

Dans ce TD, nous allons présenter trois méthodes d'optimisation : la recherche exhaustive, les algorithmes gloutons, la recherche locale. Nous travaillerons sur les données de l'exercice 5 du TD 1 et sur un petit problème de sac à dos.

### Recherche exhaustive

La recherche exhaustive est la méthode la plus simple, mais aussi souvent la plus coûteuse, pour trouver la ou les solutions optimales à un problème. Comme son nom l'indique, on va énumérer l'ensemble des solutions possibles, les parcourir une à une et conserver celle qui retourne le meilleur résultat. Comme le nombre de solutions possibles est fini, la recherche finit toujours. Toutefois, si l'on a  $N$  variables binaires, on aura à énumérer  $2^N$  solutions possibles.

#### Exercice 1 :

Proposez un algorithme de recherche exhaustive pour le problème de sac à dos, et écrivez son pseudo code. Faites le "tourner" sur les données.

Proposez un algorithme de recherche exhaustive pour le problème des entrepôts, et écrivez son pseudo code.

### Algorithme glouton

Un algorithme glouton est un algorithme approché pour lequel, à chaque itération, on fixe la valeur d'une (ou plusieurs) des variables décrivant le problème sans remettre en cause les choix antérieurs. Le principe est de partir d'une solution incomplète (voire totalement indéterminée), que l'on complète, de proche en proche, en effectuant des choix définitifs. A chaque étape, on prend la décision qui permet d'optimiser la fonction objectif pour la solution partielle. Bien que très simples, ces algorithmes peuvent fournir pour certains problèmes de très bonnes solutions (par exemple, l'algorithme de Kruskal pour la recherche d'un arbre couvrant de poids minimum). Le principal avantage est la rapidité des ces algorithmes. En revanche, la solution trouvée peut être arbitrairement loin de la solution optimale.

#### Exercice 2 :

Proposez un algorithme glouton pour le problème des entrepôts, et écrivez son pseudo-code. Faites le "tourner" sur les données du TD 1, en représentant graphiquement les étapes et les flux.

### Recherche locale

Les méthodes de recherche locale regroupent un ensemble de méthodes basées sur la notion de voisinage de solutions dans un espace de recherche. Une méthode de recherche locale se caractérise principalement par un espace des configurations, une relation de voisinage définie entre les configurations, un mécanisme d'exploration du voisinage et une fonction d'évaluation de la qualité de chaque configuration.

Une méthode de recherche locale débute avec une solution initiale et explore l'espace de recherche en remplaçant, à chaque itération, la solution courante par une de ses voisines. Ce processus itératif, qui peut être considéré comme une suite de mouvements dans l'espace de recherche, s'arrête quand certaines conditions prédéfinies sont atteintes et retourne la meilleure solution trouvée.

Il existe deux principales méthodes de choix d'un voisin : il est possible de les énumérer jusqu'à en trouver un qui améliore la fonction d'évaluation, on parle alors de *première amélioration* ou *première descente* ; ou on peut rechercher le meilleur voisin, on parle alors de *meilleure amélioration* ou *plus forte descente*.

Comme l'espace des solutions est fini, cette procédure s'arrête toujours, et la dernière configuration trouvée ne possède pas de voisin strictement meilleur qu'elle-même. Autrement dit, la recherche locale retourne toujours un optimum local.

#### Exercice 3 :

Proposez un algorithme de recherche locale et écrivez son pseudo-code.