|  |
| --- |
| **MESSOU FRANCK JUNIOR ABOYA**  **20-ESATIC0067BF**  **MASTER 1 SIGL 2020-2021** |

**DEVOIR DE RECHERCHE OPÉRATIONNELLE**

**Compagnie de transport naval à Abidjan : cas de Citrans**

**DEFINITION DU PROBLEME MULTIOBJECTIF**

Afin de satisfaire sa clientèle, une compagnie navale dispose de 6 bateau-bus avec nombre de place différent pour chaque flotte et souhaite les mettre en service sur des lignes de transport. Ici, ce qui intéresse la compagnie, c’est de faire le maximum de profit journalier tout en minimisant avec le COVID19 et ses conséquences économiques, le personnel nécessaire pour chaque voyage ainsi que la durée du trajet par jour pour une durée de vie maximale des engins. Les places réservées aux clients sont comme suit :

* Le premier bateau-bus dispose de **B1** places
* Le deuxième bateau-bus dispose de **B2** places
* Le troisième bateau-bus dispose de **B3** places
* Le quatrième bateau-bus dispose de **B4** places
* Le cinquième bateau-bus dispose de **B5** places
* Le sixième bateau-bus dispose de **B6** places

Les coûts de tickets de voyages, du bénéfice minimum par voyage, du temps passé par voyage, du temps de service par jour ainsi que du nombre de personnel nécessaire par voyage sont respectivement comme suit :

* Ligne A : **tA** $ ; rapporte au minimum **rA** $/jour ; **m1** minute/voyage ; **m1,max** minute/jour  ; **n1** personnel à bord
* Ligne B : **tB** $ ; rapporte au minimum **rB** $/jour ; **m2** minute/voyage ; **m2,max** minute/jour  ; **n2** personnel à bord
* Ligne C : **tC** $ ; rapporte au minimum **rC** $/jour ; **m3** minute/voyage ; **m3,max** minute/jour  ; **n3** personnel à bord

Par la suite, nous modélisons le problème multi-objectif ainsi défini suivant une approche mathématique plus solvable.

𝑀𝑎𝑥 𝑃𝑟𝑜𝑓𝑖𝑡 = (v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6) + (v1b **tB**B1 + v2b **tB**B2 + v3b **tB** B3 + v4b **tA**B4+ v5b **tB**B5+ v6b **tB**B6)

+ (v1c **tC**B1 + v2c **tC**B2 + v3c **tC**B3 + v4c **tC**B4+ v5c **tC**B5+ v6c **tC**B6)

𝑀𝑖𝑛 durée = (v1a + v1b+ v1c) **m1** + (v2a + v2b+ v2c) **m2** + (v3a + v3b+ v3c) **m3**

𝑀𝑖𝑛 personnel = (v1a + v1b+ v1c) **n1** + (v2a + v2b+ v2c) **n2** + (v3a + v3b+ v3c) **n3**

Sous contraintes de :

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rA (1)**

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rb (2)**

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rc (3)**

v1a + v1b+ v1c **≤ m1,max**  **(4)**

v2a + v2b+ v2c **≤ m2,max**  **(5)**

v3a + v3b+ v3c **≤ m3,max**  **(6)**

Avec **vi∆** est le nombre de voyages effectué par le bateau-bus i sur la ligne j.

Méthode de résolution du problème

Il existe plusieurs méthodes de résolution. Mais pour ce type de problème, nous optons pour une résolution simple. D’où

**Principe de l’algorithme de recherche Tabou**

L'idée de la recherche tabou consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui minimise la [fonction objectif](https://fr.wikipedia.org/wiki/Fonction_objectif).

Il est essentiel de noter que cette opération peut conduire à augmenter la valeur de la fonction (dans un problème de minimisation) : c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée. C'est à partir de ce mécanisme que l'on sort d'un [minimum local](https://fr.wikipedia.org/wiki/Minimum_local).

Le risque cependant est qu'à l'étape suivante, on retombe dans le minimum local auquel on vient d'échapper. C'est pourquoi il faut que l'heuristique ait de la mémoire : le mécanisme consiste à interdire (d'où le nom de *tabou*) de revenir sur les dernières positions explorées.

Les positions déjà explorées sont conservées dans une file [FIFO](https://fr.wikipedia.org/wiki/First_in,_first_out) (appelée souvent *liste tabou*) d'une taille donnée, qui est un paramètre ajustable de l'heuristique. Cette file doit conserver des positions complètes, ce qui dans certains types de problèmes, peut nécessiter l'archivage d'une grande quantité d'informations. Cette difficulté peut être contournée en ne gardant en mémoire que les mouvements précédents, associés à la valeur de la fonction à minimiser.

**Résolution du problème avec l’algorithme**

Soit **S** l’espace de solutions qui vérifie toutes les contraintes et définie par

**s =** (**v**1a,**v** 1b,**v** 1c, **v** 2a,**v** 2b,**v** 2c, **v** 3a,**v** 3b,**v** 3c, v4a,**v** 4b,v4c, v5a,**v** 5b,**v** 5c, **v** 6a,v6b,v6c)

la fonction de voisinage est définie comme suit :

**f1(s)** =( **v**1a, **v**1b, **v**1c, **v**2a, **v**2b, **v**2c, **v**3a, **v**3b, **v**3c, v4a, **v**4b, v4c, v5a, **v**5b, **v**5c, **v**6a, v6b, v6c )

**f2(s)** =( **v**2a, **v**2b, **v**2c, **v**1a, **v**1b, **v**1c, **v**3a, **v**3b, **v**3c, v4a, **v**4b, v4c, v5a, **v**5b, **v**5c, **v**6a, v6b, v6c )

**f3(s)** =( **v**2a, **v**2b, **v**2c, **v**3a, **v**3b, **v**3c, **v**1a, **v**1b, **v**1c, v4a, **v**4b, v4c, v5a, **v**5b, **v**5c, **v**6a, v6b, v6c )

**F(s):**

**f4(s)** =( **v**2a, **v**2b, **v**2c, **v**3a, **v**3b, **v**3c,v4a, **v**4b, v4c, **v**1a, **v**1b, **v**1c, v5a, **v**5b, **v**5c, **v**6a, v6b, v6c )

**f5(s)** =( **v**2a, **v**2b, **v**2c, **v**3a, **v**3b, **v**3c,v4a, **v**4b, v4c, v5a, **v**5b, **v**5c, **v**1a, **v**1b, **v**1c, **v**6a, v6b, v6c )

**f6(s)** =( **v**2a, **v**2b, **v**2c, **v**3a, **v**3b, **v**3c,v4a, **v**4b, v4c, v5a, **v**5b, **v**5c, **v** 6a, v6b, v6c , **v**1a, **v**1b, **v**1c)

la fonction de coût est définie comme suit :

𝑀𝑎𝑥 𝑃𝑟𝑜𝑓𝑖𝑡 = (v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6) + (v1b **tB**B1 + v2b **tB**B2 + v3b **tB** B3 + v4b **tA**B4+ v5b **tB**B5+ v6b **tB**B6)

+ (v1c **tC**B1 + v2c **tC**B2 + v3c **tC**B3 + v4c **tC**B4+ v5c **tC**B5+ v6c **tC**B6)

𝑀𝑖𝑛 agrégation = (v1a + v1b+ v1c) **m1** + (v2a + v2b+ v2c) **m2** + (v3a + v3b+ v3c) **m3 +**

(v1a + v1b+ v1c) **n1** + (v2a + v2b+ v2c) **n2** + (v3a + v3b+ v3c) **n3**

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rA (1)**

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rb (2)**

v1a **tA**B1 + v2a **tA**B2 + v3a **tA**B3 + v4a **tA**B4+ v5a **tA**B5+ v6a **tA**B6 ≥ **rc (3)**

v1a + v1b+ v1c **≤ m1,max**  **(4)**

v2a + v2b+ v2c **≤ m2,max**  **(5)**

v3a + v3b+ v3c **≤ m3,max**  **(6)**

coût(s) = profit(s) – agrégation(s)

**ALGORITHME**

L = Ø; //liste des points tabous

s= (1, 2, 3, 1, 2, 3, 1, 2, 3) // initialisation de la solution **s**

Xk = 50 //nombre d’itération maximale

x = 0 et l = 0 // initialisation de du compteur et de la liste tabou

Tant que x ≤ Xk // compteur inférieur à 50

y = f(s)

si y n’est pas de L alors

C = coût(y) - coût(s)

Si C ≥ 0 et contraintes satisfaites alors

L = L + s //ajout solution à la liste L

s = y

x = x + 1

retourner s. //solution

Instance du problème

Avec la libéralisation des bateaux-bus à Abidjan en Côte-d’Ivoire, la concurrence s’annonce vive entre la Société de transport lagunaire (STL) et Citrans, qui a lancé ses activités.

Citrans prévoyait de transporter aux alentours de 40 000 passagers par jour dès le début et d’exploiter six gares. Citrans souhaite donc maximiser ses profits journalier tout en passant le moins de temps possible sur un trajet avec un nombre de personnel optimal.

Le problème qui en résulte est de maximiser les profit tout en minimisant la durée du transport (attentes, embarquement, débarquements,...).

Dans la suite de notre analyse, nous supposons des prix fictifs.

Les places réservées aux clients sont comme suit :

* Le premier bateau-bus dispose de **50** places
* Le deuxième bateau-bus dispose de **75** places
* Le troisième bateau-bus dispose de **100** places
* Le quatrième bateau-bus dispose de **70** places
* Le cinquième bateau-bus dispose de **85** places
* Le sixième bateau-bus dispose de **90** places

Les coûts de tickets de voyages, du bénéfice minimum par voyage, du temps passé par voyage, du temps de service par jour ainsi que du nombre de personnel nécessaire par voyage sont respectivement comme suit :

* Ligne Plateau-Cocody : **500** F ; rapporte au minimum **300 000** F/jour ; **20** minute/voyage ; **250** minute/jour  ; **4** personnel à bord
* Ligne Plateau-Traicheville : **300** F ; rapporte au minimum **250 000** F/jour ; **25** minute/voyage ; **300** minute/jour  ; **4** personnel à bord
* Ligne Plateau-Yopougon : **200** $ ; rapporte au minimum **180 000** F/jour ; **25** minute/voyage ; **325** minute/jour  ; **5** personnel à bord

**RÉSOLUTION**

**Voir ci-joints le code source écrit en python**

**Dossier parent : mesabo.py**

Sur github : [mesabo.py](https://github.com/mesabo/recherche_operationnelle_algo_tabou_citrans/blob/main/mesabo.py)

À tester ici [programiz.com](https://www.programiz.com/python-programming/online-compiler/)

Aperçu de la résolution ci-dessous :

