Convergence of an entropic Frank-Wolfe method

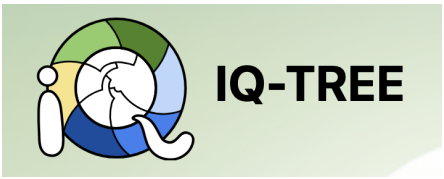
Nonlinear Approximation for High Dimensional Problems 2025

James Ashton Nichols (james.ashton.nichols@gmail.com)

July 3, 2025

Australian National University

Biological problems - IQ-Tree and COVID Variants



- Main "Phylogenetic inference" software.
- Used to distinguish COVID variants (Delta, Omicron...).
- Developed at ANU and Vienna University

Input:

```
7 28
Frog        AAATTTGGTCCTGTGATTCAGCAGTGAT
Turtle      CTTCCACACCCCAGGACTCAGCAGTGAT
Bird        CTACCACACCCCAGGACTCAGCAGTAAT
Human       CTACCACACCCCAGGAAACAGCAGTGAT
Cow         CTACCACACCCCAGGAAACAGCAGTGAC
Whale       CTACCACGCCCCAGGACACAGCAGTGAT
Mouse       CTACCACACCCCAGGACTCAGCAGTGAT
```

Output:

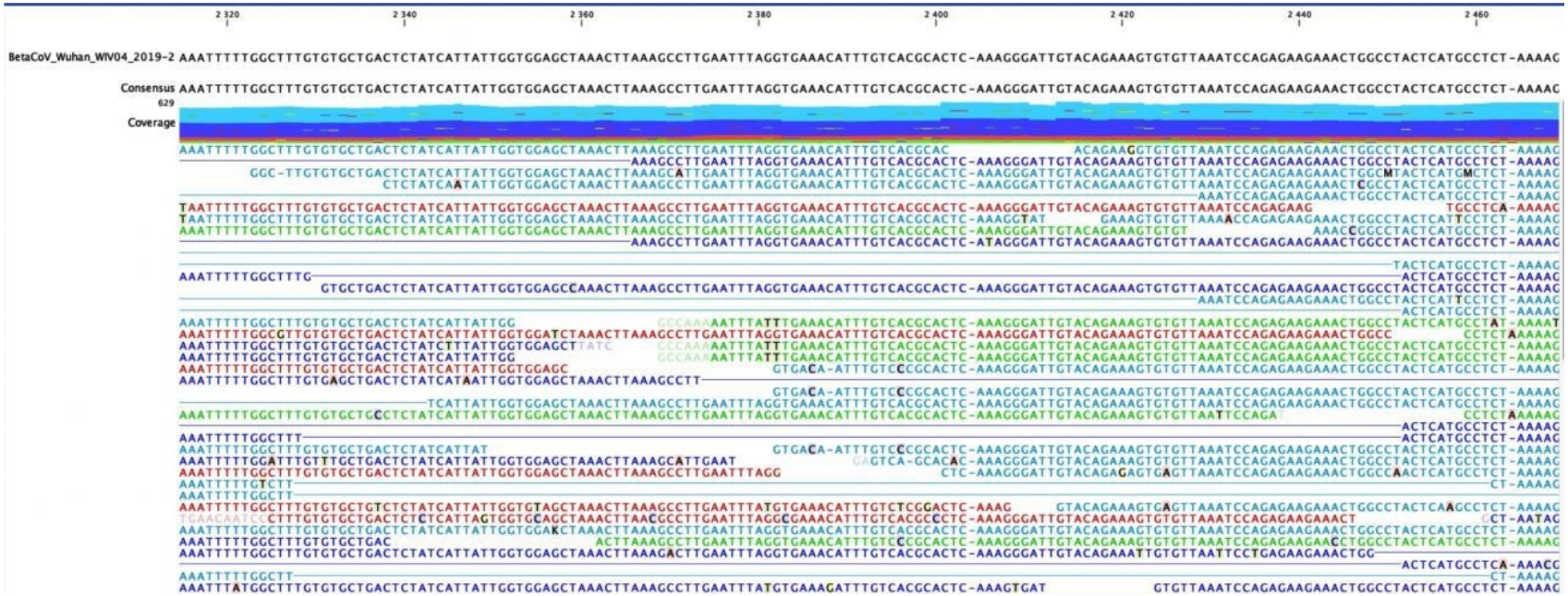
```
|
|      +--------------------Frog
+------|
       |                +-----------------Turtle
       |          +-----|
       |          |     |      +-----------------------Sphenodon
       |          |     |   +--|
       |          |     |   |  +--------------------------Lizard
       |          |     +---|
       |          |         |      +---------------------Crocodile
       |          |         +------|
       |          |                +------------------Bird
       +----------|
                  |                  +----------------Human
                  |               +--|
                  |               |  |  +--------Seal
                  |               |  +--|
                  |               |     |   +-------Cow
                  |               |     +---|
```

Biological problems - IQ-tree and COVID variants

- IQ-Tree ver1: 37 hours to infer a tree for 17,000 COVID sequences
- IQ-Tree ver2 (published May 2020): reduced this to three minutes
- At pandemic peak, 10,000 sequences were published per day

COVID genome as sequence at Institute L. Pasteur, January 24, 2020

$\sim 10^4$ base pairs



The Frank-Wolfe Method

[FRANK & WOLFE, 1965]

To find $\min_{x \in K} f(x)$ where $K \subset \mathbb{R}^d$ is compact and **convex**, F.W. takes an **iterative** approach based on **gradients**

The Frank-Wolfe Method

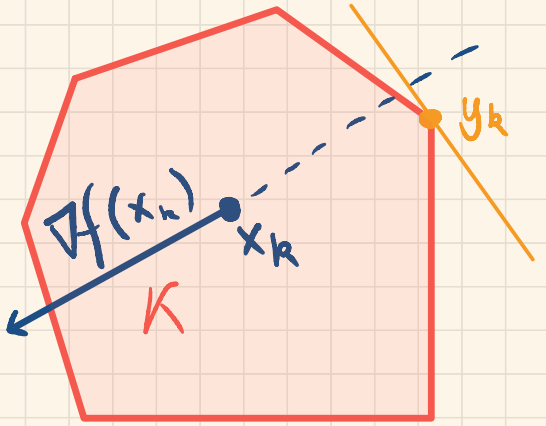
[FRANK & WOLFE, 1965]

To find $\min_{x \in K} f(x)$ where $K \subset \mathbb{R}^d$ is compact and **convex**, F.W. takes an **iterative** approach based on **gradients**

ALGORITHM (F.W.)

1. Given $x_k \in K$, find

$$y_k = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle$$



The Frank-Wolfe Method

[FRANK & WOLFE, 1965]

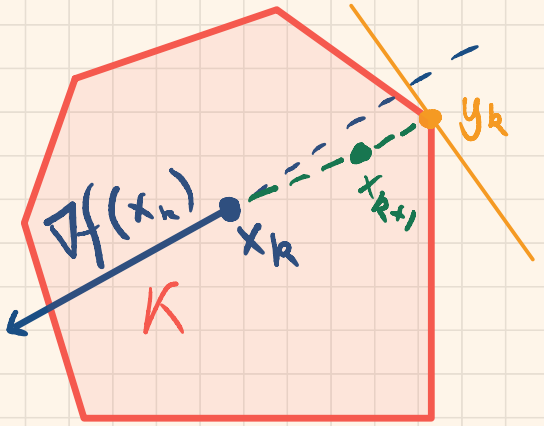
To find $\min_{x \in K} f(x)$ where $K \subset \mathbb{R}^d$ is compact and **convex**, F.W. takes an **iterative** approach based on **gradients**

ALGORITHM (F.W.)

1. Given $x_k \in K$, find

$$y_k = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle$$

2. Set $x_{k+1} = x_k + \gamma_k (y_k - x_k)$



The Frank-Wolfe Method

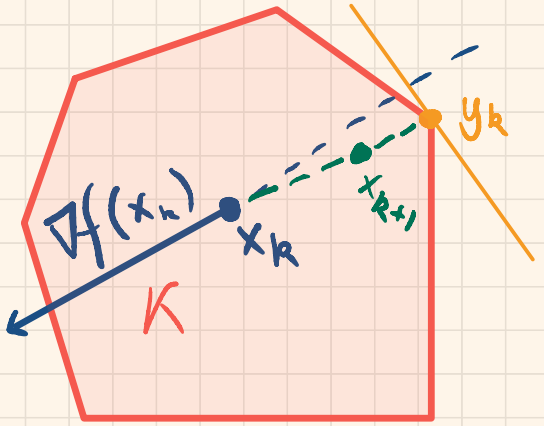
[FRANK & WOLFE, 1965]

To find $\min_{x \in K} f(x)$ where $K \subset \mathbb{R}^d$ is compact and **convex**, F.W. takes an **iterative** approach based on **gradients**

ALGORITHM (F.W.)

1. Given $x_k \in K$, find

$$y_k = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle$$



2. Set $x_{k+1} = x_k + \gamma_k (y_k - x_k)$

where OPTION 1: $\gamma_k = 2/k+2$

OPTION 2 (LINE-SEARCH): γ_k minimises $f(x_k + \gamma(y_k - x_k))$

Linear Minimisation

One drawback of FW: we must solve

$$\operatorname*{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle$$

- If K is a convex polytope, we use linear programming, $O(d^3)$ complexity in $\mathbb{R}^d$
- Otherwise we seek an approximate linear solver

Regularising with entropy

Assume now we're dealing with **probability vectors**, ie. $K \subseteq \Sigma_d := \{ x \in [0,1]^d : \sum_{i=1}^{d} x_i = 1 \}$

We have an **approximate linear program**

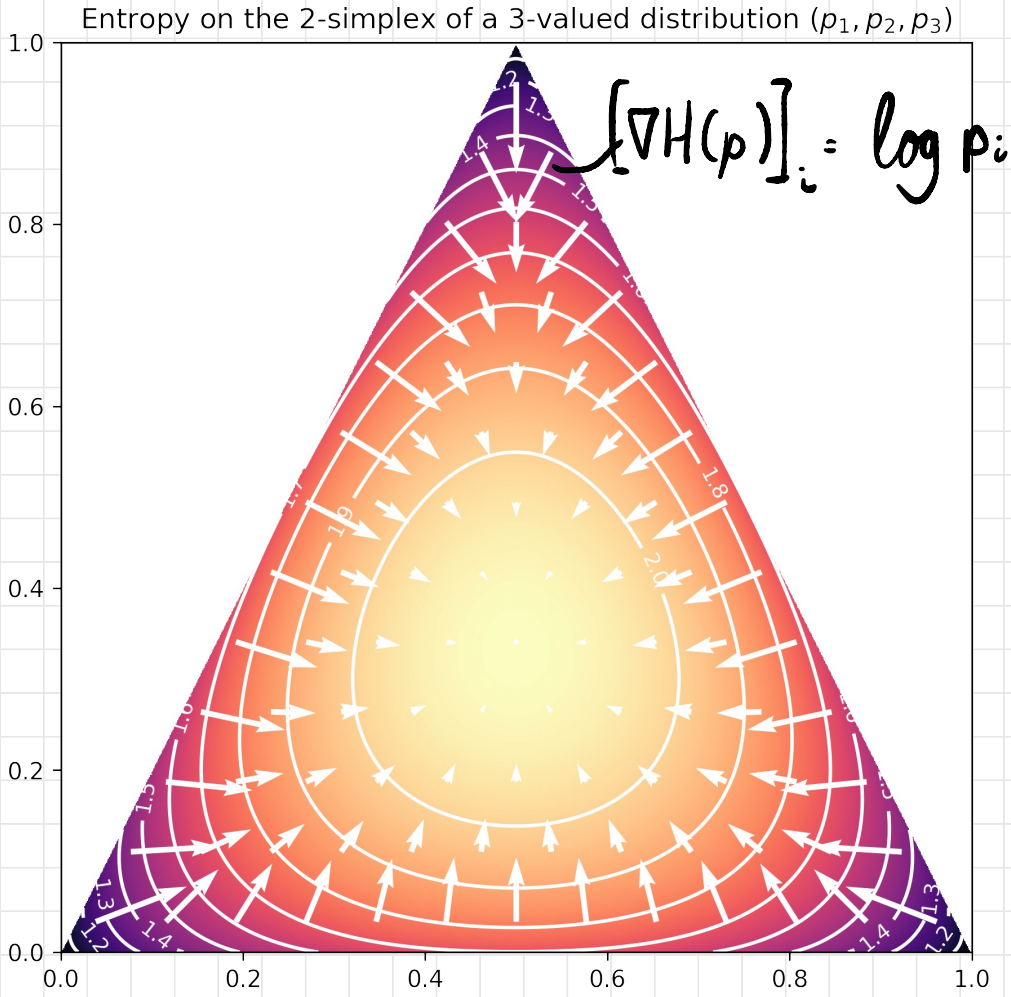
$$\min_{x \in K} \langle x, c \rangle \quad \approx \quad \min_{x \in K} \langle x, c \rangle - \varepsilon H(x)$$

where $\varepsilon > 0$ is small and **entropy** H is

$$H(x) := -\sum_{i=1}^{d} x_i \log x_i$$

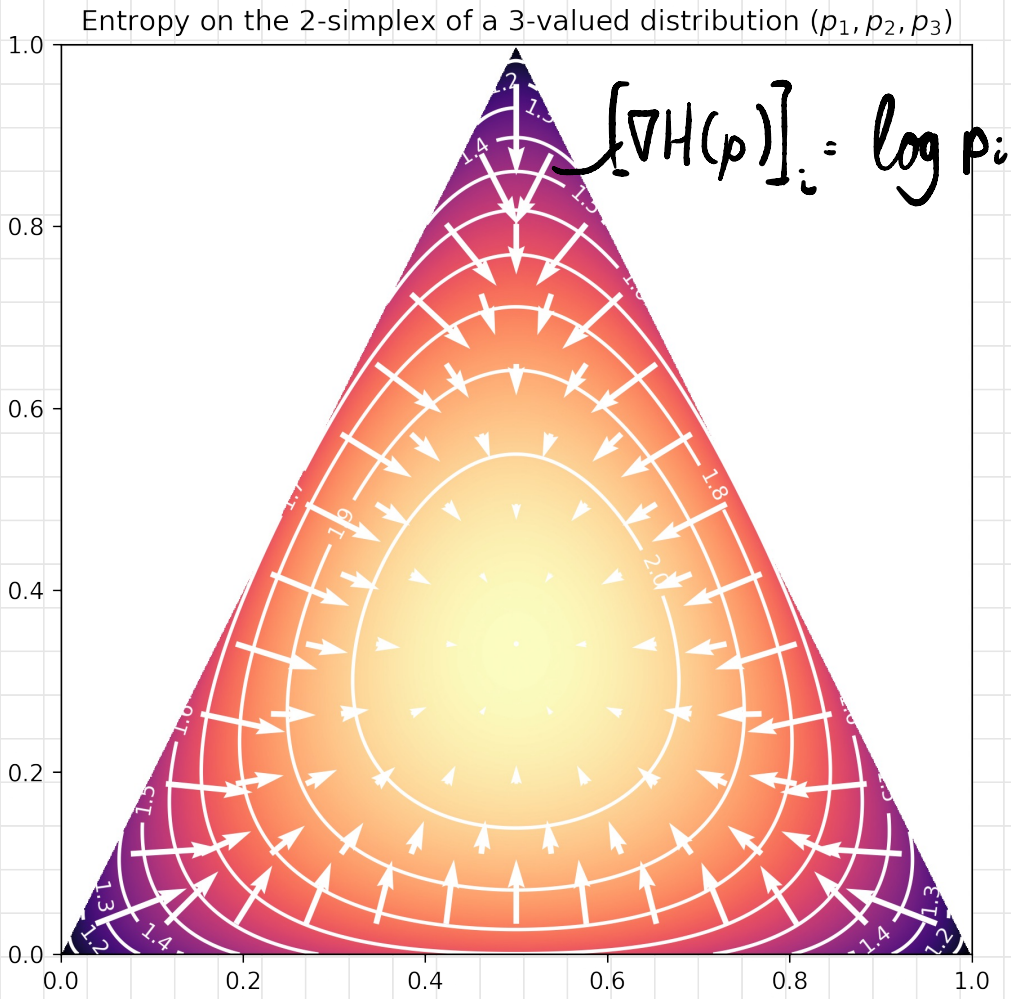
∇H Illustrated

- $\nabla H : \Sigma_d \to \mathbb{R}^{d-1}$ is a bijection
- $\nabla H(x) = c/\varepsilon$ always has a solution
- $\| \nabla H(x) \| \to \infty$ on $\partial \Sigma_d$



∇H Illustrated

- $\nabla H: K \to \mathbb{R}^{\dim(K)}$ is a bijection
- $\nabla H(x) = c/\varepsilon$ always has a solution
- $\| \nabla H(x) \| \to \infty$ on ∂K



Soft-max & Duality

Log-sum-exp "Soft-max"

$$h(c) := \log \sum_{i=1}^{d} e^{c_i}$$

$$[\nabla h(c)]_i = \frac{e^{c_i}}{\sum_{i=1}^{d} e^{c_i}} = e^{c_i - h(c)}$$

```
c = np.array([0.5, 1, 14, 2])
log_sum_exp = np.log(np.exp(c).sum())
softmax = np.exp(c - log_sum_exp)
print(f'c = {c}'); print(f'h(c) = {log_sum_exp:0.5f}'); print(f'nab_h(c) = {softmax}')
```

✓ 0.0s

```
c = [ 0.5  1.  14.   2. ]
h(c) = 14.00001
nab_h(c) = [1.3709e-06 2.2603e-06 9.9999e-01 6.1442e-06]
```

Note: $\nabla h(c) \in \Sigma_d$ for any $c \in \mathbb{R}^d$

Soft-max & Duality

Log-sum-exp

$$h(c) := \log \sum_{i=1}^{d} e^{c_i}$$

"Soft-max"

$$[\nabla h(c)]_i = \frac{e^{c_i}}{\sum_{i=1}^{d} e^{c_i}} = e^{c_i - h(c)}$$

Log-sum-exp is **convex dual** to entropy

(FENCHEL-LEGENDRE)

$$h^*(x) := \sup_{c \in \mathbb{R}^d} \langle x, c \rangle - h(c) = -H(x)$$

[BOYD & VANDENBERGHE, 2004]

Soft-max & Duality

Log-sum-exp
"Soft-max"

$$h(c) := \log \sum_{i=1}^{d} e^{c_i}$$

$$[\nabla h(c)]_i = \frac{e^{c_i}}{\sum_{i=1}^{d} e^{c_i}} = e^{c_i - h(c)}$$

Log-sum-exp is **convex dual** to entropy
(FENCHEL-LEGENDRE)

$$h^*(x) := \sup_{c \in \mathbb{R}^d} \langle x, c \rangle - h(c) = -H(x)$$

[BOYD & VANDENBERGHE, 2004]

And **gradients invert!**

$$\nabla h(\nabla h^*(x)) = x \qquad \nabla h^*(\nabla h(c)) = "c"$$

$$\nabla h : \mathbb{R}^d \to \Sigma_d \qquad \nabla h^* : \Sigma^d \to \mathbb{R}^d$$

Solving entropic LP

[ERLANDER, 1981]
[CLASEN, 1965]

$$x^*_{c/\varepsilon} = \operatorname*{argmin}_{x \in \Sigma_d} \langle x, c \rangle - \varepsilon H(x)$$

$$\Rightarrow \quad \nabla H(x^*_{c/\varepsilon}) = c/\varepsilon$$

$$\Rightarrow \quad x^*_{c/\varepsilon} = \nabla h(-c/\varepsilon) = e^{-c/\varepsilon - h(-c/\varepsilon)}$$

Solving entropic LP

[ERLANDER, 1981]
[CLASEN, 1965]

$$x^*_{c/\varepsilon} = \operatorname*{argmin}_{x \in \Sigma_d} \langle x, c \rangle - \varepsilon H(x)$$

$$\Rightarrow \quad \nabla H(x^*_{c/\varepsilon}) = c/\varepsilon$$

$$\Rightarrow \quad x^*_{c/\varepsilon} = \nabla h(-c/\varepsilon) = e^{-c/\varepsilon - h(-c/\varepsilon)}$$

When satisfying constraints it's a little trickier,

Lagrangian: $\mathcal{L}(x, \lambda) = \langle x, c \rangle - \varepsilon H(x) + \langle Ax - b, \lambda \rangle$

$$\nabla_\lambda \mathcal{L} = 0 \Rightarrow x^*_{c/\varepsilon} = e^{-c/\varepsilon + A^T\lambda - h(-c/\varepsilon + A^T\lambda)}$$

$x \in K := \{x \in \Sigma_d : Ax = b\}$

Solving entropic LP

[ERLANDER, 1981]
[CLASEN, 1965]

$$X^*_{c/\varepsilon} = \operatorname*{argmin}_{x \in \Sigma_d} \langle x, c \rangle - \varepsilon H(x)$$

$$\Rightarrow \quad \nabla H(x^*_{c/\varepsilon}) = c/\varepsilon$$

$$\Rightarrow \quad x^*_{c/\varepsilon} = \nabla h(-c/\varepsilon) = e^{-c/\varepsilon - h(-c/\varepsilon)}$$

When satisfying constraints it's a little trickier.

Lagrangian: $\mathcal{L}(x, \lambda) = \langle x, c \rangle - \varepsilon H(x) + \langle Ax - b, \lambda \rangle$

$$\nabla_\lambda \mathcal{L} = 0 \Rightarrow x^*_{c/\varepsilon} = e^{-c/\varepsilon + A^T\lambda - h(-c/\varepsilon + A^T\lambda)}$$

$K := \{x \in \Sigma_d : Ax = b\}$

How do we find λ?

Satisfying constraints

$K := \{ x \in \Sigma_d : Ax = b \}$

$$x^*_{c/\varepsilon} = \operatorname*{argmin}_{x \in K} \langle x, c \rangle - \varepsilon H(x) = e^{-c/\varepsilon + A^T\lambda - h(-c/\varepsilon + A^T\lambda)}$$

- "Newton-Kantorovich": [ERLANDER, 1981]
  Solve $A e^{-c/\varepsilon + A^T\lambda - h(-c/\varepsilon + A^T\lambda)} = b$ for λ with Newton's method
- Sinkhorn's method: [SINKHORN, 1965] [CUTURI, 2013]
  If $A_{ij} \in \{0, 1\}$ we can scale $x^*_{c/\varepsilon}$ in marginals, iteratively setting each λ_j in a loop.

A RELATIONSHIP BETWEEN ARBITRARY POSITIVE MATRICES AND DOUBLY STOCHASTIC MATRICES

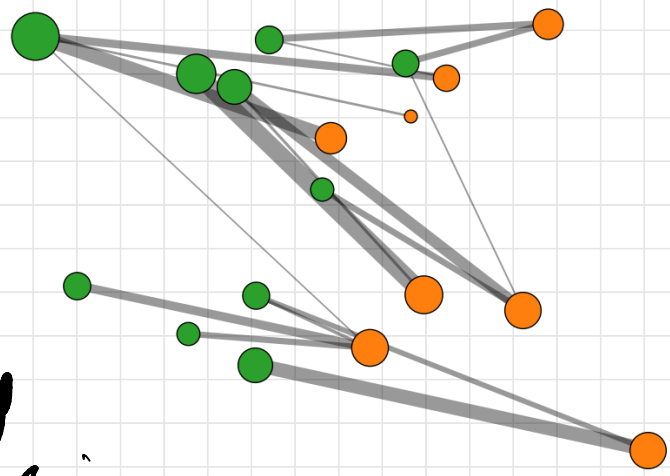
By Richard Sinkhorn

University of Houston

1. Introduction. Suppose one observes n transitions of a Markov chain with N states and stochastic matrix $P = (p_{ij})$. The usual estimate of p_{ij} is $t_{ij} = a_{ij}/\lambda_i$ where a_{ij} is the number of transitions from i to j which are observed, and $\lambda_i = \sum_j a_{ij}$. (Cf. [1].) This amounts to a normalization of the rows of $A = (a_{ij})$, and can be expressed as a matrix equation $T = D_1 A$ where $T = (t_{ij})$ and $D_1 = \operatorname{diag}[\lambda_1^{-1}, \cdots, \lambda_N^{-1}]$.

Solving O.T.

$$\min_{P \in U(\alpha, \beta)} \langle P, C \rangle_F$$



is a linear programming problem:

- ☑ Objective is linear
- ☑ Constraint set $U(\alpha, \beta)$ is a linear polytope

[KANTOROVIC, 1939]

But LP solvers will be $\sim O((mn)^3)$!

Entropic optimal transport

Sinkhorn Distances:
Lightspeed Computation of Optimal Transport

[CUTURI, 2013]

Marco Cuturi

$$\min_{P \in U(\alpha, \beta)} \langle P, C \rangle_F - \varepsilon H(P)$$

Entropic optimal transport

Sinkhorn Distances:
Lightspeed Computation of Optimal Transport

[CUTURI, 2013]

Marco Cuturi

$$\max_{\substack{\lambda \in \mathbb{R}^n \\ \theta \in \mathbb{R}^m}} \min_{P \in \mathbb{R}^{n \times m}} \underbrace{\langle P, C \rangle_F - \varepsilon H(P) - \langle P \mathbb{1}_m - \alpha, \lambda \rangle - \langle P^T \mathbb{1}_n - \beta, \theta \rangle}_{= \mathcal{L}(P, \lambda, \theta)}$$

Entropic optimal transport

Sinkhorn Distances:
Lightspeed Computation of Optimal Transport

[CUTURI, 2013]

Marco Cuturi

$$\max_{\substack{\lambda \in \mathbb{R}^n \\ \theta \in \mathbb{R}^m}} \min_{P \in \mathbb{R}^{n \times m}} \underbrace{\langle P, C \rangle_F - \varepsilon H(P) - \langle P \mathbb{1}_m - \alpha, \lambda \rangle - \langle P^T \mathbb{1}_n - \beta, \theta \rangle}_{= \mathcal{L}(P, \lambda, \theta)}$$

- Critical point $\nabla_{\lambda,\theta} \mathcal{L}(P, \lambda, \theta) = 0 \Rightarrow P_{ij} = \underbrace{e^{\lambda_i/\varepsilon}}_{u} \underbrace{e^{-C_{ij}/\varepsilon}}_{K_\varepsilon} \underbrace{e^{\theta_j/\varepsilon}}_{v}$

Entropic optimal transport

Sinkhorn Distances:
Lightspeed Computation of Optimal Transport

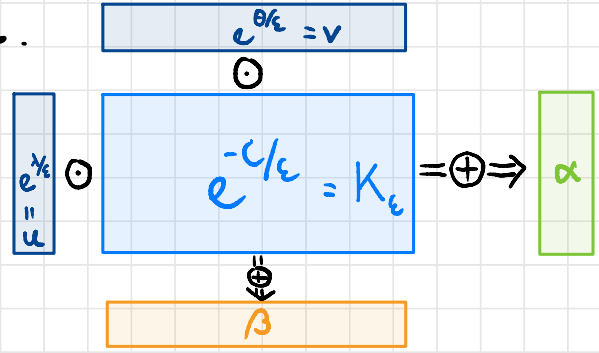
[CUTURI, 2013]

Marco Cuturi

$$\max_{\substack{\lambda \in \mathbb{R}^n \\ \theta \in \mathbb{R}^m}} \min_{P \in \mathbb{R}^{n \times m}} \underbrace{\langle P, C \rangle_F - \varepsilon H(P) - \langle P\mathbb{1}_m - \alpha, \lambda \rangle - \langle P^T\mathbb{1}_n - \beta, \theta \rangle}_{= \mathcal{L}(P, \lambda, \theta)}$$

- Critical point $\nabla_{\lambda,\theta} \mathcal{L}(P, \lambda, \theta) = 0 \Rightarrow P_{ij} = e^{\lambda_i/\varepsilon} e^{-C_{ij}/\varepsilon} e^{\theta_j/\varepsilon}$, with $e^{\lambda_i/\varepsilon} = u$, $e^{-C_{ij}/\varepsilon} = K_\varepsilon$, $e^{\theta_j/\varepsilon} = v$
- To satisfy the marginals, $P\mathbb{1}_m = \alpha$ & $P^T\mathbb{1}_n = \beta$, we use **Sinkhorn's method**.

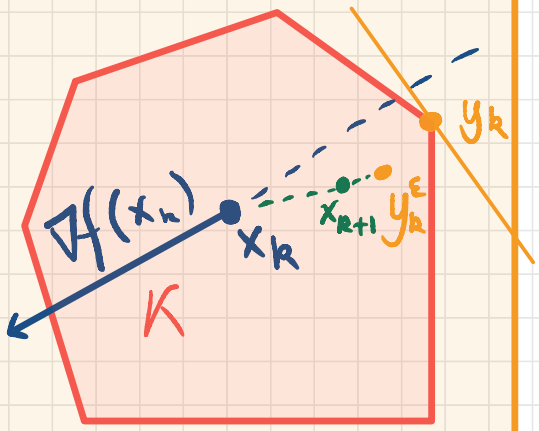
1. Set u s.t. $P\mathbb{1}_m = \alpha$
2. Set v s.t. $P^T\mathbb{1}_n = \beta$
3. Repeat. $P \to P_\varepsilon^*$



Entropic Frank-Wolfe

ALGORITHM [N. & STONE, 2025]

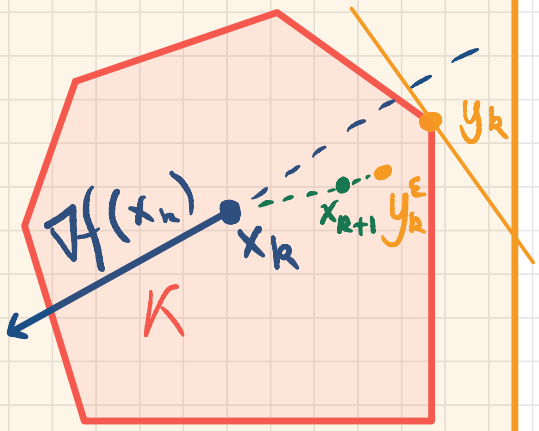
1. Solve $y_k^\varepsilon = \underset{y \in K}{\operatorname{argmin}} \langle y, \nabla f(x_k) \rangle - \varepsilon H(y)$



Entropic Frank-Wolfe

ALGORITHM [N. & STONE, 2025]

1. Solve $y_k^\varepsilon = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle - \varepsilon H(y)$
2. If $\langle y_k^\varepsilon - x_k, \nabla f(x_k) \rangle \geq 0$, halve ε, repeat step 1.

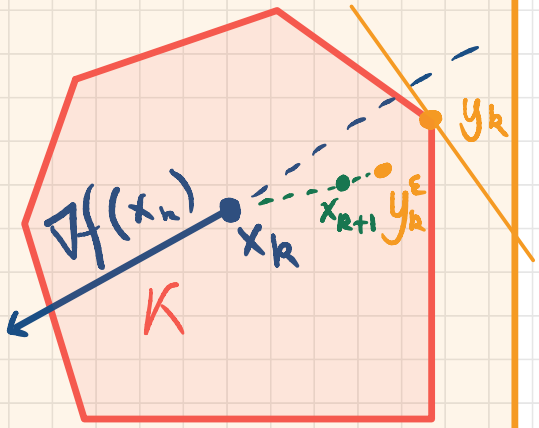


Entropic Frank-Wolfe

ALGORITHM [N. & STONE, 2025]

1. Solve $y_k^\varepsilon = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle - \varepsilon H(y)$
2. If $\langle y_k^\varepsilon - x_k, \nabla f(x_k) \rangle \geq 0$, halve ε, repeat step 1.
3. Set $x_{k+1} = x_k + \gamma_k (y_k^\varepsilon - x_k)$

   where γ_k minimises $f(x_k + \gamma(y_k^\varepsilon - x_k))$

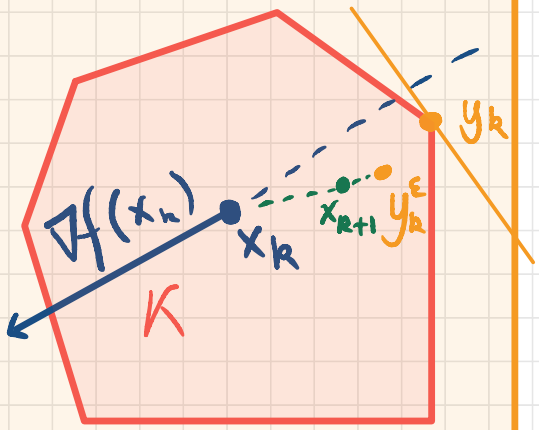


Entropic Frank-Wolfe

ALGORITHM [N. & STONE, 2025]

1. Solve $y_k^\varepsilon = \operatorname{argmin}_{y \in K} \langle y, \nabla f(x_k) \rangle - \varepsilon H(y)$
2. If $\langle y_k^\varepsilon - x_k, \nabla f(x_k) \rangle \geq 0$, halve ε, repeat step 1.
3. Set $x_{k+1} = x_k + \gamma_k (y_k^\varepsilon - x_k)$

   where γ_k minimises $f(x_k + \gamma(y_k^\varepsilon - x_k))$
4. Stop when $\langle \nabla f(x_k), y_k^\varepsilon - x_k \rangle \leq \eta$ $(\eta > 0$ STOPPING CRITERIA$)$



Error Bounds

LP: $x^* = \operatorname{argmin}_{x \in K} \langle x, c \rangle$

ENTROPIC LP $x^*_{c/\varepsilon} = \operatorname{argmin}_{x \in K} \langle x, c \rangle - \varepsilon H(x)$

We have

- $0 \leqslant \langle x^*, c \rangle - \langle x^*_{c/\varepsilon}, c \rangle \leqslant \varepsilon \left(H(x^*_{c/\varepsilon}) - H(x^*) \right)$

  [PEYRÉ & CUTURI, 2018]

- $\| x^* - x^*_{c/\varepsilon} \| \lesssim e^{-1/\varepsilon}$

  [COMINETTI & SAN MARTIN, 1994]

Convergence

① **FW:** f is **convex** then $f(x_k) - \min_{x \in K} f(x) \lesssim C_f / k$

[FRANK & WOLFE 1965] [CLARKSON 2010] [JAGGI 2013]

② **FW:** f is **non-convex** then

$$\min_{0 \le \ell < k} f(x_\ell) - f(x_{\ell+1}) \lesssim C_f / \sqrt{k}$$

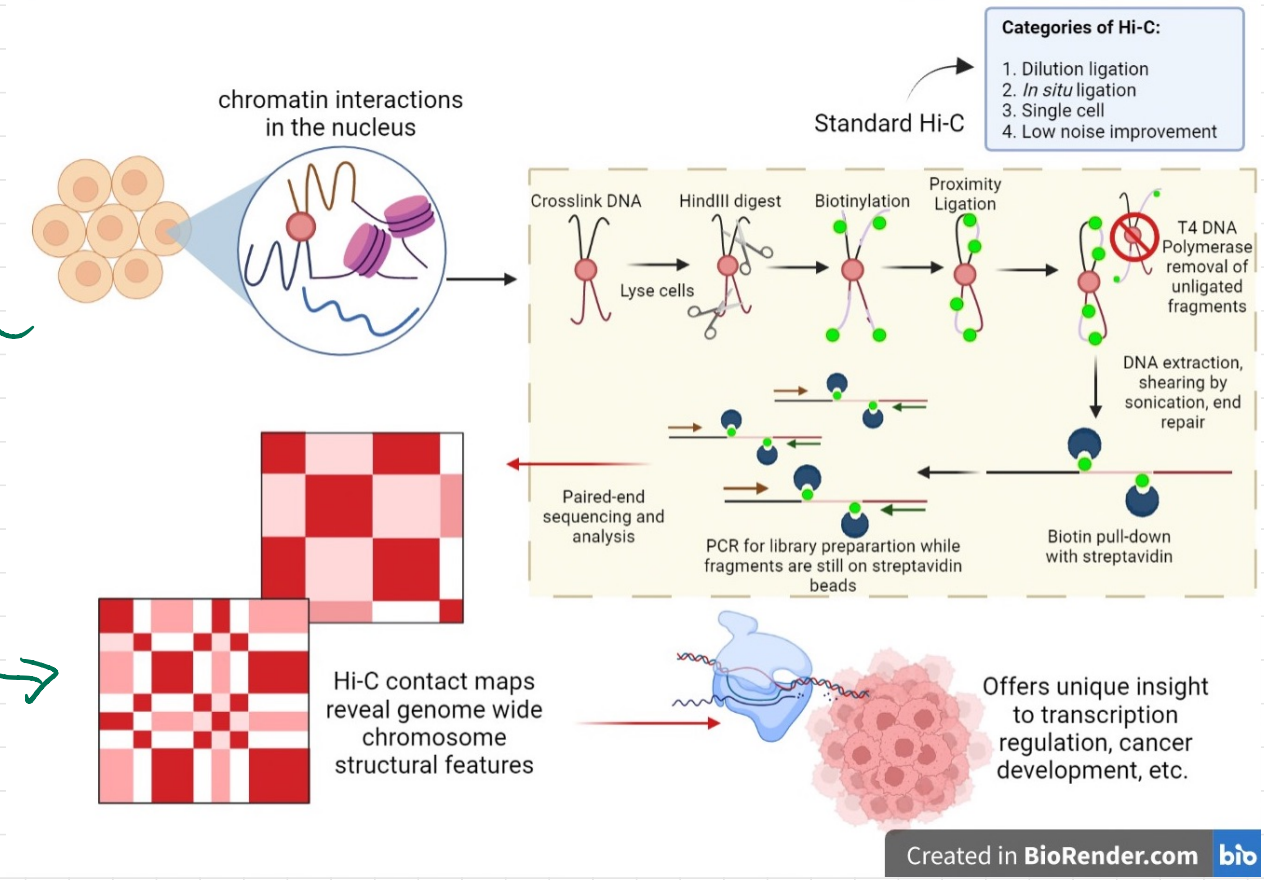
[LACOSTE-JULIEN 2013]

③ **ENTROPIC FW:** f is **non-convex** then

$$\min_{0 \le \ell < k} f(x_\ell) - f(x_{\ell+1}) \lesssim C_f / \sqrt{k}$$

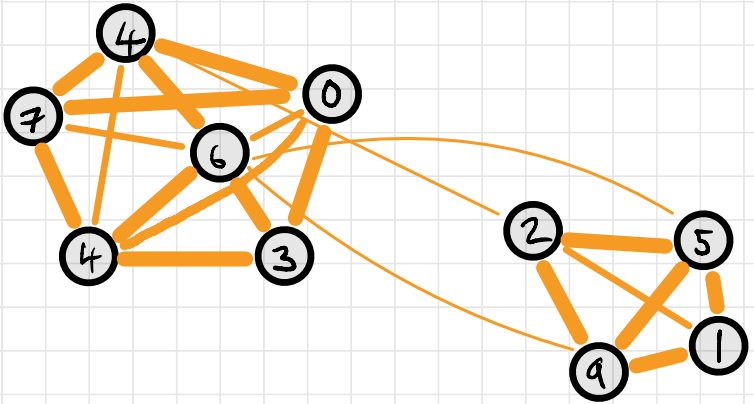
[N. & STONE, 2025]

An application

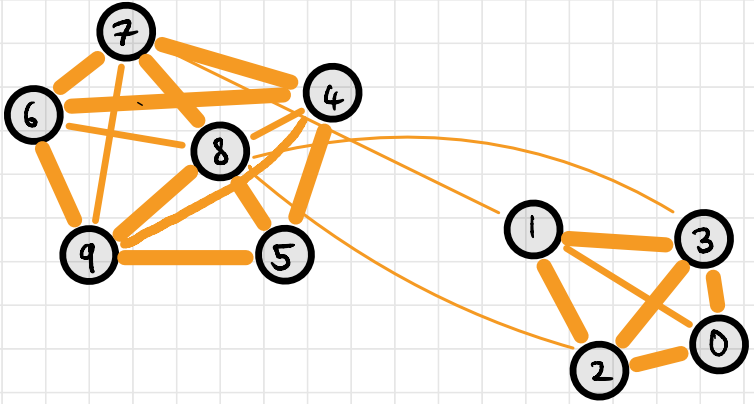


This is where I come in

Matrix Reordering



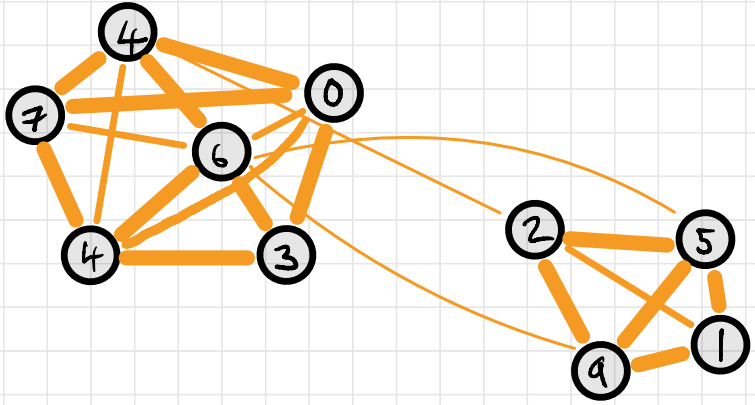
	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1.4	0	0	1.6	2.5	1.9	1.1
1	0	0	1.4	0	3.1	3	0	0	0	0
2	0	1.4	0	0	2.5	2.8	0	0	0	0.41
3	1.4	0	0	0	0	0	1.8	1.2	2.3	1.4
4	0	3.1	2.5	0	0	3.2	0.6	0	0	0
5	0	3	2.8	0	3.2	0	0.56	0	0	0
6	1.6	0	0	1.8	0.6	0.56	0	2.7	1.6	3
7	2.5	0	0	1.2	0	0	2.7	0	1.6	3.1
8	1.9	0	0	2.3	0	0	1.6	1.6	0	1.2
9	1.1	0	0.41	1.4	0	0	3	3.1	1.2	0

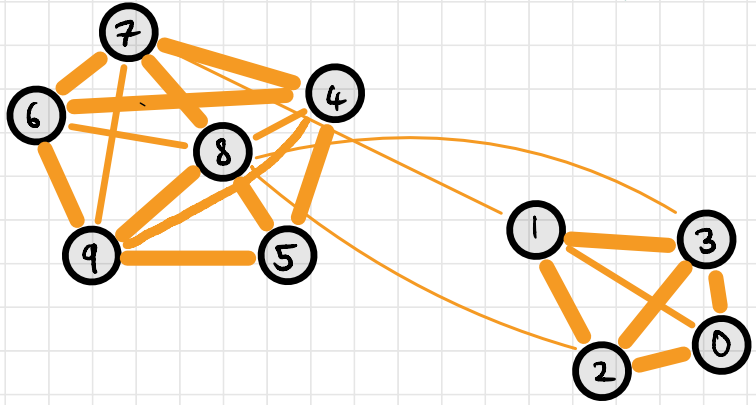


	0	1	2	3	4	5	6	7	8	9
0	0	1.4	3.1	3	0	0	0	0	0	0
1	1.4	0	2.5	2.8	0	0	0	0.41	0	0
2	3.1	2.5	0	3.2	0	0	0	0	0.6	0
3	3	2.8	3.2	0	0	0	0	0	0.56	0
4	0	0	0	0	0	1.4	2.5	1.1	1.6	1.9
5	0	0	0	0	1.4	0	1.2	1.4	1.8	2.3
6	0	0	0	0	2.5	1.2	0	3.1	2.7	1.6
7	0	0.41	0	0	1.1	1.4	3.1	0	3	1.2
8	0	0	0.6	0.56	1.6	1.8	2.7	3	0	1.6
9	0	0	0	0	1.9	2.3	1.6	1.2	1.6	0

Graph Sorting

Sorting a graph such that heavily connected nodes are **enumerated to be maximally adjacent** is the same as **permuting rows and columns** of the adjacency matrix to make it **maximally diagonal**. (A little like Cuthill - McKee)





Off-diagonal penalties

Consider a penalty matrix $C_{ij} = c(|i-j|)$,

eg. $C_{ij} = |i-j|$

$C \in \mathbb{R}^{5\times 5}$ looks like $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$.

Then, if $\langle A, C\rangle_F \leq \langle B, C\rangle_F$ we say A is more **diagonally dominant** than B.

Permutations

Let $\mathcal{P}_n$ denote the set of **permutation matrices**:

$$\mathcal{P}_n = \{P \in \{0,1\}^{n \times n} : \sum_i P_{ij} = \sum_j P_{ij} = 1\}$$

Given a permutation σ of P, $[PGP^T]_{ij} = G_{\sigma(i)\sigma(j)}$

So, our objective is to solve

$$\min_{P \in \mathcal{P}_n} \langle PGP^T, C \rangle_F = \min_{P \in \mathcal{P}_n} \sum_{i,j=1}^{n} [PGP^T]_{ij} C_{ij}$$

This is **NP-hard!**

Equivalent to
- Quadratic assignment **(QAP)**
- Travelling salesman **(TSP)**

[Koopman & Beckmann '57] [Yuille & Kosowsky '94]

Relaxation

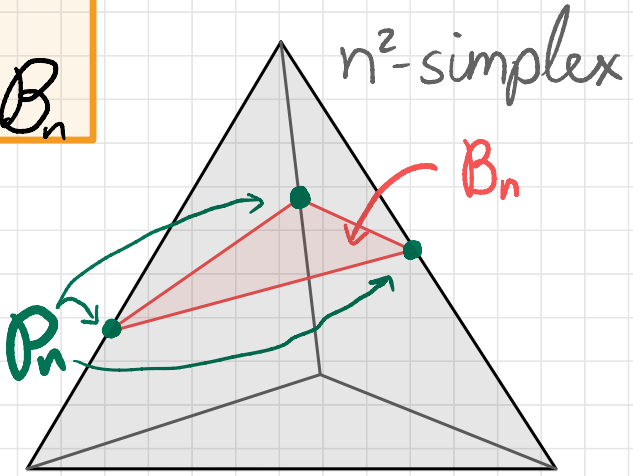
We propose the following relaxation:

Solve $\min_{P \in \mathcal{B}_n} \langle PGP^T, C \rangle_F$

where $\mathcal{B}_n = \{ P \in [0,1]^{n \times n} : \sum_i P_{ij} = \sum_j P_{ij} = 1 \}$ is the set of **bistochastic matrices**.

THEOREM (BIRKHOFF)
$\mathcal{P}_n$ is the set of extreme points of $\mathcal{B}_n$

$\mathcal{B}_n$ is a **convex polytope** and a subset of the **n^2-simplex** of "probability" vectors.



Entropic Frank-Wolfe!

We solve our problem w/ entropic Frank-Wolfe!

- $f(P) = \langle PGP^T, C \rangle_F$
- $\nabla_P(\langle PGP^T, C \rangle_F) = CPG^T + C^TPG = 2CPG$

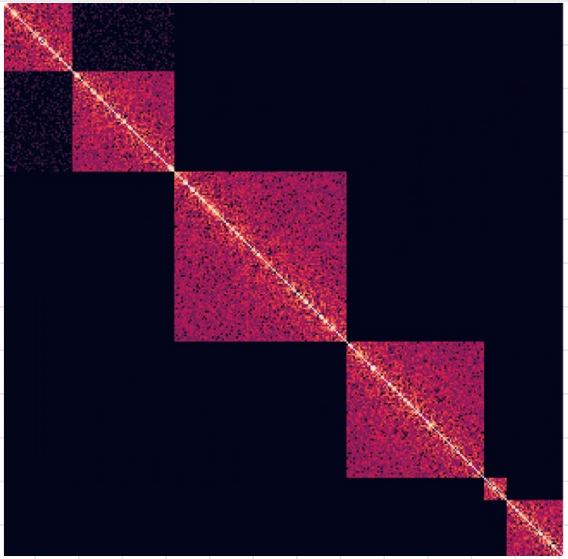
(As $C = C^T$ & $G = G^T$)

- $H(P) = -\sum_{i,j=1}^{n} P_{ij}(\log P_{ij} - 1)$
- $K = \mathcal{B}_n$

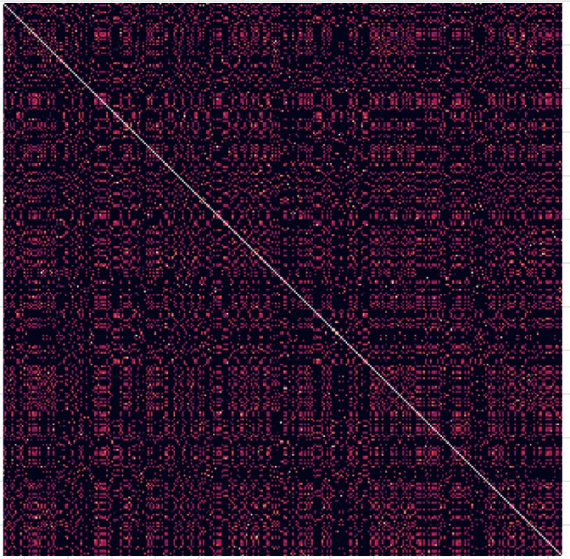
NOTE Our objective is **quadratic** and **non-convex**, we have no guarantee that $P^* \in \mathcal{P}_n \subset \mathcal{B}_n$

Some results

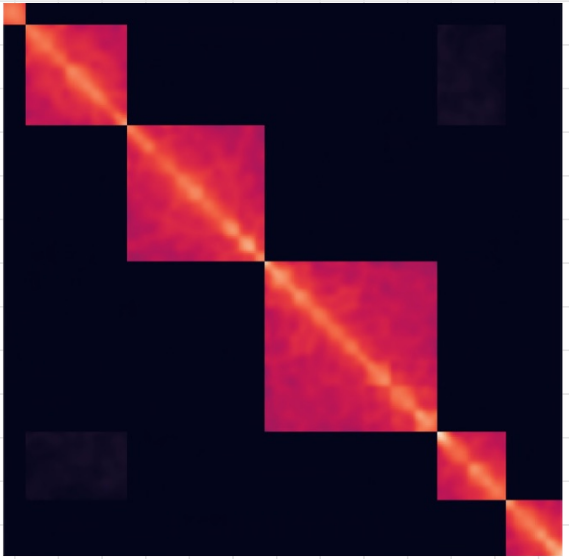
ORIGINAL RANDOM BLOCK MATRIX



SCRAMBLED MATRIX (600 × 600)



RECOVERED / ENTROPICALLY SORTED



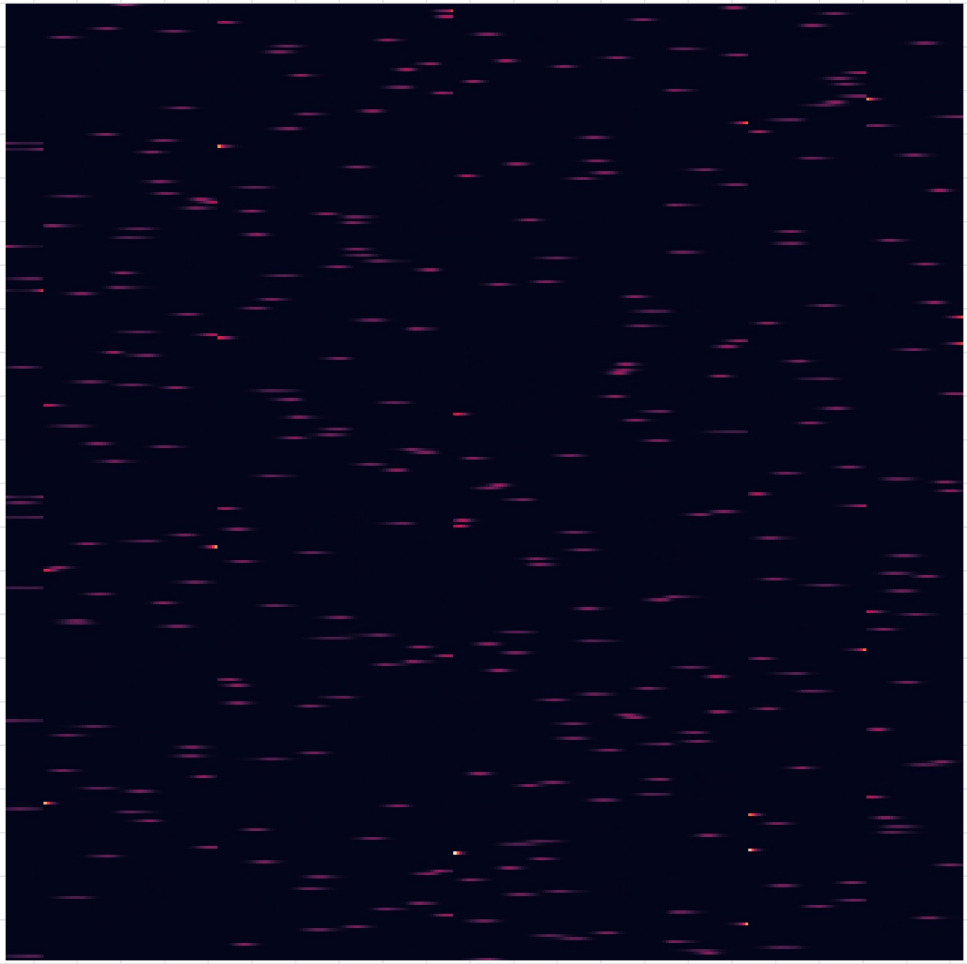
40 iterations

$\epsilon_0 = 2 \times 10^{-3}$

Some results

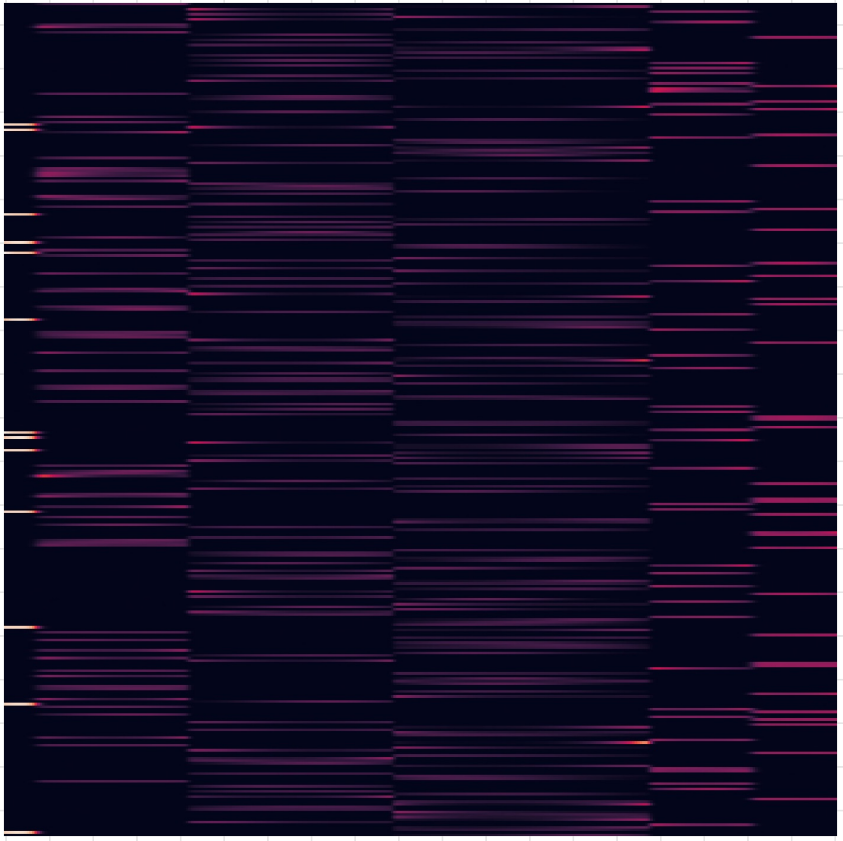
APPROXIMATE PERMUTATION (IN $\mathcal{B}_n$)

Can recover a real permutation from this

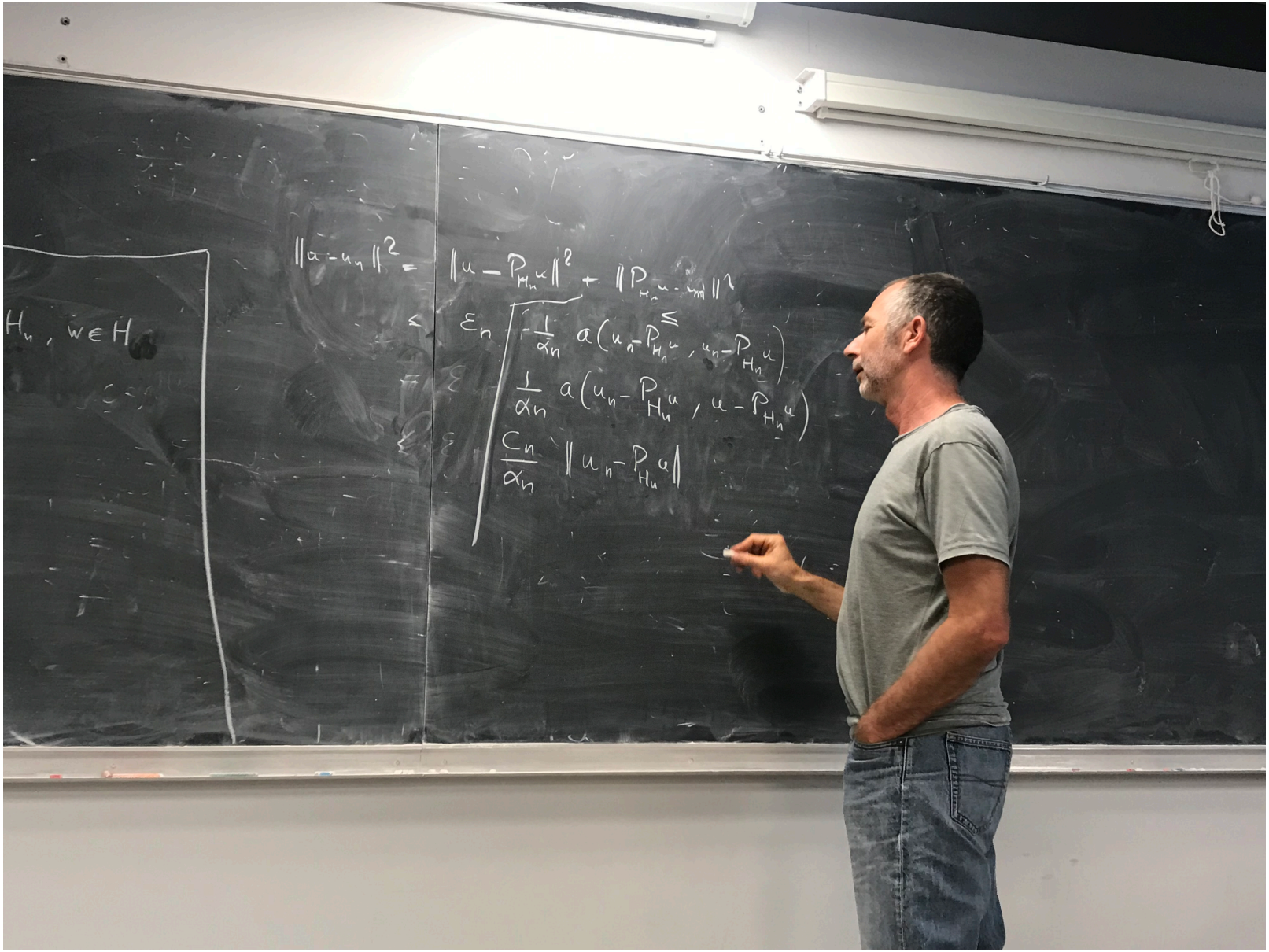


Only 3 iterations

Same 600 x 600 problem w/ 3 iterations



POV - You are about to have a great day



POV - You are witnessing a most elegant talk

Playing music

POV - You had a great three years in Paris with this guy