



Universidad Simón Bolívar

Diseño de Algoritmos I

Proyecto II

Lorenzo Fundaró - 0639559

Germán Jaber - 0639749

22 de mayo de 2010

Índice

1. Breve descripción del problema

En teoría de complejidad computacional, el “Bin Packing Problem” es un problema combinatorio de dificultad NP-HARD. Consiste en tomar objetos de volúmenes distintos y empacarlos en un número finito de contenedores de capacidad V de tal manera que se minimice el número de contenedores usados. Existen muchas variaciones a este problema, como el de 2-dimensiones, empaquetamiento lineal, por peso, por costo, etc. Estos tienen muchas aplicaciones, como por ejemplo llenar containers con fines de exportación, llenar camiones con una capacidad determinada de peso, crear respaldos de archivos en medios portátiles (pendrives, disco duro portátil, etc).

1.1. Especificación formal

Dado un contenedor de capacidad V y una lista a_1, \dots, a_n de elementos de capacidad variable, se busca encontrar un entero B junto con una B -partición $S_1 \cup S_B$ de $\{1, \dots, n\}$ tal que $\sum_{i \in S_k} a_i \leq V$ para todo $k = 1, \dots, B$. Una solución es óptima si posee un B mínimo.

2. Descripción de las heurísticas empleadas

2.1. Representación de la solución

Una solución puede verse como un vector de elementos empacados en un contenedor determinado. Cada elemento indica sus coordenadas en el contenedor al que pertenece, el número de contenedor donde está y las características respectivas al ítem como su ancho y alto.

2.2. Función Objetivo

La función objetivo consiste en comparar si luego de aplicar el operador de vecindad se ha reducido el número de contenedores utilizados. Si esto sucede se procede a utilizar la nueva solución como la mejor obtenida hasta el momento.

2.3. Operadores

El operador que permite moverse de una solución a otra obedece a un principio que permite alcanzar una solución óptima. Este principio consiste en elegir un contenedor objetivo (Target Bin) el cuál tenga la mayor posibilidad de ser vaciado por sucesivas extracciones de sus elementos, esta posibilidad se determina a través de la función: [1]

$$\varphi(S_i) = \alpha \frac{\sum_{j \in S_i} w_j h_j}{WH} - \frac{|S_i|}{n}$$

donde S_i denota el conjunto de el conjunto de ítems que están actualmente empacados en el bin i y α es un valor predefinido de peso positivo.

Una vez se determina el Target Bin se procede a extraer un elemento del mismo, y junto con los elementos del resto de los contenedores (sin incluir al Target Bin) se trata de empacar dichos elementos. Luego la función objetivo se utilizará para decidir si cambiamos si cambiamos de solución o no.

3. Pseudocódigos

3.1. Heurística Finite Best Strip

Este algoritmo consiste en ir poniendo objetos en la zona inferior izquierda del contenedor. Se comienza con el primero y se prosigue con el resto hasta agotar el ancho del contenedor. Al agotar el ancho del bin hemos conseguido lo que se denomina un "strip" o también conocido como una capa y se procede a crear un nuevo strip para colocar el resto de los elementos. Cuando se han colocado todos los elementos se procede entonces a tratar de colocar strips encima de otros siempre que quepan en un bin. El algoritmo funciona como sigue:

```
for i in 1,2,3,4: print i
```