PROJET ALGORITHME

PLAN

INTRODUCTION

1. JEU OU SITUATION PRESENTEE
2. FONCTIONNEMENT
3. UTILISATION DE LALGORITHME UTILISEE

CONCLUSION

BIBLIOGRAPHIE

INTRODUCTION

Nous remarquons que les algorithmes régissent presque tous les secteurs de notre environnement sans qu’on en prenne réellement conscience.

Par exemple se déplacer d’un point à l’autre, choisir quel itinéraire emprunter, jouer entre amis, etc.

Fort de ceci, mon choix s’est porté sur le cas Uber, sa gestion et algorithmes utilisés.

1. CAS UBER

UBER est une une application qui permet aux utilisateurs de trouver, commander et booker un véhicule pour leurs déplacements.

1. FONCTIONNEMENT

L’utilisateur se connecte sur la plateforme, il soumet sa requête. L’application soumet la requête au chauffeur le plus proche du client et lui demande sil veut effectuer la course.

Le chauffeur valide la requête s il souhaite effectuer la course, sinon il la rejette et elle est alors renvoyée au chauffeur suivant le plus proche et ainsi de suite.

Remarque :

Lorsque de nombreux passagers commandent en même temps et que le nombre de véhicules disponibles devient faible, Uber ajuste en temps réel le montant des courses en fonction du rapport entre l’offre et la demande. Le tarif augmente afin de maximiser le chiffre d’affaire et d’assurer une disponibilité auprès des clients.

Les tarifs sont recalculés toutes les 5 minutes et diminuent donc dès que le nombre de véhicules disponibles recommencent à augmenter.

1. ALGORITHME

Uber utilise comme algorithme:

-Dijkstra’s algorithm: car il recherche le chauffeur le plus proche premièrement (Shortest path) dans un lieu indiqué en tenant compte du temps que le chauffeur mettra à joindre son client (poids) et s’il n’en trouve pas il passe au suivant (graphe).

C’est sur cette base que Uber applique sa notion de Surge pricing ou tarification dynamique fondée sur Dijkstra’s algorithm premièrement et d’autres facteurs externes comme le temps, les heures de pointe qui peuvent faire varier le poids qui ici, est le temps.

On trouve aussi comme structure de données la récursion : car le client lance comme sa requête comme fonction d’appel, si le 1er chauffeur valide, le client se déconnecte et attend sa course (Base case), sinon il continue sa requête tant qu’il n’a pas trouvé de chauffeur disponible (recursive case).

CONCLUSION

Pour un peu qu’on soit curieux, nous pouvons a présent détecter un algorithme a chaque évènement ou situation qui nous entoure.

Bibliographie

<https://www.uber.com>

<http://www.journaldemontreal.com>

<https://www.google.ca/>