

Projet Bin Packing 1D

Adrien GARBANI

8 juin 2021

Les instructions pour exécuter le code peuvent être trouvés dans le fichier `README.md`

1 Borne inférieure des jeux de données

La borne inférieure du nombre de bin à utiliser est obtenue en divisant la somme de la taille de tous les objets par la taille d'un bin.

nom	taille d'un bin	nombre d'items	nombre de bin
binpack1d_00.txt	9	24	13
binpack1d_01.txt	150	250	99
binpack1d_02.txt	150	500	198
binpack1d_03.txt	1000	60	20
binpack1d_04.txt	1000	120	40
binpack1d_05.txt	1000	249	83
binpack1d_06.txt	1000	501	167
binpack1d_11.txt	100	50	25
binpack1d_12.txt	120	50	26
binpack1d_13.txt	120	500	252
binpack1d_14.txt	150	500	215
binpack1d_21.txt	10000	141	11
binpack1d_31.txt	1000	160	61

Résultats obtenus avec la commande : `./Project_opti.exe question 1.`

2 Résolution avec FirstFitDecreasing

L'implémentation peut être trouvé dans la fonction `FileData::first_fit_decreasing()` dans le fichier `src/FileData.cpp`.

nom	borne inférieure	FirstFitDecreasing
binpack1d_00.txt	13	13
binpack1d_01.txt	99	100
binpack1d_02.txt	198	201
binpack1d_03.txt	20	23
binpack1d_04.txt	40	45
binpack1d_05.txt	83	94
binpack1d_06.txt	167	190
binpack1d_11.txt	25	25
binpack1d_12.txt	26	29
binpack1d_13.txt	252	258
binpack1d_14.txt	215	220
binpack1d_21.txt	11	12
binpack1d_31.txt	61	62

Résultats obtenus avec la commande : `./Project_opti.exe question 2.`

3 Résolution linéaire

3.1 Solution de binpack1d_00.txt

La résolution linéaire donne un résultat optimal à 13 bins en environ 38ms. Le contenu de chaque bin étant :

items	total
6	6
6,3	9
5,3	8
5,4	9
5,2,2	9
4,5	9
4,5	9
4,4	8
2,7	9
2,7	9
5,4	9
8	8
8	8

Résultats obtenus avec la commande :

```
./Project_opti.exe question 3 file resources/binpack1d_00.txt verbose.
```

3.2 Limite de la résolution linéaire

4 Générateurs aléatoires

Les générateurs de solution aléatoires peuvent être trouvés dans les fonctions `FileData::solve_simple()` et `FileData::first_fit_random()` dans le fichier `src/FileData.cpp`.

Leur résultats peuvent être obtenus avec la commande :

```
./Project_opti.exe question 4 file resources/binpack1d_00.txt verbose.
```

5 Opérateurs de voisinage

Les opérateurs peuvent être trouvés dans les fonctions `Solution::move_item(...)` et `Solution::swap_items(...)` dans le fichier `src/Solution.cpp`.

Une démonstration des opérateurs peut être obtenue avec la commande : `./Project_opti.exe question 5`.

6 Recuit simulé

L'implémentation du recuit simulé peut être trouvé dans la fonction `algo::recuit()` dans le fichier `src/algo.cpp`. Il est appelé dans `src/question6.cpp` à la ligne 129.

Deux scénarios seront étudiés : recuit en partant de la solution simple (un item par bin) avec comme objectif le nombre de bin et recuit en partant de d'un first fit aléatoire et avec comme fonction objectif la somme des carrés des sommes des items de chaque bin.

6.1 Scénario 1

`./Project_opti.exe question 6 file resources/binpack1d_XX.txt objectif 1 initial 1 step 100000 temperature 1.44 lambda 0.99`

J'ai choisi arbitrairement de paramétrer la température initiale à 1.44 et le lambda à 0.99. L'étude devra permettre de raffiner ou justifier ces valeurs. La température a été choisie pour que la probabilité initiale de choisir l'option soit environ 0.5 pour une différence d'énergie de 1.

J'ai pris un nombre de pas max de 10000 pour obtenir des résultats dans un temps raisonnable.

nom	optimal trouvé	FirstFitDecreasing	borne inférieure	temps (ms)	pas
binpack1d_00.txt	13	13	13	3	500
binpack1d_01.txt	107	100	99	3495	100000
binpack1d_02.txt	217	201	198	6910	100000
binpack1d_03.txt	22	23	20	862	100000
binpack1d_04.txt	43	45	40	1643	100000
binpack1d_05.txt	90	94	83	3423	100000
binpack1d_06.txt	182	190	167	6921	100000
binpack1d_11.txt	27	25	25	787	100000
binpack1d_12.txt	29	29	26	769	100000
binpack1d_13.txt	282	258	252	7241	100000
binpack1d_14.txt	235	220	215	6935	100000
binpack1d_21.txt	12	12	11	962	100000
binpack1d_31.txt	67	62	61	2213	100000

6.2 Scénario 2

`./Project_opti.exe question 6 file resources/binpack1d_XX.txt objectif 2 initial 2 step 100000 temperature 1.44 lambda 0.99`

J'ai choisi arbitrairement de paramétrer la température initiale à 1.44 et le lambda à 0.99. L'étude devra permettre de raffiner ou justifier ces valeurs.

La température a été choisie pour que la probabilité initiale de choisir l'option soit environ 0.5 pour une différence d'énergie de 1. Le lambda a été choisi pour que la probabilité de choix soit de 0.1 après 100 étapes.

J'ai pris un nombre de pas max de 10000 pour obtenir des résultats dans un temps raisonnable.

nom	optimal trouvé	FirstFitDecreasing	borne inférieure	temps	pas
binpack1d_00.txt	13	13	13	0	0
binpack1d_01.txt	99	100	99	371	12143
binpack1d_02.txt	198	201	198	5000	81322
binpack1d_03.txt	21	23	20	690	100000
binpack1d_04.txt	41	45	40	1381	100000
binpack1d_05.txt	83	94	83	2272	79484
binpack1d_06.txt	168	190	167	5635	100000
binpack1d_11.txt	25	25	25	10	1499
binpack1d_12.txt	29	29	26	593	100000
binpack1d_13.txt	259	258	252	5498	100000
binpack1d_14.txt	216	220	215	5500	100000
binpack1d_21.txt	11	12	11	5	598
binpack1d_31.txt	62	62	61	1657	100000

On remarque que le deuxième scénario donne toujours de meilleurs résultats et est toujours plus rapide.

6.3 Scénario 2 avec $T0 = 4.48$ et $\lambda = 0.98$

`./Project_opti.exe question 6 file resources/binpack1d_XX.txt objectif 2 initial 2 step 200000 temperature 4.48 lambda 0.98`

Je choisis la température pour que la probabilité initiale de choisir l'option soit environ 0.8 pour une différence d'énergie de 1. Je choisis le lambda pour que la probabilité de choix soit de 0.1 après 100 étapes.

nom	optimal trouvé	FirstFitDecreasing	borne inférieure	temps	pas
binpack1d_00.txt	13	13	13	1	138
binpack1d_01.txt	99	100	99	387	12637
binpack1d_02.txt	198	201	198	5456	88182
binpack1d_03.txt	21	23	20	1326	200000
binpack1d_04.txt	41	45	40	2796	200000
binpack1d_05.txt	83	94	83	1819	68295
binpack1d_06.txt	167	190	167	10911	190307
binpack1d_11.txt	26	25	25	1244	200000
binpack1d_12.txt	29	29	26	1185	200000
binpack1d_13.txt	258	258	252	11174	200000
binpack1d_14.txt	216	220	215	11792	200000
binpack1d_21.txt	11	12	11	4	482
binpack1d_31.txt	62	62	61	3592	200000

6.4 Scénario 2 avec $T0 = 4.48$ et $\lambda = 0.998$

`./Project_opti.exe question 6 file resources/binpack1d_XX.txt objectif 2 initial 2 step 200000 temperature 4.48 lambda 0.998`

Je choisis la température pour que la probabilité initiale de choisir l'option soit environ 0.8 pour une différence d'énergie de 1. Je choisis le lambda pour que la probabilité de choix soit de 0.1 après 1000 étapes.

nom	optimal trouvé	FirstFitDecreasing	borne inférieure	temps	pas
binpack1d_00.txt	13	13	13	1	87
binpack1d_01.txt	99	100	99	365	12432
binpack1d_02.txt	198	201	198	1793	30088
binpack1d_03.txt	21	23	20	1324	200000
binpack1d_04.txt	41	45	40	2658	200000
binpack1d_05.txt	83	94	83	3100	110480
binpack1d_06.txt	168	190	167	11972	200000
binpack1d_11.txt	25	25	25	7	1071
binpack1d_12.txt	29	29	26	1484	200000
binpack1d_13.txt	258	258	252	11626	200000
binpack1d_14.txt	217	220	215	11929	200000
binpack1d_21.txt	11	12	11	9	1116
binpack1d_31.txt	63	62	61	3135	200000

On remarque que certains jeux de données atteignent un meilleur résultat avec un $T0$ plus grand, en effet ils ont plus de possibilité de sortir d'un minimum local au début de l'algorithme. En augmentant lambda on permet à certains jeux de données de converger plus vite vers une meilleure solution, mais d'autres jeux de données convergeront moins facilement car ils ne sortiront pas d'un minimum local.

7 Tabu Search

L'implémentation de la recherche tabou peut être trouvée dans la fonction `algo::tabou()` dans le fichier `src/algo.cpp`. Elle est appelée dans `src/question7.cpp` à la ligne 79.

`./Project_opti.exe question 7 file resources/binpack1d_XX.txt objectif 2 initial 2 step 100 buffer 10`

Pour le jeu de données `binpack1d_01.txt` avec un nombre de pas maximum de 100 et un buffer de 10 l'exécution prend 146 secondes et ne trouve pas un résultat optimal.

8 Comparaison

La méthode tabou est beaucoup plus lente que la méthode du recuit simulé. En effet elle demande de visiter tous les voisins de la solution courante à chaque pas, ce qui prend beaucoup de temps. Je n'ai pas trouvé de manière plus efficace d'implémenter la méthode tabou.