Diseño y Análisis de Algoritmos
Universidad de La Habana
Curso 2023 - 2024
Naomi Lahera Champagne C411

Bin Packing

Dado un conjunto de n elementos, cada uno con un peso w_i, y un conjunto de contenedores B, cada uno con una capacidad fija c, encuentra el número mínimo de contenedores necesarios para empacar todos los elementos.

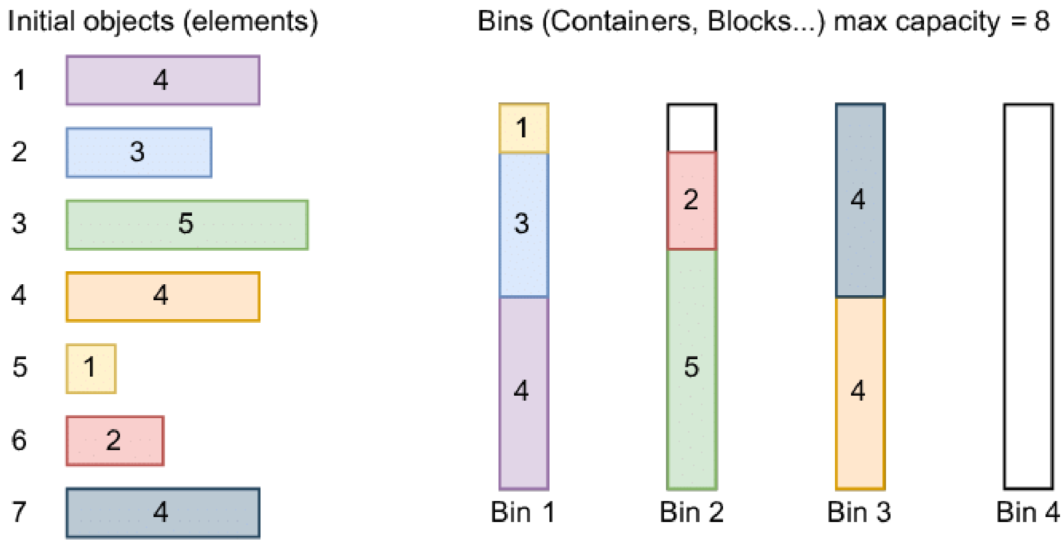

Figure 1: Posible solución a instancia de Bin Packing con $c = 8$ y $W = [4, 3, 5, 4, 1, 2, 4]$. Fuente: *An efficient cloudlet scheduling via bin packing in cloud computing. Amine Chraibi, Said Ben Alla, Abdellah Ezzati. 2022*

Introducción

En el presente documento haremos un análisis del problema Bin Packing. Dicho problema tiene como objetivo encontrar el mínimo valor que cumpla con una condición dada, lo que nos lleva a pensar que estamos tratando un problema de optimización. Cuando minimizamos en un espacio de búsqueda finito una vía efectiva es evaluar cada posible solución y quedarnos con la 'mejor', así estaríamos entonces frente a un problema de optimización combinatoria siendo $O(n!)$ la complejidad de la solución anterior. Esto nos conduce a las siguientes interrogantes, las cuales serán resultas en el este documento :

1. Estará Bin Packin en el conjunto de problemas cuya solución puede darse en tiempo polinomial?

2. Podremos comprobar en tiempo polinomial si que una posible solución es la solución correcta?

3. Podremos construir un algoritmo de aproximación para enfrentar este problema?

4. Existen algunos casos para los cuales hay solución polinomial?

A continuación se dará respuesta a las interrogantes planteadas.

Secciones

Introducción 1

Bin Packing es un problema NP? 4

Redefinición del problema 4

Bin Packin. NP-Duro 4

Partition $\leq_p$ Bin Packing 4

Subset-Sum $\leq_p$ Partition 6

Conclusión 7

Algoritmo de aproximación 8

Bin Packing es un problema NP?

Redefinición del problema

Primero, necesitamos reformular el problema de empaquetado en bins como un problema de decisión. Definamos *BIN-PACK* de la siguiente manera:

$$\textit{BIN-PACK}$$

$= \{\langle W, c, b\rangle|$ existe un mínimo de b subconjuntos disjuntos S_x que cumplen $S_1 \cup S_2 \cup \cdots \cup S_b = W$, y para cada $\langle i, w_i\rangle \in S_x, \sum w_i \leq c$ $\}$.

Supongamos que tenemos una solución a una instancia del problema y queremos comprobar su correctitud. Tendremos que comprobar entonces :

1. $S_1 \cup S_2 \cup \cdots \cup S_b = W$
2. $\forall S_x$ se cumple $\sum\limits_{w_i \in S_x} w_i \leq c$
3. No existe forma de empaquetar los objetos en $b - 1$ contenedores.

Las comprobaciones 1 y 2 pueden hacerse en tiempo polinimial, sin embargo la número 3 requiere la solución de una instancia distinta del problema ($\langle W, c, b-1\rangle$). Entonces Bin Packing solo puede ser NP si existe un algoritmo que en tiempo polinomial nos de una solución correcta. Más adelante le daremos respuesta a nuestra interrogante.

Bin Packin. NP-Duro

En esta sección demostraremos que Bin Packing es un problema NP-Duro haciendo una reducción de Subset-Sum a Bin-Packing, pero para hacer la demostración más potable haremos primero una reducción de Subset-Sum a Partition y luego otra de Partition a Bin Packing.

Partition $\leq_p$ Bin Packing

Pasemos a definir el problema *PARTITION*. Sea A un conjunto de números racionales tenemos que:

$$\text{PARTICIÓN} = \left\{ A \;\middle|\; \exists\ B,\ C \subseteq A,\ B \cap C = \emptyset,\ B \cup C = A,\ \sum_{b \in B} b = \sum_{c \in C} c \right\}$$

Conatruyamos entonces una instancia de Bin Packing a partir de una instancia de PARTITION. Enumerando los elementos de A creamos las tuplas $\langle$ i, a_i $\rangle$ y definimos:

- $W = \langle 1, a_1\rangle, \langle 2, a_2\rangle, \ldots, \langle k, a_k\rangle$
- $c = \left\lfloor \frac{\sum\limits_{a_i \in A} a_i}{2} \right\rfloor$. Definiendo $n = \frac{\sum\limits_{a_i \in A} a_i}{2}$ tenemos $c = \lfloor \frac{n}{2} \rfloor$
- $b = 2$

Tenemos ahora una instancia de Bin Packin a partir de una instancia de PARTITION con una construcción en tiempo polinimial ($O(n)$). Dicha instancia cumple $\langle A\rangle \in PARTICION \Rightarrow \langle W, c, b\rangle \in$ BIN-PACK.

Proof. Si $\langle A\rangle \in$ PARTITION entonces existe una forma de particionar A en dos conjuntos disjuntos B y C que cumplen $\sum\limits_{b \in B} b = \sum\limits_{c \in C} c$. Sea $S_1 = \{\langle i, a_i\rangle | a_i in B\}$ y $S_2 = \{\langle i, a_i\rangle | a_i in C\}$, tenemos que :

1. Como cada elemento de A esta en B o C y cada elemento de B o C tiene un único elemento asociado en W se cumple $S_1 \cup S_2 = W$.
2. Como $\sum\limits_{b \in B} b = \sum\limits_{cinC} c = \frac{n}{2}$ por ser $\langle A\rangle unainstanciadePARTITIONsecumpleque \sum\limits_{\langle i, w_1 i\rangle \in S_1} w_1 i = \sum\limits_{\langle i, w_{2i}\rangle in S_2} w_{2i} = \frac{n}{2} = c$. Luego tenemos los elementos de W pueden ser empacados en contenedores de tamaño c.
3. Como la suma de cada uno de los conjuntos $S_1 y S_2$ es exactamente c podemos afirmar que no es posible empacar los elementos de W en menos 2 contenedores de tamaño c y por ende 2 es el mínimo.

Entonces podemos concluir que $\langle A\rangle \in PARTICION \Rightarrow \langle W, c, b\rangle \in$ BIN-PACK. □

Se cumple también que $\langle A\rangle \notin PARTICION \Rightarrow \langle W, c, b\rangle \notin$ BIN-PACK.

Proof. Si $\langle A\rangle \notin PARTITION$ entonces no existe forma de particionar a A en dos conjuntos B y C que cumplan $\sum\limits_{b \in B} b = \sum\limits_{cinC} c = \frac{n}{2}$. Luego $\forall X \subseteq A$ se cumple que $\sum\limits_{x \in X} x < \frac{n}{2}$ ó $\sum\limits_{x \in X} x > \frac{n}{2}$. Entonces para empaquetar los elementos correspondientes en W tendremos que coger siempre una patición formada por dos conjuntos donde uno ellos cumple que la suma de sus elementos es estrictricamente menor que $\frac{n}{2}$ y por tanto menor estrictamente que c. La suma de los elementos del conjunto resrante es entonces estrictamente mayor que $\frac{n}{2}$ y por tanto estrictamente mayor que c siendo imposible empauqetar sus elementos en un contenedor de tamanno c. Podemos concluir que el $\langle W, b, c\rangle$ correspondiente no pertenece a Bin Packing. □

Tenemos entonces que:

1. $\langle A\rangle \notin PARTICION \Rightarrow \langle W, c, b\rangle \notin$ BIN-PACK.
2. $\langle A\rangle \in PARTICION \Rightarrow \langle W, c, b\rangle \in$ BIN-PACK.

Queda entonces demostrado Partition $\leq_p$ Bin Packing

Subset-Sum $\leq_p$ Partition

Definamos SUBSET-SUM. Sea S un conjunto de números y t un entero decimos que:

$$\text{SUBSET-SUM} = \{\langle S, t\rangle \mid \exists\ X \subseteq\ S, \sum_{x \in X} x = t\}$$

Lema 1. *Si $\langle S, t\rangle \in PARTITION$ entonces siendo $n = \sum_{s_i \in S} s_i$ podemos decir que $\langle S, n - t\rangle \in PARTITION$.*

Proof. Como $n = \sum_{s_i \in S} s_i$ si $\langle S, t\rangle \in$ PARTITION entonces se cumple que $\exists X | X S y X cumple \sum_{x \in X} x = t$ luego existe entonces $Y = S \setminus X$ que cumple $\sum_{y \in Y} x = n - t$. Podemos afirmar entonces que $\langle S, n - t\rangle \in$ PARTITION. $\square$

Siendo $n = \sum_{s_i \in S} s_i$, $\langle S, t\rangle$ de ahora en adelante asumamos que $\frac{n}{2} \leq t$ si $t \leq \frac{n}{2}$ entonces resolviendo $\langle S, n - t\rangle con \frac{n}{2} \leq n - t$ habremos resulto también $\langle S, t\rangle$.

Construyamos entonces una instancia de PARTITION en función de un ainstancia de SUBSET-SUM. Para esto definamos $A = S \cup \{\ 2t - n\}$. Esta cinstrucción se puede hacer en tiempo polinomial (O(n)).

Demostremos que si $\langle S, t\rangle \in$ SUBSET-SUM $\Rightarrow \langle A\rangle \in$ PARTITION.

Proof. Si $\langle S, t\rangle \in$ SUBSET-SUM entonces $\exists X \subseteq S$ que cumple $\sum_{x \in X} x = t$, entonces existe también $Y = S \setminus X$ que cumple $\sum_{y \in Y} y = n - t \leq t$ (pues $\frac{n}{2} \leq t$). Sean $B = X$ y $C = Y \cup \{2t - n\}$ tenemos entonces que $\exists\ B,\ C \subseteq A$, que cumplen:

1. $B \cap C = \emptyset,\ B \cup C = A$
2. $\sum_{b \in B} b = t$
3. $\sum_{c \in C} c = (\sum_{c \in C} c) + 2t - n = n - t + 2t - n = t$
4. $\sum_{b \in B} b = \sum_{c \in C} c$

Podemos concluir entonces que si $\langle S, t\rangle \in$ SUBSET-SUM $\Rightarrow \langle A\rangle \in$ PARTITION. $\square$

Demostremos ahora que si $\langle S, t\rangle \notin$ SUBSET-SUM $\Rightarrow \langle A\rangle \notin$ PARTITION.

Proof. Si $\langle S, t\rangle \notin$ SUBSET-SUM entonces $\neg\exists X \subseteq S$ que cumpla $\sum_{x \in X} x = t$. Como $A = S \cup \{\ 2t - n\}$ con $n = \sum_{s_i \in S} s_i$ se cumple que $\sum_{a \in A} a = 2t$. Para que $\langle A\rangle \in$ PARTITION tendría que existir entonces $B, C \subseteq A$ tales que $\sum_{b \in B} b = \sum_{c \in C} c = t$ pero esto no es posible porque no podemos separar $2t - n$ en dos números para satisfacer ambos conjuntos, B y C. Luego $\neg\exists B, C | B \cap C = \emptyset, B \cup C = A$ y $\sum_{b \in B} b = \sum_{c \in C} c$.

Podemos concluir entonces que si $\langle S, t\rangle \notin$ SUBSET-SUM $\Rightarrow \langle A\rangle \notin$ PARTITION. $\square$

Hasta el momento tenemos que:

1. si $\langle S, t \rangle \in$ SUBSET-SUM $\Rightarrow \langle A \rangle \in$ PARTITION.
2. si $\langle S, t \rangle \notin$ SUBSET-SUM $\Rightarrow \langle A \rangle \notin$ PARTITION.

Podemos afirmar que Subset-Sum $\leq_p$ Partition.

Hemos demostrados entonces que:

1. Subset-Sum $\leq_p$ Partition
2. Partition $\leq_p$ Bin Packing
3. Subset-Sum $\leq_p$ Partition $\leq_p$ Bin Packing

Conclusión

Estamos ahora en condiciones de responer la primera interrogante que nos planteamos. Estará Bin Packin en el conjunto de problemas cuya solución puede darse en tiempo polinomial? Demostramos que no existe en estos momentos un algoritmo que resuelva Bin Packing en tiempo polinomial, entonces no existe tampoco un algortimo que en tiempo polinomial determine la correctitud de una solución a este problema. Podemos afirmar que Bin Packing no es NP. Al no ser un problema de decisión tampoco es NP-Completo lo que nos lleva a la conclusión de que es NP-Duro.

Algoritmo de aproximación

This homework answers the problem set sequentially.

1. *Download the US Presidential Elections data set* `uspresidentialelections.dta` *from the course ILIAS site. Load the data set in* `R`.

   ```
   Copy your R Code to answer the question here.
   ```

2. *Describe the dataset. What variables does it contain? How many observations are there? What time span does it cover?*

   Please type your answer here.

   ```
   Put the right R command here.
   ```

3. *Compute measures of central tendency and variability of the variables* `vote` *and* `growth` *using* `R`. *Use the numerical measures of central tendency and variability discussed in class. Describe them in your own words and make a nice table. Plot the distribution of both variables using a boxplot and histogram. Make sure to make your plots as nice-looking as possible. Especially, include a title and label the axes.*

   Your answer goes here

   ```
   3 R commands here
   ```

   Potentially, your answer continues here.

   ```
   4 R commands here
   ```

   And more of your answer here.

   ```
   And more space for your R commands.

   # This is the code to produce the first boxplot.
   pdf(file = "box1.pdf")
   boxplot(us_data$vote, horizontal = T,
           main  = "A Boxplot of the Variable Vote",
           names = "Vote",
           xlab  = "Range of Votes")
   dev.off()
   ```

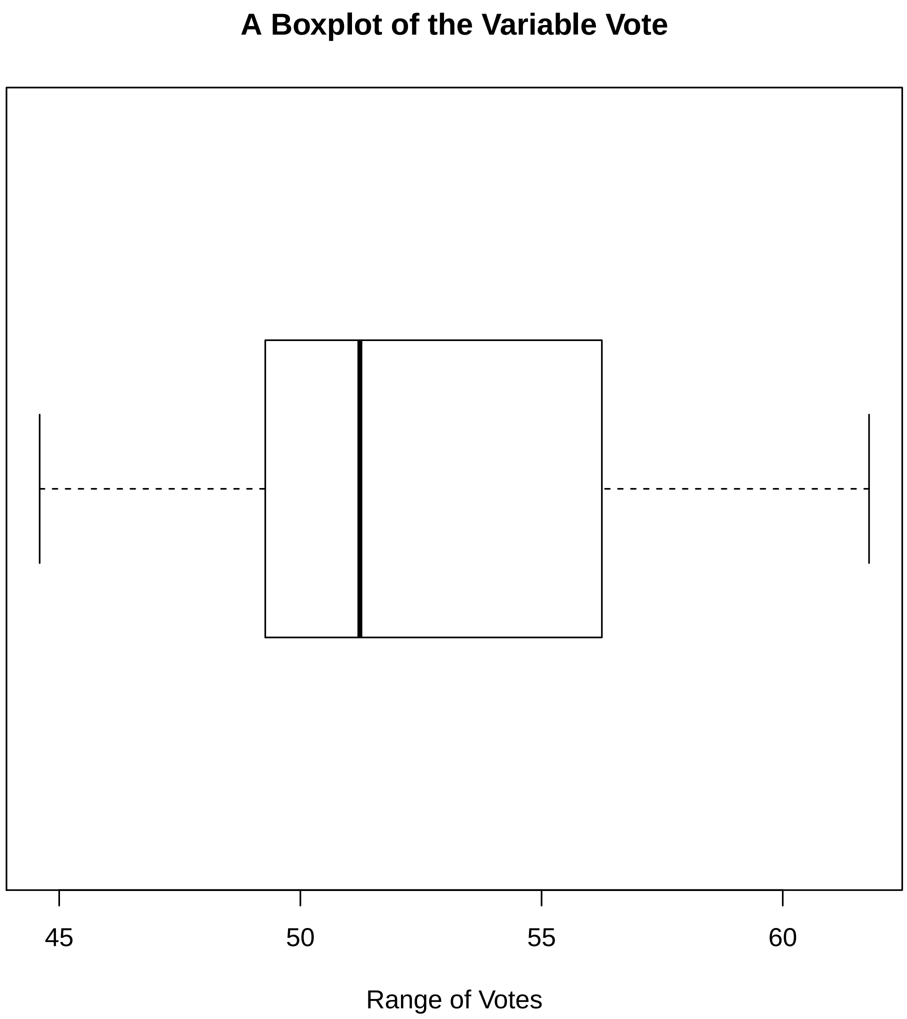

Figure 2: Boxplot of Incumbent Vote share

Variable	*Mean*	*Median*	*Mode*	*Var*	*SD*	*Range*	*IQR*
Vote	x	x	x	x	x	x	x
Growth	x	x	x	x	x	x	x

Table 1: Measures of central tendency and variability.

4. *Make a bar plot of the party affiliation of incumbent presidential candidates.*

   Include the code for the bar plot and the plot here.

5. *During the presidential campaign in 1992, Bill Clinton's campaign coined the phrase "It's the economy, stupid!" Let's investigate the relationship between the economy and electoral success. Generate a nice-looking scatterplot of economic growth and vote share. Label the data points with the year of the election. Describe the pattern that you see in your own words.*

   Include the code for the scatterplot as well as the plot here.

   Then, describe the pattern you see. In the scatterplot we can see that...

R-Code

`Finally, copy and paste the entire script here.`