Parameterized by treewidth algorithms for Hamiltonian Cycle

Bartłomiej Jachowicz

10 grudnia 2020

Table of Contents

The Hamiltonian cycle problem

Definition 1.1

Hamiltonian cycle is a graph cycle that visit each node from the graph exactly once.

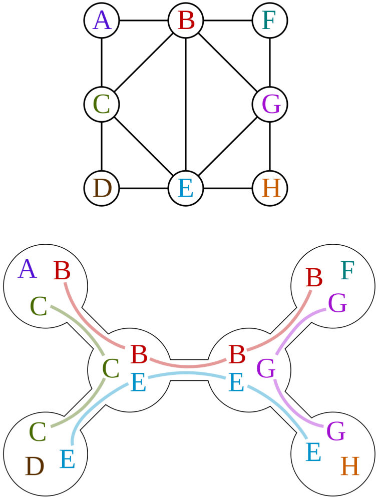
Tree decomposition

Definition 1.2

Tree decomposition of a graph G *is pair* $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, *such that:*

1. T *is a tree.*
2. $\bigcup_{t \in V(T)} X_t = V(G)$. *Each node of* G *is associated with at least one tree node.*
3. *For every edge* $(u, v) \in E(G)$ *there is a subset* X_t *that contains both* u *and* v.
4. *For each* $u \in V(G)$ *set* $T_u = \{t \in V(T) \mid u \in X_t\}$ *induce connected subtree of a tree* T.

Tree decomposition



Treewidth

Definition 1.3

The width of tree decomposition $\mathcal{T} = (T, \{X_t\})$ *is is one less than the maximum node size* $X_t \in T$

Treewidth

Definition 1.4

The width of tree decomposition $\mathcal{T} = (T, \{X_t\})$ *is is one less than the maximum node size* $X_t \in T$

Definition 1.5

Treewidth of a graph G *is the minimal width of tree decomposition out of all tree decompositions for a graph* G.

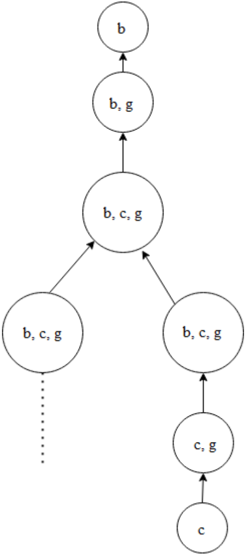
Nice tree decomposition

Definition 1.6

Nice tree decomposition is a tree decomposition $\mathcal{T} = (T, \{X_t\})$, such that T is a rooted tree with a root r and following conditions are satisfied:

- *$X_r = \varnothing$ for a root node r.*
- *For each leaf node t in a tree T: $X_t = \varnothing$.*
- *Each node t in a tree T which is neither leaf or a root has one out of four different types:*
  - ***introduce node for a vertex** v: t has one child t' such that $X_t = X_{t'} \cup \{v\}$ and $v \notin X_{t'}$.*
  - ***introduce node for an edge** (u, v): t has one child t' such that $X_t = X_{t'}$. Each edge from a graph G should be introduced exactly one time.*
  - ***forget node for a vertex** v: t has one child t' such that $X_t = X_{t'} \backslash \{v\}$ and $v \in X_t$.*
  - ***join node**: t has two children t', t'' such that $X_t = X_{t'} = X_{t''}$.*

Nice tree decomposition



Parameterized algorithm

Theorem 2.1

Let $G = (V, E)$ be a given graph of treewidth k. There exists algorithm that decide Hamiltonian Cycle problem for a graph G in time $2^{O(klogk)}n^{O(1)}$ if a nice tree decomposition of width k is given.

Parameterized algorithm

Theorem 2.2

Let $G = (V, E)$ be a given graph of treewidth k. There exists algorithm that decide Hamiltonian Cycle problem for a graph G in time $2^{O(klogk)}n^{O(1)}$ if a nice tree decomposition of width k is given.

Algorithm idea: For each node in a tree decomposition we want to create a set of function and matching pairs. We will build this sets starting from the leaf nodes up to the root. Every set element will represent a set of paths which are a trace of Hamiltonian Cycle on vertices from input graph already introduced to a nice tree decomposition.

Parameterized algorithm

Input: graph $G = (V, E)$ and a nice tree decomposition $\mathcal{T} = (T, \{X_t\})$.

Algorithm: For each node $t \in T$ with a set of vertices X_t algorithm build dynamically set $HC(X_t)$ of pairs (s_{deg}, M) where $s_{deg} : X_t \rightarrow \{0, 1, 2\}$ and M is a matching defined on set $\{v \in X_t : s_{deg}(v) = 1\}$. Algorithm compute those sets for every nodes based on node type and sets computed for his children. For node t we denote his children as t' and t''.

Output: There exist a Hamiltonian Cycle in a graph G if set $HC(r) \neq \varnothing$, where r is a root of a nice tree decomposition r.

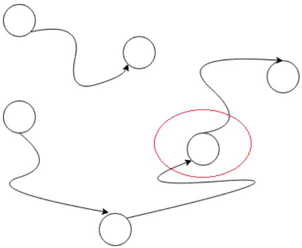
Leaf node:
$HC(X_t) = \{(\emptyset_f, \emptyset)\}$ where $\emptyset_f$ is function defined for empty set.

Introduce vertex node:
$HC(X_t) = \{(s_{deg}, M') : (s'_{deg}, M') \in HC(X_{t'})\}$ where s_{deg} is defined as:

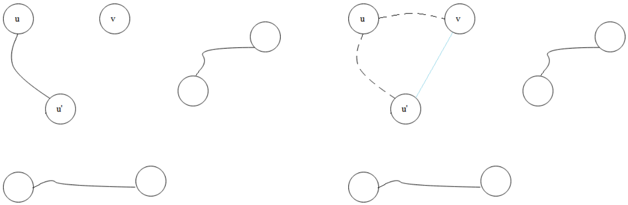
$$s_{deg}(x) = \begin{cases} s^{'}_{deg}(x) & \text{if } x \in X_{t'}, \\ 0 & \text{if } x \notin X_{t'}. \end{cases}$$

Parameterized algorithm

Forget node for a vertex v:
$HC(X_t) = \{(s_{deg}, M') : (s'_{deg}, M') \in HC(X_{t'})$ and $s'_{deg}(v) = 2\}$ where s_{deg} is a restriction of function s'_{deg} to the set X_t.

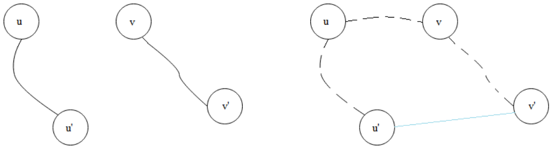


Introduce node for an edge (u, v):
Set $HC(X_t)$ is computed in different way according to value of $s'_{deg}(u)$ and $s'_{deg}(v)$:
1. $s'_{deg}(u) = s'_{deg}(v) = 0$: increase both vertices degree and add new pair to the matching M
2. $(s'_{deg}(u) = 1$ and $s'_{deg}(v) = 0)$ or $(s'_{deg}(u) = 0$ and $s'_{deg}(v) = 1)]$



Parameterized algorithm

Introduce node for an edge (u, v):
Set $HC(X_t)$ is computed in different way according to value of $s'_{deg}(u)$ and $s'_{deg}(v)$:
3. $s'_{deg}(u) = s'_{deg}(v) = 1$



4. $s'_{deg}(u) = 2$ or $s'_{deg}(v) = 2$: we can not add any new pair, at least one vertex has already degree 2

Parameterized algorithm

Join node:

Here we have two sets $HC(X_{t'}), HC(X_{t''})$ because join node has two children.

Algorithm adds element (s_{deg}, M) to set $HC(X_t)$ if:

- $s^{'}_{deg}(v) + s^{''}_{deg}(v) \leqslant 2$ for every $v \in X_t$
- $G' = (X_t, (M' \cup M''))$ is an acyclic graph

and pair (s_{deg}, M) is defined as follows:

- $s_{deg} = s^{'}_{deg}(v) + s^{''}_{deg}(v)$ for every $v \in V$,
- $(u, v) \in M$ iff u and v are ends of the same path in $G' = (X_t, (M' \cup M''))$

Improve parameterized algorithm

Theorem 2.3

Let $G = (V, E)$ *be a given graph of treewidth* k. *There exists algorithm that decide Hamiltonian Cycle problem for a graph* G *using Gaussian elimination in time* $2^{O(k)}n^{O(1)}$ *if a nice tree decomposition of width* k *is given.*

Improve parameterized algorithm

Algorithm idea: To reduce previous algorithm time which is caused by the number of possible matchings it has to compute we will use Gaussian elimination.
For each node t we compute partitions of set X_t and build matrix from them. Hence we can remove those cuts which are linear combination of other cuts and reduce the number of matching we need to store for a node. To perform in time mentioned in theorem algorithm uses additional matrix of smaller size defined on set called cuts.

Improve parameterized algorithm

Definition 2.1

Let $\Pi(U)$ be a set of all partitions of set U and $\Pi_2(U)$ be a set of all partitions on two elements sets.
For two partitions P and Q we define $P \sqcup Q$ as a minimal partition of U containing both partition P and Q

Definition 2.2

Define partition matrix as:

$$\mathcal{M}[P][Q] = \begin{cases} 1 & \textit{jeśli } P \sqcup Q = \{U\}, \\ 0 & \textit{w przeciwnym przypadku.} \end{cases} \tag{1}$$

Improve parameterized algorithm

Definition 2.3

Define cuts for set of vertices U:

$$\boldsymbol{cuts}(U) := \{(V_1, V_2) \ : \ V_1 \cup V_2 = U \wedge V_1 \cap V_2 = \varnothing \wedge w_1 \in V_1\}, \tag{2}$$

where w_1 is arbitrary chosen vertex from U.

Definition 2.4

Define $\mathcal{C} \in \{0,1\}^{\Pi_2(U) \times \boldsymbol{cuts}(X)}$ as:

$$\mathcal{C}[M][(V_1, V_2)] = \begin{cases} \textit{1} & \textit{if every pair in matching } M \textit{ on set } U \\ & \textit{is in } V_1 \textit{ or in } V_2, \\ \textit{0} & \textit{otherwise.} \end{cases}$$

Tests results

Presented algorithms were tested on small random graphs generated by self implemented graph generator. For instances of few hundred vertices and edges both algorithms performed very well with similar results.

Additionally I tested them with use of graphs from *Flinders Hamiltonian Cycle project*. This is a collection of 1000 instances for Hamiltonian Cycle problem with average size over 3000 vertices. For bigger instances the improved algorithm is much faster than standrad one.

Instance	$\|V\|$	$\|E\|$	Treewidth	Standard	Improved
graph0039	280	430	8	191512	159652
graph0344	2023	4646	6	87200	43744

The end