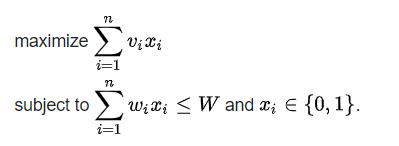
Problème du sac à dos

**Explication du problème :**

* Remplir le sac à dos de sorte de maximiser la valeur des objets inclus tout en respectant la contrainte de poids du sac.

**Formulation mathématique :**

* On suppose qu’on peut apporter une fraction de chaque objet i:

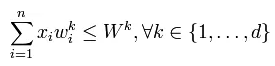


**Différentes variantes:**

* **Sac à dos multidimensionnel :**

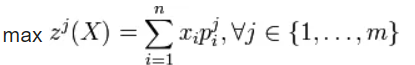
On considère ici que le sac à dos à dimensions, avec *d > 0*. Par exemple, on peut imaginer une boîte. Chaque objet a trois dimensions, et il ne faut déborder sur aucune des dimensions.

La formule mathématique de cette variante est :



* **Sac à dos multi-objectif :**

Une variante du problème consiste, à partir d'objets ayant plusieurs valeurs, à maximiser plusieurs fonctions objectives, c'est le problème du sac à dos multi-objectif. On rentre donc dans le domaine de l'optimisation multi-objectif.



**\sum_{i=1}^n x_iw_i \le W**

* **Sac à dos quadratique :**

Le problème de sac à dos quadratique est noté *QKP*. On a ici un gain *gij* supplémentaire lorsque deux objets (*i* et *j*) sont pris simultanément. Par exemple, disons que vous souhaitez maximiser la qualité de votre café lors d'une expédition avec un sac à dos. On peut comprendre qu'il est plus intéressant d'apporter une cuillère et un sucre plutôt qu'un seul des deux.

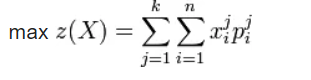
La fonction objectif s'écrit alors :



* **Sac à dos multiple :**

Le problème de sac à dos multiple (MKP) consiste à répartir un ensemble d'objets dans plusieurs sacs à dos de capacités différentes. La valeur d'un objet dépend maintenant du sac dans lequel il est placé.

Si on a k sacs à dos, on notera x_i^j=1 si l'objet i est placé dans le sac j. La formulation du problème devient :



sous contraintes :

\sum_{i=1}^n x_i^jw_i \le W^j, \forall j \in \{1, \dots, k\} : on ne dépasse pas la capacité des sacs ;

\sum_{j=1}^k x_i^j \le 1, \forall i \in \{1, \dots, n\} : un objet n'est mis que dans un sac.

**Domaine d’utilisation :**

* Machine Learning(décision tree)
* cryptographie
* Chargement de bateau ou d'avion
* Découpe de matériaux

**Résolution du problème :**

* **Méthode exacte :**

Méthode exacte, qui demande un temps de calcul beaucoup plus long (si le problème est difficile à résoudre). Il n’existe pas une méthode exacte universellement plus rapide que toutes les autres

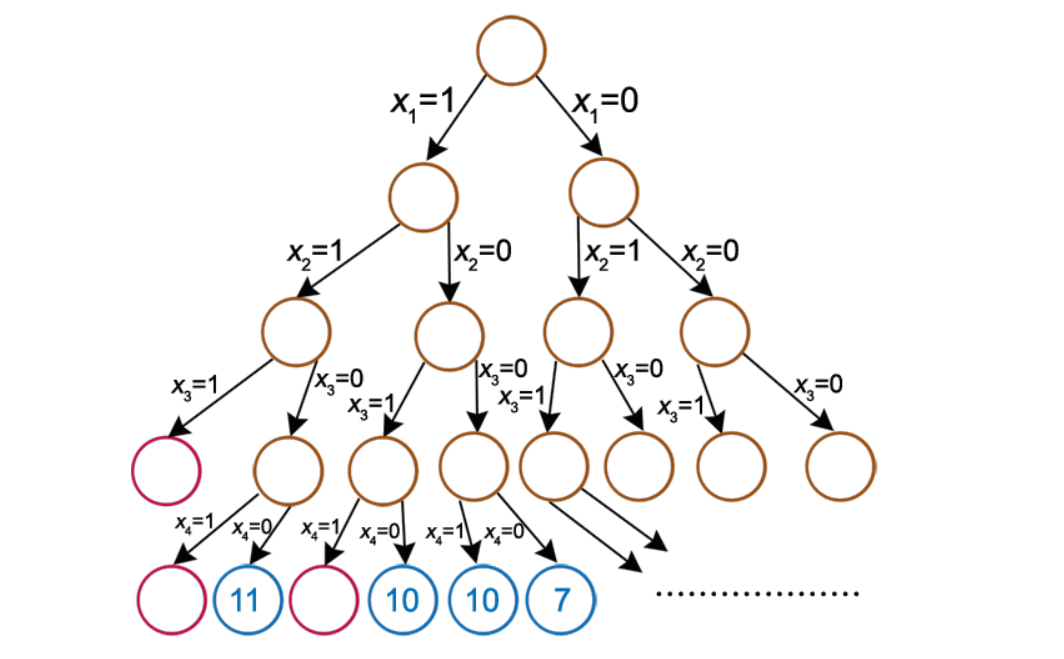
Nous allons présenter un exemple d’algorithme de ce type, nommé procédure par séparation et évaluation (PSE), ou en anglais branch and bound. Nous exposerons ici qu’une version simplifiée d’une PSE.

nous utilisons un « arbre de recherche » constitué :

* de nœuds ou sommets, où un nœud représente une étape de construction de la solution
* d’arcs pour indiquer certains choix faits pour construire la solution.

À partir d’une étape N, l’algorithme construit deux nouvelles étapes : dans celle de gauche, on conserve toutes les décisions prises à l’étape N et on ajoute la décision de mettre un nouvel objet (ici l’objet i) dans le sac ; dans celle de droite, on conserve également toutes les décisions prises à l’étape N et on ajoute la décision de ne pas mettre l’objet i.

L’arbre de recherche commence par un seul nœud, où aucune décision n’a été prise pour les objets. Une fois tous les objets sélectionnés, les nœuds finaux, aussi appelés nœuds feuilles, représentent chacun une solution finale.



Les nœuds en rouge représentent une solution impossible.

Les nœuds en bleu sont des nœuds feuilles correspondant à une solution réalisable.

À la fin de l’algorithme, il suffit de calculer la valeur du sac pour chaque nœud feuille et de prendre la solution avec la plus grande valeur.

**Algorithme de résolution de problème :**

* Méthodes exactes  :
  + Programmation dynamique
  + L’algorithme A
  + Simplexe
  + Backtracking
* Méthodes approchées  :

A base de solution unique :

* Le Recuit Simulé \*
* La recherche tabou

             A base de population de solution :

* L’algorithme évolutionnaire
* L’algorithme génétique \*
* La recherche par harmonies

A base d’intelligence par essaim :

* Le Système immunitaire
* L’optimisation par Colonies de fourmis \*
* L’optimisation par Colonies d’abeilles
* La recherche coucou
* Essaim de particule \*