

# Chapitre 1 : Introduction à l'Intelligence Artificielle (IA)

## 1. Définition de l'IA

L'Intelligence Artificielle (IA) peut être définie différemment selon les perspectives :

- **Alan Turing** : L'IA est ce qui rend difficile la distinction entre une tâche réalisée par un humain et une machine.
- **Charles Darwin** : L'IA est ce qui permet la survie de l'individu le plus apte, capable de s'adapter à son environnement.
- **Thomas Edison** : L'IA est tout ce qui fonctionne efficacement et génère du profit.
- **Microsoft** : L'IA désigne les logiciels qui imitent les comportements et les capacités humaines.

## 2. Charges de travail principales de l'IA

L'IA comprend plusieurs domaines clés :

- **Apprentissage automatique (Machine Learning)** : Enseigner à un modèle informatique à faire des prédictions à partir de données.
  - **Apprentissage supervisé** : L'algorithme apprend à partir de données étiquetées.
  - **Apprentissage non supervisé** : L'algorithme identifie des structures dans des données non étiquetées.
  - **Apprentissage par renforcement** : L'algorithme apprend en interagissant avec un environnement.
- **Détection d'anomalies** : Identification automatique d'erreurs ou d'activités inhabituelles.
- **Vision par ordinateur** : Capacité des logiciels à analyser des images et vidéos.
- **Traitement du langage naturel (NLP)** : Compréhension et génération du langage humain.
- **Exploration des connaissances** : Extraction d'informations à partir de données non structurées.

## 3. Caractéristiques de l'IA

L'IA se distingue par plusieurs éléments :

- **Manipulation des informations symboliques** : Contrairement aux systèmes classiques, l'IA utilise des concepts abstraits et des règles (ex. température de 39,2°C → patient févreux).

- **Utilisation des méthodes heuristiques** : Contrairement aux algorithmes déterministes, les heuristiques permettent de trouver des solutions approximatives plus rapidement.
- **Traitement d'informations incomplètes** : L'IA utilise des raisonnements approximatifs et statistiques.
- **Représentation des connaissances** : Stockage et traitement des connaissances sous forme de bases de données spécialisées.

## 4. Historique de l'IA

- **Années 50** : Optimisme initial, mais échecs dans les jeux et la reconnaissance vocale.
- **Années 60** : Développement des premiers systèmes heuristiques.
- **Années 70** : Expansion avec les systèmes experts et la compréhension du langage naturel.
- **Années 80** : Intégration de l'IA dans l'économie avec des applications industrielles.
- **Années 90** : Commercialisation (assistants intelligents, robots domestiques).

## 5. Applications de l'IA

L'IA est omniprésente dans divers domaines :

- **Recherche et recommandations** : Google (suggestions de recherche), Netflix (recommandations de films).
- **Santé** : Diagnostic assisté par IA, robots chirurgicaux.
- **Entreprise** : Automatisation des tâches répétitives, analyse des données clients.
- **Éducation** : Correction automatique, adaptation de l'apprentissage aux besoins des élèves.
- **Industrie** : Robots industriels collaborant avec les humains.

## 6. Paradigmes de programmation

- **Programmation procédurale** : Basée sur des instructions séquentielles.
- **Programmation logique** : Basée sur des règles logiques (ex. Prolog).

# Chapitre 2 : Résolution des problèmes en IA

## 1. Introduction

L'IA cherche à résoudre des problèmes difficiles pour les humains en s'appuyant sur des règles et des stratégies adaptées.

## 2. Principe de la résolution des problèmes en IA

- L'objectif est de concevoir des **méthodes générales** applicables à divers types de problèmes.
- Un algorithme doit être indépendant du domaine concerné.
- La description du problème et la stratégie de résolution doivent être séparées.

## 3. General Problem Solver (GPS)

- Développé en **1959** par **Newell et Simon**.
- Inspiré de la manière dont les humains résolvent des problèmes.
- Fonctionnement : Réduction des différences entre l'état initial et l'objectif.
- Malgré ses ambitions, il s'est limité à la résolution d'énigmes simples.

## 4. Types de problèmes en IA

- **Problèmes de satisfaction de contraintes (CSP)** : Trouver une solution respectant des contraintes (ex. Sudoku, séquençage ADN).
- **Problèmes de planification** : Trouver une séquence d'actions menant à un objectif (ex. emploi du temps, transport, jeux comme le Solitaire).

## 5. Représentation graphique d'un problème

Un problème peut être modélisé sous forme de graphe où :

- Les **nœuds** représentent des **états**.
- Les **arcs** représentent des **opérations** qui transforment un état en un autre.
- La **solution** est un chemin du nœud initial vers un nœud final.

## 6. Représentation d'un problème en IA

Un problème est défini par un triplet **(I, O, B)** :

- **I** : États initiaux.
  - **O** : Ensemble des opérations possibles.
  - **B** : États finaux (solutions acceptables).
- 

### Exercices et Études de cas

#### 1. Problème des 4 reines

- **Objectif** : Placer 4 reines sur un échiquier 4x4 sans qu'elles se menacent.
- **Représentation** :
  - **I** : Échiquier vide.
  - **O** : Placer une reine.
  - **B** : Disposition correcte des reines.

#### 2. Problème du voyageur de commerce

- **Objectif** : Trouver le chemin le plus court reliant plusieurs villes et revenant au point de départ.
- **Représentation** :
  - **I** : Ville de départ.
  - **O** : Déplacement d'une ville à une autre.
  - **B** : Toutes les villes sont visitées une seule fois avant de revenir au point de départ.

#### 3. Problème de la traversée de la rivière

- **Objectif** : Faire traverser un fermier, un loup, un mouton et un chou sans incidents.
- **Contraintes** :
  - Si le loup et le mouton restent seuls, le loup mange le mouton.
  - Si le mouton et le chou restent seuls, le mouton mange le chou.
- **Représentation** :
  - **I** : Tout le monde est sur la rive gauche.
  - **O** : Transport d'un élément à la fois avec le fermier.
  - **B** : Tout le monde est sur la rive droite sans incident.

## Conclusion:

- ✓ L'IA vise à imiter l'intelligence humaine à travers des algorithmes.
- ✓ Elle repose sur des techniques comme l'apprentissage automatique et le traitement du langage naturel.
- ✓ Contrairement aux algorithmes classiques, l'IA utilise des méthodes heuristiques.
- ✓ L'IA peut gérer des informations incomplètes et incertaines.
- ✓ Son développement a progressé depuis les années 50, avec une intégration économique à partir des années 80.
- ✓ Les domaines d'application sont variés : santé, entreprise, éducation, industrie.
- ✓ L'IA cherche à résoudre des problèmes complexes en imitant les stratégies humaines.
- ✓ Le **General Problem Solver** (GPS) est un modèle initial de résolution de problèmes basé sur la réduction des différences entre état initial et but.
- ✓ Il existe deux grands types de problèmes :
  - **Problèmes de satisfaction de contraintes (CSP)** : Ex. Sudoku.
  - **Problèmes de planification** : Ex. emploi du temps, transport.
    - ✓ Un problème est modélisé sous forme de **graphe** (états et transitions).
    - ✓ Il est défini par un triplet **(I, O, B)** :
      - **I** : État initial.
      - **O** : Opérations possibles.
      - **B** : État final recherché.

## Questions-Réponses :

### Chapitre 1 : Introduction à l'IA

#### ? Qu'est-ce que l'Intelligence Artificielle (IA) ?

- ✓ C'est la capacité d'un programme informatique à imiter l'intelligence humaine pour réaliser des tâches complexes.

#### ? Quels sont les principaux domaines de l'IA ?

- ✓ Apprentissage automatique, vision par ordinateur, traitement du langage naturel, détection d'anomalies, exploration des connaissances.

#### ? Quelle est la différence entre IA et informatique classique ?

- ✓ L'IA utilise des heuristiques et apprend des données, tandis que l'informatique classique suit des règles strictes et déterministes.

#### ? Quels sont les types d'apprentissage en IA ?

- ✓ Apprentissage supervisé, non supervisé et par renforcement.

#### ? Quels sont quelques domaines d'application de l'IA ?

- ✓ Santé (diagnostic), entreprises (automatisation), éducation (correction automatique), industrie (robots).

**? Quels sont les paradigmes de programmation en IA ?**

✓ Programmation procédurale et programmation logique.

---

**Chapitre 2 : Résolution des problèmes en IA**

**? Pourquoi l'IA est-elle utilisée pour résoudre des problèmes ?**

✓ Parce que certains problèmes ont trop de possibilités pour être résolus par des méthodes classiques.

**? Qu'est-ce que le General Problem Solver (GPS) ?**

✓ Un modèle de résolution de problèmes basé sur la réduction des écarts entre l'état initial et l'objectif.

**? Quels sont les types de problèmes en IA ?**

✓ Problèmes de satisfaction de contraintes (CSP) et problèmes de planification.

**? Comment représenter un problème en IA ?**

✓ Par un triplet (I, O, B) :

- **I** : état initial,
- **O** : opérations possibles,
- **B** : état final recherché.

**? Comment modéliser un problème en IA ?**

✓ Sous forme de graphe, où les nœuds représentent les états et les arcs les transitions possibles.

**? Quels sont quelques problèmes classiques en IA ?**

✓ Problème des 4 reines, problème du voyageur de commerce, problème de la traversée de la rivière.

# Chapitre 3 : Résolution d'un problème par recherche aveugle

## 1. Introduction aux méthodes non informées (recherche aveugle)

✦ **Définition** : Une recherche **non informée** ne possède aucune connaissance spécifique du domaine (ex. emplacement de l'objectif, proximité, coût).

✦ **Principe** : L'algorithme explore systématiquement l'espace de recherche sans information sur la solution.

✦ **Fonctionnement** : Il génère tous les états possibles jusqu'à atteindre la solution, ce qui le rend coûteux en mémoire et en temps.

💡 **Autre nom** : Recherche aveugle (Blind Search).

---

## 2. Algorithme général des méthodes non informées

L'algorithme suit ces étapes :

1 **Initialiser** l'arbre ou le graphe de recherche avec l'état initial.

2 **Vérifier** s'il reste des nœuds à explorer.

3 **Sélectionner** un nœud pour l'expansion (selon la stratégie choisie).

4 **Tester** si ce nœud est une solution :

- ✓ Si oui → retourner la solution.
  - ✗ Sinon → générer ses successeurs et les ajouter à la structure de recherche.
- 5 **Répéter** jusqu'à trouver la solution ou ne plus avoir de nœuds à explorer.

## 4. Recherche en largeur (BFS - Breadth First Search)

✦ **Stratégie** : Explore tous les nœuds d'un niveau avant de passer au suivant.

✦ **Structure utilisée** : File d'attente (FIFO)

✦ **Avantages** :

✓ Garantie de trouver la solution optimale (si coût uniforme).

✓ Fonctionne bien pour des solutions proches de l'état initial.

✦ **Inconvénients** :

✗ Consomme beaucoup de mémoire (stocke tous les nœuds).

✗ Lent si la solution est profonde.

---

## 5. Recherche en profondeur (DFS - Depth First Search)

◆ **Stratégie** : Suit un chemin jusqu'à la profondeur maximale avant de revenir en arrière.

◆ **Structure utilisée** : Pile (LIFO)

◆ **Avantages** :

✓ Moins gourmand en mémoire que BFS.

✓ Peut être plus rapide si la solution est profonde.

◆ **Inconvénients** :

✗ Peut explorer des chemins inutiles.

✗ Ne garantit pas toujours la solution optimale.

✗ Risque de boucles infinies sur des graphes sans borne.

---

## 6. Comparaison BFS vs DFS

Critère	BFS (Largeur)	DFS (Profondeur)
Exploration	Niveau par niveau	En profondeur d'abord
Structure	File d'attente (FIFO)	Pile (LIFO)
Solution optimale	✓ Oui (si coût uniforme)	✗ Pas garanti
Mémoire	✗ Grande (stocke tous les nœuds)	✓ Faible (stocke seulement les branches explorées)
Temps	✗ Lent si la solution est loin	✓ Plus rapide si la solution est en profondeur
Boucles infinies	✓ Pas de risque	✗ Risque élevé

---

## Conclusion

✓ **Les recherches non informées** sont utiles quand on n'a pas d'information sur l'objectif.

✓ **BFS** est sûr mais gourmand en mémoire.

✓ **DFS** est rapide mais peut être inefficace si mal utilisé.

✓ D'autres méthodes comme l'approfondissement itératif ou la recherche bidirectionnelle permettent d'optimiser l'exploration.

✓ **Recherche non informée** → Aucune connaissance sur la solution, explore tout l'espace de recherche.

✓ **Méthodes non informées** → Fonctionnent sans heuristique, juste en testant tous les états possibles.



- ✓ **Algorithme général** → Explore les nœuds un par un selon une stratégie jusqu'à trouver la solution.
- ✓ **BFS (Recherche en largeur)** → Explore niveau par niveau, **garantit la solution optimale** mais consomme beaucoup de mémoire.
- ✓ **DFS (Recherche en profondeur)** → Suit un chemin jusqu'au bout avant de revenir en arrière, **économique en mémoire** mais peut explorer inutilement.
- ✓ **DFS limitée** → Comme DFS, mais avec une profondeur maximale pour éviter les explorations infinies.
- ✓ **Approfondissement itératif** → Combine BFS et DFS en augmentant progressivement la profondeur maximale.
- ✓ **Coût uniforme** → Explore toujours le chemin le moins coûteux en premier.
- ✓ **Recherche bidirectionnelle** → Recherche simultanée depuis l'état initial et l'état final, plus rapide si applicable.

## Questions-Réponses :

### ? Qu'est-ce qu'une recherche non informée ?

- ✓ Une méthode de recherche qui ne possède aucune information sur la proximité de la solution et explore tous les états possibles.

### ? Pourquoi appelle-t-on ces méthodes "recherche aveugle" ?

- ✓ Parce qu'elles ne tiennent pas compte de l'emplacement de l'objectif et explorent l'espace de recherche de manière exhaustive.

### ? Quels sont les principaux types de recherche non informée ?

- ✓ Recherche en largeur (BFS), recherche en profondeur (DFS), recherche en profondeur limitée, approfondissement itératif, recherche par coût uniforme, recherche bidirectionnelle.

### ? Quelle est la structure de données utilisée pour chaque méthode ?

- ✓ **BFS** → File d'attente (FIFO).
- ✓ **DFS** → Pile (LIFO).
- ✓ **Coût uniforme** → File de priorité.
- ✓ **Bidirectionnelle** → Deux arbres de recherche.

---

### ? Comment fonctionne la recherche en largeur ?

- ✓ Explore tous les nœuds d'un niveau avant de passer au suivant.

### ? Quels sont les avantages de BFS ?

- ✓ Trouve toujours la solution optimale (si le coût des actions est uniforme).

### ? Quels sont les inconvénients de BFS ?

- ✓ Consomme beaucoup de mémoire car elle stocke tous les nœuds explorés.

---

### **? Comment fonctionne la recherche en profondeur ?**

- ✓ Explore un chemin jusqu'à la profondeur maximale avant de revenir en arrière.

### **? Quels sont les avantages de DFS ?**

- ✓ Consomme moins de mémoire que BFS.
- ✓ Peut être plus rapide si la solution est en profondeur.

### **? Quels sont les inconvénients de DFS ?**

- ✓ Ne garantit pas la solution optimale.
- ✓ Peut tomber dans des boucles infinies si le graphe est infini.

---

### **? Pourquoi utiliser la recherche en profondeur limitée ?**

- ✓ Pour éviter les boucles infinies et limiter l'exploration à une certaine profondeur.

### **? Comment fonctionne l'approfondissement itératif ?**

- ✓ Combine BFS et DFS en explorant progressivement des profondeurs croissantes.

### **? Quels sont les avantages de la recherche par coût uniforme ?**

- ✓ Trouve le chemin le moins coûteux en explorant en priorité les nœuds avec le plus faible coût cumulé.

### **? Comment fonctionne la recherche bidirectionnelle ?**

- ✓ Démarre la recherche à la fois depuis l'état initial et depuis l'état final, les deux explorations se rejoignent au milieu.