Approximation Scheme for Lowest Outdegree Orientation and Graph Density Measures

Łukasz Kowalik

Warsaw University

(work partially done when in

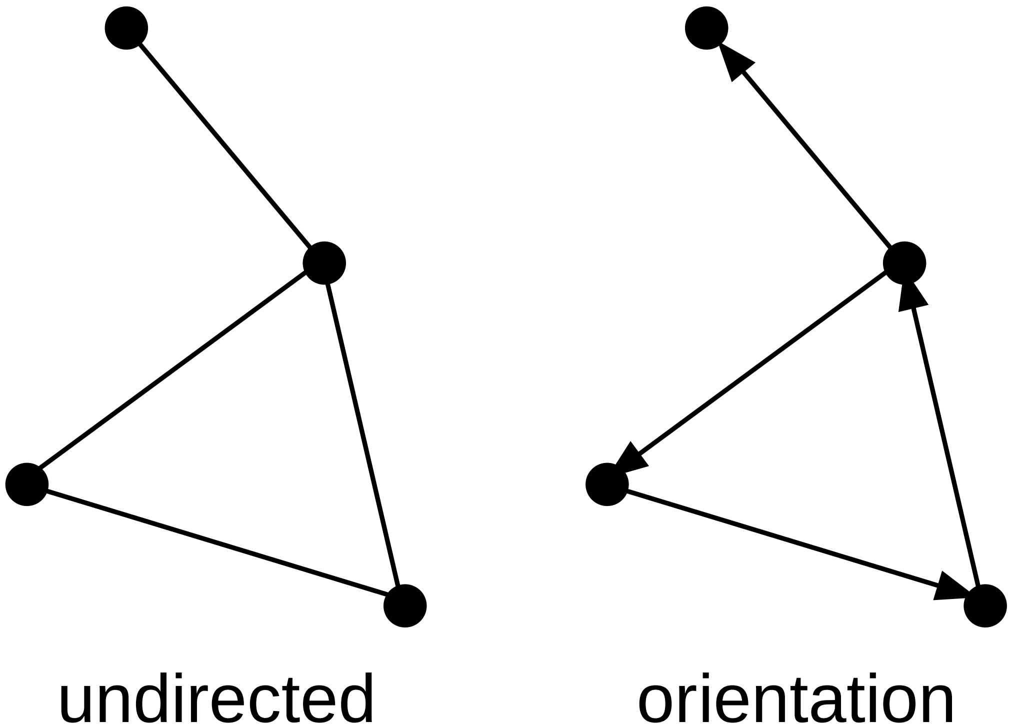
Max-Planck-Institut für Informatik, Saarbrücken)

Outline

1. Statement of the problem,
2. Some applications and related problems,
3. Previous results,
4. My results,
5. Sketch of the algorithm,
6. Open Problems.

Orientation

Orientation of an undirected graph G is a directed graph $\vec{G}$ that is obtained from G by replacing each edge $uv \in E(G)$ by either arc (u, v) or arc (v, u).



Outdegrees

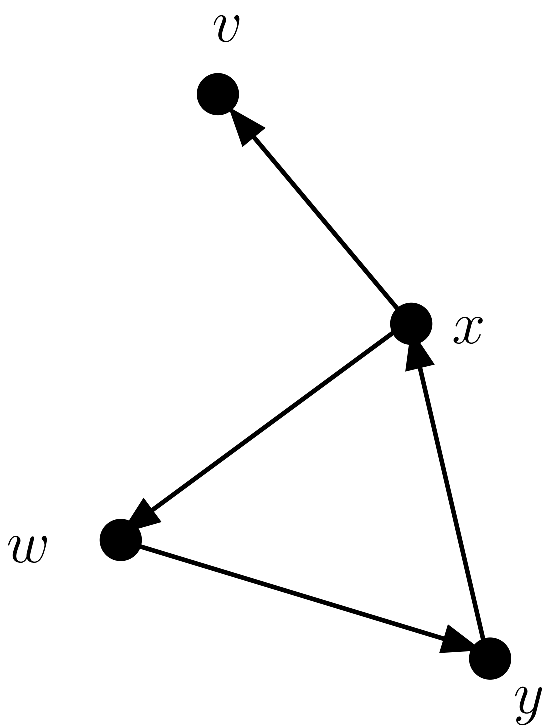
Consider anorientation $\vec{G}$
(or, more generally, a directed graph).

Outdegree of vertex v is the number of edges leaving v. We denote it as $\mathrm{outdeg}(v)$.

Outdegree of orientation $\vec{G}$ is the largest of outdegrees of its vertices :

$$\mathrm{outdeg}(\vec{G}) = \max_{v \in V(G)} \mathrm{outdeg}(v)$$

Outdegrees, cont'd.



$$\mathrm{outdeg}(x) = 2, \ \mathrm{outdeg}(y) = 1$$

$$\mathrm{outdeg}(\vec{G}) = \mathrm{outdeg}(x) = 2$$

Our problem

INPUT: an undirected graph G,

OUTPUT: orientation of G with smallest possible outdegree (i.e. maximum outdegree of a vertex).

Pseudoarboricity

Maximum density of a graph G is defined as

$$d^*(G) = \max_{\substack{J \subseteq G \\ J \neq \emptyset}} \frac{|E(J)|}{|V(J)|}.$$

In another words: density of the densest subgraph.

We define *pseudoarboricity* of G as $P(G) = \lceil d^*(G) \rceil$.

Theorem (Frank & Gyárfás 1976).
For any graph G its pseudoarboricity is equal to the smallest possible outdegree of orientation of G.

Arboricity

Arboricity of a graph G is defined as

$$\mathrm{arb}(G) = \max_{\substack{J \subseteq G \\ |V(G)| \geq 2}} \left\lceil \frac{|E(J)|}{|V(J)| - 1} \right\rceil .$$

Corollary. $P(G) \leq \mathrm{arb}(G) \leq P(G) + 1$.

Theorem (Nash-Williams). For any graph G its arboricity is equal to the smallest number k such that edges of G can be partitioned into k forests.

Related problems

For given graph G:

- determine arboricity of G,
- find the relevant partition into forests,
- determine maximum density of G,
- find the densest subgraph,

An application: Labeling scheme

Let $\vec{G}$ be orientation of G. Then w and v are adjacent in G iff

$$v \in \mathrm{Adj}_{\vec{G}}(w) \quad \text{or} \quad w \in \mathrm{Adj}_{\vec{G}}(v).$$

Such a label has at most $(\mathrm{outdeg}(\vec{G}) + 1)\lceil \log n \rceil$ bits.

Corollary. Assign to each vertex v a label which consists of its id and id's of vertices in $\mathrm{Adj}_{\vec{G}}(v)$. Then one can decide whether two verices u and v are adjacent given only their labels.

Finding lowest outdegree orientations

Finding orientation with outdegree $P(G)$:

- $\mathcal{O}(nm\log^3 n)$ – network flows
  Picard and Queyranne, 1982
- $\mathcal{O}(nm\log(n^2/m))$ – parametric flows
  Gallo, Grigariadis, Tarjan, 1989
- $\mathcal{O}(m\min\{m^{1/2}, n^{2/3}\}\log P(G))$ – matroid partitioning;
  Gabow and Westermann, 1988
- $\mathcal{O}(m^{3/2}\log P(G))$ – flows, Dinitz's algorithm
  Aichholzer, Aurenhammer and Rote, 1995

Finding low outdegree orientations

Finding orientation with outdegree $2P(G)$:

- $\mathcal{O}(m)$ Aichholzer, Aurenhammer, Rote 1995;
  Arikati, Maheshwari, Zaroliagis 1997

My result: $\tilde{\mathcal{O}}(m/\varepsilon)$ algorithm for finding orientation with outdegree $\lceil (1+\varepsilon)d^*(G) \rceil$, for any fixed value of $\varepsilon > 0$.

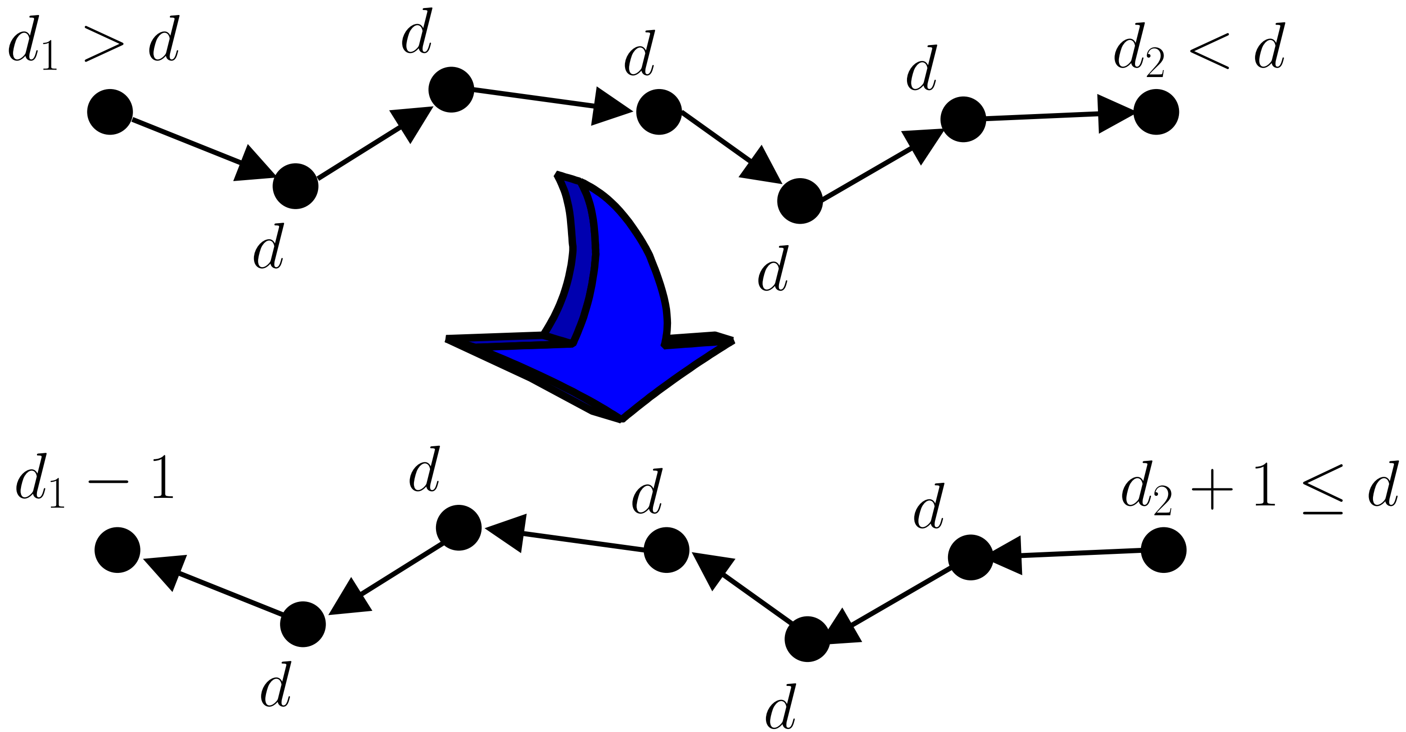
Recall: Optimal outdegree $= P(G) = \lceil d^*(G) \rceil$.

It gives also: Approximation schemes for determining *values* of pseudoarboricity, abroricity and maximum density (with some additive errors).

Idea

Finding an orientation of outdegree d

Basic tool: reverting a path from a vertex of outdegree $> d$ to a vertex of outdegree $< d$.



Idea, cont'd.

Lemma. Let $\vec{G}$ be an orientation with outdegree d of some n-vertex graph G of maximum density d^* and let $d > d^*$. Then for any vertex v the distance in $\vec{G}$ to a vertex with outdegree smaller than d does not exceed $\log_{d/d^*} n$.

Corollary. If $d \geq (1+\varepsilon)d^*$ then the paths which we revert have length $\leq \log_{1+\varepsilon} n$.

Idea, cont'd.

Algorithm TEST(d)**:** Start with arbitrary orientation. Revert the paths as long as they have length $\leq \log_{1+\varepsilon} n$. If the length never exceeds $\log_{1+\varepsilon} n$ some orientation of outdegree d is found. Otherwise we know that $d < (1+\varepsilon)d^*$ and algorithm returns 'FAIL' message.

Main algorithm: Using binary search find the smallest d such that TEST(d) does not return 'FAIL' message.

Efficient implementation of TEST

We reformulate the problem in terms of network flows.

Reverting a path corresponds to sending a flow through augmenting path.

We use Dinic's maximum flow algorithm.

How Dinic's alg. works here?

- In one phase finds a bunch of paths to revert,
- Always chooses *shortest path*,
- One phase takes linear time,
- After each phase length of the shortest path increases,
- After $\leq \log_{1+\varepsilon} n$ phases we stop the algorithm.

Corollary. One run of TEST routine takes $O(m \log_{1+\varepsilon} n) = O(m(\log n)/\varepsilon)$ time.

Total time complexity: $\mathcal{O}(m \log_{1+\varepsilon} n \log d^*)$.

Open Problem I

Observation. The approximation scheme for computing pseudoarboricity gives approximation algorithm for computing arboricity (with additional additive error 1). But NOT for finding the relevant partition into forests.

Problem. Given an orientation of graph G with outdegree d find a partition of G into

- $d + 1$ forests
- αd forests, for some $\alpha < 2$.

The algorithm should have time complexity $\tilde{O}(m)$.

Open Problem II

Find an efficient approximation scheme for the densest subgraph problem.

State of art:

- $\mathcal{O}(nm \log(n^2/m))$ parametric flow exact algorithm,
- simple 2-approximation $\mathcal{O}(m)$-time algorithm (Charikar 2000).

Outline

1. Statement of the problem,
2. Some applications and related problems,
3. Previous results,
4. My results,
5. Sketch of the algorithm,
6. Open Problems.