Evaluation IBM

1. IA

L'approche réaliste doit permettre d'expliquer comment les sciences parviennent avec succès à prédire les phénomènes. Elle énonce qu' "une théorie scientifique acceptée et consolidée décrit exactement comment les choses se passent dans le monde". Cette approche appréhende la complexité en cherchant à comprendre ce qui fonctionne, pour qui, et dans quels contextes. L'approche utilitariste quant à elle repose sur du biomimétisme pour aborder des problèmes complexes.

2. Programmation logique/chaînage avant

Il s'agit d'un algorithme pour ranger une suite par ordre croissant.
Lorsque l'instruction input=[2,0,5,4,9]; est exécutée, on obtient en sortie result=[0,2,4,5,9] et input=[].

Il s'agit d'un tri par insertion : la complexité est 0(n^2) en moyenne et dans le pire des cas et 0(n) dans le meilleur des cas.
C'est le tri souvent utilisé naturellement pour trier des cartes à jouer : les valeurs sont insérées les unes après les autres dans une liste triée (initialement vide). C'est souvent le plus rapide et le plus utilisé pour trier des entrées de petite taille.

3. Smart City

Le projet SmartDeliveries permet de combattre la saturation des routes en centre ville, la pollution, les coûts sans sacrifier l'activité économique, en optimisant la mobilité : prévoir le trafic à l'avance, prévoir les déplacements des véhicules, optimiser les chemins à emprunter.
Un site où l'on pourrait renseigner tous les jours les déplacements professionnels que l'on va réaliser dans la journée permettrait de proposer des itinéraires en évitant les zones qui sont encombrées et feraient perdre du temps.

Voici des logiciels pouvant présenter des similitudes avec ce projet :

- ➔ Waze
- ➔ Coyote : L'application donne des indications sur les zones de dangers, les bouchons, les accidents, les perturbations routières (travaux, météo, etc..)

- ➔ AnyLogic : La modélisation de simulation AnyLogic fournit une bibliothèque de trafic routier, qui offre la flexibilité et la puissance nécessaire pour parvenir à une ingénierie et à une conception du trafic routier plus efficace. Une visualisation claire est un moyen simple et rapide pour faciliter le développement de modèles d'infrastructures routières, avec des cartes de densité présentant la congestion et des animations montrant le flux du trafic.

- ➔ Citilog : crée une mobilité plus intelligente grâce à l'analyse vidéo et à l'intelligence artificielle appliquées aux solutions de gestion du trafic. Leurs solutions visent à réduire les coûts et les risques pour les opérateurs de trafic, en fournissant des outils plus intelligents et en fin de compte une route plus sûre pour les voyageurs qu'ils desservent.

- ➔ C-The Difference : Combler le vide qui sépare les implémentations les plus avancées des STI-C dans un environnement urbain et les déploiements et opérations à plus grande échelle en ciblant les professionnels responsables des opérations et de la planification des transports urbains, les décideurs politiques et les décisionnaires finaux.

4. Trafic routier

a)
Les principales variables mesurées par un détecteur de trafic sont le débit (mesurable par des dispositifs de comptage fixes), la vitesse (représente la vitesse de déplacement du flot de véhicules) et le taux d'occupation. Ce dernier représente le pourcentage de temps durant lequel un point de la route est « occupé » par un véhicule au-dessus de lui. Le taux d'occupation est exprimé en pourcentage. Il est à noter que sa qualité de mesure par des capteurs classiques est très dégradée lorsque les vitesses des véhicules sont basses.

b)
Le diagramme fondamental d'un détecteur de trafic illustre la densité du trafic. Pour comprendre le fonctionnement d'une section ou d'un réseau routier, l'approche consiste à mesurer l'évolution de la vitesse du trafic en fonction du débit et d'en déduire des paramètres de fonctionnement de la section, comme la vitesse libre ou la capacité.
Il s'agit d'une diagramme utile pour prévoir la congestion car on peut y trouver des régularités en fonction de l'heure et du jour.

c)
Les modèles de prévision du trafic sont généralement conçus et évalués sur les données des autoroutes, qui sont moins variables que les données des réseaux urbains et ne sont pas soumises aux effets des feux de signalisation.
Dans les réseaux urbains, les détecteurs peuvent être denses dans certaines parties d'un réseau urbain mais pas dans d'autres, de sorte que les emplacements contenant des informations prédictives peuvent être difficiles à identifier.

5. Temps de parcours

a)
Les principales variables prédictives du temps de parcours d'un camion de livraison en ville sont :

- Durée
- Heure
- Jour de la semaine
- Origine (x, y)
- Destination (x, y)
- Catégories de routes, nb voies
- Durée statique
- Distance (vol d'oiseau)
- Id mission
- Durée estimée avec les règles

b)
Voici des facteurs potentiels affectant les temps de parcours et difficiles à mesurer avec les données fournies dans les fichiers fournis en TP :

- Expérience du conducteur
- Motivation du conducteur (s'il doit se dépêcher…)
- Circonstances locales

6. Prescriptive Analytics

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.*

FAUX.
Un problème NP est un problème de décision vérifiant les propriétés suivantes :

- il est possible de vérifier une solution efficacement (en temps polynomial) ; la classe des problèmes vérifiant cette propriété est notée NP ;

- Tous les problèmes de la classe NP se ramènent à celui-ci via une réduction polynomiale ; cela signifie que le problème est au moins aussi difficile que tous les autres problèmes de la classe NP.

De plus, une propriété de la définition implique que s'il existe un algorithme polynomial pour résoudre un quelconque des problèmes NP-complets, alors tous les problèmes de la classe NP peuvent être résolus en temps polynomial.

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

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)

FAUX.
Il y a une seule solution : u=1, v=2, w=3, Y=4 et x=5.

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

7. Programmation par contraintes

Dans le cadre de la programmation par contraintes, les problèmes sont modélisés à l'aide de variables de décision et de contraintes, où une contrainte est une relation entre une ou plusieurs variables qui limite les valeurs que peuvent prendre simultanément chacune des variables liées par la contrainte.

1: Recherche arborescente
Dans le cas de la résolution sur domaines finis, il est en théorie possible d'énumérer toutes les possibilités et de vérifier si elles violent ou non les contraintes. C'est la méthode Générer et tester : on génère toutes les affectations possibles et on vérifie si elles correspondent à des solutions. Cependant, cette méthode n'est pas faisable pour des problèmes de taille moyenne en raison du grand nombre de combinaisons possibles.

2: Filtrage

Une des principales parties de la résolution, appelée « filtrage », a pour but d'éviter cette énumération exhaustive. Elle consiste à déduire à partir des contraintes les valeurs impossibles. Lorsqu'une variable ne possède plus qu'un candidat, celle-ci est instanciée.

Dès qu'une variable est affectée, on essaye de filtrer les valeurs pour les autres variables. On remplace la variable par sa valeur dans toutes les contraintes. On peut filtrer si une contrainte ne contient plus qu'une variable.

Cependant, le filtrage seul ne permet pas d'instancier toutes les variables, et il est donc nécessaire de scinder le problème en plusieurs parties (par exemple en instanciant une variable à chacune de ses valeurs possibles) et relancer le filtrage sur l'une de ces parties et ce, de manière récursive jusqu'à obtenir l'instanciation de toutes les variables.