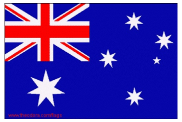
Two-Layer Planarization:

Improving on Parameterized Algorithmics

Henning Fernau

Universität Tübingen 

The University of Newcastle 

University of Hertfordshire (to come) 

January 22, 2005

Overview

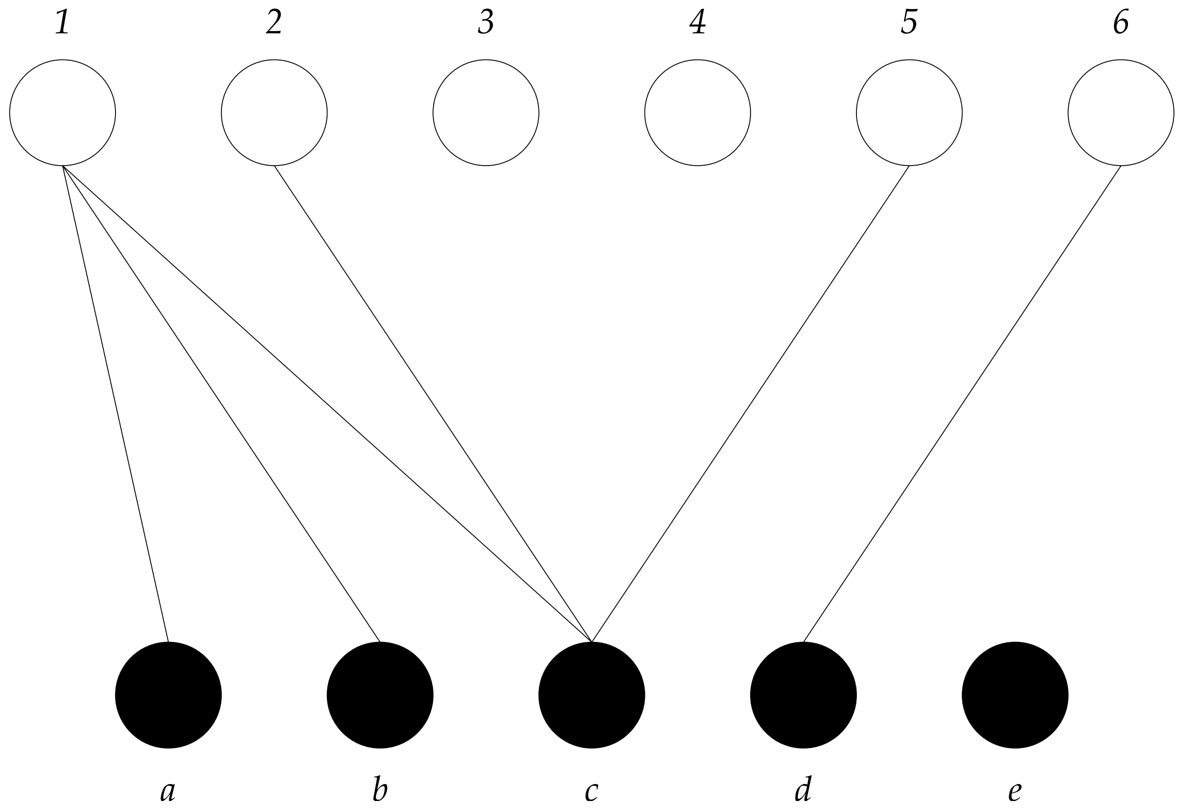
- the problems:
  ONE-LAYER PLANARIZATION and
  TWO-LAYER PLANARIZATION

- the methodology: parameterized algorithmics

- our solutions: kernelization and search tree techniques;
  how and why do they work?

- conclusions

Biplanarity



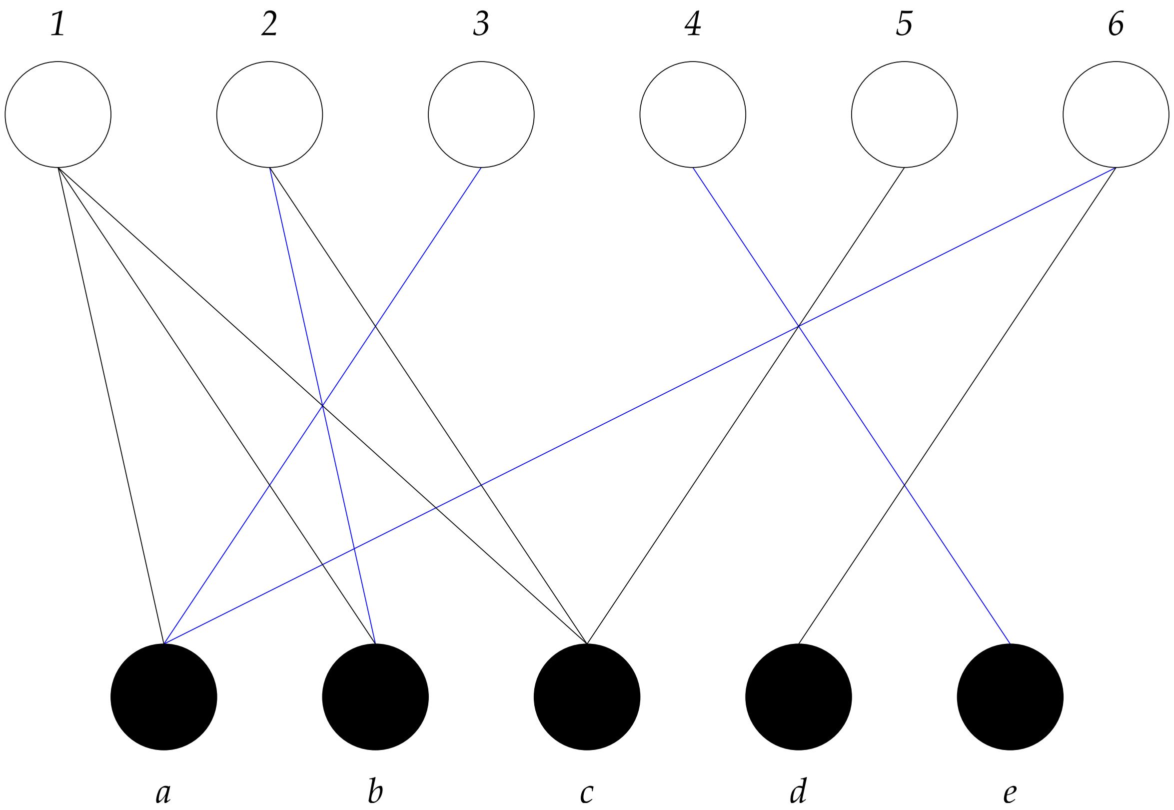
What if:

- graph is not biplanar?

Delete a few edges

- Graph is not bipartite?
- To which layer are vertices assigned?
- In which order do vertices occur on a layer?

Adding four edges. . .



The problems

TWO-LAYER PLANARIZATION

Can k edges be deleted from a given graph G so that the remaining graph is biplanar?

ONE-LAYER PLANARIZATION

Given a bipartite graph $G = (V_1, V_2; E)$, an ordering $<_1$ on V_1, and a parameter k: can k edges be deleted from G so that the remaining graph has a biplanar drawing that respects V_1, V_2 and $<_1$?

Background: Sugiyama Layering Algorithm

Problem: Construct a "hierarchical" drawing of a given directed graph.

Method: Work in three phases:

- Assign vertices to "layers."
- Fix ordering of vertices within layers s. t. # of crossing-producing edges is minimized.
- Determine concrete positions of vertices according to ordering.

Doing the 2nd step layerwisely gives ONE-LAYER PLANARIZATION; in the beginning TWO-LAYER PLANARIZATION.

The Curse of Combinatorics

Folklore: Nearly all "interesting" computational problems in graph theory are "(NP-)hard".

Examples:

- Find a vertex cover of size $\leq k$.

- Find an independent set of size $\geq k$.

- ...

Parameterized Problems and Algorithms

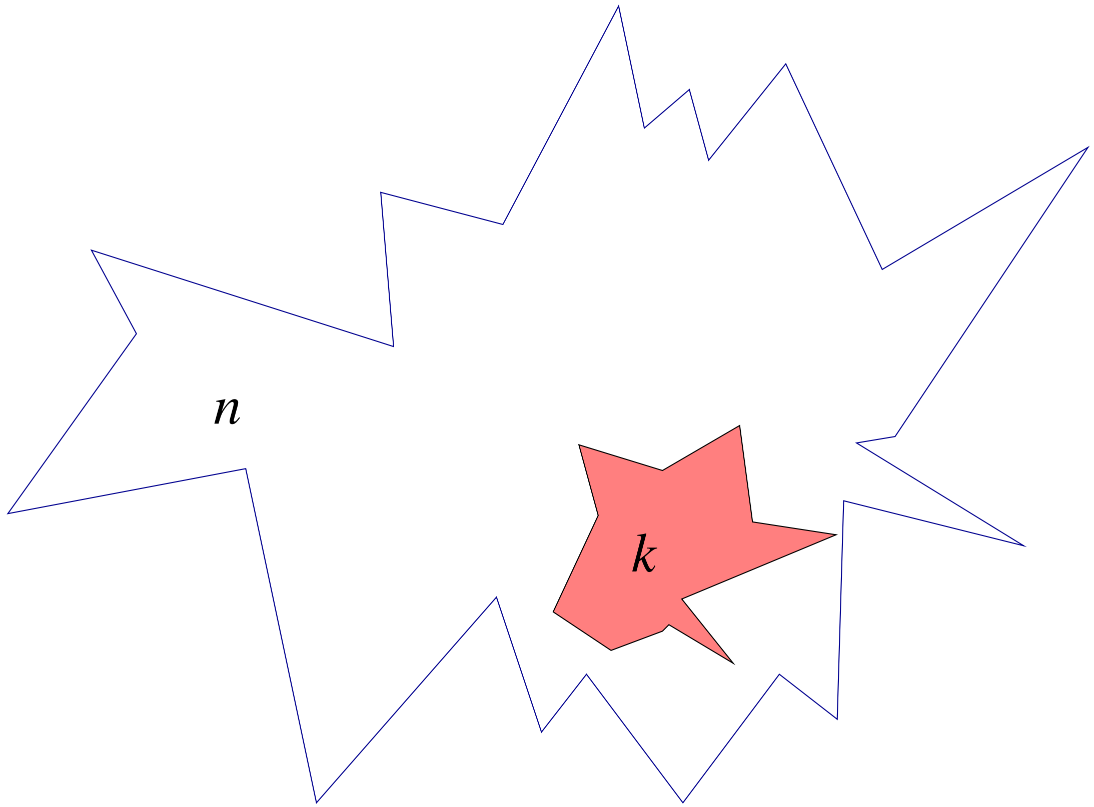
Idea: measure an instance in two dimensions:

- overall size n and
- parameter [size] k.

The parameter can (in principle) be *anything*.
For minimization, take the natural parameter:
the entity to be minimized.

Controlled Explosion:

The idea of parameterized algorithms: the parameter is small.



Parameterized Algorithms and Feasibility

Three equivalent characterizations of fixed parameter tractable $\mathcal{FPT}$:

- running time $\mathcal{O}(f(k)p(n))$ for arbitrary f and a polynomial p;
- running time $\mathcal{O}(f(k) + p(n))$ for arbitrary f and a polynomial p;
- being pol.-time reducible to a problem kernel, i.e., another instance of the same problem of size $\mathcal{O}(f(k))$ and parameter $\mathcal{O}(g(k))$ for arbitrary f, g.

Contrast: running time $\mathcal{O}(n^k)$.

Parameterized Algorithms Design—Standard Methodology

1. Look for a suitable parameter.

2. Find appropriate reduction rules to get a small kernel
   $\rightsquigarrow$ data reduction.

3. Develop search tree algorithm.

Remark 1: Other "parameterized methodologies" available, but not "practical."

Remark 2: Notion of W[1]-hardness $\rightsquigarrow$ methodology may fail.

Complexity of biplanarization problems

Theorem: (Eades, Whitesides 1994) TWO-LAYER PLANARIZATION and ONE-LAYER PLANARIZATION are $\mathcal{NP}$-complete (even under very restricted circumstances)

Theorem: (Dujmović and 11 more authors GD'01) TLP as well as OLP are in $\mathcal{FPT}$.

More specifically:

Theorem: OLP is solvable in time $\mathcal{O}(3^k|G|)$.

Theorem: TLP is solvable in time $\mathcal{O}(k \cdot 6^k + |G|)$.

Our contributions

- We derive a small kernel for ONE-LAYER PLANARIZATION.

- We get a better running time estimate for a (still rather trivial) search tree algorithm for ONE-LAYER PLANARIZATION.

- We get a better running time estimate for a (still rather trivial) search tree algorithm for TWO-LAYER PLANARIZATION.

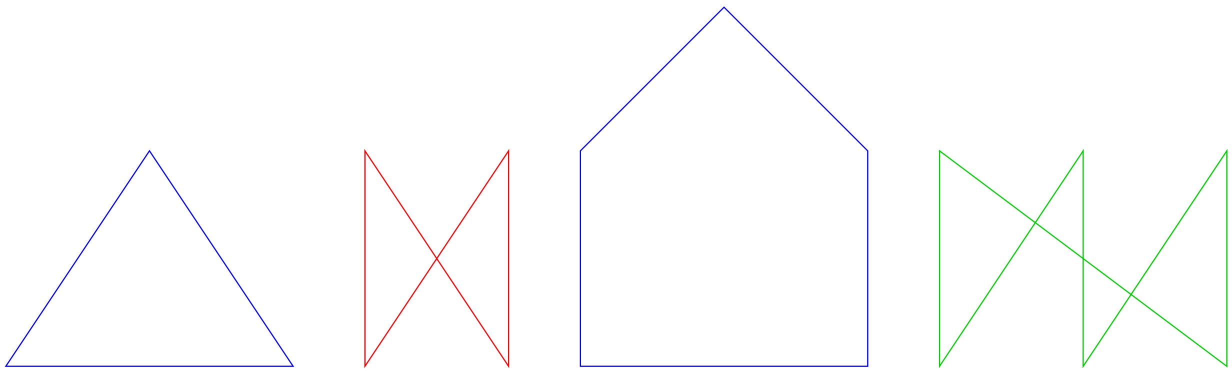
The main theorems

Theorem 1: Any ONE-LAYER PLANARIZATION instance $(G, <, k)$ can be reduced such that $|E(G)| \leq k^3$. The kernel can be found in time $\mathcal{O}(|G|^2)$.

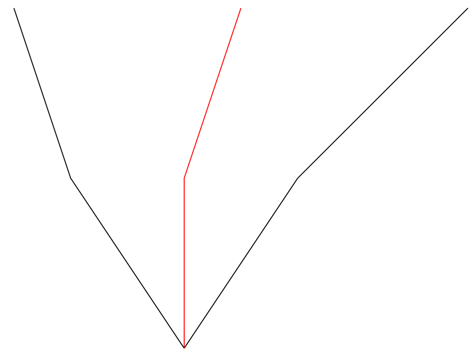
Theorem 2: OLP can be solved in $\mathcal{O}(k^3 \cdot 2.5616^k + |G|^2)$ time.

Theorem 3: TLP can be solved in $\mathcal{O}(k^2 \cdot 5.1926^k + |G|)$ time.

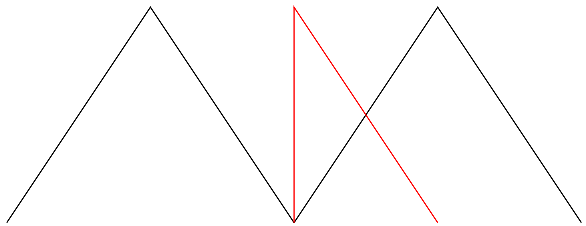
Forbidden structures I: cycles



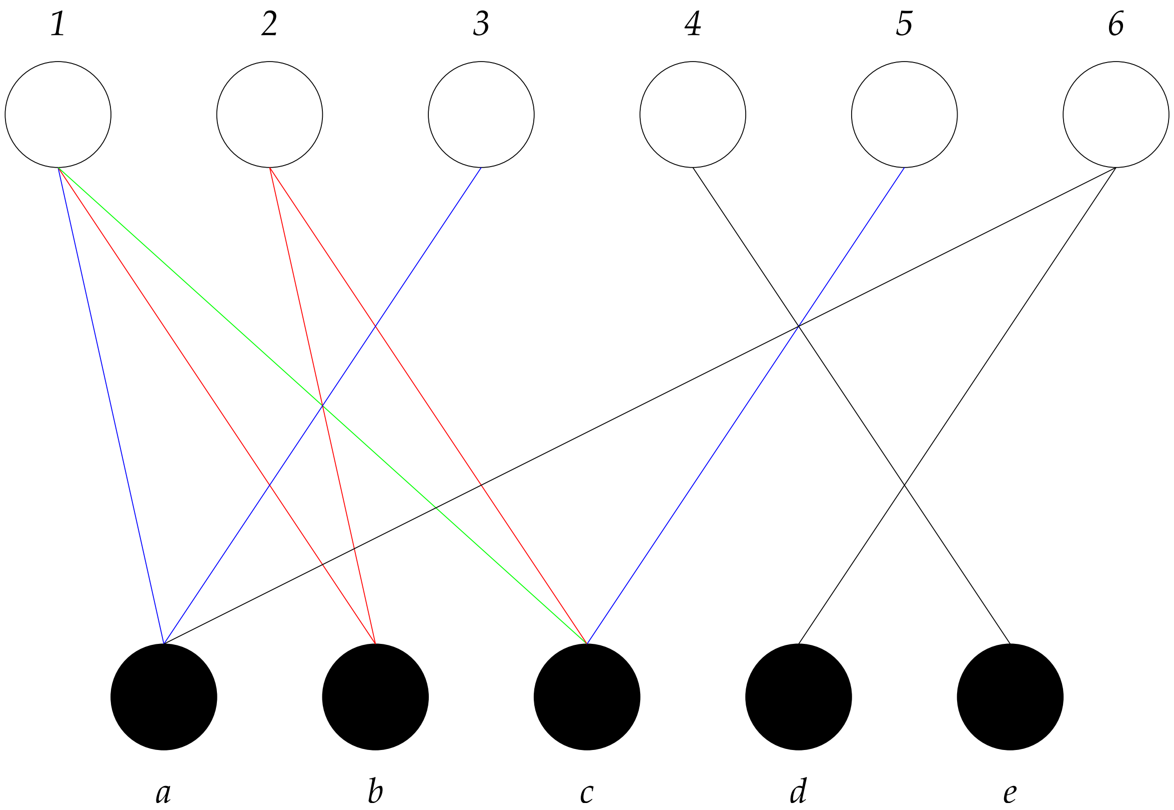
Forbidden structures II: 2-claws



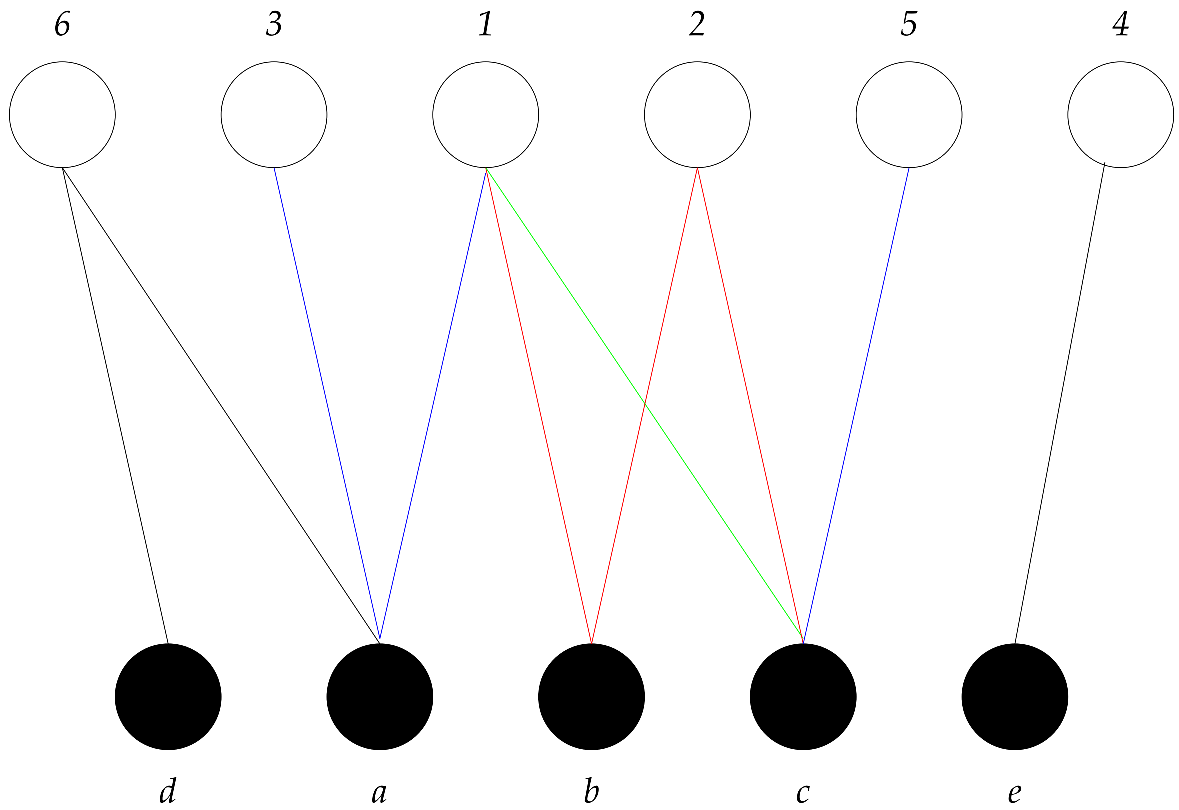
Forbidden structures IIa: 2-claws



Forbidden structures in our example



A possible solution



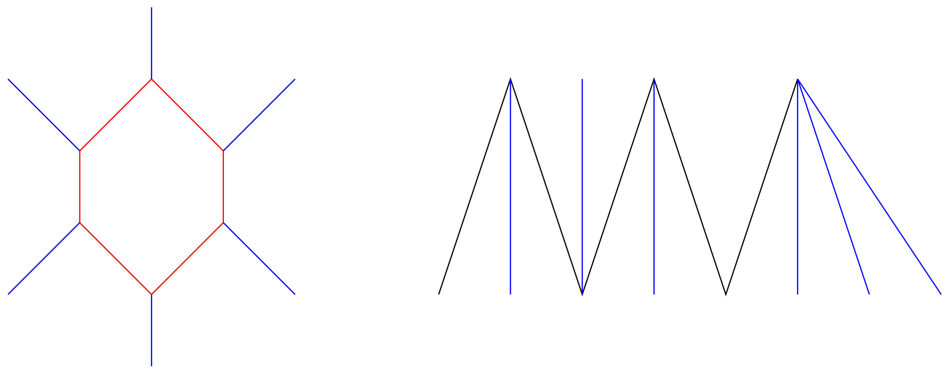
Small forbidden structures (D. et al.)

The *non-leaf degree* of a vertex v in graph G is the number of non-leaf edges at v in G, and is denoted by $\deg'_G(v)$.

Lemma : If there exists a vertex v in a graph G such that $\deg'_G(v) \geq 3$, then G contains a 2-claw or a 3- or 4-cycle containing v.

Otherwise: trivial (but some work to be done).

Otherwise: Wreaths and Caterpillars



Search tree algorithm I

Require: a graph $G = (V, E)$, a positive integer k
Ensure: YES iff there is a biplanarization set $B \subseteq E$, $|B| \leq k$

if $\exists v \in V$: $\deg'_G(v) \geq 3$ and $k \geq 0$ **then**
  Determine 2-claw, 3-cycle or 4-cycle C containing v.
  Branch at all edges $e \in C$.
else
  return $k \geq \#$ wreath components
end if

Search tree algorithm II

Any ideas for improvements?

- Oberserve similarity with 6-HITTING SET.

- BUT: return $k \geq \#$ wreath components

- Try to mimic best 6-HITTING SET algorithm!
  Can the corresponding reduction rules be translated?

Reduction rules

1. structure domination: A forbidden structure f is *dominated* by another structure f' if $c(f') \subset c(f)$. Then, mark f as virtual.

2. small structures: If $|c(f)| = 1$, put the only non-virtual edge into the solution that is constructed.

3a isolates: If e is an edge of degree zero, then mark e virtual.

$c(f)$: non-virtual edges of f
BUT: How to avoid branching at "edges of small degree"?
$\leadsto$ another 2-claw specific rule (omitted)

Algorithm 1 **Search tree algorithm III**

Require: a graph $G = (V, E)$, a positive integer k

Ensure: YES iff there is a biplanarization set $B \subseteq E$, $|B| \leq k$

Exhaustively apply reduction rules
if $\exists v \in V$: $\text{deg}'_G(v) \geq 3$ and $k \geq 0$ **then**
    Determine non-dominated 2-claw, 3-cycle or 4-cycle C with smallest number of non-virtual edges.
    Determine edge $e \in C$ with largest number of non-dominated forbidden structures containing e.
    Binary branch at $e \in C$
    {turning e virtual if e is not assumed to be in the biplan. set}
else
    return $k \geq \#$ wreath components
end if

The time analysis

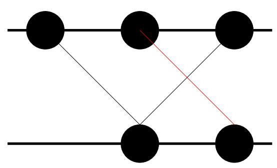
$T^\ell(k)$: size of a search tree assuming that $\geq \ell$ forbidden structures in the given instance (with parameter k) have size ≤ 5, i.e.: $|c(f)| \leq 5$.

$$T^0(k) \leq T^0(k-1) + T^2(k).$$
$$T^1(k) \leq 2T^0(k-1) + 2T^1(k-1) + T^2(k-1).$$
$$T^2(k) \leq \max \left\{ \begin{array}{l} 2T^1(k-1) + 3T^2(k-1), \\ T^0(k-1) + 16T^0(k-2), \\ 2T^0(k-1) + 9T^0(k-2), \\ 3T^0(k-1) + 4T^0(k-2), \\ 4T^0(k-1) + T^0(k-2) \end{array} \right\}.$$

Lemma: $T^0(k) \leq 5.1926^k$

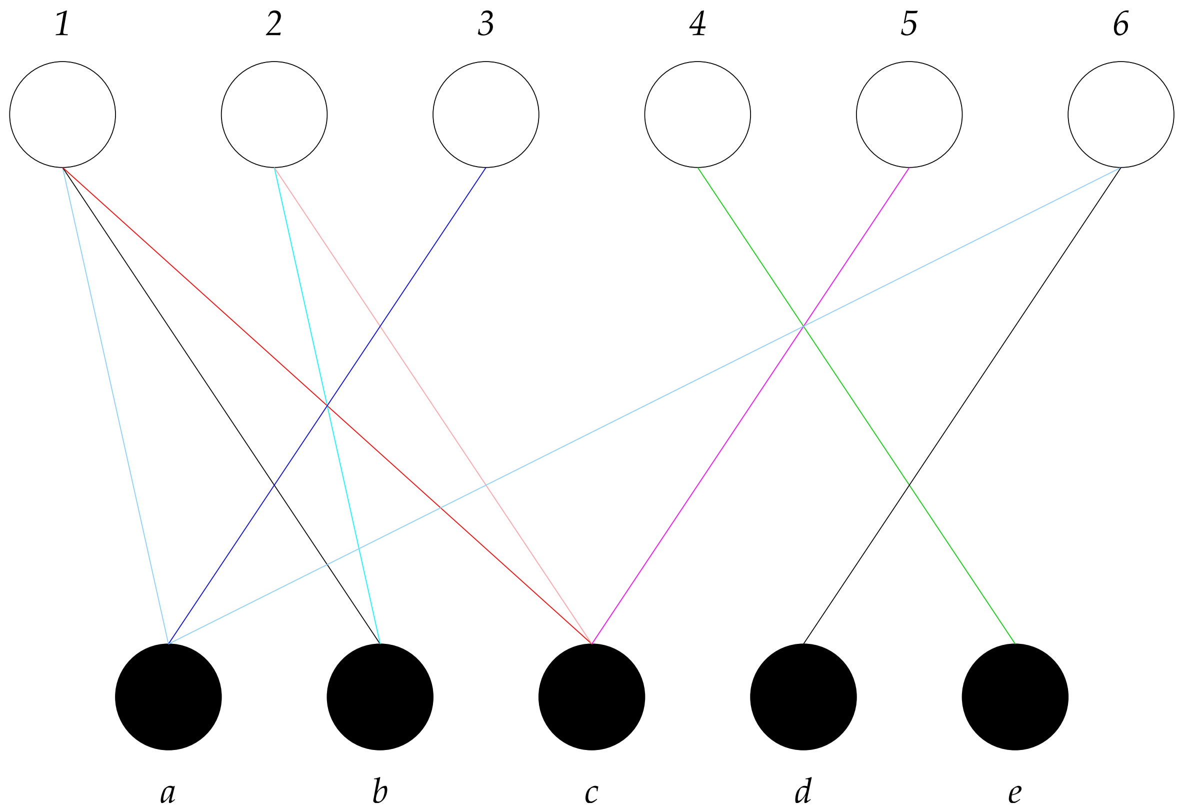
ONE-LAYER PLANARIZATION

- Relation to 3-HITTING SET.

- New forbiden structure: 

- The translation of (kernelization) rules for 3-HITTING SET (basically) also yields a kernelization for OLP.

Forbidden structures in our example



Summary of 2-Layer Drawing Problems

	time $\mathcal{O}^*(\cdot)$	kernel size $\mathcal{O}(\cdot)$	where?
TLP	5.1926^k	k	this paper
OLP	2.5616^k	k^2	this paper
OSCM	1.4656^k	$(2+\epsilon)k^2$	[DujFerKauGD04]
TSCM	$2^{32(2+2k)^3}$	??	[DujetalESA01]

OSCM= ONE-SIDED CROSSING MINIMIZATION

TSCM= TWO-SIDED CROSSING MINIMIZATION