Projet Simulation et Monte Carlo
Travelling Salesman Problem (TSP)

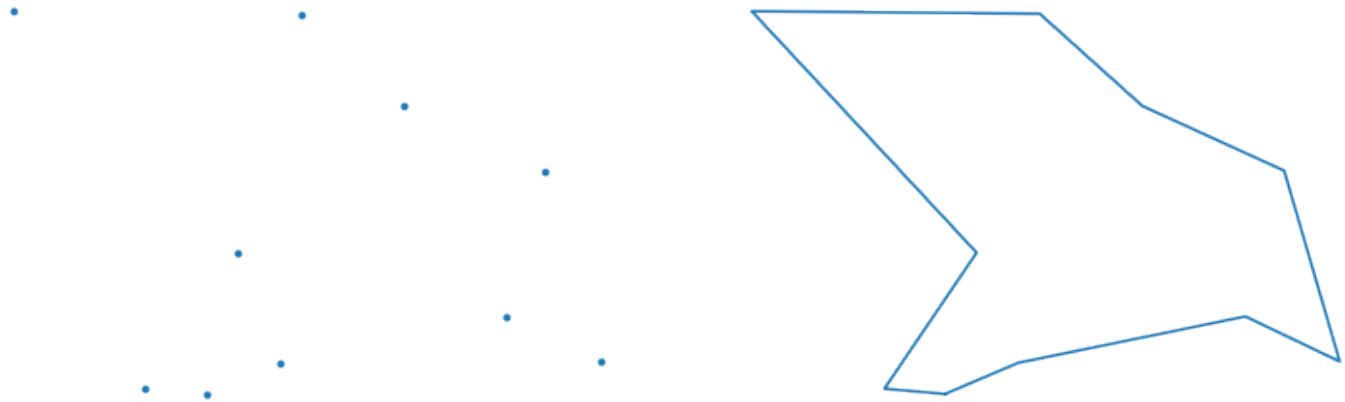
Maxime Blin, William Lambert, Dimitri Meunier

ENSAE ParisTech

Mai 2019

Problème TSP

Le problème du voyageur de commerce revient à minimiser le trajet effectué entre M villes, sous la contrainte de passer par chaque ville exactement une fois. C'est un problème d'optimisation combinatoire.



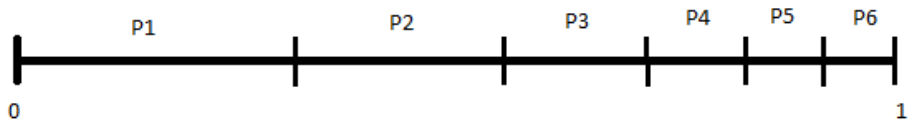
Essayer tous les trajets coûte $M!$. **Idée :** Simuler des 'tours' en explorant efficacement l'espace des solutions.

Simulation de N vecteurs selon la loi multinomiale

Chaque expérience de la loi multinomiale est simulée via la **méthode de la transformée inverse :**

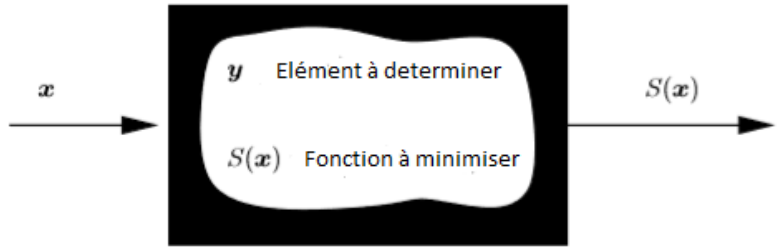
$$j = \min \left\{ j' \in \{1, \ldots, k\} : \left(\sum_{i=1}^{j'} p_i \right) - U \geq 0 \right\}$$

avec U une simulation de loi $\mathcal{U}[0,1]$. En répétant cette étape n fois et en rassemblant les issus on obtient une observation multinomiale de loi $\text{Mult}(n; p_1, \cdots, p_k)$.



Trier les probabilités par ordre décroissant permet d'améliorer la vitesse de calcul.

Méthode CE (Cross Entropy) pour l'optimisation



But : déterminer $\text{argmin}_{x \in \chi} S(x)$. On se place dans un modèle paramétrique $X \sim f(.\,, P)$:

- générer $X_i \sim f(.\,, P_0)$
- conserver les observations qui donnent les meilleurs scores (selon le quantile $q\%$ des scores)
- trouver le nouveau paramètre P_1 qui rend cet échantillon tronqué 'vraisemblable'

À la fin on estime la solution du problème par $x^* = E_{f(.\,,\hat{P})}[X]$.

Très souvent $f(.\,, \hat{P})$ est concentrée autour de son mode qui est notre solution (loi dégénérée).

Notations

- Les villes sont numérotées de 0 à M-1, 0 sera la ville de référence
- χ est l'ensemble des tours admissibles, $\chi = \{0, \sigma, 0\}$ avec $\sigma \in \Sigma_M$ l'ensemble des permutations de $\{1, \cdots, M-1\}$
- D est la matrice (symétrique) des distances entre les villes
- S est l'application qui à un tour associe sa longueur à partir des entrées de D

On souhaite donc minimiser S sur χ
Problèmes. Quelle est la famille paramétrique des tours ?
Comment les simuler ?

Simulation des tours

Soit P une matrice de transition de taille $M \times M$ dont la diagonale est nulle. P_{ij} est la probabilité de passer de la ville i à la ville j. Partant de la ville 0 le tour est généré de la façon suivante:

- On simule la ville suivante à partir des probabilités de la ligne correspondant à la ville actuelle (i) de la matrice P. C'est la réalisation d'une loi Mult$(1; p_{i,0}, \cdots, p_{i,M-1})$
- On met à 0 la colonne correspondant à la nouvelle ville et on renormalise les probabilités de la matrice
- On réitère jusqu'à obtention d'un tour complet

La loi de chaque tour est entièrement paramétrisée par P. Les tours sont issus de la famille paramétrique associée à une chaîne de Markov **sans remise** de matrice de transition P.

Application au problème TSP

On cherche à déterminer $\hat{P}$ telle que les tours simulés donne la distance la plus faible possible.

- Initialisation de la matrice de transition P^0 (uniforme)
- Simulation de N tours selon P^t
- On met à jour P à partir des $q\%$ des meilleurs tours. P_{ij}^{t+1} = proportion des meilleurs tours où la transition ij apparaît
- Répéter jusqu'au critère d'arrêt (5 derniers quantiles égaux)

Problème. Si la transition ij n'est pas visitée par les N tours, P_{ij}^t bloque à 0 et l'exploration de l'ensemble des solutions est mauvaise.

Solution 1 : Smooth Update

$$P_{ij}^{t+1} = \alpha Q_{ij}^{t} + (1 - \alpha) P_{ij}^{t}$$

avec Q_{ij}^{t} la proportion des meilleurs tours où la transition ij apparaît.

Exemple pour 10 villes, convergence rapide vers une matrice dégénérée :

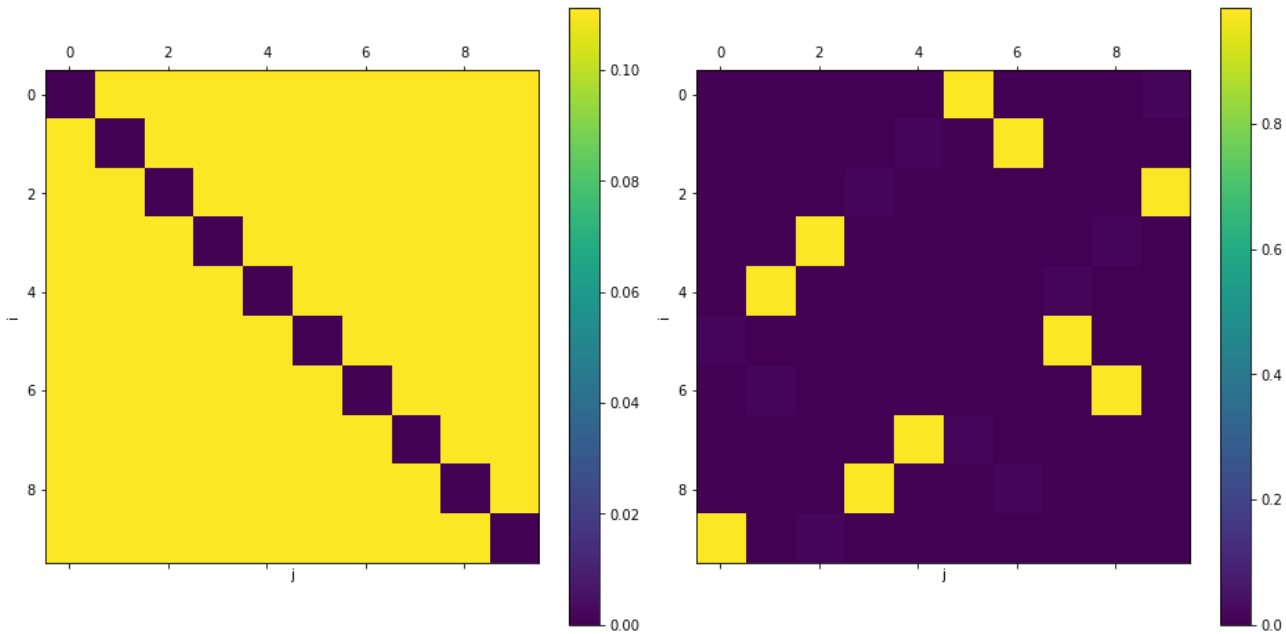


Figure: Matrice P: entrée et sortie de l'algorithme TSP, après 25 updates

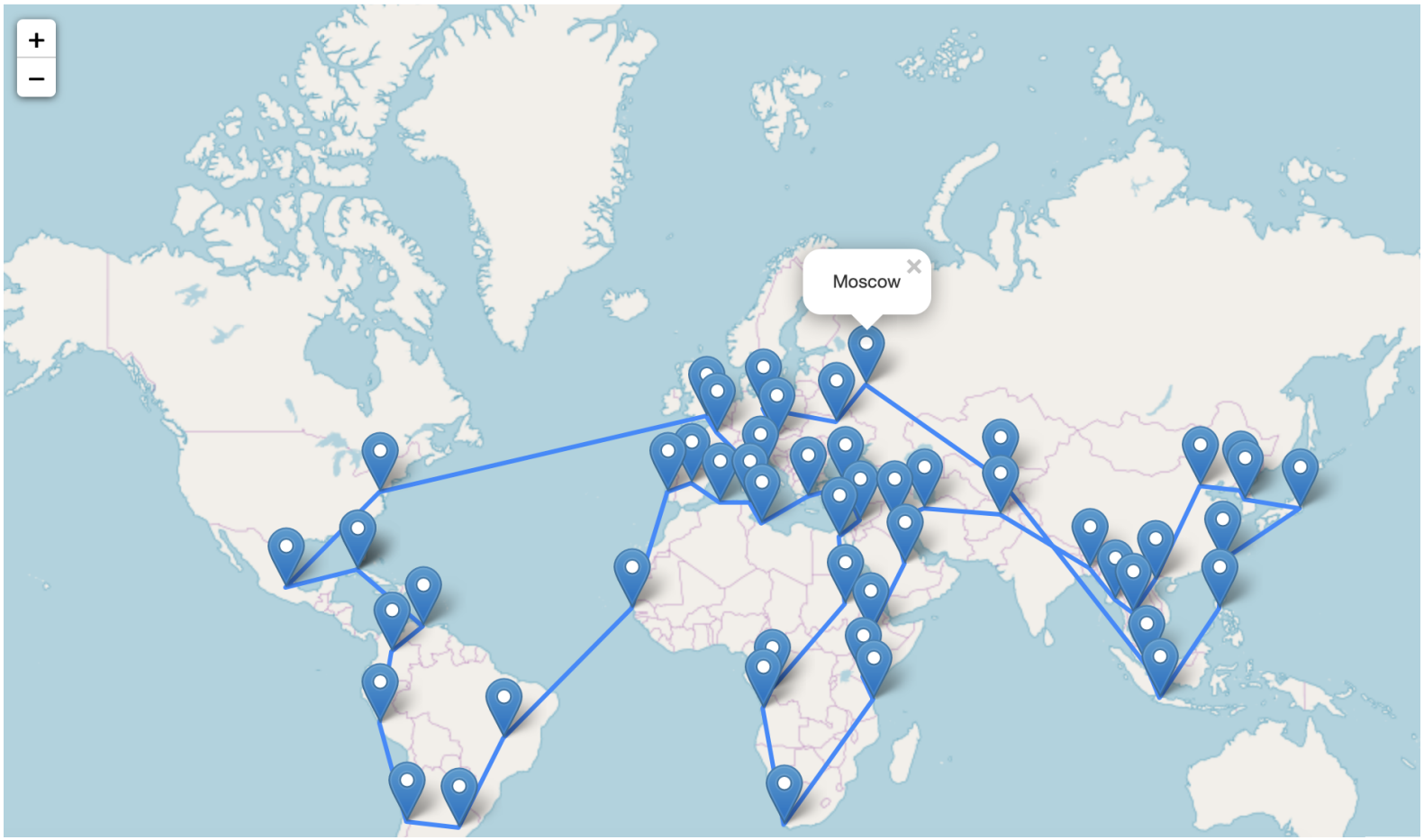
Solution 2 : Approche Bayésienne

Loi a priori: $(p_1, ..., p_{M-1}) \sim Dir(\alpha_1, ..., \alpha_{M-1})$
On montre facilement que la loi a posteriori suit la loi $Dir(\alpha_1 + n_1, ..., \alpha_{M-1} + n_{M-1})$ avec $(n_1, ..., n_{M-1})$ une simulation de Mult$(n; p_1, \cdots, p_{M-1})$ (démonstration en annexe).
On considère chaque ligne de P comme les réalisations du loi de Dirichlet. L'algorithme est le suivant :

- Intialiser les poids des Dirichlet
- Simuler les lignes de P selon ces Dirichlet
- Faire l'update des poids des Dirichlets en ne conservant que les q% des meilleurs tours. Simuler à nouveau P avec les nouveaux poids.
- Répéter jusqu'à stabilisation du quantile q

Application

Trouver le tour du monde optimal passant une seul fois par M capital. Exemple de solution avec M = 50 :



Détails pratiques

Les deux approches requièrent de choisir différents hyperparamètres.

Approche SoftUpdate

- initialisation de P, poids uniformes par défaut
- q, quantile pour la troncature des meilleurs tours. Par défaut $q = 0.01$ dans nos algorithmes.
- α, pondération entre l'ancienne matrice et la nouvelle dans le SoftUpdate. Si $\alpha = 1$, pas de SoftUpdate. Par défaut $\alpha = 0.9$
- N, nombre de tours simulés pour chaque update. En pratique il est conseillé de prendre $N = kM^2$. Par défaut $k = 10$.

Approche Bayésienne N et q sont choisis comme précédemment. Il n'y a pas de paramètre α. P n'a pas besoin d'être initialisé, on initialise les poids de Dirichlets à 1 par défaut de sorte que leur loi soit uniforme sur le $(M-1)-$simplexe.

Annexe: Démonstration du résultat bayésien

Loi a priori: $(p_1, ..., p_{M-1}) \sim Dir(\alpha_1, ..., \alpha_{M-1})$
C'est à dire $\pi(p) = \frac{1}{B(\alpha)} \prod_{i=1}^{M-1} p_i^{(\alpha_i - 1)}$
Vraisemblance de la loi multinomiale: $\frac{n!}{n_1!...n_{M-1}!} p_1^{n_1} ... p_{M-1}^{n_{M-1}}$
Loi a posteriori:

$$\begin{aligned} \pi(P|X) &\propto \pi(p) f(X|p) \\ &\propto \prod_{i=1}^{M-1} p_i^{\alpha_i - 1} p_i^{n_i} \\ &\propto \prod_{i=1}^{M-1} p_i^{\alpha_i + n_i - 1} \end{aligned}$$

Donc $P|X \sim Dir(\alpha_1 + n_1, ..., \alpha_{M-1} + n_{M-1})$
La loi de Dirichlet est conjugée pour la loi multinomiale