

Cours 1 : Introduction + Agent.....	3
Définition de l'IA.....	3
Fondations de l'IA :.....	3
Agent et environnement.....	3
Définition.....	3
Fonction et programme.....	4
Exemple : Vacuum world.....	4
Agent relationnel.....	4
Environnement de tâche - PEAS (Performance measure, Environment, Actuators, Sensors).....	5
Type d'agents.....	5
Cours 2 : recherche Non Informée.....	6
Agent de Planification.....	6
Problème de recherche.....	6
Graphe d'état et arbres de recherche.....	6
Stratégies de recherche non informée.....	7
Depth-First Search [DFS] - Recherche en profondeur.....	7
Breadth-First Search [BFS] - Recherche en largeur.....	7
Cours 3 : Recherche Informé.....	9
Récap des Stratégies de recherche.....	9
Uniform Cost Search (UCS).....	9
Méthodes Heuristiques.....	9
Hill Climbing.....	10
Hill Climbing V2.....	10
Best-First.....	11
Beam Search Algorithme.....	11
A* Algorithme.....	12
Recap.....	12
Cours 4 : AI Adversarial Search.....	13
Historique.....	13
Concepts de Base des Jeux.....	13
Types de jeux.....	13
Jeux standards.....	13
Algorithme Minimax / Alpha-beta.....	14
Minimax.....	14
Alpha-Bêta.....	14
Cours 5 : ML.....	15
Définition du ML.....	15
Diff entre IA d'origine et la nouvelle IA.....	15
Types d'apprentissage.....	15
Algorithmes de ML.....	15
Classification supervisée.....	15

Régression Supervisée.....	15
Clustering non supervisé.....	16
Evaluation des modèles de ML.....	16
Overfitting & Underfitting.....	16
Méthodes de validation croisée.....	16
Cours 6 : NLP.....	17
Définition.....	17
Historique.....	17
Niveaux et Problèmes.....	17
Application.....	18
Analyse d'Opinion.....	18
Détection de Fake News.....	18
Résumé Automatique.....	18
Recherche d'Information.....	18
Correction Orthographique et Grammaticale.....	18
Traduction Automatique.....	18
Développement d'Outils.....	18

Cours 1 : Introduction + Agent

Définition de l'IA

- Acting Humanly :
 - Test de turing (1950 - Alan Turin) : une machine passe le test de Turing si elle peut imiter de manière convaincante un comportement humain dans une conversation
- Thinking Humanly : créer de programmes qui pensent humainement, en s'inspirant des expériences psychologiques et techniques d'imagerie cérébrale
- Thinking Rationally : L'approche des lois de la pensée
- Acting Rationally : L'approche des agents rationnels définit un agent relationnel comme un programme informatique qui fonctionne de manière autonome, perçoit son environnement, s'adapte aux changements et poursuit des objectifs. La rationalité consiste à maximiser l'utilité attendue des décisions prises, indépendamment du processus de pensée qui les sous-tend

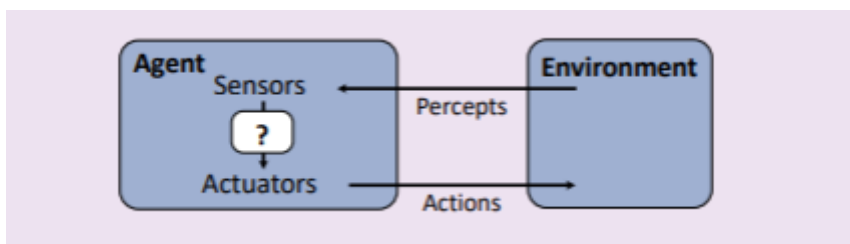
Fondations de l'IA :

Elle est basé sur mathématique, philosophie, psychologie, linguistique, informatique, économie, Neuroscience

Agent et environnement

Définition

- Agent : une entité qui perçoit son environnement à travers des capteurs (sensors) et agit sur cet environnement via des effecteurs (actuators)



- Exemple d'agent :

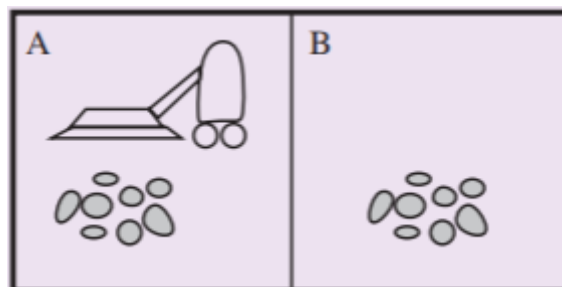
Agent	Sensors	Actuators
Humain	Vision, audition, toucher, odorat, goût, proprioception	Muscle, sécrétations, modifications de l'état cérébral

Calculatrice de poche	Capteurs d'états de touches	Affichage des chiffres
-----------------------	-----------------------------	------------------------

Fonction et programme

- Fonction de l'agent : mappe les histoires de percepts (P^*) aux actions (A). Elle détermine la réponse de l'agent à toute séquence de percepts. $f: P^* \rightarrow A$
- Programme de l'agent (I) s'exécute sur une machine M pour implémenter la fonction de l'agent $f = \text{Agent}(I, M)$. La fonction de l'agent dépend de la machine M et le programme I

Exemple : Vacuum world



Deux Zones : A et B

- Percepts [Zone, état] (exp : [A, Sale])
- Actions : Left, Right, Suck, NoOp

Agent function		Agent program
Percept sequence	Action	function Reflex-Vacuum-Agent([location,status]) returns an action if status = Dirty then return Suck else if location = A then return Right else if location = B then return Left
[A,Clean]	Right	
[A,Dirty]	Suck	
[B,Clean]	Left	
[B,Dirty]	Suck	
etc	etc	

Agent relationnel

- Un agent rationnel choisit l'action qui **maximise** la valeur attendue de la **mesure de performance**. Par exemple, un agent aspirateur rationnel pourrait être évalué sur la base de *la propreté des carrés nettoyés par unité de temps*. La rationalité inclut l'exploration et l'apprentissage, surtout dans des environnements inconnus

Environnement de tâche - PEAS (Performance measure, Environment, Actuators, Sensors)

Exemples:

- Pacman :
 - **Mesure de performance** : Critères pour évaluer le succès de l'agent. [-1 per step, +10 food, +500 win]
 - **Environnement** : Le monde dans lequel l'agent opère. [pacman dynamics, y compris le comportement des fantômes]
 - **Effecteurs** : Mécanismes par lesquels l'agent agit. [Left Right, Up, Down]
 - **Capteurs** : Mécanismes par lesquels l'agent perçoit l'environnement [Entire state est visible]
- Taxi automatisé
 - **Mesure de performance** : Revenus, satisfaction des clients, coûts du véhicule, amendes, primes d'assurance.
 - **Environnement** : Rues, autres conducteurs, clients, météo, police.
 - **Effecteurs** : Direction, frein, accélérateur, affichage/haut-parleur.
 - **Capteurs** : Caméra, radar, accéléromètre, capteurs moteur, microphone, GPS

Type d'agents

Les agents peuvent être classés selon leur complexité et leur généralité croissantes :

1. **Agents réflexes simples** : Réagissent directement aux percepts.
2. **Agents réflexes avec état** : Tiennent compte de l'historique des percepts.
3. **Agents basés sur des buts** : Agissent pour atteindre des buts spécifiques.
4. **Agents basés sur l'utilité** : Agissent pour maximiser une fonction d'utilité

Cours 2 : recherche Non Informée

C'est une stratégie de recherche qui ne possède aucune information supplémentaire sur l'espace de recherche au-delà de la définition du problème elle-même.

Agent de Planification

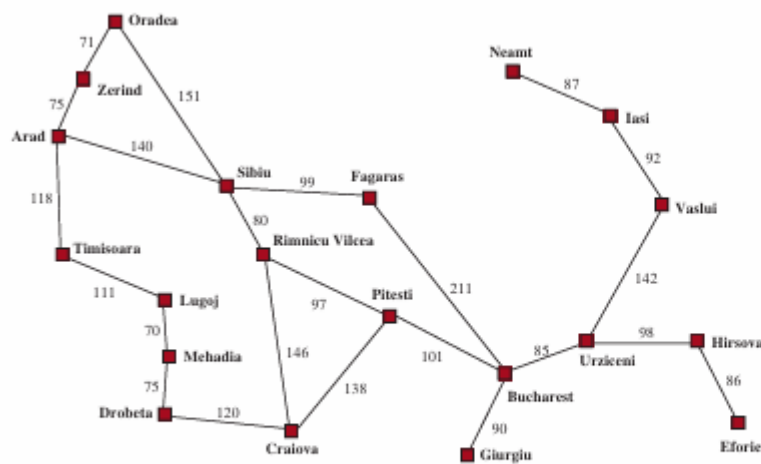
Agent de planification décide de l'action basée sur l'évaluation des séquences d'actions futures. A besoin d'un modèle de l'évolution du monde en réponse aux actions. A un but généralement bien défini

Problème de recherche

est défini par 5 éléments

- État initial (s_0)
- Actions ($A(s)$)
- Modèle de transition ($Result(s, a)$)
- Test de but ($G(s)$)
- Coût d'action ($c(s, a, s')$)

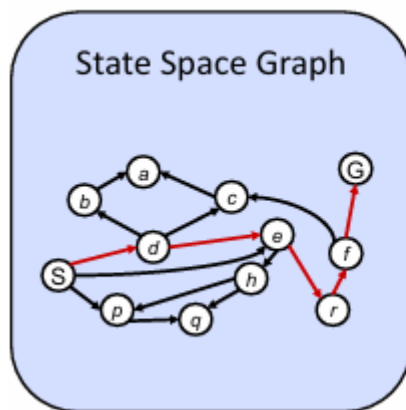
Exemple : tours de Hanoi, Voyager en Roumanie



- **State space:**
 - Cities
- **Initial state:**
 - Arad
- **Actions:**
 - Go to adjacent city
- **Transition model:**
 - Reach adjacent city
- **Goal test:**
 - $s = \text{Bucharest?}$
- **Action cost:**
 - Road distance from s to s'
- **Solution?**

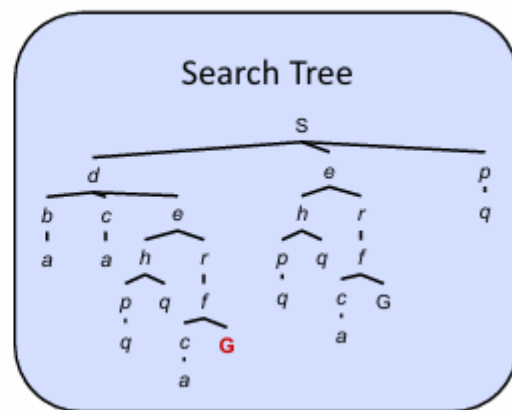
Graphe d'état et arbres de recherche

Les graphes d'état représentent **mathématiquement** un problème de recherche où chaque **état n'apparaît qu'une fois**. En revanche, les arbres de recherche sont construits à la demande et chaque nœud **représente un chemin entier** dans le graphe d'état.



Each NODE in the search tree is an entire PATH in the state space graph.

We construct the tree on demand – and we construct as little as possible.



Stratégies de recherche non informée

On juge les Algorithmes de recherche selon

- Complétude (Si la sol existe, on va la trouver)
- Optimalité (trouver chemin de moindre coût)
- Complexité temporelle
- complexité spatiale

Depth-First Search [DFS] - Recherche en profondeur

- Explore les nœuds les plus profonds en premier.
- Utiliser une pile (LIFO) pour la gestion de la frontière.
- Non optimale et non complète, mais espace de stockage linéaire

Breadth-First Search [BFS] - Recherche en largeur

- Explore les nœuds les plus proches de la racine en premier.
- Utilise une file (FIFO) pour la gestion de la frontière.
- Complète et optimale pour les coûts d'action unitaires, mais complexité spatiale exponentielle.

Aspect	BFS	DFS
Complétude	Oui (si Sol. existe)	Non (Peut ne pas trouver une sol. si on atteint le nombre limite de profondeur)
Optimalité	Oui (Si les coûts d'actions uniformes)	Non
Complexité temporelle	Exponentielle	Exponentielle

Complexité Spatiale	Exponentielle	Linéaire
Meilleures Performance	- Coût uniforme des actions - Profondeur limitée ou faible par rapport à la branche	- Profondeur Inconnue ou très grande - Utilisation de mémoire limitée
Moins Performant	- Espace de recherche largement branchés (du à la mémoire)	- Profondeur limite-Recherche de sol. optimales

Cours 3 : Recherche Informé

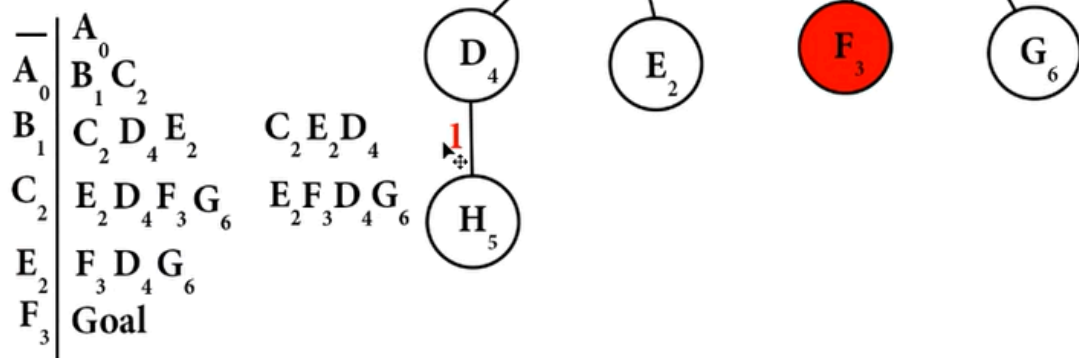
Récap des Stratégies de recherche

- **Recherche Systématique Non-informée** : BFS et DFS
- **Recherche avec Information sur les Arcs** : Uniform Cost Search UCS (algorithme de Dijkstra).
- **Recherche avec Information sur les Nœuds** : Hill-Climbing, Best-First et Beam Search
- **Recherche avec Information sur les Nœuds et les Arcs**: A*

Uniform Cost Search (UCS)

- Information sur les arcs
- UCS développe le nœud avec le coût de chemin le plus bas ($g(n)$).
- Utilise une file de priorité pour gérer la frontière.
- Convient pour trouver le chemin de moindre coût mais peut être inefficace sans une bonne heuristique

Uniform Cost Search (UCS)



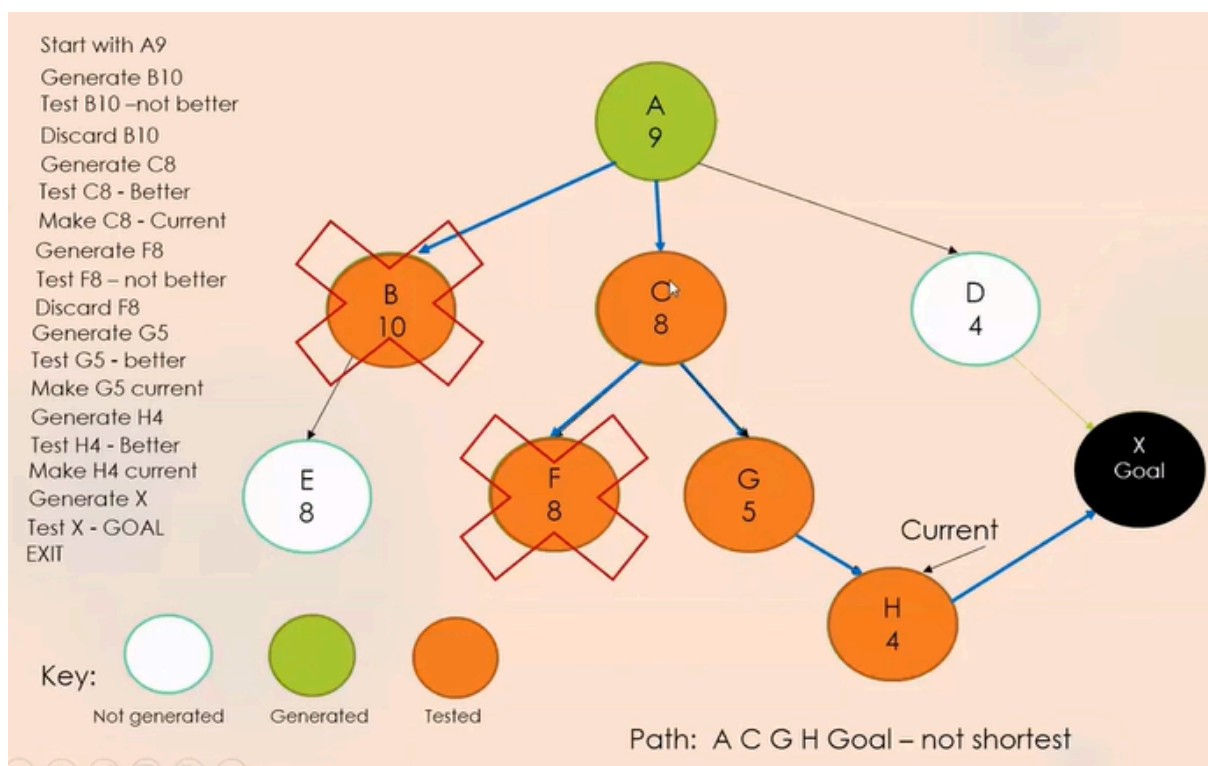
Méthodes Heuristiques

- **Recherche Systématique** : Explore tous les chemins possibles, ce qui peut prendre *beaucoup de temps*.
- **Recherche Heuristique** : Utilise des heuristiques pour prioriser les chemins qui semblent plus prometteurs, *réduisant ainsi le temps de recherche*.

- Exemple : Trouver l'itinéraire en Roumanie, en utilisant la distance à vol d'oiseau jusqu'à Bucarest. Puzzle 8, heuristiques : nombre de tuiles mal placées, Distance de Manhattan (Somme des distances horizontales et Verticales des tuiles % leurs position final
- Heuristique Admissible : si elle ne surestime jamais le coût pour atteindre le but
 - h est admissible si $0 \leq h(n) \leq h^*(n)$ avec $h^*(n)$ le coût réel

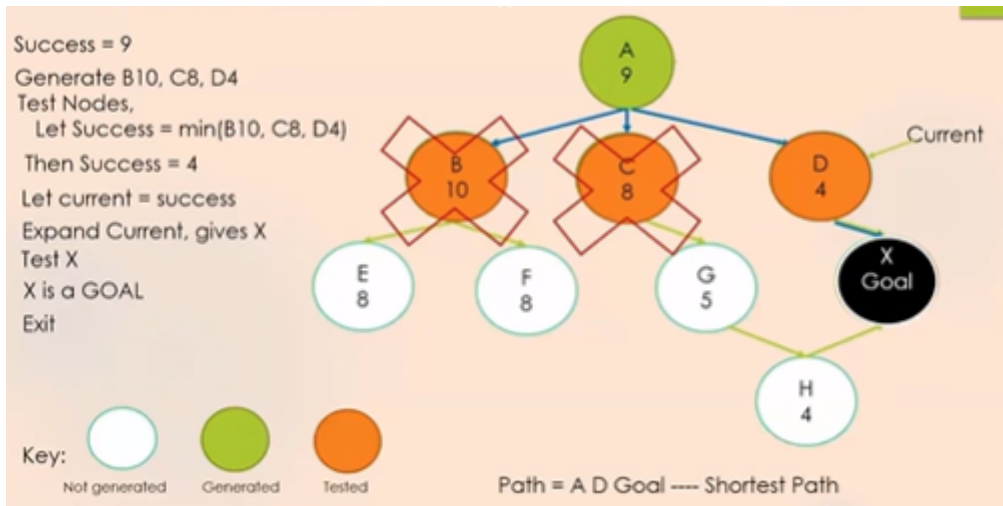
Hill Climbing

- commencer n'importe où, puis se déplacer vers l'État voisin le plus prometteur.
- Problèmes :
 - L'algo peut s'arrêter avant d'atteindre l'état but
 - Il ne considère qu'un seul chemin à la fois



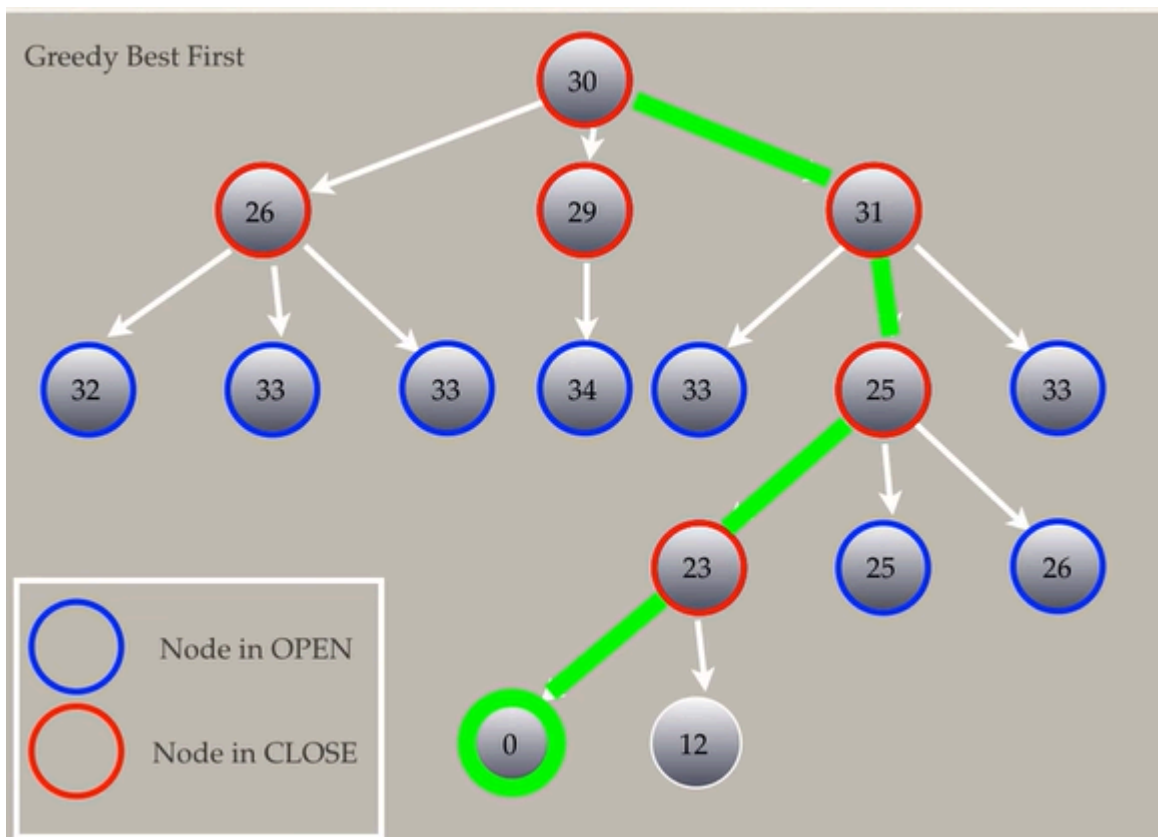
Hill Climbing V2

Cette variation Choisit l'état produit par le meilleur mouvement comme état suivant.



Best-First

- Combine les recherches en profondeur et en largeur.
- Suivre un chemin à la fois, mais changer dès qu'un chemin plus prometteur apparaît.
- Méthode :
 - Sélectionner le nœud le plus prometteur parmi tous les nœuds générés.
 - Développer ce nœud et ajouter ses successeurs à la liste des nœuds à explorer.



Beam Search Algorithm

- une amélioration du Best-First
- une amélioration du Best-First
- Efficacité en Mémoire
- Rapidité

A* Algorithme

- Définit une fonction $f(n)$ comme combinaison du coût réel $g(n)$ et d'une estimation heuristique $h(n)$.
- Expands le nœud avec la plus basse valeur de $f(n) = g(n) + h(n)$.

Recap

Aspect	UCS (Dijkstra)	Hill Climbing	Best-First Search	A*
Complétude	Oui (si coût positif et fini)	Non (peut se bloquer)	Non (peut ne pas explorer tout les chemins)	Oui (si heuristique admissible)
Optimalité	Oui	Non (peut se bloquer)	Non (ne garantit pas une sol optimal)	Oui (si heuristique admissible)
Complexité temporelle	$O(b^d)$	$O(\text{infini})$, dans le pire des cas (ça dépend du nombre d'itération)	$O(b^m)$ avec m profondeur max)	$O(b^d)$ dans le pire des cas, avec d la profondeur de la sol
Complexité Spatiale	$O(b^d)$	$O(1)$ ne stocke que l'état actuel	$O(b^m)$ avec b facteur de branchement	$O(b^d)$ (nécessite une grande quantité de mémoire)

⚠ Complexité temporelle et spatiale de A* Peut être exponentielle, mais est généralement plus efficace que UCS car l'heuristique guide la recherche de manière plus informative.

Cours 4 : AI Adversarial Search

Historique

Jeu	Date
Checkers	<ul style="list-style-type: none">- 1950 : First computer player- 1994: First computer world champion: Chinook defeats Tinsley
Chess	<ul style="list-style-type: none">- 1945-1960: Zuse, Wiener, Shannon, Turing, Newell & Simon, McCarthy (Computer players)- 1997: Deep Blue defeats human champion Gary Kasparov
Go	<ul style="list-style-type: none">- 1968: Zobrist's program plays legal Go, barely (b>300!)- 1968-2005: various ad hoc approaches tried, niveau Débutant- 2005-2014: Monte Carlo tree search -> Niveau Amateur- 2016-2017: AlphaGo defeats human world champions

Concepts de Base des Jeux

Types de jeux

Un jeu peut être définie sur plusieurs axes :

- Déterministe (Sans facteur aléatoire exp échecs) ou stochastique
- Information parfaite (Connaissance complète de l'état de jeu exp échecs) ≠(jeu de carte)
- un ou plusieurs joueurs
- tour par tour ou simultané
- Somme nulle (Les gains d'un joueur = pertes de l'autre)

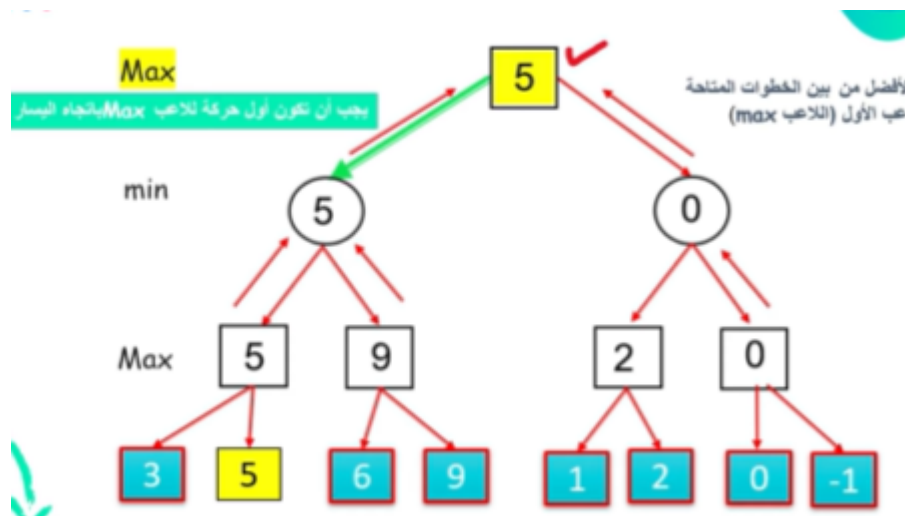
Jeux standards

Un jeu standard est déterministe, observable, à deux joueurs, à tour de rôle et à somme nulle.

Algorithme Minimax / Alpha-beta

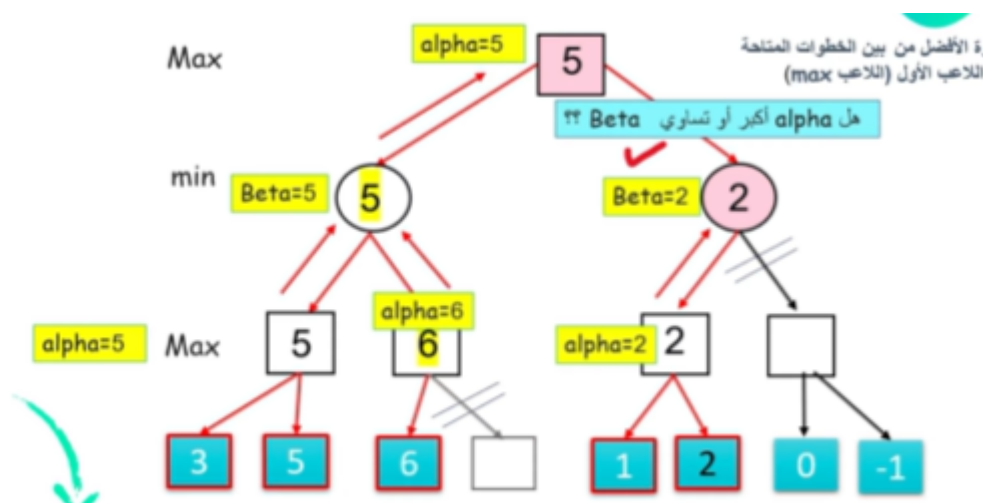
Minimax

Principe : choisir l'action qui mène à l'état avec la meilleure valeur minimax, en supposant que tous les futurs mouvements seront optimaux.



Alpha-Bêta

- Améliore l'efficacité de minimax en évitant d'explorer des branches de l'arbre de recherche qui ne peuvent pas influencer la décision finale.



Cours 5 : ML

Définition du ML

- En **1959, Arthur Samuel** a introduit le terme "machine learning" en le définissant comme le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés.
- c'est un domaine de l'IA qui se concentre sur apprendre des données disponibles pour faire des prédictions sur des données invisibles sans programmation explicite

Diff entre IA d'origine et la nouvelle IA

Approche basée sur des règles	Approche statistique
Automatise les tâches répétitives	ML établit des méthodes pour apprendre indépendamment, automatisant
Précision de 100%	Précision de 70% (30% de réponses fausses)

Types d'apprentissage

Supervisé	Non Supervisé
Utilisation des données étiquetées	Détection de relations entre données sans étiquetées
Classification (étiquette de classe discrète) et Régression (Prévoir une quantité continue)	Clustering (diviser un ensemble de données en groupes)

Algorithmes de ML

Classification supervisée

- Naïve Bayes
- Régression Logistique
- Support Vector Machine (SVM)
- Arbre de Décision
- K-NN

Régression Supervisée

- Régression Linéaire (Analyse de l'efficacité du marketing, prévision des ventes.)

Clustering non supervisé

- K-means (exp d'appli : Segmentation de la clientèle, segmentation d'images, systèmes de recommandation)

Evaluation des modèles de ML

Overfitting & Underfitting

- Overfitting (Surapprentissage)
 - Se produit lorsqu'un modèle apprend trop bien les détails et le bruit des données d'entraînement au point de nuire à sa performance sur les nouvelles données
 - Raison : trop de caractéristiques utilisées ou réutilisation des échantillons de formation dans les tests
 - Solutions :
 - Réduire le nombre de caractéristiques.
 - Augmenter la quantité de données.
 - Utiliser la validation croisée (cross-validation)
- Underfitting (Sous-apprentissage)
 - Se produit lorsqu'un modèle est trop simple pour capturer les tendances sous-jacentes des données.
 - Raisons possibles : Utilisation d'un estimateur trop simple.
 - Solutions :
 - Ajouter plus de caractéristiques.
 - Utiliser un estimateur différent.

Méthodes de validation croisée

C'est une technique d'évaluation d'un modèle en divisant l'ensemble de données en plusieurs sous-ensembles pour la formation et le test.

Hold-out method	Division aléatoire de l'ensemble de données en un ensemble de formation et un ensemble de test
K-fold cross-validation	Division en K sous-échantillons. Chaque sous-échantillon est utilisé à son tour comme ensemble de validation, et les K-1 sous-échantillons restants sont utilisés comme ensemble de formation.
Leave-one-out cross-validation, (LOO-CV)	Semblable à k-fold, mais avec un sous-échantillon contenant un seul point de données tenu à l'écart, et le reste des données utilisé pour la formation

Cours 6 : NLP

Définition

- NLP désigne l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et comprendre des énoncés linguistiques à des fins de communication. Il utilise des outils et techniques issus de diverses disciplines :
 - **Linguistique** : descriptions explicites.
 - **Mathématiques** : Passage entre linguistique et l'informaticien
 - **Informatique** : optimisation et implémentation des algorithmes et programmes.
 - **Intelligence artificielle et psychologie expérimentale** : représentation des connaissances.

Historique

- 1943 : Modèle de circuit booléen du cerveau par McCulloch et Pitts.
- 1950 : Machine de Turing.
- 1956 : Adoption du terme "Intelligence Artificielle" lors de la réunion de Dartmouth.
- 1952-69 : Développements initiaux avec de grands espoirs (GPS de Newell et Simon, LISP de McCarty).
- **1966-73 : Découverte de la complexité computationnelle, recul de la recherche sur les réseaux neuronaux.**
- 1969-79 : Développement des systèmes à base de connaissances (systèmes experts comme MYCIN et Dendral).
- 1980 : L'IA devient une industrie.
- 1986 : Retour des réseaux neuronaux.
- 1987 : L'IA devient une science.
- 1995 : Émergence des agents intelligents.

Niveaux et Problèmes

- Traitement d'image -OCR : Extraction de texte à partir d'images
- Traitement de Voix - Reconnaissance et Synthèse Vocale : Reconnaissance et synthèse vocale
- Traitement de texte
 - Prétraitement de texte :
 - Normalisation : Convertir les données en format standardisé
 - Segmentation : Diviser le texte en phrases/mots
 - Tokenisation : Fractionner le texte en unités minimales (tokens) [mots, ponctuations, etc]
 - Analyse morphologique : étude de la structure des mots. Spécifie comment les mots sont construits en identifiant les composants lexicaux et leurs propriétés (goes → go + es). Permet ainsi d'enlever l'ambiguïté des mots

- Analyse syntaxique : étude de la structure des phrases. Identifier la structure des phrases et les relations entre les mots. L'ambiguïté syntaxique est traitée à ce niveau
- Analyse sémantique : identification du sens des phrases hors contexte. Permet la traduction des phrases et traite l'ambiguïté sémantiques
- Analyse pragmatique : compréhension du sens des phrases dans leur contexte. Ce niveau est crucial pour comprendre les intentions et significations implicites dans une conversation

Application

Analyse d'Opinion

- Détection des sentiments (positif, négatif, neutre) dans un texte.
- Distinction entre objectivité et subjectivité.

Détection de Fake News

- Identification et gestion des fausses informations.

Résumé Automatique

- Techniques de synthèse pour réduire la taille des textes tout en conservant l'essentiel de l'information.

Recherche d'Information

- Indexation et recherche de documents pertinents basées sur leurs caractéristiques analytiques.

Correction Orthographique et Grammaticale

- Identification et correction des fautes d'orthographe et de grammaire.

Traduction Automatique

- Utilisation d'outils en ligne pour la traduction entre différentes langues (BabelFish, Bing Translator, Reverso, Google Translate).

Développement d'Outils

- Utilisation de bibliothèques et plateformes telles que NLTK, GATE, UIMA pour le développement d'applications NLP.