Projet
Problème de localisation de machines et d'ordonnancement de tâches avec délai de livraison et date d'échéance

Version du : 27 septembre 2023

Description du problème

Le problème considéré dans ce projet est un problème d'optimisation de la localisation de machines puis de l'affection et de l'ordonnancement de tâches sur ces machines. Deux exemples d'applications sont les suivants :

- opérations portuaires de conteneurs : les navires (machines) doivent être affectés à des postes d'amarrage (localisations), et les séquences de chargement ou de déchargement des conteneurs (tâches) doivent être déterminées
- logistique : un nombre déterminé de centres de distribution (machines) doivent être placées en différents points (localisations) du territoire à couvrir, et les zones de demande ou clients (tâches) sont affectés à ces centres

Soit $\mathcal{K}$ l'ensemble des sites possibles pour localiser les m machines qui doivent être installées ($m \leqslant |\mathcal{K}|$). Chaque site $k \in \mathcal{K}$ peut accueillir au plus une machine pour un coût égal à c_k. Un ensemble $\mathcal{J}$ de tâches doivent être affectées et ordonnancées sans préemption sur ces machines. Chaque machine peut exécuter au plus une tâche à la fois. Chaque tâche $j \in \mathcal{J}$ a une durée p_j (en nombre entier de périodes de temps). Toutes les tâches sont positionnées dans une certaine localisation dès le début de l'horizon de planification. Si une tâche j est affectée à une machine située sur le site $k \in \mathcal{K}$, alors un temps de transport noté r_{jk} et un coût e_{jk} sont à considérer. On suppose ici que r_{jk} et e_{jk} sont fonction de la distance D_{jk} (en km) entre la localisation de j au début de l'horizon de planification et le site k, c'est à dire, $r_{jk} = D_{jk}/u_{jk}$ et $e_{jk} = D_{jk}f_{jk}$ avec u_{jk} la vitesse de transport (km par période de temps) et f_{jk} le coût de transport (€/km). Quand une tâche j est terminée, elle doit être retransportée vers sa localisation d'origine avec une date d'échéance égale à d_j. Si une tâche est de retour à une date $t > d_j$, alors cela entraîne un coût de retard égal à $t - d_j$. L'objectif de ce problème est de minimiser la somme pondérée du coût d'installation des machines sur les différent sites (poids égal à λ_1), du coût de transport des tâches (poids égal à λ_2) et du coût des retards (poids égal à λ_3).

Comme en ordonnancement, on notera C_j la date de fin d'une tâche j et T_j son retard par rapport à la date d'échéance (c'est à dire $T_j = \max\{0, C_j - d_j\}$).

Travail à faire

Etude et modélisation du problème

Question 1. En utilisant des résultats de cours, pourquoi le problème est-il NP-difficile ?
Indice : Il est possible de trouver un problème NP-difficile étudié en cours comme étant un cas particulier.

Question 2. Donner deux formulations du problème comme un programme linéaire en nombres entiers. L'un des modèles devra utiliser des variables indicées sur les périodes et l'autre devra s'en passer.
Expliquer vos deux modèles en détail (signification des variables et des contraintes) et veiller à écrire des contraintes sous la forme la plus forte.

Définition d'une méthode exacte de résolution du problème

Le but de cette partie est de construire deux méthodes exactes de résolution du problème plus efficace que la résolution d'un modèle PLNE par un solveur. Une première méthode est basée sur la décomposition de Benders classique et l'autre sur une adaptation de cette méthode pour la rendre plus efficace pour la résolution du problème décrit (cette méthode parfois appelée décomposition de Benders combinatoire ou décomposition de Benders basée sur la logique).

Décomposition de Benders classique

Question 3. Supposons que le choix des sites où placer les machines est connu ainsi que l'affectation des tâches à ces sites et l'ordre dans lesquels les tâches seront planifiées. Il faut noter qu'une solution du problème initial n'est pas entièrement caractérisée car le coût de retard des tâches n'est pas calculable uniquement avec ces informations. On note $\overline{\mathcal{K}}$ les sites choisis pour l'emplacement des machines ($|\overline{\mathcal{K}}| = m$). Notons $\overline{\mathcal{J}}_k$ la **liste ordonnée** des tâches affectées à la machine du site $k \in \overline{\mathcal{K}}$ (on pourra noter $j(l)$ la tâche occupant la position l dans la liste $\overline{\mathcal{J}}_k$)

a) Formaliser précisément le problème restant à résoudre pour obtenir une solution complète au problème initial (définir les données du problème, les décisions à prendre, les contraintes, la fonction objectif). Quelle est la caractéristique principale de ce problème que l'on peut exploiter pour accélérer sa résolution ?

b) Modéliser ce problème comme un programme linéaire.

c) Ecrire le problème maître correspond à l'application de la décomposition de Benders sur le problème initial (c.à.d., une formulation PLNE du problème uniquement avec des variables associées aux décisions de choix des sites et de séquencement des tâches sur les machines, ainsi qu'une ou des variables pour le coût du sous-problème.)

Décomposition de Benders combinatoire

Question 4. Supposons que le choix des sites où placer les machines est connu ainsi qu'uniquement l'affectation des tâches à ces sites. Une solution du problème initial n'est pas entièrement caractérisée car ces informations. On note $\widetilde{\mathcal{J}}_k$ l'**ensemble** des tâches affectées à la machine du site $k \in \mathcal{K}$ et $T\big(\widetilde{\mathcal{J}}_k\big)$ la somme du coût de retard des tâches de $\widetilde{\mathcal{J}}_k$.

a) En utilisant la notation de Graham introduite dans la partie ordonnancement, quel est précisément le problème restant à résoudre au niveau de la machine d'un site $k \in \mathcal{K}$ tel que $|\widetilde{\mathcal{J}}_k| \neq \varnothing$?

b) Expliquer précisément pourquoi $T\big(\widetilde{\mathcal{J}}_k\big) \geqslant \sum_{j\in\widetilde{\mathcal{J}}_k} p_j + \min_{j\in\widetilde{\mathcal{J}}_k}\{r_{jk}\} - \max_{j\in\widetilde{\mathcal{J}}_k}\{d_j - r_{jk}\}$.

c) Si $\widetilde{\mathcal{J}}'_k \supset \widetilde{\mathcal{J}}_k$, expliquer précisément pourquoi $T\big(\widetilde{\mathcal{J}}'_k\big)$ (la somme du coût de retard des tâches de $\widetilde{\mathcal{J}}'_k$) sera forcément supérieure ou égale à $T\big(\widetilde{\mathcal{J}}_k\big)$

d) Soit $z_{jk} = 1$ si la tâche j est affectée à la machine du site k, 0 sinon et soit π_k la somme du coût de retard des tâches affectées à la machine $k \in \mathcal{K}$, montrer que $\pi_k \geqslant T\big(\widetilde{\mathcal{J}}_k\big)\big(\sum_{j\in\mathcal{J}} z_{jk} - |\widetilde{\mathcal{J}}_k| + 1\big)$

e) En déduire une formulation PLNE du problème initial contenant uniquement les variables de choix des sites et d'affections des tâches aux sites ainsi que les variables $\{\pi_k\}_{k\in\mathcal{K}}$.

Indice : Un nombre exponentiel (mais fini) de contraintes est à introduire. Ce nombre provient du fait qu'on recourt à une énumération.

f) Ecrire un algorithme de résolution de cette formulation basée sur l'algorithme de résolution utilisée lorsqu'on a recourt à une décomposition de Benders classique.

Etude d'extensions du problème

Chaque extension ci-dessous est traitée de façon indépendante. Pour chaque question, décrire comment adapter l'un de vos 2 modèles PLNE.

Question 5. Chaque site $k \in \mathcal{K}$ peut maintenant accueillir au plus $1 \leqslant m_k \leqslant m$ machines.

Question 6. On désire maintenant prendre en compte le cas où chaque machine consomme de l'électricité lors de la planification des tâches et le prix de l'électricité varie en fonction des périodes de temps. Soit $\mathcal{P}$ l'ensemble des périodes de temps et c_{kt} le prix de l'électricité sur le site $k \in \mathcal{K}$ lors de la période $t \in \mathcal{P}$. On note e_j la consommation d'électricité demandée à chaque période de son exécution par la tâche $j \in \mathcal{J}$. Si la quantité maximale consommée d'électricité consommée durant une période de temps dépasse un seuil E, alors le surplus d'électricité demandée durant cette période a un coût de 50% plus important. Le coût de l'électricité consommée est pénalisée dans la fonction objectif par un coefficient λ_4.

Résolution heuristique du problème

Question 7. Décrire une heuristique pour construire des solutions "de bonne qualité" au problème (utile pour les instances de grande taille du dossier B_instances. Votre heuristique doit comporter une phase de construction d'une ou plusieurs solutions initiales et aussi une phase d'amélioration de ces solutions. Vous pouvez prendre inspiration des résultats aux questions précédentes et du fichier Heuristiques.pdf disponibles sur Moodle. Vous pouvez par exemple résoudre des sous-problèmes par un solveur de PLNE.
La description de votre heuristique devra être suffisamment précise afin de permettre son implémentation à la seule lecture de vos explications. Il faut donc bien expliciter les opérations à effectuer.

Expérimentations numériques

Mettre en place des expérimentations numériques pour comparer la résolution des programmes linéaires en nombres entiers par un solveur et votre heuristique. Un temps limite (minimum 5 minutes et de préférence > 10 minutes) doit être imposé.

Les instances à utiliser sont disponibles sur Moodle. Elles sont regoupées en deux dossiers : A_instances et B_instances selon leur taille. Pour les instances du dossier A_instances, on a $|\mathcal{J}| \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, $|\mathcal{K}| \in \{4, 6, 8, 10\}$, et $m \in \{2, 3, 4, 5\}$. Pour les instances du dossier B_instances, on a $|\mathcal{J}| \in \{150, 200, 250, 300\}$, $|\mathcal{K}| \in \{20, 40, 60, 80, 100\}$, et $m \in \{10, 20, 30, 40, 50\}$. Il n'est pas nécessaire de faire des expérimentations numériques sur toutes les instances, mais il faudra forcément tester votre heuristique sur quelques instances du dossier B_instances. Veiller à faire un choix cohérent.

Pour l'implémentation, le langage de programmation utilisé doit être C++, Java ou Python. Il est conseillé de bien organiser son code (plusieurs fichiers, plusieurs fonctions, ...) et de le rendre lisible en l'agrémentant de quelques commentaires. Votre code devra contenir forcément les classes (ou structures) suivantes :

- `PMSLPData` : classe contenant les données du problème
- `PMSLPSolution` : classe contenant les données d'une solution du problème
- `PMSLPMIPModel1` et `PMSLPMIPModel2` : classes contenant la résolution du premier et du deuxième modèle de programmation linéaire en nombres entiers

  Pour chaque classe, la fonction permettant de lancer la résolution du modèle s'appellera `solve`. Elle prendra en paramètre un objet de la classe `PMSLPData` et elle retournera un objet de la classe `PMSLPSolution`.

  La fonction permettant de lancer la résolution du modèle s'appellera `solve`. Elle prendra en paramètre un objet de la classe `PMSLPData` et elle retournera un objet de la classe `PMSLPSolution`.
- `PMSLPHeuristic` : classe contenant l'heuristique

  La fonction permettant de lancer l'heuristique s'appellera `solve`. Elle prendra en paramètre un objet de la classe `PMSLPData` et elle retournera un objet de la classe `PMSLPSolution`.

Vous êtes libres d'ajouter d'autres classes et structures. Il est fortement recommandé d'utiliser les fonctions déjà implémentées dans le langage utilisé.

La résolution de programmes linéaires en nombres entiers pourra se faire à l'aide du solveur Gurobi ou à l'aide du solveur libres (Cbc, HiGHs, SCIP).

Votre programme devra pouvoir être lancé de l'une des façons suivantes dépendamment du langage choisi :

- En C++ : `./pmslpSolver cheminVersInstance nomMethode tempsLimite`
- En Java : `java -jar pmslpSolver.jar cheminVersInstance nomMethode tempsLimite`
- En Python : `python pmslpSolver.py cheminVersInstance nomMethode tempsLimite`

où `nomMethode` $\in \{\text{MIP1}, \text{MIP2}, \text{H}\}$ et `tempsLimite` est un entier indiquant le temps maximum autorisé en secondes.
L'exécution de votre programme devra (en cas de succès à trouver une solution) écrire un fichier résultat ayant le format suivant :

- une ligne pour chaque tâche avec le numéro du site sur lequel elle est exécutée et la période de début d'exécution

Dans l'exemple ci-dessous, le coût de la solution a été calculé comme étant égal à 147. Les tâches 1, 2 et 3 sont réalisées respectivement sur les sites 2, 1 et 2 et débutent aux périodes 1, 7 et 3.

```
2 1
1 7
2 3
147
```

Rendu

Ecrire un rapport clair mais concis comprenant :

— les réponses aux questions

— l'étude numérique comparative des performances des méthodes de résolution (tout ce qui suit doit être sous forme de texte : introduction des expérimentations menées, présentation des tableaux de résultats (explication des colonnes ...), analyse des résultats

Il est possible de faire mention des difficultés rencontrées et un bilan de ce que vous avez personnellement retenu de ce projet dans la conclusion.

Au final, vous devez déposer sur Moodle une archive (extensions acceptées : .zip, .tar, .tar.xz, .tar.bz2, .tar.gz, .rar, .7z) contenant :

— votre rapport

— votre code (avec tous les fichiers nécessaires à la compilation si vous utilisez C++)

— l'exécutable associé à votre code si l'implémentation a été effectuée en C++ ou Java

Barème

Le barème suivant est fourni à titre indicatif. Il est susceptible d'évoluer légèrement lors de la correction du projet.

Réponses aux questions (/29pts) :

— Question 1 (/1pt)

— Question 2 (/5pts)

— Question 3 (/6pts) a) 1pt b) 2pts c) 3pts

— Question 4 (/8pts) a) 1pt b) 1pt c) 1pt d) 1 pt e) 2pts f) 2pts

— Question 5 (/2pts)

— Question 6 (/3pts)

— Question 7 (/4pts)

Pour obtenir le maximum de points à chaque question, vos réponses doivent être claires, précises et justifiées.

Expérimentations numériques (/11pts) :

— Implémentation correcte des deux programmes linéaires en nombres entiers (/4pts)

— Implémentation correcte de l'heuristique et résultats "de bonne qualité" (/3pts)

— Présentation des expérimentations et de leur rôle (que cherchez-vous à évaluer et comment ?) et discussion des résultats obtenus + conclusion (/4pts)

Le non respect des consignes d'implémentation données dans la partie intitulé "Expérimentations numériques" sera pénalisé.