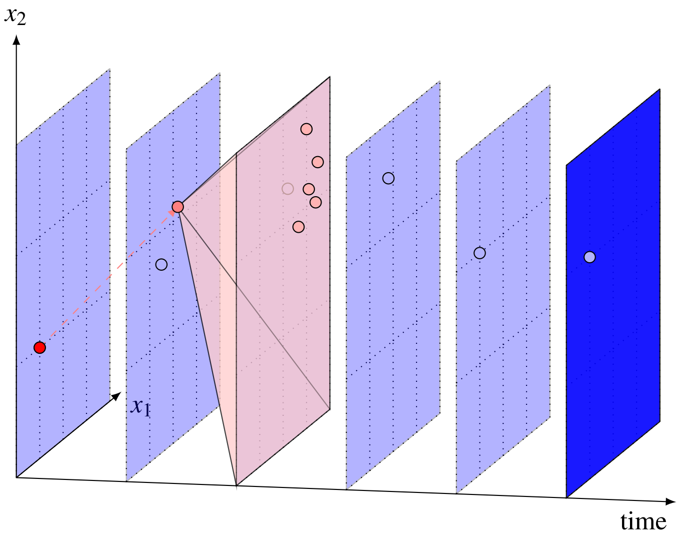

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



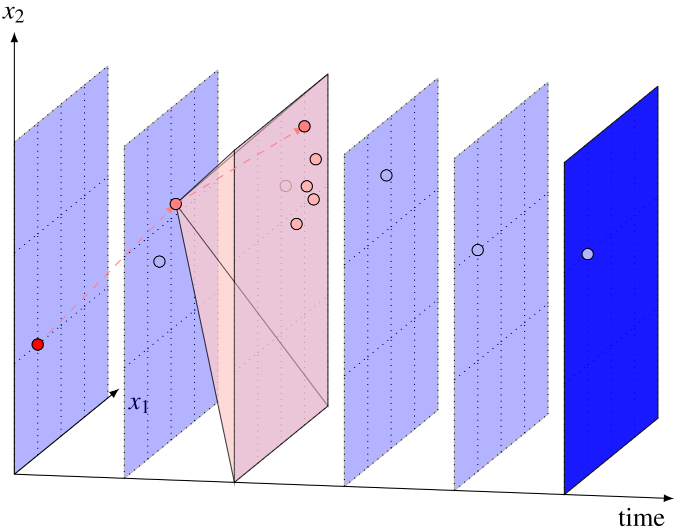

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



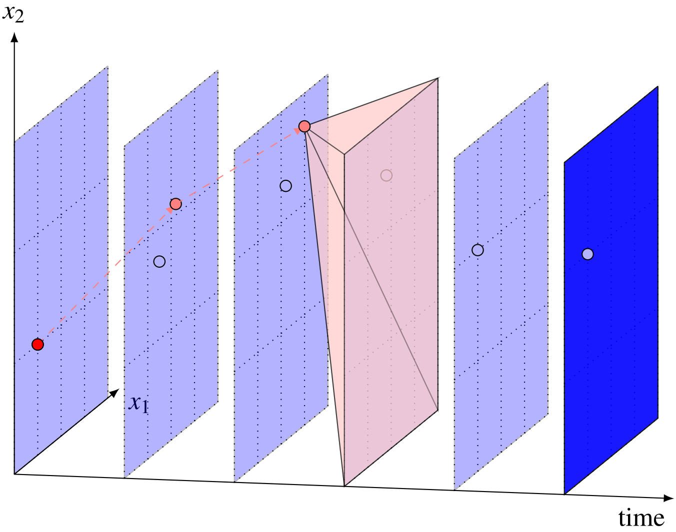

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



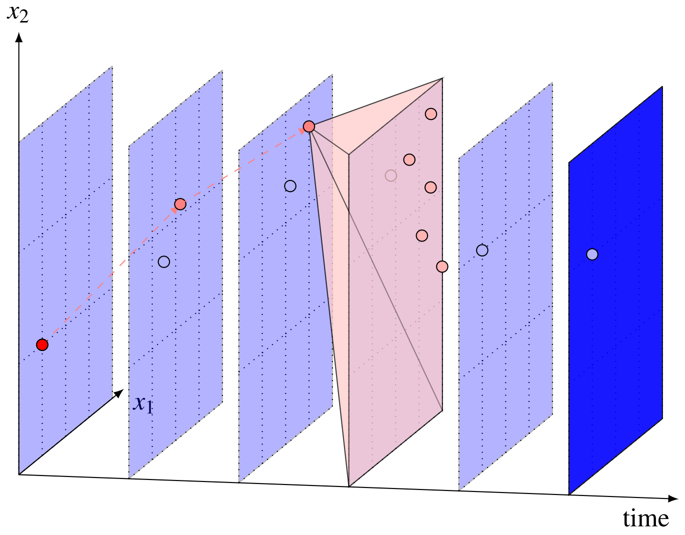

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



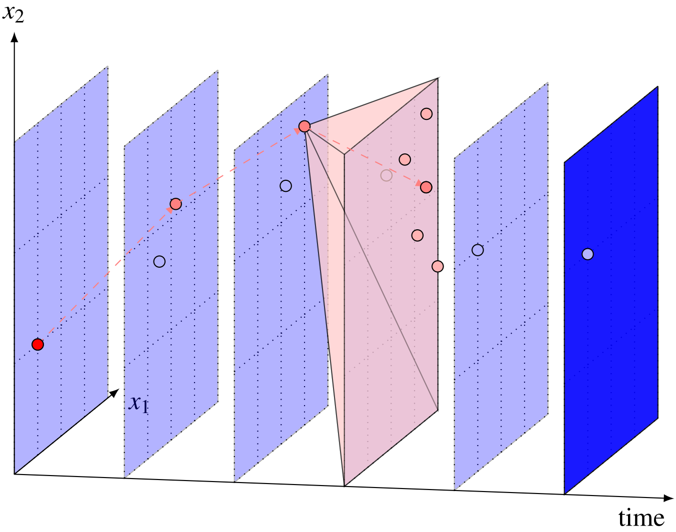

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



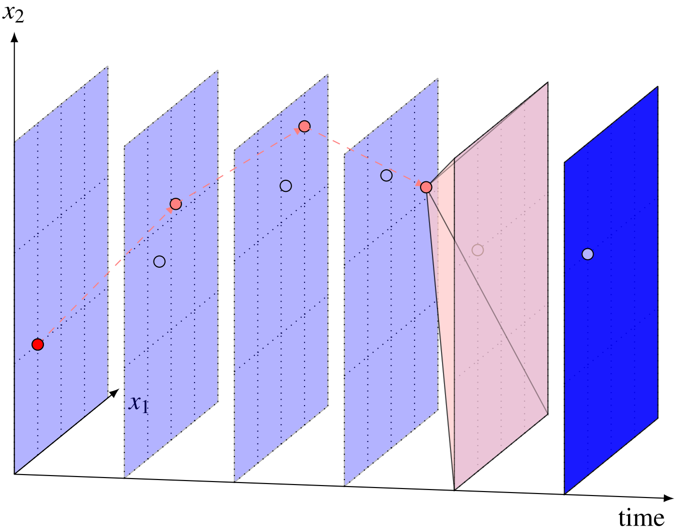

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



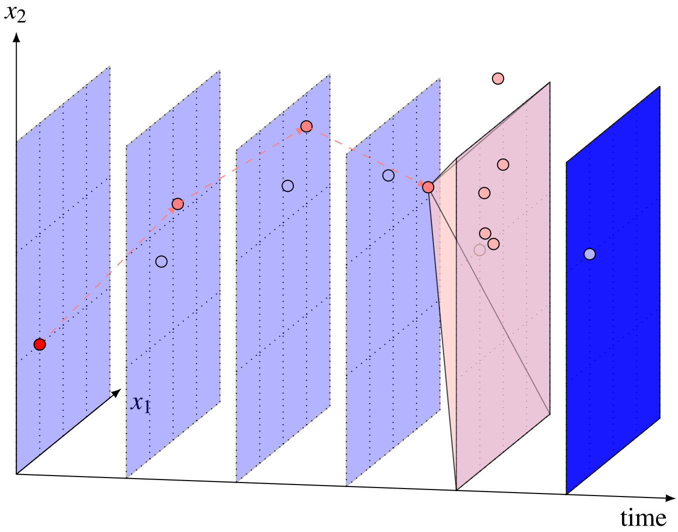

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



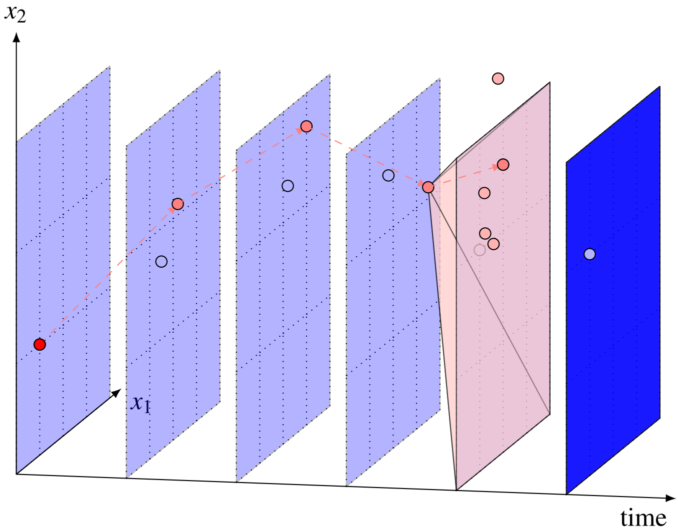

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



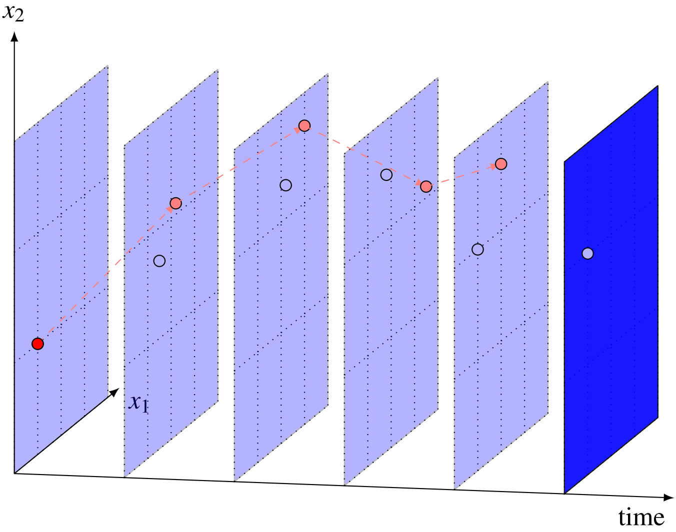

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



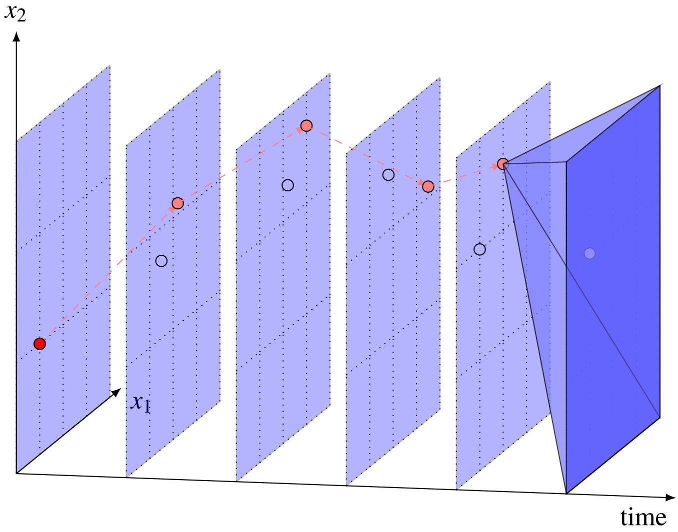

second forward pass : computing trajectory

# Trajectory Following Dynamic Programming



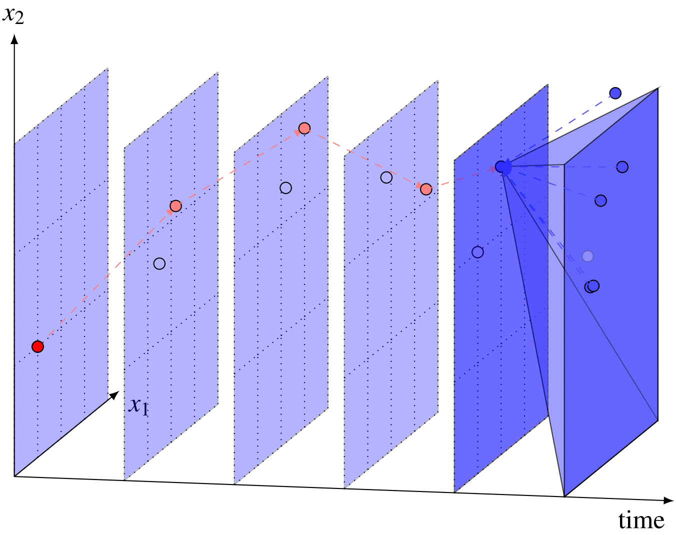

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



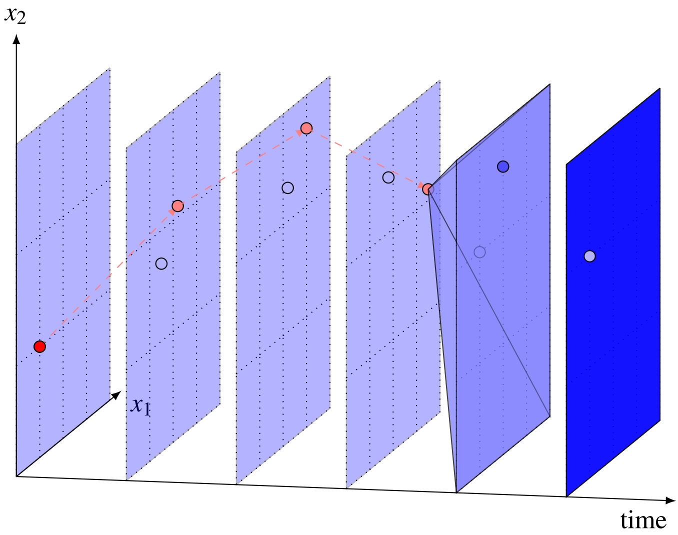

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



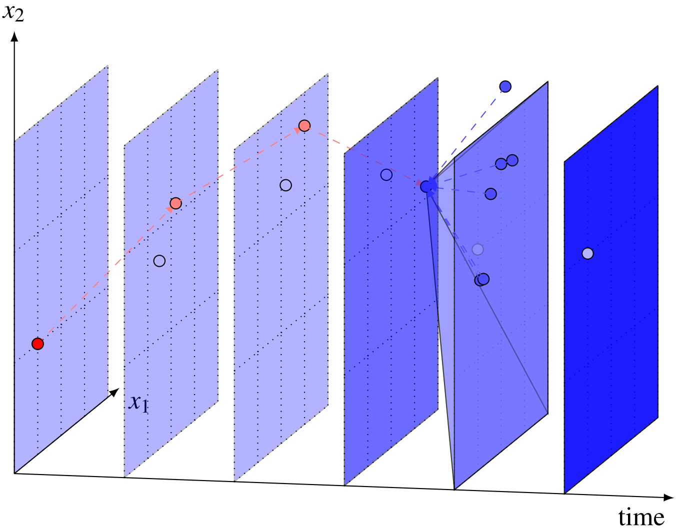

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



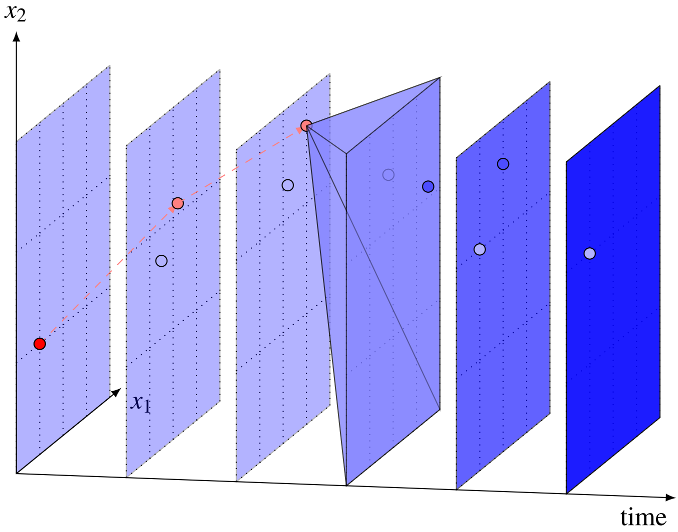

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



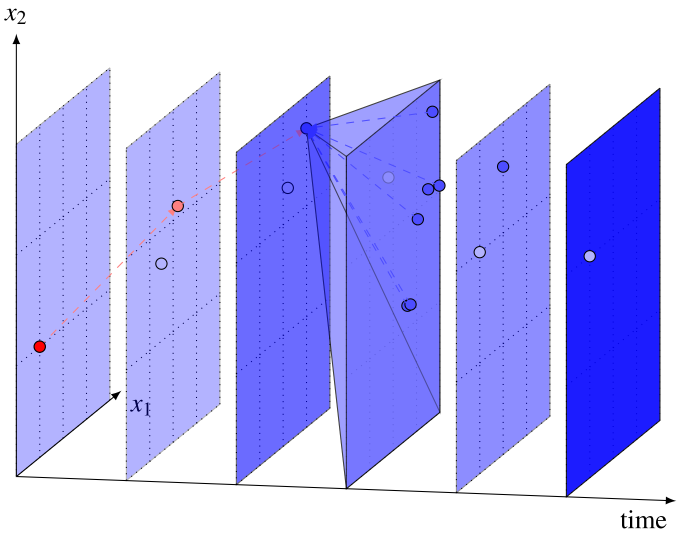

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



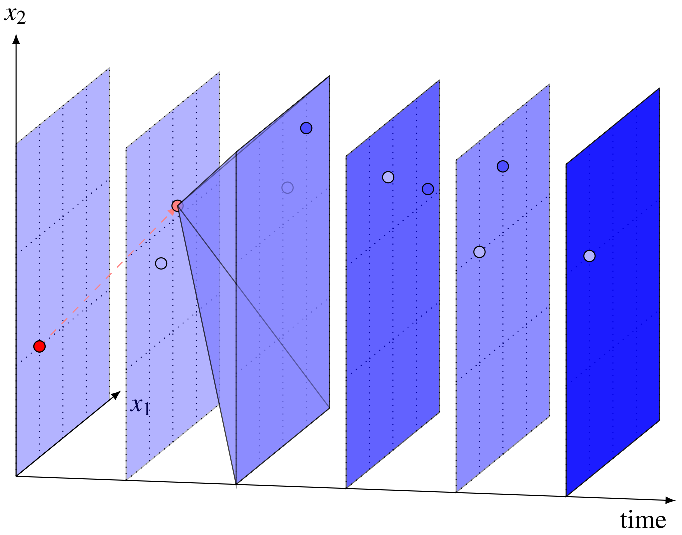

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



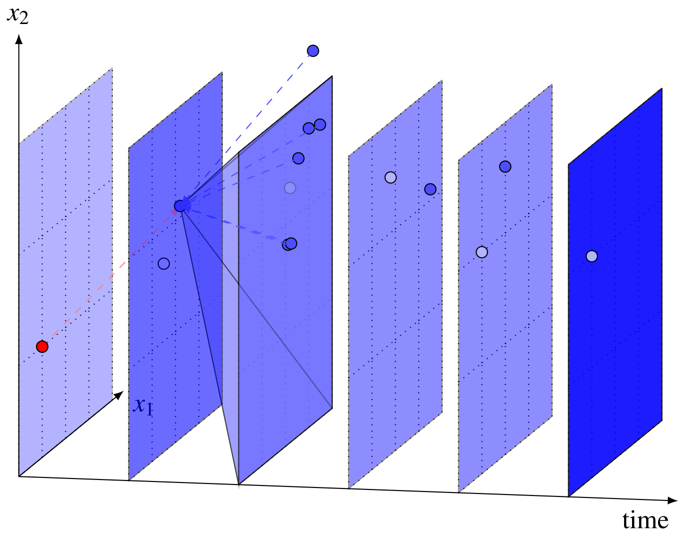

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



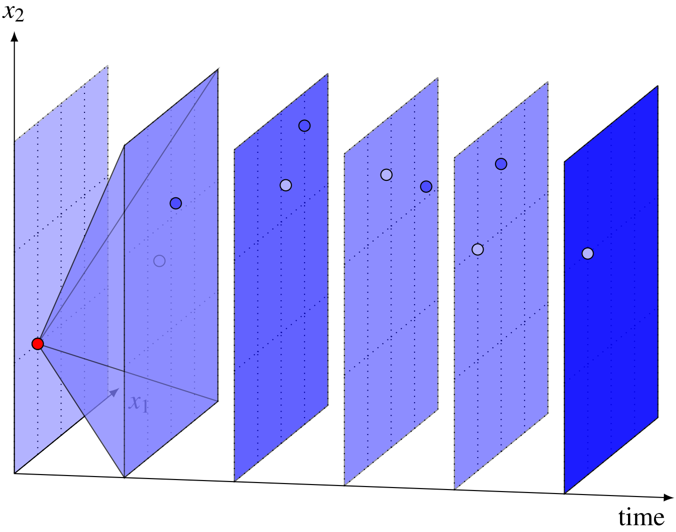

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



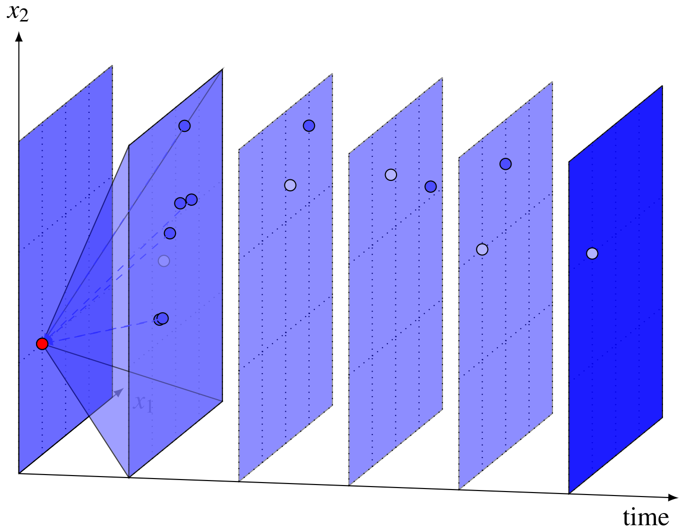

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



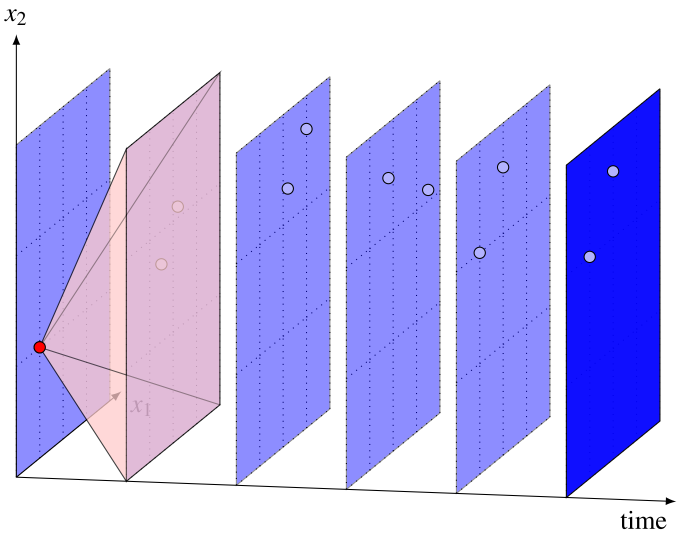

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



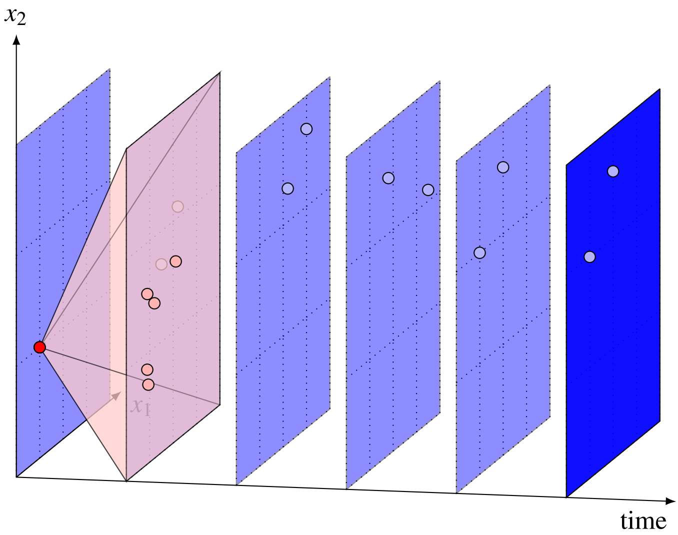

second backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



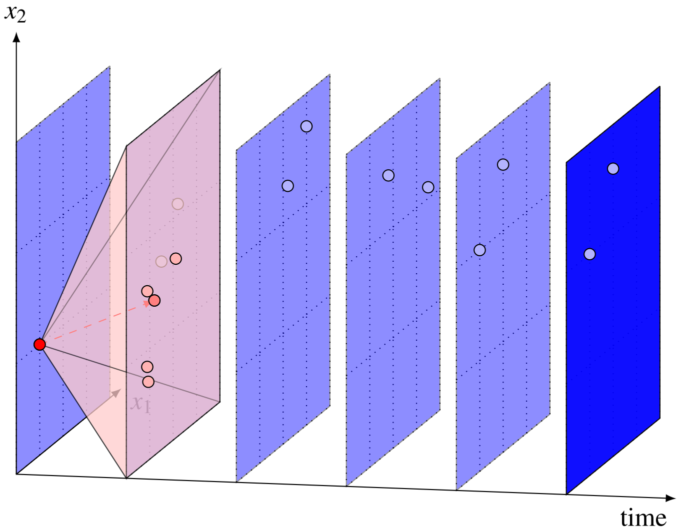

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



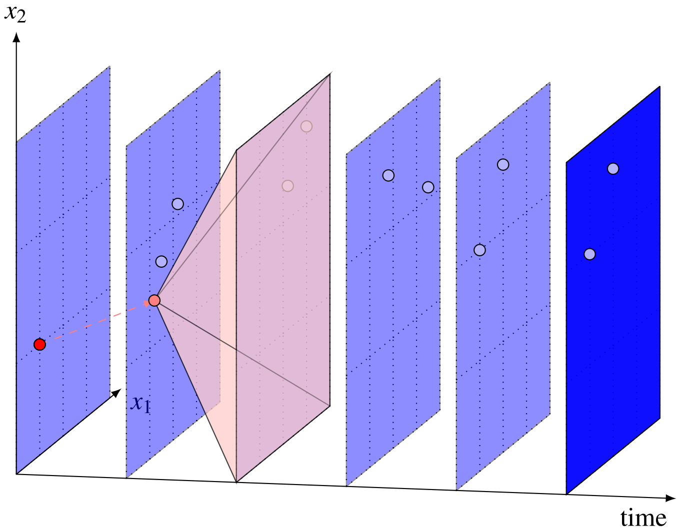

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



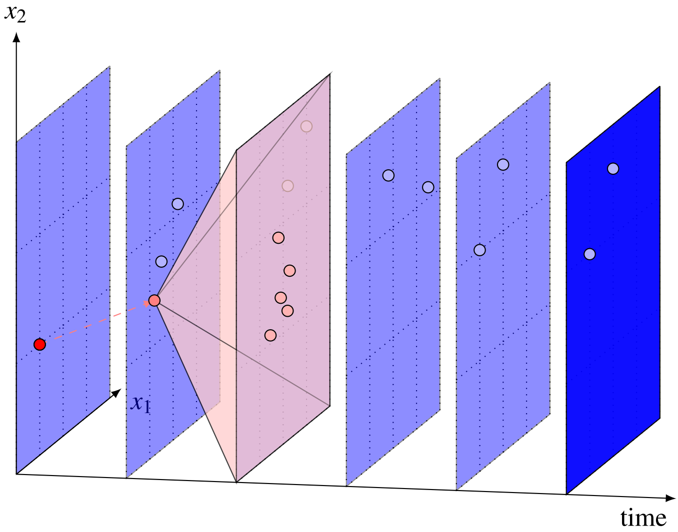

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



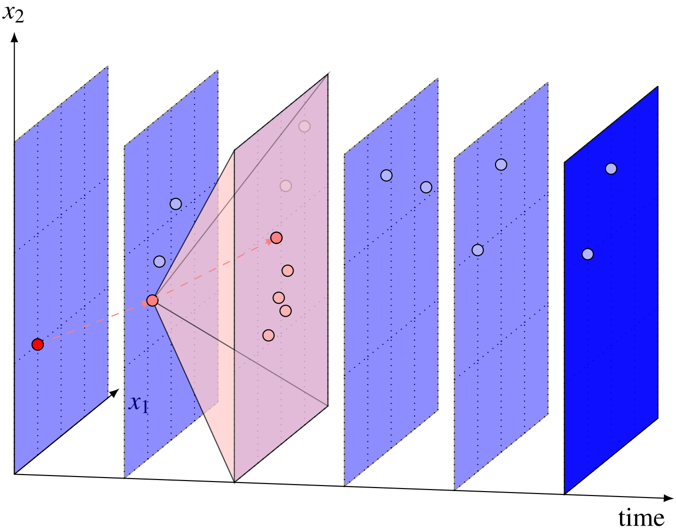

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



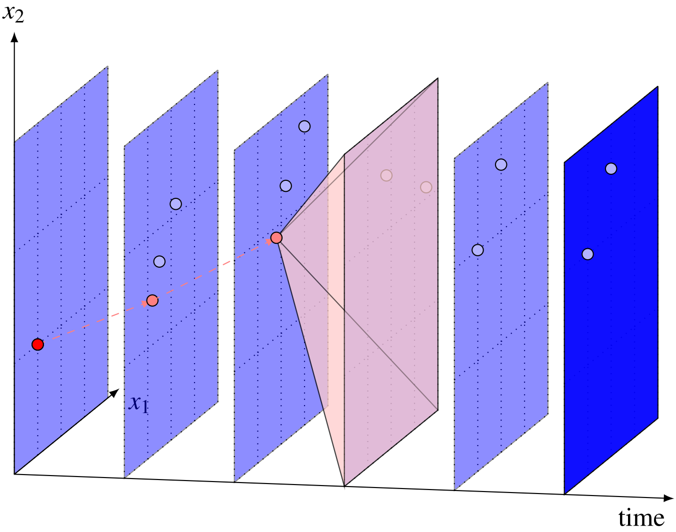

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



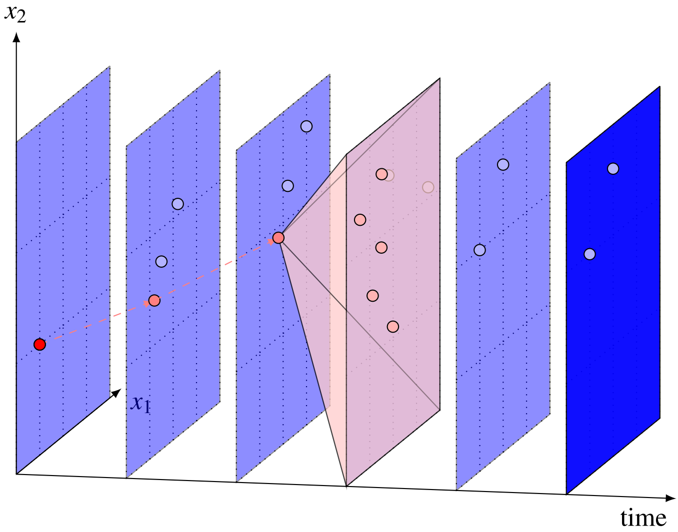

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



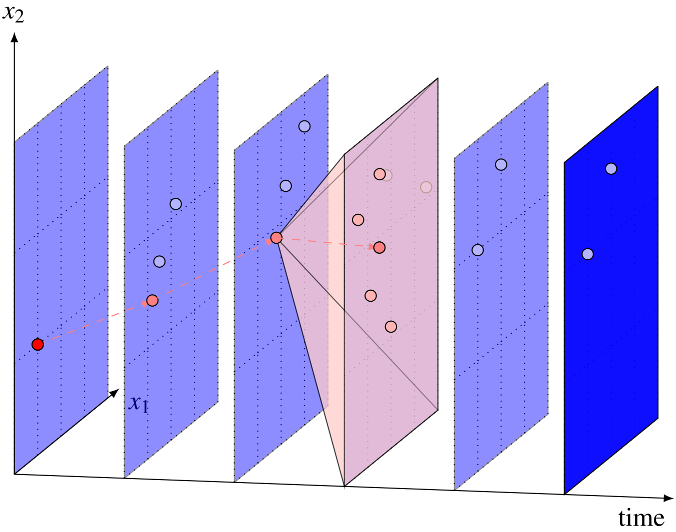

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



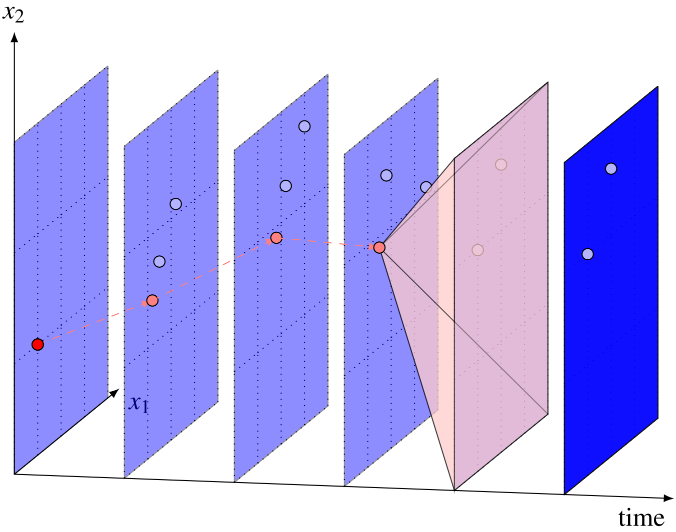

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



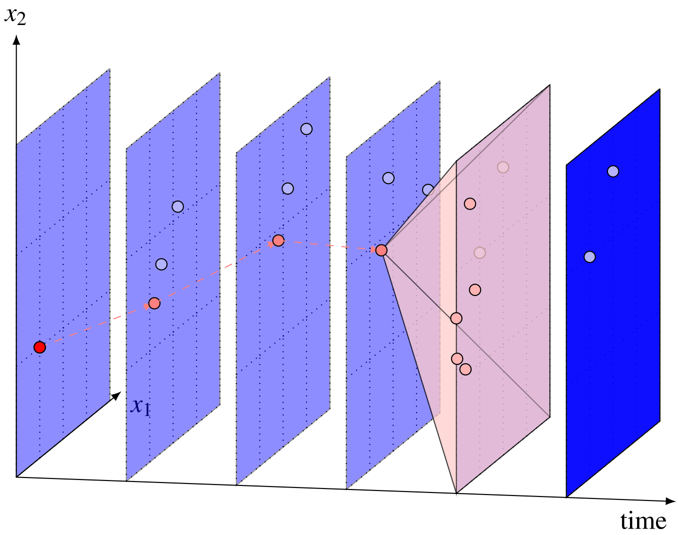

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



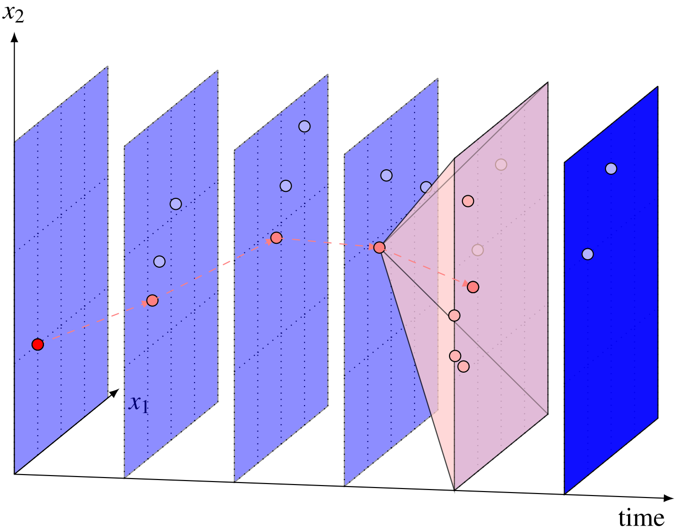

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



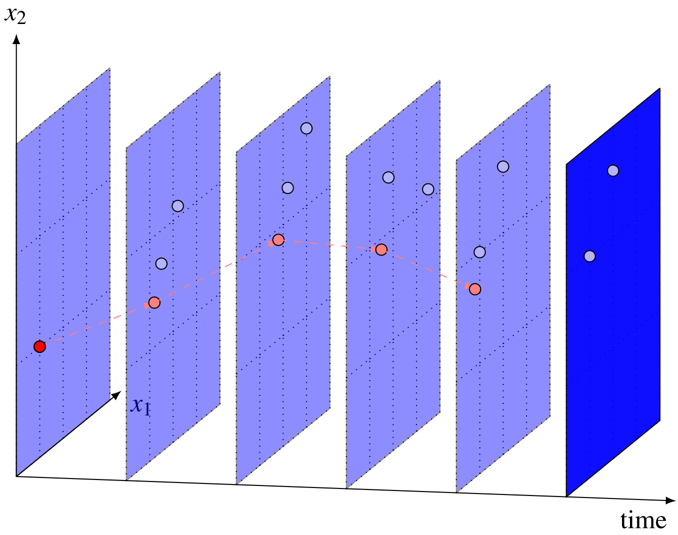

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



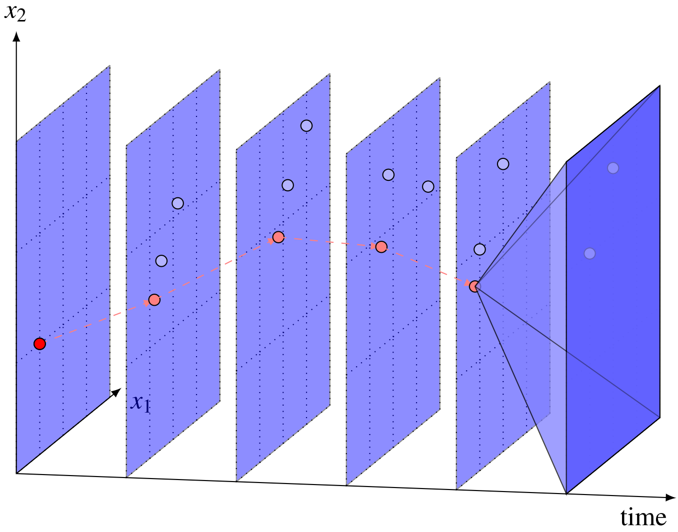

third forward pass : computing trajectory

# Trajectory Following Dynamic Programming



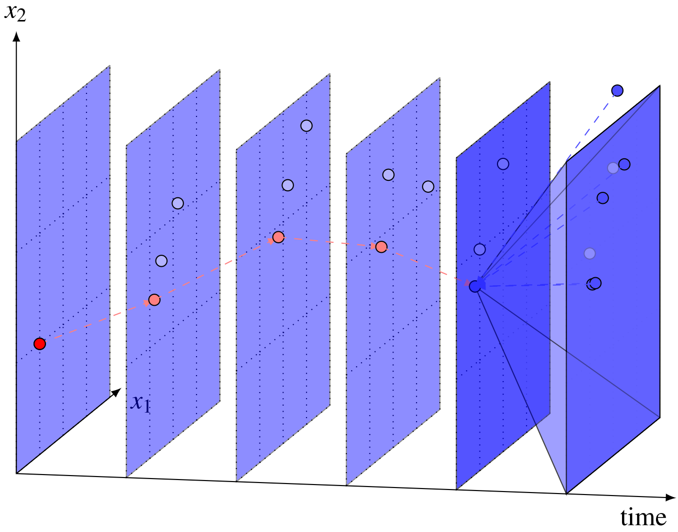

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



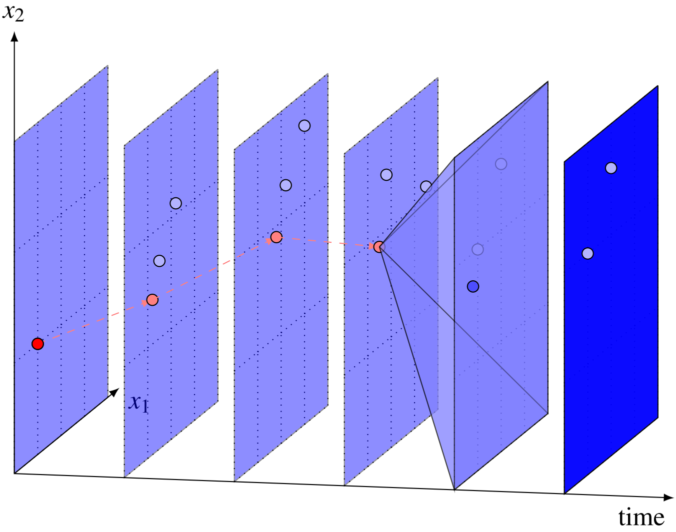

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



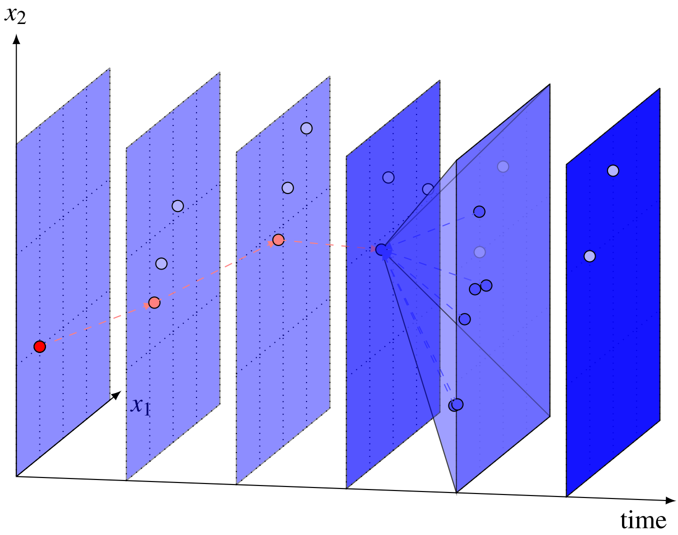

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



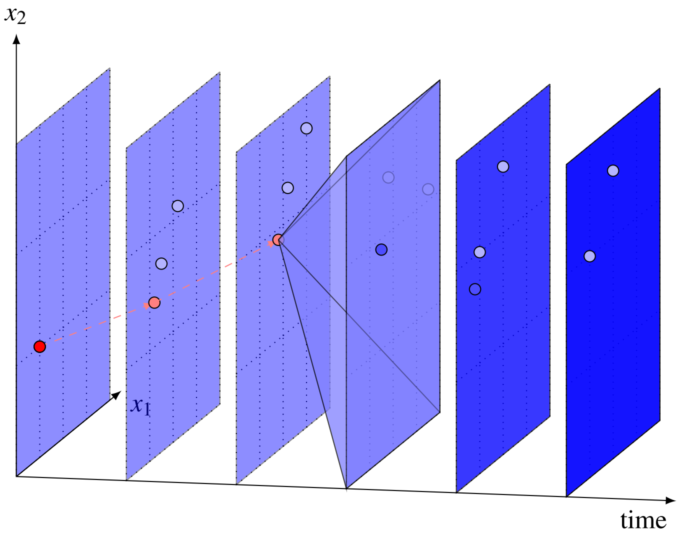

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



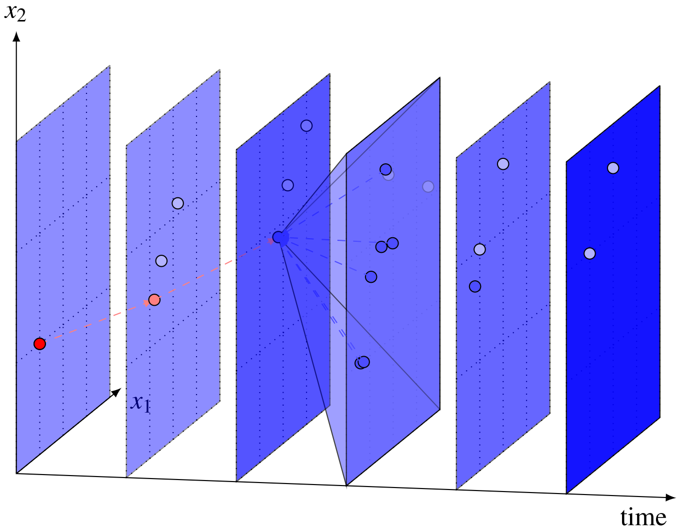

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



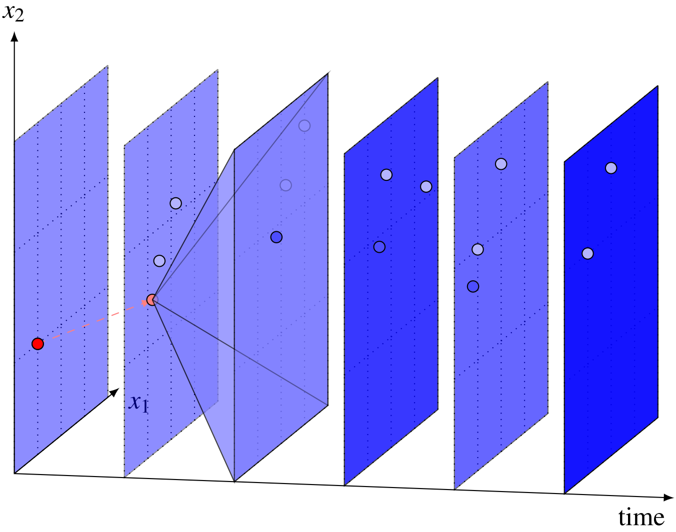

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



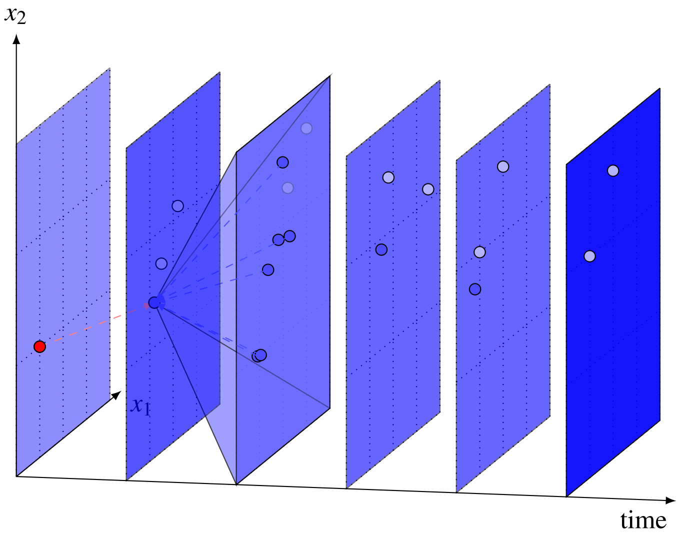

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



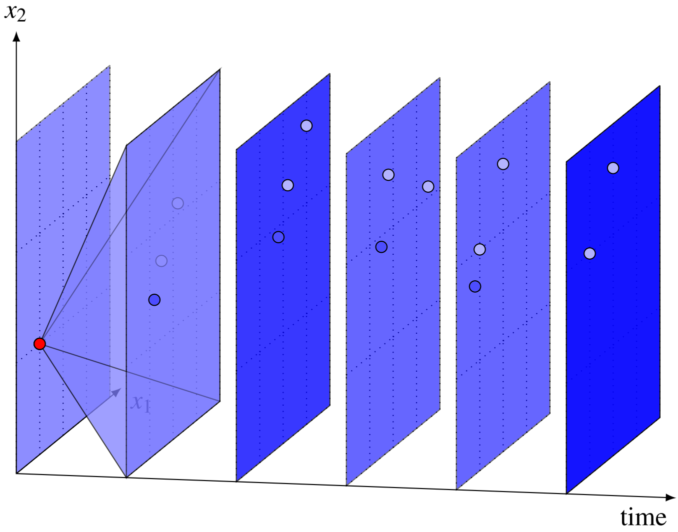

third backward pass : refining approximation (adding cuts)

# Trajectory Following Dynamic Programming



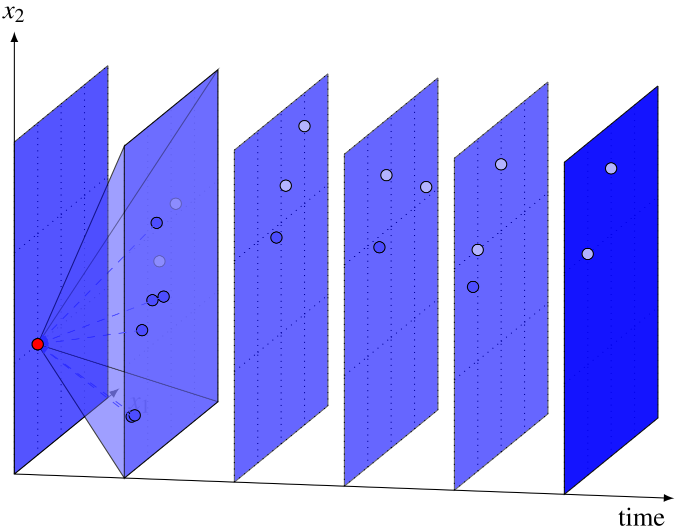

third backward pass : refining approximation (adding cuts)

# 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 $\xi_t^k$ | $\mathcal{F}_t$ | $\underline{V}_t^k$ | $\overline{V}_t^k$ | 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

# 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 | × | ✓ | ✓ |
| 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).

# 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 ?