Numerical Optimal Transport

http://optimaltransport.github.io

Algorithmic Foundations

Gabriel Peyré

www.numerical-tours.com

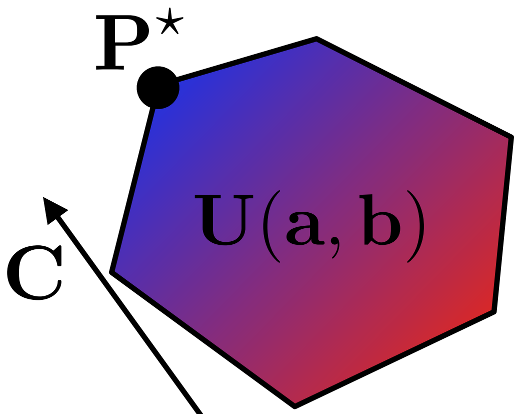
Overview

- **Linear Programming**
- PDE-based
- Semi-discrete (dedicated part)
- Sinkhorn (dedicated part)

Linear Programming

Transportation polytope:

$$\mathbf{U}(\mathbf{a},\mathbf{b}) \overset{\text{def.}}{=} \left\{ \mathbf{P} \in \mathbb{R}_+^{n\times m} \; ; \; \mathbf{P}\mathbb{1}_n = \mathbf{a}, \mathbf{P}^\top \mathbb{1}_m = \mathbf{b} \right\}$$

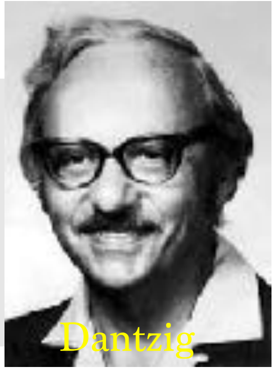


Linear program:

$$\min \left\{ \sum_{i,j} \mathbf{P}_{i,j} \mathbf{C}_{i,j} \; ; \; \mathbf{P} \in U(\mathbf{a},\mathbf{b}) \right\}$$

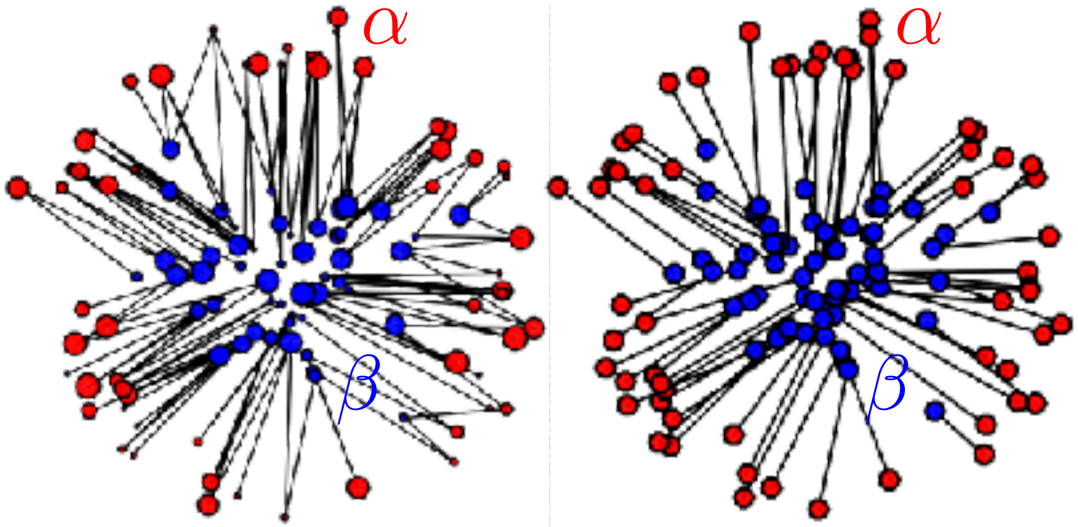
Theorem: $\exists \mathbf{P}^\star$ solution extremal point of $\mathbf{U}(\mathbf{a},\mathbf{b})$

$$|\{(i,j) \; ; \; \mathbf{P}^\star_{i,j} \neq 0\}| \leqslant n+m-1$$



Example: if $n = n$, $\mathbf{a} = \mathbf{b} = \mathbb{1}/n$, $\mathbf{P}^\star$ permutation matrix.

$\rightarrow$ up to $n!$ extremal points.



Interior Point Methods

Linear programming:

$$x_0 \in \underset{x}{\operatorname{argmin}} \ \{\langle x,\, c\rangle \ ; \ i = 1, \ldots, m, \langle a_i,\, x\rangle \leqslant b_i\}$$

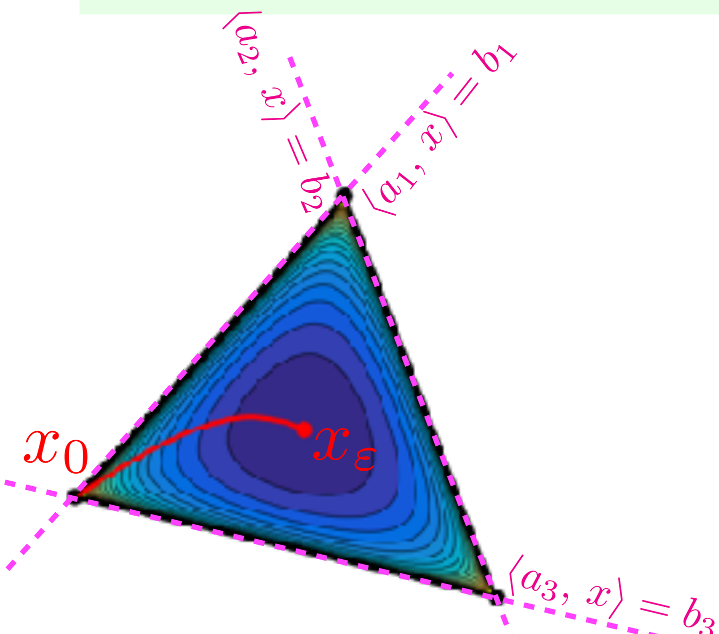
$\varepsilon \to 0$

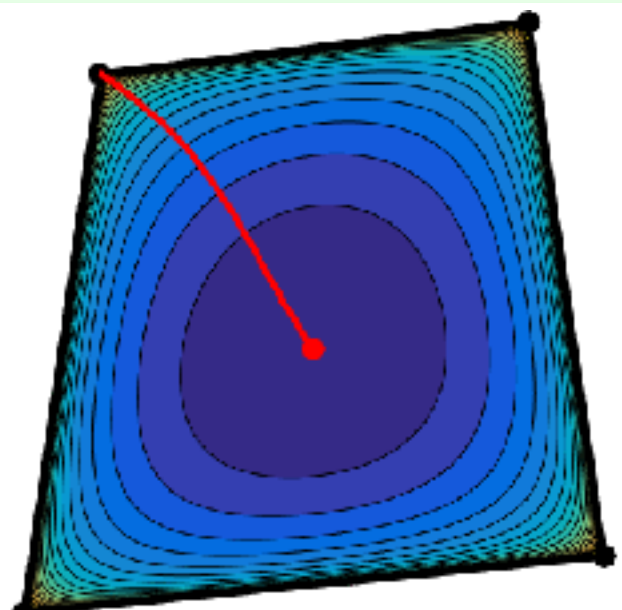
Log-barrier approximation:

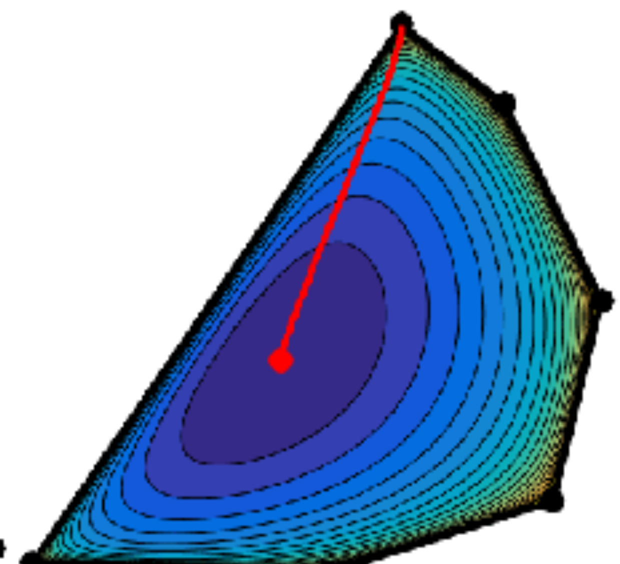
$$x_\varepsilon \overset{\text{def.}}{=} \underset{x}{\operatorname{argmin}} \ \langle x,\, c\rangle - \varepsilon \textstyle\sum_i \log(b_i - \langle a_i,\, x\rangle)$$

Interior point method:

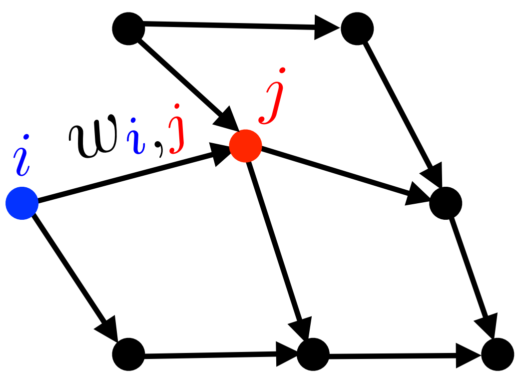
$O(\sqrt{m}\log(\frac{m}{\tau}))$ Newton iterations computes feasible $\hat{x}_\varepsilon$ with $\langle \hat{x}_\varepsilon - x_0,\, c\rangle \leqslant \tau$







Network Flow



Divergence on a graph:

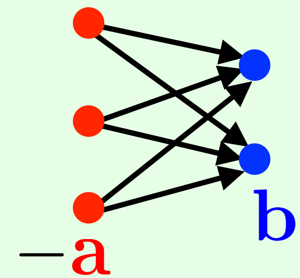
$$\mathrm{div}(\mathbf{s})_i \overset{\text{def.}}{=} \sum_{(i,k)\in G} \mathbf{s}_{i,k} - \sum_{(k,i)\in G} \mathbf{s}_{k,i}$$

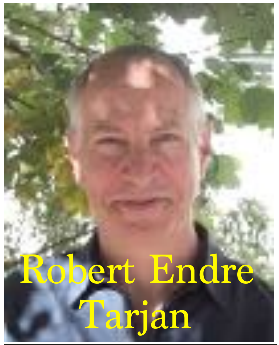
Min-cost flow:

$$\min_{\mathbf{s}\geqslant 0} \; \{\langle \mathbf{s},\, w\rangle \; ; \; \mathrm{div}(\mathbf{s}) = h\}$$

Optimal transport: bi-partite graph.

$$w_{i,j} = \mathbf{C}_{i,j} \qquad h = (-\mathbf{a}, \mathbf{b})$$





Theorem: on a graph with E edges and V vertices,

$\exists$ a network simplex algorithm of complexity

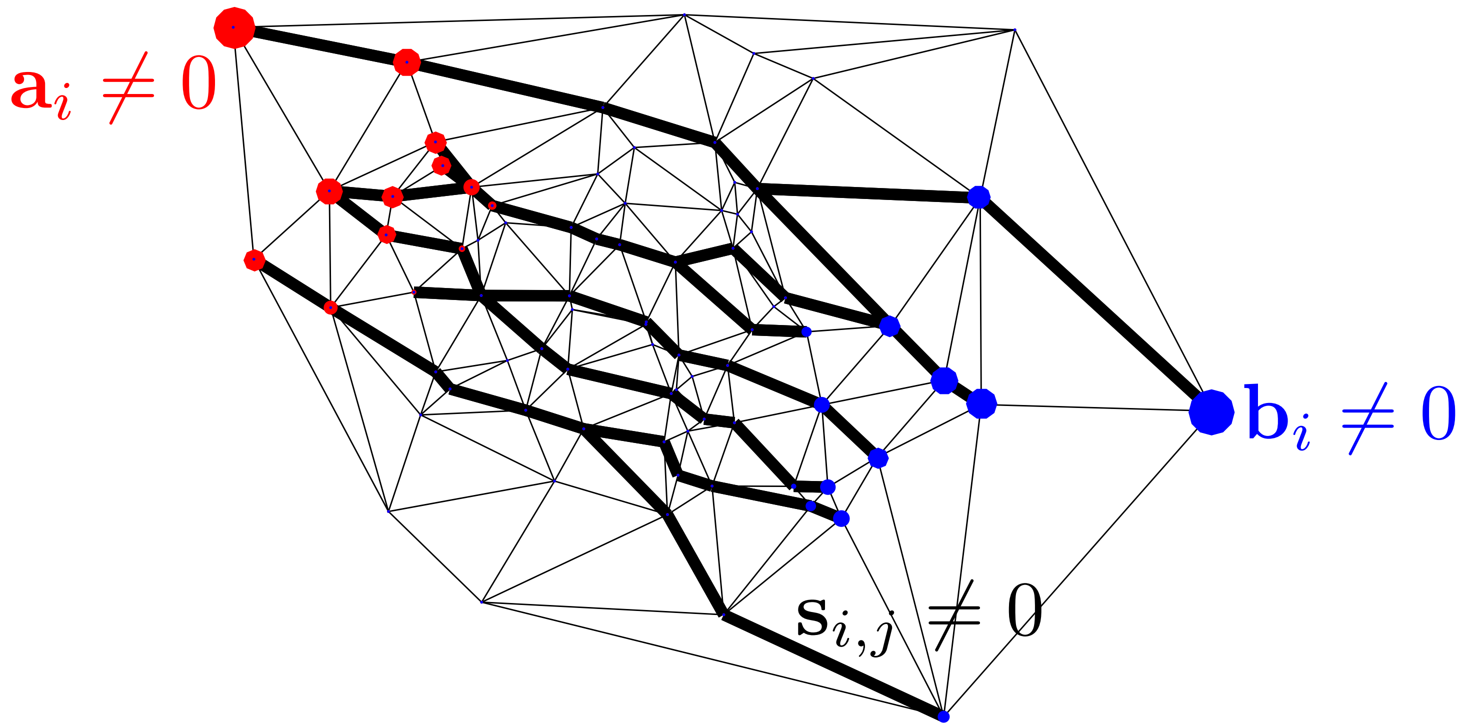
$O(VE\log V \log(V\|\mathbf{C}\|_\infty))$ if $\mathbf{C}_{i,j} \in \mathbb{Z}$.

OT simplex: $n = m$, complexity $O(n^3 \log(n)^2)$.

W1 as a Reduced Min-cost Flows

$$\mathbf{C}_{i,j} = \mathrm{GeodDist}_{\mathbf{w}}(i,j)$$

Proposition: [Beckmann]
$$\mathrm{W}_1(\mathbf{a},\mathbf{b}) = \min_{\mathbf{s}\in\mathbb{R}_+^{\mathcal{E}}} \left\{ \sum_{(i,j)\in\mathcal{E}} \mathbf{w}_{i,j}\mathbf{s}_{i,j} \;:\; \mathrm{div}(\mathbf{s}) = \mathbf{a}-\mathbf{b} \right\}$$



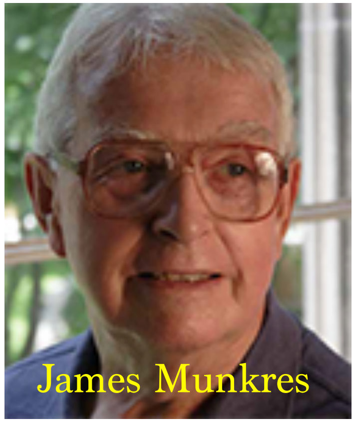
Network simplex, $E, V = O(n)$ (e.g. regular graph):

W_p in $O(n^3\log(n)^2)$ $\longrightarrow$ W_1 in $O(n^2\log(n)^2)$

W1 as a Reduced Min-cost Flows

$$d = \| \cdot \|, p = 1 : W_1(\mu, \nu) = \min_{\mathrm{div}(v) = \mu - \nu} \int \|u(x)\| \mathrm{d}x \quad \to \text{max-flow.}$$

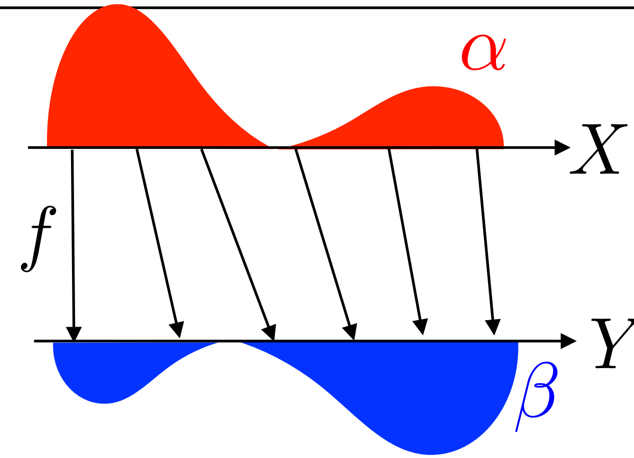
Hungarian and Auction Algorithms



Overview

- Linear Programming
- **PDE-based**
- Semi-discrete (dedicated part)
- Sinkhorn (dedicated part)

Monge-Ampère equation



$$\min_{\beta = f_\sharp \alpha} \int_X \|x - f(x)\|^2 \mathrm{d}\alpha(x)$$

Densities: $\frac{\mathrm{d}\alpha}{\mathrm{d}x} = \rho_\alpha, \frac{\mathrm{d}\beta}{\mathrm{d}x} = \rho_\beta$

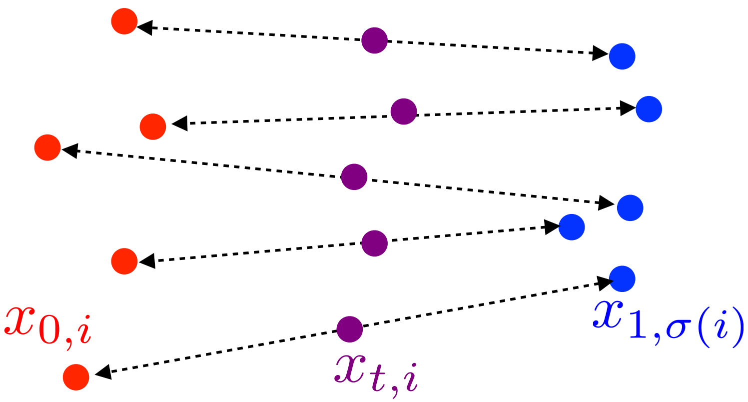
Theorem: [Brenier] Unique $f = \nabla\varphi$ solving

$$\det(\partial^2\varphi(x))\rho_\beta(\nabla\varphi(x)) = \rho_\alpha(x) \qquad \varphi \text{ convex}$$

$\rightarrow$ Finite-elements / finite-differences discretization of the cone of convex functions.

$\rightarrow$ non-classical boundary conditions.

Displacement Interpolation



Optimal assignement: $\min_{\sigma} \|x_0 - x_1 \circ \sigma\|$

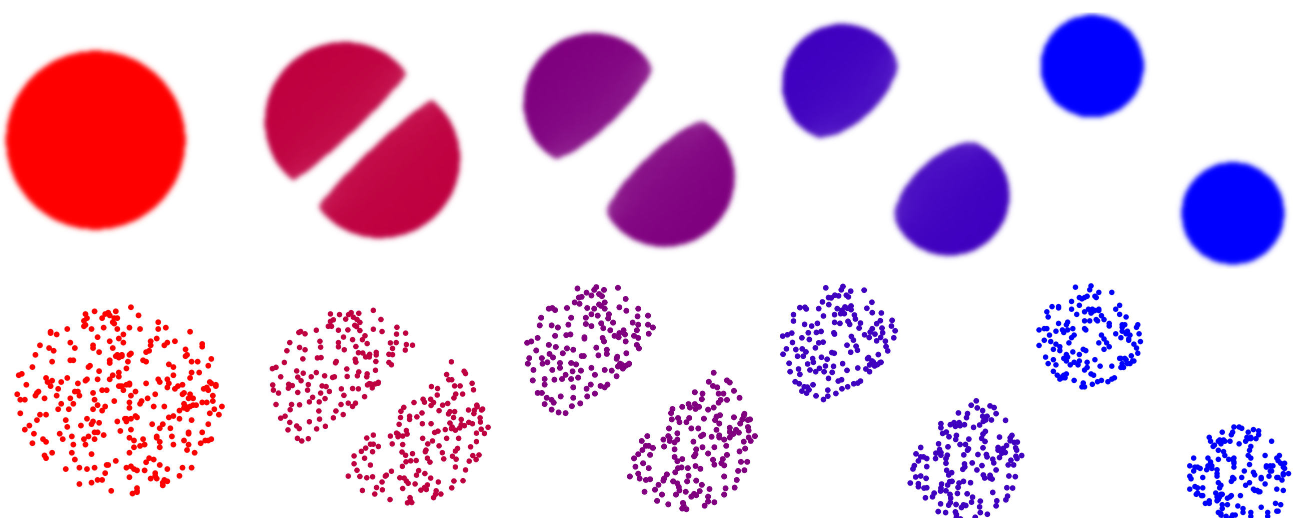
Displacement interpolation: $\alpha_t \overset{\text{def.}}{=} \frac{1}{n} \sum_i \delta_{x_{t,i}}$

$x_t = (1-t) x_0 + t x_1 \circ \sigma$

Monge map $\psi_\sharp \alpha = \beta$:

$$\alpha_t \overset{\text{def.}}{=} ((1-t)\mathrm{Id} + t\psi)_\sharp \alpha = (t\mathrm{Id} + (1-t)\psi^{-1})_\sharp \beta$$

Optimal coupling $\pi \in \mathcal{U}(\alpha, \beta)$: $\alpha_t \overset{\text{def.}}{=} ((1-t)P_{\mathcal{X}} + tP_{\mathcal{Y}})_\sharp \pi$



$t = 0$ $t = 1/2$ $t = 1$

Benamou-Brenier

Overview

- Linear Programming
- PDE-based
- **Semi-discrete (dedicated part)**
- Sinkhorn (dedicated part)

Overview

- Linear Programming
- PDE-based
- Semi-discrete (dedicated part)
- **Sinkhorn (dedicated part)**

Conclusion: Toward High-dimensional OT

Monge Kantorovich Dantzig Brenier Otto McCann Villani