Lecture 2 (12.10.2012) Author: Kamil Yurtsever

Minors and Iterative Compression

1 Theory of graph minors

We define following operations on graphs:

$-e$ - edge removal

$-v$ - vertex removal (with removal of incident edges)

$/e$ - edge contraction (with collapse of arising self-loops and multiple edges)

Here we should note that in some situations we won't collapse self-loops and multiple edges created by edge contraction.

Definition 1. Relation of *being a minor* is defined as follows.

Graph H is a minor ($\leq_m$) of graph G iff H is obtainable from G by a sequence of aforementioned operations ($-e$, $-v$, $/e$).

Example 2. Given lattice graph 5×5, graph K_4 is its minor as we can show sequence of operations that transform first one into the second one. However graph K_5 isn't its minor as $-e$, $-v$ and $/e$ preserve planarity.

Definition 3. Relation of *being a minor* can also be defined equivalently this way.

H is a minor of G iff for each vertex u in H exists set V_u of vertices from G, such that

1. $\forall_{u,u' \in V(H)} u \neq u' \rightarrow V_u \cap V_{u'} = \emptyset$
2. $\forall_{u \in V(H)} G[V_u]$ is connected
3. $\forall_{u,u' \in V(H)} (u, u') \in E(H) \rightarrow E_G(V_u, V_u') \neq \emptyset$

Such family of sets is called model of H in G.

We can see that this definition implies the earlier one. We can choose connected components from graph G and remove all other vertices. Then we contract these components into singular vertices and declare mapping from these new vertices to vertices of graph H.

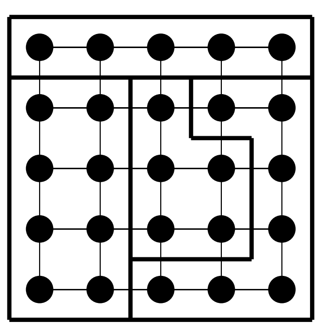

Figure 1: Model of K_4 in 5×5 lattice.

Definition 4. Relation of *being a topological minor*

Graph H is a topological minor ($\leq_{top}$) of graph G iff for each vertex u in H exists vertex v_u in G such that for every pair $u, u' \in E(H)$ exists path $p_{uu'}$ in G which connects v_u with $v_{u'}$ and all such paths are vertex disjoint (except for the endpoints).

Example 5. Given lattice graph 5×5, graph K_4 is its topological minor as we can show vertex mapping and appropriate vertex disjoint paths.

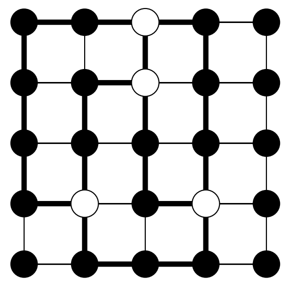

Figure 2: K_4 as a topological minor in 5×5 lattice.

Definition 6. Relation of *being a topological minor* can be alternatively defined so.

$H \leq_{top} G \iff H$ can obtained from G using $-e$, $-v$ and $/uv$, but we allow for edge contraction only if $deg(u) = 2$ or $deg(v) = 2$.

Edge contraction with this constraint corresponds with contracting of paths selected in earlier definition.

Let's take a look at complexity of checking if graph H is a minor of graph G.

From now on we will asume that $|V(H)| = k$ and $|V(G)| = n$. When checking if $H \leq_m G$ we can check all possible reductions, which there are $(k+1)^n$. However we will find better FPT solution.

We can check $H \leq_{top} G$ in $\mathcal{O}\left(n^k \cdot T(DPP)\right)$, where $T(DPP)$ is time of solving Disjoint paths problem. This is because first we can guess k vertices. Then we just have to solve problem of joining these vertices with disjoint paths. This problem is known as Disjoint paths problem.

Disjoint paths problem (DPP)
Input: A graph G and sequence of k pairs (s_i, t_i)
Question: Sequence of vertex disjoint paths p_i where i-th path joins s_i with t_i

We won't prove next few facts.

Theorem 7. *Directed DPP (i.e. input graph is directed) is NP-hard even for $k = 2$[6].*

There is an XP ($n^{\mathcal{O}(k)}$) algorithm for directed DPP in planar graphs[11] but we don't know if this problem is in FPT.

Theorem 8. *$H \leq_m G$ can be checked in $\mathcal{O}(f(k) \cdot n^3)$[9] so this problem is FPT.*

$f(k)$ in this theorem is very big - some kind of power tower of height k. This result is achieved by basing on another theorem

Theorem 9. *DPP can be solved in $\mathcal{O}(f(k) \cdot n^{\mathcal{O}(1)})$[9].*

Again $f(k)$ is very big but for planar graphs we know that $f(k) = 2^{2^k}$[1].

Theorem 10. *$H \leq_{top} G$ is also FPT(k) with $f(k)n^3$[7].*

f(k) is similar to ones above.

1.1 Ordering properties of minors

Both $\leq_m$ and $\leq_{top}$ form an ordering. Also $H \leq_{top} G \to H \leq_m G$.

Definition 11. $(\Omega, \leq)$ is a *quasi ordering* iff

(i) $a \leq b$ and $b \leq c \to a \leq c$

(ii) $a \leq a$

However we **don't** require third property of well ordering, that if $a \leq b$ and $b \leq a$ then $a = b$.

Definition 12. $(\Omega, \leq)$ is a *well quasi ordering* iff

$$\forall_{x_1,x_2,...}\exists_{i_1,i_2,...}x_{i_1} \leq x_{i_2} \leq ... \wedge i_1 < i_2 < ...$$

equivalently we can say:
There are no infinite antichains or infinite strictly descendening sequences.

The question is if $\leq_m$ is WQO. We can't have infinite strictly descending sequence with relation to $\leq_m$. This is because if $G \leq_m H$ and $G \neq H$ then $|E(G)| + |V(G)| \leq |E(H) + V(H)|$ and with every step this number has to become lower until we get an empty graph after finite number of steps. There are also no infinite antichains with relation to $\leq_m$ but we won't prove it here.

Theorem 13. *$\leq_m$ is WQO[10]*

However (again without proof)

Theorem 14. *$\leq_{top}$ isn't WQO*

Now we will show algorithm for checking planarity of a graph that uses WQO property of $\leq_m$. Let $\mathcal{P}$ be all planar graphs. We know that if $H \leq_m G$ and G is planar then H is planar as well. Therefore $\mathcal{P}$ is class closed under taking minor, i.e. if graph is in $\mathcal{P}$ then all its minors are planar as well.

Definition 15. $\mathcal{K}$ - minimal non-planar graphs
$K \in \mathcal{K} \iff K \notin \mathcal{P} \wedge \hat{K} \leq_m K \to \hat{K} \in \mathcal{P}$

We can see that $\mathcal{K}$ is an antichain and as $\leq_m$ is WQO we know that this set is finite. Also we know that graph G is planar iff $\neg\exists_{K \in \mathcal{K}} K \leq_m G$. Therefore this algorithm is correct

function IsPlanar(G)
  for all $K \in \mathcal{K}$ **do**
    if $minor(K, G)$ **then**
      return false
    end if
  end for

return true
end function

As $\mathcal{K}$ is fixed this algorithm runs in $\mathcal{O}(n^3)$, as checking $minor(K, G)$ is FPT($|K|$) but possible graphs are fixed (also $|K|$ is constant) and we have only fixed amount of these graphs.

Of course we know that $\mathcal{K} = \{K_{3,3}, K_5\}$. However this algorithm works for any class of graphs closed under taking a minor (e.g. planar graphs on sphere with three handles).

Theory of minors will return in later lectures.

2 Iterative compression

FEEDBACK VERTEX SET (FVS) **Input:** A graph G and number K. **Question:** If there is set of vertices $X \subseteq V(G)$ such that $|X| \leq k$ and $G \setminus X$ is a forest.

We will create algorithm for this problem with complexity $\mathcal{O}(5^k \cdot n^{\mathcal{O}(1)})$ using iterative compression technique.

Let $V(G) = \{v_1, ..., v_n\}$ and $G_i = G[\{v_1, ..., v_i\}]$. For G_k we can easily find solution $X_k = V(G_k) = \{v_1, ..., v_k\}$. Given solution X_{i-1} for graph G_{i-1} with property $|X_{i-1}| \leq k$, we can create solution $X_i = X_{i-1} \cup \{v_i\}$ for G_i. Of course $|X_{i-1} \cup \{v_i\}| \leq k + 1$.

Now if we could compress this solution we could solve FVS by iteratively expanding and compressing solution.

COMPRESSION FVS **Input:** Graph G, number k and set Z of vertices from G where $|Z| = k + 1$ and $G \setminus Z$ is a forest. **Question:** What is set of vertices $X \subseteq V(G)$, $|X| \leq k$ such that $G \setminus Z$ is a forest.

To solve this problem at first we guess $X \cap Z$. There are 2^{k+1} possibilities and in each we get $X_Z \subseteq Z$. Then we feed $G \setminus X_Z$, $k - |X_Z|$ and $Z \setminus X_Z$ to another subproblem — DISJOINT COMPRESSION FVS.

DISJOINT COMPRESSION FVS **Input:** Graph G, number k and set Z of vertices from G where $|Z| = k + 1$ and $G \setminus Z$ is a forest. **Question:** Is there $X \subseteq V(G) \setminus Z$ such that $|X| \leq k$ and $G \setminus X$ is a forest.

First, we have to notice that if $G[Z]$ isn't a forest then this problem has no solution. Otherwise we will simplify this problem by considering these options, given $v \in V(G) \setminus Z$

- $deg(v) = 1$ or $deg(v) = 0$. In such situation v is of no concern for us as it won't be in any cycle. We remove v.
- $deg(v) = 2$. If all its neighbors aren't in Z then if it were in a solution we could just as well take any of its neighbors, therefore we contract it with its neighbor. If v has both of its neighbors in Z we just let it be. If it has one neighbor in Z then, as we don't want to forget about some cycles, we will contract v with its neighbor in Z. If it creates multiple edge that means that we had a triangle between the contracted vertex v, some vertex in Z and some other vertex u. We can't remove vertex from Z, so we have to remove v or u. As v had no other neighbors (as it had degree 2) we can greedily add u to the solution and remove it from the graph.

Now if v is a leaf of forest (a vertex of degree at most 1 in $G \setminus Z$) then there are at least two edges e_1, e_2 leading to Z.

if e_1, e_2 lead to the same connected component of $G[Z]$ **then**
  remove v
  $k \leftarrow k - 1$
else
  branch
    remove v
    $k \leftarrow k - 1$
  else branch
    $Z \leftarrow Z \cup \{v\}$
  end branch
end if

As we can see with every branch number of vertices lowers by one or number of connected components in $G[Z]$ lowers at least by one. We know that at the beginning $G[Z]$ has $\leq k+1$ connected components. If we define potential $\phi = k + (cc(G[Z]) - 1)$ (where $cc(G)$ is number of connected components of a graph), then we know that $\phi \leq 2k$ and it lowers by one with every branch. Therefore complexity of this algorithm is $\mathcal{O}^\star(2^{2k})$. It will be called for every guessed subset of Z from Compression FVS of which there are $\mathcal{O}^\star(2^{k+1})$.

However when we call Disjoint compression FVS from Compression FVS with k we set new k_1 which is smaller but we know that number of connected components in new Z is limited by $k - k_1$. Now we can think in terms of this k about complexity of Disjoint compression FVS and then complexity is

$$\sum_{k_1=0}^{k} \binom{k+1}{k_1} 4^{k-k_1} = \sum_{k_1=0}^{k} \binom{k+1}{k_1} 4^{k-k_1} \cdot 1^{k_1-1} = 5^{k+1}$$

This algorithm was first presented in 2008[4].

There is also an algorithm $\mathcal{O}^\star((1+2\sqrt{2})^k)$[3], and there is also $\mathcal{O}^\star(3^k)$[5] randomized algorithm that will be shown in later lecture.

Also there is an old $\mathcal{O}^\star(4^k)$[2] randomized algorithm:

1. Remove vertices with degree 0 or 1.

2. For all vertices with degree 2 contract them. If that creates self-loops remove that vertex and add it to the solution. If that creates multiple edge, branch and remove one of the ends adding it to the solution.

3. Randomly choose end of an edge, add it to the solution and remove it from the graph.

4. Repeat whole procedure until graph becomes a forest or solution becomes too big.

Lemma 16. *If min deg$(G) \geq 3$ then for all $X \subseteq V$ where $G[V \setminus X]$ is a forest, # of edges with at least one end in X is greater or equal $\frac{|E|}{2}$.*

If this lemma is true we will have $\frac{1}{2}$ probability of choosing good edge and then probablity $\frac{1}{2}$ of choosing the right vertex.

Proof

Let say $V \setminus X$ has a vertices. If $a \leq \frac{|E|}{2}$ then of course lemma is true, so we assume $a > \frac{|E|}{2}$. If minimal degree is 3 then there are $3a$ ends of edges there. At most $2a$ of them is used in there. Therefore a edges have to come out of X so at least $\frac{|E|}{2}$ edges come out of X.

Iterative compression was originally created for solving Odd cycle transversal problem[8], however we won't describe how it can be used to solve this problem here.

Odd cycle transversal **Input:** Graph G, number k. **Question:** Set $X \subseteq G$ where $\lvert X \rvert \leq k$ and $G \setminus X$ is bipartite graph.

References

[1] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Tight bounds for linkages in planar graphs. In *ICALP (1)*, pages 110–121, 2011.

[2] Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *J. Artif. Intell. Res. (JAIR)*, 12:219–234, 2000.

[3] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set new measure and new structures. In *SWAT*, pages 93–104, 2010.

[4] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008.

[5] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.

[6] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoret. Comput. Sci.*, (10):111–121, 1980.

[7] Martin Grohe, Ken ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *STOC*, pages 479–488, 2011.

[8] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

[9] Neil Robertson and Paul Seymour. Graph minors. xiii. the disjoin paths problem. *Journal of Combinatorial Theory*, Series B(63):65–100.

[10] Neil Robertson and P.D. Seymour. Graph minors. xx. wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325 – 357, 2004.

[11] Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994.