

# RAPPORT D'ANALYSE

DORYAN BONIFASSI  
GUILLAUME CASAGRANDE – JEREMY JUNAC  
PIERRE-EMMANUEL NOVAC



# INTRODUCTION

Les problèmes de bin-packing ont toujours été sensibles du fait de leur NP difficulté. Cependant, de nombreuses heuristiques répondent plus ou moins au problème.

Dans ce document, nous analyserons cinq algorithmes différents afin de montrer leur efficacité les uns par rapport aux autres. Nous vous présenterons ensuite les résultats de nos tests. Enfin, nous interpréterons ces données afin de vous proposer la meilleure solution à votre problème.

Dans ce rapport, nous parlons de camions comme étant le conteneur dans lequel nous souhaitons ranger nos objets.

Avant tout, commençons par présenter succinctement les algorithmes que nous allons traiter.

## CONTENU

Introduction.....	1
Description des algorithmes .....	3
Next fit.....	3
Worst fit.....	3
First fit.....	3
Best fit.....	4
Almost worst fit.....	4
Analyse.....	5
Métriques .....	5
Analyse approfondie de chaque algorithme.....	6
REmplissage des camions par Next Fit : .....	8
Remplissage des camions par Worst Fit : .....	9
Remplissage des camions par First Fit : .....	10
Remplissage des camions par Best Fit : .....	11
Remplissage des camions par Almost Worst Fit : .....	12
Analyse globale sur la totalité de nos exemples. ....	13
Stabilité des algorithmes.....	14
Interprétation .....	16
Généralisation.....	16
Un cas réel : les plus gros poids arrivent en premier .....	16
Recommandation .....	18

# DESCRIPTION DES ALGORITHMES

## NEXT FIT

Cet algorithme essaye de ranger l'objet dans le dernier camion. S'il ne rentre pas dans ce camion, il est rangé dans un nouveau camion vide. La complexité est de  $O(n)$  avec  $n$  le nombre d'objet à ranger. On ne parcourt jamais les précédents camions.

## WORST FIT

Cet algorithme essaye de ranger l'objet dans le camion le moins rempli. S'il ne rentre pas dans ce camion, il est rangé dans un nouveau camion vide. La complexité est de  $O(n^2)$  dans le pire des cas avec  $n$  le nombre d'objet à ranger, où chaque camion ne peut contenir qu'un objet. Il peut être optimisé en conservant un ordre dans les camions à l'aide par exemple d'un tas min, dans lequel on ne consulterait que le sommet pour ranger un objet. La complexité serait alors de  $O(n \cdot \log(n))$ .

## FIRST FIT

Cet algorithme essaye de ranger l'objet dans n'importe quel camion en les parcourant depuis le début. S'il ne rentre dans aucun camion, il est rangé dans un nouveau camion vide. La complexité est en  $O(n^2)$  dans le pire des cas avec  $n$  le nombre d'objet à ranger, où chaque camion ne peut contenir qu'un objet.

## BEST FIT

De manière similaire à « Worst fit », cet algorithme essaye de ranger l'objet dans le camion le plus rempli dans lequel il peut rentrer. S'il ne rentre dans aucun camion, il est rangé dans un nouveau camion vide. La complexité est aussi de  $O(n^2)$ , et la même optimisation peut être appliquée avec un tas max. Elle serait cependant un peu moins efficace puisqu'il faudrait redescendre le tas jusqu'à trouver un camion où l'objet peut rentrer, sans oublier de supprimer les camions pleins.

## ALMOST WORST FIT

De manière similaire à « Worst fit », cet algorithme essaye de ranger l'objet dans le deuxième camion le moins rempli, ou dans le camion le moins rempli s'il n'existe pas d'autre camion qui pourrait contenir l'objet. La complexité est ici moins bonne puisqu'on tri les camions entre eux pour chaque objets, elle est donc de  $O(n^2 \cdot \log(n))$  avec un algorithme de tri efficace. On peut appliquer la même optimisation que « Worst fit », la complexité serait alors aussi de  $O(n \cdot \log(n))$ .

# ANALYSE

## MÉTRIQUES

Nous jugeons les différents algorithmes grâce à certains critères.

Premièrement, le nombre de camions utilisés, car c'est là le cœur même du problème. Cette métrique bien que simple reste primordiale. Notre but est donc de trouver l'algorithme qui propose le plus petit nombre de camions utilisés.

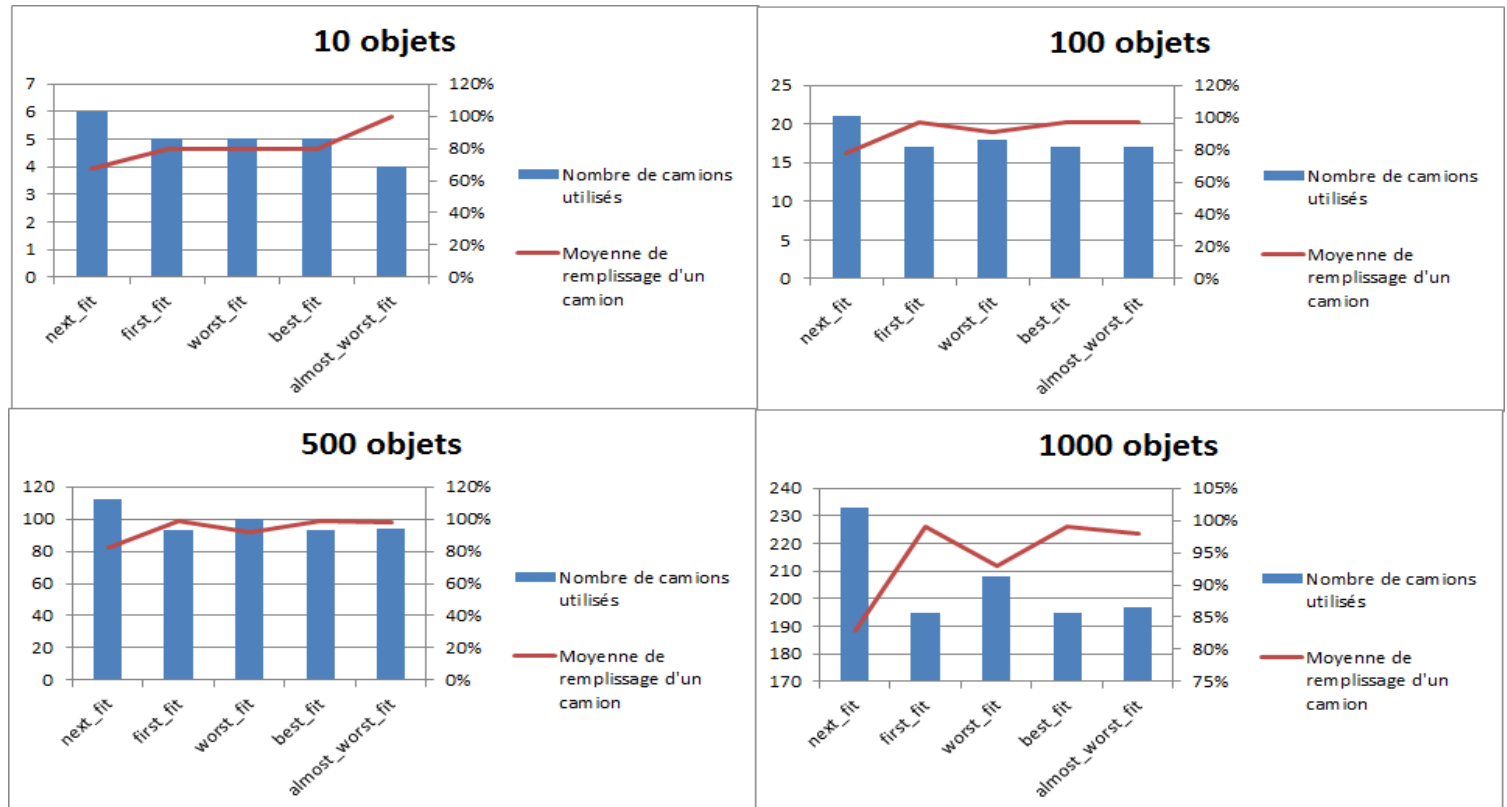
Nous regardons bien sûr le nombre d'accès à chaque camion. Cette métrique permet de se rendre compte du nombre d'actions que chaque algorithme effectue et donc de son efficacité. Cela permet aussi de représenter les déplacements qui seraient nécessaires si l'algorithme était réalisé de manière physique, c'est à dire en essayant de rentrer l'objet physiquement dans chaque camion. Cela nous donne un indice sur la complexité de l'algorithme.

Ensuite, nous avons calculé la moyenne de remplissage des camions. Plus ce pourcentage est élevé, plus les camions sont remplis.

Enfin, nous avons découpé et classé les camions selon leur taux de remplissage. Nous avons donc cinq catégories : les camions pleins, les presque pleins, les bien remplis, les peu remplis et les presque vides. Grâce à ces chiffres, nous pourrions remarquer l'efficacité de chaque algorithme.

## ANALYSE APPROFONDIE DE CHAQUE ALGORITHME

Pour comparer les performances de nos algorithmes, nous avons utilisé quatre exemples. Ceux-ci utilisent un nombre d'objets variés afin d'observer les différentes réactions des algorithmes. Voici les chiffres qui en découlent.

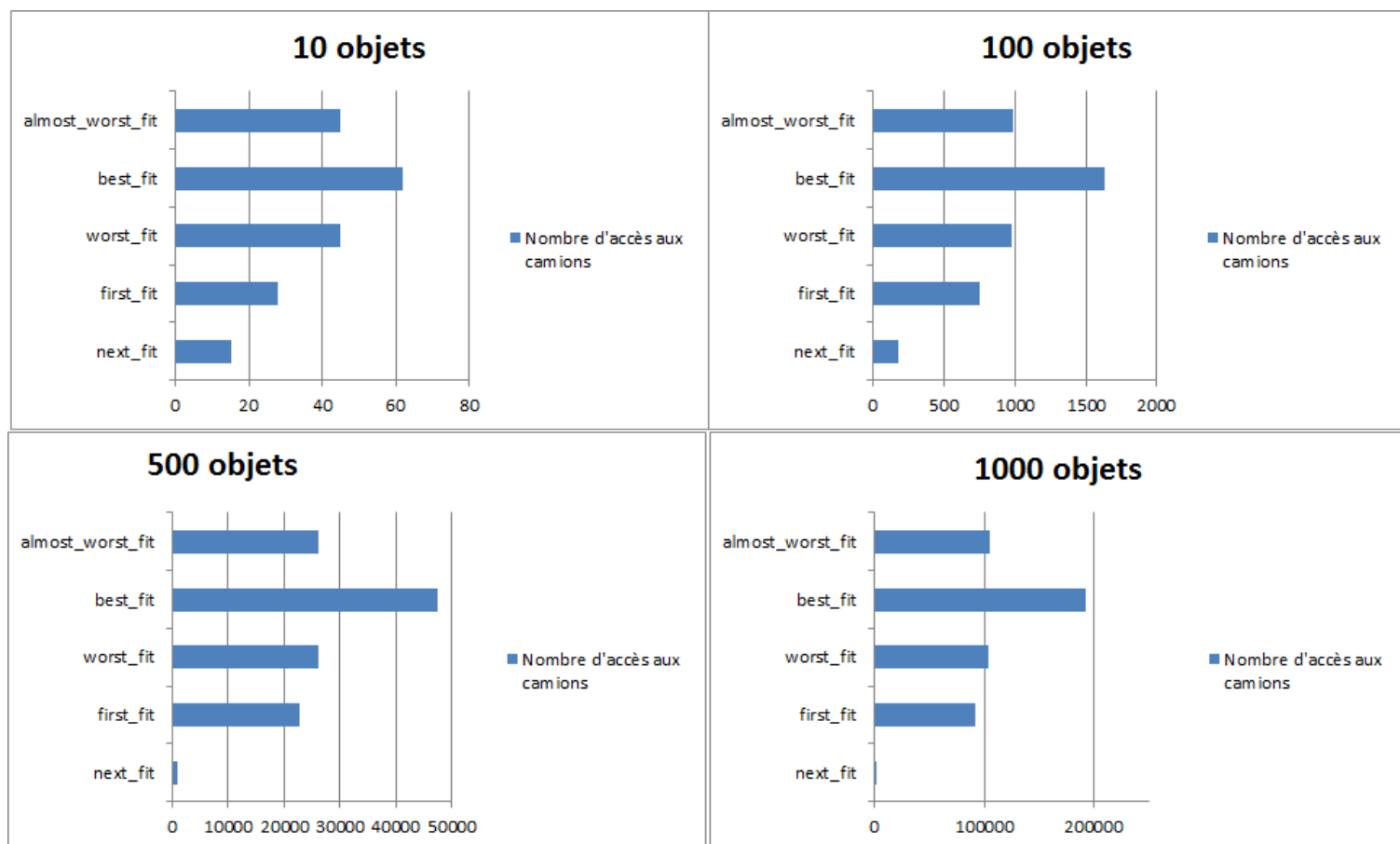


Nous pouvons remarquer grâce à ces graphes que l'algorithme Next Fit est toujours le moins efficace, que ce soit pour un petit nombre d'objets à traiter ou pour un grand nombre.

De plus, les algorithmes First Fit, Best Fit et Almost Worst Fit ont souvent le même nombre de camions utilisés et une moyenne de remplissage très proche les uns des autres. Cependant, on remarque que lorsque le nombre d'objets est faible, l'algorithme Almost Worst Fit est plus efficace de par sa moyenne de remplissage élevée (100% pour 10 objets) lorsque les autres algorithmes sont capés à 80%.

A l'inverse, lorsque le nombre d'objet est grand, on constate que les algorithmes First Fit et Best Fit ont de meilleurs résultats (99% de remplissage pour 500 et 1000 objets contre seulement 98% pour Almost Worst Fit).

Nous notons aussi que plus le nombre d'objets à traiter est grand, plus les algorithmes Next Fit et Worst Fit donnent de mauvais résultats.

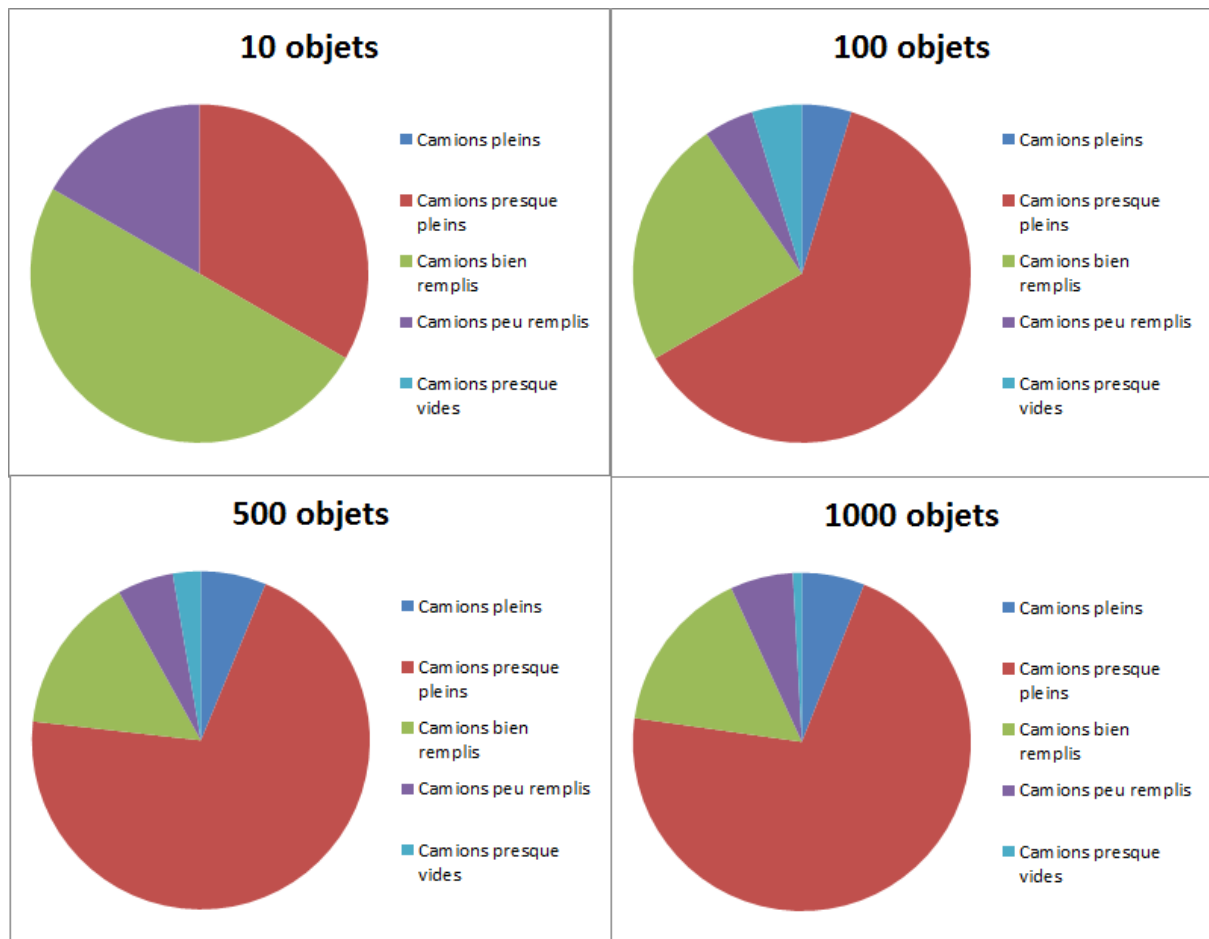


A la vue de ces graphes, nous remarquons instantanément que l'algorithme Next Fit accède très peu aux camions. Alors que Best Fit au contraire en réalise énormément. Le nombre d'accès aux camions peut représenter le temps que va durer le remplissage. Next Fit a beau utiliser beaucoup de camions, il reste une solution rapide pour ce problème.

Nous voyons aussi que les algorithmes Almost Worst Fit et Worst Fit utilisent sensiblement le même nombre d'accès aux camions.



## REEMPLISSAGE DES CAMIONS PAR NEXT FIT :



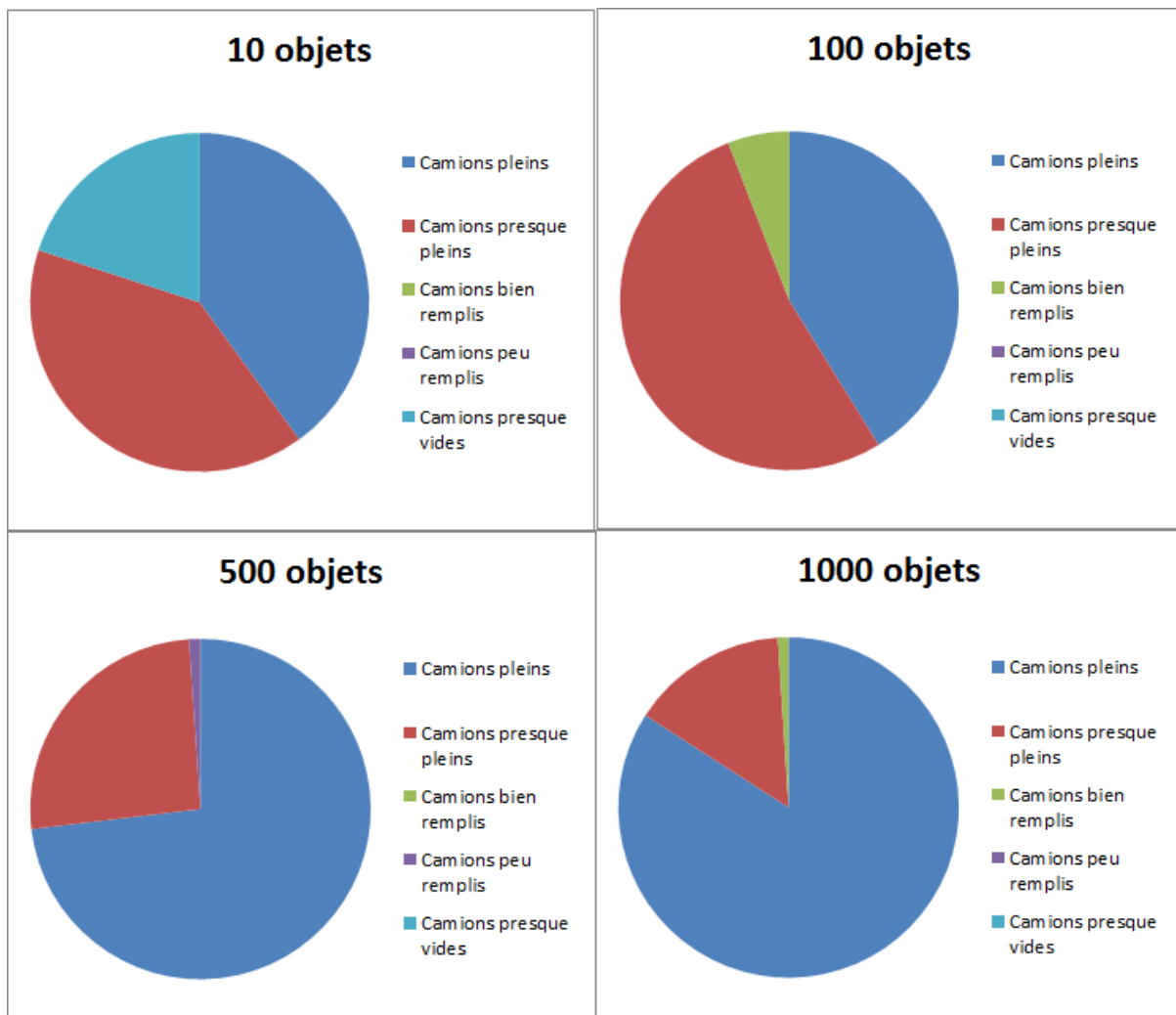
On remarque dans un premier temps que très peu de camions sont pleins. Plus le nombre d'objet augmente, plus les camions sont presque pleins. Pour minimiser les camions peu remplis, on peut réappliquer un algorithme. Par exemple, on peut décider de vider les camions dont le remplissage est inférieur à 40% et recommencer l'algorithme avec ces objets. Ou on peut coupler des camions presque vides avec des camions bien remplis, et passer tous les objets dans un seul. Cette méthode a quand même un inconvénient, car on doit forcément faire un deuxième algorithme.

## REEMPLISSAGE DES CAMIONS PAR WORST FIT :



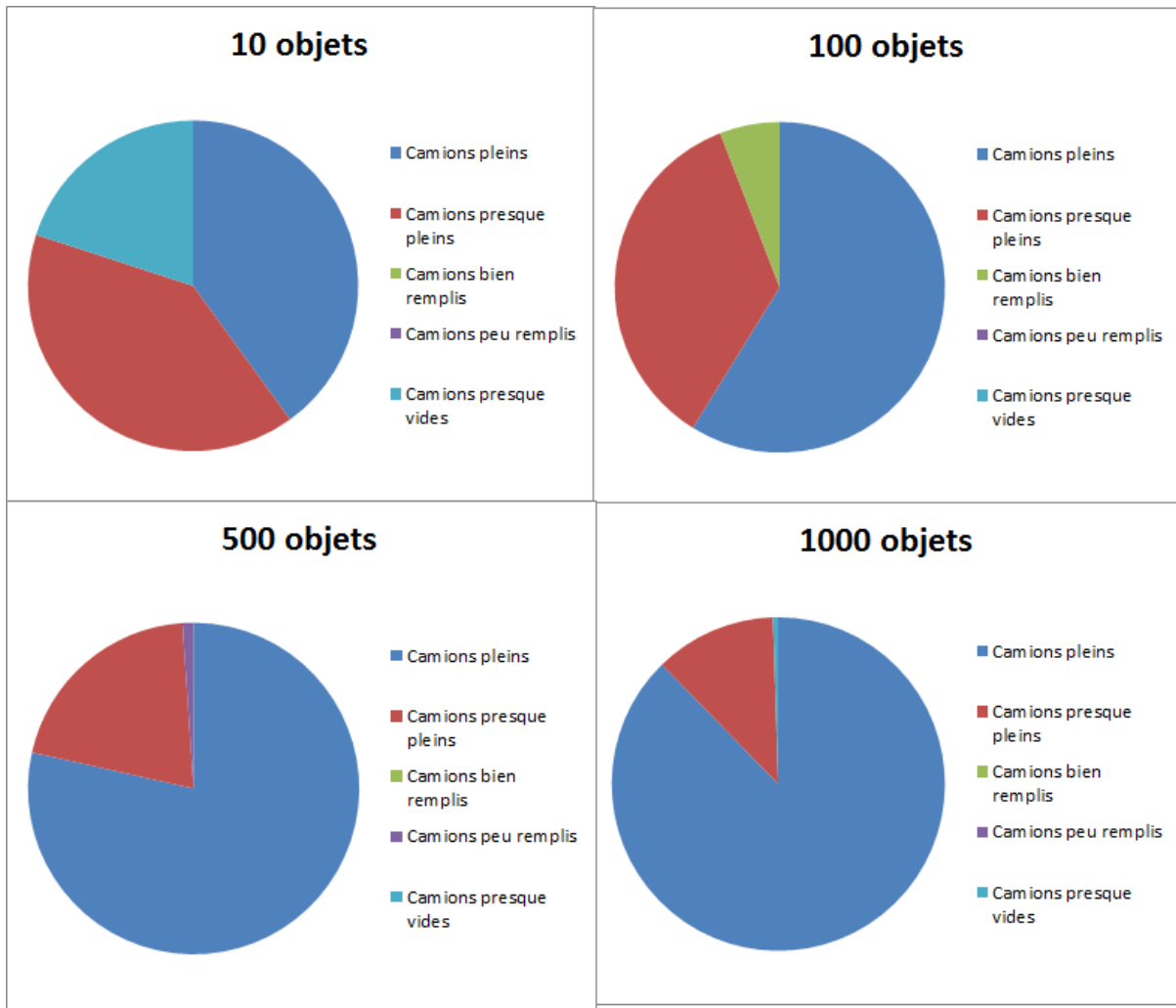
Nous pouvons voir que les solutions proposées par cet algorithme ne comportent aucun camion presque vide ou peu rempli. La grande majorité des camions sont presque pleins. Cette solution peut donc convenir à un client qui souhaite utiliser un minimum de camions sans pour autant les remplir entièrement.

## REEMPLISSAGE DES CAMIONS PAR FIRST FIT :



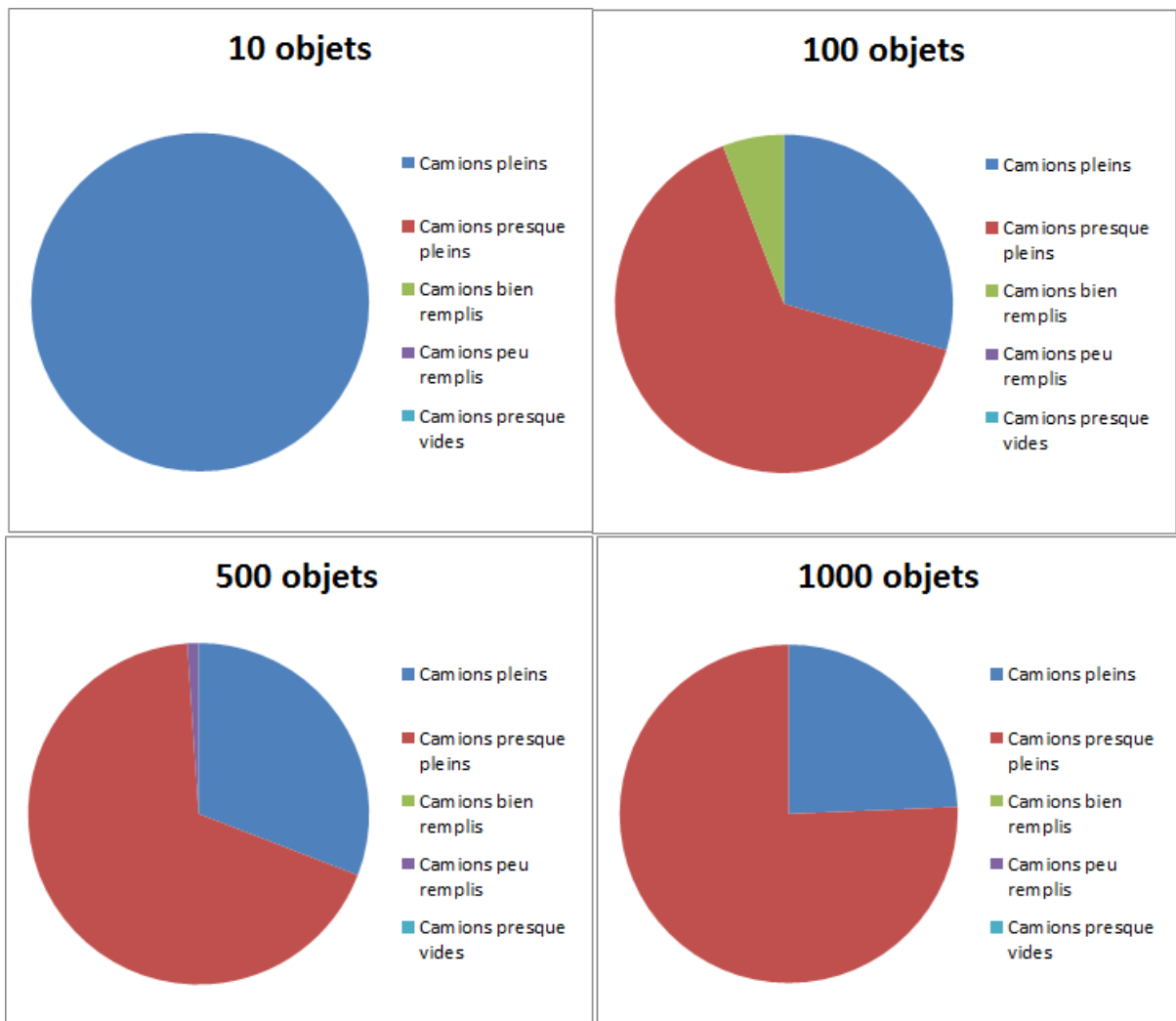
Ici, la majorité des camions sont totalement pleins. On remarque pourtant que pour 10 objets, nous avons 1 camion sur 5 qui est presque vide. Cependant, dans le cas des 10 objets, cela paraît logique, car si les camions ont une taille maximum de 10, et que la somme des objets fait 51, il est normal qu'au moins un camion soit assez vide.

## REPLISSAGE DES CAMIONS PAR BEST FIT :



De manière équivalente à l'algorithme précédent, celui-ci propose une solution comportant majoritairement des camions pleins. On remarque aussi que les résultats sont très proches de l'algorithme précédent pour 10, 500 et 1000 objets.

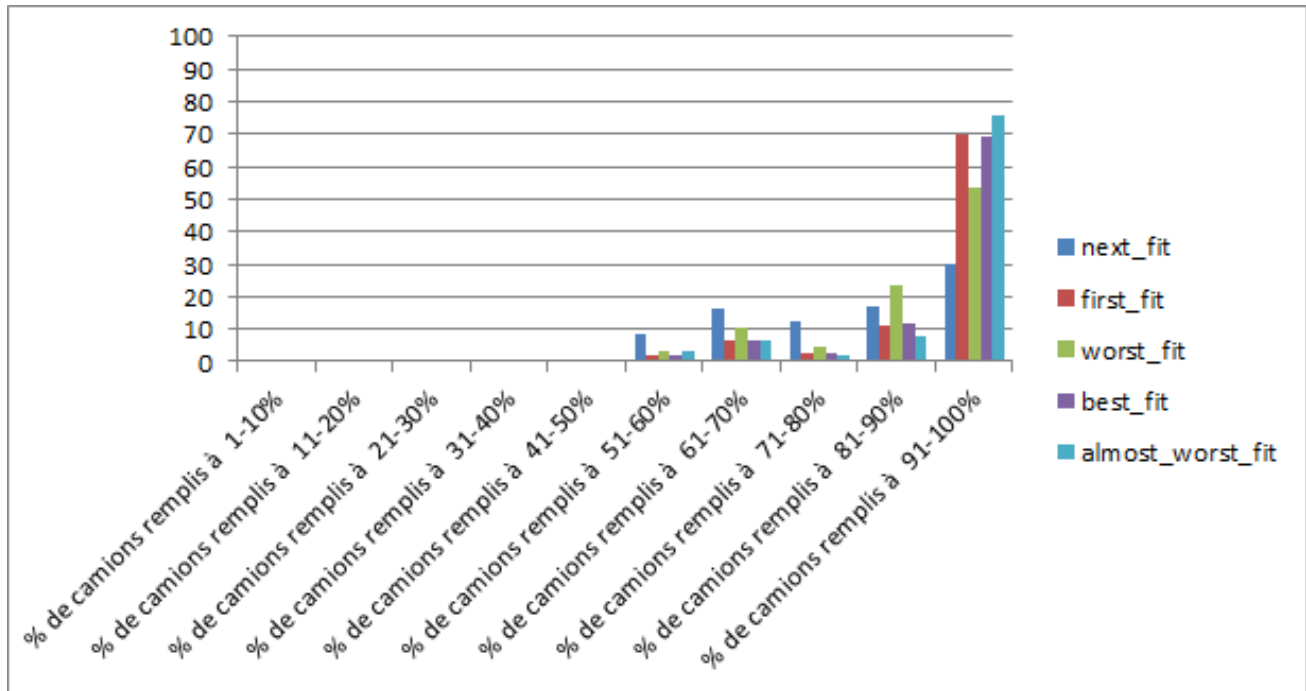
## REEMPLISSAGE DES CAMIONS PAR ALMOST WORST FIT :



Dans un premier temps, nous constatons que le résultat pour 10 objets est parfait. Seulement, cela peut être un coup de chance, on ne peut pas tirer de conclusion hâtive sur un seul exemple avec 10 objets. De plus, il n'y a aucun camion presque vide. Pourtant, on remarque aussi que le nombre de camion presque pleins est grand par rapport aux deux algorithmes précédents.

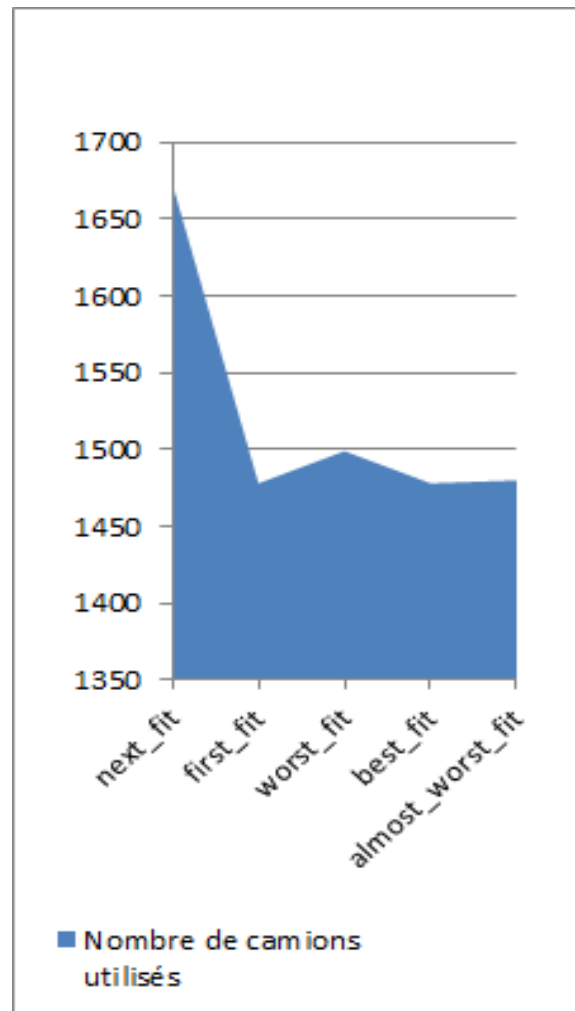
## ANALYSE GLOBALE SUR LA TOTALITÉ DE NOS EXEMPLES.

Afin de fournir des statistiques concrètes, nous avons développé et utilisé un algorithme permettant de consigner et rassembler tous les résultats de nos exemples.



Nous pouvons voir directement que tous les camions sont remplis à plus de 50%. On note aussi que l'algorithme Almost Worst Fit reste celui qui propose le plus de camions totalement pleins.

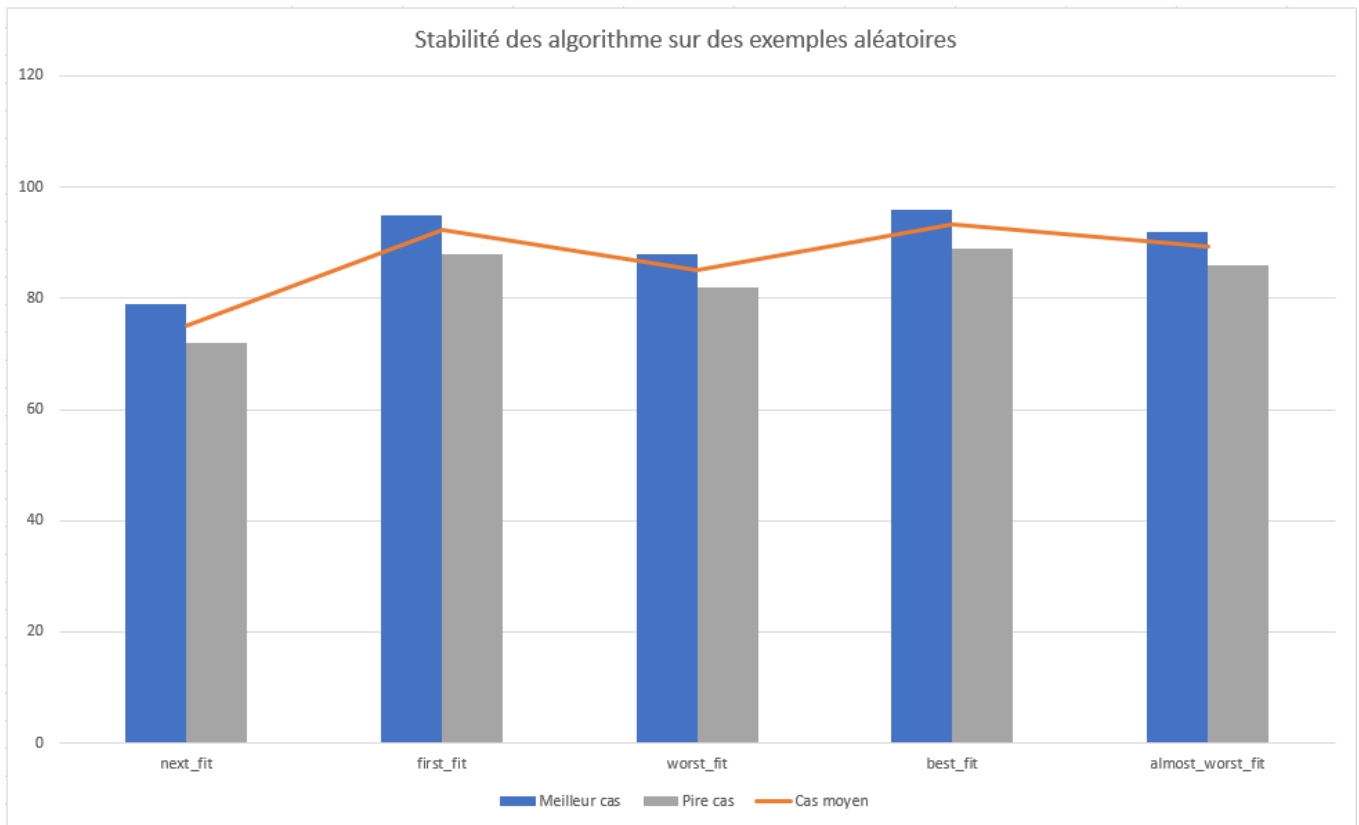
Nous voyons ici le nombre cumulé de camions utilisés par les différents algorithmes sur un ensemble d'exemples. On note que First Fit et Best Fit utilisent le moins de camions (1478) contre 1480 pour Almost Worst Fit.



## STABILITÉ DES ALGORITHMES

Les problèmes de stabilité sont des problèmes récurrents sur les algorithmes. Un exemple flagrant est le *Bubble sort* qui est excellent quand la liste est quasiment triée, et relativement mauvais (par rapport au Quick sort par exemple) avec une liste à l'envers.

Nous allons donc comparer le pire des cas et le cas moyen de ces algorithmes. Nous avons lancé une série de 1000 simulation aléatoire avec des camions de taille 100 et 200 objets compris entre 1 et 100, dont les résultats sont les suivants:



On observe donc que sur des exemples aléatoires que les algorithmes sont relativement stables, car tous les cas sont compris dans un intervalle de taille 10% centrés sur le cas moyen. Il y a donc peu de chance pour qu'un algorithme devienne catastrophique (voir plus bas pour un cas particulier).



## INTERPRÉTATION

Grâce à ces graphes, nous comprenons clairement que, ces algorithmes sont totalement différents. Dans un premier temps, les résultats de Next Fit et de Worst Fit ne sont pas comparables aux trois autres algorithmes. Bien que Next Fit n'accède que très peu aux camions, les résultats qui en découlent sont décevants. En ce qui concerne Worst Fit, nous pouvons en dire que ses résultats sont beaucoup trop éloignés d'une potentielle solution "parfaite" (s'il en existe une bien sûr), et que son nombre d'accès aux camions est bien trop élevé pour le résultat.

Ensuite, nous pouvons dire que les algorithmes First Fit et Best Fit sont très similaires de par leurs résultats. Ils ont tous les deux sensiblement le même nombre de camions utilisés et un remplissage moyen qui tend vers 100% en augmentant le nombre d'objets. Cependant, le nombre d'accès aux camions de Best Fit est bien supérieur à celui de First Fit. Les deux algorithmes sont donc similaires en résultat, mais First Fit est bien plus efficient.

En ce qui concerne Almost Worst Fit, nous pouvons dire que cet algorithme est intéressant parce que son taux de remplissage de camion est meilleur que les autres algorithmes. Pourtant, First Fit est sensiblement plus efficace en terme de camion utilisés.

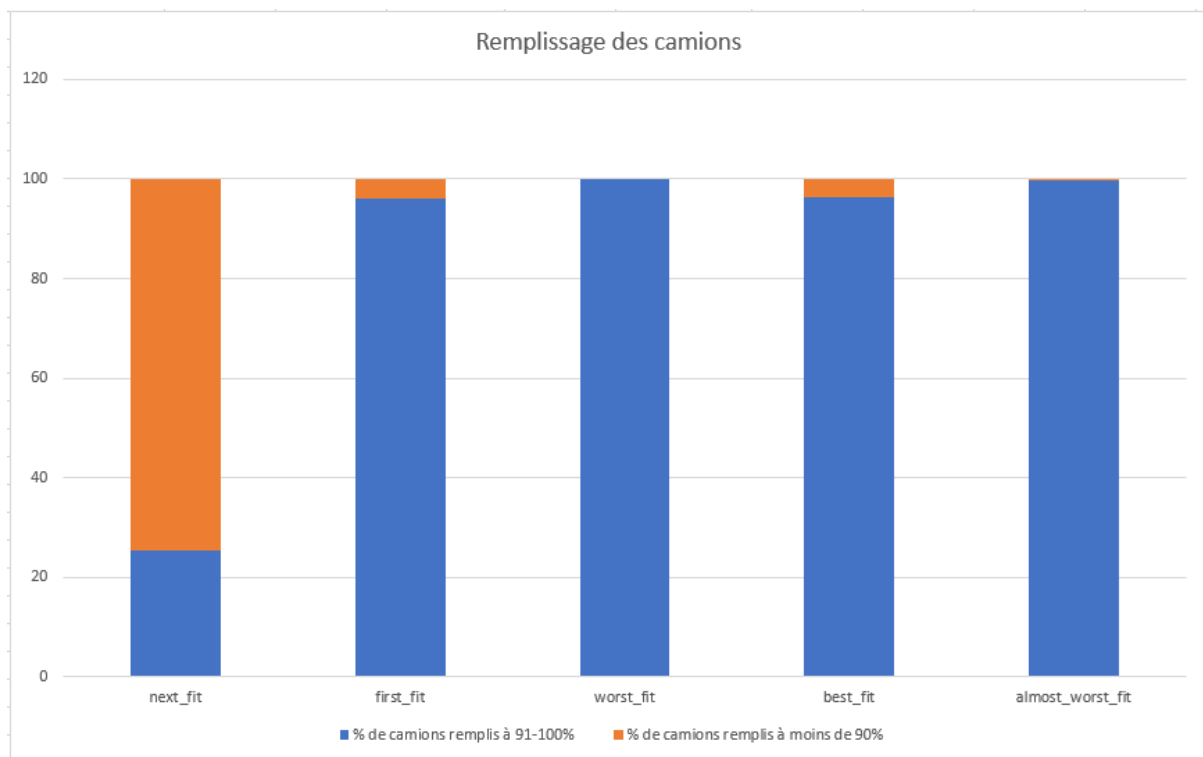
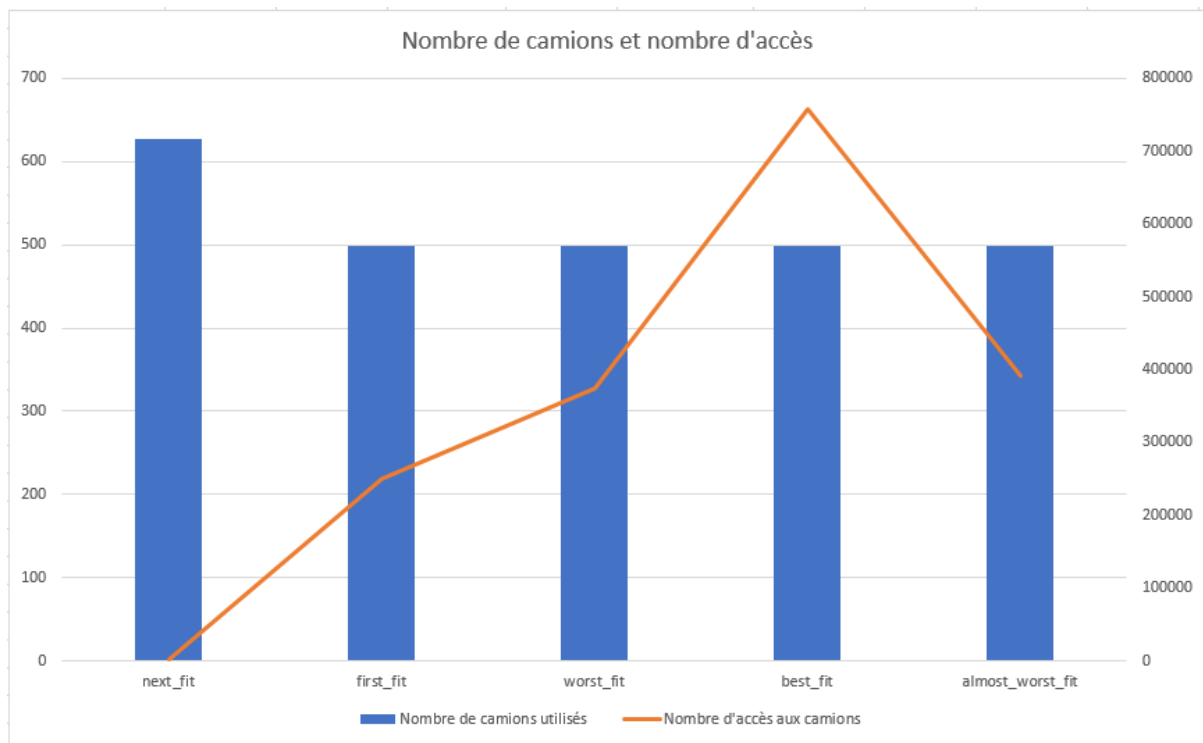
## GÉNÉRALISATION

Si nous suivons aveuglément notre interprétation, il est clair que First Fit serait le candidat idéal. Cependant, lorsqu'il est question de cas réels, il est rare d'avoir un nombre d'objets totalement aléatoire avec un ordre sans logique apparente.

Dans le cas du problème du Bin Packing, l'ordre d'arrivée des données est primordial.

### UN CAS RÉEL : LES PLUS GROS POIDS ARRIVENT EN PREMIER

Quand on doit résoudre un tel cas dans la vie réelle, on peut décider de tirer le meilleur parti possible des algorithmes. En effet, les objets n'arrivent pas dans un ordre aléatoire. Il semble logique de placer les plus gros en premier. Voyons comment se comportent les algorithmes dans ce cas particulier.



Dans ce cas plus proche de la vie réelle, on voit bien que deux algorithmes sortent du lot: le Worst Fit et le Almost Worst Fit. Aussi, dans ce cas, l'algorithme First Fit revient à faire un Best Fit, car vu que les objets arrivent de manière décroissante, les camions sont triés de manière décroissante (donc le premier qui rentre est le meilleur).

Pourtant, dans le cas contraire, lorsque les objets les plus petits arrivent en premier, tous les résultats sont quasiment identiques.

## RECOMMANDATION

Faire une recommandation n'est pas aisé. En effet, il n'y a pas d'algorithme magique. Il faudra donc choisir le bon algorithme en fonction du problème.

Dans le cas où les données sont massives et non triées, l'algorithme First Fit propose les meilleurs résultats pour une consommation de ressource très faible par rapport aux autres. S'il n'y a pas beaucoup d'objets à traiter, alors on sera tenté d'utiliser l'algorithme Almost Worst Fit. Mais si les données sont triées par ordre décroissant, les algorithmes Almost Worst Fit et Worst Fit sont tout indiqués.

Bien que ces algorithmes résolvent le problème posé (certains mieux que d'autres), l'utilisation de ces algorithmes seuls n'est peut-être pas tout à fait suffisante. En effet, après avoir utilisé un algorithme une première fois, il est probable que certains camions ne soient pas entièrement pleins. Les objets qui sont dans ces camions devraient être retraités par un algorithme de nouveau afin de trouver une solution encore meilleure.