AA-Entrega-4

Oriol Miró López-Feliu

April 2023

1 Problema 2

Consideramos el siguiente escenario: tenemos un conjunto de n ciudades con distancias mínimas entre ellas (verifican la desigualdad triangular). Queremos seleccionar un subconjunto C de k ciudades en las que queremos ubicar un centro comercial. Asumiendo que las personas que viven en una ciudad comprarán en el centro comercial más cercano, se quiere buscar una ubicación C de manera que todas las ciudades tengan un centro comercial a una distancia menor que r en la que intentamos minimizar r sin perder cobertura. Para ello diremos que C es un r-recubrimiento si todas las ciudades están a una distancia como mucho r de una ciudad en C. Sea $r(C)$ el mínimo r para el que C es un r-recubrimiento. Nuestro objetivo es encontrar C con k vértices para el que $r(C)$ es mínimo.

(a)

Si $k \geq n$, entonces podemos seleccionar todas las ciudades como el conjunto C de k ciudades. Como resultado, cada ciudad tendría un centro comercial a distancia cero, y por lo tanto, $r(C)$ sería cero. Esta solución es óptima ya que cubre todas las ciudades y minimiza r.

(b)

Para ver que el algoritmo tiene una tasa de aproximación de 2 nos basamos en el siguiente argumento:

Si nos imaginamos el grafo de las ciudades, podemos dividir el grafo en k núcleos de puntos de manera que el radio de cada núcleo de puntos sea mínimo. Entonces, la solución óptima, C^* consistirá en los k puntos que son el centro de esos núcleos. Este argumento se puede generalizar para cualquier longitud de n y k.

Entonces, nuestro algoritmo nos proporcionará una solución C, la qual primero escogerá un punto aleatoriamente y seguidamente irá añadiendo los puntos más alejados. Volviendo al argumento de los núcleos, este algoritmo no escogerá los puntos que son centrales del núcleo, sino aquellos que están en el borde.

Sea pues $r^* = r(C^*)$, esta r^* será la máxima distancia entre el centro de uno de los k nucleos hasta su punto más lejano. Ahora consideremos $r = r(C)$. Esta distancia, para cubrir todo el núcleo que tiene asignado, tendrá que cubrir hasta el extremo opuesto de núcleo. Sea x el punto del núcleo seleccionado por C, y el punto seleccionado por C^* y z un tercer punto del núcleo, en el peor caso $dist(x, y) = dist(y, z)$, con lo qual necesitamos una $r = 2 * r^*$ para cubrir el nucleo entero. Vease que si en el núcleo de puntos de radio mayor $dist(x, y) \leq dist(y, z)$ entonces el $r^* = dist(y, z)$ y por tanto $r \leq 2 * r^*$. En el caso opuesto, si $dist(x, y) \geq dist(y, z)$, entonces $r^* = dist(x, y)$ y una vez más $r \leq 2 * r^*$.

Hemos demostrado entonces que el algoritmo produce un resultado C con una tasa de aproximación 2.