Spanning Trees with many Leaves in Regular Bipartite Graphs

Emanuele G. Fusco and Angelo Monti

Dipartimento di Informatica, Università di Roma “La Sapienza”
via Salaria, 113-00198 Rome, Italy.
{fusco, monti}@di.uniroma1.it

ISAAC 07 December 17–19, 2007
Outline

1. Spanning Trees with many Leaves
2. Bipartite Graphs
3. Our Results
4. Conclusions and Open Problems

E.G. Fusco and A. Monti
Spanning Trees with many Leaves in Regular Bipartite Graphs
Outline

1 Spanning Trees with many Leaves

2 Bipartite Graphs

3 Our Results

4 Conclusions and Open Problems
The problem of finding spanning trees with many leaves has been thoroughly investigated:
Spanning trees with many leaves

The problem of finding spanning trees with many leaves has been thoroughly investigated:

- known to be NP-Hard (see, e.g., [GJ79]);
Spanning trees with many leaves

The problem of finding spanning trees with many leaves has been thoroughly investigated:

- known to be NP-Hard (see, e.g., [GJ79]);
- 3-approximation algorithms by Lu and Ravi [LR98];
The problem of finding spanning trees with many leaves has been thoroughly investigated:

- known to be NP-Hard (see, e.g., [GJ79]);
- 3-approximation algorithms by Lu and Ravi [LR98];
- 2-approximation algorithm by Solis-Oba [SO98];
The problem of finding spanning trees with many leaves has been thoroughly investigated:

- known to be NP-Hard (see, e.g., [GJ79]);
- 3-approximation algorithms by Lu and Ravi [LR98];
- 2-approximation algorithm by Solis-Oba [SO98];
- remains NP hard even if the input is restricted to $d$-regular graphs for any fixed $d \geq 3$ (Lemke [Lem88]);
Spanning trees with many leaves

The problem of finding spanning trees with many leaves has been thoroughly investigated:

- known to be NP-Hard (see, e.g., [GJ79]);
- 3-approximation algorithms by Lu and Ravi [LR98];
- 2-approximation algorithm by Solis-Oba [SO98];
- remains NP hard even if the input is restricted to $d$-regular graphs for any fixed $d \geq 3$ (Lemke [Lem88]);
- 7/4 approximation algorithm for cubic graphs by Lorys and Zwozniak [LZ02].
Outline

1. Spanning Trees with many Leaves
2. Bipartite Graphs
3. Our Results
4. Conclusions and Open Problems
A variation of the problem

We consider a variation of the problem introduced by Rahman and Kaykobad in [RK05].
A variation of the problem

We consider a variation of the problem introduced by Rahman and Kaykobad in [RK05].
We are given a bipartite graph $G$. Nodes in $G$ are black or white depending on the partited set they belong to. We want to find a spanning tree of $G$ with the maximum number of black leaves.
A variation of the problem

We consider a variation of the problem introduced by Rahman and Kaykobad in [RK05].
We are given a bipartite graph $G$. Nodes in $G$ are black or white depending on the partitited set they belong to. We want to find a spanning tree of $G$ with the maximum number of black leaves. This problem is NP hard for planar bipartite graphs (Li and Toulouse [LT06]).
A variation of the problem

We consider a variation of the problem introduced by Rahman and Kaykobad in [RK05].
We are given a bipartite graph $G$. Nodes in $G$ are black or white depending on the partited set they belong to. We want to find a spanning tree of $G$ with the maximum number of black leaves. This problem is NP hard for planar bipartite graphs (Li and Toulouse [LT06]).
We restrict the input to regular bipartite graphs.
Outline

1. Spanning Trees with many Leaves
2. Bipartite Graphs
3. Our Results
4. Conclusions and Open Problems
The problem of finding a spanning tree with the maximum number of black leaves is NP hard for $d$-regular bipartite graphs for any fixed $d \geq 4$. 
NP-hardness

Theorem

The problem of finding a spanning tree with the maximum number of black leaves is NP hard for $d$-regular bipartite graphs for any fixed $d \geq 4$.

Gadgets

Gadget $Ga1$
NP-hardness

Theorem

The problem of finding a spanning tree with the maximum number of black leaves is NP hard for $d$-regular bipartite graphs for any fixed $d \geq 4$.

Gadgets

Gadget $Ga1$

Gadget $Ga2$
NP-hardness

The problem is NP-hard for any fixed \( d \geq 4 \):

Graph construction
Let $\lambda_i$ be the number of black nodes of degree $i$ in a spanning tree $T$ of a $d$-regular bipartite graph with $n$ black nodes. We have that $\sum_{i=1}^{d} \lambda_i = n$. 
Let $\lambda_i$ be the number of black nodes of degree $i$ in a spanning tree $T$ of a $d$-regular bipartite graph with $n$ black nodes. We have that $\sum_{i=1}^{d} \lambda_i = n$. All edges have endpoints in one black node and one white node so we also have $\sum_{i=1}^{d} i \lambda_i = 2n - 1$. 
Let $\lambda_i$ be the number of black nodes of degree $i$ in a spanning tree $T$ of a $d$-regular bipartite graph with $n$ black nodes. We have that $\sum_{i=1}^{d} \lambda_i = n$. All edges have endpoints in one black node and one white node so we also have $\sum_{i=1}^{d} i \lambda_i = 2n - 1$.

**Lemma**

*Let $T$ be a spanning tree of $G_d$, we have*

$$\lambda_1(T) = 1 + \sum_{i=3}^{d} (i - 2) \lambda_i(T)$$
Let $\lambda_i$ be the number of black nodes of degree $i$ in a spanning tree $T$ of a $d$-regular bipartite graph with $n$ black nodes. We have that $\sum_{i=1}^{d} \lambda_i = n$. All edges have endpoints in one black node and one white node so we also have $\sum_{i=1}^{d} i \lambda_i = 2n - 1$.

**Lemma**

Let $T$ be a spanning tree of $G_d$, we have $\lambda_1(T) = 1 + \sum_{i=3}^{d} (i - 2)\lambda_i(T)$

**Lemma**

Let $T$ be a spanning tree of $G_d$, then $\lambda_1(T) \leq \left\lfloor \frac{(d-2)n+1}{d-1} \right\rfloor$. 
Our algorithm

1: \( F \leftarrow \emptyset \)
2: \( i \leftarrow 1 \)
3: repeat
4: Find a black node \( v \) with all neighbors outside \( F \)
5: Build a tree \( T_i \) with root \( v \) and add to \( T_i \) all white neighbors of \( v \)
6: Augment \( T_i \) as long as it is possible to add one black node with at least 2 white neighbors outside \( F \).
7: until All black nodes have a neighbor in \( F \)
8: build \( T_A \) by connecting \( F \) and all the isolated nodes in a tree
9: return \( T_A \)
Every tree $T_i$ in the forest $\mathcal{F}$ has a black node of degree $d$. 
Useful properties

Every tree $T_i$ in the forest $\mathcal{F}$ has a black node of degree $d$. Every black node in $\mathcal{F}$ has degree at least 3.
Useful properties

Every tree $T_i$ in the forest $\mathcal{F}$ has a black node of degree $d$. Every black node in $\mathcal{F}$ has degree at least 3. Every isolated black node has $d - 1$ neighbors in a tree $T_i$. 
Useful properties

Every tree $T_i$ in the forest $\mathcal{F}$ has a black node of degree $d$. Every black node in $\mathcal{F}$ has degree at least 3. Every isolated black node has $d - 1$ neighbors in a tree $T_i$. 
Approximation ratio

Lemma

For any $G_d$ with $d \geq 4$ there exists a spanning tree $T_A$ such that

$$\lambda_1(T_A) \geq \left\lfloor \frac{d - 1}{2d} n + \frac{(d - 1)^2}{2d} \right\rfloor$$
Approximation ratio

Lemma

For any $G_d$ with $d \geq 4$ there exists a spanning tree $T_A$ such that

$$\lambda_1(T_A) \geq \left\lceil \frac{d - 1}{2d} n + \frac{(d - 1)^2}{2d} \right\rceil$$

Theorem

The problem of finding a Spanning Tree with the maximum number of black leaves for a $d$-regular bipartite graph $G_d$, $d \geq 4$, can be approximated by an algorithm running in linear time with approximation ratio $\leq 2 - 2/(d - 1)^2$. 

E.G. Fusco and A. Monti Spanning Trees with many Leaves in Regular Bipartite Graphs
Cubic bipartite graphs

Our algorithms only achieves approximation ratio 2 on cubic bipartite graphs:
Local optimization step

Such a tree $T_i$ is not allowed to be added to the forest $\mathcal{F}$:
Approximation ratio on cubic bipartite graphs

Lemma

For any $G_3$ there exists a spanning tree $T_A$ such that

$$\lambda_1(T_A) \geq \left\lceil \frac{n}{3} \right\rceil + 1.$$
Lemma

For any $G_3$ there exists a spanning tree $T_A$ such that
\[ \lambda_1(T_A) \geq \left\lceil \frac{n}{3} \right\rceil + 1. \]

Theorem

The problem of finding a Spanning Tree with the maximum number of black leaves for a cubic bipartite graph $G_3$ can be approximated by an algorithm running in linear time with approximation ratio $\leq 1.5$. 
Tight analysis
Outline

1. Spanning Trees with many Leaves
2. Bipartite Graphs
3. Our Results
4. Conclusions and Open Problems
If we know the minimum degree $d$ of an $n$-nodes connected graph $G$, what can we say about its Spanning Trees? More formally:
If we know the minimum degree $d$ of an $n$-nodes connected graph $G$, what can we say about its Spanning Trees? More formally:

$l(n, d)$

Let $l(n, d)$ be the maximum integer $m$ such that every connected $n$-nodes graph with minimum degree at least $d$ has a spanning tree with at least $m$ leaves.
If we know the minimum degree $d$ of an $n$-nodes connected graph $G$, what can we say about its Spanning Trees? More formally:

$$l(n, d)$$

Let $l(n, d)$ be the maximum integer $m$ such that every connected $n$-nodes graph with minimum degree at least $d$ has a spanning tree with at least $m$ leaves.

For $d \leq 5$ we have that:

$$l(n, d) = \frac{d - 2}{d + 1} n + c_d$$
For $d \geq 6$ the exact value $l(n,d)$ is unknown.
For $d \geq 6$ the exact value $l(n, d)$ is unknown. Alon proved by probabilistic methods that for a large enough $d$:

$$l(n, d) \leq (1 + o(1)) \frac{d - \ln(d + 1)}{d + 1} n$$

(See [CWY00] for more information on this topic.)
On regular bipartite graphs

Let $l_B(n, d)$ to be the maximum $m$ such that every $G_d$ with $n$ black nodes has a spanning tree with at least $m$ black leaves.

Structural properties for regular bipartite graphs

- $l_B(n, 2) = 1$;
On regular bipartite graphs

Let $l_B(n, d)$ to be the maximum $m$ such that every $G_d$ with $n$ black nodes has a spanning tree with at least $m$ black leaves.

**Structural properties for regular bipartite graphs**

- $l_B(n, 2) = 1$;
- $l_B(n, 3) = \lceil \frac{n}{3} \rceil + 1$;
Necklace
On regular bipartite graphs

Let $l_B(n, d)$ to be the maximum $m$ such that every $G_d$ with $n$ black nodes has a spanning tree with at least $m$ black leaves.

Structural properties for regular bipartite graphs

- $l_B(n, 2) = 1$;
- $l_B(n, 3) = \left\lceil \frac{n}{3} \right\rceil + 1$;
- $\left\lceil \frac{d-1}{2d} n + \frac{(d-1)^2}{2d} \right\rceil \leq l_B(n, d) \leq \left\lceil \frac{d-2}{d} n \right\rceil + 1$. 

Open Problems

Can we sharpen the upper and lower bounds for $l_B(n, d)$ for any $d \geq 4$? Can we give exact values?
Open Problems

- Can we sharpen the upper and lower bounds for $l_B(n, d)$ for any $d \geq 4$? Can we give exact values?
- Is it the problem NP hard for cubic bipartite graphs? (Our contribution for this class of graphs is a linear time algorithm with approximation ratio 1.5.)
Open Problems

- Can we sharpen the upper and lower bounds for $l_B(n, d)$ for any $d \geq 4$? Can we give exact values?
- Is it the problem NP hard for cubic bipartite graphs? (Our contribution for this class of graphs is a linear time algorithm with approximation ratio 1.5.)
- Is it the problem NP hard for $d$-regular planar graphs?
Thank you for your attention!
Connected domination and spanning trees with many leaves.

*Computers and Intractability; A Guide to the Theory of NP-Completeness*.

The maximum leaf spanning tree problem for cubic graphs is np-complete.

Approximating maximum leaf spanning trees in almost linear time.

Variations of the maximum leaf spanning tree problem for bipartite graphs.


Approximation algorithm for the maximum leaf spanning tree problem for cubic graphs.

Complexities of some interesting problems on spanning trees.

2-approximation algorithm for finding a spanning tree with maximum number of leaves.