

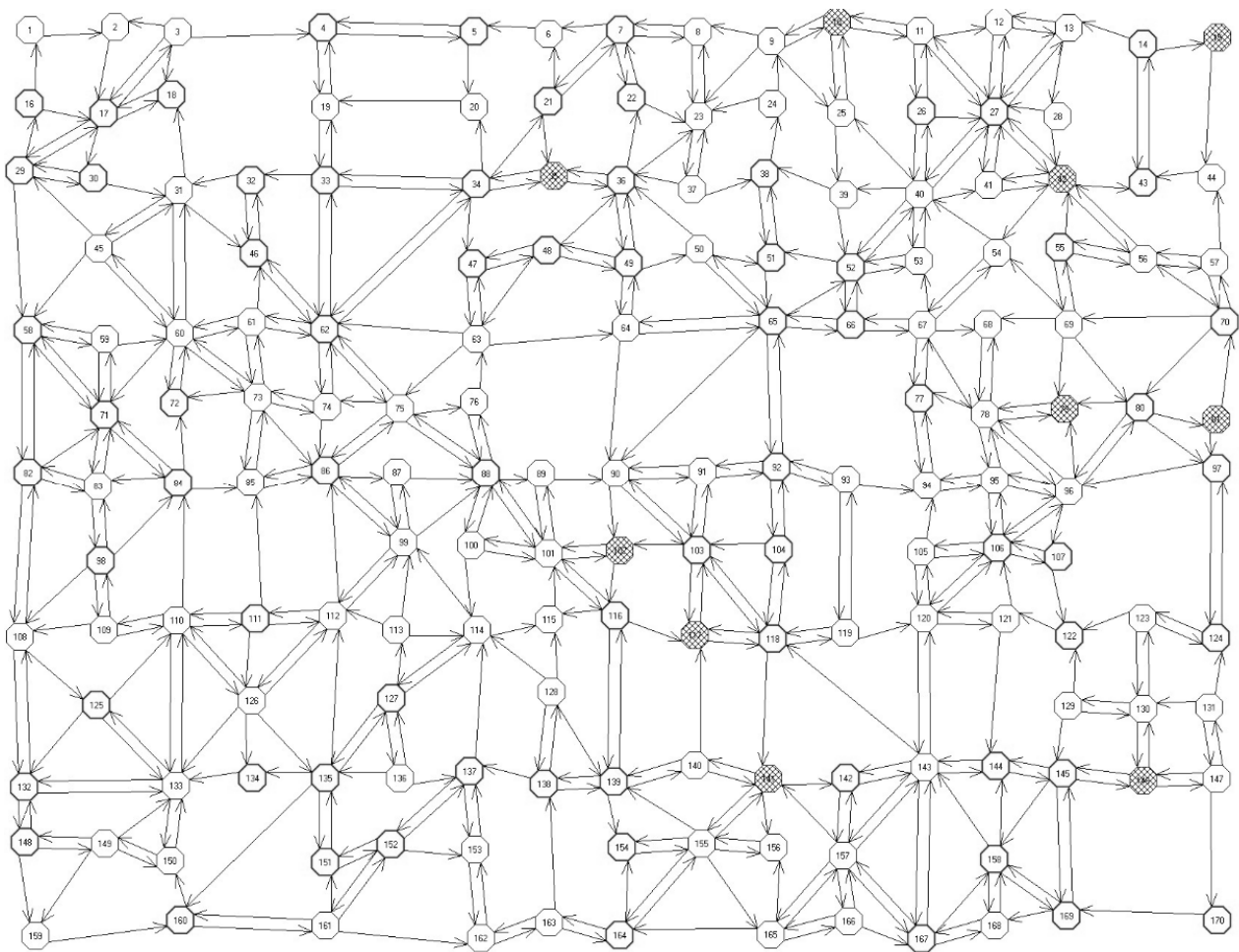
TP1 Outils d'Aide À la Décision

Introduction et Heuristiques pour le Flowshop

Partie 1.

Télécharger le fichier .rar qui se trouve ici : <http://www.isima.fr/~lacomme/TPRO/graphe.rar>

Il s'agit d'un graphe qui ressemble au graphe suivant :



Faire un programme « simple » permettant de lire le graphe et de calculer le plus court chemin du nœud i au nœud j .

Partie 2.

Description : le Flowshop de permutation est un des problèmes d'ordonnancement en atelier les plus simples. On considère n produits devant passer sur m machines, dans le même ordre (par exemple découpe, ébavurage, polissage, traitement de surface, peinture, séchage). Le temps de passage $t_{ij} > 0$ du produit i sur la machine j est spécifique à chaque produit et chaque machine. Deux conditions sont nécessaires pour qu'un produit i sur la machine j puisse passer à

la machine suivante $j+1$: l'opération sur la machine courante j doit être terminée et la machine $j+1$ doit avoir terminé l'opération sur le produit précédent (elle doit être libre). La séquence de passage des produits sur les machines est la même quelle que soit la machine. On cherche donc la séquence des produits optimisant un critère de production. On considèrera ici le critère du *makespan* (durée totale d'activité de l'atelier) que l'on cherchera à minimiser. Pour le Flowshop, le *makespan* correspond à la date de fin de traitement de la dernière machine sur le dernier produit de la séquence, si le premier produit commence sur la première machine à $t = 0$.

Instances : le jeu d'instances le plus connu pour le Flowshop de permutation a été généré par Éric Taillard. Il est disponible à l'adresse suivante : <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html> ou recherche Google sur Éric Taillard. Chaque fichier contient 10 instances de même taille, avec des temps de traitement différents. Pour chaque instance, les deux premières lignes contiennent la description de l'instance (nombre de produits, nombre de machines ainsi qu'une borne inférieure et supérieure sur la *makespan* optimal)

Heuristique : parmi les nombreuses heuristiques proposées, on s'intéressera à l'heuristique NEH (Nawaz, Ensore, Ham) qui repose sur l'algorithme suivant

$$T_i = \sum_{j=1}^m t_{ij}$$

- trier les produits selon leur temps total T_i décroissant
- prendre les deux premiers et les placer dans l'ordre le plus intéressant
- pour chaque produit suivant, l'insérer dans la position la plus intéressante au sens du *makespan*

Travail à réaliser :

1. télécharger les instances
2. générer un code permettant de lire et afficher une instance et d'évaluer une séquence
3. implémenter une heuristique basique ne faisant que la première étape de NEH
4. implémenter NEH
5. générer un script permettant de tester NEH sur toutes les instances
6. produire un tableau reportant, pour chaque instance, les bornes inf et sup, la valeur obtenue par NEH, son temps de calcul et l'écart relatif par rapport à la borne inf.
7. (facultatif) randomiser NEH en choisissant la place de chaque produit au hasard parmi les $\alpha\%$ meilleurs