

Trabajo Práctico 2: Problema de Empaquetamiento

El presente trabajo busca evaluar el desarrollo y análisis de un problema NP Completo, la implementación de una solución exacta así como una aproximada y su análisis. La primera fecha de entrega del TP será el 05/06/23, mientras que la segunda fecha de entrega (con reducción de nota) será el 12/06/23.

Definición del problema

Dado un conjunto de n objetos cuyos tamaños son $\{T_1, T_2, \dots, T_n\}$, con $T_i \in (0, 1]$, se debe empaquetarlos usando la mínima cantidad de envases de capacidad 1.

Enunciado

1. Demostrar que el problema de empaquetamiento es NP-Completo.
2. Programar un algoritmo por Backtracking/Fuerza Bruta que busque la solución exacta del problema. Indicar la complejidad del mismo. Realizar mediciones del tiempo de ejecución, y realizar gráficos en función de n .
3. Considerar el siguiente algoritmo: Se abre el primer envase y se empaqueta el primer objeto, luego por cada uno de los objetos restantes se prueba si cabe en el envase actual que está abierto. Si es así, se lo agrega a dicho envase, y se sigue con el siguiente objeto. Si no entra, se cierra el envase actual, se abre uno nuevo que pasa a ser el envase actual, se empaqueta el objeto y se prosigue con el siguiente.

Este algoritmo sirve como una aproximación para resolver el problema de empaquetamiento. Implementar dicho algoritmo, analizar su complejidad, y analizar cuán buena aproximación es. Para esto, considerar lo siguiente: Sea I una instancia cualquiera del problema de empaquetamiento, y $z(I)$ una solución óptima para dicha instancia, y sea $A(I)$ la solución aproximada, se define $\frac{A(I)}{z(I)} \leq r(A)$ para todas las instancias posibles. Calcular $r(A)$ para el algoritmo dado, demostrando que la cota está bien calculada. Realizar mediciones utilizando el algoritmo exacto y la aproximación, con el objetivo de verificar dicha relación.

4. [Opcional] Implementar alguna otra aproximación (u algoritmo greedy) que les parezca de interés. Comparar sus resultados con los dados por la aproximación del punto 3. Indicar y justificar su complejidad.

Se recomienda realizar varias ejecuciones con distintos conjuntos de datos del mismo tamaño y promediar los tiempos medidos para obtener un punto a graficar. Repetir para valores de n crecientes hasta valores que sean manejables con el hardware donde se realiza la prueba.

Ejemplo

Si tenemos el siguiente ejemplo: $T = \{0.4; 0.8; 0.5; 0.1; 0.7; 0.6; 0.1; 0.4; 0.2; 0.2\}$, la solución exacta es:

E1 = {0.5; 0.4; 0.1}
E2 = {0.8; 0.2}
E3 = {0.7; 0.2; 0.1}
E4 = {0.6; 0.4}
Total de 4 envases

La solución dada por la aproximación dada será:

E1 = {0.4}
E2 = {0.8}
E3 = {0.5; 0.1}
E4 = {0.7}
E5 = {0.6; 0.1}
E6 = {0.4; 0.2; 0.2}
Total de 6 envases

Datos de entrada

Los datos deben venir en un archivo de texto con el siguiente formato:

```
n
<linea en blanco>
T1
T2
T3
```

...

Tn

Invocación

Se debe invocar al programa como:

```
./tdatp2 <E>|<A>|<A2> <datos.txt>
```

Donde E indica que se debe calcular la solución exacta, A indica que se debe calcular la solución usando la aproximación propuesta por el curso, A2 indica que se debe calcular la solución usando la aproximación propuesta por el grupo.

Salida del programa

La salida por salida estándar deberá ser:

```
<Solución Exacta>|<Solución Aproximada>|<Solución Aproximada Alumnos>: #Envases  
<Tiempo de ejecución en mseg>
```