Auxiliar 9 - "Algoritmos Aproximados"

Profesores: Pablo Barceló
Gonzalo Navarro
Auxiliar: Dustin Cobas

P1. k-Center Clustering

El problema de *k-center clustering* se define como sigue. Se nos entrega un conjunto $P = \{p_1, \ldots, p_n\}$ de n puntos en un espacio métrico, y un entero k. Se pide que encontremos k bolas en este espacio que en su conjunto engloben a todos los puntos, de modo que el mayor de sus radios sea el mínimo posible. En otras palabras, se nos pide encontrar un conjunto $C = \{c_1, \ldots, c_k\}$ de k centros, tal que se minimiza:

$$costo(C) = \max_{i \in [n]} \min_{j \in [k]} d(p_i, c_j)$$

La intuición es que cada punto en P se asigna a su centro más cercano: el conjunto de puntos asignado a un centro específico forma un *cluster*. El radio de un *cluster* es la distancia de su centro al punto más lejano a este en el *cluster*. Finalmente, el costo de un *clustering* con k centros es el máximo radio de los *clusters* generados.

a) Suponga que $costo(C)$ es conocido, diseñe una 2-aproximación para este caso.

b) Diseñe una 2-aproximación sin tener la información anterior.

P2. Bin Packing

En el problema de *bin packing* tenemos un conjunto de n items, con tamaños $a_1, \ldots, a_n \in (0, 1]$. Por otro lado, tenemos un conjunto $B = \{1, ..., n\}$ de bins con capacidad 1. Queremos asignar items a los bins, de modo de minimizar el número de bins utilizados (no vacíos). Este problema es NP-completo.

a) Muestre una 2-aproximación para el problema de *bin packing*.

b) Muestre que no es posible lograr un factor de aproximación menor a $3/2$ en tiempo polinomial si $P \neq NP$. Para ello, recuerde que el problema de partir un conjunto X de números positivos en dos conjuntos de igual suma (una versión de *Partition*) es NP-completo.

c) Muestre una $3/2$-aproximación para el problema de *bin packing*.

Soluciones

P1. k-Center Clustering

a) La 2-aproximación se obtiene tomando un centro entre los puntos (de manera arbitraria) y agregando a su cluster todos los puntos que estén a distancia $\leq 2 \times costo(C)$ y luego repetir este proceso con los puntos restantes si me quedan centros por elegir pero no puntos los elijo arbitrariamente. Es simple notar que este algoritmo entrega una 2-aproximación, sin embargo, puede que el procedimiento no consiga cubrir todos los puntos con los centros escogidos. Veamos que esto último no sucede, pues al considerar un punto se agregan a su cluster todos aquellos que estaban en la solución óptima (y probablemente más) y por lo tanto el siguiente centro considerado pertenecerá a otro cluster de la solución óptima, si el algoritmo no alcanza a cubrir todos los puntos, significa que los cluster de la solución óptima tampoco lo hace, lo que es una contradicción.

b) Consideremos el siguiente algoritmo avaro que entrega en K el conjunto de centros:

- Elegir punto $k_1 \in P$ arbitrariamente.
- Repetir $k-1$ veces:
  - Elegir $k_i \in P \setminus K$ más lejano a K (donde la distancia de un punto a un conjunto se considera como el mínimo entre las distancias del punto y los puntos del conjunto) y agregarlo a K.

Veamos ahora que esto es una 2-aproximación. Supondremos que C es el conjunto de centros óptimo y dividiremos el análisis en 2 casos.

Caso 1 Cada cluster de C contiene exactamente un punto en K
Consideremos $p \in P$, $\bar{c} \in C$ el centro del cluster donde se encuentra p y $\bar{k} \in K$ el punto de K que está en el cluster de $\bar{c}$. Tenemos que $d(p, \bar{k}) \leq d(p, \bar{c}) + d(\bar{c}, \bar{k})$ (por desigualdad triangular), pero cada una de estas distancias es una distancia de un punto al centro de su cluster, que es menor o igual al radio de su cluster que es menor o igual a $costo(C)$ y por lo tanto $d(p, \bar{k}) \leq 2costo(C)$. Como esto se cumple para cualquier punto p el algoritmo es una 2-aproximación.

Caso 2 Hay dos centros $\bar{k}_1, \bar{k}_2 \in K$ ambos en el mismo cluster de centro $\bar{c} \in C$.
Sin mucha pérdida de generalidad supongamos que $\bar{k}_1$ fue añadido antes que $\bar{k}_2$ por el algoritmo. Consideremos ahora el momento justo antes de agregar $\bar{k}_2$ a K, entonces se cumple que $|ALG| \leq costo(K)$ (pues consideramos un conjunto de centros con menos puntos), además $costo(K) = d(\bar{k}_2, K)$ (pues $\bar{k}_2$ es el punto más alejado de K en ese momento), en particular esta última distancia es $\leq d(\bar{k}_2, \bar{k}_1)$ que por desigualdad triangular es $\leq d(\bar{k}_2, \bar{c}) + d(\bar{c}, \bar{k}_1)$ que nuevamente es $\leq 2costo(C)$, por lo que el algoritmo es una 2-aproximación.

P2. Bin Packing

Llamemos A a la suma de los tamaños de los items, $A = \sum_{i=1}^{n} a_i$. A partir de esto notamos que $|OPT| \geq \lceil A \rceil$.

a) El primer algoritmo que veremos será tomar los items e ir poniéndolos en los bins mientras quepan. De este modo si en algún momento un item no cabe en el espacio del bin que se está procesando, agregamos el item a un nuevo bin y continuamos procesando ese bin.

   Para ver que esto es una 2-aproximación notemos que como resultado del algoritmo, se cumple que para i impar, la suma de los pesos que contiene un i-ésimo procesador junto con la que contiene el $i+1$-ésimo es mayor a 1 (si no fuese así podría haber puesto el contenido de $i+1$ en i). Este argumento se puede aplicar sobre $\left\lfloor \frac{|ALG|}{2} \right\rfloor$ bins y por lo tanto $A > \left\lfloor \frac{|ALG|}{2} \right\rfloor$ y por lo tanto $\lceil A \rceil - 1 \geq \left\lfloor \frac{|ALG|}{2} \right\rfloor$ y así $|OPT| - 1 \geq \frac{|ALG|-1}{2}$ y finalmente $|ALG| \leq 2|OPT|$.

b) Supongamos que tenemos una $k < 3/2$-aproximación de Bin Packing. Consideremos una instancia $a_1, a_2, \ldots, a_n$ de Partition y la correspondiente instancia de Bin Packing con $s_i = 2a_i/A$. Veamos que la secuencia es aceptada por Partition ssi la solución del Bin Packing es 2. Como tengo una aproximación $< 3/2$ de Bin Packing, la solución que me de esta aproximación será 2 ssi la secuencia es aceptada por Partition. Finalmente si esta aproximación es polinomial, habremos encontrado una solución polinomial a Partition que es NP-completo y por lo tanto $P = NP$.

c) El algoritmo que obtiene una 3/2-aproximación considera los items en orden decreciente, llamemos a este orden de los items $s_1 \geq s_2 \geq \ldots \geq s_n$. Los elementos se procesan secuencialmente en este orden. Para cada item este se ubica en el primer bin que quepa.

   Para el análisis de este algoritmos consideremos el bin $j = \lceil (2/3)|ALG| \rceil$ y pongamonos en los siguientes 2 casos:

   - **El bin j tiene elemento de peso $> 1/2$**
     Esto significa que todos los bins anteriores (por el funcionamiento del algoritmo) tienen al menos un item de tamaño 1/2 (pues los items están ordenados decrecientemente) y por lo tanto, sabemos que hay al menos j items de tamaño $> 1/2$ que significa que $|OPT| \geq j \geq (2/3)|ALG|$ y por lo tanto $|ALG| \leq (3/2)|OPT|$.
   - **El bin j (ni ninguno que le sigue) tiene un item de peso $> 1/2$**
     Por lo tanto los bins $j, j+1, \ldots, |ALG|$ contienen al menos $2(|ALG| - j) + 1$ items y ninguno de ellos cabe en ninguno de los bins $1, 2, \ldots, j-1$ y por lo tanto al unir algunos de esos items en $j-1$ bins, tenemos que $A \geq \lceil (2/3)|ALG| \rceil$ y por lo tanto $|ALG| \leq (3/2)|OPT|$.