University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Adam Starak
Student no. 361021

Title in English

Master's thesis
in COMPUTER SCIENCE

Supervisor:
dr Michał Pilipczuk
Instytut Informatyki

May 2017

Supervisor's statement

Hereby I confirm that the presented thesis was prepared under my supervision and that it fulfils the requirements for the degree of Master of Computer Science.

Date Supervisor's signature

Author's statement

Hereby I declare that the presented thesis was prepared by me and none of its contents was obtained by means that are against the law.

The thesis has never before been a subject of any procedure of obtaining an academic degree.

Moreover, I declare that the present version of the thesis is identical to the attached electronic version.

Date Author's signature

Abstract

W pracy przedstawiono prototypową implementację blabalizatora różnicowego bazującą na teorii fetorów σ-ρ profesora Fifaka. Wykorzystanie teorii Fifaka daje wreszcie możliwość efektywnego wykonania blabalizy numerycznej. Fakt ten stanowi przełom technologiczny, którego konsekwencje trudno z góry przewidzieć.

Keywords

parameterized algorithm

Thesis domain (Socrates-Erasmus subject area codes)

11.3 Informatyka

Subject classification

D. Software
D.127. Blabalgorithms
D.127.6. Numerical blabalysis

Tytuł pracy w języku polskim

Tytuł po polsku

Contents

Introduction

Blabalizator różnicowy jest podstawowym narzędziem blabalii fetorycznej. Dlatego naukowcy z całego świata prześcigają się w próbach efektywnej implementacji. Opracowana przez prof. Fifaka teoria fetorów σ-ρ otwiera w tej dziedzinie nowe możliwości. Wykorzystujemy je w niniejszej pracy.

Chapter 1

Basic definitions

1.1. Structures

A simple graph G is a pair (V, E) where V denotes a set of vertices and E denotes a set of undirected edges. Let $deg_G(v)$ denote a degree of vertex v in graph G. Let $G \setminus \{v\}$ be the abbreviation for $G' = (V(G) \setminus \{v\}, E(G) \setminus \{(u, v) : u \in V(G)\})$.A *tree* T is a graph where two vertices are connected by excatly one path. A *spanning tree* T of a graph G is a graph which includes all of the vertices of G, with minimum possible number of edges. A *star* S is a tree of size at least 2 for which at most 1 vertex has a degree greater than 1. A vertex in a *star* that has the greatest degree is called a *center* while the others are called *rays*.

1.2. Parameterized complexity

Definition 1.1. *Parameterized problem*

Definition 1.2. *FPT algorithm*

Definition 1.3. *Kernel*

Definition 1.4. *Kernelization algorithm*

1.3. Graph decomposition

Definition 1.5. *Path decomposition and pathwidth*

Definition 1.6. *Tree decomposition and treewidth*

Definition 1.7. *Nice tree decomposition*

Chapter 2

Spanning Star Forest Problem

For a given graph G, we say that S is a *Spanning Star Forest* if every connected component C is a star. In the *Spanning Star Forest Problem* given a graph G, the objective is to determine whether there exists a *Spanning Star Forest*.

It turns out that the problem formulated in such a way is relatively simple. Although, various parametrizations described in this paper make it more complex. The following lemma easily clarifies all the concerns about it's hardness.

Lemma 2.1. *A graph G has a* Spanning Star Forest *if and only if it does not contain any isolated vertices.*

Proof. If G has a *Spanning Star Forest* S, then trivially for all $v \in V(G)$ $1 \leq deg_S(v) \leq deg_G(v)$. Thus, none of the vertices is isolated.

For the opposite direction, we prove the lemma by induction on $|V(G)|$. Assume $|V(G)| = 2$. The statement trivially holds because a graph representing an edge is a correct *Spanning Star Forest*. Let $|V(G)| > 2$. Suppose that there does not exist a vertex v such that $G \setminus \{v\}$ has no isolated vertices. Then, it holds that for all $v \in V(G)$ $deg_G(v) = 1$ so G itself is a correct *Spanning Star Forest*. Now, suppose that v is a vertex such that $G \setminus \{v\}$ has no isolated vertices. From the inductive assumption, let S be a *Spanning Star Forest* of a graph $G \setminus \{v\}$, u be a vertex such that $(u, v) \in E(G)$ and w be a vertex such that $w \in N_S(u)$. Consider the 3 following cases:

1. $deg_S(u) > 1$. Then, $S' = \big(V(S) \cup \{v\}, E(S) \cup \{(u, v)\}\big)$ is a correct solution for graph G.

2. $deg_S(u) = deg_S(w) = 1$. Then, $S' = \big(V(S) \cup v, E(S) \cup (u, v)\big)$ is a correct solution for graph G.

3. $deg_S(w) > 1$. Then, $S' = \big(V(S) \cup \{v\}, (E(S) \cup \{(u, v)\}) \setminus \{(u, w)\}\big)$ is a correct solution for graph G.

Observe that in graph G there are no isolated vertices. Thus, one can always extend a solution inductively.

□

Application of Lemma 2.1 yields the following result for *Spanning Star Forest Problem*.

Theorem 2.1. *Decision version of* Spanning Star Forest Problem *can be solved in linear time.*

Proof. Given an input $G = (V, E)$ the answer is YES if for all $v \in V(G)$ $deg_G(v) \neq 0$ and NO otherwise. □

2.1. Obtaining a solution

In this section we focus on obtaining an arbitrary solution for a given instance of *Spanning Star Forest Problem*. Firstly, let us introduce 2 claims in order to normalize the instance and make the algorithm look more clear.

Claim 2.1. *Family of disjoint* Spanning Star Forests *is a* Spanning Star Forest.

Claim 2.2. *G has a* Spanning Star Forest *if and only if it's spanning tree T has.*

The first claim can be trivially proven by the definition of *Spanning Star Forest Problem* while the second one follows directly from Lemma 2.1. Equipped with this information, all that is left to do, is to design an algorithm which solves *Spanning Star Forest Problem* for trees.

Data: Graph G
Result: Spanning Star Forest of T
$T \leftarrow SpanningTree(G)$;
$S \leftarrow \emptyset$;
for *v: postorder(T) and* $v \notin$ *V(S)* **do**
    if *v is not a root* **then**
        S $\leftarrow S \cup \{(u,v)\}$ where $u = parent(v)$
    else
        S $\leftarrow S \cup \{(u,v)\}$ where u is any of the root's children
    end
end
return S

Algorithm 1: Obtaining a Spanning Star Forest from a tree.

Lemma 2.2. *Algorithm 1 is correct.*

Proof. Assume contrary, that the algorithm yields an incorrect solution S. Consider the first case: a path $(u,v),(v,w),(w,z)$ exists in S where u is v's child, v is w's child and w is z's child. But, if u is w's grandchild and $(u,v),(v,w) \in S$, then it means that w is a root. Contradiction because w cannot be z's child. Now, suppose the alternative relationship: u is v's child, v and z are w's children. Provided that vertices were visited in postorder, edge (v,w) should not have been added because v was introduced by u and w was introduced by z. □

Theorem 2.2. *A solution for Spanning Star Forest Problem can be found in linear time.*

Proof. Spanning tree of any graph can be found in linear time. The main loop has n iterations (every vertex is visited once), each of which takes constant time. Thus, the total runtime is linear. □

2.2. Spanning Star Forest parameterized by the number of stars

In *Spanning Star Forest Problem* parameterized by the number of stars, given a graph G and a natural number k, the objective is to determine whether there exists a *Spanning Star Forest* S such that the number of connected components is at most k.

It is natural to ask whether one can find a solution that minimizes the number of connected components. The problem formulated in that way looks slightly different than the previous one. From the other hand, the problem resembles *Dominating Set Problem*, which is defined as follows:

Definition 2.1. Dominating Set Problem*: Given a graph G and a positive integer k find a set D such that $|D| \leq k$ and every vertex from the graph is adjacent to one of the vertices from D.*

It turns out, that the second comparison is true and *Spanning Star Forest Problem* parameterized by the number of stars is NP-Complete. But, before we begin, let us introduce one more definition and a lemma that supports a reduction.

Definition 2.2. *Dominating mapping: Given an instance (G, k) of Dominating Set Problem and a solution D, a dominating mapping is a function $m : V(G) \setminus D \to D$ such that satisfies $(x, m(x)) \in E(G)$ for all $x \in Dom(m)$.*

Lemma 2.3. *Let (G, k) be an instance of* Dominating Set Problem *without isolated vertices and let D be a solution of minimal size. Then, there exists a dominating mapping m such that m is surjective.*

Proof. Suppose contrary that such a mapping does not exist i.e. for every mapping m there exists a vertex $v \in D$ such that $v \notin im(m)$. Let us break the proof into 4 cases:

1. Suppose $N_G(v) = \emptyset$. Contradiction, G has no isolated vertices.
2. Suppose $u \in N_G(v) \cap D$. Contradiction, D was said to be a solution of minimal size whereas $D \setminus \{u\}$ is a valid, smaller solution.
3. Suppose $u \in N_G(v) \setminus D$ and $w \in N_G(u) \cap im(m)$. If $|m^{-1}(w)| = 1$, then $((D \setminus \{v, w\}) \cup u)$ is a valid, smaller solution for a graph G. Contradiction.
4. Suppose $u \in N_G(v) \setminus D$ and $w \in N_G(u) \cap im(m)$. If $|m^{-1}(w)| > 1$ then a mapping:

$$m'(x) = \begin{cases} v, & \text{if x = u} \\ m(x), & \text{otherwise} \end{cases}$$

   is a valid mapping that satisfies $im(m) \subset im(m')$. Thus, one can create a new mapping m'' inductively such that m'' is surjective. Contradiction, we assumed that no such mapping exists.

Since all the possible cases led to a contradiction, we may claim that there exists a dominating mapping f such that f is surjective. □

Armed with the lemma, we are ready to prove the main theorem of the chapter.

Theorem 2.3. Spanning Star Forest Problem *parameterized by the number of stars is NP-Complete.*

Proof. Membership in NP: given an oracle (O, k), we check whether the number of components in O is less than k and whether every connected component forms a star. The task can be easily done in polynomial time.

We show hardness by a reduction from *Dominating Set Problem* that completes the proof. Let (G, k) be an instance of it. We create a graph G' as follows: for every isolated vertex $v \in V(G)$ introduce a vertex v' and an edge (v, v'). Now, we claim that (G, k) is a YES-instance for *Dominating Set Problem* if and only if $(G'.k)$ is a YES-instance for *Spanning Star Forest Problem* parameterized by the number of stars.

The backward implication is simple. Suppose S is a solution for (G', k). We claim that a set D representing centers of stars is a correct *Dominating Set*. Obviously $|D| \leq k$ because there are at most k connected components. Every vertex from G' is adjacent to one of the centers. If there exists a vertex $v' \in D$ such that $v' \notin V(G)$ we transform the solution as follows: $D := (D \setminus \{v'\}) \cup \{v\}$.

To prove the forward implication, let D be a solution of minimal size for (G, k). Obviously, D is also a minimal solution for a graph G'. Thus, by lemma 2.3. there exists a mapping m that is surjective. Now, we claim that a graph $S = (V(G'), \{(x, m(x)) : x \in Dom(m)\})$ is a correct solution for *Spanning Star Forest Problem*. Trivially, there are no isolated vertices in S. Moreover, there is no path of length 4 because S consists of edges (v, u) such that $v \in D$, $u \notin D$ and for all $u \in V(S) \setminus D$ $deg_S(u) = 1$.

□

The theorem implies that *Spanning Star Forest Problem* parameterized by the number of stars is as hard as *Dominating Set Problem*. Thus, we can immediately obtain the following corollary.

Corollary 2.1. Spanning Star Forest Problem *parameterized by the number of stars is* W[2]-complete.

The problems look so similar that one could ask whether the reverse reduction is true. Indeed, with a small twist to the previous idea one can prove the reverse reduction instantly.

Theorem 2.4. *There exists a reduction from* Spanning Star Forest Problem *parameterized by the number of stars to* Dominating Set Problem.

Proof. Let (G, k) be an instance of *Spanning Star Forest Problem*. We create an instance (G', k') for *Dominating Set* as follows: let $G' = G$ and if G contains an isolated vertex, then $k' = 0$. Otherwise, the value remains the same. Now, we claim that (G, k) is a YES-instance for *Spanning Star Forest Problem* if and only if (G', k') is a YES-instance for *Dominating Set*.

To prove the following reduction one can use the method which was described in Theorem 2.4 with a little remark: if an instance (G, k) contains an isolated vertex, then obviously it is a NO-instance for *Spanning Star Forest Problem* and so is (G', k') for *Dominating Set* because G' is not an empty graph. □

One can observe now the immediate corollary of the theorem 2.3 and theorem 2.4.

Corollary 2.2. *Every theorem that is true for* Dominating Set Problem *if and only if it is true for a* Spanning Star Forest Problem *parameterized by the number of stars.*

As an example, the following theorem described in can be transfered to *Spanning Star Forest Problem* parameterized by the number of stars.

Theorem 2.5. *Unless* CNF-SAT *can be solved in time $\mathcal{O}^*((2 - \epsilon')^n)$ for some $\epsilon' > 0$ there do not exist constant $\epsilon > 0$, $k \geq 3$ and an algorithm solving* Dominating Set Problem *parameterized by the number of stars in time $\mathcal{O}^*(N^{k-\epsilon})$, where N is the number of vertices of the input graph.*

Chapter 3

Spanning Star Forest Problem with isolated edges

In this chapter, we significantly change the problem. Let G be a graph and $F \subseteq E(G)$. In the Spanning Star Forest Problem with isolated edges the question that we want to answer now is that whether there exists a *Spanning Star Forest* S such that $F \subseteq E(S)$. Hardness of the problem lays in deciding which ends of the isolated edges are centers and which are rays. We used three different parameters: number of isolated edges, number of non-isolated edges and treewidth.

3.1. Instance normalization

Observe that a star is a primitive structure. The star's maximal radius is equal to 3. It means that we can look at the problem rather locally than globally. Notice that this time we do not have any limit on the number of connected components. As it was said before, the hardness of the problem lays in choosing which of the endings of an isolated. Thus, it might worth trying to normalize instance i.e. try to remove vertices that are "far enough" from isolated vertices.

Let (G, F) be an arbitrary instance of Spanning Star Forest Problem with isolated edges. Firstly, consider edge cases.

Claim 3.1. *If graph G contains an isolated vertex, it is a NO-instance*

Claim 3.2. *If in graph G there exists a path of size at least 3 made from isolated edges, it is a NO-instance*

Claim 3.3. *If there exists a vertex v such that it is connected to the both endings of an isolated edge, then remove v from G.*

Suppose that isolated edges form a star of size at least 3. Then, the center is already set. Thus, we can remove from the instance all the vertices that are adjacent to the pre-created centers.

Claim 3.4. *Let $C = \{v : |N_G(v) \cap V(F)| > 1\}$. Update $G = G \setminus N_G(C)$.*

Now, let $V_P = \{v : (v, u) \in (E(G) \setminus F) \text{ and } v \notin V(F)\}$ and $V_{NP} = V(G) \setminus V_P$. Finally, $G_{NP} = G[V_{NP}]$ and $G_P = G[V_P]$. Notice an immediate consequence of the partitioning of graph G.

Claim 3.5. *G_P always has a solution*

To prove the claim we can apply lemma 2.1. G_P does not have any isolated edges nor isolated vertices. All that is left to do is to prove that edges between G_P and G_{NP}, that were lost during partitioning, does not have any effect on the solution. The following theorem proves the intuition.

Lemma 3.1. *An instance (G, F) has a solution if and only if (G_{NP}, F) has one.*

Proof. The backward implication is trivial. Suppose S is a solution for an instance (G_{NP}, F). We can partition G into G_P and G_{NP} and find a solution, say S', for a graph G_P. Then, $S \cup S'$ is a correct solution for G.

Now, consider the forward implication. By the claim 3.5 vertices $V(G) \setminus V(G_{NP})$ form a star forest. Let S be a solution for an instance (G, F). Assume contrary that there exists a vertex $v \in V(G_{NP})$ does not belong to any star. Trivially, vertices from $V(F)$ are covered. Thus, $v \in V(G_{NP}) \setminus V(F)$. But, $\{(v, u) : (v, u) \in E(G_{NP})\} = \{(v, u) : (v, u) \in E(G)\}$ which means that there exists an edge $(v, u) \in E(S)$ such that $(v, u) \in E(G_{NP})$. Contradiction. $\square$

3.2. Parametrization by the number of isolated edges

3.3. Parametrization by the number of non-isolated edges

3.4. Parametrization by treewidth

Bibliography

[Bea65] Juliusz Beaman, *Morbidity of the Jolly function*, Mathematica Absurdica, 117 (1965) 338–9.

[Blar16] Elizjusz Blarbarucki, *O pewnych aspektach pewnych aspektów*, Astrolog Polski, Zeszyt 16, Warszawa 1916.

[Fif00] Filigran Fifak, Gizbert Gryzogrzechotalski, *O blabalii fetorycznej*, Materiały Konferencji Euroblabal 2000.

[Fif01] Filigran Fifak, *O fetorach σ-ρ*, Acta Fetorica, 2001.

[Głomb04] Gryzybór Głombaski, *Parazytonikacja blabiczna fetorów — nowa teoria wszystkiego*, Warszawa 1904.

[Hopp96] Claude Hopper, *On some Π-hedral surfaces in quasi-quasi space*, Omnius University Press, 1996.

[Leuk00] Lechoslav Leukocyt, *Oval mappings ab ovo*, Materiały Białostockiej Konferencji Hodowców Drobiu, 2000.

[Rozk93] Josip A. Rozkosza, *O pewnych własnościach pewnych funkcji*, Północnopomorski Dziennik Matematyczny 63491 (1993).

[Spy59] Mrowclaw Spyrpt, *A matrix is a matrix is a matrix*, Mat. Zburp., 91 (1959) 28–35.

[Sri64] Rajagopalachari Sriniswamiramanathan, *Some expansions on the Flausgloten Theorem on locally congested lutches*, J. Math. Soc., North Bombay, 13 (1964) 72–6.

[Whi25] Alfred N. Whitehead, Bertrand Russell, *Principia Mathematica*, Cambridge University Press, 1925.

[Zen69] Zenon Zenon, *Użyteczne heurystyki w blabalizie*, Młody Technik, nr 11, 1969.