

# Optimisation combinatoire par méthodes incomplètes

## Expérimentations sur le problème du sac à dos

2012

Le TP s'étale sur 3 séances de 2h. Son objectif est de tester certaines méthodes incomplètes sur le cas particulier du sac à dos.

On rappelle les données du problème général :

soit  $E = \{o_1, o_2, \dots, o_n\}$  un ensemble d'objet.

Chaque objet  $o_i$  possède un poids  $w_i$  et rapporte une valeur  $c_i$ .

Soit  $W$  le poids total que peut contenir le sac.

On cherche à remplir le sac avec un ensemble d'objet  $S \subset E$  tel que  $\sum_{o_i \in S} w_i \leq W$

en maximisant la somme des gains  $\sum_{o_i \in S} c_i$

Pour cela vous implémenterez en Java les classes qui permettront d'appliquer à ce problème les méthodes locales suivantes (il faudra définir le voisinage d'une solution) :

- une recherche opportuniste de base (un mouvement est fait au hasard dans le voisinage s'il améliore la solution courante)
- une recherche le long de la plus grande pente (mouvement sur le meilleur voisin)
- une recherche par la méthode tabou ou par recuit simulé (au choix selon les groupes).

Des instances de tests seront fournies dont il faudra essayer d'approcher l'optimum.

Une instance sera stocké dans un fichier texte au format suivant (abstrait à gauche/exemple à droite) :

W
w1 c1
w2 c2
w3 c3
...

37
50 10
13 6
20 3
...

Vous rendrez un petit rapport qui résumera vos résultats selon les méthodes et les réglages effectués. (2 pages maxi), ainsi qu'une archive du code, par mail à [philippe.muller@irit.fr](mailto:philippe.muller@irit.fr).

**Bonus** Comparer avec un autre algorithme approché vu en cours (meilleure densité, algorithme récursif avec approximation, ...).

**Conseils** Pour stocker un ensemble d'objets, vous pouvez prendre un type Collection (comme HashSet).

Pour ne pas vous embêter avec la lecture des données, utiliser quelque chose du genre:

```
try{
    BufferedReader fic = new BufferedReader(new FileReader(filename));

    objets = new HashSet<Objet>();

    while((ligne = fic.readLine()) != null){
        s = ligne.split("\\s+");
        if(debut == 0) {
            capacity=Long.valueOf(s[0]);
            debut=1;
        } else {
            o=new Objet(Long.valueOf(s[0]),Long.valueOf(s[1]));
            objets.add(o);
        }
    }
}catch(IOException ioe){ }
```

Pour prendre un objet au hasard dans une collection, faire par exemple

```
Random rand;
rand = new Random();
(Objet)one.toArray()[rand.nextInt(one.size())];
```