Université de Nantes
M1 Ingénierie

Optimisation stochastique
Exercices

Ex 1. Soit $\mathcal{X}$ la boule unité de $\mathbb{R}^2$ et U, V deux variables aléatoires indépendantes de loi uniforme sur $[0, 1]$.

1. Générer un vecteur de loi uniforme sur $\mathcal{X}$ par la méthode d'acceptation-rejet (remarquer que la loi uniforme sur $\mathcal{X}$ est la loi de (U, V) conditionnellement à l'événement $(U, V) \in \mathcal{X}$).
2. Soit $R = \sqrt{U}$ et $T = 2\pi V$, montrer que $(R\sin(T), R\cos(T))$ est de loi uniforme sur $\mathcal{X}$.
3. Générer des échantillons de tailles $N = 100$ par les deux méthodes et vérifier graphiquement qu'elles fonctionnent.
4. Comparer les temps de calcul des deux méthodes.
5. Reprendre l'algorithme d'acceptation-rejet pour générer la loi uniforme sur la boule unité de $\mathbb{R}^d$. Représenter l'évolution du temps de calcul en fonction de la dimension d.

Ex 2. Soit la fonction $J : (x, y, z) \mapsto |(x - y)(y - z)(x - z)|$ définie sur $[-1, 1]^3$.

1. Déterminer l'ensemble des minimiseurs de J. En déduire que J n'est pas convexe.
2. Implémenter l'algorithme de recherche aléatoire à l'aveugle pour un échantillon iid $U_1, ..., U_N$ de loi uniforme sur $[-1, 1]^3$ pour $N = 1000$.
3. Représenter l'évolution du processus (X_n, Y_n, Z_n) obtenu et de $J(X_n, Y_n, Z_n)$.
4. Justifier que $J(X_n, Y_n, Z_n)$ converge presque sûrement. Qu'en est-il de X_n, Y_n, Z_n?

Ex 3. Soit $J : (x, y) \mapsto x^2 + y^2 + 30(\sin^2 x + \sin^2 y)$, $(x, y) \in \mathbb{R}^2$.

1. Justifier que l'ensemble des minima locaux de J est contenu dans une boule centrée en 0 de rayon r suffisamment grand.
2. Tracer le graphe de J.
3. Rechercher le minimiseur de J par un algorithme d'exploration aléatoire.

Ex 4. Soit $J : x \mapsto x^4 - 16x^2 + 5x$.

1. Tracer le graphe de la fonction sur l'intervalle $[-4, 4]$. Au vu du graphe, justifier que le minimiseur de J sur $\mathbb{R}$ se trouve sur cet intervalle.
2. Implémenter plusieurs fois l'algorithme d'exploration localisée avec des incréments iid de loi uniforme sur $[-1, 1]$ en partant d'un point aléatoire de $[-4, 4]$. Que constate-ton?
3. Y a t-il le même problème avec l'exploration à l'aveugle?
4. Refaire l'exercice avec $J_2 : (x, y) \mapsto J(x) + J(y)$ sur $[-4, 4]^2$.

Ex 5. On veut classer un ensemble de points en deux groupes homogènes.

1. Importer les données `iris` accessibles par la commande `sklearn.datasets.load_iris()` de la librarie `sklearn`.
2. Représenter dans l'espace les deux premières variables du jeu de données.
3. Préciser la fonction objectif J utilisée et son domaine de définition adaptés au problème de classification non supervisée (càd où on ne connaît pas à l'avance les caractéristiques de chaque groupe).
4. Implémenter une méthode automatique de classification en 2 groupes homogènes basée sur un algorithme d'exploration aléatoire. Comparer les méthodes d'exploration à l'aveugle et localisée dans ce cas.
5. Reprendre les questions précédentes pour 3 groupes.
6. Comparer les groupes obtenus avec les variétés d'iris (variables `target_names` du jeu de données).

Ex 6. On s'intéresse au problème du voyageur de commerce.

1. Générer $n = 20$ points aléatoirement (qui représentent des villes numérotées de 1 à n) uniformément sur $[0, 1]^2$.
2. On cherche l'itinéraire de longueur minimale qui parcourt tous les points et revient à son point de départ. Formaliser le problème mathématiquement en précisant la fonction objectif J et son domaine de définition.
3. Générer un itinéraire aléatoirement, calculer sa distance totale et le représenter graphiquement.
4. Rechercher l'itinéraire le plus court par un algorithme d'exploration aléatoire à l'aveugle. Représenter graphiquement l'évolution de l'itinéraire le plus court (on pourra utiliser la fonction `matplotlib.pyplot.pause`).
5. Comparer le résultat avec un algorithme d'exploration aléatoire localisée, avec comme opération aléatoire à chaque itérations
   - la permutations de deux indices successifs.
   - la permutations de deux indices quelconques.
6. Reprendre l'exercice avec $n = 50$ villes.

Ex 7. On s'intéresse au comportement d'une bille qui se déplace le long d'une courbe $y = f(x)$ sur un plan vertical, sous le seul effet de la gravité (et sans vitesse intiale). La courbe brachistochrone est la courbe qui minimise le temps de parcours de la bille entre deux points donnés. On prend $A = (0, 1)$ comme point de départ, $B = (1, 0)$ comme point d'arrivée et on note g la constante gravitationnelle. On cherche à approcher numériquement la courbe brachistochrone.

1. Soit $x_1, ..., x_n$ une discrétisation de l'intervalle $[0, 1]$, la courbe d'équation $y = f(x)$ est approchée par interpolation linéaire en $(x_i, f(x_i))$. Etant donné un vecteur d'ordonnées $y_1, ..., y_n$, calculer la vitesse v_i de la bille au point d'abscisse x_i pour tout i (utiliser le fait que $v_i^2 + 2y_i g$ est constant lorsque $y_i \leq 1$ par le principe de conservation de l'énergie).
2. Sur l'intervalle $[x_i, x_{i+1}]$, on calcule

- la distance parcourue: $d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$,
- l'accélération (supposée constante sur l'intervalle): $a_i = (y_i - y_{i+1})g/d_i$,
- le temps de parcours: $t_i = (v_{i+1} - v_i)/a_i$.

Ecrire une fonction qui à un vecteur $(y_1, ..., y_n)$ associe le temps de parcours global (on prendra des x_i régulièrement espacés).

3. On pose $g = 9.81$. Rechercher la courbe brachistochrone par un algorithme stochastique.
4. Comparer les résultats pour $n = 4$ et $n = 50$.

Ex 8. Soit $J(x) = |x - 1| \sin(5x) + 2\cos(2x) + 5$, $x \in [-5, 5]$.

1. Représenter la mesure de Gibbs associée à J pour des températures $T = 10, 1, 0.1, 0.01$.
2. Implémenter l'algorithme du recuit simulé sur un échantillon de taille $N = 200$ pour minimiser la fonction J avec:
   - des incréments de loi uniforme sur $[-3, 3]$.
   - les températures $T_i = c/\log(1 + i)$ pour $c > 0$, $i = 1, ..., N$.
3. Représenter l'évolution de l'algorithme en "temps réel" en affichant la température.
4. Reprendre les questions précédentes pour maximiser J.
5. Faites varier les paramètres (N, températures et loi des incréments) et comparer les performances de l'algorithme.
6. Répéter l'expérience en représentant plusieurs trajectoires simultanément.

Ex 9. Reprendre le problème du voyageur de commerce en implémentant un algorithme de recuit-simulé et en calibrant au mieux les paramètres. Comparer les performances avec la recherche aléatoire localisée.

Ex 10. On recherche un mot de passe oublié. A chaque tentative, on nous donne le nombre de caractères corrects (et bien placés).

1. Formaliser le problème en précisant la fonction objectif J et son domaine de définition.
2. On prendra pour simplifier un mot de passe composé de 8 lettres écrites en minuscule. Tester l'efficacité de méthodes d'exploration aléatoires.
3. Proposer des procédés de sélection, croisement et mutation et implémenter un algorithme génétique pour retrouver le mot de passe.
4. Faire varier les paramètres (taille M de la population, pourcentage d'individus sélectionnés, taux de mutation, etc...).

Ex 11. On veut implémenter la méthode du gradient stochastique pour la régression logistique sur les données `pima`, qui cherchent à prédire si un individu est diabétique ($y_i \in \{1, 0\}$) en fonction de données médicales $x_{i,1}, ..., x_{i,p}$, pour $i = 1, ..., n$.

1. Charger le fichier `pima.csv` et préciser la taille de l'échantillon n et le nombre p de variables explicatives.

2. La régression logistique consiste à modéliser la probabilité de l'événement $\{y_i = 1\}$ par une fonction de la forme

$$f(\beta_0, ..., \beta_p | x_i) = \frac{1}{1 + \exp(-\beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p})}.$$

   On minimise la fonction de coût $J : (\beta_0, ..., \beta_p) \mapsto \sum_{i=1}^{n} (y_i - f(\beta_0, ..., \beta_p | x_i))^2$.

3. Implémenter l'algorithme de gradient stochastique avec les pas d'apprentissage $\gamma = 1, \gamma_n = 1/\sqrt{n}$, $\gamma_n = 1/n$ et vérifier s'il y a convergence.

4. Comparer avec la solution donnée par la fonction `LogisticRegression()` de la librairie `sklearn.linear_model`.

Ex 12. Soit ϕ_{μ,σ^2} la densité de la loi normale $\mathcal{N}(\mu, \sigma^2)$, on considère le modèle de mélange Gaussien de densité $x \mapsto \sum_{j=1}^{J} p_j \phi_{\mu_j, \sigma_j^2}(x)$, où $p_1, ..., p_J$ sont des poids positifs de somme 1.

1. Générer un échantillon $X = (X_1, ..., X_n)$ iid de taille $n = 200$ sous une loi de mélange avec:
   - $J = 3$, $p_1 = 1/2$, $p_2 = 1/3$ et $p_3 = 1/6$
   - $\mu_1 = 0, \mu_2 = 2, \mu_3 = 6$
   - $\sigma_1^2 = 4, \sigma_2^2 = 1$ et $\sigma_3^2 = 2$.

2. Coder la log-vraisemblance du modèle pour l'échantillon X généré.

3. Implémenter l'algorithme EM pour rechercher des estimateurs du maximum de vraisemblance des p_j, μ_j et σ_j^2.

4. Représenter sur un même graphique l'histograme de X et la densité estimée.