University Côte d'Azur | Graphs
Master 1 : Informatique et Interactions | April-June 2024

Graph Algorithms :

Homework : June 4th, 2024

You can answer in french or english. All answers must be formally explained. All sections are independent.

1 Vertex Cover in bipartite graphs

A **matching** $M \subseteq E$ in a graph $G = (V, E)$ is a set of pairwise disjoint edges, i.e., for all $e, f \in M$, $e \cap f = \emptyset$. A path $P = (v_1, \cdots, v_\ell)$ in G is M**-alternating** if, for every $1 < i < \ell$, $\{v_{i-1}, v_i\} \in M$ or $\{v_i, v_{i+1}\} \in M$. The path P is M**-augmenting** if it is M-alternating and if v_1 and v_ℓ are nor **covered** by M (they are not ends of some edge in M). A **vertex cover** $Q \subseteq V$ is a set of vertices that "touch" all edges, i.e., for every $e \in E$, $e \cap Q \neq \emptyset$.

Question 1 *State the Berge Theorem.*

Question 2 *Let $G = (V, E)$ be a graph, $M \subseteq E$ be a matching of G and let $Q \subseteq V$ be a vertex cover of G. Show that $|M| \leq |Q|$.*

A set of vertices $I \subseteq V$ is a **stable set** of $G = (V, E)$ if there are no edges between the vertices in I, i.e., if for all $u, v \in I$, $\{u, v\} \notin E$.

A graph $G = (V = A \cup B, E)$ is **bipartite** if its vertex-set V can be partitioned into two stable sets A and B.

Question 3 *Let $G = (A \cup B, E)$ be a bipartite graph. Show that A is a vertex cover of G.*

A matching $M \subseteq E$ of a graph $G = (V, E)$ **saturates** a set of vertices $W \subseteq V$ if every vertex in W is the end of some edge in M (M "touches" all vertices of W), i.e., $\forall v \in W$, $\exists e \in M$ such that $v \in e$.

Question 4 *Let $G = (A \cup B, E)$ be a bipartite graph with a matching M that saturates A. Show that A is a vertex cover of G with minimum cardinality.*

Hint : Show that $|M| = |A|$.

In questions 5 to 9, the following notation is used.

Let $G = (A \cup B, E)$ be a bipartite graph and $M \subseteq E$ be a maximum matching of G. Let us assume that M does not saturate A and let $U \subseteq A$ be the set of the vertices of A not covered by M. Let $X \subseteq V$ be the vertices linked to the vertices of U by a M-alternating path (i.e., X is the set of vertices w such that there exists a M-alternating path from w to a vertex of U). Let $A' = X \cap A$ and $B' = X \cap B$. Note that $U \subseteq A'$. An example is illustrated in Figure 1.

Question 5 *Show that $N(A') = B'$.*

Recall that $N(A')$ is the set of neighbours of the vertices in A'.

Question 6 *Show that B' is saturated by M (i.e., the vertices of B' are covered by M : $\forall v \in B'$, $\exists e \in M$ such that $v \in e$)* *Hint : by contradiction, using Berge's theorem.*

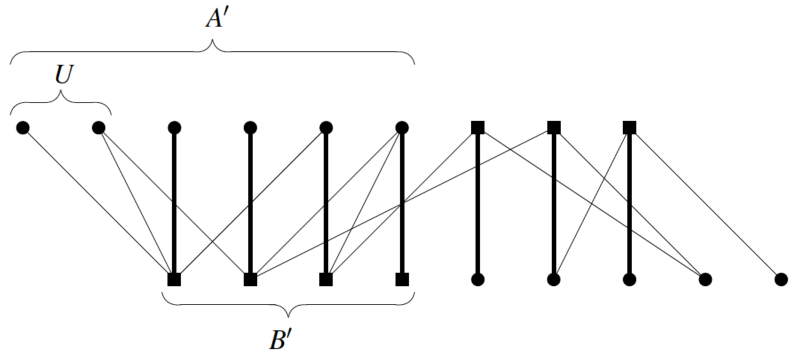

FIGURE 1 – Example of a bipartite graph $G = (A \cup B, E)$ and a maximum matching M illustrating the notation used in questions 5 to 9. The set A corresponds to the top vertices, and the set B corresponds to the bottom vertices. The edges in bold are the ones of the matching M.

In questions 7 to 9, let us set $Y = B' \cup (A \setminus A')$.

Question 7 *Show that Y is a vertex cover of G.*

Hint : for every $e \in E$, show that $e \cap Y \neq \emptyset$.

Question 8 *Show that $|Y| = |M|$.*

Question 9 *Show that Y is a vertex cover with minimum cardinality.*

Question 10 *Show from previous questions and results seen in the lecture that a minimum size vertex cover can be computed in polynomial-time in the class of bipartite graphs.*

Question 11 *Give a minimum vertex cover of the graph in Figure 1.*

2 Maximum Cut Problem

Question 12 *Let $c \geq 1$. Define a c-approximation algorithm for a minimization problem.*

Let $G = (V, E)$ be a graph. A *cut* in G is a partition of V into two sets. Let $S \subseteq V$ be a subset of vertices. The *cost of the cut* $(S, V \setminus S)$, denoted by *cost*(S), equals the number of edges between S and $V \setminus S$, i.e., the size of the set $\{\{x, y\} \in E \mid x \in S, y \in V \setminus S\}$.

The Maximum Cut Problem takes a graph $G = (V, E)$ as input and the objective is to find a cut with maximum cost.

Question 13 *Let $G = (A \cup B, E)$ be a bipartite graph (i.e., A and B are stable sets). Give a maximum cut of G. What is its cost ? (prove that the solution is optimal)*

Question 14 *Give an exponential-time algorithm that computes a maximum cut in arbitrary graphs. Prove that its time-complexity is $O(m \cdot 2^n)$.*

The Maximum Cut Problem is NP-hard, meaning that it does not admit a polynomial-time algorithm unless $P = NP$. The goal of next questions is to analyze an approximation algorithm for it.

Definition : Let $(S, V \setminus S)$ be a cut. A vertex v is *movable* if

— either $v \in S$ and $cost(S) < cost(S \setminus \{v\})$;
— or $v \in V \setminus S$ and $cost(S) < cost(S \cup \{v\})$.

That is, v is movable if moving v on the other side strictly increases the cost of the cut.

Algorithm 1 2-approximation algorithm for MAXIMUM CUT

Require: A graph $G = (V, E)$
Ensure: A cut $(S, V \setminus S)$
1: $S = \emptyset$
2: **while** There is a movable vertex v **do**
3: Move the vertex v on the other side, that is :
— if $v \in S$ then replace S by $S \setminus \{v\}$
— if $v \in V \setminus S$ then replace S by $S \cup \{v\}$
4: **return** S

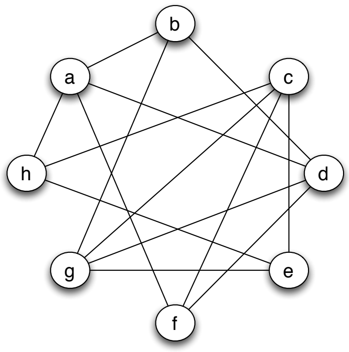

FIGURE 2 – A graph with 8 nodes and 14 edges

Question 15 *To simplify, only in this question, we assume that the Line 1 of Algorithm 1 is replaced by $S = \{g, d\}$. Apply Algorithm 1 on the example depicted in Figure 2.*

Question 16 *What is the maximum number of iterations of the While loop of Algorithm 1 ?*

Let us assume that checking if a vertex is movable can be done in constant time. What is the order of magnitude of the time-complexity of Algorithm 1 ?

Notation : Let $G = (V, E)$ be a graph, $v \in V$ and $X \subseteq V$. Let $deg_X(v)$ denote the degree of v in X, that is the size of the set $\{w \in X \mid \{v, w\} \in E\}$. Let $d(v)$ denote the (classical) degree of v, i.e., $d(v) = deg_V(v)$.

Question 17 *Let $(S, V \setminus S)$ be a solution computed by Algorithm 1.*

1. *Let $v \in S$. Show that $deg_S(v) \leq \lfloor d(v)/2 \rfloor$.*
2. *Similarly, show that $deg_{V \setminus S}(v) \leq \lfloor d(v)/2 \rfloor$ for any $v \in V \setminus S$.*

Question 18 *Let $(S, V \setminus S)$ be a solution computed by Algorithm 1. Let $X = \{\{u, v\} \in E \mid u, v \in S\}$ be the set of edges between nodes in S. Let $Y = \{\{x, y\} \in E \mid x, y \notin S\}$ be the set of edges between nodes in $V \setminus S$. Let $Z = E \setminus (X \cup Y)$ be the set of edges between S and $V \setminus S$.*

1. *By summing the degree of the nodes in S, and using previous question, show that $2|X| \leq |Z|$.*
2. *Similarly, show that $2|Y| \leq |Z|$.*
3. *Deduce that $|Z| \geq |E|/2$.*

Question 19 *Prove that Algorithm 1 is a 2-approximation algorithm for the* Maximum Cut *problem.*

3 Solving a "real life" problem using graphs

Puzzle : We have several pieces (an example is depicted in Figure 3) and the goal is to align all of them in order to create a "wave" (an example of solution is given in Figure 4). Two pieces can be adjacent if they share a vertical side of same height. Note also that any piece x may be swapped (to obtain a symmetrical piece $\bar{x}$ along the vertical axis) [1].

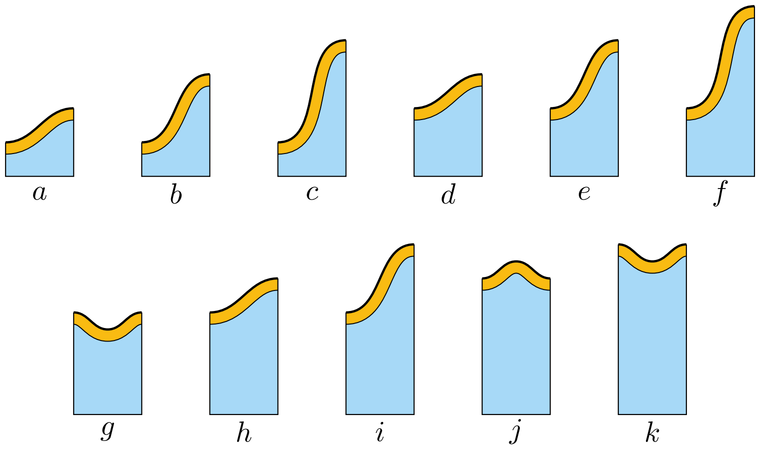

Figure 3 – Example of pieces of the puzzle.

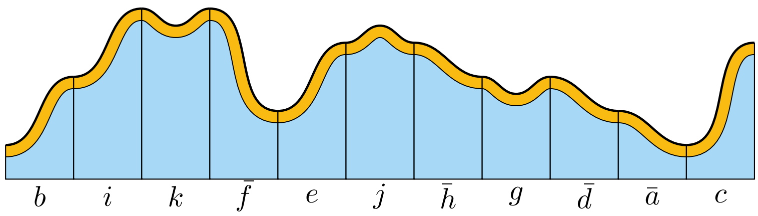

Figure 4 – Example of a solution for the pieces given in Figure 3.

Note that each piece is only defined by the heights of its two vertical sides. For instance, the piece a in Fig. 3 can be denoted by $a = (1, 2)$ since one of its sides has height 1 and the other 2.

1. Such a puzzle was a test in Koh-Lanta (French version of Survivor) few years ago.

The general problem. Given a set of n pieces $P = \{p_i = (a_i, b_i)\}_{1 \le i \le n}$, is there a solution (i.e., a wave aligning the n pieces and respecting the rules) ? If yes, how to find a solution ?

To solve this problem, let us model it by the following graph $G(P)$. The vertices correspond to the heights of the sides of the pieces. That is, there is a vertex labeled x if there is a piece p_i such that $x = a_i$ or $x = b_i$. Moreover, there is an edge between vertices x and y if there is a piece (x, y) (i.e., a piece with one side of height x and the other side of height y). If several pieces (x, y) are similar (i.e., each of these pieces has one side of height x and one side of height y), there as many edges between the vertices x and y (i.e., parallel edges are allowed). Also, each piece (x, x) (i.e., with its two vertical sides of same height x) corresponds to a self-loop in the vertex x. An example of the graph $G(P)$ is depicted in Figure 5.

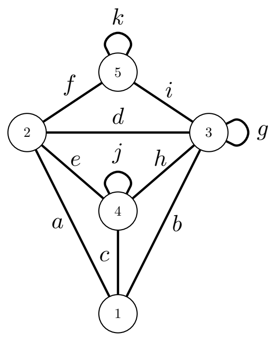

FIGURE 5 – Example of the graph $G(P)$ for the set P of pieces given in Figure 3. Note that the sides of the pieces of this example have exactly 5 different heights that we denote by $1, 2, 3, 4, 5$ and that correspond to the five vertices of $G(P)$. For instance, the piece $a = (1, 2)$ corresponds to the edge between vertices 1 and 2.

Question 20 *Explain how to represent the solution given in Figure 4 in the graph $G(P)$ given in Figure 5.*

Question 21 *Design an algorithm that, given a set of pieces $P = \{p_i = (a_i, b_i)\}_{1 \le i \le n}$, either computes a solution or certifies that no solution exists. Explain your answer (at most* 10 *lines).*