Ampliació d'Algorismia **QP 2016-2017**

Algorismes d'aproximació

1 Algorismes d'aproximació

1. Consider the Max Clique problem. Given an undirected graph $G = (V, E)$ compute a set of vertices that induce a complete subgraph with maximum size.

   For each $k \geq 1$, define G^k to be the undirected graph (V^k, E^k), where V^k is the set of all ordered k-tuples of vertices from V. E^k is defined so that $(v_1, \ldots, v_k)$ is adjacent to $(u_1, \ldots, u_k)$ iff, for $1 \leq i \leq k$, either $(v_i, u_i) \in E$ or $v_i = u_i$.

   - (a) Prove that the size of a maximum size clique in G^k is the k-th power of the corresponding size in G.
   - (b) Argue that if there is a constant approximation algorithm for Max Clique, then there is a polynomial time approximation schema for the problem.

2. Consider the MAXIMUM COVERAGE problem: Given sets $S_1, \ldots, S_m$ over a universe of elements $U = \{1, \ldots n\} = \cup_{i=1}^{n} S_i$ and a positive integer k. Choose k sets that cover as many elements as possible. Es decir

   $$opt(x) = \max\{| \cup_{i \in I} S_i| \mid |I| = k\}.$$

   Provide a greedy approximation algorithm with rate $\frac{e}{e-1} \approx 1.58$.

3. 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 compraran en el centro comercial más próximo se quiere buscar una ubicación C de manera que todas las ciudades tengan un centro comercial a 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 distancia como mucho r de una ciudad en C. Sea $r(C)$ el mínimo r para el que C es un r-cover. Nuestro objetivo es encontrar C con k vértices para el que $r(c)$ es mínima.

   (a) Demuestra que si $k \geq n$, la solución formada por todas las ciudades es óptima.
   A partir de ahora asumiremos que $k \leq n$.

   (b) Diseña un algoritmo que dado C calcule $r(C)$. Analiza su coste.

   (c) El problema es NP-dificil?

   (d) Suponiendo que S es el conjunto de ciudades, considera el siguiente algoritmo:

   Seleccionar cualquier ciudad $s \in S$ y define $C = \{s\}$
   while $|C| \neq k$ **do**
   seleccionar una ciudad $s \in S$ que maximiza la distancia de s a C;

$C = C \cup \{s\}$;
end while;
return C

Demuestra que es un algoritmo de aproximación con tasa de aproximación 2.

4. Consider the following algorithms that have as input a boolean formula in 3-CNF

   Algorithm A

   (a) Let c_0 be the number of true clauses when all variables are set to value 0.
   (b) Sea c_1 be the number of true clauses when all variables are set to value 1.
   (c) return the maximum among c_0 and c_1.

   Algorithm B

   (a) For each variable, toss a fair coin to determine its value, 0 or 1.
   (b) return the number of true clauses under this random assignment.

   Taking as value provided by the algorithm the expected number of clauses in algorithm B, is there a constant r for which the algorithms are r-approximations for MaxSat?

5. Consider the following algorithm:

   function EDGES(G:graph)
   $U = \emptyset$; $E = E(G)$;
   while $E \neq \emptyset$ **do**
   select any edge $e = (u, v) \in E$;
   $U = U \cup \{u, v\}$;
   $E = E - \{e' \in E \mid e' \text{ incident to } e\}$
   end while;
   return U
   end function

   Prove that Edges is a 2-approximation for the Minimum vertex cover problem.

6. *Planificació.*

   Tenim un conjunt de n tasques. La tasca i es descriu per un parell $T_i = (s_i, d_i)$ on s_i es el seu temps de disponibilitat i d_i la seva duració. L'objectiu es planificar totes les tasques en un processador de manera que: (a) cap tasca s'executa abans del seu temps de disponibilitat, (b) les tasques s'executen sense interrumció, i (d) el temps de finalització del procesament de totes les tasques sigui mínim.

   Se sap que, quan és possible interrompre qualsevol tasca en execució i reiniciar-le més tard des del punt d'aturada, l'algorisme voraç que executa en cada punt de temps la tasca (disponible) més propera al seu temps de finalització, aconsegueix el desitjat mínim. Anomenarem a aquest algorisme Voraç1.

   Considereu el procediment següent:

   (a) Executa Voraç1.

(b) Ordena les tasques en l'ordre ascendent del seu temps de finalització d'acord amb la solució proporcionada per versió Voraç1.

(c) Les tasues s'executeu en aquest ordre i sense interrupció, afegint el temps d'espera necesari fins que la tasca estigui disponible.

Demostreu que aquest algorisme es una 2-aproximació pel problema plantejat.

7. Consider the following randomized algorithm.

function RWVC(G:weighted graph)
  $U = \emptyset$;
  while $E \neq \emptyset$ **do**
    Select an edge $e = (v, t)$;
    Randomly choose x from $\{v, t\}$ with
    $P[x = v] = \frac{w(t)}{w(v)+w(t)}$;
    $U = U \cup \{x\}$;
    $E = E - \{e \mid x \text{ is an end-point of } e\}$;
  end while;
  return U
end function

Prove that RWVC is a randomized 2-approximation for Minimum weighted vertex cover problem.

8. Provide an integer programming formulation for the following problems. Formulate their corresponding LP relaxation an their duals.

   - Max Flow
   - Max Weighthed Cut
   - Max Set Cover
   - TSP

9. Consider the network design problems seen in class. Prove that the primal-dual algorithm P-D-NetDes provides a 2-approximation for Tree Partition and Steiner Forest.

10. The Set Packing problem is defined as follows: Given a family of sets $S_1, \ldots, S_m \subseteq U$ such that, for $1 \leq i \leq m$, $|S_i| = 3$ and has profit $c(S_i)$, find a subset of these sets that maximizes the profit, while each element is covered at most once. Consider a straightforward integer linear programming formulation of the problem:

$$\max \sum_{i=1}^{m} c(S_i) x_i$$
$$\sum_{i \mid u \in S_i} x_i \leq 1 \qquad \forall u \in U$$
$$x_i \in \{0, 1\} \qquad 1 \leq i \leq m$$

Consider the following algorithm that, first computes an optimal solution x^* of the LP obtained by relaxing $x_i \in [0, 1]$, and second performs the following rounding algorithm

- (1) Choose a set S_i to be in the solution with probability $x_i^*/6$.
- (2) If an element $u \in U$ is covered by more than one set, remove all the sets in the solution that contain u.

Show that the proposed algorithm is a randomized 12-approximation for Set packing

11. *Sales de joc*

La Societat d'Amics dels Videojocs (SAV) té una col·lecció de m locals a la ciutat de Barcelona i un total de n socis. Volen determinar en quins locals els hi convé obrir una sala de joc juntament amb una assignació de socis a les sales de joc.

La SAV per una part ha fet una estimació del cost d'adequar un local com a sala de joc i així, per cada local i, te una estimació del cost l_i. Per una altre part, la SAV vol tenir en compte el cost que té per als socis desplaçar-se fins al local assignat. Així disposa dels valors $c(i,j)$ que indican la distancia que habitualment el soci $i$ha de recorre per arribar al local j.

L'objectiu de la SAV es trobar una solució en la que es minimitze la suma dels costos d'adequació dels locals seleccionats més la suma de les distancies dels desplaçaments dels socis a les sales assignades.

Q1. Doneu una formalització com problema de programació entera d'aquest problema. Feu servir una variable x_j perla seleccion del local j y una variable y_{ij} per la possible assignació del soci i al local j.

Considereu el problema de programació lineal obtingut desprès de relaxar les condicions d'integritat. Sigui x^*, y^* una solució òptima del programa relaxat.

Considereu el següent procès:

(a) Per cada soci i, sigui $\tilde{c}_i = \sum_j c(i,j)y_{ij}^*$, la distància mitjana del soci i a les sales asignades en y^*.
(b) Per cada soci i, sigui $S_i = \{j \mid c(i,j) \leq 2\tilde{c}_i\}$.
(c) Per i,j, if $j \notin S_i$, sigui $\tilde{y}_{ij} = 0$ en cas contrari $\tilde{y}_{ij} = y_{ij}^*/\sum_{j\in S_i} y_{ij}^*$.
(d) Per cada local j, sigui $\tilde{x}_j = \min(2x_i^*, 1)$.

Q2. Demostreu que, per tot i i tot j, $\tilde{y}_{ij} \leq 2y_{ij}^*$.

Q3. Demostreu que $\tilde{x}$, $\tilde{y}$ és una solució factible del problema de programació lineal i que $\sum_{i,j} c(i,j)\tilde{y}_{ij} \leq 2\sum_{i,j} c(i,j)y_{ij}^*$.

Ara, donats $\tilde{x}$, $\tilde{y}$, considereuel següent procès, mentre quedin socis sense assignar a una sala:

(a) Seleccionem el soci i no assignat amb $\tilde{c}_i$ mínim.
(b) Obrim una sala de joc al local j que minimitzi el valor $\min_{j\in S_i} l_i$
(c) Assignem el soci i a la sala j.
(d) Tots els socis i' tals que $S_i \cap S_{i'} \neq \emptyset$ s'assignen a la sala j.

Q4. Demostreu que la solució així obtinguda es una 6 aproximació per al problema plantejat.

12. The k-SET COVERING problem is as follows: Given a family of sets $S_1, \ldots, S_m \in U$ of cardinality at most k with cost $c(S_i)$, find a subset of these sets that minimizes the total cost, while each element in U has to be covered at least once.

   (a) Provide a integer programming formulation of the problem.

   (b) Let x' be an optimum basic solution for the LP relaxation of the IP formulation.

   Consider the following iterative rounding algorithm:

   while $U \neq \emptyset$ **do**
      Compute an optimum basic solution x'
      Choose i with $x'_i \geq 1/k$
      Buy set S_i, delete elements in S_i from the instance
   end while
   Output the bought sets

   Prove that this algorithm gives a k-approximation.

   Hint How much does the value of the optimum fractional solution decrease in each iteration compared to the bought set?