Projet A3.
Introduction à la recherche opérationnelle.

Antoine Desrier, Louis Lecru,
Dominique Paul, Nathaniel Raimbault,
Matthieu Saumard, Nesma Settouti.

31-05-2024

Authentique : *L. Lecru* : - A l'ISEN, nous envisageons de donner un projet à propos d'un problème de chargement de conteneurs aux classes de troisième année, visant à minimiser le nombre de conteneurs.

- Un ami : C'est génial. Je vais essayer de le mettre en œuvre pour limiter le nombre de semi-remorques transportant nos machines en pièces détachées de notre site de production vers l'usine où les machines doivent être assemblées.

Quelques mois plus tard.

- *L'ami* (Soupirant) : Bon ... Finalement je n'avais pas les dimensions des pièces à ma disposition...

Presque sûrement entendu lors d'un déménagement : "Attendez, si on tourne ce carton et qu'on déplace cette caisse, je pourrai glisser cette boîte dans le coffre".

1 Modalité d'évaluation.

Ce projet est d'une durée de 30 heures. Il peut être réalisé par groupes de deux ou trois personnes au maximum. Il est demandé d'aborder la partie 1 et la partie 2. Le langage de programmation est Python. Le code expliqué devra être déposé sous Moodle au format Nom-Prénom1-Nom-Prénom-2. Une partie de l'évaluation constituera en la présentation de vos résultats au format PowerPoint pour une durée maximale de 15 minutes. Cette présentation devra contenir :

- Présentation de la philosophie des algorithmes approchés (heuristiques).
- Démonstration d'une ou plusieurs résolutions du problème de chargement du train de marchandises.
- Présentations des résultats de vos algorithmes pour les cas $d = 1, d = 2, d = 3$ online et offline (voir tableau partie 2).
- Etre capable de tester et expliquer un bout du code à la demande du jury.

On ajoute un brin de compétition pour ce projet :

- **Challenge Précision** : Le binôme parvenant à déterminer une organisation faisant intervenir un nombre minimum de wagons gagnera le challenge Précision.
- **Challenge temps** : Le binôme parvenant à déterminer une organisation convenable (peu éloignée du résultat du premier challenge) avec un algorithme s'exécutant en un temps record gagnera le challenge temps.

Introduction :

A propos de la recherche opérationnelle.

Plutôt qu'un simple arsenal de méthodes mathématico-informatiques destiné à l'optimisation des processus de production et de diffusion des produits ou à celle d'organisations dans le secteur tertiaire, la recherche opérationnelle peut se définir comme *l'ensemble des méthodes et techniques rationnelles d'analyse et de synthèse des phénomènes d'organisation utilisables pour élaborer de meilleures décisions*. Chaque fois que l'Homme, des machines, des produits se trouvent en relations actives, on dira que l'on a affaire à un *phénomène d'organisation*. Les problèmes relatifs à ces problèmes d'organisation se signalent tout d'abord, généralement, par leur caractère hautement combinatoire. Pour peu que le hasard y soit mêlé et que la concurrence s'y manifeste, les situations deviennent encore plus compliquées. [...]

Aussi, la recherche opérationnelle n'est pas le moyen d'échapper aux jugements de bon sens. Bien au contraire, elle constitue la méthode adéquate pour ramener l'attention de l'Homme aux domaines où sa raison individuelle peut s'exercer efficacement, c'est à dire au niveau des choix - une fois résolus et explorés, par des techniques adéquates, les problèmes combinatoires, aléatoires ou concurrentiels qui dépassent l'esprit humain même le mieux constitué. [...]

Hâtons-nous de préciser d'abord, que, dans la vie, on ne rencontre que rarement des situations assez simples pour être justiciables directement d'un des algorithmes élémentaires. Bien souvent, un problème concret nécessite la mise en oeuvre d'un ensemble de méthodes, les unes s'inspirant des procédés classiques, les autres résultant de recherches originales. Parfois, pris par le temps, l'analyste se contente d'adapter une *méta-heuristique* à son problème et n'obtient qu'une solution approchée. Pour certaines

heuristiques (algorithmes approchés), il existe des garanties de performances.

Répétons aussi que l'on peut avoir avantage, dans la pratique, à substituer à la notion d'optimum (mathématique), utilisée dans les schémas plus simples, le concept de solution "très satisfaisante", ou même seulement "bonne". C'est notamment le cas lorsqu'un critère unique et précis n'a pu être défini.

Insistons enfin sur les relations entre chercheurs, chercheuses opérationnels et "utilisateurs". Toute équipe de Recherche opérationnelle doit, sous peine d'échec, travailler en parfait accord avec les utilisateurs, et préparer, en collaboration avec eux et leur organisateur, les améliorations et changements qu'elle préconise et dont il lui faut établir clairement l'efficacité. Tout ceci est valable, qu'il s'agisse d'une décision exceptionnelle prise au niveau le plus élevé, ou qu'il soit tout simplement question d'améliorer des procédures décisionnelles répétitives. Dans ce dernier cas, il y a intérêt à instituer une collaboration généralement acceptée, entre l'Homme et la machine. [1]

Objectifs du projet :

Ce projet de mathématique a pour objectif de présenter un problème d'optimisation pouvant être rencontré dans la logistique (stockage entrepôts, palettes), dans l'industrie pour la découpe de câbles, en informatique pour le rangement de fichiers voire même un déménagement. Sa réalisation permettra de découvrir (ou revoir) la notion d'algorithme approché (heuristique), la notion de complexité d'un algorithme et le compromis entre précision d'une solution et rapidité de l'exécution d'un algorithme. Enfin, un des intérêts final est de vous permettre d'être source de propositions en entreprise face à des problèmes d'organisations. La bonne réalisation de ce projet devrait, nous l'espérons, vous mener à vous poser dans le futur les questions suivantes si vous rencontrez un problème d'organisation :

- Existe-t-il une meilleure solution ?
- Comment peut-on avoir accès à cette meilleure solution ?
- Peut-on avoir accès à cette solution rapidement ?
- Est-il pratique de la mettre en œuvre sur le terrain ?
- Dans le cas où un algorithme approché est nécessaire, quel est l'écart entre une solution approchée et la solution optimale ?

Pour aller plus loin :

De nombreux exemples industriels sont évoqués dans le livre pdf *Livre blanc de la recherche opérationnelle en France*, disponible sous Moodle. Sa lecture est vivement recommandée pour :

- avoir une vue d'ensemble sur les nombreuses entreprises ayant recours à la recherche opérationnelle.
- avoir une culture générale de différents problèmes d'optimisation pouvant être rencontrés en entreprise.
- avoir une vision d'ensemble sur différentes entreprises proposant leurs services pour répondre à des problèmes d'optimisation. La réalisation de ce projet pourra également encourager la réalisation d'un algorithme ou logiciel "maison" au besoin en entreprise.

Au besoin, voici une courte bibliographie d'ouvrages abordant certains de ces thèmes :

Références

[1] Robert Faure, Bernard Lemaire, Christophe Picouleau, *Précis de recherche opérationnelle. Méthodes et exercices d'applications* 7e édition, Dunod, 2014.

[2] Michel Gondran, Michel Minoux *Graphes et algorithmes*. 4e édition, Collection Edf R et D, Lavoisier, 2009.

Partie 1.

Avant de se lancer sur un projet de taille plus industrielle, intéressons-nous au problème suivant :

Nous avons en face de nous une série d'objets ayant tous un poids propre, ainsi qu'une "utilité" mesurée par un réel. Nous souhaitons maintenant choisir certains de ces objets en nous assurant que la charge totale ne dépasse pas une constante C, la meilleure façon de le faire étant pour nous d'optimiser la somme des utilités des objets contenus dans le sac.

Pour cela, nous associerons à chaque objet o d'une liste L une variable binaire $prendre[o]$, valant 1 si l'objet doit être mis dans le sac, et 0 sinon. Nous cherchons donc à résoudre le $MILP$ (Mixed Integer Linear Programming) suivant :

$$max : \sum_{o \in L} utilite_o \times prendre_o$$

tel que :

$$\sum_{o \in L} poids_o \times prendre_o \leq C$$

Une cycliste souhaite étudier la composition de son sac à dos pour un parcours à travers les collines des Monts d'Arrées. La masse des objets pouvant être transportée dans le sac à dos est une constante C.

	A	B	C
1	**Objet**	**Masse**	**Utilité**
2	Pompe	0,2	1,5
3	Démonte-pneus	0,1	1,5
4	Gourde	1	2
5	Chambre à air	0,2	0,5
6	Clé de 15	0,3	1
7	Multi-tool	0,2	1,7
8	Pince multiprise	0,4	0,8
9	Couteau suisse	0,2	1,5
10	Compresses	0,1	0,4
11	Désinfectant	0,2	0,6
12	Veste de pluie	0,4	1
13	Pantalon de pluie	0,4	0,75
14	Crème solaire	0,4	1,75
15	Carte IGN	0,1	0,2
16	Batterie Portable	0,5	0,4
17	Téléphone mobile	0,4	2
18	Lampes	0,3	1,8
19	Arrache Manivelle	0,4	0
20	Bouchon valve chromé bleu	0,01	0,1
21	Maillon rapide	0,05	1,4
22	Barre de céréales	0,4	0,8
23	Fruits	0,6	1,3
24	Rustines	0,05	1,5

1. On fixe N le nombre d'objets dans le sac à dos. Sans tenir compte de la contrainte de charge maximale, combien de sac à dos contenant N objets peut-on réaliser ? Calculer ce nombre de sacs à dos pour $N \in \{1, 2, 10, 23\}$.

2. Combien d'organisations de sac à dos peut-on former au total, sans tenir compte de la charge maximale de sac à dos ? On inclura les sacs à dos avec les 23 objets, et le sac à dos vide.

3. Résoudre le problème à la main pour $C = 0.6$.

4. Ecrire un algorithme papier, le plus simple qui vous vienne à l'esprit, pour lequel :

   - on peut faire varier le nombre d'opérations effectuées par l'algorithme ;
   - dénombrer très facilement le nombre d'opérations (comparaisons, additions, multiplications, ...) que l'algorithme effectue pour se terminer.
   - l'algorithme se termine.

5. Écrire ce programme sous Python. A l'aide de la librairie *time* déterminer le temps de calcul de votre CPU pour réaliser n opérations. On fera varier n afin d'obtenir une idée du temps de réponse de l'algorithme en fonction du nombres d'opérations à effectuer. On notera T la durée (moyenne estimée) en secondes de la réalisation d'une opération.

6. A l'aide des approximations ci-dessus, et en supposant que chaque opération possède le même coût en temps, combien de temps faudrait-il pour tester toutes les organisations de sac à dos (dans le cas où il n'y a pas de contrainte sur la masse) ?

7. Rédiger un algorithme papier pour la résolution exacte du problème du sac à dos. On l'appellera A.

8. Rédiger cet algorithme exact A sous Python.

9. A l'aide de A, déterminer la (ou les) composition(s) du sac à dos dans le cas où $C = 2$, $C = 3$, $C = 4$ et $C = 5$. On précisera le temps de calcul pour chaque valeur de C. Gardez ces valeurs pour les challenges temps et précision.

10. Rédiger à présent un algorithme papier heuristique pour la résolution du problème du sac à dos. On l'appellera B. La base de données de ce sac à dos est disponible sous Moodle.

11. Rédiger cet algorithme approché B sous Python.

12. A l'aide de B, déterminer la (ou les) composition(s) du sac à dos dans le cas où $C = 2$, $C = 3$, $C = 4$ et $C = 5$. On précisera le temps de calcul pour chaque valeur de C. Gardez ces valeurs pour les challenges temps et précision.

13. Comparer A et B.

14. Ecrire un nouvel algorithme B' approché renvoyant une solution approchée à ce problème en une durée inférieure à 2 secondes. On comparera cette solution approchée à la solution exacte obtenue précédemment. Gardez ces valeurs pour les challenges temps et précision.

2 Partie 2.

Un train de fret de la SNCF doit transporter des conteneurs au sein desquels sont chargés différentes marchandises depuis un site industriel d'une entreprise vers un second site industriel. Tous les conteneurs sont de la même dimension : longueur de 11,583 mètres, largeur de 2,294 mètres et hauteur de 2,569. On peut charger un unique conteneur par wagon. L'objectif de ce problème est de déterminer une affectation des marchandises dans les conteneurs de façon à minimiser le nombre total de wagons. Toutes les marchandises doivent être stockées à bord, on supposera donc que le nombre de conteneurs peut être infini. On suppose qu'à tout instant chaque conteneur peut être chargé. Les caractéristiques des marchandises sont disponibles dans la base de données sous Moodle intitulée **donneesmarchandises**. On notera d la dimension suivant laquelle on traite le problème :

- Pour **d=1** on ne tient compte que de la longueur des marchandises. Les marchandises sont donc assimilables à des segments de droite. Il n'est pas possible de les mettre des marchandises côte à côte, ni de les superposer.
- Pour **d=2** on suppose qu'on tient compte de la surface occupée au sol par les marchandises. On peut mettre les marchandises côte à côte éventuellement, mais on ne peut pas les superposer en hauteur. Les marchandises sont assimilables à des rectangles. On tient donc compte des variables : longueur et largeur.
- Pour **d=3** on tient compte du volume occupé par chaque marchandise. On pourra éventuellement empiler les marchandises. Les marchandises sont assimilables à des parallélépipèdes rectangles.

On distinguera également deux cas différents, suivants si les marchandises peuvent être triées ou non.

- **Off line** On suppose que l'on peut trier les marchandises avant des les charger dans les conteneurs.
- **On line** On suppose que les marchandises ne peuvent pas être triées. On affectera donc les marchandises par ordre d'apparition dans la liste. La première marchandise (ligne 2) étant affectée en première, et la dernière (ligne 101) en dernier.

On a donc les cas suivants à traiter :

Online	Nombre wagons	dimension non occupée	Temps calcul
d=1			
d=2			
d=3			

Offline	Nombre wagons	dimension non occupée	Temps calcul
d=1			
d=2			
d=3			

Consigne : Traiter dans l'ordre $d = 1$ Offline et Online, dimension $d = 2$ Offline et Online, dimension $d = 3$ Offline et Online. Pour chacun de ces 6 cas vous :

1. Mettrez au point un algorithme qui renvoie au moins une solution. Cet algorithme pourra être un algorithme exact ou heuristique en fonction de la complexité du problème. Stocker les résultats pour les challenges.
2. Evaluerez la complexité de votre algorithme. Un usage de la librairie *time* pour représenter le temps de calcul en fonction du nombre de marchandises serait un bon début pour représenter l'évolution du temps de calcul en fonction de la taille des données en entrée.
3. **(Difficile)** Pour aller plus loin, vous pourriez essayer d'obtenir des informations sur sa complexité (complexité au pire, borne supérieure ...)