Tropical Dynamic Programming and Entropic Regularization of the Nested Distance

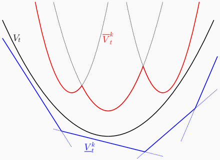
(Duy-Nghi) Benoît Tran

October 07th 2021

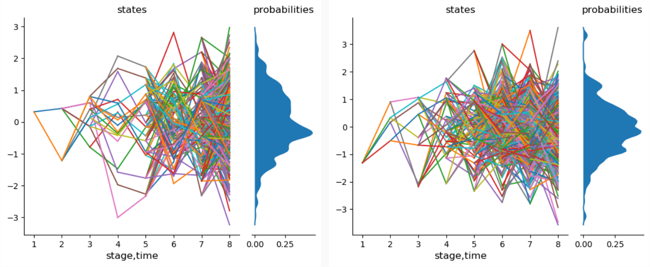
FGV EMAp

Outline

1. **Tropical Dynamic Programming**
with M. Akian (Ecole Polytechnique) and
J-P. Chancelier (Ecole des Ponts ParisTech)



2. **Entropic Regularization of the Nested Distance**
with Z. Qu (Hong Kong University)



Part content

Multistage Stochastic optimization Problems (MSP)

Multistage Stochastic optimization Problem

$$\min_{(\mathsf{X},\mathsf{U})} \mathbb{E}\left[\sum_{t=0}^{T-1} c_t^{\mathsf{W}_{t+1}}\left(\mathsf{X}_t, \mathsf{U}_t\right) + \psi\left(\mathsf{X}_T\right)\right]$$

$$\text{s.t. } \mathsf{X}_0 = x_0 \text{ given}, \forall t \in [\![0, T-1]\!]$$

$$\mathsf{X}_{t+1} = f_t^{\mathsf{W}_{t+1}}\left(\mathsf{X}_t, \mathsf{U}_t\right)$$

$$\sigma\left(\mathsf{U}_t\right) \subset \sigma\left(\mathsf{X}_0, \mathsf{W}_1, \ldots, \mathsf{W}_{t+1}\right) \quad \text{(Hazard-Decision)}$$

Assumption (Finite support independent noises)

The sequence $(\mathsf{W}_t)_{t\in[\![1,T]\!]}$ is made of independent random variables each with finite support

Bellman operators and Dynamic Programming

MSP can be solved by Dynamic Programming

Bellman operators and Dynamic Programming

MSP can be solved by Dynamic Programming

- Pointwise Bellman operator

  for all $w \in \mathrm{supp}\,(\mathsf{W_{t+1}})$ and $\phi : \mathbb{X} \to \overline{\mathbb{R}}$

$$\mathcal{B}_t^w\,(\phi) : x \in \mathbb{X} \mapsto \min_u \Big(c_t^w\,(x, u) + \phi\big(f_t^w\,(x, u)\,\big)\Big) \in \overline{\mathbb{R}}$$

Bellman operators and Dynamic Programming

MSP can be solved by Dynamic Programming

- Pointwise Bellman operator
  for all $w \in \mathrm{supp}\,(\mathsf{W_{t+1}})$ and $\phi : \mathbb{X} \to \overline{\mathbb{R}}$
  $$\mathcal{B}_t^w(\phi) : x \in \mathbb{X} \mapsto \min_u \Big(c_t^w(x,u) + \phi\big(f_t^w(x,u)\big)\Big) \in \overline{\mathbb{R}}$$
- (Average) Bellman operator
  $$\mathfrak{B}_t(\phi) : x \in \mathbb{X} \mapsto \mathbb{E}_{W_{t+1}}\Big[\mathcal{B}_t^{\mathsf{W_{t+1}}}(\phi)(x)\Big] \in \overline{\mathbb{R}}$$

MSP can be solved by Dynamic Programming

- Pointwise Bellman operator
  for all $w \in \operatorname{supp}(\mathbf{W}_{t+1})$ and $\phi : \mathbb{X} \to \overline{\mathbb{R}}$
  $$\mathcal{B}_t^w(\phi) : x \in \mathbb{X} \mapsto \min_u \Big(c_t^w(x,u) + \phi\big(f_t^w(x,u)\big)\Big) \in \overline{\mathbb{R}}$$
- (Average) Bellman operator
  $$\mathfrak{B}_t(\phi) : x \in \mathbb{X} \mapsto \mathbb{E}_{W_{t+1}}\Big[\mathcal{B}_t^{\mathbf{W}_{t+1}}(\phi)(x)\Big] \in \overline{\mathbb{R}}$$
- Dynamic Programming Equations
  $$V_T = \psi \quad \text{and} \quad \forall t \in [\![0, T-1]\!],\ V_t = \mathfrak{B}_t(V_{t+1})$$

Bellman operators and Dynamic Programming

MSP can be solved by Dynamic Programming

- Pointwise Bellman operator

  for all $w \in \mathrm{supp}\left(\mathsf{W}_{\mathsf{t+1}}\right)$ and $\phi : \mathbb{X} \to \overline{\mathbb{R}}$

$$\mathcal{B}_t^w\left(\phi\right) : x \in \mathbb{X} \mapsto \min_u \Big(c_t^w\left(x,u\right) + \phi\big(f_t^w\left(x,u\right)\big)\Big) \in \overline{\mathbb{R}}$$

- (Average) Bellman operator

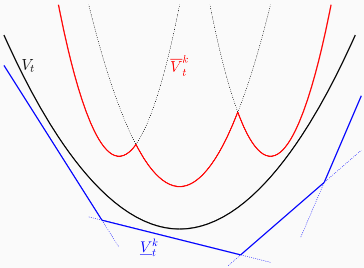
$$\mathfrak{B}_t\left(\phi\right) : x \in \mathbb{X} \mapsto \mathbb{E}_{W_{t+1}}\left[\mathcal{B}_t^{\mathsf{W}_{\mathsf{t+1}}}\left(\phi\right)\left(x\right)\right] \in \overline{\mathbb{R}}$$

- Dynamic Programming Equations

$$V_T = \psi \ \text{ and } \ \forall t \in [\![0, T-1]\!],\ V_t = \mathfrak{B}_t\left(V_{t+1}\right)$$

  - V_t is called the value function at time $t \in [\![0, T]\!]$
- The value of MSP is equal to $V_0\left(x_0\right)$

Build an algorithm that simultaneously generates upper and lower approximations of V_t as min-plus linear and max-plus linear combinations of basic functions



For all $t \in [\![0, T]\!]$, construct increasing sequences of basic functions $\left(\underline{F}_t^k\right)_{k\in\mathbb{N}}$ and $\left(\overline{F}_t^k\right)_{k\in\mathbb{N}}$

$$\begin{cases} \underline{V}_t^k = \sup_{\underline{\phi}\in\underline{F}_t^k} \underline{\phi} \\ \overline{V}_t^k = \inf_{\overline{\phi}\in\overline{F}_t^k} \overline{\phi} \end{cases}$$

Build an algorithm that simultaneously generates upper and lower approximations of V_t as min-plus linear and max-plus linear combinations of basic functions

- Generalizes the Min-plus algorithm for deterministic control problems (McEneaney 2007, Qu 2014) giving upper approximations as infima of quadratics
- and the Stochastic Dual Dynamic Programming (SDDP) algorithm (Pereira and Pinto 1991, Shapiro 2011, ...) giving lower approximations as suprema of affine cuts

Assumption (Lipschitz dynamic, costs and constraints)

For every time $t < T$ and $w \in \mathrm{supp}\,(\mathbf{W}_{t+1})$,

- *dynamics f_t^w are Lipschitz continuous*
- *cost c_t^w are Lipschitz continuous on $\mathrm{dom}\, c_t^w$*
- *constraint set-valued mapping $\mathcal{U}_t^w$ is Lipschitz continuous on X_t,*

$$d_{\mathcal{H}}\left(\mathcal{U}_t^w\left(x_1\right), \mathcal{U}_t^w\left(x_2\right)\right) \leq L_{\mathcal{U}_t^w} \|x_1 - x_2\|$$

Proposition (Lipschitz MSP implies regularity of $\mathfrak{B}_t$)

If $V : \mathbb{X} \to \overline{\mathbb{R}}$ L_{t+1}-Lipschitz on X_{t+1},
then $\mathfrak{B}_t\left(V\right)$ is L_t-Lipschitz on X_t for some constant $L_t > 0$ which only depends on the data of the MSP problem and L_{t+1}.

Constraint set-valued mapping

For each noise $w \in \mathrm{supp}(\mathsf{W}_{t+1})$, $t \in [\![0, T-1]\!]$, define the constraint set-valued mapping $\mathcal{U}_t^w : \mathbb{X} \rightrightarrows \mathbb{U}$

$$\mathcal{U}_t^w(x) := \{u \in \mathbb{U} \mid c_t^w(x,u) < +\infty \text{ and } f_t^w(x,u) \in X_{t+1}\}\,.^1$$

Assumption (Recourse assumption)

The set-valued mapping $\mathcal{U}_t^w$ is non-empty compact valued

Proposition (Known domains of V_t)

Under the recourse assumption, $\mathrm{dom}\, V_t = X_t$

[1] $\forall w \in \mathrm{supp}(\mathsf{W}_{t+1})\,, X_t^w := \pi_{\mathbb{X}}\left(\mathrm{dom}\; c_t^w\right)$, and $X_t := \cap_{w \in \mathrm{supp}(\mathsf{W}_{t+1})} X_t^w$.

How to select new basic functions ?

Input: sequence $(x_t)_{t\in[\![0,T]\!]}$ of trial points, sequence $(F_t)_{t\in[\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t\in[\![0,T]\!]}$ of basic functions

How to select new basic functions ?

Input: sequence $(x_t)_{t\in[\![0,T]\!]}$ of trial points, sequence $(F_t)_{t\in[\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t\in[\![0,T]\!]}$ of basic functions

<u>Case $t = T$</u>

Tightness Assumption (local property)

$$\phi_T(x_T) = V_T(x_T)$$

Validity Assumption (global property)

$$\phi_T \geq V_T \quad \text{(Min-plus lin. combinations case)}$$

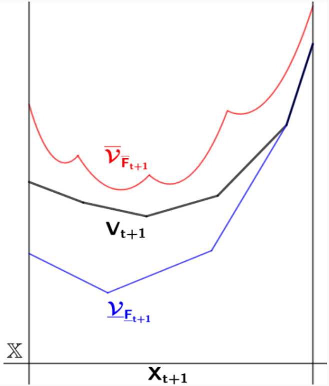
$$\phi_T \leq V_T \quad \text{(Max-plus lin. combinations case)}$$

How to select new basic functions ?

Input: sequence $(x_t)_{t\in[\![0,T]\!]}$ of trial points, sequence $(F_t)_{t\in[\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t\in[\![0,T]\!]}$ of basic functions

Notation: $\mathcal{V}_{F_{t+1}}$ the sup or inf of basic functions in F_{t+1}

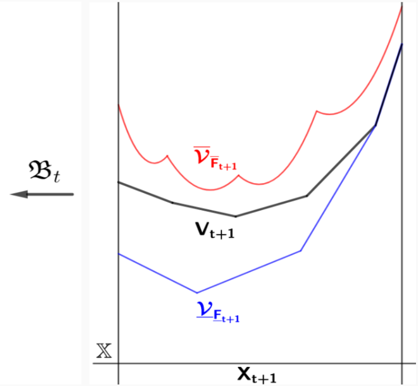


How to select new basic functions ?

Input: sequence $(x_t)_{t\in[\![0,T]\!]}$ of trial points, sequence $(F_t)_{t\in[\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t\in[\![0,T]\!]}$ of basic functions

Notation: $\mathcal{V}_{F_{t+1}}$ the sup or inf of basic functions in F_{t+1}

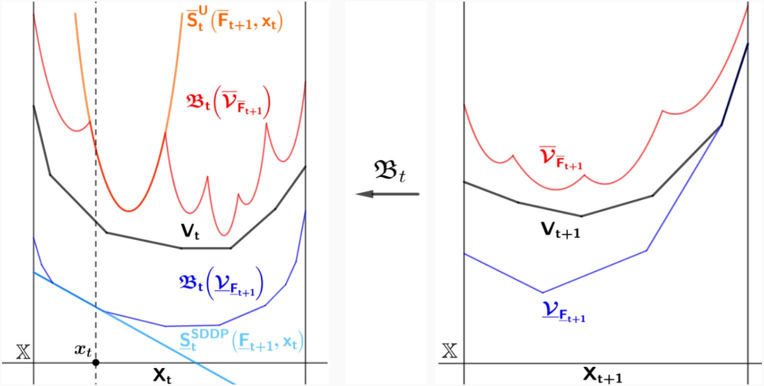


How to select new basic functions ?

Input: sequence $(x_t)_{t \in [\![0,T]\!]}$ of trial points, sequence $(F_t)_{t \in [\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t \in [\![0,T]\!]}$ of basic functions

Notation: $\mathcal{V}_{F_{t+1}}$ the sup or inf of basic functions in F_{t+1}



How to select new basic functions ?

Input: sequence $(x_t)_{t \in [\![0,T]\!]}$ of trial points, sequence $(F_t)_{t \in [\![0,T]\!]}$ of sets of basic functions

Output: sequence $(\phi_t)_{t \in [\![0,T]\!]}$ of basic functions

<u>Case $t < T$</u>

Tightness Assumption (local property)

$$\phi_t \left(x_t\right) = \mathfrak{B}_t \left(\mathcal{V}_{F_{t+1}}\right) \left(x_t\right)$$

Validity Assumption (global property)

$$\phi_t \geq \mathfrak{B}_t \left(\mathcal{V}_{F_{t+1}}\right) \quad \text{(Min-plus lin. combinations case)}$$

$$\phi_t \leq \mathfrak{B}_t \left(\mathcal{V}_{F_{t+1}}\right) \quad \text{(Max-plus lin. combinations case)}$$

Problem-child trajectory of Baucke and al. (2018)

Input: two sequences of functions $\underline{V}_0, \ldots, \underline{V}_T$ and $\overline{V}_0, \ldots, \overline{V}_T$

Output: Problem-child trajectory, states $(x_0^*, \ldots, x_T^*)$.

Initial state x_0^* is given, then **for** $t < T$

Problem-child trajectory of Baucke and al. (2018)

Input: two sequences of functions $\underline{V}_0, \dots, \underline{V}_T$ and $\overline{V}_0, \dots, \overline{V}_T$

Output: Problem-child trajectory, states $(x_0^*, \dots, x_T^*)$.

Initial state x_0^* is given, then **for** $t < T$

1. For all $w \in \mathrm{supp}(\mathsf{W}_{t+1})$, compute optimal control at x_t^*

$$u_t^w \in \operatorname*{arg\,min}_{u \in U} \Big(c_t^w(x_t^*, u) + \underline{V}_{t+1}\big(f_t^w(x_t^*, u)\big) \Big)$$

Problem-child trajectory of Baucke and al. (2018)

Input: two sequences of functions $\underline{V}_0, \dots, \underline{V}_T$ and $\overline{V}_0, \dots, \overline{V}_T$

Output: Problem-child trajectory, states $(x_0^*, \dots, x_T^*)$.

Initial state x_0^* is given, then **for** $t < T$

1. For all $w \in \operatorname{supp}(\mathsf{W}_{t+1})$, compute optimal control at x_t^*
$$u_t^w \in \operatorname*{arg\,min}_{u \in U} \Big(c_t^w(x_t^*, u) + \underline{V}_{t+1}\big(f_t^w(x_t^*, u)\big)\Big)$$
2. Compute "the worst" noise
$w^* \in \operatorname{arg\,max}_{w \in \mathsf{W}_{t+1}} \big(\overline{V}_{t+1} - \underline{V}_{t+1}\big)\big(f_t^w(x_t^*, u_t^w)\big)$

Problem-child trajectory of Baucke and al. (2018)

Input: two sequences of functions $\underline{V}_0, \dots, \underline{V}_T$ and $\overline{V}_0, \dots, \overline{V}_T$

Output: Problem-child trajectory, states $(x_0^*, \dots, x_T^*)$.

Initial state x_0^* is given, then **for** $t < T$

1. For all $w \in \operatorname{supp}(\mathbf{W}_{t+1})$, compute optimal control at x_t^*
$$u_t^w \in \operatorname*{arg\,min}_{u \in U} \Big(c_t^w(x_t^*, u) + \underline{V}_{t+1}\big(f_t^w(x_t^*, u)\big) \Big)$$
2. Compute "the worst" noise
$w^* \in \operatorname{arg\,max}_{w \in \mathbf{W}_{t+1}} \big(\overline{V}_{t+1} - \underline{V}_{t+1}\big)\big(f_t^w(x_t^*, u_t^w)\big)$
3. Set $x_{t+1}^* = f_t^{w^*}\left(x_t^*, u_t^{w^*}\right)$

Problem-child trajectory of Baucke and al. (2018)

Input: two sequences of functions $\underline{V}_0, \dots, \underline{V}_T$ and $\overline{V}_0, \dots, \overline{V}_T$

Output: Problem-child trajectory, states $(x_0^*, \dots, x_T^*)$.

Initial state x_0^* is given, then **for** $t < T$

1. For all $w \in \mathrm{supp}(\mathsf{W}_{t+1})$, compute optimal control at x_t^*

$$u_t^w \in \operatorname*{arg\,min}_{u \in U} \Big(c_t^w(x_t^*, u) + \underline{V}_{t+1}\big(f_t^w(x_t^*, u)\big) \Big)$$

2. Compute "the worst" noise
   $w^* \in \arg\max_{w \in \mathsf{W}_{t+1}} \big(\overline{V}_{t+1} - \underline{V}_{t+1}\big)\big(f_t^w(x_t^*, u_t^w)\big)$
3. Set $x_{t+1}^* = f_t^{w^*}\left(x_t^*, u_t^{w^*}\right)$

Interpretation

Problem child trajectory = "Worst" optimal trajectory of the lower approximations

Tropical Dynamic Programming (TDP) algorithm

Algorithm 1 Tropical Dynamic Programming (TDP)

Input: Selection functions and $(\mathsf{W_t})_{t \in [\![1,T]\!]}$ independent r.v. with finite support.

Output: Sequence of sets $\left(\overline{F}_t^k\right)_{k\in\mathbb{N}}$, $\left(\underline{F}_t^k\right)_{k\in\mathbb{N}}$

Tropical Dynamic Programming (TDP) algorithm

Algorithm 2 Tropical Dynamic Programming (TDP)

Input: Selection functions and $(\mathbf{W_t})_{t \in [\![1,T]\!]}$ independent r.v. with finite support.

Output: Sequence of sets $\left(\overline{F}_t^k\right)_{k \in \mathbb{N}}$, $\left(\underline{F}_t^k\right)_{k \in \mathbb{N}}$

1: For every $t \in [\![0, T]\!]$, $\overline{F}_t^0 := \emptyset$ and $\underline{F}_t^0 := \emptyset$

Tropical Dynamic Programming (TDP) algorithm

Algorithm 3 Tropical Dynamic Programming (TDP)

Input: Selection functions and $(\mathrm{W_t})_{t\in[\![1,T]\!]}$ independent r.v. with finite support.

Output: Sequence of sets $\left(\overline{F}_t^k\right)_{k\in\mathbb{N}}$, $\left(\underline{F}_t^k\right)_{k\in\mathbb{N}}$

1: For every $t \in [\![0, T]\!]$, $\overline{F}_t^0 := \emptyset$ and $\underline{F}_t^0 := \emptyset$

2: **for** $k \geq 0$ **do**

3: Forward. Compute Problem-child trajectory $\left(x_t^k\right)_{t\in[\![0,T]\!]}$ using $\overline{V}_t^k = \inf_{\overline{\phi}\in\overline{F}_t^k} \overline{\phi}$ and $\underline{V}_t^k = \sup_{\underline{\phi}\in\underline{F}_t^k} \underline{\phi}$

5: **end for**

Algorithm 4 Tropical Dynamic Programming (TDP)

Input: Selection functions and $(\mathrm{W_t})_{t\in[\![1,T]\!]}$ independent r.v. with finite support.

Output: Sequence of sets $\left(\overline{F}_t^k\right)_{k\in\mathbb{N}}$, $\left(\underline{F}_t^k\right)_{k\in\mathbb{N}}$

1: For every $t \in [\![0,T]\!]$, $\overline{F}_t^0 := \emptyset$ and $\underline{F}_t^0 := \emptyset$

2: **for** $k \geq 0$ **do**

3: Forward. Compute Problem-child trajectory $\left(x_t^k\right)_{t\in[\![0,T]\!]}$ using $\overline{V}_t^k = \inf_{\overline{\phi}\in\overline{F}_t^k} \overline{\phi}$ and $\underline{V}_t^k = \sup_{\underline{\phi}\in\underline{F}_t^k} \underline{\phi}$

4: Backward. Compute new basic functions $\left(\overline{\phi}_t\right)_{t\in[\![0,T]\!]}$ and $\left(\underline{\phi}_t\right)_{t\in[\![0,T]\!]}$ and update
$\overline{F}_t^{k+1} := \overline{F}_t^k \cup \left\{\overline{\phi}_t\right\}$ and $\underline{F}_t^{k+1} := \underline{F}_t^k \cup \left\{\underline{\phi}_t\right\}$, $t \in [\![0,T]\!]$

5: **end for**

Convergence to limits $\underline{V}_t^*$ and $\overline{V}_t^*$

Under finite independent noises, Lipschitz data and recourse assumptions we have

Existence of an approximating limit

The sequence of functions $\left(\underline{V}_t^k\right)_{k\in\mathbb{N}}$ (resp. $\left(\overline{V}_t^k\right)_{k\in\mathbb{N}}$) generated by TDP converges uniformly on every compact set included in the domain of V_t to a function $\underline{V}_t^*$ (resp. $\overline{V}_t^*$).

Some features of TDP

Convergence to limits $\underline{V}_t^*$ and $\overline{V}_t^*$

Under finite independent noises, Lipschitz data and recourse assumptions we have

Existence of an approximating limit

The sequence of functions $\left(\underline{V}_t^k\right)_{k\in\mathbb{N}}$ (resp. $\left(\overline{V}_t^k\right)_{k\in\mathbb{N}}$) generated by TDP converges uniformly on every compact set included in the domain of V_t to a function $\underline{V}_t^*$ (resp. $\overline{V}_t^*$).

Some features of TDP

- No need to discretize the state space

Convergence to limits $\underline{V}_t^*$ and $\overline{V}_t^*$

Under finite independent noises, Lipschitz data and recourse assumptions we have

Existence of an approximating limit

The sequence of functions $\left(\underline{V}_t^k\right)_{k\in\mathbb{N}}$ (resp. $\left(\overline{V}_t^k\right)_{k\in\mathbb{N}}$) generated by TDP converges uniformly on every compact set included in the domain of V_t to a function $\underline{V}_t^*$ (resp. $\overline{V}_t^*$).

Some features of TDP

- No need to discretize the state space
- $\left(\underline{V}_t^k\right)_k$ and $\left(\overline{V}_t^k\right)_k$ are monotonic

Convergence to limits $\underline{V}_t^*$ and $\overline{V}_t^*$

Under finite independent noises, Lipschitz data and recourse assumptions we have

Existence of an approximating limit

The sequence of functions $\left(\underline{V}_t^k\right)_{k\in\mathbb{N}}$ (resp. $\left(\overline{V}_t^k\right)_{k\in\mathbb{N}}$) generated by TDP converges uniformly on every compact set included in the domain of V_t to a function $\underline{V}_t^*$ (resp. $\overline{V}_t^*$).

Some features of TDP

- No need to discretize the state space
- $\left(\underline{V}_t^k\right)_k$ and $\left(\overline{V}_t^k\right)_k$ are monotonic
- $\underline{V}_t^*$ and $\overline{V}_t^*$ are close to V_t on "interesting points", but may be far from V_t elsewhere.

Asymptotic convergence of TDP

Under finite independent noises, Lipschitz data and recourse assumptions we have

Convergence of TDP [Akian, Chancelier, T., 2020]

Denote by $\left(x_t^k\right)_{0 \leq t \leq T}$ the k-th Problem-child trajectory.

For every accumulation point x_t^* of $\left(x_t^k\right)_{k \in \mathbb{N}}$, we have

$$\overline{V}_t^k\left(x_t^k\right) - \underline{V}_t^k\left(x_t^k\right) \underset{k \to +\infty}{\longrightarrow} 0 \quad \text{and} \quad \overline{V}_t^*\left(x_t^*\right) = V_t\left(x_t^*\right) = \underline{V}_t^*\left(x_t^*\right)$$

This result generalizes the convergence of SDDP *à la* [Philpott and al. (2013)] and [Baucke and al. (2018)] seen as a specific instance of TDP for the linear-polyhedral framework

Idea of the proof, details in [Akian, Chancelier, T., 2020]

- $\left(\underline{V}_t^k\right)_k$ (resp. $\left(\overline{V}_t^k\right)_k$) converges uniformly to $\underline{V}_t^*$ (resp. $\overline{V}_t^*$) on the domain of V_t by Arzela-Ascoli theorem

Idea of the proof, details in [Akian, Chancelier, T., 2020]

- $\left(\underline{V}_t^k\right)_k$ (resp. $\left(\overline{V}_t^k\right)_k$) converges uniformly to $\underline{V}_t^*$ (resp. $\overline{V}_t^*$) on the domain of V_t by Arzela-Ascoli theorem
- Exploiting monotonicity of the approximations and that each operator $\mathcal{B}_t^w$ is order preserving

$$
\begin{aligned}
0 &\leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \\
&\leq \sum_{w \in \mathrm{supp}(\mathbf{W}_{t+1})} \mathbb{P}\left[\mathbf{W}_{t+1} = w\right] \left[\left(\overline{V}_{t+1}^k - \underline{V}_{t+1}^k\right)\left(f_t^w\left(x_t^k, u_t^k(w)\right)\right)\right]
\end{aligned}
$$

Idea of the proof, details in [Akian, Chancelier, T., 2020]

- $\left(\underline{V}_t^k\right)_k$ (resp. $\left(\overline{V}_t^k\right)_k$) converges uniformly to $\underline{V}_t^*$ (resp. $\overline{V}_t^*$) on the domain of V_t by Arzela-Ascoli theorem
- Exploiting monotonicity of the approximations and that each operator $\mathcal{B}_t^w$ is order preserving

$$
\begin{aligned}
0 &\leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \\
&\leq \sum_{w \in \mathrm{supp}(\mathbf{W}_{t+1})} \mathbb{P}\left[\mathbf{W}_{t+1} = w\right] \left[\left(\overline{V}_{t+1}^k - \underline{V}_{t+1}^k\right)\left(f_t^w\left(x_t^k, u_t^k(w)\right)\right)\right]
\end{aligned}
$$

- PC-trajectory is the “worst” optimal trajectory

$$
0 \leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \leq \overline{V}_{t+1}^k\left(x_{t+1}^k\right) - \underline{V}_{t+1}^k\left(x_{t+1}^k\right)
$$

Idea of the proof, details in [Akian, Chancelier, T., 2020]

- $\left(\underline{V}_t^k\right)_k$ (resp. $\left(\overline{V}_t^k\right)_k$) converges uniformly to $\underline{V}_t^*$ (resp. $\overline{V}_t^*$) on the domain of V_t by Arzela-Ascoli theorem
- Exploiting monotonicity of the approximations and that each operator $\mathcal{B}_t^w$ is order preserving

$$
\begin{aligned}
0 &\leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \\
&\leq \sum_{w \in \mathrm{supp}(\mathbf{W}_{t+1})} \mathbb{P}\left[\mathbf{W}_{t+1} = w\right]\left[\left(\overline{V}_{t+1}^k - \underline{V}_{t+1}^k\right)\left(f_t^w\left(x_t^k, u_t^k(w)\right)\right)\right]
\end{aligned}
$$

- PC-trajectory is the “worst” optimal trajectory

$$
0 \leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \leq \overline{V}_{t+1}^k\left(x_{t+1}^k\right) - \underline{V}_{t+1}^k\left(x_{t+1}^k\right)
$$

- Taking the limit in k

$$
0 \leq \overline{V}_t^*\left(x_t^*\right) - \underline{V}_t^*\left(x_t^*\right) \leq \overline{V}_{t+1}^*\left(x_{t+1}^*\right) - \underline{V}_{t+1}^*\left(x_{t+1}^*\right)
$$

Idea of the proof, details in [Akian, Chancelier, T., 2020]

- $\left(\underline{V}_t^k\right)_k$ (resp. $\left(\overline{V}_t^k\right)_k$) converges uniformly to $\underline{V}_t^*$ (resp. $\overline{V}_t^*$) on the domain of V_t by Arzela-Ascoli theorem
- Exploiting monotonicity of the approximations and that each operator $\mathcal{B}_t^w$ is order preserving

$$0 \leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right)$$
$$\leq \sum_{w \in \mathrm{supp}(\mathbf{W}_{t+1})} \mathbb{P}\left[\mathbf{W}_{t+1} = w\right]\left[\left(\overline{V}_{t+1}^k - \underline{V}_{t+1}^k\right)\left(f_t^w\left(x_t^k, u_t^k(w)\right)\right)\right]$$

- PC-trajectory is the “worst” optimal trajectory

$$0 \leq \overline{V}_t^{k+1}\left(x_t^k\right) - \underline{V}_t^{k+1}\left(x_t^k\right) \leq \overline{V}_{t+1}^k\left(x_{t+1}^k\right) - \underline{V}_{t+1}^k\left(x_{t+1}^k\right)$$

- Taking the limit in k

$$0 \leq \overline{V}_t^*\left(x_t^*\right) - \underline{V}_t^*\left(x_t^*\right) \leq \overline{V}_{t+1}^*\left(x_{t+1}^*\right) - \underline{V}_{t+1}^*\left(x_{t+1}^*\right)$$

- Conclude by backward recursion on t

Linear-polyhedral framework

Linear dynamics $(x, u) \mapsto f_t^w(x, u)$

Polyhedral costs $(x, u) \mapsto c_t^w(x, u)$ (convex polyhedral epigraph)

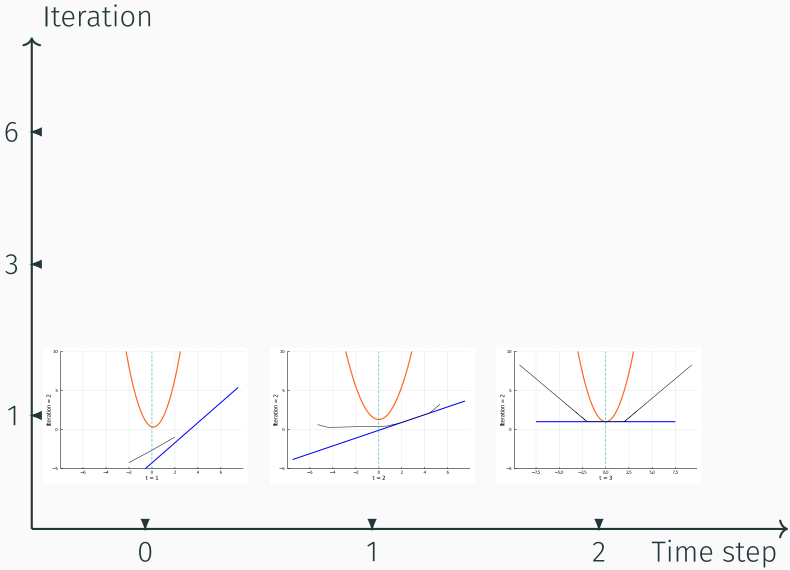
Proposition (Linear-polyhedral MSP are Lipschitz MSP)

Linear-polyhedral MSP are Lipschitz MSP

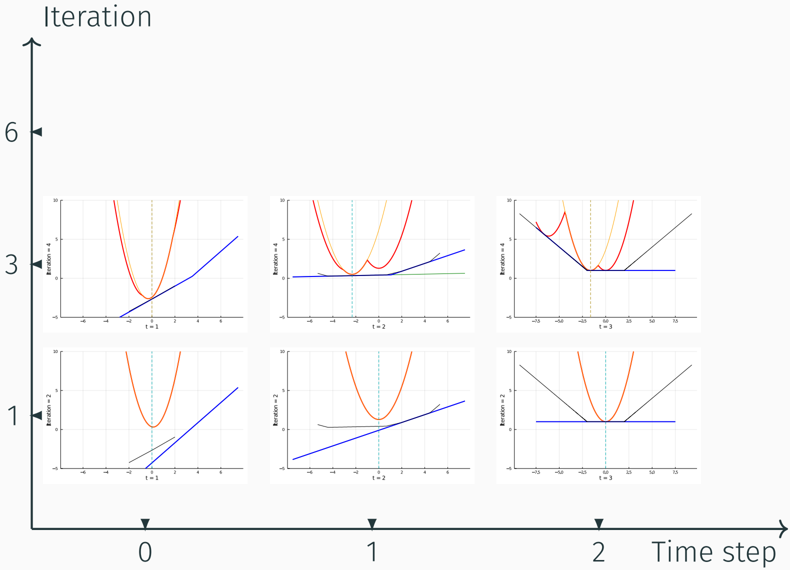
Proof.

The constraint mapping $\mathcal{U}_t^w$ has a convex polyhedral graph thus (*e.g.* [Rockafellar-Wets, Variational Analysis]) is Lipschitz with an explicit constant □

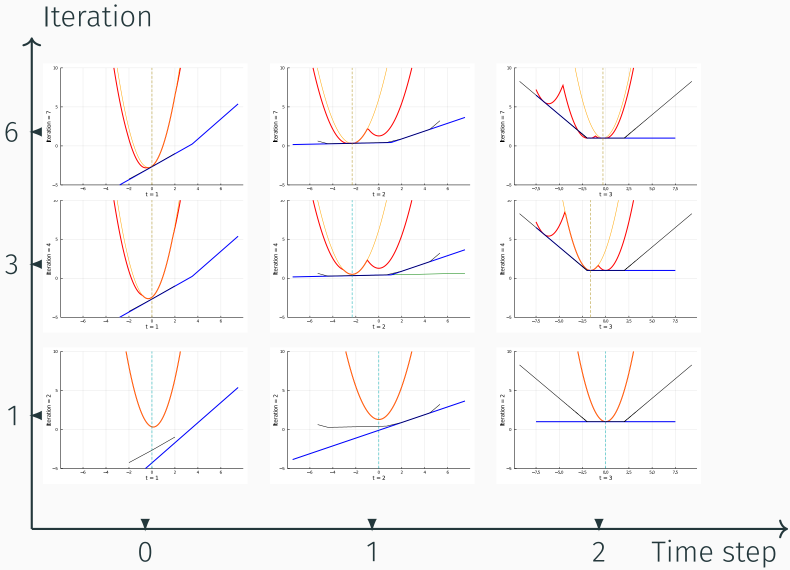
U-SDDP on a linear-polyhedral example



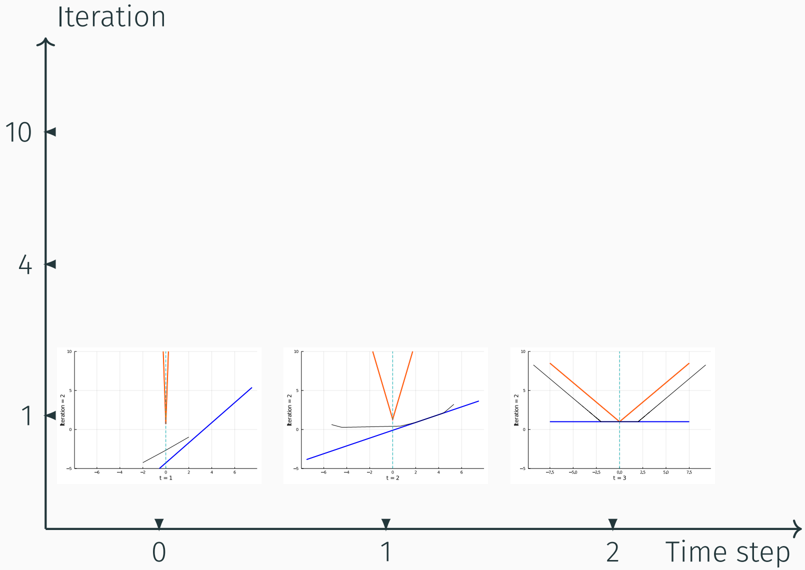
U-SDDP on a linear-polyhedral example



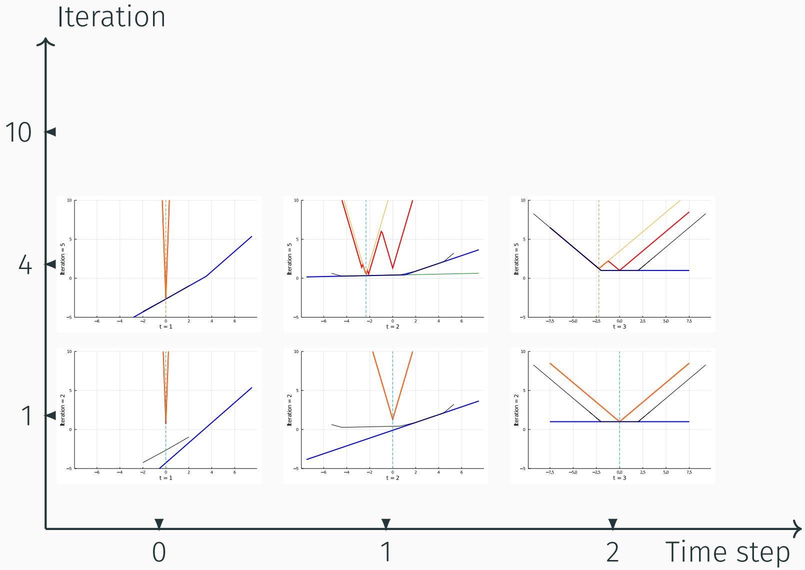
U-SDDP on a linear-polyhedral example



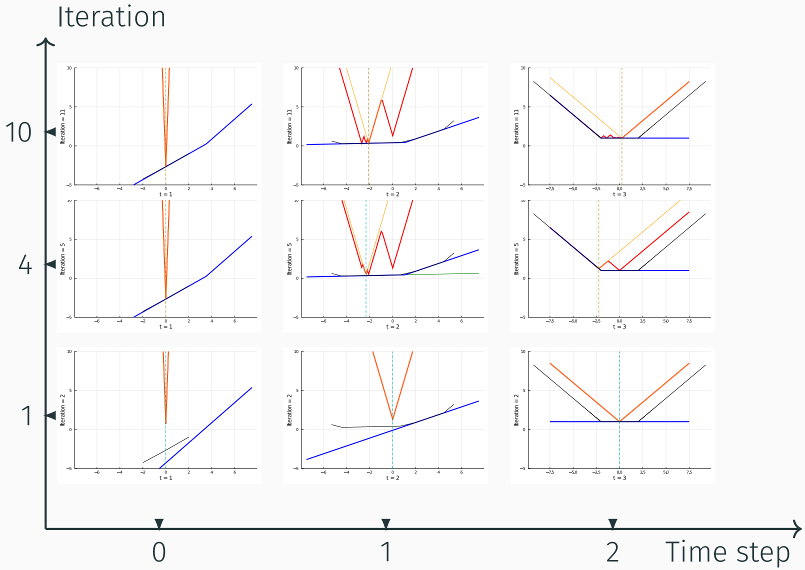
V-SDDP on a linear-polyhedral example



V-SDDP on a linear-polyhedral example



V-SDDP on a linear-polyhedral example



Complexity of TDP

- G. Lan obtained complexity of SDDP (and EDDP) in 2020
  Precision of $T\epsilon$ archived after at most $T(\frac{D}{\epsilon}+1)^N$ iterations
  D diameter state spaces
  N dimension state/control (decision) space

Complexity of TDP

- G. Lan obtained complexity of SDDP (and EDDP) in 2020
  Precision of $T\epsilon$ archived after at most $T(\frac{D}{\epsilon}+1)^N$ iterations
  D diameter state spaces
  N dimension state/control (decision) space
- Straightforward modifications of Lan's proof yield the same complexity result for TDP

Complexity of TDP

- G. Lan obtained complexity of SDDP (and EDDP) in 2020
  Precision of $T\epsilon$ archived after at most $T(\frac{D}{\epsilon}+1)^N$ iterations
  D diameter state spaces
  N dimension state/control (decision) space
- Straightforward modifications of Lan's proof yield the same complexity result for TDP
- For TDP, overall complexity depends on the complexity of computing basic functions

Selection mapping	Computational difficulty
SDDP	$\mathrm{Card}\left(\mathsf{W}_{t+1}\right)$ LPs
U	$\mathrm{Card}\left(\mathsf{W}_{t+1}\right) \cdot \mathrm{Card}\left(F\right)$ QPs
V	one LP

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions
- Approximations refined along the Problem-child trajectory without discretizing state space

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions
- Approximations refined along the Problem-child trajectory without discretizing state space
- Gap between upper and lower approximation vanishes along the Problem-child trajectory

Conclusion on TDP

- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions
- Approximations refined along the Problem-child trajectory without discretizing state space
- Gap between upper and lower approximation vanishes along the Problem-child trajectory
- Generalizes [Philpott and al. (2013)] and [Baucke and al. (2018)] for a variant of SDDP

Conclusion on TDP

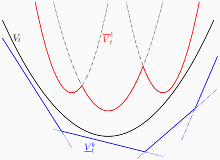
- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions
- Approximations refined along the Problem-child trajectory without discretizing state space
- Gap between upper and lower approximation vanishes along the Problem-child trajectory
- Generalizes [Philpott and al. (2013)] and [Baucke and al. (2018)] for a variant of SDDP
- Additional results (deterministic case) in [Akian, Chancelier, T. (2018)]

Conclusion on TDP

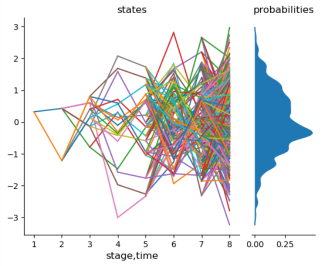
- Monotonic approximations $\left(\overline{V}_t^k\right)_k$ and $\left(\underline{V}_t^k\right)_k$ of V_t
- Min-plus linear or Max-plus linear combinations of basic functions
- Tight and Valid basic functions
- Approximations refined along the Problem-child trajectory without discretizing state space
- Gap between upper and lower approximation vanishes along the Problem-child trajectory
- Generalizes [Philpott and al. (2013)] and [Baucke and al. (2018)] for a variant of SDDP
- Additional results (deterministic case) in [Akian, Chancelier, T. (2018)]
- Currently working with Vincent Guigues on regularization techniques for SDDP

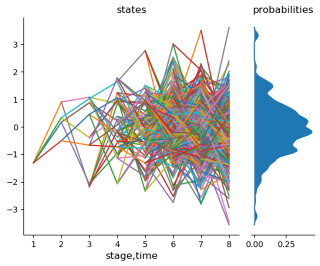
Recall of the outline

1. Tropical Dynamic Programming
with M. Akian (Ecole Polytechnique) and
J-P. Chancelier (Ecole des Ponts ParisTech)



2. Entropic Regularization of the Nested Distance
with Z. Qu (Hong Kong University)





Part content

A distance between scenario trees

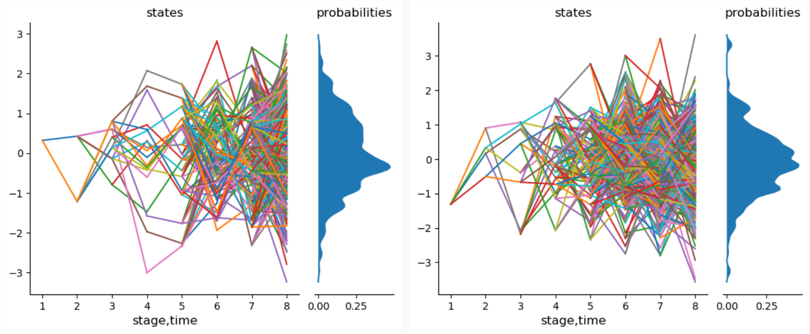


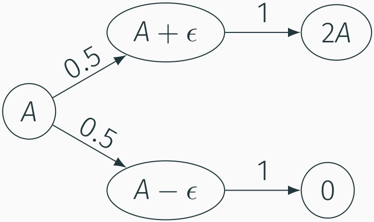
Figure 3: Two scenario trees X and Y, Nested Distance is $\mathrm{ND}_2(X, Y) = 1.009$ its entropic regularization is $\mathrm{END}_2(X, Y) = 1.011$.

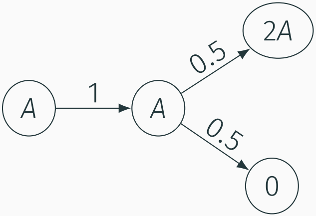
Scenario tree

A stochastic process $(X_t)_{t \in [\![1,T]\!]}$ is a scenario tree if it is also discrete and finite in space

Buying an object with random prices at the best average price.

$$v(Z) = \min_{\mathrm{u}} \left\{ \mathbb{E}\left[\sum_{t=0}^{2} Z_t \mathrm{u}_t\right] \;\middle|\; \begin{array}{c} \mathrm{u}_t \in \{0,1\}, \\ \mathrm{u}_t \text{ is } \mathcal{F}_t\text{-measurable}, \\ \sum_{t=0}^{T} \mathrm{u}_t = 1, \end{array} \right\}.$$





Proximity in Wasserstein metric

$$\mathrm{W}(X, Y) = 2\epsilon$$

Arbitrarily large gap in values

$$|v(X) - v(Y)| = \frac{A - \epsilon}{2}$$

Wasserstein distance is not suited for MSP

For every $L > 0$, there exists a initial price A s.t.

$$\underbrace{|v(X) - v(Y)|}_{\frac{A-\epsilon}{2}} \not\leq L \cdot \underbrace{\mathrm{W}(X, Y)}_{2\epsilon}$$

From the Wasserstein distance to the Nested Distance (2/2)

Wasserstein distance is not suited for MSP

For every $L > 0$, there exists a initial price A s.t.

$$\underbrace{|v(X) - v(Y)|}_{\frac{A-\epsilon}{2}} \not\leq L \cdot \underbrace{\mathrm{W}(X, Y)}_{2\epsilon}$$

The Nested Distance is suited for MSP

There exists $L > 0$ s.t. for every initial price A

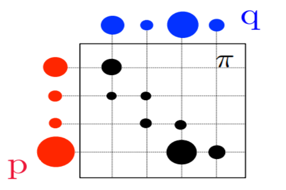
$$|v(X) - v(Y)| \leq L \cdot \underbrace{\mathrm{ND}(X, Y)}_{=A+\epsilon} \quad \text{(Pflug and Pichler, 2012)}$$

Optimal Transport

$\mathrm{OT}(p, q; c) = \min_{\pi \in \mathbb{R}_+^{n \times m}} \sum_{\substack{1 \le i \le n \\ 1 \le j \le m}} c_{ij}\pi_{ij}$ s.t. $\pi \mathbf{1}_m = p, \pi^T \mathbf{1}_n = q$

π transport plan if it satisfies the mass constraints

$$\begin{cases} \pi \mathbf{1}_m = p \\ \pi^T \mathbf{1}_n = q \end{cases}$$

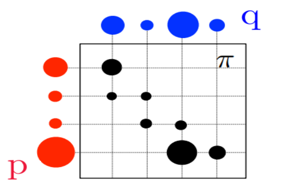


Optimal Transport

$$\mathrm{OT}(p, q; c) = \min_{\pi \in \mathbb{R}_+^{n \times m}} \sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} c_{ij} \pi_{ij} \text{ s.t. } \pi \mathbf{1}_m = p, \pi^T \mathbf{1}_n = q$$

π transport plan if it satisfies the mass constraints

$$\begin{cases} \pi \mathbf{1}_m = p \\ \pi^T \mathbf{1}_n = q \end{cases}$$



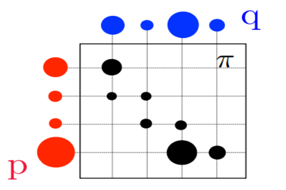
Discrete entropy of $\pi \in \mathbb{R}_+^{n \times m}$, $H(\pi) = -\sum_{i,j} \pi_{ij} \log(\pi_{ij})$

Optimal Transport

$$\mathrm{OT}(p,q;c) = \min_{\pi\in\mathbb{R}_+^{n\times m}} \sum_{\substack{1\le i\le n\\1\le j\le m}} c_{ij}\pi_{ij} \text{ s.t. } \pi\mathbf{1}_m = p, \pi^T\mathbf{1}_n = q$$

π transport plan if it satisfies the mass constraints

$$\begin{cases}\pi\mathbf{1}_m = p\\ \pi^T\mathbf{1}_n = q\end{cases}$$



Discrete entropy of $\pi\in\mathbb{R}_+^{n\times m}$, $H(\pi) = -\sum_{i,j}\pi_{ij}\log(\pi_{ij})$

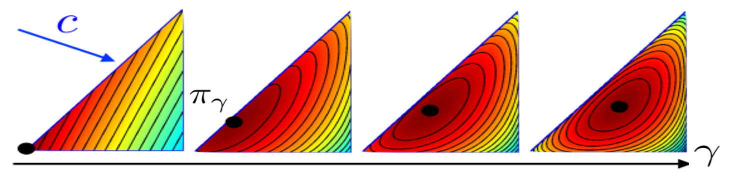
Regularized Optimal Transport

$$\mathrm{OT}_\gamma(p,q;c) = \min_{\pi\in\mathbb{R}_+^{n\times m}} \sum_{\substack{1\le i\le n\\1\le j\le m}} c_{ij}\pi_{ij} \underbrace{-\gamma H(\pi)}_{\text{strongly convex}}$$

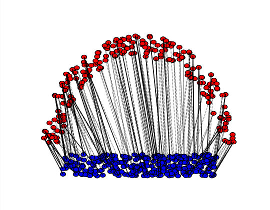
$$\text{s.t. } \pi\mathbf{1}_m = p, \pi^T\mathbf{1}_n = q$$

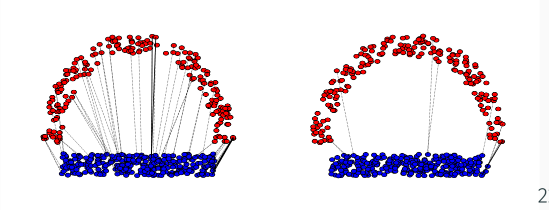
Optimal Transport and Regularized Optimal Transport (2/2)

Regularization pushes the optimal transport plan away from the boundary, illustration from Peyré and Cuturi (2019)



The regularized optimal transport plan is stable, diffuse





Part content

Sinkhorn's Algorithm to solve regularized OT (Peyré Cuturi 2019)

The optimal regularized transport plan is a rescaling of the Gibbs kernel G. $\pi^* = \mathrm{diag}\left(u^*\right) G\, \mathrm{diag}\left(v^*\right), u^*, v^* > 0$, where G is the Gibbs kernel defined by $G_{ij} = \exp\left(\frac{c_{ij}}{\gamma}\right)$.

Sinkhorn's Algorithm to solve regularized OT (Peyré Cuturi 2019)

The optimal regularized transport plan is a rescaling of the Gibbs kernel G. $\pi^* = \mathrm{diag}\,(u^*)\, G\, \mathrm{diag}\,(v^*)\,, u^*, v^* > 0$, where G is the Gibbs kernel defined by $G_{ij} = \exp\left(\frac{c_{ij}}{\gamma}\right)$.

Alternatively rescaling the lines and columns of G (Sinkhorn's algorithm) converges to π^*.

$$\begin{cases} u_{k+1} = \mathbb{1}_n \ ./\ (G v_k) \quad (./ \text{ entrywise division}) \\ v_{k+1} = \mathbb{1}_m \ ./\ (G u_{k+1})\,, \end{cases}$$

Sinkhorn's Algorithm to solve regularized OT (Peyré Cuturi 2019)

The optimal regularized transport plan is a rescaling of the Gibbs kernel G. $\pi^* = \mathrm{diag}(u^*)\, G\, \mathrm{diag}(v^*)$, $u^*, v^* > 0$, where G is the Gibbs kernel defined by $G_{ij} = \exp\left(\frac{c_{ij}}{\gamma}\right)$.

Alternatively rescaling the lines and columns of G (Sinkhorn's algorithm) converges to π^*.

$$\begin{cases} u_{k+1} = \mathbb{1}_n \ ./\ (G v_k) & (./ \text{ entrywise division}) \\ v_{k+1} = \mathbb{1}_m \ ./\ (G u_{k+1}), \end{cases}$$

Sinkhorn's algorithm converges linearly to π^*.

Sinkhorn's Algorithm to solve regularized OT (Peyré Cuturi 2019)

The optimal regularized transport plan is a rescaling of the Gibbs kernel G. $\pi^* = \mathrm{diag}(u^*)\, G\, \mathrm{diag}(v^*)$, $u^*, v^* > 0$, where G is the Gibbs kernel defined by $G_{ij} = \exp\left(\frac{c_{ij}}{\gamma}\right)$.

Alternatively rescaling the lines and columns of G (Sinkhorn's algorithm) converges to π^*.

$$\begin{cases} u_{k+1} = \mathbb{1}_n \ ./ \left(G v_k\right) & \text{(./ entrywise division)} \\ v_{k+1} = \mathbb{1}_m \ ./ \left(G u_{k+1}\right), & \end{cases}$$

Sinkhorn's algorithm converges linearly to π^*.

Overall complexity when $m = n$. For every $\epsilon > 0$, setting $\gamma = \frac{\epsilon}{4\log(n)}$, Sinkhorn's algorithm computes π^* in $O\left(n^2 \log(n) \epsilon^{-3}\right)$ operations s.t. $\sum_{ij} \pi^*_{ij} c_{ij} \leq \mathrm{OT}(p, q; c) + \epsilon$.

Part content

Nested Distance between scenario trees

. X and Y two scenario tree, $r \geq 1$, $d(x, y) = \|x - y\|_r$ over $\mathbb{R}^T$

. Compute recursively backward in time functions
$c_t : \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \to \overline{\mathbb{R}}$

$$\begin{cases} c_T(x_{1:T}, y_{1:T}) = d(x_{1:T}, y_{1:T}), \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T}, \\ \\ \end{cases}$$

Nested Distance between scenario trees

. X and Y two scenario tree, $r \geq 1$, $d(x,y) = \|x-y\|_r$ over $\mathbb{R}^T$

. Compute recursively backward in time functions
$c_t : \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \to \overline{\mathbb{R}}$

$$\begin{cases} c_T(x_{1:T}, y_{1:T}) = d(x_{1:T}, y_{1:T}), \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T}, \\ c_t(x_{1:T}, y_{1:T}) = \mathrm{OT}\Big(P_{t+1}(\cdot \mid X_{1:t} = x_{1:t}), \tilde{P}_{t+1}(\cdot \mid Y_{1:t} = y_{1:t}); c_{t+1}^r\Big)^{1/r}, \\ \quad \forall t \in [\![1, T-1]\!], \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T}. \end{cases}$$

Nested Distance between scenario trees

- X and Y two scenario tree, $r \geq 1$, $d(x,y) = \|x-y\|_r$ over $\mathbb{R}^T$
- Compute recursively backward in time functions $c_t : \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \to \overline{\mathbb{R}}$

$$
\begin{cases}
c_T(x_{1:T}, y_{1:T}) = d(x_{1:T}, y_{1:T}), \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T}, \\
c_t(x_{1:T}, y_{1:T}) = \mathrm{OT}\left(P_{t+1}(\cdot \mid X_{1:t} = x_{1:t}), \tilde{P}_{t+1}(\cdot \mid Y_{1:t} = y_{1:t}); c_{t+1}^r\right)^{1/r}, \\
\quad \forall t \in [\![1, T-1]\!], \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T}.
\end{cases}
$$

$\mathrm{ND}_r(X,Y) := \mathrm{OT}\left(P_T, \tilde{P}_T, c_1^r\right)^{1/r}$ is the r-Nested Distance

Entropic regularization of the Nested Distance

. X and Y two scenario tree, $r \geq 1$, $d(x,y) = \|x-y\|_r$ over $\mathbb{R}^T$

. Compute recursively backward in time functions
$c_t : \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \to \overline{\mathbb{R}}$

$$\begin{cases} c_T(x_{1:T}, y_{1:T}) = d(x_{1:T}, y_{1:T}), \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \\ c_t(x_{1:T}, y_{1:T}) = \mathrm{OT}_\gamma\left(P_{t+1}(\cdot \mid X_{1:t} = x_{1:t}), \tilde{P}_{t+1}(\cdot \mid Y_{1:t} = y_{1:t}); c_{t+1}^r\right)^{1/r} \\ \quad \forall t \in [\![1, T-1]\!], \ \forall (x_{1:T}, y_{1:T}) \in \mathbb{X}_{1:T} \times \mathbb{Y}_{1:T} \\ \quad \gamma = \max\limits_{\substack{x_{1:t+1} \in x_{1:t}^+ \\ y_{1:t+1} \in y_{1:t}^+}} c_{t+1}^r(x_{1:t+1}, y_{1:t+1})/30. \end{cases}$$

$\mathrm{END}_r(X,Y) := \mathrm{OT}_\gamma\left(P_T, \tilde{P}_T, c_1^r\right)^{1/r}$ is the Entropic regularization of the r-Nested Distance

Nested Distance (ND) vs Entropic Nested Distance (END)

Main property of the Nested Distance is preserved

$$|v(X) - v(Y)| \leq L \cdot \mathrm{ND}_r\left(X, Y\right) \leq L \cdot \mathrm{END}_r\left(X, Y\right).$$

Nested Distance (ND) vs Entropic Nested Distance (END)

Main property of the Nested Distance is preserved

$$|v(X) - v(Y)| \leq L \cdot \mathrm{ND}_r(X, Y) \leq L \cdot \mathrm{END}_r(X, Y).$$

Toy problem with varying horizon T

Horizon T	ND_2 (ms)	END_2 (ms)	Speedup	Rel. error (%)
2	0.26	0.014	16	0.14
4	3.8	0.14	25	0.25
6	115	6.3	33	0.51
8	1077	28	35	0.35
10	18205	493	36	0.41

Average results after 10 runs, Jupyter notebook in Julia 1.5.2 of this experiment is available at `https://github.com/BenoitTran/END`

Perspectives on regularizations of the Nested Distance

- Pichler and Weinhardt (2021): dual characterization of regularized ND and upper bound on approximation error

Perspectives on regularizations of the Nested Distance

- Pichler and Weinhardt (2021): dual characterization of regularized ND and upper bound on approximation error
- Blondel and al. (2018): OT problems with sparsity inducing regularization term

Perspectives on regularizations of the Nested Distance

- Pichler and Weinhardt (2021): dual characterization of regularized ND and upper bound on approximation error
- Blondel and al. (2018): OT problems with sparsity inducing regularization term
- Entropic regularization computes approximate value of OT problem but optimal transport plan is different: optimal transport plan is sparse whereas regularized optimal transport plan is dense.

Perspectives on regularizations of the Nested Distance

- Pichler and Weinhardt (2021): dual characterization of regularized ND and upper bound on approximation error
- Blondel and al. (2018): OT problems with sparsity inducing regularization term
- Entropic regularization computes approximate value of OT problem but optimal transport plan is different: optimal transport plan is sparse whereas regularized optimal transport plan is dense.
- Work in progress: Sparse regularization of the Nested Distance for scenario tree reduction algorithms

Webpage: https://benoittran.github.io/
E-mail: benoit.tran@tutanota.com

Thank you !