Boosting Reduction Tree via Wasserstein Barycenters

Daniel Mimouni[1,2], Paul Malisani[2], Jiamin Zhu[2], Welington de Oliveira[1]

1 **Centre de Mathématiques Appliquées - CMA, Mines Paris PSL, France**

2 **IFP Energies nouvelles, France**

PGMO days
Scenarios methods in stochastic control and applications

Outline

I. Multistage Stochastic Optimization Problem Context

II. Kovacevic and Pichler's Reduction Tree Method

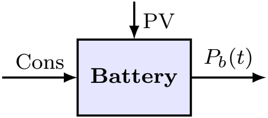
III. The Probability Optimization Step is a Wasserstein Barycenters Problem

IV. Applications

I. Multistage Stochastic Optimization Problem Context

Some Context

- Renewable energy integration challenges: Unlike fossil fuels, renewable power generation is variable and weather-dependent, making grid stability more complex.
- Demand-side flexibility as a solution: Adapting consumer energy use to match real-time conditions helps optimize renewable energy use but requires advanced management systems.
- Optimization-based management systems: Stochastic optimization techniques enable effective scheduling and resource allocation in uncertain conditions, essential for integrating renewables.



The effective power demand:

$$f_t := P_m(t) = \text{Cons} - \text{PV} + \frac{1}{\rho_c} \max\{P_b(t), 0\} + \rho_d \min\{P_b(t), 0\} \tag{1}$$

The stage-wise cost function:

$$c_t(P_m(t), P_b(t), (\text{Cons}, \text{PV})) = p_r^b(t) \max\{P_m(t), 0\} + p_r^s(t) \min\{P_m(t), 0\} \tag{2}$$

Optimization problem

Multistage stochastic optimization problem:

$$\min_{u_1} c_1(x_1, u_1, \xi_1) + \min_{u_2} \mathbb{E}_{\xi_2} \left[c_2(x_2, u_2, \xi_2) + \cdots + \min_{u_T} \mathbb{E}_{\xi_T} \left[c_T(x_T, u_T, \xi_T) \right] \right] \tag{3}$$

Under the following constraints:

$$x_{t+1} = f_t(x_t, u_t, \xi_t), \qquad t = 1, \ldots, T-1 \tag{4a}$$

$$(u_t, x_t) \in K_t \subset \mathbb{R}^m \times \mathbb{R}^n, \qquad t = 1, \ldots, T \tag{4b}$$

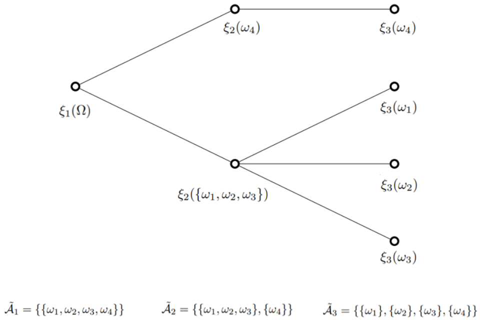
$$x_1 = x^0, \tag{4c}$$

Resolution methods

- MPC solves deterministic optimization problem at each time step thus does not use the statistical properties of the future random variables, potentially yielding far from sub-optimal decisions.

- SDDP is a sequential decomposition method, that needs strong assumption like stage-wise independence.

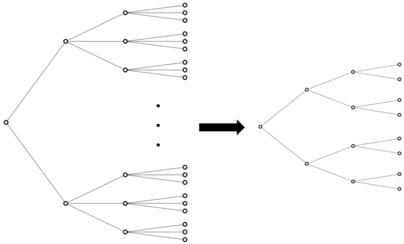
- PHA is a scenario decomposition techniques that decomposes the problem per scenario while keeping the whole time horizon in individual (scenario-based) subproblems.



In practice, the scenario process $\{\xi_t\}$ is approximated by a scenario tree

Reducing Trees

- For statistical representativity, the scenario tree should be large
- For computation tractability, the scenario tree should be small



How to Compare Trees?

The Nested Distance

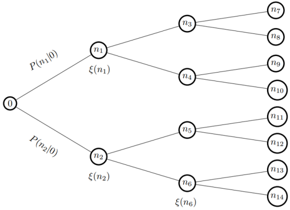
- Has good stability results[1].
- Takes filtration into account.

[1] See Pflug and Pichler 2012

Distance between processes

Let two T-period scenario trees with set of nodes $\mathcal{N}, \mathcal{N}'$:

- The ancestors of $n \in \mathcal{N}$ are $\mathcal{A}(n)$.
- The distance between two nodes at stage t, is $\mathtt{d}_{n_1,n_2}$.
- The transport mass between nodes at stage t, is noted $\pi_{i,j}$ or $\pi(i,j)$.



The Nested Distance

For $\iota \in [1,\infty)$, the process distance of order ι between $\mathbf{P}$ and $\mathbf{P'}$ is the ι^{th} root of the optimal value of the following LP:

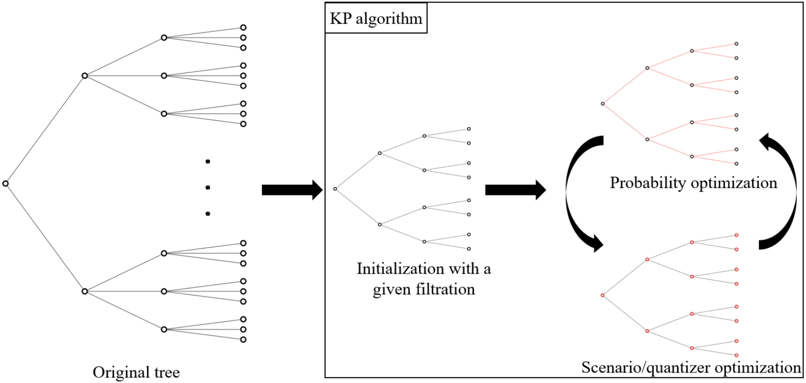
$$\mathrm{ND}_\iota(\mathbf{P},\mathbf{P'}) := \begin{cases} \min\limits_{\pi} & \sum\limits_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi(i,j)\mathtt{d}^{\iota}_{i,j} \\ \text{s.t.} & \sum_{\{j:n\in\mathcal{A}(j)\}} \pi(i,j|m,n) = P(i|m), \quad (m\in\mathcal{A}(i), n) \\ & \sum_{\{i:m\in\mathcal{A}(i)\}} \pi(i,j|m,n) = P'(j|n), \quad (n\in\mathcal{A}(j), m) \\ & \pi_{i,j} \geq 0 \text{ and } \sum_{i,j}\pi_{i,j} = 1. \end{cases} \tag{NDT}$$

[2]

[2] Multimarginal optimal transport problem.

II. Kovacevic and Pichler's Reduction Tree Method

Kovacevic and Pichler's algorithm (KP)



KP algorithm: to approximate a tree, a smaller tree with a given filtration is improved in order to minimize the distance with the original tree. The probabilities and the scenario values are alternatively optimized until convergence.

Probability Optimization

Given the stochastic process quantizers $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ and structure of $(\mathcal{N}', A')$, we are looking for the optimal probability measure P' to approximate $\mathbf{P} := (\Xi^{T+1}, \mathcal{F}, P)$, regarding the nested distance.

Recursive Problem

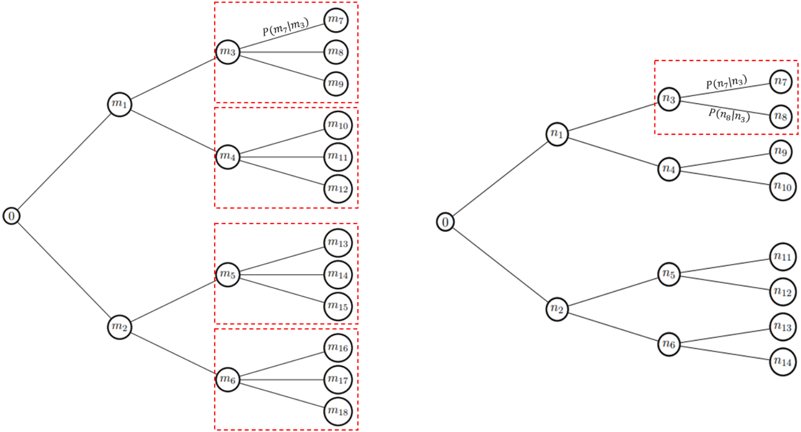
$$\begin{cases} \min\limits_{\pi} & \sum_{m \in \mathcal{N}_t} \pi(m,n) \sum_{i \in m+, j \in n+} \pi(i,j|m,n) \delta_\iota(i,j) & \\ \text{s.t.} & \sum_{j \in n+} \pi(i,j|m,n) = P(i|m), & (i \in m+) \\ & \sum_{i \in m+} \pi(i,j|m,n) = \sum_{i \in \tilde{m}+} \pi(i,j|\tilde{m},n), & (j \in n+ \text{ and } m, \tilde{m} \in \mathcal{N}_t) \\ & \pi(i,j|m,n) \geq 0. & \end{cases} \tag{RP}$$

- Computationally expensive due to the solving of potentially large-scale LPs repeatedly. Can be untractable for large-scale scenario trees.

III. The Probability Optimization Step is a Wasserstein Barycenters Problem

Wasserstein Barycenter (WB) within the KP Algorithm

- In the Scenario Reduction problem with seek $\mathbf{P}'$ (with given filtration $\mathcal{F}'_t$) that minimizes $\mathrm{ND}_2(\mathbf{P}, \mathbf{P}')$
- Our first contribution is to notice than the steps of the KP algorithm is a Wasserstein Barycenter problems (WB)



(left) Original tree, *(right)* Approximated tree. The probabilities $(P(n_7|n_3), P(n_8|n_3))$ are computed as the Wasserstein Barycenter of the set of (known) probabilities associated to the boxed subtrees on the left.

The recursive problem (RP) is a Wasserstein Barycenter problem.

Given $t \in \{1, \ldots, T\}$ and $n \in \mathcal{N}'_t$, problem (RP) reads as:

$$\begin{cases} \min\limits_{\pi} & \sum_{m \in \mathcal{N}_t} \pi(m,n) \sum_{i \in m+, j \in n+} \pi(i,j|m,n)\delta_\iota(i,j) \\ \\ \text{s.t.} & \sum_{j \in n+} \pi(i,j|m,n) = P(i|m), \qquad (i \in m+) \\ & \sum_{i \in m+} \pi(i,j|m,n) = \sum_{i \in \tilde{m}+} \pi(i,j|\tilde{m},n), \qquad (j \in n+ \text{ and } m, \tilde{m} \in \mathcal{N}_t) \\ & \pi(i,j|m,n) \geq 0. \end{cases} \tag{RP}$$

- $\sum_{i \in m+} \pi(i,j|m,n) = P'(j|n)$ for $j \in n+$, for all $m = m_1, \ldots, m_M$.

From a large LP to an Optimal Transport Problem

The recursive problem (RP) is a Wasserstein Barycenter problem.

Given $t \in \{1, \dots, T\}$ and $n \in \mathcal{N}_t'$, problem (RP) reads as:

$$\left\{\begin{array}{llll}
\min\limits_{P',\pi} & \pi(m_1,n) \sum\limits_{i \in m_1+, j \in n+} \pi(i,j|m_1,n)\delta_\iota(i,j) \quad + \cdots + \quad \pi(m_M,n) \sum\limits_{i \in m_M+, j \in n+} \pi(i,j|m_M,n) & & \\
\text{s.t.} & \sum_{j \in n+} \pi(i,j|m_1,n) & = P(i|m_1), & (i \in m_1+) \\
& \qquad \ddots & \vdots & \\
& \qquad\qquad \sum_{j \in n+} \pi(i,j|m_M,n) & = P(i|m_M), & (i \in m_M+) \\
& \sum_{i \in m_1+} \pi(i,j|m_1,n) & = P'(j|n), & (j \in n+) \\
& \qquad \ddots & \vdots & \\
& \qquad\qquad \sum_{i \in m_M+} \pi(i,j|m_M,n) & = P'(j|n), & (j \in n+) \\
& \sum_{j \in n+} P'(j|n) = 1, \pi(i,j|m_1,n) \geq 0 \quad \cdots & \pi(i,j|m_M,n) \geq 0, &
\end{array}\right. \tag{WB}$$

- $\sum_{i=1}^{M} \pi(m_i, n) = 1$ per definition. Kovacevic and Pichler fix $\pi(m,n)$ with the values computed from the previous iteration.

From a large LP to an Optimal Transport Problem

The recursive problem (RP) is a Wasserstein Barycenter problem.

Given $t \in \{1, \dots, T\}$ and $n \in \mathcal{N}'_t$, problem (RP) reads as:

$$
\begin{cases}
\min\limits_{P',\pi} & \alpha_1^n \sum\limits_{i \in m_1+, j \in n+} \pi(i,j|m_1,n)\delta_\iota(i,j) \quad + \cdots + \quad \alpha_M^n \sum\limits_{i \in m_M+, j \in n+} \pi(i,j|m_M,n)\delta_\iota(i,j \\
\text{s.t.} & \sum_{j \in n+} \pi(i,j|m_1,n) \qquad = P(i|m_1), \quad (i \in m_1+) \\
& \qquad \ddots \qquad \vdots \\
& \qquad \sum_{j \in n+} \pi(i,j|m_M,n) \qquad = P(i|m_M), \; (i \in m_M+) \\
& \sum_{i \in m_1+} \pi(i,j|m_1,n) \qquad = P'(j|n), \quad (j \in n+) \\
& \qquad \ddots \qquad \vdots \\
& \qquad \sum_{i \in m_M+} \pi(i,j|m_M,n) \qquad = P'(j|n), \quad (j \in n+) \\
& \sum_{j \in n+} P'(j|n) = 1, \pi(i,j|m_1,n) \geq 0 \quad \cdots \quad \pi(i,j|m_M,n) \geq 0,
\end{cases}
\tag{WB}
$$

This is a Wasserstein barycenter problem, with:

- The right constraints on mass conservation,
- The left constraints on unicity of the barycenter.

Meilleure illustration? The Wasserstein distance

Let empirical (discrete) measures being defined like:

$$\texttt{supp}(\nu) := \{\xi'_1, \ldots, \xi'_S\} \quad \text{and} \quad \nu = \sum_{s=1}^{S} q_s \underline{\delta}_{\xi'_s}. \tag{5}$$

The Wasserstein distance

Given two probability measures $\mu, \nu \in P(\mathbb{R}^d)$, their Wasserstein distance is the ι-th root of

$$W_\iota^\iota(\mu, \nu) := \min_{\pi \in U(\mu,\nu)} \sum_{r,s} D_{r,s} \pi_{r,s}. \tag{WD}$$

- D is the cost matrix between the measures,
- π is the transport matrix between the measures,
- $U(\mu, \nu)$ is the set of all transport plans having marginals μ and ν.

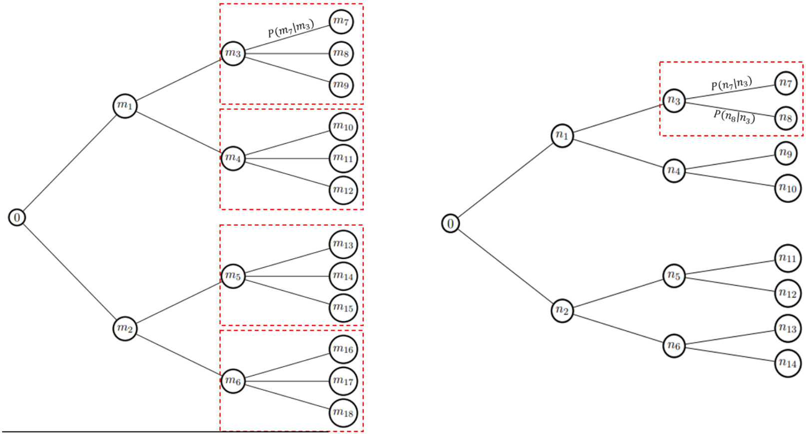
Wasserstein Barycenter

Given M measures $\{\nu^1, \ldots, \nu^M\}$ in $P(\mathbb{R}^d)$, an ι-*Wasserstein barycenter* with weights $\alpha \in \Delta_M$ is a solution to the following optimization problem:

$$\min_{\mu \in P(\mathbb{R}^d)} \sum_{m=1}^{M} \alpha_m W_\iota^\iota(\mu, \nu^m). \tag{6}$$

Wasserstein Barycenter (WB) within the KP Algorithm

- Our first contribution is to notice than the steps of the KP algorithm is a Wasserstein Barycenter problems (WB)
- (WB) can be solved with specialized techniques: MAM (Method of Averaged Marginals of [3]), IBP (Iterative Bregman Projection of [4]).



[3] [Mimouni, D. W., Malisani, P., Zhu, J., & de Oliveira, W. SIAM Journal on Mathematics of Data Science (2024)]

[4] [Benamou, J. D., Carlier, G., Cuturi, M., Nenna, L., & Peyré, G. SIAM Journal on Scientific Computing. (2015)]

Boosted Algorithm

Scenario tree reduction via nested distance and Wasserstein barycenters

▷ Step 0: input

Let the original scenario tree $\mathbf{P} = (\Xi^{T+1}, \mathcal{F}, P)$ and a smaller scenario tree $\mathbf{P'}^0 = (\Xi^{T+1}, \mathcal{F}', {P'}^0)$ be given.
Choose a tolerance $\texttt{Tol} > 0$

for $k = 0, 1, 2, \ldots$ **do**
 ▷ **Step 1:** Improve the scenario values (quantizers)
  If $\iota = 2$ use an analytic solution otherwise do a gradiant descent.

 ▷ **Step 2:** Improve the probabilities
  for $t = T-1, \ldots, 0$ **do** ▷ Recursivity
   for all $n \in \mathcal{N}'_t$ **do** ▷ Wasserstein barycenters
    Set $\alpha_m^n \leftarrow \pi^k(m, n)$, $m \in \mathcal{N}_t$
    Use IBP, or MAM to compute $\pi^{k+1}(\cdot, \cdot|\cdot, n)$ solving (WB)
   end for
  end for

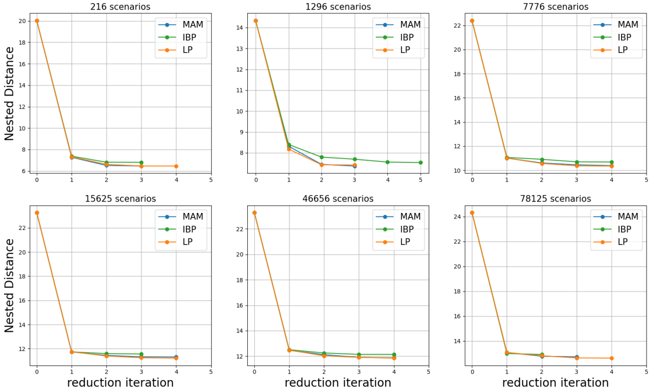
 ▷ **Step 3:** Stopping test
  if $\delta_\iota^k(0,0) - \delta_\iota^{k+1}(0,0) \leq \texttt{Tol}$ **then**
   Define $P'(n_T) = \sum_{m_T \in \mathcal{N}_T} \pi^{k+1}(m_T, n_T)$ for all $n_T \in \mathcal{N}'_T$ then $P'(n) = \sum_{j \in n+} P'(j)$ for all $n \in \mathcal{N}'_t, t \neq T$
   Set $\mathrm{ND}_\iota(\mathbf{P}, \mathbf{P}') \leftarrow \delta_\iota^{k+1}(0,0)$
   Stop and return with the reduced tree $\mathbf{P}' = (\Xi^{T+1}, \mathcal{F}', P')$ and nested distance $\mathrm{ND}_\iota(\mathbf{P}, \mathbf{P}')$
  end if
end for

IV. Applications

Reduction scenario applications

Scenario tree reduction employing different solvers to compute the WBs:

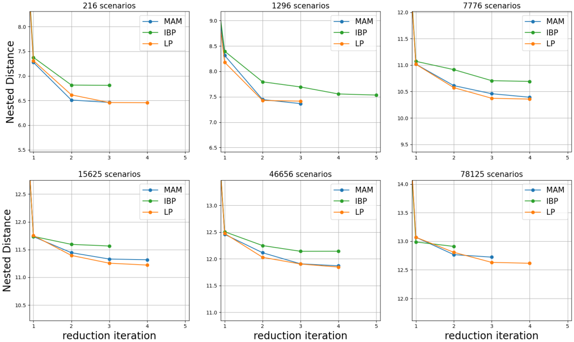
- A classic LP (KP setting),
- Iterative Bregmann Projection algorithm [5],
- MAM.



Evolution of the Nested Distance along the reduction iterations for different initial trees.

[5] see the work of D. Bennammou and G. Peyré

Reduction scenario applications



Evolution of the Nested Distance along the reduction iterations for different initial trees with a zoom.

	LP	IBP	MAM	MAM 4 processors
T=4, cpn=6	0.17	0.49	2.21	0.56
T=5, cpn=6	1.54	14.83	18.23	6.28
T=6, cpn=6	74.25	161.19	344.83	124.44
T=7, cpn=5	487.58	323.76	816.46	341.62
T=7, cpn=6	4905	2136	2541	1256
T=8, cpn=5	13797	4334	3458	1635

Table: Total time (in seconds) per method for the studied trees.

Impact of the initialization

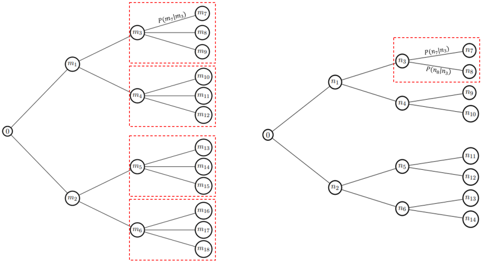
- *Kmeans method*, starting from 100 scenarios it creates 25 clusters using the Euclidean norm, and then computes the 25 corresponding barycenters;
- *Fast Forward Selection (FFS) method*, introduced by Heitsch and Römisch. The method iteratively selects scenarios that minimize the Wasserstein distance to the remaining scenarios. At each step, the scenario that best approximates the distribution is added to the reduced set until the desired number of 25 scenarios is reached, ensuring an efficient yet effective reduction.

Scenario set	Filtrations	initial ND	reduced ND
1	**Kmeans**	2757	1219
	FFS	1384	658
2	**Kmeans**	1699	1092
	FFS	1936	896
3	**Kmeans**	1653	832
	FFS	1499	716
4	**Kmeans**	1858	963
	FFS	1161	566
5	**Kmeans**	1968	1054
	FFS	917	540

Comparison of the ND to the original tree before and after tree reduction using different initialization techniques.

Take-away messages

- New approach to tackle scenario tree reduction
- New easy-to-implement and memory efficient algorithm for reducing scenario trees
- Can leverage parallelization of transport optimal techniques
- Makes more accessible (because more efficient) a technique that keeps maximal information from the initial modelization



Thank you!

D. Mimouni, P. Malisani, J. Zhu, W. de Oliveira. Scenario Tree Reduction via Wasserstein Barycenters.
Submitted to Annals of Operational Research, 2024

- Preprint available at
  https://arxiv.org/pdf/2309.05315.pdf
- Python code is freely available at
  https://github.com/dan-mim/Nested_tree_reductionb

Contact:

✉ daniel.mimouni@ifpen.fr

https://dan-mim.github.io

Appendix

Stability Results for ND

Stability result for the ND

Consider the value function $\mathrm{val}(\mathbf{H})$ of stochastic optimization problem seen earlier so that $\mathrm{val}(\mathbf{H}) := \mathrm{val}(\xi^H)$, and L_2 a constant, then it holds[6]:

$$|\mathrm{val}(\mathbf{H}) - \mathrm{val}(\mathbf{G})| \leq L_2 \cdot \mathrm{dl}_2(\mathbf{H}, \mathbf{G})^2 \tag{7}$$

- It is not the case when using the WD.

[6] See Pflug and Pichler 2012

Probability Optimization in the KP algorithm

Given the stochastic process quantizers $\{\xi'(n) \in \Xi : n \in \mathcal{N}'\}$ and structure of $(\mathcal{N}', A')$, we are looking for the optimal probability measure P' to approximate $\mathbf{P} := (\Xi^{T+1}, \mathcal{F}, P)$, regarding the nested distance.

Large non-convex optimization problem

$$
\begin{cases}
\min\limits_{\pi, P'} & \sum\limits_{i \in \mathcal{N}_T, j \in \mathcal{N}'_T} \pi(i,j) \mathtt{d}^{\iota}_{i,j} \\
\text{s.t.} & \sum_{j \in n+} \frac{\pi(i,j)}{\pi(m,n)} = P(i|m), \quad (\forall m \in \mathcal{A}(i), n) \\
 & \sum_{i \in m+} \frac{\pi(i,j)}{\pi(m,n)} = P'(j|n), \quad (\forall n \in \mathcal{A}(j), m) \\
 & \pi_{i,j} \geq 0 \text{ and } \sum_{i,j} \pi_{i,j} = 1 \\
 & P'(j|j-) \geq 0.
\end{cases} \tag{8}
$$

- This is a bilinear problem
- There is a large number of decision variables and bilinear constraints.

From Bilinear to Recursive Problem

- $\pi(i,j) = \pi(i,j|m,n) \times \pi(m,n)$,
- $\delta_\iota(m,n) := \sum_{i\in m+, j\in n+} \pi(i,j|m,n)\delta_\iota(i,j)$ for $m \in \mathcal{N}_t, n \in \mathcal{N}'_t$,
- $\delta_\iota(i,j) = \mathtt{d}_\iota(\xi_i, \xi'_j)^\iota =: \mathtt{d}^\iota_{i,j}$ for the leaves i, j of the trees.

$$\begin{aligned}
\sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi(i,j)\mathtt{d}^\iota_{i,j} &= \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \pi(i,j)\delta_\iota(i,j) && \text{(9a)}\\
&= \sum_{i\in\mathcal{N}_T, j\in\mathcal{N}'_T} \sum_{m\in i-, n\in j-} \pi(i,j|m,n)\pi(m,n)\delta_\iota(i,j) && \text{(9b)}\\
&= \sum_{n\in\mathcal{N}'_{T-1}} \sum_{m\in\mathcal{N}_{T-1}} \pi(m,n) \underbrace{\sum_{i\in m+, j\in n+} \pi(i,j|m,n)\delta_\iota(i,j)}_{\delta_\iota(m,n)} && \text{(9c)}\\
&= \sum_{n\in\mathcal{N}'_{T-1}} \sum_{m\in\mathcal{N}_{T-1}} \pi(m,n)\delta_\iota(m,n). && \text{(9d)}
\end{aligned}$$

Evaluate the ND recursively

$\delta_\iota(0,0) = \mathrm{ND}(\mathbf{P}, \mathbf{P}')$

The Method of Averaged Marginals

MAM algorithm

Input: Initial plan $\pi = (\pi^1, \ldots, \pi^m)$ and parameter $\rho > 0$
Set $S^m \leftarrow \mid \texttt{supp}(q^m) \mid$, for $m = 1, \ldots, M$
Define $a_m \leftarrow (\frac{1}{S^m})/(\sum_{j=1}^{M} \frac{1}{S^j})$ and set $p^m \leftarrow \sum_{s=1}^{S^m} \pi^m_{rs}$, $m = 1, \ldots, M$
Set $D^m \leftarrow \alpha_m\, (\delta_\iota(i,j))_{(i,j)\in m+\times n+}$ and set $q^m = (P(i|m))_{i\in m+}$

while not converged **do**

$\quad p \leftarrow \sum_{m=1}^{M} a_m p^m$ ▷ Average the marginals

$\quad$**for** $m = 1, \ldots, M$ **do**
$\quad\quad$**for** $s = 1, \ldots, S^m$ **do**
$\quad\quad\quad \pi^m_{:s} \leftarrow \texttt{Proj}_{\Delta(q^m_s)}\left(\pi^m_{:s} + 2\frac{p-p^m}{S^m} - \frac{1}{\rho} D^m_{:s}\right) - \frac{p-p^m}{S^m}$
$\quad\quad$**end for**
$\quad\quad p^m \leftarrow \sum_{s=1}^{S^m} \pi^m_{rs}$ ▷ Update the m^{th} marginal
$\quad$**end for**

end while

The Iterative Bregman Projection

IBP algorithm

Input: Given α_m for $m = 1, \ldots, M$, $\lambda > 0$, initialize v^0 and u^0 with an arbitrary positive vector, for example $\mathbb{1}_S$ Initialize p^0, for example $\mathbb{1}_R / R$

Set $D^m \leftarrow \alpha_m \left(\delta_\iota(i, j)\right)_{(i,j) \in m+ \times n+}$ and set $q^m = (P(i|m))_{i \in m+}$

Define $K^m = e^{-\lambda D^m}$ for all $m = 1, \ldots, M$

while not converged **do**

▷ Projections onto the constraints

for m=1,...,M **do**

$$v^{m,k+1} = \frac{q^m}{(K^m)^T u^{m,k}}$$

$$u^{m,k+1} = \frac{p^{k+1}}{K^m v^{m,k+1}}$$

end for

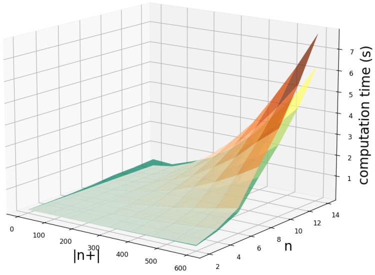
▷ Approximation of the barycenter

$$p^{k+1} = \prod_{m=1}^{M} (K^m v^{m,k+1})^{\alpha_m}$$

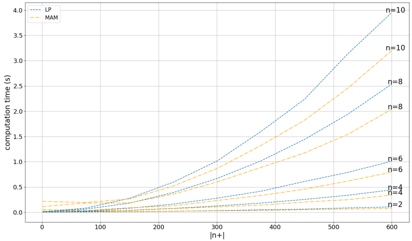
end while

return $\pi^m = \mathrm{diag}(u^m) K^m \mathrm{diag}(v^m)$ for all $m = 1, \ldots, M$

Impact of the tree structure

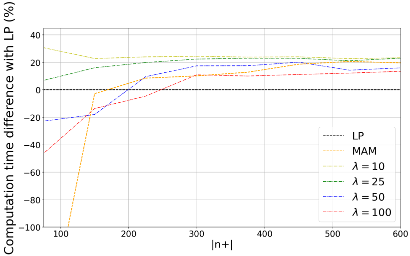


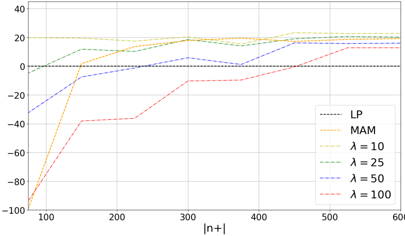
Influence of the tree structure on the computation time of a stage, depending on the method in use: MAM in green and LP in orange.



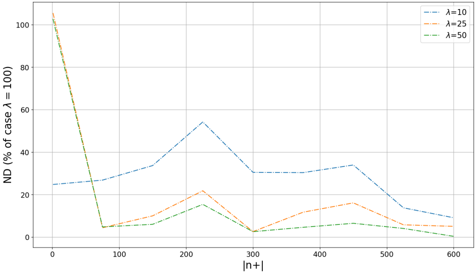
Influence of the tree structure on the computation time for small n.

Impact of the tree structure





Speed comparison with IBP for different λ: A positive time difference means the method is faster than LP. Each curve is obtained by averaging the ND accuracy over $n \in \{2, 4, 6, 8, 10, 12, 14, 16\}$.



Average influence of λ in the precision. Each curve is obtained by averaging the ND accuracy over $n \in \{2, 4, 6, 8, 10, 12, 14, 16\}$.