

Parameterized Algorithms for
Directed Maximum Leaf Problems

Gregory Gutin

Royal Holloway, University of London

joint with Noga Alon, Fedor Fomin,
Michael Krivelevich and Saket
Saurabh

Fixed Parameter Tractability

A parameterized problem Π can be considered as a set of pairs (I, k) where I is the **problem instance** and k (usually an integer) is the **parameter**. Π is called **fixed-parameter tractable (FPT)** if membership of (I, k) in Π can be decided in time $O(f(k)|I|^c)$, where $|I|$ is the size of I , $f(k)$ is a computable function, and c is a constant independent from k and I .

Examples: k -Vertex Cover is FPT ($O(1.271^k + kn)$, Chen, Kanj, Jia, 2001); k -Terminal Steiner Problem in Graphs is FPT; k -Dominating Set in Planar Graphs is FPT; k -Clique is W[1]-complete, and thus is likely not FPT (unless $\text{FPT} = \text{W}[1]$); k -Dominating Set is W[2]-complete.

FPT Results and Books

Many important parameterized undirected graph theory and constraint satisfaction problems have been classified. Books:

- G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer-Verlag, (1999)
- J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, (2006)
- R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, (2006)

FPT Results for Digraphs

Very few parameterized problems on digraphs have been classified:

- Directed k -Cycle and k -Path problems (N. Alon, R. Yuster and U. Zwick, 1994)
- k -Kernel in Planar Digraphs is FPT (G. Gutin, T. Kloks, C.M. Lee and A. Yeo, 2005)
- k -Feedback Vertex (Arc) Set in Tournaments are FPT (V. Raman and S. Saurabh, 2006)
- k -Kernel in Digraphs is $W[2]$ -hard (G. Gutin, T. Kloks, C.M. Lee and A. Yeo, 2005)

Undirected Max Leaf Problem

Spanning MaxLeaf: Given an undirected connected graph G , find a spanning tree of G with maximum number of leaves.

Complexity: It's NP-hard.

MaxLeaf: Given an undirected connected graph G , find a tree of G with maximum number of leaves.

Remark: G has a tree with $\geq k$ leaves iff G has a spanning tree with $\geq k$ leaves.

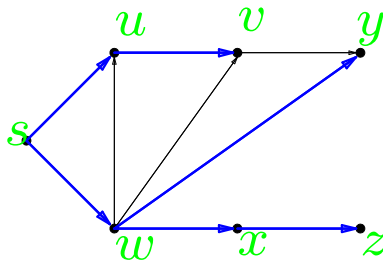
k -Leaf: Given a connected graph G and a parameter k , check whether G contains a spanning (arbitrary) tree with at least k leaves. The problem is FPT.

Out-Trees and Out-Branchings

We say that a subdigraph T of a digraph D is an **out-tree** if T is an oriented tree with only one vertex s of in-degree zero (its **root**).

The vertices of T of out-degree zero are **leaves**.

If T is a spanning out-tree, i.e. $V(T) = V(D)$, then T is an **out-branching** of D .



Proposition (Bang-Jensen, Gutin, Digraphs, 2000) A digraph D has an out-branching iff D has only one source strong component (a strong component C is **source** if there are no arcs entering C).

Directed k -Leaf Problems

Directed k -Leaf: Given a connected digraph D and a parameter k , check whether D contains an out-tree with at least k leaves ($\ell(D) \geq k$?)

Directed Spanning k -Leaf: Given a connected digraph D and a parameter k , check whether D contains an out-branching with at least k leaves ($\ell_s(D) \geq k$?)

Problem (M. Fellows) Are the problems FPT?

Approaches to Prove FPT for Undirected k -Leaf

(a) The Graph Minors Theory of Robertson and Seymour is a powerful (non-constructive) technique for establishing membership in FPT. However, Graph Minors Theory for digraphs is still in a preliminary stage.

(b) Bodlaender used the following arguments: If G contains a star $K_{1,k}$ as a minor, then G has a spanning tree with $\geq k$ leaves. Otherwise, $pw(G) \leq f(k)$ and dynamic programming can be used to decide whether there is a tree with $\geq k$ leaves.

This approach does not work on digraphs because the existence a big out-tree as a minor does not imply the existence of an out-branching or out-tree with many leaves in the original graph.

Approaches to Prove FPT for Undirected k -Leaf

(c) The most efficient approach. Kleitman and West showed that every connected undirected graph G on n vertices with minimum degree at least 3 has a spanning tree with at least $n/4 + 2$ leaves.

Bonsma, Brueggermann and Woeginger (2003) combined this inequality with clever preprocessing rules to obtain the fastest known algorithm for the k -Leaf problem, running in time $O(n^3 + 9.4815^k k^3)$.

It is not clear how to devise a similar approach for digraphs.

Tree Decompositions of Graphs

A **tree decomposition** of an (undirected) graph G is a pair (X, U) where U is a tree whose vertices we will call **nodes** and $X = (\{X_i \mid i \in V(U)\})$ is a collection of subsets of $V(G)$ (**bags**) such that (a) $\bigcup_{i \in V(U)} X_i = V(G)$, (b) for each edge $\{v, w\} \in E(G)$, there is an $i \in V(U)$ such that $v, w \in X_i$, and (c) for each $v \in V(G)$ the set of nodes $\{i \mid v \in X_i\}$ forms a subtree of U .

The **width** of a tree decomposition $(\{X_i \mid i \in V(U)\}, U)$ equals $\max_{i \in V(U)} \{|X_i| - 1\}$. The **treewidth** of a graph G , $tw(G)$, is the minimum width over all tree decompositions of G .

If in the definitions we restrict U to be a path, then we have the definitions of **path decomposition** and **pathwidth**.

Path Decompositions and Vertex Separation

Let G be a graph and let $\sigma = (v_1, v_2, \dots, v_n)$ be an ordering of $V(G)$. For $j \in [n]$ put $V_j = \{v_i : i \in [j]\}$ and denote by ∂V_j all vertices of V_j that have neighbors in $V \setminus V_j$. Setting

$$vs(G, \sigma) = \max_{i \in [n]} |\partial V_i|,$$

we define the **vertex separation** of G as

$$vs(G) = \min\{vs(G, \sigma) : \sigma \text{ is an ordering of } V(G)\}.$$

Proposition KP (Kirousis and Papadimitriou, 1985) *For any graph G , $vs(G) = pw(G)$.*

The family \mathcal{L}

For every digraph D , $\ell(D) \geq \ell_s(D)$.

$$\mathcal{L} = \{D : \ell_s(D) = 0 \text{ or } \ell_s(D) = \ell(D)\}.$$

\mathcal{L} includes strong digraphs, acyclic digraphs, semicomplete multipartite digraphs, quasi-transitive digraphs, etc.

Example: D_n with $1 \rightarrow 2 \rightarrow \dots \rightarrow n$, $n \rightarrow \{2, 3, \dots, n-1\}$; $\ell_s(D) = 1$, $\ell(D) = n - 1$. $D_n \notin \mathcal{L}$.

Directed Spanning k -Leaf on Digraphs in \mathcal{L}

Lemma 2 *Let D be a digraph in \mathcal{L} and $\ell_s(D) > 0$. Then either $\ell_s(D) \geq k$ or the underlying undirected graph of D is of pathwidth at most k^3 .*

Proof: Let $D \in \mathcal{L}$, let $0 < \ell_s(D) < k$, and let T be an out-branching with p leaves.

$V(T)$ can be decomposed into a family \mathcal{P} of $p \leq k - 1$ disjoint paths.

For $p \in \mathcal{P}$, $W(P)$ is the set of out-neighbors of vertices of P outside P . We have $|W(P)| \leq k - 1$ by Lemma 1.

Define $U_1 = \{v \in W(P) : P \in \mathcal{P}\}$;
We have $|U_1| \leq (k - 1)^2$.

Proof Cont'd

Obtain D_1 from D : for every path $P \in \mathcal{P}$ and every vertex $v \in U_1 \cap V(P)$ we delete all arcs from v and to v except those of the path P itself. Thus for every two paths $P, Q \in \mathcal{P}$ there is no arc in D_1 that goes from P to Q .

For $P = u_1u_2 \cdots u_q \in \mathcal{P}$, consider $D_1[V(P)]$; P is a Hamiltonian path. We call u_iu_j in $D_1[V(P)]$ **forward** if $i \leq j - 2$; $S[P]$ are terminal vertices of forward arcs.

We claim that $|S[P]| \leq (k - 2)(k - 1)$.

For each vertex $v \in S[P]$, delete all forward arcs terminating at v but one. The number of forward arcs in the new digraph $f = |S[P]|$.

Proof Cont'd

Suppose that there are $2(k - 1)$ indices

$$i_1 < j_1 \leq i_2 < j_2 \leq \cdots \leq i_{k-1} < j_{k-1}$$

such that each $u_{i_s}u_{j_s}$ is a forward arc. Then the arcs of the paths $u_{i_s}u_{j_s}u_{j_s+1} \cdots u_{i_{s+1}-1}u_{i_{s+1}}$ ($1 \leq s \leq k - 2$) and

$$\{u_{i_{k-1}}u_{j_{k-1}}\} \cup \{u_{i_s}u_{i_{s+1}} : 1 \leq s \leq k - 1\}$$

form an out-tree with k leaves, a contradiction.

Consider the graph G whose vertices are the forward arcs and a pair u_iu_j, u_su_r are adjacent if $[i, j - 1]$ and $[s, r - 1]$ intersect. G is an interval graph and, thus, a perfect graph. By the above, $\alpha(G) < k - 1$. Thus, $\chi(G) = \omega(Q) \geq f/(k - 2)$.

Proof Cont'd

Let Q be a clique in G , let $V(Q) = \{u_{i_s}u_{j_s} : 1 \leq s \leq g\}$ and let $h = \min\{j_s - 1 : 1 \leq s \leq g\}$. Each interval $[i_s, j_s - 1]$ contains h . Therefore, we can form an out-tree with vertices

$$\{u_1, u_2, \dots, u_h\} \cup \{u_{j_s} : 1 \leq s \leq g\}$$

in which $\{u_{j_s} : 1 \leq s \leq g\}$ are leaves. Hence $g \leq k - 1$ and $\frac{f}{k-2} \leq g \leq k - 1$ implying $f \leq (k - 2)(k - 1)$.

Define $U_2 = \{v \in S[P] : P \in \mathcal{P}\}$; $|U_2| \leq p(k - 2)(k - 1) \leq (k - 2)^2(k - 1)$.

Let D_2 be the graph obtained from D_1 after applying the trimming procedure as before around all vertices of U_2 .

Proof Cont'd

Put $U = U_1 \cup U_2$. Let C be a connectivity component of D_2 . Observe that C is a Hamiltonian path $P = v_1v_2 \dots v_q \in \mathcal{P}$ together with its backward arcs.

For every $j \in [q]$ let $V_j = \{v_i : i \in [j]\}$. If for some j the set V_j contained at least k vertices, say $\{v'_1, v'_2, \dots, v'_t\}$ with $t \geq k$, having in-neighbors in the set $\{v_{j+1}, v_{j+2}, \dots, v_q\}$, then D would contain an out-tree with at least k leaves, a contradiction.

Thus, $vs(UN(C)) \leq k$. By Proposition KP, $pw(UN(C)) \leq k$. Hence $pw(UN(D_2)) \leq k$.

Let (X_1, X_2, \dots, X_p) be a path decomposition of $UN(D_2)$ of width at most $k-1$. Then $(X_1 \cup U, X_2 \cup U, \dots, X_p \cup U)$ is a path decomposition of $UN(D)$ of width at most $k + |U| \leq k^3$.

Directed Spanning k -Leaf is FPT for digraphs in \mathcal{L}

Proof: Let $D \in \mathcal{L}$ and $\ell_s(D) > 0$. The proof of Lemma 2 can be turned into a polynomial time algorithm to either build an out-branching of D with at least k leaves or to show that $pw(UN(D)) \leq k^3$ and provide the corresponding path decomposition.

Now the algorithm follows by a simple dynamic programming over the decomposition.

Alternatively, the property of containing an out-branching with at least k leaves can be formulated as a monadic second order formula. Thus, by the fundamental theorem of Courcelle, the problem for all digraphs D with $pw(UN(D)) \leq k^3$ can be solved in $O(f(k) \cdot |V(D)|)$ time.

Directed k -Leaf is FPT for all digraphs

Let R_v be the set of vertices reachable from a vertex $v \in V(D)$ in D . Observe that D has an out-tree with k leaves iff there exists a $v \in V(D)$ such that $D[R_v]$ has an out-tree with k leaves.

Notice that each $D[R_v]$ has an out-branching rooted at v . Thus, we can prove the following theorem, using the arguments in the previous proofs.

Lemma 3 *For a digraph D and $v \in V(D)$, either we have $\ell(D[R_v]) \geq k$ or the underlying undirected graph of $D[R_v]$ is of pathwidth at most k^3 .*

Theorem *Directed k -Leaf is FPT for all digraphs.*