

**✔️Question 1: Pseudo-code de chaque algorithme génétique pour le TSP**

Algorithme génétique standard pour le TSP:

Début

Générer une population initiale de solutions (tournées)

Évaluer chaque solution en calculant son coût total

Répéter jusqu'à ce qu'un critère d'arrêt soit atteint:

Sélectionner les parents pour la reproduction

Appliquer les opérateurs de reproduction (crossover) et mutation pour créer une nouvelle génération

Évaluer chaque solution de la nouvelle génération

Remplacer la population actuelle par la nouvelle génération

Retourner la meilleure solution trouvée

Algorithme génétique avec élitisme pour le TSP:

Début

Générer une population initiale de solutions (tournées)

Évaluer chaque solution en calculant son coût total

Répéter jusqu'à ce qu'un critère d'arrêt soit atteint:

Sélectionner les parents pour la reproduction (y compris les meilleures solutions de la génération précédente)

Appliquer les opérateurs de reproduction (crossover) et mutation pour créer une nouvelle génération

Évaluer chaque solution de la nouvelle génération

Remplacer la population actuelle par la nouvelle génération, en conservant les meilleures solutions de la génération précédente

Retourner la meilleure solution trouvée

**✔️Question 2: Explication de l'utilité des différents paramètres de l'algorithme génétique**

Les algorithmes génétiques comportent plusieurs paramètres qui affectent leur fonctionnement et leur performance. Voici quelques-uns des paramètres les plus importants :

Taille de la population : le nombre de solutions dans la population. Une plus grande population peut aider à éviter de rester coincé dans des optima locaux, mais augmente également le temps de calcul.

Taux de mutation : la probabilité qu'un gène soit modifié au cours de la reproduction. Un taux de mutation élevé peut aider à diversifier la population, mais peut également conduire à une exploration inefficace de l'espace de recherche.

Taux de croisement : la probabilité qu'un croisement se produise lors de la reproduction. Un taux de croisement élevé peut conduire à une convergence plus rapide vers la solution optimale, mais peut également conduire à la perte de diversité dans la population.

Critère d'arrêt : la condition qui détermine quand l'algorithme doit s'arrêter. Il peut s'agir d'un nombre maximal d'itérations, d'un temps d'exécution maximal, d'une certaine valeur de la fonction objectif, etc.

Fonction de sélection des parents : la méthode utilisée pour sélectionner les parents pour la reproduction. Les méthodes populaires incluent la sélection par roulette, la sélection par tournoi, la sélection par rang, etc.

Elitisme : la pratique de conserver les meilleures solutions de la génération précédente dans la nouvelle génération. Le fait de conserver les meilleures solutions peut aider à maintenir un niveau élevé de qualité de la population au fil du temps.

✔️Question 3: Application de l'algorithme génétique pour résoudre le problème du voyageur de commerce

Pour utiliser l'algorithme génétique pour résoudre le problème du voyageur de commerce, nous pouvons représenter chaque solution candidate comme une permutation des villes à visiter. Ainsi, chaque chromosome de la population représente une tournée possible.

Nous pouvons définir une fonction d'évaluation qui calcule la longueur de chaque tournée et utilise cette mesure pour évaluer la qualité de chaque solution candidate. Ensuite, nous pouvons utiliser les opérateurs génétiques de croisement et de mutation pour générer une nouvelle population de solutions à partir de la génération précédente.

L'algorithme d'élite peut être particulièrement utile pour ce problème, car il permet de conserver les meilleures solutions de chaque génération, augmentant ainsi la probabilité de converger vers une solution optimale.

En fin de compte, l'algorithme génétique peut être utilisé pour trouver une solution approximative au problème du voyageur de commerce en utilisant une approche heuristique qui évolue au fil du temps. Cependant, il est important de noter que l'algorithme génétique peut ne pas garantir la solution optimale en raison de la nature heuristique de l'approche.

Une variante de l'algorithme génétique appelée "métaheuristique de colonies de fourmis" (Ant Colony Optimization) peut également être utilisée pour résoudre le problème du voyageur de commerce. Cette approche est inspirée du comportement collectif des colonies de fourmis dans leur recherche de nourriture.

Dans cette approche, chaque fourmi représente une solution candidate, et chaque ville est associée à une phéromone qui est déposée par les fourmis lorsqu'elles visitent une ville. Les fourmis sont attirées par les phéromones plus fortes, ce qui leur permet de découvrir des solutions prometteuses.

À mesure que les fourmis visitent les villes et déposent des phéromones, les chemins les plus courts accumulent une plus grande quantité de phéromones, attirant ainsi davantage de fourmis et augmentant la probabilité de trouver une solution optimale.

En fin de compte, les algorithmes génétiques et les métaheuristiques de colonies de fourmis sont des approches efficaces pour résoudre le problème du voyageur de commerce. Cependant, le choix de l'approche appropriée dépendra de la complexité du problème et des ressources disponibles pour l'optimisation.

Exemple :

Problème : Un voyageur doit visiter 3 villes (A, B, C) une seule fois et revenir à sa ville de départ (A) en parcourant la distance la plus courte possible. Les distances entre les villes sont les suivantes :

A B C

A 0 2 3

B 2 0 4

C 3 4 0

Solution : Utiliser l'algorithme génétique pour trouver une solution approchée.

Pseudo-code de l'algorithme génétique pour le problème du voyageur de commerce :

Générer une population initiale de chemins, chaque chemin représentant une permutation des villes à visiter. Par exemple : [A, B, C] et [A, C, B].

Evaluer la distance totale de chaque chemin en utilisant la distance euclidienne entre chaque paire de villes. Par exemple, pour le chemin [A, B, C] la distance totale est : 2 + 4 + 3 = 9.

Tant que le critère d'arrêt n'est pas atteint (par exemple, le nombre maximal d'itérations est atteint) :

a. Sélectionner deux parents pour la reproduction en utilisant une méthode de sélection (par exemple, la sélection par roulette ou la sélection par tournoi).

b. Effectuer la reproduction pour créer une nouvelle génération de chemins en utilisant une méthode de croisement (par exemple, le croisement en ordre).

c. Appliquer une méthode de mutation pour perturber certains chemins sélectionnés au hasard (par exemple, la mutation par inversion).

d. Évaluer la distance totale de chaque nouveau chemin créé.

e. Appliquer une méthode de remplacement pour remplacer la génération précédente par la nouvelle (par exemple, l'élitisme).

Retourner le chemin le plus court trouvé.