Métaheuristiques sur QAP

1 Description

Le problème d'assignement quadratique (Quadratic Assignemnt Problem - QAP) est un problème d'optimisation combinatoire décrit dans le cours précédant. Vous avez déjà réalisé (ou vous êtes sur le point de terminer) un algorithme évolutionnaire résolvant ce problème. Dans ce TP, vous allez réaliser les métaheuristiques Hill climbing, recherche taboue et mimetic algorithm afin de résoudre ce problème.

La description précise des métaheuristiques utilisées est issue de l'article de recherche :
"A New Genetic Algorithm for the Quadratic Assignment Problem", Zvi Drezner, INFORMS Journal on Computing, vol. 15-3, 2003, 320–330

2 But

Comparer les performances des différentes métaheuristiques dans la résolution de ce problème et comparer ces résultats aux résultats obtenus par Drezner.

3 Métaheuristiques

Nous allons réaliser les métaheuristiques à l'aide de la bibliothèque EO que vous avez déjà utilisé. Une partie Moving Object (MO) est spécialement dédiée à cet effet. Télécharger la partie MO à l'adresse : http://www.lifl.fr/ cahon/paradisEO/mo/index.html

3.1 Hill Climbing

Le programme principal se situe dans le fichier hillClimbingQAP.cpp.

Vous devez complèter les classes :

- *IncrEvalQAP* dans le fichier incrEvalQAP.h qui réalise l'évaluation incrémentale. Pour cela, utiliser la formule 2 p.324 de l'article.
- *InitPermMove* dans le fichier initPermMove.h qui initialise le mouvement pour la recherche dans le voisinage des solutions. Pour cela inspirez-vous, des exemples donnés dans MO (mo/examples/tsp/share/two_opt_init.*).
- *NextPermMove* dans le fichier nextPermMove.h qui calcule le mouvement suivant pour l'exploration du voisinage des solutions. Pour cela inspirez-vous, des exemples donnés dans MO (mo/examples/tsp/share/two_opt_next.*).

3.2 Recherche taboue

Le programme principal se situe dans le fichier tabouQAP.cpp.

Vous devez complèter les classes :

- *TabuListQAP* dans le fichier tabuListQAP.h qui implémente la liste taboue. Pour cela, utiliser le paragraphe 2.2.2 p.323 (The simple Tabu Search) de l'article and aussi les exemples de MO.
- *AspirCritQAP* dans le fichier AspirCritQAP.h qui implémente un critère d'aspiration de la recherche taboue. Le méchanisme d'aspiration est expliqué au 2(b) de l'algorithme du paragraph 2.2.2.

3.3 Algorithme mimétique

Un algorithme mimétique est un algorithme qui combine une métaheuristique manipulant une unique solution (HC, TS, SA...) avec un algorithme évolutionnaire. L'association des métaheuristiques a pour but d'obtenir les avantages de chaque méthode.

Pour obtenir un algorithme mimetique, il vous suffit de remplacer la mutation de l'algorithme évolutionnaire (MutationQAP) par l'une des métaheuristiques ci-dessus (hill_climb ou tabu_search).

4 Travail Demandé

1 Implémenter les *trois* métaheuristiques : Hill-Climbing, recherche taboue, et algorithme mimetique.

2 Comparer les trois métaheuristiques par l'exécution d'un grand de fois chaque métaheuristique sur les instances de QAP proposés par Drezner.
Vous présenterez (au moins) la moyenne, l'écart type, la valeur minimale et maximale.

3 Analyser les résultats par rapport à ceux de l'article de Drezner, conclure.