Oliver Tonnesen
V00885732

Project 3

MATH 423
October 31, 2019

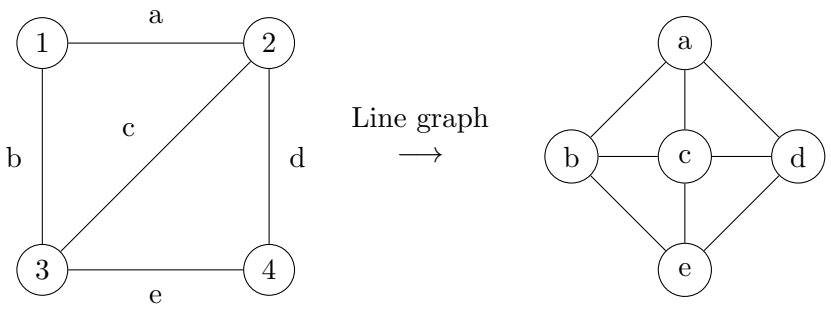
1 Definition and characterizations

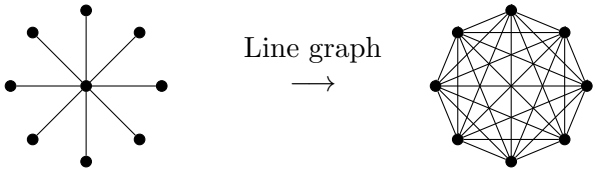
1.1 Definition of a line graph

A line graph of a simple graph G, $L(G)$, is the graph obtained by taking $E(G)$ to be its vertex set, and connecting two vertices with an edge if the corresponding edges in G share a common vertex.

Some definitions we use throughout this survey, assume H is a suspected line graph and G is its root graph if it exists:
g-clique: If $g \in G$, a g-clique in H is a clique in H, corresponding to the star graph centred at g in G.
Half-named: A vertex in H is "half-named" when it corresponds to an edge of G, one of whose vertices belongs to a fully discovered g-clique in H.
Fully named: A vertex in H is "fully named" when it corresponds to an edge of G, both of whose vertices belong to a fully discovered g-clique in H.
Even triangle: A triangle such that every vertex in the graph is adjacent to exactly two or zero vertices in the triangle.
Odd triangle: A triangle such that every vertex in the graph is adjacent to exactly one or three vertices in the triangle.
Cross vertex: Given two vertices in H $x - y$ and $y - z$, a cross vertex is one of the form the name $x - z$.

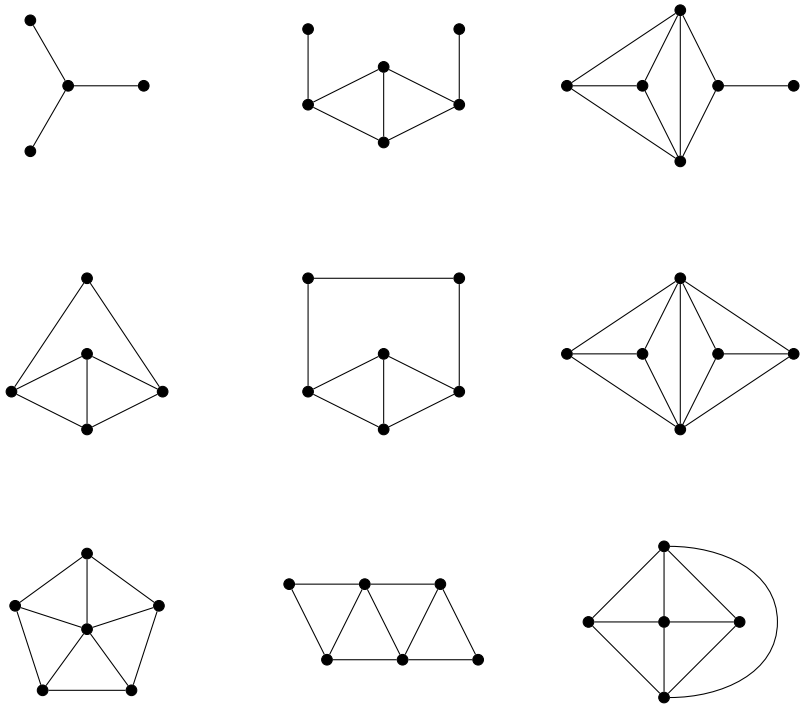
1.2 Examples





1.3 Characterizations of line graphs by forbidden structures and by forbidden subgraphs

A graph is the line graph of some simple graph if and only if it contains none of these nine graphs as induced subgraphs [2]:



Another characterization [2]: A graph is a line graph of some simple graph if and ly if it is claw-free and no induced diamond subgraph of G has two odd triangles.

2 Recognitions

2.1 Lehot algorithm for constructing the root graph of a graph if it exists

We're given a graph $G = (V(G), E(G)$. The algorithm described in [1] has 9 steps:

Step 1: Select two adjacent vertices in $V(G)$, and name them $1-2$ and $2-3$. Call these vertices the "basic vertices."

Step 2: Find all vertices adjacent to both basic vertices. If there are at least three such vertices, go to step 5. If ther are none, go to step 6. If there is one, continue to step 3. If there are two, go to step 4.

Step 3: Call the one vertex x. If the triangle $(x, 1-2, 2-3)$ is odd, call x $2-4$ and go to step 6. Otherwise call x $1-3$ and go to step 7.

Step 4: Call the two vertices x and y. If x y, then $(x, 1-2, 2-3)$ and $(y, 1-2, 2-3)$ are two distinct triangles. If one triangle, say $(x, 1-2, 2-3)$ is odd, then call x $2-4$, call y $1-3$ (note that this makes it a cross vertex) dnd go to step 6. Otherwise both triangles are even, and the only three possibilities are the following:

Step 5: There are at least three vertices adjacent to both basic vertices. Find two that are nonadjacent, a and b. We check if a is adjacent to a third vertex in this set. If it is, then we call b the cross vertex. If not, then we call a the cross vertex. If there are no such a and b, continue to step 6, otherwise go to step 7.

Step 6: All the vertices in the g-clique are named, and we use them to half-name all the vertices adjacent to them, then go to step 8.

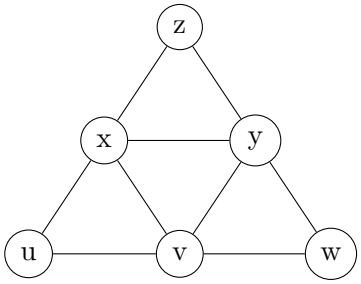
Step 7: All the vertices in the g-clique are named, and the cross vertex is fully named. We use the vertices of the g-clique to half-name the vertices adjacent to them, then go to step 8.

Step 8: If there remain no more vertices that are not fully named, continue to step 9. Otherwise choose any half-named vertex. By definition, it is adjacent to a fully-named vertex. If this vertex does not belong to an already

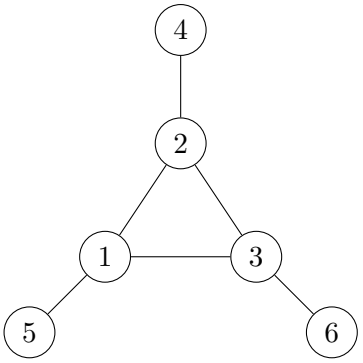
discovered g-clique, then there must be a fully-named vertex which does belong to an already discovered clique, and which is defined as the cross vertex of the two vertices. These two vertices fully name the previously-chosen half-named vertex. Use the vertices of this clique to half-name all the adjacent vertices with the number in their name which is not the clique number (g), then go back to step 8.

Step 9: Compare $L(G)$ to H. If they are isomorphic, then H is a line graph, otherwise it is not.

2.2 Apply Lehot algorithm to construct the root graph of the graph below



Starting at step 1 of the algorithm, we choose x and y as our basic vertices, and name $x = 1-2$ and $y = 2-3$. Continuing to step 2, we find two vertices adjacent to both basic vertices: z and v. There are two such verties, so we continue to step 4. z and v are not adjacent, so they make two distinct triangles. The triangle (x, y, z) is odd, so we call $z = 2 - 4$, and $v = 1 - 3$. We now skip to step 6, and use our fully named $1 - clique$ to successively half-name all vertices adjacent to it: $w = 3 - 6$, and $u = 1 - 5$. Now at step 8, we choose a half-named vertex, u. It is adjacent to a fully named vertex, x. x and v fully name u. We continue this for the remaining vertices, and we obtain the graph below:



It is simple to check that the line graph of the graph below is isomorphic to the original graph, and so the graph we were given was indeed a line graph, and the graph we constructed is its root graph.

3 Optimization

3.1 Find a maximum clique in a line graph

Given a graph $L(G)$, finding a vertex of degree $\Delta(G) \neq 3$ in G corresponds to finding a maximum matching in $L(G)$. Since this algorithm runs in linear time, if $L(G)$ is indeed the line graph of some simple graph G, then we can find a maximum clique – normally a very computationally hard problem – efficiently by transforming one from a degree $\Delta(G)$ vertex in G.

3.2 Find a maximum independent set in a line graph

Given a graph $L(G)$, finding a maximum matching in G corresponds to finding a maximum independent set in $L(G)$. Since this algorithm runs in linear time, if $L(G)$ is indeed the line graph of some simple graph G, then we can find a maximum independent set – normally a very computationally hard problem – efficiently by transforming one from a maximum matching in G.

References

[1] P Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM*, 21:569–575, 1974.

[2] A van Rooij and H Wilf. The interchange graphs of a finite graph. *Acta Mathematica Hungarica*, 16:263–269, 1974.