**Lisez seulement abeg… (j’aimerai me rassurer être à la ligne)**

Si le problème est le CVRPTW (Capacitated Vehicle Routing Problem with Time Windows), qui est une extension du VRP avec des fenêtres de temps, voici une approche générale pour le résoudre à l'aide d'un algorithme génétique :

1. Modélisation :
   * Définir une représentation des individus de la population, qui peut être une séquence ordonnée des clients à visiter pour chaque camion, ainsi que les plages horaires de visite pour chaque client.
   * Définir une fonction d'évaluation qui mesure la qualité d'une solution en termes de distance totale parcourue, de respect des contraintes de capacité des véhicules et des fenêtres de temps de visite.
   * Définir les opérations génétiques telles que la sélection, le croisement et la mutation pour générer de nouvelles solutions.
2. Génération de la population initiale :
   * Générer une population initiale de solutions en respectant les contraintes de capacité des véhicules, les fenêtres de temps de visite, etc.
3. Boucle principale de l'algorithme génétique :
   * Évaluer la qualité de chaque solution dans la population.
   * Sélectionner les meilleurs individus pour la reproduction.
   * Appliquer les opérations génétiques (croisement et mutation) pour générer de nouvelles solutions.
   * Évaluer à nouveau la qualité des nouvelles solutions.
   * Remplacer l'ancienne population par la nouvelle.
   * Répéter ces étapes jusqu'à atteindre une condition d'arrêt prédéfinie.
4. Contraintes spécifiques au CVRPTW :
   * Pour gérer les fenêtres de temps de visite, nous pouvons ajuster les opérateurs génétiques pour respecter les contraintes temporelles. Par exemple, lors du croisement, nous pouvons échanger les séquences de clients entre les camions tout en conservant l'ordre temporel des visites.
   * Pour gérer la capacité des véhicules, nous pouvez définir des contraintes de capacité pour chaque véhicule et utiliser des opérateurs génétiques spécifiques pour l'affectation des clients et la planification des tournées pour chaque véhicule.
5. Étude statistique :
   * Comme pour le VRP, effectuer une étude statistique du comportement de l'algorithme en exécutant plusieurs instances du CVRPTW avec différentes configurations et tailles.
   * Mesurer des métriques telles que la qualité de la solution (distance totale, respect des contraintes temporelles), le temps de convergence, le nombre d'itérations nécessaires, etc.
   * Analyser les résultats pour évaluer les performances de l'algorithme et déduire des informations prédictives sur son comportement pour des cas d'utilisation plus vastes.

Nous pouvons adapter ces étapes en fonction de nos spécifications précises et des contraintes du CVRPTW que nous souhaitons inclure. Des bibliothèques Python comme DEAP peuvent également être utiles pour l'implémentation de l'algorithme génétique pour le CVRPTW.

**Livrable**

Voici une structure de base du livrable technique :

1. Fichiers de données :

- Inclure les fichiers d'instances de problème utilisés pour nos tests en incluant des exemples de différentes tailles et complexités.

- Prédisposer des valeurs optimales ou de bornes inférieures pour ces instances, si possible.

- Inclure le code correspondant au générateur aléatoire pour créer des instances.

2. Code Python de l'algorithme :

- Code source commenté de notre algorithme de résolution du CVRPTW.

3. Codes pour l'étude expérimentale :

- Code source commenté permettant de réaliser l'étude expérimentale de notre algorithme.

- Cela peut inclure le calcul des paramètres des graphes, des bornes inférieures, ainsi que des statistiques descriptives et prédictives. (Kevin🤨)

4. Script de tests de montée en charge :

- Effectuer des tests de montée en charge pour évaluer les performances de notre algorithme sur des instances de grande taille, et inclure le script correspondant. (L’évolution dont parlait Faris & Kevin)

- Préciser les paramètres utilisés pour les tests, tels que la taille de l'instance, le nombre de camions, les contraintes, etc.

**Livrable rédigé (Notebook)**

- Ensemble de notre travail et des résultats obtenus dans un notebook jupi\_chouuu.

- Inclure des visualisations graphiques des résultats et des statistiques obtenues.

- Expliquer chaque étape de notre démarche et fournir des explications détaillées des choix effectués.

- Ajouter des commentaires et des explications pour rendre le notebook clair et compréhensible.

Archiver le livrable technique complet (tous les fichiers et codes). Pour le livrable rédigé (pas le Notebook), envoyer au format PDF pour **Faris** ne modifie rien.

Merci. A Jeudi.😘