

# QCM Intelligence Artificielle

Voici un autre sujet d'examen différent, toujours basé sur ton cours d'Intelligence Artificielle, mais avec un **nouveau QCM** et un **exercice avec les algorithmes génétiques** cette fois-ci.

---



## EXAMEN D'INTELLIGENCE ARTIFICIELLE — BAC+2

Durée : 2h — Total : 20 points

---



### PARTIE I — QCM (10 points)

Choisissez la (ou les) bonne(s) réponse(s). Certaines questions peuvent avoir plusieurs bonnes réponses.

---

1. Le Deep Learning est principalement utilisé pour : (1 pt)

- ☐ Créer des bases de données
  - ☐ Résoudre des équations différentielles
  - ☐ Traiter des images, sons ou vidéos
  - ☐ Construire des fonctions trigonométriques
- 

2. Quelle est la principale différence entre **optimisation continue** et **optimisation combinatoire** ? (1 pt)

- ☐ L'optimisation combinatoire travaille sur un ensemble fini et discret
- ☐ L'optimisation continue utilise des entiers uniquement
- ☐ L'optimisation combinatoire ne trouve jamais de solution
- ☐ L'optimisation continue ne peut pas être modélisée mathématiquement

---

3. Les métaheuristiques sont particulièrement utiles pour : (1 pt)

- ☐ Les problèmes à solution triviale
  - ☐ Les problèmes d'optimisation simples
  - ☐ Les problèmes NP-difficiles où les méthodes classiques échouent
  - ☐ Résoudre uniquement des équations algébriques
- 

4. Quelle méthode est inspirée du comportement des colonies d'insectes ? (1 pt)

- ☐ Recuit simulé
  - ☐ Recherche tabou
  - ☐ Algorithmes de colonies de fourmis
  - ☐ Descente de gradient
- 

5. Un problème NP-complet est : (1 pt)

- ☐ Facile à résoudre avec des équations simples
  - ☐ Résoluble rapidement en toutes circonstances
  - ☐ Difficile à résoudre en temps raisonnable quand la taille du problème augmente
  - ☐ Toujours impossible à résoudre
- 

6. Quelle est la condition pour qu'une solution soit un **optimum global** dans un problème de minimisation ? (1 pt)

- ☐ Elle est trouvée au hasard
  - ☐ Elle satisfait  $\forall s \in S, f(s^*) \leq f(s)$
  - ☐ Elle ne dépend pas des contraintes
  - ☐ Elle maximise la fonction coût
-

7. L'algorithme de sélection **par tournoi** dans un AG (algorithme génétique) : (1 pt)

- ☐ Sélectionne toujours le plus faible
  - ☐ Ne prend jamais les meilleurs
  - ☐ Sélectionne parmi un sous-ensemble d'individus les plus adaptés
  - ☐ Est moins influencé par les valeurs extrêmes que la roulette
- 

8. Quel est le rôle de la mutation dans les algorithmes génétiques ? (1 pt)

- ☐ Elle détruit la population
  - ☐ Elle augmente la diversité génétique
  - ☐ Elle est obligatoire à chaque génération
  - ☐ Elle n'est jamais utilisée
- 

9. Un algorithme génétique fonctionne généralement avec : (1 pt)

- ☐ Une seule solution unique
  - ☐ Une population de solutions qui évolue
  - ☐ Des calculs exacts uniquement
  - ☐ Des itérations sans aucune règle
- 

10. Le rôle de la **fonction de performance (fitness)** est de : (1 pt)

- ☐ Générer aléatoirement des individus
  - ☐ Évaluer la qualité/adaptation d'un individu
  - ☐ Calculer la température
  - ☐ Créer des mutations
- 



## PARTIE II — EXERCICE (10 points)

Sujet : Algorithme génétique simple pour un problème de minimisation

---

On veut **minimiser** la **fonction suivante** :

Nombre de bits à 1 dans un individu binaire (fonction  $w_i = \sum x_i$ )

### Données : Population initiale

python

```
population = [  
    [1, 0, 0, 1, 1, 1],  
    [0, 1, 0, 0, 1, 0],  
    [1, 1, 1, 0, 1, 1],  
    [1, 0, 1, 1, 0, 0],  
    [0, 0, 1, 0, 0, 0],  
    [0, 1, 0, 1, 1, 1]  
]
```

### Objectifs :

1. Écrire une fonction `fitness(individu)` qui retourne le nombre de bits à 1. (2 pts)
2. Implémenter une sélection par **tournoi** pour choisir les parents. (2 pts)
3. Appliquer un **croisement à un point** entre deux parents. (2 pts)
4. Appliquer une **mutation** avec une probabilité de 0.2 par bit. (2 pts)
5. Générer une **nouvelle population**, et afficher l'individu le plus adapté (ayant le moins de 1 ). (2 pts)

---

Souhaitez-tu que je te donne **le corrigé Python** de cette partie pratique aussi ?