

Étude de cas : Problème d'affectation généralisé (GAP)

Travail écrit à rendre pour le lundi 4 novembre 2024 au plus tard
Les soutenances auront lieu le jeudi 7 novembre 2024 à partir de 9h

1 Sujet

On s'intéresse à un problème d'école en optimisation combinatoire : le **problème d'affectation généralisé (GAP)**.

Étant donnés :

- T tâches et M agents
- b_i , la quantité de ressource disponible de l'agent i ($i = 1, \dots, M$)
- c_{it} , le profit (ou gain) associé à l'affectation de la tâche t à l'agent i ($i = 1, \dots, M$ et $t = 1, \dots, T$)
- r_{it} , la quantité de ressource de l'agent i consommée par la tâche t ($i = 1, \dots, M$ et $t = 1, \dots, T$)

Le problème d'*affectation généralisé* consiste à maximiser le profit associé à l'affectation de T tâches à M agents tel que

1. chaque tâche est affectée à exactement un agent
2. la contrainte de capacité de chaque agent est respectée

Une formulation classique de ce problème est la suivante :

$$\begin{aligned} \max \quad & f(x) = \sum_{i=1}^M \sum_{t=1}^T c_{it} x_{it} \\ \text{s.c.} \quad & \sum_{i=1}^M x_{it} = 1 \quad t = 1, \dots, T \end{aligned} \tag{1}$$

$$\sum_{t=1}^T r_{it} x_{it} \leq b_i \quad i = 1, \dots, M \tag{2}$$

$$x_{it} \in \{0, 1\} \quad i = 1, \dots, M; t = 1, \dots, T \tag{3}$$

avec $x_{it} = 1$ si la tâche t est affectée à l'agent i ($x_{it} = 0$ sinon)

Les contraintes (1) garantissent que chaque tâche est affectée à un agent unique.

Les contraintes (2) traduisent les limitations de ressources des agents.

Hypothèses sur les données :

- c_{it} et b_i sont des entiers positifs
- $\{t : r_{it} \leq b_i\} \neq \emptyset \forall i \in \{1, \dots, M\}$

L'objectif de ce devoir est la résolution approchée du problème d'affectation généralisé par l'intermédiaire d'une métaheuristique.

2 Travail à réaliser

1. Déterminer une heuristique pour trouver une solution réalisable du problème d'affectation généralisé. Détailler les différentes étapes de l'algorithme et le tester sur les jeux de données proposés.
2. Proposer plusieurs structures de voisinage adaptées au problème d'affectation généralisé. Les illustrer sur un exemple.
3. Mettre en oeuvre une méthode de *montée* en utilisant une des structures de voisinage proposées. Tester la méthode de *montée* en la démarrant à partir de la solution trouvée par l'heuristique proposée en 1.
4. Mettre en oeuvre une métaheuristique et la tester sur les jeux de données proposés.

3 Description des jeux de données

Les algorithmes doivent au moins être testés sur un ensemble de 16 fichiers de données à récupérer sur le lien suivant :

<http://cedric.cnam.fr/~porumbed/meta/>

Le format de chaque fichier de données est le suivant:

- Nombre d'instances dans le fichier (P)
- Pour chaque instance p ($p=1,\dots,P$) sont listés:

nombre d'agents (M), *nombre de tâches* (T)

Pour chaque agent i ($i = 1, \dots, M$), *les gains (ou coûts) par tâche* : c_{it} ($t = 1, \dots, T$)

Pour chaque agent i ($i = 1, \dots, M$), *les quantités de ressource consommées par tâche* : r_{it} ($t = 1, \dots, T$)

Les capacités des agents : b_i ($i=1,\dots,M$)

Les 12 premiers fichiers à traiter sont les fichiers gap1.txt,...,gap12.txt. Ils comportent chacun 5 instances. Dans ce premier ensemble de fichiers, le GAP est considéré en MAXIMISATION.

Les valeurs optimales des 5 instances de chaque fichier sont données dans *table 1*. Dans le deuxième ensemble de 4 fichiers (gapa.txt, gapb.txt,gapc.txt,gapd.txt), le GAP est considéré en MINIMISATION.

Les valeurs optimales (ou meilleurs majorants connus) des 6 instances de chaque fichier sont données dans *table 2*.

D'autres fichiers de données seront transmis en complément la semaine du 21 octobre.

4 Tableau de résultats

Présenter un tableau récapitulatif des expériences numériques :

- en colonnes, doivent figurer, pour chaque approche heuristique, la valeur d'une meilleure solution trouvée, le temps de résolution et le saut à la valeur optimale (ou meilleure borne connue).
- en lignes, les instances traitées.

gap1	gap2	gap3	gap4
c515-1 336	c520-1 434	c525-1 580	c530-1 656
c515-2 327	c520-2 436	c525-2 564	c530-2 644
c515-3 339	c520-3 420	c525-3 573	c530-3 673
c515-4 341	c520-4 419	c525-4 570	c530-4 647
c515-5 326	c520-5 428	c525-5 564	c530-5 664
gap5	gap6	gap7	gap8
c824-1 563	c832-1 761	c840-1 942	c848-1 1133
c824-2 558	c832-2 759	c840-2 949	c848-2 1134
c824-3 564	c832-3 758	c840-3 968	c848-3 1141
c824-4 568	c832-4 752	c840-4 945	c848-4 1117
c824-5 559	c832-5 747	c840-5 951	c848-5 1127
gap9	gap10	gap11	gap12
c1030-1 709	c1040-1 958	c1050-1 1139	c1060-1 1451
c1030-2 717	c1040-2 963	c1050-2 1178	c1060-2 1449
c1030-3 712	c1040-3 960	c1050-3 1195	c1060-3 1433
c1030-4 723	c1040-4 947	c1050-4 1171	c1060-4 1447
c1030-5 706	c1040-5 947	c1050-5 1171	c1060-5 1446

Table 1: Valeurs optimales des instances gap1.txt à gap12.txt

gapa	gapb	gapc	gapd
a5-100 1698	b5-100 1843	c5-100 1931	d5-100 6353
a10-100 1360	b10-100 1407	c10-100 1402	d10-100 6347
a20-100 1158	b20-100 1166	c20-100 1243	d20-100 6185
a5-200 3235	b5-200 3552	c5-200 3456	d5-100 12742
a10-200 2623	b10-200 2827	c10-200 2806	d10-200 12430
a20-200 2339	b20-200 2339	c20-200 2391	d20-200 12241*

Table 2: Valeurs optimales ou majorants (suivis de *) pour les instances gapa.txt, gapb.txt, gapc.txt, gapd.txt à traiter

5 Travail à rendre

- Un rapport, dans lequel vous devrez :
 - décrire les heuristiques et métaheuristiques proposées ainsi que les algorithmes associés
 - détailler les voisinages et paramétrages choisis,
 - analyser les résultats numériques que vous aurez obtenus sur la batterie d'instances proposée.
- Une archive du code comprenant un readme.txt

Le rapport sous format .pdf et l'archive associée au travail sont à envoyer aux adresses suivantes agnes.plateaualfandari@lecnam.net et daniel.porumbel@cnam.fr **au plus tard le lundi 04/11/2024**

6 RECOMMANDATIONS

- Ce travail est à réaliser en binôme.

- Le code source doit être bien commenté.
- Le codage des algorithmes peut être effectué dans le langage de votre choix.

7 Soutenances

La présentation orale aura lieu le jeudi 07/11/2024 à partir de 9h.