À remettre : vendredi 22 décembre 2023, 23:59

(English version below)

Le second rendu comptant pour le CC de l'UE Algorithmique Avancée porte donc sur le voyageur de commerce. On vous demande de rendre :

- votre implémentation d'une recherche locale, avec liste tabou, pour résoudre ce problème ;
- votre programme linéaire en nombres entiers, en langage *ZIMPL*, pour résoudre ce problème ;
- un petit rapport, de 2 à 3 pages, décrivant le travail que vous avez réalisé et les résultats obtenus sur les instances proposées.

Vous devez déposer dans le devoir ci-dessous :

- votre rapport au format pdf, non compressé et non archivé ;
- une archive contenant vos deux programmes (MH et PLNE)
  - le programme ZIMPL - un seul fichier .zpl, assez générique pour qu'on puisse facilement changer le nom du fichier d'instance
  - votre programme pour la recherche avec métaheuristique, non compilé (les fichiers source donc), avec un "readme" indiquant comment on peut tester votre programme.
- ATTENTION : l'archive ne doit pas contenir le rapport, **le rapport au format pdf doit être déposé *en dehors de l'archive*** ; le non-respect de cette consigne entraînera une pénalité.

Il y a sur Moodle 4 instances du problème de voyageur de commerce, avec respectivement 5, 25, 50 et 101 points de passage.

Votre rapport devra décrire :

1. Pour la recherche avec métaheuristique :
   - le langage utilisé ;
   - quelles méthodes vous avez implémentées (recherche selon la plus grande pente, avec - ou non - liste tabou) ;
   - le(s) voisinage(s) implémenté(s) ;
2. Pour le programme ZIMPL :
   - le modèle mathématique (variables, objectif, contraintes)
3. Les résultats :
   - la meilleure solution obtenue sur l'instance tsp50.txt (avec l'une ou l'autre méthode) : la séquence des points de passage, et la distance parcourue ;
   - un tableau récapitulant les résultats, donnant, pour chacune des 4 instances / fichiers :
     - pour le programme ZIMPL avec SCIP : si optimum atteint, ou sinon écart à l'o[illegible]num (en %), temps de calcul, valeur de la meilleure solution trouvée ;
     - pour votre implémentation de MH : temps de calcul, valeur de la meilleure solution trouvée, éventuellement en différenciant suivant les voisinages / algos implémentés

The second coursework for the Advanced Algorithms CC is on the travelling salesman problem. You are asked to submit:

- your implementation of a local search, with tabu list, to solve this problem ;
- your integer linear program, in the *ZIMPL* language, to solve this problem;
- a short report, 2-3 pages, describing the work you've done and the results obtained on the proposed instances.

You must submit the following below:

- your report in pdf format, uncompressed and unarchived ;
- an archive containing your two programs (MH and PLNE)
  - the ZIMPL program - a single .zpl file, generic enough to allow easy renaming of the instance file
  - your metaheuristic search program, uncompiled (i.e. the source files), with a "readme" indicating how your program can be tested.

- BEWARE: the archive must not contain the report; **the report in pdf format must be filed *outside the archive***; failure to do so will result in a penalty.

There are 4 instances of the travelling salesman problem on Moodle, with 5, 25, 50 and 101 points respectively.

Votre rapport devra décrire :

1. For search with metaheuristics :
    - the language used ;
    - which methods you have implemented (steepest hill descent, with - or without - tabu list);
    - the neighborhood(s) implemented;
2. For the ZIMPL program :
    - mathematical model (variables, objective, constraints)
3. Results:
    - the best solution obtained on the tsp50.txt instance (using either method): the sequence of points, and the distance covered;
    - a table summarizing the results, giving, for each of the 4 instances / files :
        - for the ZIMPL program with SCIP: if optimum reached, or if not, deviation from optimum (in %), calculation time, value of best solution found ;
        - for your MH implementation: calculation time, value of the best solution found, possibly differentiating according to the neighborhoods / implemented algorithms

---

Ajouter un travail

Statut de remise

Statut des travaux remis	Aucun devoir n'a encore été remis
Statut de l'évaluation	Non évalué
Temps restant	1 jour 14 heures restants
Dernière modification	-
Commentaires	Commentaires (0)