Trees with many leaves

N. V. Gravin

Abstract

In this note we consider the problem of finding a spanning tree with large number of leaves. We overview some known facts about this question. In details We consider Storer's algorithm of finding a spanning tree with many leaves for cubic graphs and make sure that such problem of finding a tree with maximal possible number of leaves is NP-complete.

Contents

1 Introduction

We are given a graph and we want to find a spanning tree in this graph. Moreover we want to pick out the most “nontrivial” tree. And a good criteria for such nontriviality is the number of leaves in the tree.

The first natural question arising in this connection is to find a spanning tree with maximal possible number of leaves but this problem was shown to be Np-complete [1] by Paul Lemke in 1988. Also in the same period of time there emerged questions of the lower bounds for number of leaves in spanning trees and how to find such trees. And in 1988 Linial posed the following conjecture:

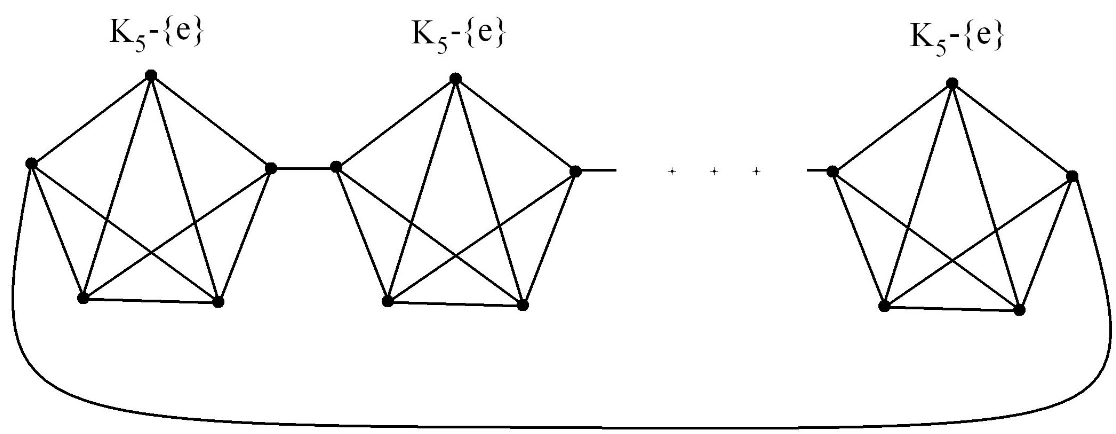
Notations.
$\delta(G)$:= Minimum degree of the graph G.
$L(T)$:= Number of leaves in the tree T
$L(G)$:= Maximal number of leaves over all spanning trees of the graph G.

Conjecture (Linial). *Let G be a graph on N vertices with $\delta(G) = k$. Then*

$$L(G) \geq \frac{k-2}{k+1}N + c_k$$

where c_k depends only on k.

It is easy to see the tightness of the bound in Linial's conjecture.



A series of examples for $k = 4$.
This “necklace” example extends to another values of k.

At present there exist some following results concerning Linial's conjecture.

- $\delta(G) = 3$ Linial's conjecture holds [3] (Storer 1981)
- $\delta(G) = 4$ Linial's conjecture holds [2] (Jerrold, R. Griggs, Mingshen Wu 1992)
- $\delta(G) = 5$ Linial's conjecture holds [2] (Jerrold, R. Griggs, Mingshen Wu 1992)
- $\delta(G) \geq 6$ open problem
- $\delta(G) \to \infty$ Linial's conjecture fails. (N. Alon 1990)

There are series of graphs G_k with $\delta(G_k) = k$ such

$$L(G_k) \leq \left(1 - \frac{\log(k)}{k+1}\right) |V(G_k)| \left(1 + \frac{O(1)}{k+1}\right)$$

2 Strorer's Algorithm

Definition 1. Cubic is a graph where all vertices have degree equal to three.

In this section we suppose G be a cubic connected graph on N vertices. And we want to pick out a spanning tree with at least $\lfloor \frac{1}{4}N \rfloor + 2$ leaves according to Linial's conjecture for $\delta(G) = 3$ and $c_k = 2$.

2.1 Algorithm description

We will construct such spanning tree consequently at each step of the algorithm having a partial tree of G and always seeking to enlarge it in a proper way.

Definition 2. A leaf v of a partial tree T of G called **dead** if and only if v has no adjacent to it vertices outside T. And by $D(T)$ we denote the number of dead leaves in the partial tree T of G.

Consider cost function involving the number of leaves, dead leaves and vertices of T

$$f(T) := 3L(T) + D(T) - |V(T)|.$$

So at each step of the algorithm we will seek for such enlargement of partial tree T that $f(T)$ do not decrease.

2.2 Algorithm's proof

Suppose for a while that we always can enlarge our partial tree that $f(T)$ do not decrease. Then starting from some partial tree T_0 at the end we get some spanning tree T_1 of G with $f(T_1) \geq f(T_0)$.

Let us remember that $f(T_1) = 3L(T_1) + D(T_1) - |V(T_1)|$. As T_1 is a spanning tree of G then $V(T_1) = N$ and $D(T_1) = L(T_1)$ because all leaves are dead in a spanning tree. So we get $f(T_1) = 4L(T_1) - N \geq f(T_0)$.

Let us take as T_0 a vertex and all its neighbors. Then we get $f(T_0) \geq 3 * 3 + 0 - 4$, so $f(T_0) \geq 5$. Then $4L(T_1) \geq N + 5$ and we are done.

So the only thing we have to prove is why we always are able to enlarge partial tree T and do not decrease $f(T)$.

Let us suppose contrary to this claim. Then the following assertions hold:

1. There are no non leaf vertices of T adjacent to vertex outside T as in other case we can enlarge T and do not decrease $f(T)$.
2. There are no leaf vertices of T adjacent to at least two vertices outside T.
3. There are no leaves of T adjacent to an outside vertex with two neighbors also outside T.

Then every vertex outside T either has not edges into T, or has at least two neighbors in T in other case we get contradiction with the third assertion. As T is not jet a whole spanning tree and G is connected then there is a vertex v outside T adjacent to at least two vertices in T. By the first assertion all neighbors of v are leaves in T and by the second assertion both of this two neighbors have only v as a neighbors outside T. So adding v to T we increase number of dead leaves and it means that we do not decrease $f(T)$. Then we arrive at a contradiction and we are done.

3 NP-completeness

Theorem (Lemke). *A maximum leaf spanning tree problem for cubic graphs is NP-complete.*

Proof. So we consider the following decision problem associated with the maximum leaf spanning tree problem:

INSTANCE: A cubic graph G and an integer number k.
QUESTION: Does G posses a spanning tree with at least k leaves?

Instead of this decision problem let us consider more specific decision problem:

INSTANCE: A cubic graph G.
QUESTION: Does G posses a spanning tree with at least $\frac{|V(G)|}{2}+1$ leaves?

The last question has an equivalent reformulation for cubic graphs:

EQUIVALENT QUESTION: Does G posses a spanning tree with no vertices of degree two?

Remark 1. Indeed why this is equivalent reformulation?

Proof.

Notations.
$a_1 :=$ number of vertices in spanning tree T with degree 1.
$a_2 :=$ number of vertices in spanning tree T with degree 2.
$a_3 :=$ number of vertices in spanning tree T with degree 3.
$N :=$ number of vertices in T.

We know that $a_1+a_2+a_3=N$ as T is a spanning tree and contains all the vertices of G. Also we know that $a_1+2a_2+3a_3=2(N-1)$ as every tree on N vertices contains exactly $N-1$ edges. So subtract the second equality from the doubled first one we get $a_1-a_3=2$.

Suppose now T has at least $\frac{|V(G)|}{2}+1$ leaves. It means $a_1 \geq \frac{|N|}{2}+1$ and so by the third equality $a_3 \geq \frac{|N|}{2}-1$. And as $a_1+a_2+a_3=N$ we get $a_2=0$. It means that T has no vertices of degree two.

In the other direction. Let T has no vertices of degree two then $a_2=0$ and so $3a_1+3a_3-a_1-3a_3=3N-2(N-1)$. And we get $a_1=\frac{N}{2}+1$. □

Let us return to the proof of theorem. The proof will be by reduction of known NP-complete problem EXACT COVER BY 3-SETS [4] to ours.

The EXACT COVER BY 3-SETS is as follows:

INSTANCE: Positive integers n and m, subsets $S_1, S_2, ..., S_m$ of $\{1,2,....,n\}$, with $|S_i|=3$ for all $i \in \{1,2,...,m\}$.

QUESTION: Is there a subset $Q \subseteq \{1,2,...,m\}$ such that $\bigcup_{i\in Q} S_i = \{1,2,...,n\}$ and $\forall i_1, i_2 \in Q, i_1 \neq i_2 \Rightarrow S_{i_1} \cap S_{i_2} = \emptyset$?

Given an instance of EXACT COVER BY 3-SETS we construct a graph G as follows:

Define the numbers $a_0, a_1, ..., a_n$ by:

$a_0 = 2m$
$a_j = |\{i | j \in S_i\}|$ for $j = 1, 2, ..., n$.

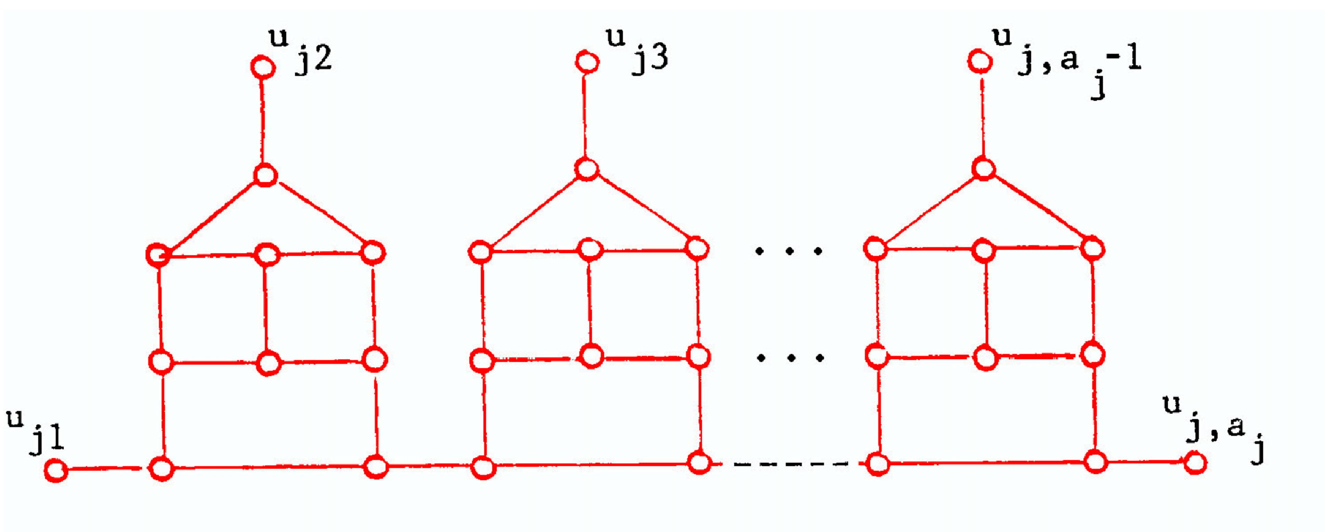
For the construction given below we need $a_j \geq 3 \quad \forall j$. If this is not the case, we can always make it so by adding duplicate sets to $\{S_1, S_2, ..., S_m\}$.

The vertex set V of G is defined to be the union the pairwise disjoint sets $U_0, U_1, ..., U_n, W$, and X, where:

$|U_j| = 10a_j - 18$ for $j = 0, 1, ..., n$
$W = \{w_1, w_2, ..., w_m\}$
$X = \{x_1, x_2, ..., x_m\}$.

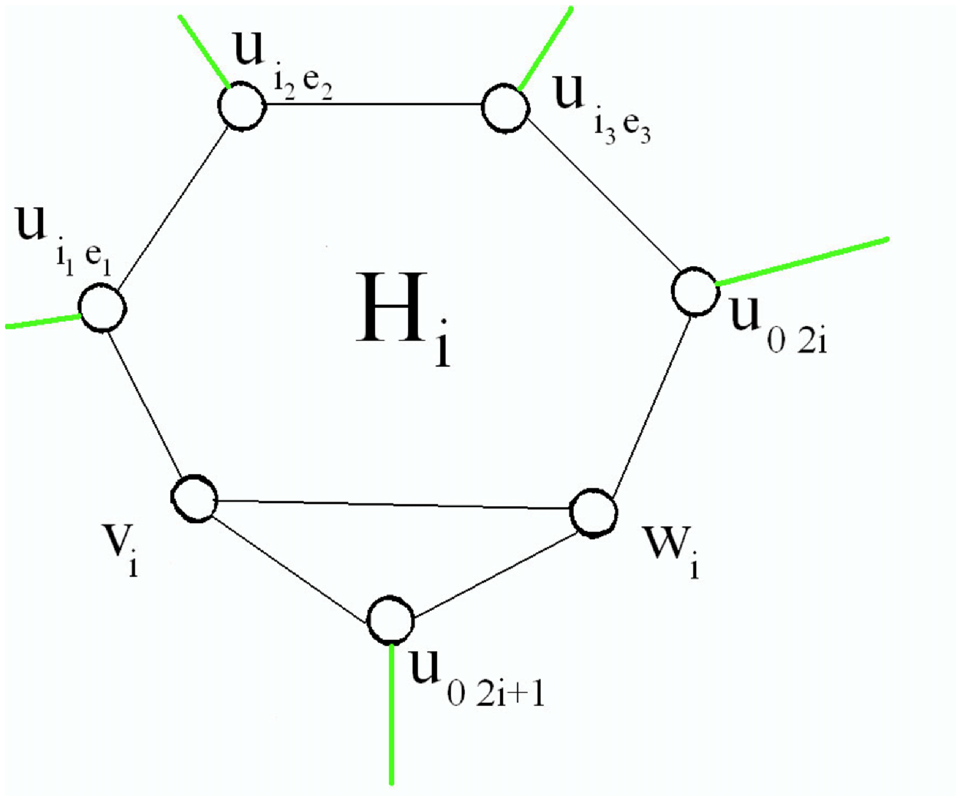
Now we have to describe the edge set E of G. It will be a union of edge sets of some graphs with vertices taken in some subsets of $V(G)$. And now we describe this graphs. They consist of two series of graphs $G_j, j = 0, 1, ..., n$ and $H_i, i = 1, 2, ..., m$.

We take G_j with the U_j vertex set as shown on the figure ($j = 0, 1, ..., n$)



So every G_j has exactly a_j vertices of degree one and all remaining vertices have degree three. By definition of a_j we can put in one to one correspondence $u_{jk}, k \in \{1, 2, ..., a_j\}$ where $j = 1, 2, ..., n$ with inclusion of j-th element into some 3-set S_i. Let us fix this correspondence.

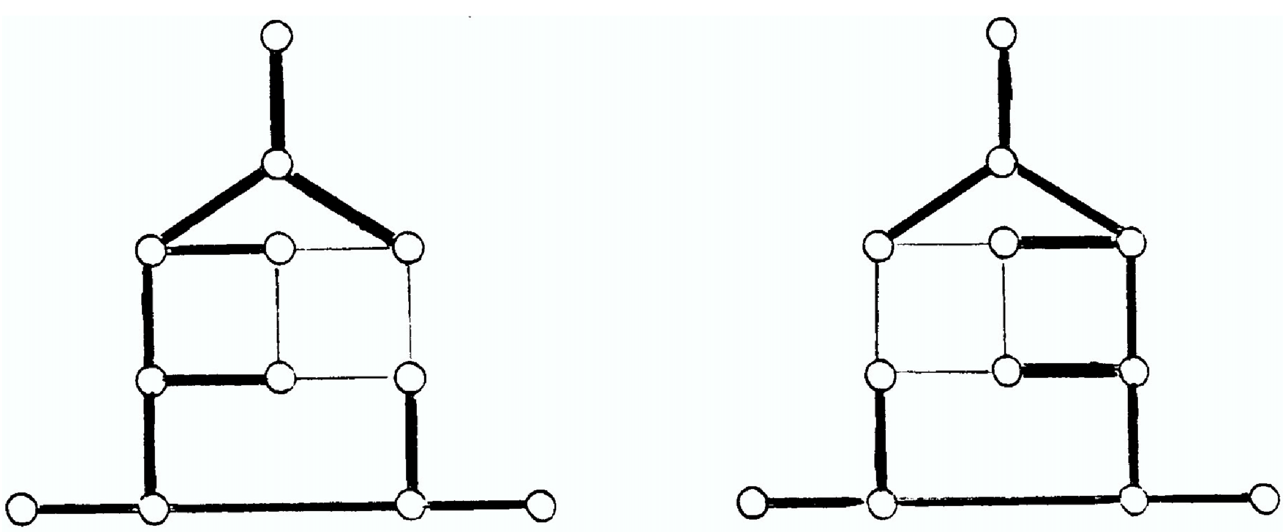
We take H_i corresponding to $S_i = \{i_1, i_2, i_3\}$ on the $\{u_{i_1,e_1}, u_{i_2,e_2}, u_{i_3,e_3}, u_{0,2i}, u_{0,2i+1}, w_i, x_i\}$ set of vertices, where u_{i_1,e_1} corresponds to the inclusion of i_1 element into S_i, u_{i_2,e_2}, u_{i_3,e_3} correspond to the inclusion of i_2 and i_3 elements into S_i, $u_{0,2i}, u_{0,2i+1} \in U_0$ and $w_i \in W, x_i \in X$. The edges we draw for H_i as depicted on the following figure.



Green edges do not belongs to H_i but belongs to corresponding graphs of type G_j.

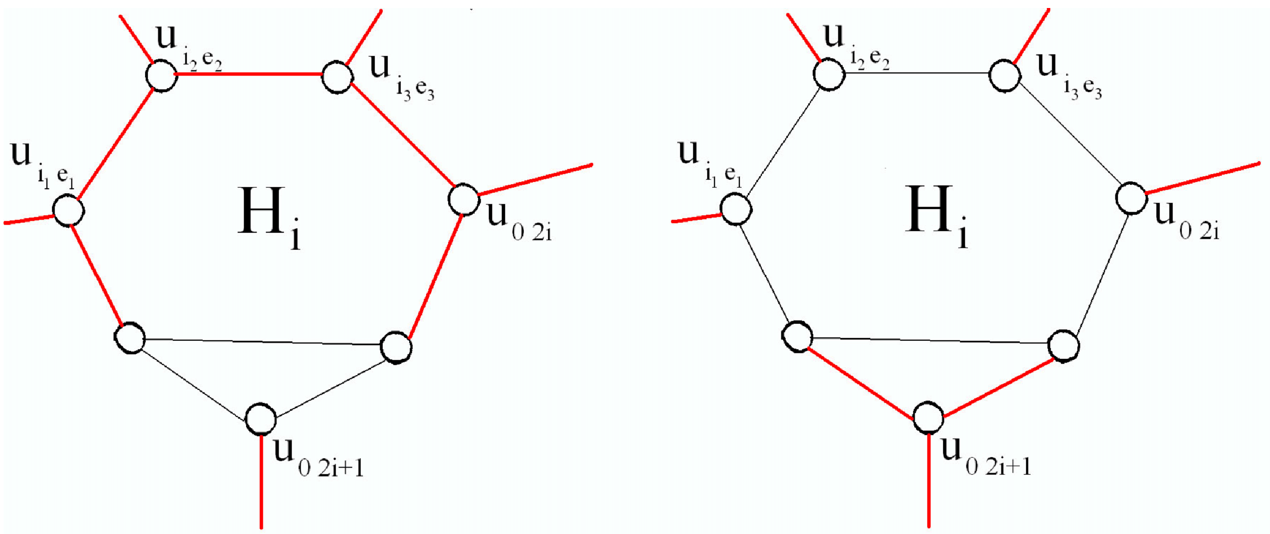
So now we get a cubic graph. We are only to show why the constructed graph has a spanning tree without vertices of degree two if and only if the corresponding problem of EXACT COVER BY 3-SETS has positive answer. Let us just do it for "only if" part of the statement because for "if" part it will be easy to construct such a spanning tree relying on reasoning of "only if" part.

If we consider the unit block on nine vertices of G_j then there are only two possibilities to assign edges so that nine interior vertices has odd degree (see the following figure).



So every G_j should be a connected graph with all edges coming to vertices of degree one in G_j assigned into the spanning tree.

For H_i there are also only two possibilities to assign edges so that seven interior vertices have odd degree (see the following figure).



In the first case we join $G_0, G_{i_1}, G_{i_2}, G_{i_3}$ into one component of connectivity in the second case we make no connection between this four components. So in the first case we take S_i into covering set, and in the second case we does not take it. Then we get EXACT COVERING BY 3-SETS we are needed in as in the other case we have joined G_0 and G_i twice or we have had disconnected spanning tree.

□

References

[1] P. Lemke, *The maximum leaf spanning tree problem for cubic graphs is NP-complete*, preprint, The Institute for Mathematics and its Applications, 1988

[2] Jerrold, R. Griggs, Mingshen Wu. *Spanning trees in graphs of minimum degree 4 or 5*, Discreet Mathematics 104 (1992) p. 167–183.

[3] J. A. Storer, *Constructing full spanning trees for cubic graphs*, Inform. Process. Lett. 13 (1981), №1, p. 8-11.

[4] R. M. Karp, *Reducibility among combinatorial problems*, R. E. Miller and J. W. Thatcher (eds.), in *Complexity of Computer Computations*, Plenum Press, New York, 1972, pp. 85-103.