Les heuristiques sont des méthodes spécifiques qui exploitent au mieux la structure du problème dans le but de trouver une solution raisonnable (non nécessairement optimale) en un temps réduit. L'utilisation de ce type d'algorithmes s'impose car les méthodes de résolution exactes sont de complexité exponentielle, et échouent à trouver la solution pour des instances de tailles moyennes voire petites, comme on l'a constaté lors du chapitre précédant. L'usage des heuristiques est donc pertinent pour surmonter ces limites.

Dans ce chapitre, nous allons présenter la conception détaillée des heuristiques sur lesquelles notre choix d'implémentation s'est porté et qui sont:

1. Next Fit (NF)
2. Next Fit Decreasing (NFD)
3. First Fit (FF)
4. First Fit Descreasing (FFD)
5. Best Fit (BF)
6. Best Fit Decreasing (BFD)

Dans le but d'explorer ces méthodes, comparer leurs performances, montrer leurs avantages et découvrir leurs limites, nous effectuerons des tests empiriques et comparatifs sur le même benchmark utilisé pour les tests des méthodes exactes (Benchmark Scholl).

1 Next Fit (NF)

1.1 Principe

Si l'article tient dans la même boite que l'article précédent, il est placé avec ce dernier. Sinon, on ouvre une nouvelle boite et le mettre là-dedans.

- NF est un algorithme simple d'une complexité de O(n).

1.2 Pseudocode

Algorithm 1 Next Fit

```
for Tous les articles i = 1, 2,. . . , n do
  if l'article i s'inscrit dans la boîte actuelle then
    Ranger l'article i dans la boîte actuelle
  else
    Créer une nouvelle boîte, en faire la boîte actuelle et ranger l'article i
    dedant.
  end if
end for
```

2 Next Fit Decreasing (NFD)

2.1 Principe

Le NFD est une amélioration de l'algorithme Next-Fit. Cet algorithme ordonne les articles par ordre décroissant des poids,puis applique l'algorithme NF.

2.2 Pseudocode:

Algorithm 2 Next Fit Decreasing

```
Triez les articles par ordre décroissant
Appliquer Next-Fit à la liste triée
```

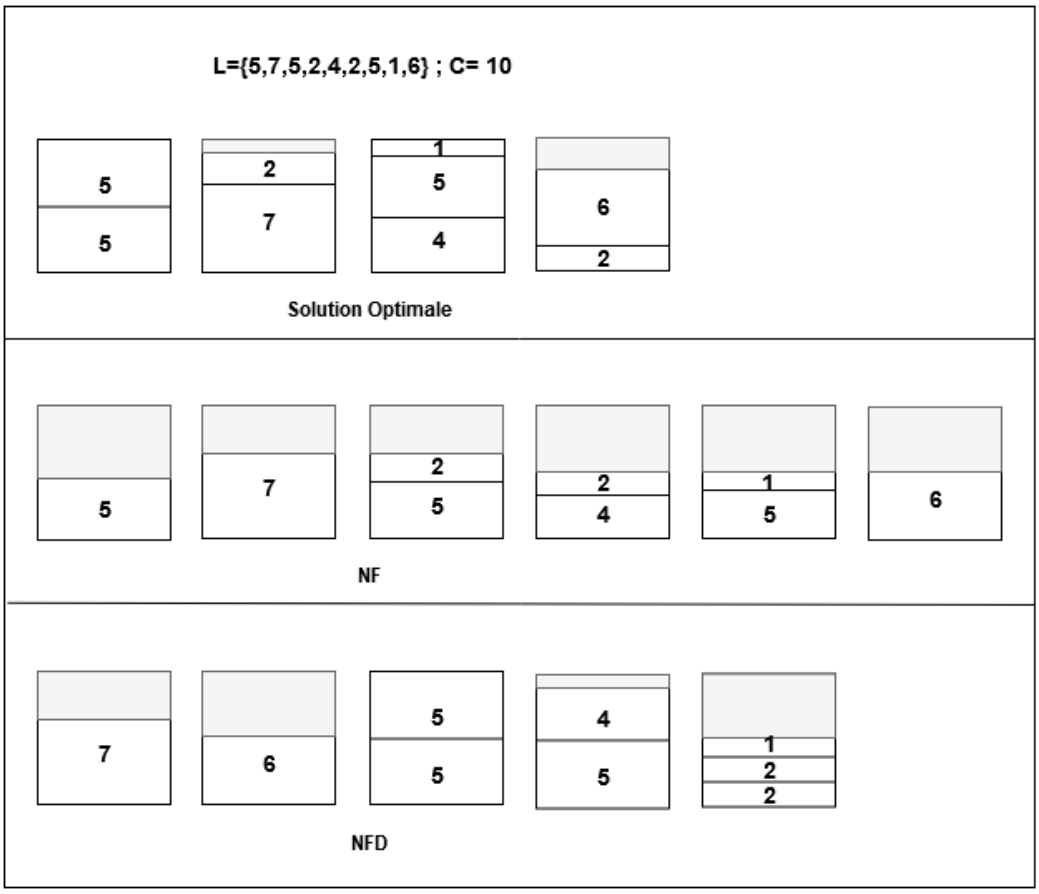

Figure 1: Exemple NF et NFD

3 First Fit (FF)

3.1 principe

Ranger chaque article courant dans la première boîte, entre celles déjà ouvertes, qui peut le contenir sinon ouvrir une nouvelle boîte et on le range dedans.

- L'algorithme First Fit implémenté a une complexité de $O(n^2)$.

3.2 Pseudocode

Algorithm 3 First Fit

```
for Tous les articles i = 1, 2,. . . , n do
  for Toutes les boîtes j = 1, 2,. . . m do
    if l'article i s'inscrit dans la boîte j then
      Ranger l'article i dans la boîte j
      Quitter la boucle (passer à l'article suivant)
    end if
  end for
  if l'article i ne rentre dans aucune boîte disponible then
    Créer une nouvelle boîte et ranger l'article i dedans
  end if
end for
```

4 First Fit Descreasing (FFD)

4.1 Principe

Le FFD est une amélioration de l'algorithme First-Fit. Cet algorithme ordonne les poids dans un ordre décroissant puis leur applique l'algorithme FF.

- L'algorithme First Fit peut être implémenté en O(nlog n) en utilisant les arbres de recherche binaires

4.2 Pseudocode

Algorithm 4 First Fit Decreasing

```
Triez les articles par ordre décroissant
Appliquer First-Fit à la liste triée
```

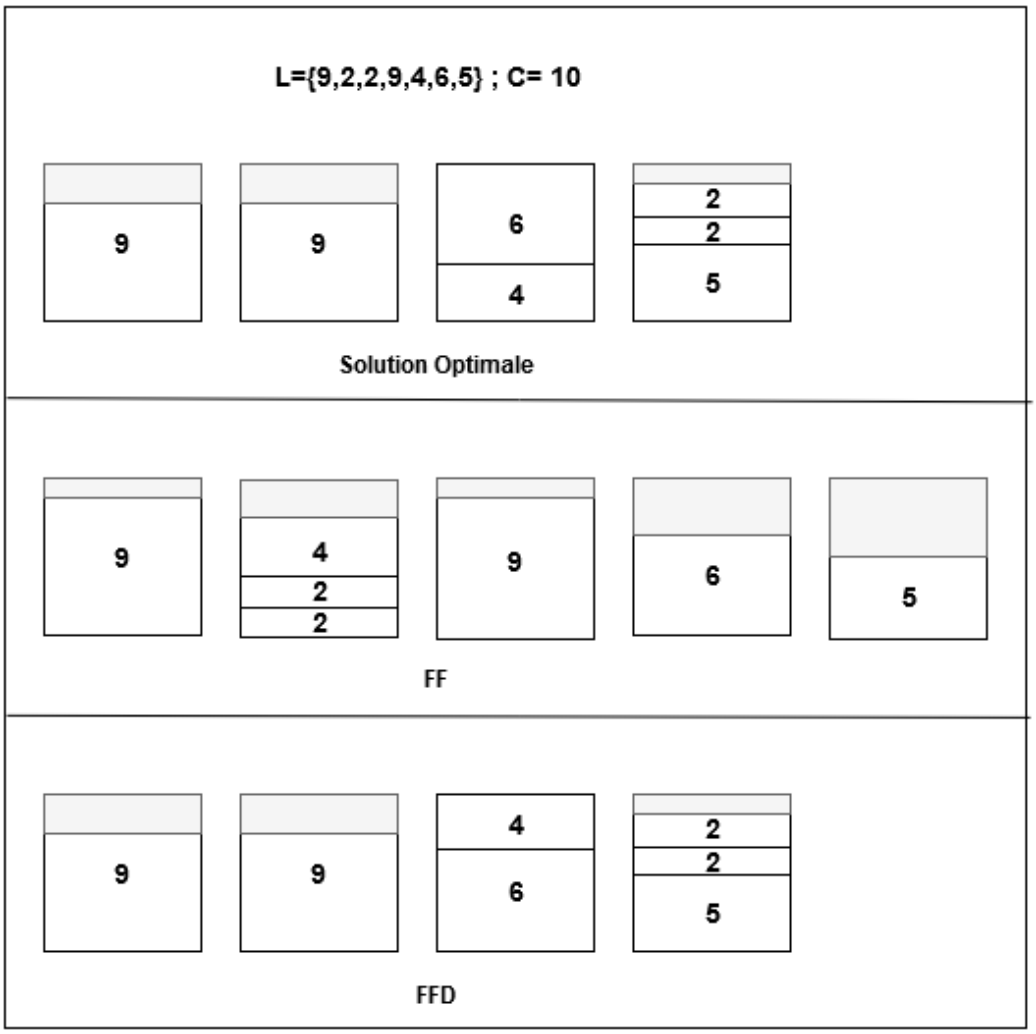

Figure 2: Exemple FF et FFD

5 Best Fit (BF)

5.1 principe

Ranger chaque article courant dans la boîte la mieux remplie, entre celles déjà ouvertes, qui peut le contenir sinon ouvrir une nouvelle boîte et le ranger dedans.

- L'algorithme Best Fit implémenté a une complexité de O(n^2).

5.2 Pseudocode

Algorithm 5 Best Fit

for Tous les articles i = 1, 2,. . . , n **do**
  for Tous les boîtes j = 1, 2,. . . m **do**
    if l'article i s'inscrit dans la boîte j **then**
      Calculer la capacité restante dans la boîte j
    end if
  end for
  Ranger l'article i dans la boîte j, où j est la boîte ayant la capacité restante minimale après avoir ajouté l'article(c'est-à-dire que "l'article lui convient le mieux").
  if une telle boîte n'existe pas (l'article ne peut être rangé dans aucune boîte) **then**
    Créer une nouvelle boîte et ranger l'article i dedans
  end if
end for

6 Best Fit Decreasing (BFD)

6.1 principe

Le BFD est une amélioration de l'algorithme Best-Fit. Cet algorithme ordonne les poids dans un ordre décroissant puis applique l'algorithme BF.

6.2 Pseudocode

Algorithm 6 Best Fit Decreasing

Triez les articles par ordre décroissant
Appliquer Best-Fit à la liste triée

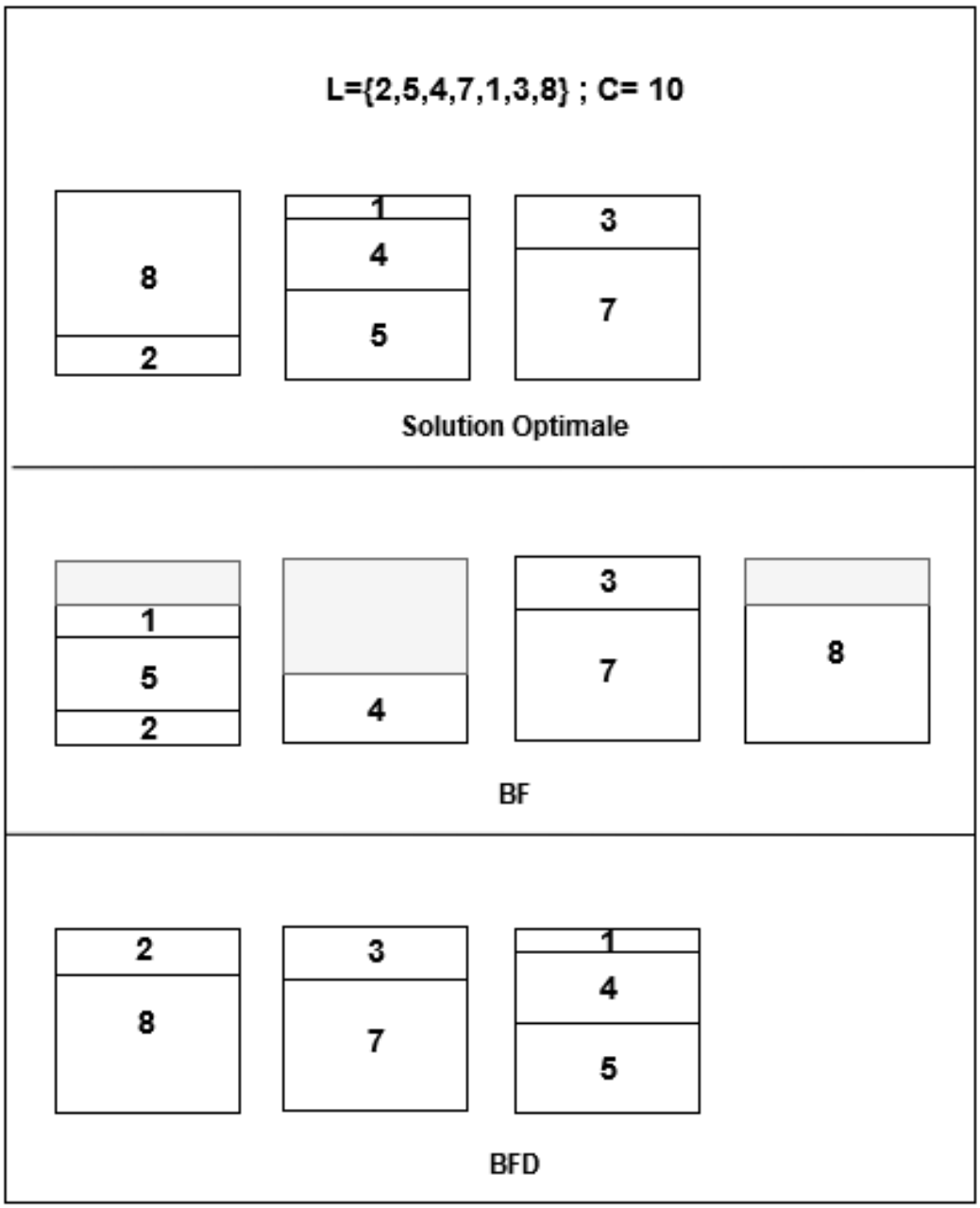

Figure 3: Exemple BF et BFD