Ampliació d'Algorismia:
Aproximation, Parameterization and Streaming **QP 2022-2023: Sheet 1**

The definition of approximation algorithm.

- Let $\mathcal{P}$ be an optimization problem.
- For any instance x of $\mathcal{P}$ let $\mathsf{opt(x)}$ be the cost of an optimal solution.
- Let $\mathcal{A}$ be an algorithm such that for any instance x of $\mathcal{P}$ computes a feasible solution with cost $\mathcal{A}(x)$.

$\mathcal{A}$ is an r-approximation for $\mathcal{P}$ ($r \geq 1$) if for any instance x of $\mathcal{P}$

$$\frac{1}{r} \leq \frac{\mathsf{opt}(x)}{\mathcal{A}(x)} \leq r$$

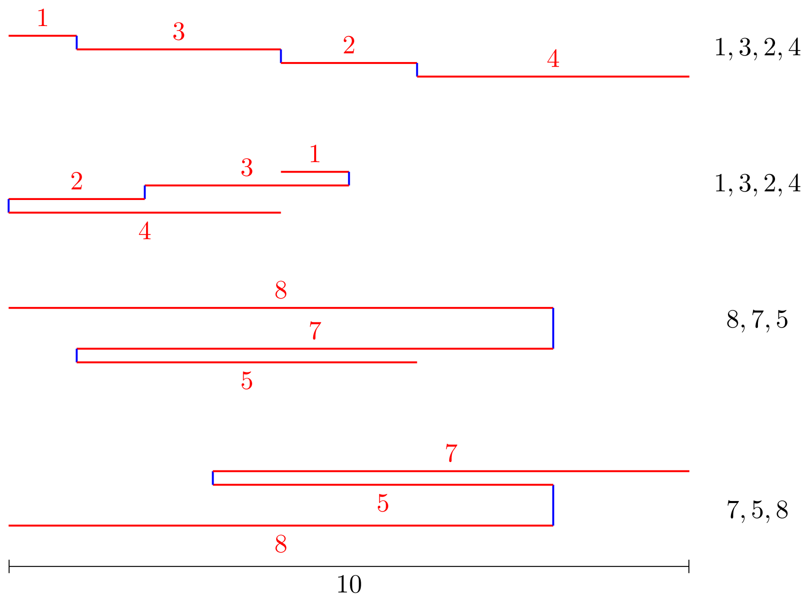
$\mathcal{P}$ is r-approximable in polynomial time if there is a polynomial time computable r-approximation for $\mathcal{P}$.

Un exemple adaptat de la llista de problemes de Algorismia

Un metre de fuster (com el de la figura de sota) està format per uns quants segments de fusta habitualment iguals. Cada segment és rígid i s'uneix al previ i/o al següent pels extrems de manera que es pot rotar completament a les unions.

En aquest problema considerarem una generalització de metre de fuster en el que els segments poden tenir longituds diferents encara que tots tenen la mateixa amplada. A més cada segment té com a molt 100cm de llargada. Així un metre de fuster està format per n segments de llargades $l_1, \dots, l_n$ (en aquest ordre) on totes les longituds dels segments son enters al interval $[0, 100]$. Per simplificar la notació considerarem també els extrems $A_0, \dots A_n$, on A_0 és l'extrem lliure del primer segment, A_1 és l'extrem comú al primer i segon segment, etc, i A_n és l'extrem lliure del segment n-ésim.

Volem analitzar el problema de plegar el metre per tal de ficar-lo a dintre d'una caixa. Per exemple, si els segments són de longitud 1, 3, 2 i 4, el metre es pot guardar en una caixa de longitud 10 (plegar a l'interval [0,10]), però podem fer-ho també en una caixa de longitud 5 (a l'interval [0,5]). Si els segments tenen longitud 8, 7 i 5 en aquest ordre, es pot guardar plegat en una caixa de 8, però si els segments son 7, 5 i 8, llavors la caixa més petita en la que es pot plegar té longitud 10 metres. A la figura de sota teniu una representació estilitzada i bidimensional d'aquests plegaments.



1. Considereu l'algorisme següent

   procedure Fold inside interval($L(n)$)
       Let $m = \max L[i]$
       Place A_0 at position 0
       for $i = 1, \dots, n$ **do**
           if it is possible to place A_i to the left of A_{i-1} inside $[0, 2m]$ **then**
               place A_i to the left of A_{i-1}
           else
               place A_i to the right of A_{i-1}

   Demostreu que Fold inside interval determina un plegat que ens permet ficar el metre dintre de l'interval $[0, 2m]$ on m es la longitud del segment més llarg i analitzeu-ne el seu cost.

2. Considereu el problema Min-Fold: Donat un metre de fuster, format per n segments de llargades $l_1, \dots, l_n \in \mathbb{N}$ (en aquest ordre), $0 < l_i \leq 100$, trobar la llargada ℓ més petita que ens permeti ficar el metre dintre de l'interval $[0, \ell]$. És Fold inside interval una 2-aproximació a Min-Fold?

Una solución.

1. Solo necesitamos comprobar que cuando colocamos A_i a la izquierda de A_{i-1} la posición de A_i está dentro de $[0, 2m]$. En este caso sabemos que A_i no se puede colocar a la derecha, por lo tanto si $j \in [0, 2m]$ es la posición de A_{i-1}, tenemos que $j + L[i] > 2m$, como $L[i] \leq m$, tenemos que $j > m$. Por lo tanto $j - L[i] > 0$. Por lo tanto el plegado nos permite colocar el metro en $[0, 2m]$.

2. Teniendo en cuenta que m es el segmento de longitud máxima, cualquier plegado en $[0, k]$ requiere que $k \geq m$. En particular la solución optima en $[0, k^*]$ tiene que cumplir $k^* \geq m$ y como la que obtenemos en (a) es $2m$, $2m \leq 2k^*$ con lo que podemos concluir que Fold inside interval es una 2-aproximación.

Exercises

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

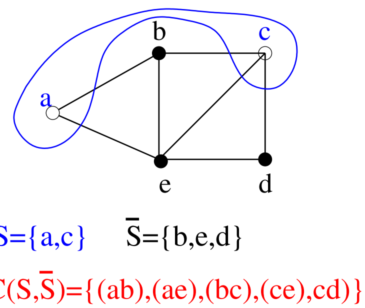


Figura 1: Exemple de MAX-CUT, amb cost òptim 5

$G = (V, E)$ 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 $(v_1, v_2, \ldots, v_n)$.
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 correctessa i doneu la complexitat de l'algorisme golafre.

   (b) Demostreu que aquest algorisme golafre és una 2-aproximació al MAX-CUT

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 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-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. 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.

   Is there a constant r for which the algorithms is a r-approximations for MaxSat ?

4. 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.

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 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.

6. 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

   $P[x = v] = \frac{w(t)}{w(v)+w(t)}$;

   $U = U \cup \{x\}$;

   $E = E - \{e \mid x \text{ is an end-point of } e\}$;

   return U

   When the algorithm is randomized, we take $\mathcal{A}(x)$ as the average cost of the solutions produced by $\mathcal{A}$ on input x.

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

7. 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 $(v_1, v_2, \ldots, v_n)$.
   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?.

8. 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.

9. 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, that is

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

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

10. 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.

11. 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.

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

    (a) Considereu l'algorisme següent:

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

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

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

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

    Q1. Provide a integer programming formulation for the problem (IP-SC). For this consider m variables $x_1, \ldots x_m$ where variable x_i indicates whether set S_i is or is not in the packing.

    Let LP-SC be the Linear Programming problem obtained by relaxing the $x_i \in \{0,1\}$ conditions in IP-SC by $x_i \in [0,1]$.

    Consider the following algorithm that, first computes an optimal solution x^* to LP-SC, and second rounds the solution according to the the following randomized 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.

    Q2. Show that the proposed algorithm is a randomized 12-approximation for Set packing

14. ***Min makespan scheduling.*** Consider the following scheduling problem: Given n jobs, m machines and, for each $1 \leq i \leq m$ and each $1 \leq j \leq n$, the amount of time t_{ij} required for the i-th machine to process the j-th job, find the schedule for all n jobs on these m machines that minimizes the makespan, i.e., the maximum processing time over all machines. You can assume that once a job starts in a machine it must run in this machine until it finishes.

    A solution for the problem can be represented by 0-1 variables, x_{ij}, $1 \leq i \leq m$ and $1 \leq j \leq n$, indicating whether the i-th machine will process the j-th job.

    (i) Using those variables provide an integer programming formulation for the problem and its corresponding LP relaxation.

    Le us call *fractional optimal schedule* an optimal solution of the LP and opt the makespan of an optimal solution of the IP.

    Observe that if $t_{ij} > \mathsf{opt}$, then job j will not be assigned to machine i in an optimal solution. This suggest that we can set an upper bound T and set $x_{ij} = 0$, whenever $t_{ij} > T$. Of course, we need to seek a suitable bound T so that a good assignment of tasks to machines is still possible.

    (ii) Provide an LP formulation to check if, for a given bound T, a fractional optimal schedule in which $x_{ij} = 0$ when $t_{ij} > T$ exists.

    (iii) Show how to find in polynomial time the value T^*, the minimum among all the integers T's for which the previous condition holds.

    Assume that x^* is a fractional optimal schedule solution in which $x_{ij} = 0$ when $t_{ij} > T^*$.

    Let $J = \{j \mid \exists i \, 0 < x^*_{ij} < 1\}$ and $M = \{1, \ldots, m\}$. Consider the bipartite graph $H = (M, J, E)$ where $E = \{(j, i) \mid < x^*_{ij} < 1\}$; that is, there is an edge (j, i) connecting j to i if and only if the j-th job is partially assigned to the i-th machine.

    Assuming that a maximum matching in H covers all the vertices in J:

    (iv) Provide a rounding algorithm to obtain, from x^*, a feasible solution to Min makespan scheduling. Your rounding algorithm should provide a 2-approximate solution.

    (v) Justify whether, the process to compute such a 2-approximate solution requires polynomial time or not.

15. 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.

Q1. Provide a integer linear programming formulation for the problem.

Let x' be an optimum 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
Output the bought sets

Q2. 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?

16. *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.

17. La DAFIB vol posar en marxa una nou mètode per recaptar més fons a la festa de la FIB. La DAFIB porta un total de n regals embolicats. Cada paquet està etiquetat amb 5 identificadors diferents. Els identificadors es treuen d'un conjunt de m identificadors possibles.

    Després es posen a la venda els identificadors. Un Fiber pot comprar tants identificadors com vulgui, però només hi ha un identificador de cada tipus a la venda. Per cada identificador, es porta a terme una subhasta amb un preu mínim de partida de M euros. Per aconseguir un regal cal aportar al menys tres dels cinc identificadors al paquet. Quan un Fiber reclama regals, aporta tots el identificadors que ha comprat i obté tots els regals aconseguits.

    La DAFIB sap que, si els participants no es coordinen massa, es relativament difícil reclamar regals i que a més es venen molts identificadors a un preu molt més alt del mínim. Encara i això volen assegurar que el valor dels regals es recuperable. Per això, donat un etiquetat dels n paquets E i un preu de partida M, defineixen la *venda mínima* ($V(E, M)$) com el preu que dels identificadors que compraria un Fiber en la següent situació extrema: només participa ell a la subhasta (al no haver-hi competència mai pag més del preu mínim per un identificador); vol reclamar tots els regals però pagant el mínim possible.

    (a) Proporcioneu una fórmulació com a problema de programació lineal entera del problema d'obtenir $V(E, M)$, donats E i M. Justifiqueu per què la vostra fórmulació és correcta.

    (b) Fent servir el apartat anterior i la tècnica de relaxació i arrodoniment, proporcioneu un algorisme d'aproximació per al problema. Justifiqueu la seva correctesa i analitzeu el cost de l'algorisme i la qualitat de la solució proporcionada.

18. The 3-Set Cover 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 minimizes the profit, while their union covers all elements of U. You can assume that U is the union of all the given sets.

    Provide a integer linear programming formulation for the problem. Use this to devise a primal-dual algorithm and analyze whether it provides a constant approximation for the problem.

19. Consider the Min 5-Hitting set problem: Given a collection of subsets $\mathcal{S} = (S_1, \ldots, S_m)$ of $U = \{1, \ldots, n\}$, so that, for $1 \leq i \leq m$, $|S_i| = 5$, find a subset $A \subseteq U$ of minimum cardiality so that $a \cap S_i \neq \emptyset$, for $1 \leq i \leq n$.

    Provide a integer linear programming formulation for the problem. Use this to devise a primal-dual algorithm and analyze whether it provides a constant approximation for the problem.