EISTI	ING2 – Mathématiques et Informatique TP : Late Acceptance Hill-Climbing *EISTI*	
	Matière : **ML-OPT-IA - Applications**	*Échéance* : **5 mars 2018**
		Nombre de pages : **2**

1 Problème du sac à dos

Soit un ensemble de n objets donnés par des couples (v_i, w_i) oú 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 :

$$max \ \ \Sigma v_i x_i$$
$$s.c. \ \ \ \Sigma v_i x_i \leq W \text{ et } x_i \in \{0, 1\}$$

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.

a. Résoudre les instances données dans le fichier `knapsack.zip` [1]. Le format d'entrée est détaillé dans le fichier `handout.pdf`.

2 Le problème du voyageur de commerce

Chaque tournée est une permutation des sommets. On définit comme voisin d'une tournée la tournée obtenue en sélectionnant deux arêtes non adjacentes et

1. Les 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=2IbJf4oXOxU`

en les "croisant" de l'unique manière possible.

Chaque tournée a donc $\frac{n(n-3)}{2}$ tournées voisines. Attention, le parcours le long du cycle est orienté (même si le graphe ne l'est pas), ce qui exige de parcourir l'une des sous-chaînes de la tournée voisine en sens inverse.

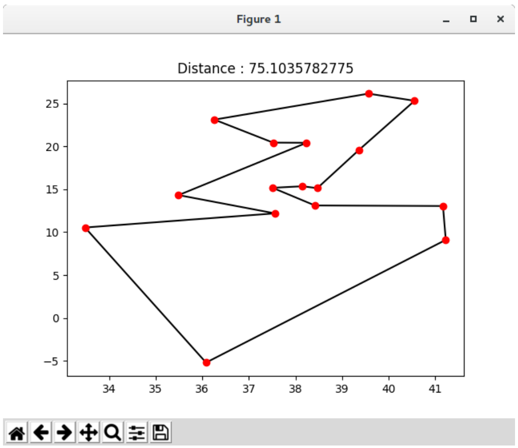

FIGURE 1 – Affichage de résolution du TSP.

Exercice 2.

a. Résoudre les instances données dans le fichier `tsp.zip`. Le format d'entrée est détaillé dans le fichier `handout.pdf`. On notera entre autres que l'on dispose de coordonnées de points et qu'il faut donc calculer les distances nécessaires.