

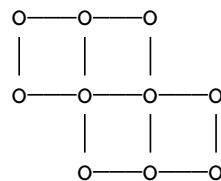
Answers can be done in English or in French. Documents (course and personal notes) allowed. The official lecture notes will be available.

Vertex Cover

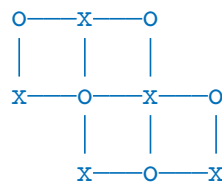
Question 1. Explain what is a “minimum vertex cover” for a given graph.

SOLUTION. This is a subset of vertices S of a graph G , with minimum cardinality, such that for every edge $u - v$ of G , either $u \in S$ or $v \in S$.

Question 2. Find a minimum vertex cover for the graph G_0 depicted below. (You can draw the graph with your solution).



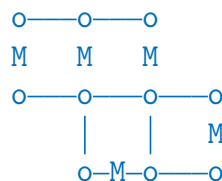
SOLUTION.



It is easy to check that every edge is of the form $x-o$. Therefore, the solution (x) is a vertex cover. Next question will prove the size is optimal.

Question 3. For G_0 , find a matching of maximum size, i.e., a set of independent edges of maximum cardinality. (You can draw the graph with your solution).

SOLUTION.



The matching is of maximum size because it covers all the vertices of the graph. There are 10 vertices and 5 edges in the matching.

Question 4. For a general graph G , what is the relation between the size of a matching and the size of a vertex cover?

SOLUTION. For every matching M and vertex cover C for G , we must have $|C| \geq |M|$, because otherwise an edge of M is not covered, a contradiction with the definition of C .

We consider the following algorithm (seen in the lecture), that applies to any connected graph G . See Figure 1 for an example.

1. Compute for G a DFS spanning tree T .
2. Return the set S of inner vertices of T , i.e., all its vertices but its leaves.

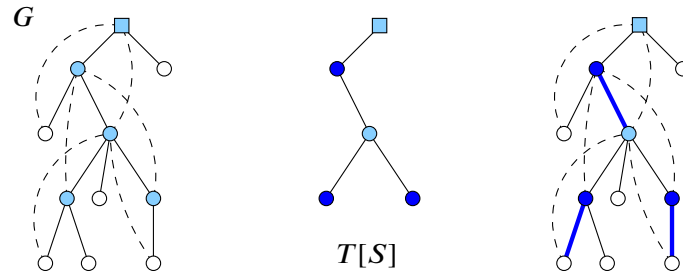


FIGURE 1 – A vertex cover S (colored vertices) for G , computed from a DFS spanning tree T . Edges of $G \setminus T$ are dotted. A 2-coloring for $T[S]$ shows that G (in fact T) contained a matching of size at least $|S|/2$ (bold edges in the left picture).

We have seen that Approx_VC2 is a 2-approximation for the minimum vertex cover, assuming the graph is connected.

Question 5. Explain what does it mean that Approx_VC2 is a 2-approximation ?

SOLUTION. It means that the algorithm returns a vertex cover computed in polynomial time whose size is at most twice the optimal value, the approximation factor is 2.

Question 6. Show that the tree T in Figure 1 contains a matching of size 5.

SOLUTION. One can choose five leaves with as many independent parents. This defines a matching of size five.

Question 7. What is the approximation factor of Approx_VC2 for the graph G , given the tree T , as in Figure 1 ? Justify.

SOLUTION. The algorithm returns a vertex cover S with $|S| = 5$. On the other hand, we have seen that it has a matching of size five. Therefore, the minimum vertex cover is 5. The approximation factor is therefore 1 in this case.

Question 8. Find a critical instance for Approx_VC2, noting that the input graph must be connected. Justify.

SOLUTION. Consider a path with $2k$ edges (and $2k + 1$ vertices) where one extremity is chosen to be the root of the DFS spanning tree. The optimal solution is k , whereas the algorithm will return $2k$ vertices, i.e., all the $2k + 1$ vertices excepted the last vertex, the leaf.

In a rooted tree, a *pre-leaf* is a vertex that has a child which is a leaf. For instance, the tree T in Figure 1 has 5 pre-leaves, and it contains a matching of the same size.

Question 9. Do you think that, in general, the number of pre-leaves is a good lower bound in order to upper bound the approximate factor of Approx_VC2 ? i.e., discuss of the relation between minimum vertex cover for general graph G and the number of pre-leaves in any DFS spanning tree T .

SOLUTION. Le nombre de pre-leaves est bien une borne inférieure à la taille de tout vertex cover, car on peut toujours construire un matching de cette taille. Malheureusement, cela ne permet pas de borner “efficacement” la taille du vertex cover, par exemple si T est un chemin, donc avec une feuille. On en déduit un vertex cover de taille au moins 1, alors que le vertex cover optimal est de taille $\Omega(n)$. Le rapport n’est pas constant.

We propose the following modification for Approx_VC2. At Step 2, return all vertices of T with degree > 1 in T (instead of inner vertices).

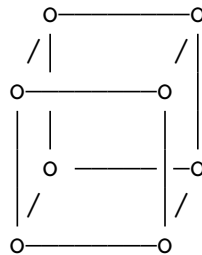
Question 10. Does this modification lead to a 2-approximation for minimum vertex cover? Justify.

SOLUTION. The only difference with the original algorithm is that the root might be not returned, if its degree is one. This is a problem if there is an edge between a leaf and the root. The returned set might be not a vertex cover. This occurs in Figure 1 if we remove the leaf attached to the root of T .

Sphere Cover

Consider a set P of points in the 3D Euclidean space, that is $P \subset \mathbb{R}^3$. The *circularity* of P is the largest number c such that there exists a plane $H \subset \mathbb{R}^3$ with a circle $C \subset H$ containing c points of P .

For instance, if P is the set of the 8 points corresponding to the vertices of a cube (see the figure below), then the circularity of P is 4. Indeed, (1) each face defines a plane H with a circle C containing 4 vertices (there are planes other than these 6 faces containing 4 vertices as well); and (2) no planes of $\mathbb{R}^3$ can contain a circle with more than 4 vertices of the cube.



We will assume to be given two routines running in constant time :

- $\text{circle}(u, v, w)$ that, given 3 points $u, v, w \in \mathbb{R}^3$, returns a circle C containing u, v, w .
- $\text{check}(p, C)$ that says whether a point p belongs to a circle C or not.

In the case where u, v, w are aligned, we can assume that $\text{circle}(u, v, w)$ returns a special circle C with infinite radius (in fact a line), and that $\text{check}(p, C)$, in this case, says whether p is aligned with u, v, w or not.

Question 11. Give an algorithm that computes the circularity of any point set $P \subset \mathbb{R}^3$. Describe its principle and give its complexity expressed as a function of $n = |P|$.

SOLUTION. [3 pts | $\sum 3.0$] Pour chaque triplet u, v, w de P , calculer $C = \text{circle}(u, v, w)$. Puis vérifier combien de points de P appartiennent à C , avec $\text{check}(\cdot, C)$. Cela prend un temps $O(n)$ pour chaque triplet, soit $O(n^4)$ en tout.

A *sphere cover* for the point set $P \subset \mathbb{R}^3$ is a collection $\mathcal{C}$ of spheres such that every point of P belongs to the surface of some sphere of $\mathcal{C}$. We consider the decision problem SPHERE COVER defined as follows :

Instance : A point set $P \subset \mathbb{R}^3$ and $k \in \mathbb{N}$.

Question : Does P has a sphere cover of size k ?

Recall that it takes a constant time to check whether a points $p \in \mathbb{R}^3$ belongs to the surface of some sphere described by its center u and radius r . Indeed, it is sufficient to check that the Euclidean distance between u and p equals r .

Question 12. Give a reduction rule showing that SPHERE COVER admits a kernel whenever the problem is parametrized by k and the circularity c of P . (A complete justification is required here! You have to show that your reduction is valid and is polynomial (give its complexity), and you have to express the size of the kernel as function of the parameters.) [Hint : Observe that the circularity of any subset of P is at most c , and that four points uniquely defined a sphere that can be computed (center and radius) in constant time.]

SOLUTION. [3 pts | $\sum 6.0$] Prendre une sphère contenant plus de ck points où c est la circularité de P . La remarque est que deux sphères s'intersectent en un cercle, cercle qui ne peut contenir qu'au plus c points de P par définition de c . Soit S un ensemble de $> ck$ points appartenant à une certaine sphère. Si on ne prend pas la sphère couvrant S alors tout autre sphère ne pourra contenir qu'au plus c points. Il en faudra donc

> k sphères pour couvrir S . Il y a que $\binom{n}{4} = O(n^4)$ sphères possibles à tester (pour définir une sphère quatre points étant nécessaires et suffisants), ce qui est polynomial et prend un temps en $O(n^5)$ pour tester toutes les sphères et points de P . Et avec k sphères couvrant chacune au plus ck points, on ne couvre que ck^2 points au maximum. Donc le noyau est de taille ck^2 .

END.