



Informatique

## Fiche

## Introduction à la programmation en Python Informatique



1

Présentation

2

## 1 Présentation

**Définition Problème d'optimisation** Un problème d'optimisation est un problème dans lequel on souhaite minimiser (ou maximiser) une fonction dépendant d'un jeu de paramètres.

### ■ Exemple

#### **Problème du rendu de monnaie**

Supposons qu'un commerçant doit rendre 24,95 € à un client. Il dispose de pièces et de billets d'une valeur de 100, 50, 20, 10, 5, 2, 1, 0,50, 0,20, 0,10, 0,05, 0,02, 0,01 €. Quelle(s) combinaison(s) de billets et de pièces permet(tent) de rendre le moins de billets et de pièces au client?

#### **Problème du sac à dos**

Supposons qu'on doit remplir un sac à dos ayant une capacité limitée (15 kg). Pour remplir ce sac on dispose de différentes boîtes de masses et de valeurs données (1 kg – 1€, 2 kg – 1€, 1 kg – 2€, 4 kg – 10€, 10 kg – 4€). Comment remplir le sac pour maximiser la valeur transportée?

Pour résoudre des problèmes d'optimisation, il existe différentes stratégies. On peut par exemple opter pour la **force brute**. Dans ce cas, on recherche l'ensemble des combinaisons de paramètres d'entrée permettant de répondre à l'objectif. Parmi l'ensemble des solutions déterminées, on recherche celle(s) qui répond(ent) « le mieux » à l'objectif.

Ce type de résolution présente un inconvénient majeur : suivant le type de problème, l'évaluation de l'ensemble des combinaisons peut prendre un temps de calcul très important.

**Définition Algorithme gloutons** Un algorithme glouton permet de trouver un optimum local permettant de répondre au problème. Pour cela, à chaque itération de l'algorithme, on va choisir un optimum local.

- **Avantage :** facilité de la mise en œuvre.
- **Inconvénient :** l'optimum trouvé est un optimum local (et pas un optimum global). La solution n'est donc pas forcément la meilleure.
- **Situations canoniques :** dans ce cas, le résultat de l'algorithme glouton est l'optimum global.

### ■ Exemple

#### **Problème du rendu de monnaie**

On souhaite rendre 31€ avec des pièces de 22, 10, et 1€. L'algorithme glouton proposera 1 pièce de 22€ et 9 pièces de 1€. Or la solution optimale est 3 pièces de 10€ et 1 pièce d'1€.

Le fait d'avoir des pièces de 20, 10, 5, 2 et 1€ (comme dans notre système monétaire) permet à l'algorithme glouton de donner la solution optimale. La situation est canonique.