TP2 outils d'Aide à la Décision Heuristiques pour le Job-Shop

Duhamel - Lacomme

ISIMA 2ème année F2F3, 2 séances, **noté**

1. Se Connecter sur la OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html). Dans la section Job-Shop, accéder aux instances classiques pour le Job-Shop (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt). À titre d'exemple, voilà ce qu'il est possible d'obtenir, pour les instances La01-La20 :

							(Rego and Duarte, 2008)
Instances	n_j	n_m	n	Opt.	S*	Gap	Time
LA01	10	5	50	666	666	0.00	1
LA02	10	5	50	655	655	0.00	2
LA03	10	5	50	597	597	0.00	1
LA04	10	5	50	590	590	0.00	1
LA05	10	5	50	593	593	0.00	1
LA06	15	5	75	926	926	0.00	1
LA07	15	5	75	890	890	0.00	1
LA08	15	5	75	863	863	0.00	1
LA09	15	5	75	951	951	0.00	1
LA10	15	5	75	958	958	0.00	1
LA11	20	5	100	1222	1222	0.00	1
LA12	20	5	100	1039	1039	0.00	1
LA13	20	5	100	1150	1150	0.00	1
LA14	20	5	100	1292	1292	0.00	1
LA15	20	5	100	1207	1207	0.00	1
LA16	10	10	100	945	947	0.21	1
LA17	10	10	100	784	784	0.00	2
LA18	10	10	100	848	848	0.00	1
LA19	10	10	100	842	846	0.48	1
LA20	10	10	100	902	917	1.66	3
Avg.						**0.12**	**1.20**

2. Travail à faire : réaliser un programme (C, C++ ou Java) comportant au minimum les fonctionnalités suivantes :

   a. lecture d'un fichier Laxxx et d'une séquence S (cf. le cours) pour effectuer l'affichage de la solution et du *makespan*

   b. écriture d'une méthode nommée évaluer() ayant comme paramètre d'entrée un vecteur par répétition (vecteur de Bierwith) comme vu en cours. Cette procédure oriente le graphe en fonction du vecteur et calcule le plus long chemin. La procédure retourne : les dates de débuts et toutes les opérations et le plus long chemin.

   c. écriture d'une méthode nommée rechercheLocale() ayant comme paramètre d'entrée un vecteur par répétition et un nombre maximal d'itérations. Cette procédure réalise une amélioration de type descente (seules les modifications strictement améliorantes sont réalisées) en utilisant la notion de **blocks**. Elle retourne le nouveau vecteur.

On précisera dans le rapport (cf. plus loin) quelles opérations sont réalisées sur le vecteur pour inverser un arc dans le graphe.

d. écriture d'une procédure testerDouble() qui permet de s'assurer qu'une solution n'a pas déjà été envisagée

e. écriture d'une procédure algorithmeGénétique() pour implanter un algorithme génétique

3. Travail à rendre : un compte rendu d'environ 10 pages, le source des programmes réalisés et tous les scripts / instances / fichiers supplémentaires utilisés pour obtenir les résultats. Le compte-rendu comprendra

   a. un descriptif des points nécessaires à l'optimisation d'un Job-Shop

      i. évaluation d'un graphe

      ii. génération d'un graphe à partir d'une séquence

      iii. conception d'une recherche locale efficace

      iv. conception d'un algorithme mémétique

   b. une description algorithmique des points précédents

      i. procédure évaluer()

      ii. procédure rechercheLocale()

      iii. procédure algorithmeGénétique()

      iv. procédure testerDouble()

   c. une étude portant sur 10 séquences pour montrer la pertinence de la recherche locale et une étude sur la convergence de l'algorithme mémétique