

	ING2 – Mathematics and Computer Science	
	TP : Late Acceptance Hill-Climbing	
	<i>CY Tech</i>	
	Subject : ML-OPT-IA - Applications	Due date : January 17, 2023
		Number of pages : 2

1 Problème du sac à dos

Soit un ensemble de n objets donnés par des couples (v_i, w_i) où chacun des v_i est une valeur (par exemple en \$) et chacun des w_i est un poids (par exemple en kg). On dispose d'un sac à dos (en anglais *knapsack*) de contenance W . L'objectif est de maximiser la valeur des objets dans le sac sans que celui-ci ne déborde. Pour chaque objet, on peut décider soit de le mettre dans le sac, soit non (les objets ne sont pas fractionnables). On veut donc résoudre le problème d'optimisation linéaire en nombres entiers suivant :

$$\begin{aligned} \max \quad & \sum v_i x_i \\ \text{s.t.} \quad & \sum w_i x_i \leq W \text{ et } x_i \in \{0, 1\} \end{aligned}$$

Chaque configuration peut être codée par n bits qui codent chacun à la décision prise pour l'objet correspondant. Les voisins d'une configuration donnée sont obtenus en sélectionnant un objet et en inversant la décision : s'il n'est pas dedans, on l'ajoute, si il y est on le retire. Chaque configuration a donc exactement n voisins.

Cependant, le voisin obtenu peut être infaisable. Il existe plusieurs méthodes pour résoudre ce problème. En voici deux parmi les plus simples :

- on refuse systématiquement les configurations infaisables ;
- lorsque le sac déborde, on enlève un par un des objets au hasard jusqu'à ce que la configuration soit à nouveau faisable.

Exercice 1.

L'objectif est de résoudre les instances données dans le fichier `knapsack.zip`¹. Le format d'entrée est détaillé dans le fichier `handout.pdf`.

- Mélanger les objets uniformément aléatoirement et évaluer la valeur du sac. Essayer avec plusieurs tirages aléatoires.
- Trier le sac par valeur croissante de chaque objet, donner la solution.
- Faire de même par poids de chaque objet.
- On appelle densité d'un objet le rapport v_i/w_i . Trier les objets par densité décroissante et donner la solution.
- Utiliser l'algorithme du LAHC.

¹Les instances sont issues du MOOC Discrete Optimization enseigné par Pascal Van Hentenryck sur coursera. Ce MOOC est vivement recommandé. Voici le trailer https://www.youtube.com/watch?v=Y2Cv_cdKo0A et l'introduction <https://www.youtube.com/watch?v=2IbJf4oX0xU>

Vous porterez une attention particulière à la pertinence d'utiliser une métaheuristique en fonction de la taille du problème.

2 Le problème du voyageur de commerce

Exercice 2.

Pour "chaque" instance donnée dans le fichier `tsp.zip`,

- a. Utiliser l'algorithme du LAHC pour résoudre l'instance.
- b. Comparer avec l'algorithme du recuit simulé en terme de min, max, moyenne, médiane, écart-type avec *le même nombre d'itérations*.

L'objectif est d'avoir un regard critique sur le paramétrage des constantes, ainsi que le nombre d'itérations nécessaires pour une résolution convenable du problème, en fonction de sa taille.