

Jérémie Haioun

Conduite de projets en science des données

1: IA

a) Décrire la différence entre approche réaliste et approche utilitariste dans la démarche scientifique.

D'un côté (approche réaliste) il s'agit d'une approche logique, fonctionnant par contraintes
(Utilisation de l'algebre, de l'analyse et des graphes)

De l'autre (approche utilitariste) il s'agit d'une approche statistique : apprentissage supervisé ou non, reseaux neuronaux, Bayes etc...

b) Watson Studio: décrire une caractéristique importante de Watson Studio comme environnement de développement en science des données, qui le rend utile en situation de développement en entreprise.

Watson Studio propose un environnement Cloud entièrement géré. Cela permet aux utilisateurs de collaborer au sein d'un projet dans un environnement intégrant des serveurs Notebook et R studio, et de programmer sur la data en R, Python et Scala

source : <https://www.ibm.com/fr-fr/cloud/watson-studio/pricing>

2: programmation logique/chainage avant

Dans un langage à base de règles simple en chaînage avant (on appelle cela un système de production) on a le programme:

```
var input[]=[], result[][], i=1, tmp=0;
when input.length>0 and i >= input.length then result.append(input[tmp]), input.removeAt(tmp), i=1, tmp=0;
when input[i] < input[tmp] then tmp=i, i=i+1;
when input[i] >= input[tmp] then i=i+1;
```

Lorsque l'instruction `input=[2,0,5,4,9];` est exécutée, que contiendront les variables `result` et `input` en retour? Expliquer l'algorithme.

```
result = [0,2,4,5,9]
input = []
```

Cet algorithme trie par ordre croissant une liste d'entiers. Pour cela, il sélectionne par comparaison successive l'élément de la liste le plus petit et l'ajoute à la liste `result`

3: Smart City: Trouver sur internet 3 logiciels commerciaux **professionnels** destinés à remplir un rôle similaire à celui du projet SmartDeliveries. En vous basant sur la présentation commerciale, identifiez leurs principales caractéristiques démarquantes (quels fonctionnalités mettent-ils particulièrement en avant par rapport à la concurrence). Fournir les références utilisées.

Waze : <https://www.lemondeinformatique.fr/actualites/lire-un-sdk-waze-pour-les-professionnels-du-transport-63722.html> Fonctionnalités mises en avant : Gagner du temps, mise à jour des informations fiable car en direct et fait par les utilisateur

BEMOBILE : <https://www.be-mobile.com/> Fonctionnalités mises en avant : Information trafic et stationnement. Moyen de paiement

sqline : <https://sqilline.com/products/smart-delivery> Fonctionnalités mises en avant : Transparence, sécurité des données, rapidité améliorée et retour utilisateur

4: trafic routier

a- Quelles sont les principales variables mesurées par un détecteur de trafic?

Un détecteur de trafic utilise principalement : La vitesse réglementaire de chaque route, les feux de circulation, la longueur de chaque route et le type de route

b- Qu'est ce que le diagramme fondamental d'un détecteur de trafic, pourquoi est-il utile pour mesurer et prévoir la congestion

Il s'agit d'un ensemble de graphique permettant d'analyser la congestion, puisque qu'il représente le flux, l'occupation ainsi que le debit en fonction du temps.

c- quel est le débit typique maximal d'un tronçon à une voie

- en zone urbaine
- sur voie rapide ou autoroute
- Pourquoi cette différence?

5: temps de parcours

a) Quelles sont les principales variables prédictives du temps de parcours d'un camion de livraison en ville, par ordre d'importance décroissante? (déterminées en cours)

Les variables predictives les plus importantes sont :

- La distance
- Type de véhicule
- Expérience du conducteur
- Motivation du conducteur
- Circonstances locales non prises en compte par le modèle

b) Citer 2 facteurs potentiels affectant les temps de parcours et difficiles à mesurer avec les données fournies dans les fichiers fournis en TP.

6: géoréférencement

On a un fichier d'adresses tel que vu en cours:

ID, BASE, KIND, CITY, FROM, TO

1, Docteur Bouchut, Rue du, Lyon, 1, 100

etc...

a) Décrire la logique d'une fonction python qui permet de retrouver l'identifiant de tronçon (ID) correspondant à chaque adresse, en supposant que la base d'adresses est stockée dans un tableau, et que l'on a une fonction distance(a, b) qui retourne la distance de Levenshtein entre 2 chaînes

17, rue Bouchut, Lyon

17 rue du Docteur Bouchot, Lyon

Tout ce que je demande, ce sont les différentes étapes à suivre, pas forcément le programme exact.

b) proposer cette fonction en python (améliorant celle vue en cours)

linkid(s, addresses):

"""" addresses contient une liste de tronçons [id, base, kind, city, from, to], s l'adresse à trouver """"

```
return id
```

Prescriptive Analytics

Question 1.

Les affirmations suivantes sont-elles vraies ou fausses:

a- Un problème de décision est dans la classe de complexité NP si et seulement si il n'existe pas d'algorithme polynomial pour le résoudre.

VRAI

b- Dans l'industrie, la majorité des problèmes d'ordonnancement sont résolus grâce à des heuristiques.

VRAI

c- Le problème suivant possède exactement trois solutions:

u in {1,3}

v in {1,2}

w in {3,4}

x in {1,5}

y in {4,5}

allDifferent(u,v,w,x,y)

VRAI

d- L'algorithme de résolution de CP-Optimizer est un algorithme exact: si un problème d'optimisation est faisable, il garantit de trouver une solution optimale.

VRAI

Question 2.

a- En cherchant sur internet, décrivez un problème d'optimisation combinatoire non vu dans le cours dont la version de décision est un problème NP-Complet.

la plus longue sous-séquence commune à deux suites, ou deux chaînes de caractères,
est une séquence étant sous-suite des deux suites, et étant de taille maximum.
[Wikipedia]

b- Décrivez une petite instance particulière de ce problème d'optimisation (avec des valeurs pour chacune des données).

Pour les deux séquences de caractères suivantes :

« abcde »,

« ceij »,

la plus longue sous-séquence commune est « ce ».

Dans ce problème, il est nécessaire que les éléments communs soient dans le même ordre dans les différentes séquences, mais pas qu'ils soient obligatoirement consécutifs : « e » n'est pas consécutif à « c » dans la première séquence.

c- Donnez une solution faisable non-optimale et une solution optimale de cette petite instance.

Non optimale : algorithme force brute

Optimale : Programmation dynamique

Question 3.

Deux principes fondamentaux de la Programmation par Contraintes sont (1) la recherche arborescente et (2) le filtrage du domaine des variables. Décrivez brièvement ces principes, leurs rôles et la façon dont ils sont mis en oeuvre durant la résolution.

(1) Test successif de toutes les possibilités. Si les conditions ne sont pas respectées alors on revient en arrière dans les étapes

(2) Permet l'amélioration des capacités de résolution lors de la construction de l'arbre de recherche par la suppression des valeurs inconsistantes des domaines de ces variables.

Question 4.

Un problème classique en ordonnancement est le problème d'open-shop pour lequel un ensemble de n jobs doivent être exécuté sur m machines. Chaque job consiste en un ensemble de m opérations de durée connue, devant être exécutées dans un ordre arbitraire sur les machines. Plus précisément, si l'opération o_{ij} désigne la jème opération du job i, cette opération utilise la machine j et sa durée est D_{ij} . L'ordre des opérations du job i n'est pas connu d'avance: les m opérations o_{ij} d'un job i doivent être ordonnées mais cet ordre est libre. D'autre part, une machine ne peut pas effectuer plus d'une opération à la fois. L'objectif est de déterminer les dates de début et de fin de chaque opération de manière à minimiser la date de fin du plan. Ce problème d'optimisation est NP-difficile.

Les données d'entrée du problème sont donc:

- n, le nombre de jobs
- m, le nombre de machines
- D_{ij} ($i \in [1,n], j \in [1,m]$), la durée de la jème opération du job i

a- Décrivez un modèle CP Optimizer à base de variables d'intervalles et de contraintes noOverlap pour résoudre le problème d'open-shop.

b- [BONUS] Décrivez un modèle CP Optimizer pour une variante du problème d'open-shop pour laquelle les machines peuvent effectuer au plus deux opérations en parallèle.