Séance 2: Application d'un algorithme approché pour résoudre un problème d'optimisation

Sabrine Mokhtar SAE2

Problème Bin-Packing

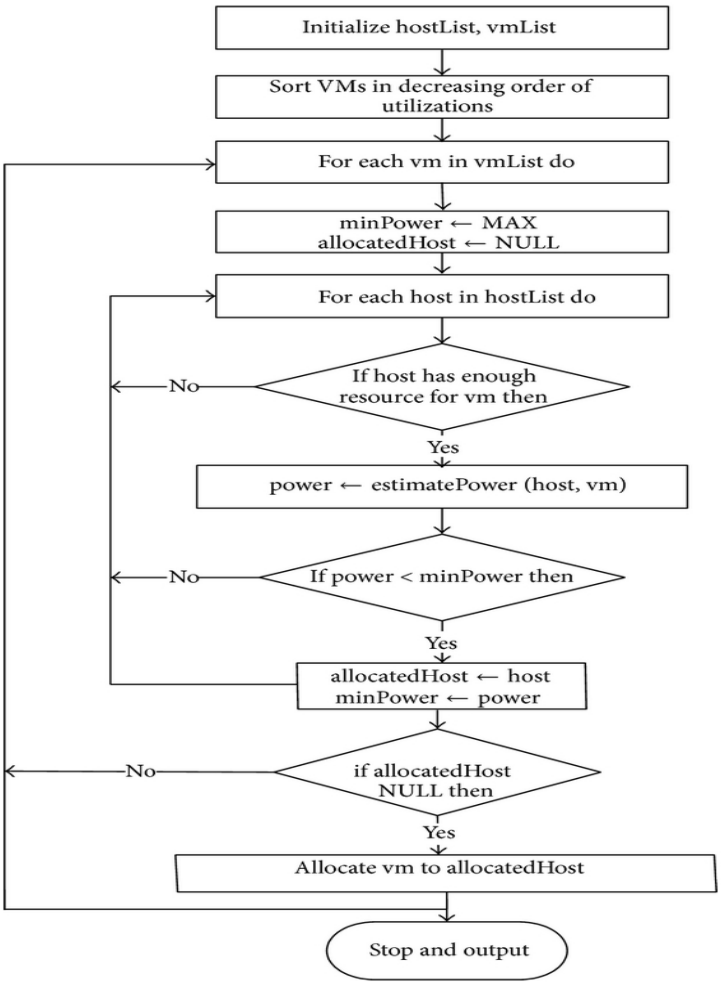
Best Fit

Question 1 : Donner le pseudo-code de l'algorithme BestFit.

L'algorithme BestFit consiste à allouer la plus petite partition libre qui répond aux exigences du processus de demande.

Dans notre problème de Bin Packing la stratégie best-fit garde les bins toujours ouverts. Cependant ,le choix du bin dans lequel l'objet i en cours va être placé dépend des valeurs des gaps (espaces libres) présentes dans les bins.Ainsi , i sera placé dans le bin de plus petit gap pouvant le contenir .

Pseudo code : une représentation purement intellectuelle



```
procedure OptimizeBandwidth(floorBursts, floatingBursts)
sortByDuration(floorBursts);
sortByDuration(floatingBursts);
solution = new Solution();
solution.packRectangle(floorBursts[0]);
for i=1,…,floorBursts.Length()-1 do
  level = searchLevelsWithEnoughSpace();
  if level then
    solution.packRectangle(floorBursts [i], level);
  else
    newLevel = solution.createLevelAboveTopMostLevel();
    solution.packRectangle(floorBursts [i], newLevel);
  endif
endforeach
for i=1,…,floatingBursts.Length()-1 do
  level = searchLevelsWithEnoughSpace();
  if level then
    solution.packRectangle(floatingBursts [i], level);
  else
    newLevel = solution.createLevelAboveTopMostLevel();
    solution.packRectangle(floatingBursts [i], newLevel);
  endif
endforeach

return solution;
end
```

Question 2 : **Expliquer l'utilité des différents paramètre de l'algorithme**.

On a comme résultat des objets ordonnés dans les bins dans la meilleure situation avec un nombre minimale de bin

Question 3 : **Comment peut-on appliquer chaque algorithme approché pour résoudre votre problème d'optimisation**

1ère étape : trier les objets à ranger dans l'ordre décroissant de hauteur c'est-à-dire les ordonner par taille (de grande taille au petite taille)

2ème étape : placer l'objet qui a le plus grande taille dans le 1er bin

3ème étape : passant en 2ème objet dans la liste triée, on test si l'espace restant dans le bin est supérieur au taille de 2ème objet :

-si espace restant > taille 2ème objet => on place le 2ème objet dans le 1er bin

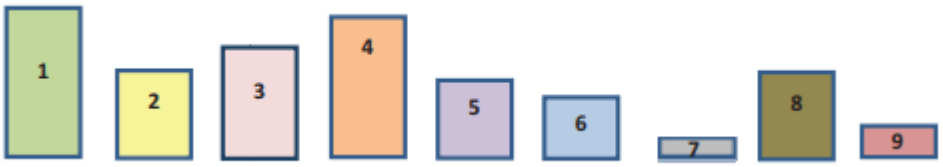
-sinon => on crée un 2ème bin et on place le 2ème objet dans le 2ème bin

4ème étape : on fait le même test pour le reste des objets.

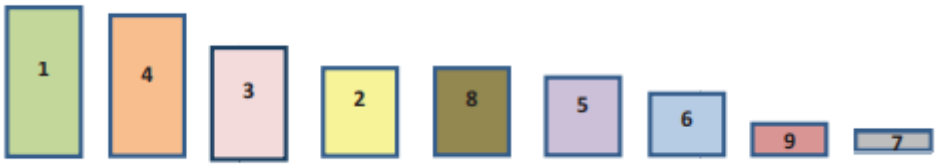
Représentation graphique :

1ère étape :

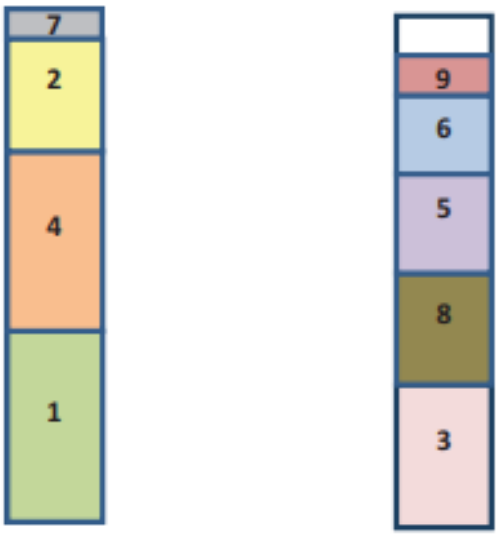
Etat initial



Après le trie :



2ème étape / 3ème étape / 4ème étape : phases de comparaison et placement des objets pour obtenir la solution finale de l'algorithme



1er bin **2ème bin**