Exact and Parameterized Algorithms for MAX INTERNAL SPANNING TREE

Henning Fernau[1] *Serge Gaspers*[2] Daniel Raible[1]

[1]University of Trier, Germany
[2]LIRMM – Université de Montpellier 2, CNRS, France

WG 2009

Outline

1. Introduction
   - Problem Definition
   - Previous Results
   - Our Results
   - New Results
2. Algorithm for graphs of max degree 3
   - Observations
   - Outline of Algorithm
   - Simplification Rules
   - Measure
   - Branching
   - Result for cubic graphs
3. Conclusion

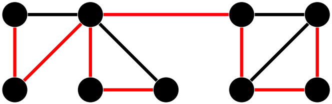
Outline

1. Introduction
   - Problem Definition
   - Previous Results
   - Our Results
   - New Results
2. Algorithm for graphs of max degree 3
   - Observations
   - Outline of Algorithm
   - Simplification Rules
   - Measure
   - Branching
   - Result for cubic graphs
3. Conclusion

Problem Definition

MAXIMUM INTERNAL SPANNING TREE (MIST)

- Input: A graph $G = (V, E)$ on n vertices, an integer k.
- Question: Does G have a spanning tree with at least k internal nodes?



Note: HAMILTONIAN PATH is a special case.

Previous Results

HAMILTONIAN PATH

- $\mathcal{O}(2^n)$ time algorithm with exponential space (Bellmann 62; Held & Karp 62)
- $\mathcal{O}(2^n)$ time algorithm with polynomial space (Kohn et al. 77; Karp 82)
- For graphs of bounded degree, faster algorithms are known for TRAVELLING SALESMAN / HAMILTONIAN CYCLE (Eppstein 03; Iwama & Nakashima 07; Björklund et al. 08)

MIST

- $\mathcal{O}(k^2)$-vertex kernel (Prieto & Sloper 05)
- $49.6^k n^{\mathcal{O}(1)}$ time algorithm (Cohen et al. 09)

Previous Results

HAMILTONIAN PATH

- $\mathcal{O}(2^n)$ time algorithm with exponential space (Bellmann 62; Held & Karp 62)
- $\mathcal{O}(2^n)$ time algorithm with polynomial space (Kohn et al. 77; Karp 82)
- For graphs of bounded degree, faster algorithms are known for TRAVELLING SALESMAN / HAMILTONIAN CYCLE (Eppstein 03; Iwama & Nakashima 07; Björklund et al. 08)

MIST

- $\mathcal{O}(k^2)$-vertex kernel (Prieto & Sloper 05)
- $49.6^k n^{\mathcal{O}(1)}$ time algorithm (Cohen et al. 09)

Our Results

MIST

- $\mathcal{O}^*(3^n)$ time algorithm for general graphs
- $\mathcal{O}((3-\epsilon)^n)$ time algorithm for graphs of bounded degree
- $\mathcal{O}(1.8669^n)$ time algorithm for graphs of maximum degree 3
- $2.1364^k n^{\mathcal{O}(1)}$ time algorithm for graphs of maximum degree 3

New Results

MIST

- $\mathcal{O}^*(2^n)$ time algorithm for general graphs (Nederlof 09)
- $3k$-vertex kernel (Fomin et al. unpublished)
- $8^k n^{\mathcal{O}(1)}$ time algorithm for general graphs

Outline

Observations

Let $t_i(T)$ denote the number of vertices u such that $d_T(u) = i$ in a spanning tree T of G.

Lemma 1

In any spanning tree T, $2 + \sum_{i \geq 3}(i-2) \cdot t_i(T) = t_1(T)$.

In cubic graphs, MIST $=$ find a spanning tree T maximizing $t_2(T)$

Lemma 2 (Prieto & Sloper 03)

An optimal solution T_o to MIST is a Hamiltonian path or the leaves of T_o are independent.

Observations

Let $t_i(T)$ denote the number of vertices u such that $d_T(u) = i$ in a spanning tree T of G.

Lemma 1

In any spanning tree T, $2 + \sum_{i \geq 3}(i-2) \cdot t_i(T) = t_1(T)$.

In cubic graphs, MIST $=$ find a spanning tree T maximizing $t_2(T)$

Lemma 2 (Prieto & Sloper 03)

An optimal solution T_o to MIST *is a Hamiltonian path or the leaves of T_o are independent.*

HAMILTONIAN PATH on cubic graphs

Lemma 3

HAMILTONIAN PATH *can be solved in time* $\mathcal{O}(1.251^n)$ *on cubic graphs.*

Proof.

Simple adaptation of the $\mathcal{O}(1.251^n)$ time algorithm for HAMILTONIAN CYCLE on cubic graphs (Iwama & Nakashima 07). □

Kernel

Lemma 4

MIST on cubic graphs has a $2k$-kernel.

Proof.

Let T be an arbitrary spanning tree of G.
If T has $\geq k$ internal nodes, answer Yes.
Otherwise, $t_3(T) + t_2(T) < k$. By Lemma 1, $t_1(T) < k + 2$. Thus, $|V| \leq 2k$. □

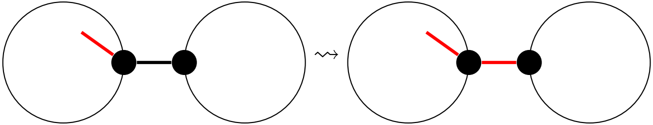
Thus, HAMILTONIAN PATH can be solved in time $1.5651^k \cdot n^{\mathcal{O}(1)}$.

Outline of Algorithm

- Maintain partial spanning tree T
- T always connected
- Look at an edge e of G incident to an edge of T and recursively solve subproblems
  - adding e to T
  - removing e from G

Simplification Rules

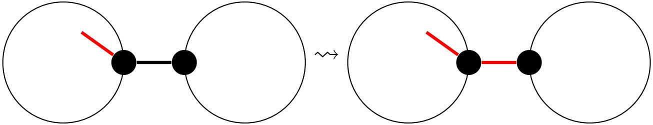
Bridge



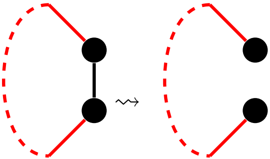
Cycle

Simplification Rules

Bridge

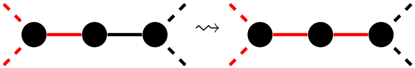


Cycle



Simplifications

Degree2

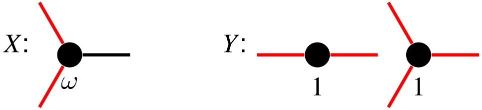


Measure

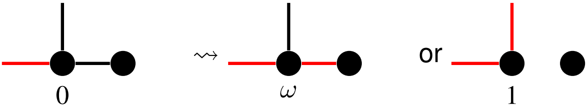
Measure of an instance:

$$\mu(G,T,k) := k - \omega|X| - |Y|, \text{ where}$$

$X := \{v \in V \mid d_G(v) = 3, d_T(v) = 2\}$,
$Y := \{v \in V \mid d_G(v) = d_T(v) \geq 2\}$, and
$\omega = 0.45346$.



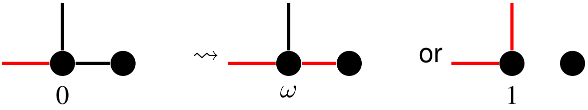
Branching



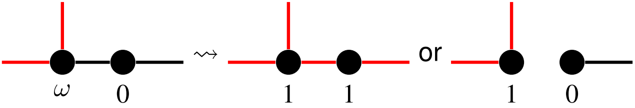
$(\omega, 1)$-branch

$(2-\omega, 1-\omega)$-branch

Branching

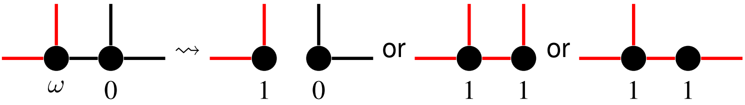


$(\omega, 1)$-branch



$(2-\omega, 1-\omega)$-branch

Branching (2)



$(1-\omega, 2-\omega, 2-\omega)$-branch

Result

Theorem 5

Deciding whether a graph of maximum degree 3 has a spanning tree with at least k internal vertices can be done in time $2.7321^k n^{\mathcal{O}(1)}$.

Improved to $2.1364^k n^{\mathcal{O}(1)}$ in the paper.

Outline

Conclusion

- Very active research around MIST

This paper

- Main Result: algorithm for graphs of maximum degree 3
- analyzed in 2 ways (w.r.t. n and w.r.t. k)
- Novel use of Measure & Conquer for parameterized analysis

Current / future work

- $\mathcal{O}((2-\epsilon_d)^n)$ time algorithms for graphs of max degree d

Open question

- For general graphs, design an algorithm faster than $8^k n^{\mathcal{O}(1)}$.

Thank you!

Questions?

Comments?