

Algorithmique et Recherche Opérationnelle : rapport de projet

Berthion Antoine - 000566199

van Praag Lucas - 000569535

Troisième Bachelier en Sciences Informatiques - Année 2024-2025

1 Introduction

Ce rapport, réalisé dans le cadre du cours d'Algorithmique et recherche opérationnelle de l'année académique 2024-2025, a pour objectif de modéliser, résoudre et analyser différentes variantes min-KP sur des sacs à dos uniques et multiples.

2 Formulation mathématique

2.1 Modèle 0/1

Cette section contient les réponses à la **section 2** du PDF de consignes pour la variante 0/1 du problème du sac à dos.

2.1.1 Modèle entier

I. Formulation du modèle

$$\min \sum_{i \in N} \sum_{j \in M} c_i x_{ij} \quad (\text{coût total minimum}) \quad (\mathcal{P})$$

$$\text{s.t.} \quad \sum_{i \in I} s_i x_{ij} \geq D_j \quad \forall j \in M \quad (\text{couverture de la demande}) \quad (1)$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in N \quad (\text{objet utilisé} \leq 1 \text{ fois}) \quad (2)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in N, \forall j \in M \quad (\text{binaires}) \quad (3)$$

II. Définition précise des éléments du modèle

Indices $N = \{1, \dots, n\}$ objets, $M = \{1, \dots, m\}$ sacs.

Constantes s_i : taille de l'objet i ,
 c_i : coût de l'objet i ,
 D_j : taille minimale demandée par le sac j .

Variables de décision

$$x_{ij} = \begin{cases} 1 & \text{si l'objet } i \text{ est placé dans le sac } j, \\ 0 & \text{sinon.} \end{cases}$$

Fonction objectif (\mathcal{P}) : minimiser la somme coût \times sélection, en incluant tous les sacs.

Contraintes (1) assure la couverture de chaque demande D_j ; (2) assure l'assignation d'un objet à au plus un sac; (3) impose la logique binaire.

III. Spécification des hypothèses et conditions de faisabilité

- a) **Données strictement positives** : $s_i > 0$, $c_i > 0$, $D_j > 0$. Ces suppositions permettent d'éviter des cas triviaux comme la sélection d'un objet gratuit, ou bien le remplissage d'un sac vide.
- b) **Capacité totale suffisante** : une condition nécessaire de faisabilité est $\sum_{i \in N} s_i \geq \sum_{j \in M} D_j$. Autrement, tous les sacs ne peuvent être remplis et le problème est infaisable.
- c) **Binarité** : la variable binaire $x_{ij} \in \{0, 1\}$, $\forall i \in N, \forall j \in M$, modélise la présence ou non de l'article i dans le sac à dos j . Dès lors, $x_{ij} = 0$ implique que l'article i n'est pas inclus dans le sac à dos j , et inversement pour $x_{ij} = 1$.

2.1.2 Maximum Knapsack - Contexte d'utilisation

Afin d'illustrer le problème **max-KP** à sacs à dos unidimensionnels multiples, on peut se placer dans un **contexte logistique** :

*"Une entreprise possède plusieurs camions, et doit décider des commandes à livrer sur un jour donné. Chaque commande rapporte un bénéfice, et pèse un certain poids (en kg). Une commande doit tenir entièrement dans un seul camion, et ne peut donc pas être partagée entre plusieurs. Chaque camion a une capacité maximale de poids (en kg) à ne pas dépasser. L'objectif final de la planification est de maximiser la **marge totale de bénéfices** de l'entreprise sans dépasser la charge maximale de chaque camion."*

Explication des paramètres :

- $N = \{1, \dots, n\}$: chaque indice $i \in \{1, \dots, n\}$ correspond à une commande.
- $M = \{1, \dots, m\}$: chaque indice $j \in \{1, \dots, m\}$ correspond à un camion.
- $\hat{N} \subseteq N$: est le sous-ensemble des commandes effectivement sélectionnées par le modèle, et est inclus dans N . Dès lors qu'une commande i est placée dans un camion j , i appartient à \hat{N} . On lie donc i à N par la variable de décision x_{ij} , qui vaut 1 lorsque la commande i est placée dans le camion j , 0 sinon.
- $\mathbf{C} = (\mathbf{C}_1, \dots, \mathbf{C}_m)^\top$: pour chaque camion $j \in M$, la quantité C_j indique la charge maximale du camion.
- $\mathbf{p} = (\mathbf{p}_1, \dots, \mathbf{p}_n)^\top$: chaque valeur $p_i, i \in N$ indique le bénéfice que rapporte la commande i .
- $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_n)^\top$: chaque valeur $w_i, i \in N$ indique le poids, en kg, de la commande i .

2.1.3 Minimum Knapsack - Contexte d'utilisation

Afin d'illustrer le problème **min-KP** à sacs à dos unidimensionnels multiples, on se place désormais dans un **contexte de connectivité** dans un graphe :

*"Dans un graphe donné, les sommets représentent des unités géographiques (comtés, etc). Afin de faire un tout nouveau découpage électoral, on souhaite regrouper des unités géographiques (normalement contiguës, et donc liées par une arête, ce qu'on abstrait ici) afin de former des districts. Le nombre de districts souhaités est fixé à l'avance. Chaque district a une valeur minimale de population à contenir. Chaque unité géographique étant associée à un coût relatif à la longueur de sa frontière ainsi qu'à une taille liée à sa quantité de population, on cherche à **minimiser la longueur de frontière** totale du nouveau district tout en respectant la contrainte de population minimale dans chacun."*

Explication des paramètres :

- $N = \{1, \dots, n\}$: chaque indice $i \in \{1, \dots, n\}$ correspond à une unité géographique.
- $M = \{1, \dots, m\}$: chaque indice $j \in \{1, \dots, m\}$ correspond à un district.
- $\hat{N} \subseteq N$: est le sous-ensemble des unités géographiques effectivement sélectionnées par le modèle, et est inclus dans N . Dès lors qu'une unité géographique i est placée dans un district j , i appartient à \hat{N} . On lie donc i à N par la variable de décision x_{ij} , qui vaut 1 lorsque l'unité géographique i est placée dans le district j , 0 sinon.
- $\mathbf{D} = (\mathbf{D}_1, \dots, \mathbf{D}_m)^\top$: pour chaque district $j \in M$, la quantité D_j indique la population minimale requise dans celui-ci.
- $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_n)^\top$: chaque valeur $c_i, i \in N$ indique le coût lié à la longueur de la frontière de l'unité géographique i .
- $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n)^\top$: chaque valeur $s_i, i \in N$ indique la taille de la population de l'unité géographique i .

*Note : cet exemple s'inspire directement du **political-districting problem**, étudié entre autres par Ricca, Scozzari, and Simeone (2013).*

2.2 Relaxation LP

Cette section se penche sur la relaxation linéaire des modèles **minKP** à sac à dos unidimensionnel unique et multiples.

Désormais, x est continue. La contrainte d'intégralité (variables de décision) $x \in \{0, 1\}$ devient $0 \leq x \leq 1$, et représente la fraction de chaque objet dans le/les sac(s) à dos.

Nous passons donc d'un modèle entier à un modèle linéaire.

Variante mono-sac

Le programme linéaire primal pour le modèle à sac à dos unique est le suivant :

$$\min \sum_{i \in N} c_i x_i \quad \text{s.t.} \quad \sum_{i \in N} s_i x_i \geq D, \quad 0 \leq x_i \leq 1.$$

En ce qui concerne la qualité de la relaxation, nous observons l'écart d'intégralité (*ou bien integrality gap*). Ce dernier représente la différence entre les solutions optimales du programme entier (discret) et fractionnel (continu).

Dans la solution optimale LP, chaque variable x_i vaut entre 0 et 1. On a deux scénarios possibles lorsqu'on veut passer à une solution entière :

- Les objets, non-fractionnés, suffisent pour atteindre la demande. Le coût entier est alors égal au coût LP.
- Les objets entiers ne suffisent pas, et pour compléter le poids manquant, un seul objet est pris partiellement. On prend donc cet objet une fois, ainsi qu'une seconde fois mais sous forme de fraction. Cela implique qu'on ajoute au plus le coût de cet objet.

On peut montrer, par l'absurde qu'il n'y aura au plus qu'un seul objet fractionné : supposons que deux objets soient partiellement pris, alors il suffirait de diminuer la fraction du plus coûteux et augmenter l'autre, tout en ne violant pas la contrainte du modèle. Le coût serait alors plus petit, ce qui contredit l'optimalité de la solution.

En conclusion, comme il suffit de compléter d'au plus un objet, le surcoût payé par rapport à la solution entière est d'au plus deux fois le coût de l'objet. Le coût entier est au maximum doublé, donc la relaxation est **bornée par 2**.

Variante multi-sacs

$$\min \sum_{i \in N} \sum_{j \in M} c_i x_{ij} \quad \text{s.t.} \quad \sum_{i \in N} s_i x_{ij} \geq D_j \quad (\forall j \in M), \quad \sum_{j \in M} x_{ij} \leq 1 \quad (\forall i \in N), \quad 0 \leq x_{ij} \leq 1.$$

La deuxième contrainte signifie qu'un objet peut être fractionné et réparti entre plusieurs sacs, tant que la somme des fractions n'excède jamais 1. Alors, un même objet peut au maximum être réparti entre m sacs. Nous considérons :

- Un objet est fractionné sur k sacs et doit alors être acheté k fois.
- Etant donné que $k \leq m$, on paie au maximum m fois son coût initial (étant donné qu'on considère une solution entière).

Alors, tout comme pour la variante mono-sac, on applique le même raisonnement. Le coût total de la solution entière augmentera d'au plus m fois le coût LP. La relaxation est donc **bornée par m** .

Ce résultat implique qu'au plus m est grand, au plus la relaxation est faible. Pour beaucoup de sacs, le potentiel écart peut devenir important.

Note : ces références ont aidé dans les raisonnements présentés plus haut. Les accès ont été obtenus via un compte d'institution : Caprara, Kellerer, and Pferschy (2000) ainsi que Kellerer, Pferschy, and Pisinger (2004).

2.2.1 Programmes linéaires duaux - modèles relaxés

Variante mono-sac

Etant donné les contraintes du PL primal, on crée deux variables de décision :

- y , qui représente le coût d'une unité de "poids".
- $z_i, \forall i \in N$, qui représente un coût implicite associé à l'objet i .

La fonction objectif se calcule comme la somme de la quantité "membre de droite \times variable duale". Elle s'exprime de la manière suivante (la négation nous vient de la contrainte \leq) :

$$\max D \times y - \sum_{i \in N} z_i$$

La contrainte, elle, suit le même raisonnement mais appliqué aux colonnes de la matrice de contrainte du primal :

$$s_i \times y - z_i \leq c_i \quad (i \in N)$$

Variante multi-sacs

En suivant le même raisonnement, les variables de décision sont :

- $y_j \geq 0$
- $\lambda_j \geq 0$.

La fonction objectif du dual est :

$$\max \sum_{j \in M} D_j \times y_j - \sum_{i \in N} \lambda_i$$

La contrainte du dual est alors :

$$s_i \times y_j - \lambda_i \leq c_i \quad (\forall i \in N, j \in M).$$

3 Protocole expérimental et résultats

3.1 Analyse au moyen des instances fournies

- a) Vérifions l'égalité des valeurs optimales du primal et du dual pour la relaxation linéaire. Rappelons-nous du théorème de la dualité forte : "si le programme linéaire est faisable et borné, alors les optimums primal et dual sont égaux". Lorsque nous testons notre modèle, nous obtenons :

- pour le *primal relaxé* :

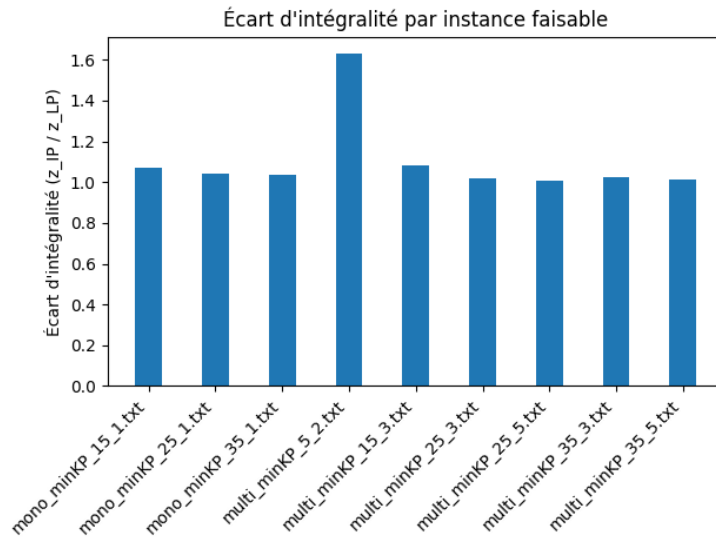
Total Cost: 4070.90476144

- pour le *dual relaxé* :

Optimal - objective value 4070.9048

Sur toutes les instances testées, la valeur renvoyée par le programme en mode 1 (*primal relaxé*), coïncide, à la tolérance près, avec celle du mode 2 (*dual relaxé*). Ce résultat confirme la dualité forte du programme linéaire.

- b) Discutons l'écart d'intégralité observé entre la solution entière optimale et celle de la relaxation linéaire. Sur plusieurs instances, nous observons des résultats avec un écart d'intégralité très faible. Le ratio $\frac{z_{IP}}{z_{LP}}$ est globalement très serré, ce qui donne une très bonne limite inférieure, à moins de 10% de la solution entière.



Le pic sur "multi_minKP_5_2.txt" est l'exemple-type de ce qui avait été mentionné en **section 2.2** : $m = 2$ sacs, et on observe un écart d'environ 1.6, ce qui est assez proche de la borne théorique 2 citée précédemment. C'est un exemple où un même objet a été fractionné sur deux sacs différents, et pour donner une solution entière à dû être dupliqué. Le coût se voit alors presque doublé.

- c) L'observation d'un écart d'égalité très faible présente plusieurs implications majeures :
- Le **min-KP**, variante 0/1, reste \mathcal{NP} -difficile. Cependant, sa variante relaxée se résout en temps polynomial. Elle est à la fois rapide et fournit des bornes serrées, ce qui confère une efficacité pratique.
 - Concernant la qualité des bornes, elles sont effectivement extrêmement serrées, proches de l'optimalité entière.
- d) Afin de résoudre la relaxation linéaire du problème en temps polynomial sans utiliser de solveur linéaire, nous pourrions le tourner en un problème de flot à coût minimal sur un réseau biparti. Cela permettrait de rester dans la classe \mathcal{P} , comme le mentionnent Ahuja, Magnanti, and Orlin (1993).

3.2 Interprétation des solutions sur certaines instances

Cette section se concentre sur l'interprétation de résultats sur les instances `multi_minKP_15_3.txt` et `multi_minKP_15_5.txt`

3.2.1 `multi_minKP_15_3.txt`

Mode	Coût total	Poids total
0 – Primal entier	8016.0	854.0
1 – Primal relaxé	7414.859	819.0
2 – Dual relaxé	7414.859	819.0

Table 1: Résultats des trois modèles sur l'instance `multi_minKP_15_3.txt`.

Tout d'abord, nous remarquons que le coût entier dépasse le coût relaxé. C'est un résultat tout à fait attendu, le modèle LP offrant plus de libertés.

Nous remarquons aussi que le théorème de la dualité forte est vérifiée : le primal relaxé coïncide avec le dual relaxé.

3.2.2 `multi_minKP_15_5.txt`

Mode	Coût total	Poids total
0 – Primal entier	n.a. (boucle à l'infini)	—
1 – Primal relaxé	17 267.824	716.0
2 – Dual relaxé	17 267.824	716.0

Table 2: Résultats des trois modèles sur l'instance `multi_minKP_15_5.txt`

Nous remarquons premièrement que le solveur n'a pas trouvé de solution (aucune convergence) pour $m = 5$. Cela peut illustrer l'explosion exponentielle d'un point de vue complexité à mesure que le nombre de sacs augmente. En revanche, la relaxation (primale et duale) reste très efficace, et de plus coïncident exactement ce qui vérifie le théorème de la dualité forte.

Malgré l'impossibilité pratique d'obtenir une solution par le modèle primal entier, les modèles relaxé fournissent une alternative rapide.

4 Conclusion

Nous avons modélisé le problème **min-KP** sous ses variantes mono-sac et multi-sacs. Pour chacune, nous avons établi :

- Le modèle entier 0/1,
- sa relaxation linéaire,
- le programme dual correspondant.

Nos expérimentations montrent que la relaxation linéaire du problème fournit, dans la grande majorité des cas, des bornes très proches de l'optimal du modèle entier.

De plus, la dualité forte est vérifiée sur toutes les instances, ce qui confirme la justesse des différentes formulations primal-dual.

En pratique, nous avons pu remarquer qu'un solveur exact est fort limité lorsque le nombre de sacs augmente.

References

- Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall. ISBN: 0-13-617549-X.
- Caprara, Alberto, Hans Kellerer, and Ulrich Pferschy (2000). “Approximation Schemes for the Multiple Subset-Sum Problem and the Multiple Knapsack Problem”. In: *SIAM Journal on Computing* 31.6, pp. 1891–1912. DOI: 10.1137/S0097539700362674.
- Kellerer, Hans, Ulrich Pferschy, and David Pisinger (2004). *Knapsack Problems*. Springer Series in Operations Research. Springer. Chap. 9 – Linear-Programming Relaxations. ISBN: 978-3-540-40286-2.
- Ricca, Federica, Andrea Scozzari, and Bruno Simeone (2013). “Political Districting: From Classical Models to Recent Approaches”. In: *Annals of Operations Research* 204.1, pp. 271–299. DOI: 10.1007/s10479-012-1267-2.