

Mini-projet de programmation par contraintes

MPRO 2021/2022

Objectif

L'objectif de ce mini-projet est d'amener les étudiants à :

1. Appliquer les connaissances théoriques qu'ils ont acquises dans le cours de PPC,
2. S'approprier l'environnement IBM ILOG CPLEX Optimisation studio sur la base d'exemples simples,
3. Manipuler les leviers classiques permettant d'améliorer les performances des programmes de PPC (modélisation, paramétrage recherche et contraintes, symétries, etc...).
4. Se familiariser, au travers d'un exemple simple, avec l'approche « génération de colonnes » qui est un mécanisme très utilisé dans l'industrie pour la décomposition de problèmes complexes (optionnel)

Trois problèmes sont proposés. Il sera demandé aux étudiants de résoudre avec CPLEX Studio, le **1er et au choix le 2eme ou le 3eme**.

Il leur sera également demandé de répondre à un ensemble de questions relatives à la méthode de résolution de ces différents problèmes.

En cas d'ambiguïté sur les énoncés, les étudiants pourront prendre des hypothèses, les décrire, et résoudre les problèmes sur la base de ces hypothèses.

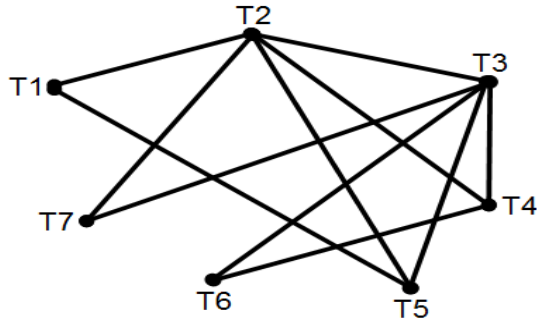
Description des problèmes

Problème 1 : Allocation de fréquences (PPC)

Le but de ce problème est d'affecter des fréquences à un ensemble de 7 émetteurs (transmitters) en évitant les interférences entre les émetteurs à proximité les uns des autres, et en minimisant la fréquence maximum utilisée, en assurant que les émetteurs pairs ont des fréquences paires, et les émetteurs impairs ont des fréquences impaires.

- Chaque fréquence est représentée par un nombre appelé le numéro de canal (chanel). Ce nombre varie entre 1 et 10.
- Les émetteurs pairs [impairs] doivent avoir des fréquences impaires [paires].
- Lorsque deux émetteurs sont à proximité, les fréquences attribuées à ces deux émetteurs doivent respecter les contraintes suivantes:
 - Valeur absolue ($\text{freq}(\text{trans1}) - \text{freq}(\text{trans2})$) $\geq \text{offset}(\text{trans1}, \text{trans2})$
 - où $\text{offset}(\text{trans1}, \text{trans2})$ dépend de trans1 et trans2

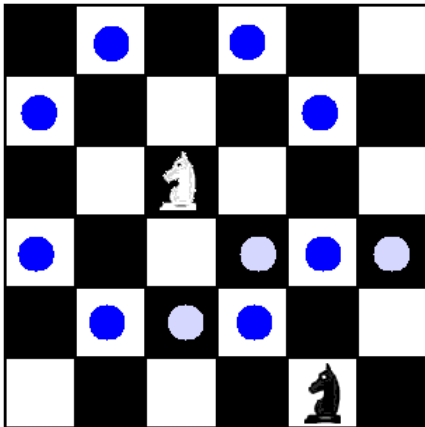
- Voici la table des valeurs d'offset:



Trans1	Trans2	offset
T1	T2	3
T1	T5	2
T2	T3	2
T2	T4	1
T2	T5	2
T3	T4	3
T3	T5	1
T6	T3	1
T6	T4	2
T7	T2	3
T7	T3	2

1. Utilisation des fonctionnalités de base d'OPL pour la recherche et l'optimisation
 - Rechercher une solution faisable
 - Chercher une solution qui minimise la fréquence maximum utilisée (utiliser « minimize »)
2. Utilisation d'OPL script (via fonction main)
 - Rechercher les 10 premières solutions (mettre en place une boucle dans le script OPL).
 - Ecrivez dans le script une procédure de recherche qui minimise la fréquence maximum utilisée (approche de type branch&bound):
 - Dès qu'une solution est trouvée avec un fréquence max N0, recommencer la recherche en limitant la fréquence max à N1=N0-1.
 - Lorsque vous n'aurez plus de solution, cela voudra dire que la dernière solution trouvée était optimale du point de vue de la fréquence max utilisée.

Problème 2 : cavaliers



Le problème consiste à positionner un nombre minimum de cavaliers sur un échiquier de telle sorte que chaque case soit contrôlée (occupée ou menacée) par au moins un cavalier.

Une case est menacée par un cavalier si ce cavalier peut se positionner sur cette case le coup suivant.

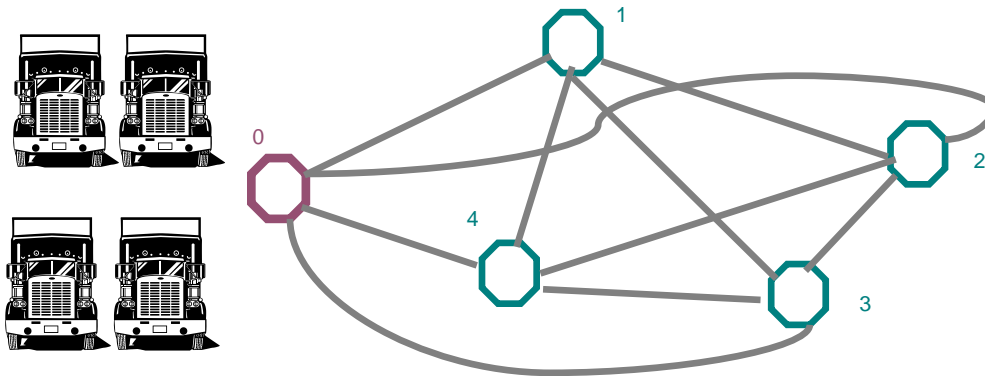
Les cavaliers se déplacent en L sur l'échiquier, sur la base d'un mouvement horizontal [vertical] suivi de deux mouvements verticaux [horizontaux].

Dans l'échiquier ci-après, les points bleus indiquent les cases contrôlées le cavalier blanc et les points blancs les cases contrôlées par le cavalier noir.

Problème 3 : Génération de colonnes (PPC + CPLEX)

Il s'agit d'un exemple simple de routing permettant d'illustrer la génération de colonnes PPC + CPLEX pour sa résolution.

- Soit un ensemble de N clients localisés sur des lieux géographiques distincts
- Soit un réseau routier qui permet de connecter tous ces clients entre eux 2 à 2.
- Chaque client C_i doit être livré de la quantité D_i de produit (même produit pour tous les clients et toutes les demandes)
- Un client ne peut être livré que par un seul camion
- Tous les camions ont la même capacité C
- On recherche le nombre minimal de camions permettant d'assurer l'ensemble des livraisons
- On ne considère pas de notion de distance sur ces routes ni de durée de transport
- Instance considérée :
 - 18 clients
 - Demande pour chaque client [4, 7, 5, 3, 6, 2, 7, 5, 8, 3, 7, 4, 1, 8, 6, 4, 3, 2]
 - Capacité des camions : 10



1. Définir le problème maître (variables, contraintes) et l'implémenter
 - Définir comment vous allez construire une première solution (triviale) et faisable pour initialiser le mécanisme.
2. Définir le sous problème (variables, contraintes) et l'implémenter
 - Trouver une route faisable (satisfaisant la contrainte de capacité)
 - Trouver la route qui a un cout réduit minimum.
3. Construire le script OPL qui implémente l'algorithme de résolution
 - Résolution du problème maître
 - Calcul de la fonction objectif du sous problème et mise à jour dans le modèle correspondant
 - Détermination de la nouvelle colonne (route) et ajout dans le problème maître
 - Poursuite tant qu'il y a des améliorations possibles.

Questions sur les problèmes

1/ Modélisation

Proposer différentes modalisations pour ces problèmes ; expliquer les avantages et inconvénients de chacune. Quels sont les points clés à considérer pour optimiser les temps de recherche de solution sur les problèmes 2 et 3)?

2/ Implémentation

Implémenter les modèles choisis avec IBM ILOG CPLEX Optimisation studio ; tester. Bien séparer les données du modèle (.dat pour les données ; .mod pour les modèles) ; utiliser les facilités de modélisation offertes par le langage pour écrire un modèle le plus concis et plus générique possible.

3/ Recherche de solutions

Essayer différents paramétrages de recherche de solution (fichiers params), et écrire des sélecteurs spécifiques avec les options de searchPhase vues dans la présentation de CPLEX Studio.

Créer des configurations d'exécution pour les différents paramétrages.

Notez les indicateurs de recherche (onglets « Journal du moteur » et statistiques)

Tester, présenter les résultats. Discuter.

4/ Propagation des contraintes

Essayer différents niveaux de propagation des contraintes (fichiers param).

Notez les indicateurs de recherche (onglets « Journal du moteur » et statistiques); Discuter.

5/ Gestion des symétries

Décrire les symétries du problème (pour tous les problèmes lorsqu'il y a lieu).

Proposer des modèles ou des contraintes additionnelles qui permettent de réduire les symétries.

Rendus

Lors de la dernière séance, vous ferez une mini-soutenance et vous rendrez un rapport de 5 pages maximum présentant les modèles utilisés en expliquant vos choix, les résultats obtenus, et les réponses aux différentes questions posées. Les codes sources de vos implémentations seront également à fournir.