

3 Algorismes d'aproximació

- 3.1. El problema del Tall Maxim (MAX-CUT). Donat un graf no dirigit $G = (V, E)$, el problema del maximum cut (MAX-CUT) es trobar la partició $S \cup \bar{S}$ de V tal que maximitze el nombre d'arestes entre S i \bar{S} ($C(S, \bar{S})$), on $S \subseteq V$ i $\bar{S} = V - S$. La maximització entre totes les possibles particiones de V . El problema del MAX-CUT és NP-hard en general. Donat

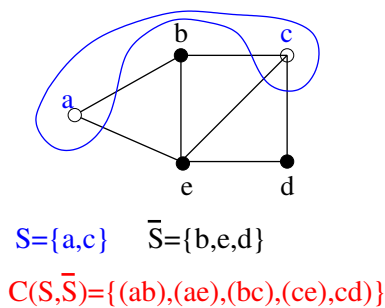


Figura 1: Exemple de MAX-CUT, amb cost òptim 5

$G = (V, E)$ considereu el següent algorisme golafre (greedy) pel problema del MAX-CUT: Ordeneu els vèrtexs en ordre decreixent respecte al seu grau (nombre veïns). Considereu el resultat de l'ordenació s'escriu com a (v_1, v_2, \dots, v_n) .

Inicialment tenim $S = \emptyset = \bar{S}$. Al primer pas $v_1 \in S$. Al pas i -èsim col·loquem v_i a S o \bar{S} de manera que maximitze $|C(S, \bar{S})|$.

- (a) Demostreu la correctesa i doneu la complexitat de l'algorisme golafre.
- (b) Demostreu que aquest algorisme golafre és una 2-aproximació al MAX-CUT

3.2. 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) Let 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 .

Is there a constant r for which the algorithm is a r -approximation for MaxSat ?

3.3. Consider the following algorithm:

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

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

3.4. 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 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 -recubrimiento. Nuestro objetivo es encontrar C con k vértices para el que $r(C)$ es mínimo.

(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) 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  $C$ ;
     $C = C \cup \{s\}$ 
return  $C$ 
```

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

3.5. 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 interrupció, 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 tasques s'executen en aquest ordre i sense interrupció, afegint el temps d'espera necessari fins que la tasca estigui disponible.

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

- 3.6. Donat com a entrada un graf no dirigit $G = (V, E)$, definim el problema de GST (graf sense triangles) com el problema de seleccionar el màxim nombre d'arestes E , de manera que el graf $G' = (V, E')$ no contingui cap triangle (i.e. no hi han 3 vèrtexs x, y, z tal que $(x, y), (x, z)$ i (y, z) siguin arestes a G'). Aquest problema és NP-hard.

Donat $G = (V, E)$ considereu el següent algorisme golafre (greedy):

Ordeneu els vèrtexs en ordre decreixent respecte al seu grau (nombre veïns). Considereu el resultat de l'ordenació s'escriu com a (v_1, v_2, \dots, v_n) .

Inicialment tenim $S = \emptyset = \bar{S}$. Al primer pas $v_1 \in S$. Al pas i -èsim col·loquem v_i a S o \bar{S} de manera que maximitze $|C(S, \bar{S})|$. Prenem com $E' = \{(u, v) \mid u \in S, v \notin S\}$

Aquest algorisme, és una 2-aproximació polinòmica al problema?.

3.7. 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, \dots, v_k) is adjacent to (u_1, \dots, 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.

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

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

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

- 3.9. The Barcelona Diagonal University campus is shared among the UPC, UB, CSIC, and other public and private entities. As a result there are many surveillance cameras placed outside the buildings whose images can be accessed by different surveillance companies and in some cases by different surveillance teams inside the company. Each camera has a unique surveillance team that can visualize the images on the spot. The UPC Chancellor needs to establish a surveillance plan to be set during the next Chancellor elections. The aim of the plan is to guarantee that the voting is done without pressure.

To establish the plan the UPC Committee has set a collection of *surveillance locations* that require visual surveillance. For each of those locations, they have collected the surveillance teams that have cameras visualizing the location. Fortunately all the surveillance locations can be covered by an already existing camera, so the task can be performed without additional infrastructure.

Due to the economical situation, the UPC Chancellor wants to expend the minimum possible amount of money. The administrative office has negotiated and agreed the surveillance cost with each of the surveillance teams. Now they are seeking a solution that allows to keep surveillance on all the surveillance locations with minimum total cost. Assume, for this exercise, that a surveillance team receives a fix amount of money for looking to the images of all the cameras assigned to it.

Consider the algorithm that repeatedly chooses the team that cover the maximum number of still uncovered locations.

Is this algorithm a constant factor approximation for the problem? Justify its correctness and efficiency.

- 3.10. The association for the promotion of the European Identity is planning a workshop formed by a series of debates over different European topics. They have a *participant's* list formed by the persons that have committed to participate in the workshop provided the association issues them a formal invitation.

The organizing committee in view of the planned topics and previous experiences has selected for each debate two disjoint lists of people: the *success* list and the *failure* list. The committee is considering those list in an optimistic perspective. The presence of at least one of the persons in the success list or the absence of at least one of the persons in the failure list of a particular debate guarantees that this debate will be successful.

The organizing committee faces the problem of selecting the subset of people to whom the association will send a formal invitation. The association wish to invite a set of people that guarantees that the number of successful debates (according to the previous rule) is maximized.

Design a 2-approximation algorithm for the problem. Justify its correctness and efficiency.

- 3.11. Considereu una formula booleana sobre un conjunt de n variables en 3-CNF, totes les clausules tenen 3 literals diferents. Una 3-MCNF és una formula en 3-CNF on, a més, cap literal és la negació d'una variable.

Per suposat una 3-MCNF sempre admet una assignació de valors a les variables que la satisfà. Per això considerem el problema min-3-MCNF que donada una formula booleana F en 3-MCNF ha de trobar una assignació de valor a les variables que satisfaci la formula amb el nombre més petit possible de variables a 1.

- (a) Considereu l'algorisme següent:

```
1: procedure G-3-MCNF( $F$ )
2:    $F$  en 3-MCNF amb clausules  $\mathcal{C} = \{C_1, \dots, C_m\}$  over variables  $x_1, \dots, x_n$ 
3:    $x_i = false$ , per  $1 \leq i \leq n$ 
4:   while  $\mathcal{C} \neq \emptyset$  do
5:     Extreu una clausula  $C$  de  $\mathcal{C}$ 
6:     Sigui  $x_i$  una de les variables a  $C$ 
7:      $x_i = true$ 
8:     Treu de  $\mathcal{C}$  totes les clausules que continguin  $x_i$ 
9:   Return  $x$ 
```

És G-3-MCNF un algorisme d'aproximació constant per min-3-MCNF?

- (b) Doneu una 3-aproximació per min-3-MCNF.