

Dans le cours, nous avons vu différents types de problèmes d'intelligence artificielle ainsi que plusieurs solutions possibles pour ceux-ci :

TYPE DE PROBLÈME	SOLUTIONS
Recherche du chemin le plus court	recherche heuristique A*
Optimisation d'une fonction générique	recherche locale : <i>hill climbing</i> , <i>simulated annealing</i> , algorithmes génétiques
Recherche pour jeux à deux adversaires	minimax, élagage alpha-beta
Problème de satisfaction de contraintes	<i>backtracking search</i> (avec/sans inférence, ordonnancement des variables, des valeurs), recherche locale <i>min-conflicts</i>
Inférence logique	méthode de résolution pour la logique du premier ordre
Raisonnement probabiliste	réseaux bayésiens (dynamiques)
Dérivation du plan optimal dans un environnement stochastique (MPD) connu	itération par valeur, itération par politique
Apprentissage automatique supervisé	classifieur des k plus proches voisins, Perceptron, régression logistique, réseaux de neurones
Apprentissage par renforcement	méthode directe, programmation dynamique adaptative, différence temporelle, <i>Q-learning</i> , recherche de plan/politique

Soit les situations suivantes, (1) identifiez le type de problème décrivant le mieux la situation et (2) justifiez votre choix en choisissant une solution possible pour ce type de problème et en décrivant brièvement comment vous l'appliqueriez à cette situation.

Voici un exemple de réponse à laquelle je m'attends.

Situations à résoudre :

1. Une compagnie de transport en commun souhaite développer un outil pour ses usagers, que ceux-ci pourraient consulter afin d'obtenir des suggestions d'itinéraires à emprunter dans leurs déplacements.

Solution : Cette situation correspond à une recherche du chemin le plus court. L'algorithme de recherche heuristique A* pourrait donc être utilisé. Chaque noeud dans le graphe de recherche serait une position possible dans le réseau de transport. Le noeud initial serait l'endroit de départ et le noeud but serait la destination que l'utilisateur souhaite atteindre. Les noeuds successeurs d'un noeud donné seraient l'ensemble des positions adjacentes dans le réseau (un noeud successeur par ligne de métro ou d'autobus couvrant la position donnée). Le coût entre deux noeuds serait le temps nécessaire pour passer entre deux positions à l'aide du mode de transport associé. Une heuristique similaire à la distance à vol d'oiseau entre les positions pourrait être utilisée par A*.

2. Une compagnie de réparation d'ordinateurs souhaite mettre sur pied un outil d'aide au diagnostic pour ses techniciens qui doivent réparer les ordinateurs de ses clients. La compagnie a accès à une base de données de faits concernant le fonctionnement d'un ordinateur (ex. : "si un pilote d'imprimante n'est pas installé, l'imprimante ne fonctionnera pas"). Ces faits sont déterministes.

Solution : Cette situation correspond à un problème d'inférence logique. On pourrait convertir la base de données de faits en formule de logique de premier ordre. La méthode de résolution pourrait alors être utilisée afin de faire l'inférence des causes à partir des "symptômes" ou problèmes observés.

Puisque les faits sont déterministes, on préfère l'inférence logique au raisonnement probabiliste.

3. On souhaite développer un jeu pour téléphone Android. C'est un jeu qui se joue à deux, ainsi on souhaite offrir la possibilité au joueur de jouer contre une intelligence artificielle. Il n'y a pas d'aspect aléatoire dans les règles du jeu : tout est déterministe. Le jeu est très simple et court à jouer, puis l'ensemble des règles du jeu sont faciles à décrire.

Solution : Cette situation correspond à un problème de jeux à deux adversaires. L'algorithme de basé sur l'élagage alpha-beta pourrait être utilisé (puisque plus efficace que minimax). Un état correspondrait à l'état du jeu, et les règles du jeu détermineraient la fonction de transition. Puisque le jeu est court, l'utilisation d'une fonction heuristique afin de diriger la recherche ne serait pas nécessaire.

On ne peut pas utiliser les algorithmes de processus de décision markoviens (itération par valeur, itération par politique), puisque c'est un jeu à deux adversaires et qu'on ne connaît pas la stratégie de l'adversaire. On ne peut donc pas définir le modèle de transition $P(s'|s, a)$ de l'environnement.

On pourrait par contre faire de l'apprentissage par renforcement, puisque l'apprentissage par renforcement ne nécessite pas de connaître le modèle de transition. Par contre, puisque le jeu est simple et déterministe, l'élagage alpha-beta serait une solution plus simple.

4. Une compagnie d'assurance automobile souhaite implémenter un outil permettant d'évaluer le risque qu'un nouveau client ait un accident et fasse une réclamation. La compagnie a accès à plusieurs statistiques sur d'anciens clients ayant ou n'ayant pas eu d'accidents. Pour chacun de ces anciens clients, on a de l'information sur leur profil : leur âge, la marque de leur voiture, s'ils habitent en ville ou en campagne, la distance quotidienne parcourue en voiture, etc. L'information est partielle, c.-à-d. que

certaines des informations du profil ne sont pas connues. Étant données de l'information (aussi partielle) sur un nouveau client, la compagnie aimerait donc avoir un outil pour déterminer le risque que ce client fasse éventuellement une réclamation pour un accident.

Solution : Cette situation correspond à un problème de raisonnement probabiliste. On pourrait utiliser un réseau bayésien afin de modéliser la relation entre le fait qu'un client ait fait une réclamation et les caractéristiques du client. Les paramètres du réseau bayésien (c'est-à-dire les probabilités a priori et conditionnelles dans le réseau) seraient obtenues ("appries") à partir des statistiques de fréquence du client. Afin de faire une prédiction pour un nouveau client, on devrait faire de l'inférence par énumération afin de calculer $P(\text{client fait réclamation}|\text{profil partiel du client})$.

On ne pourrait pas utiliser les algorithmes d'apprentissage automatique vus en classe (Perceptron, régression logistique, réseau de neurones et classifieur k plus proches voisins) à cause de la possibilité qu'il manque certaines informations dans le profil des clients. Effectivement, ces algorithmes requièrent que l'entrée \mathbf{x} soit complète, sans valeurs manquantes.