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

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

3. $deg_S(w) > 1$. Then, $S' = (V(S) \cup \{v\}, (E(S) \cup \{(u, v)\}) \setminus \{(u, w)\})$ 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 less than 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. The following theorem proves the statement:

Theorem 2.3. *Spanning Star Forest 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 for (G, k). Without a loss of generality, assume that D is an optimal solution. That is, the size of set D is minimal. To create a correct solution S for a *Spanning Star Forest* instance, apply exhaustively the following rules in order:

1. Suppose $v \in D$ and $deg_S(v) = 0$. If $N_{G'}(v) = \{u\}$, then add edge (v, u) to the solution and remove u from G'.

2. Suppose $v \in D$ and $deg_S(v) = 0$. Add a random edge (v, u) where $u \notin D$ to the solution and remove u from G'.

3. Add a random edge (v, u) where $v \in D$ and $u \notin D$ and remove u from G'.

To prove the correctness of the rules, assume contrary, that S is not a valid *Spanning Star Forest* of size at most k. Firstly, let us focus on the number of components. Obviously, S has $|D|$ connected components because only edges (v, u) where $v \in D$ and $u \notin D$ were introduced and $deg_S(u) = 1$ for every $u \notin D$ because the vertices were deleted right after introduction of an edge. Additionally, we can infer that S does not contain a path of length 4 which would prove the contrary.

So, assume that there exists $v \in V(S) \setminus D$ such that $deg_S(v) = 0$. Contradiction, one can apply rule 3 because v must have been dominated by a vertex in G. Now, assume that $v \in D$. Consider the following cases:

1. There exists $u \in N(v)$ such that $u \in D$. Contradiction, D was said to be the optimal dominating set whereas $D \setminus \{u\}$ is a valid, smaller solution for a graph G.

2. Suppose $u \in N_{G'}(v)$ and $w \in N_S(u)$. If $deg_S(w) > 1$, then rule 1 should have been applied to the edge (v, u). Otherwise, if $deg_S(w) = 1$, it is not an optimal solution for a dominating set because $((D \setminus \{v, w\}) \cup u)$ is a valid, smaller solution for a graph G.

□

Lemma 2.3. *There exists a reduction from Spanning Star Forest parameterized by the number of stars to Dominating Set.*

Proof. Let (G, k) be an instance of *Spanning Star Forest Problem*. We create an instance (G', k') for *Dominating Set Problem* 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 Problem*.

To prove the following reduction one can use the method which was described in Theorem 2.3 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 Problem* because G' is not an empty graph. □

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.