On the Segment Number of a Planar Graph

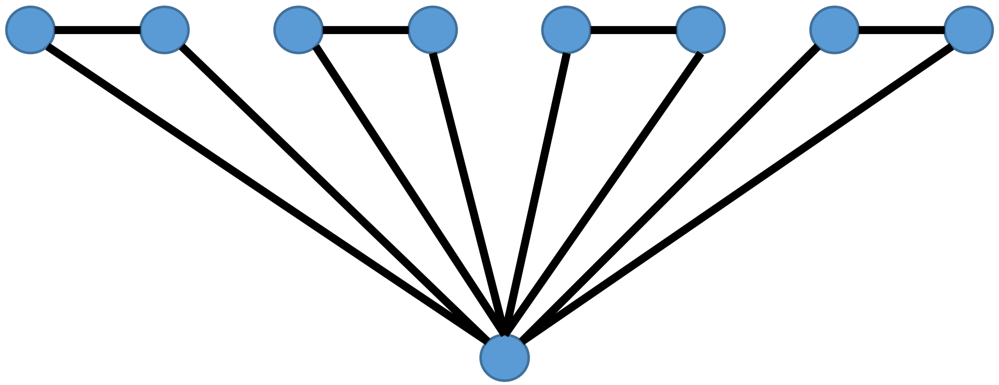
Jan Kratochvíl

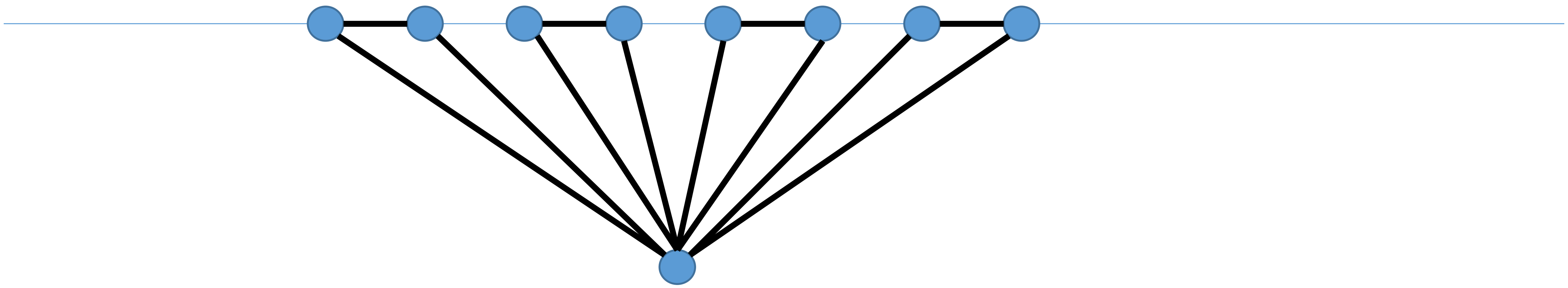
Charles University, Prague, Czech Republic

(joint work with Sabine Cornelsen, Giordano Da Lozzo, Luca Grilli, Siddharth Gupta, and Alexander Wolff)

CSGT 2024

Trojanovice, June 4, 2024

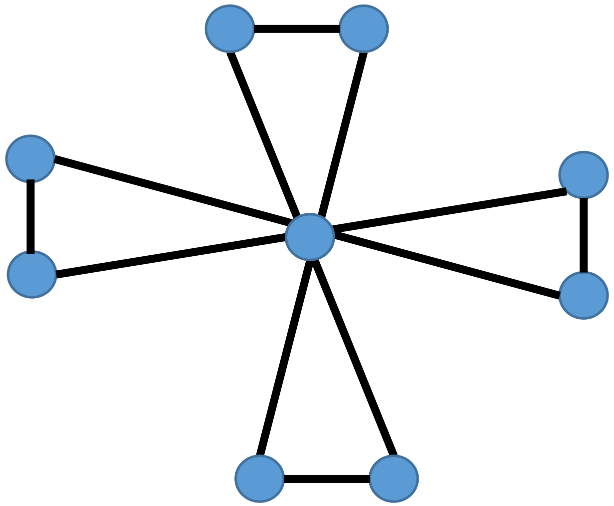




9 lines

12 segments

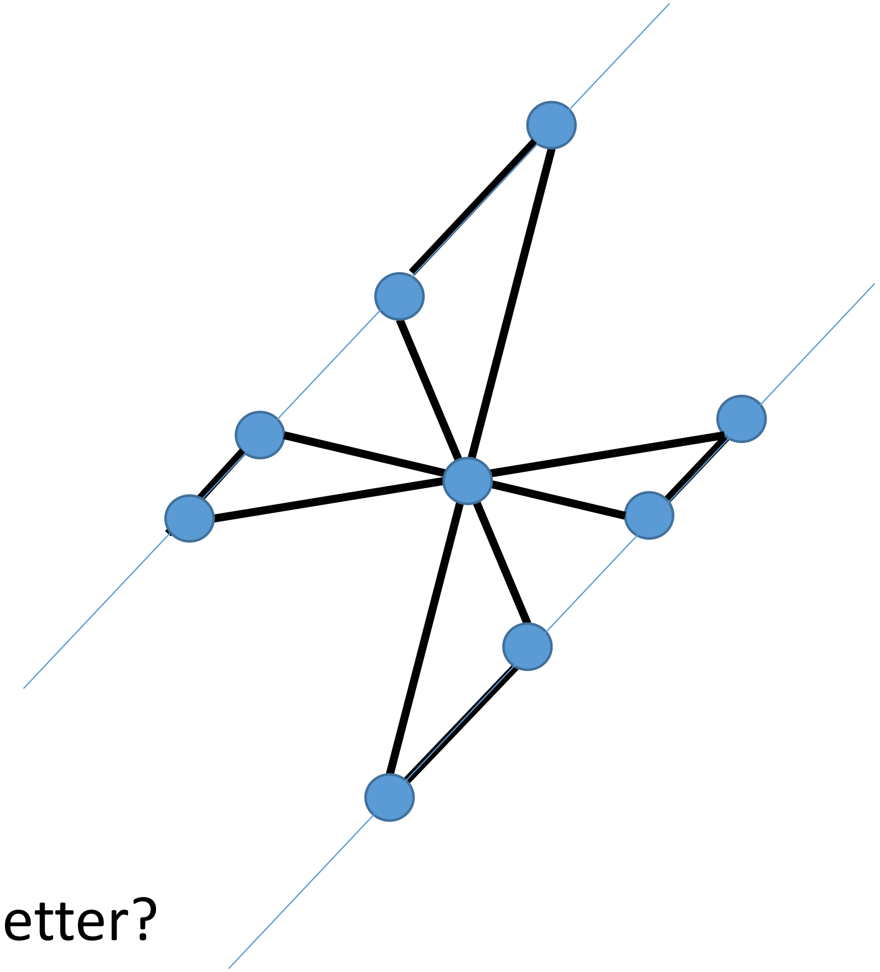
Can we do better?



8 lines

8 segments

Can we still do better?



6 lines

8 segments

Can we still do better?

line(G) = minimum number of lines covering all edges in a planar drawing of G

seg(G) = minimum number of segments containing all edges in a planar drawing of G

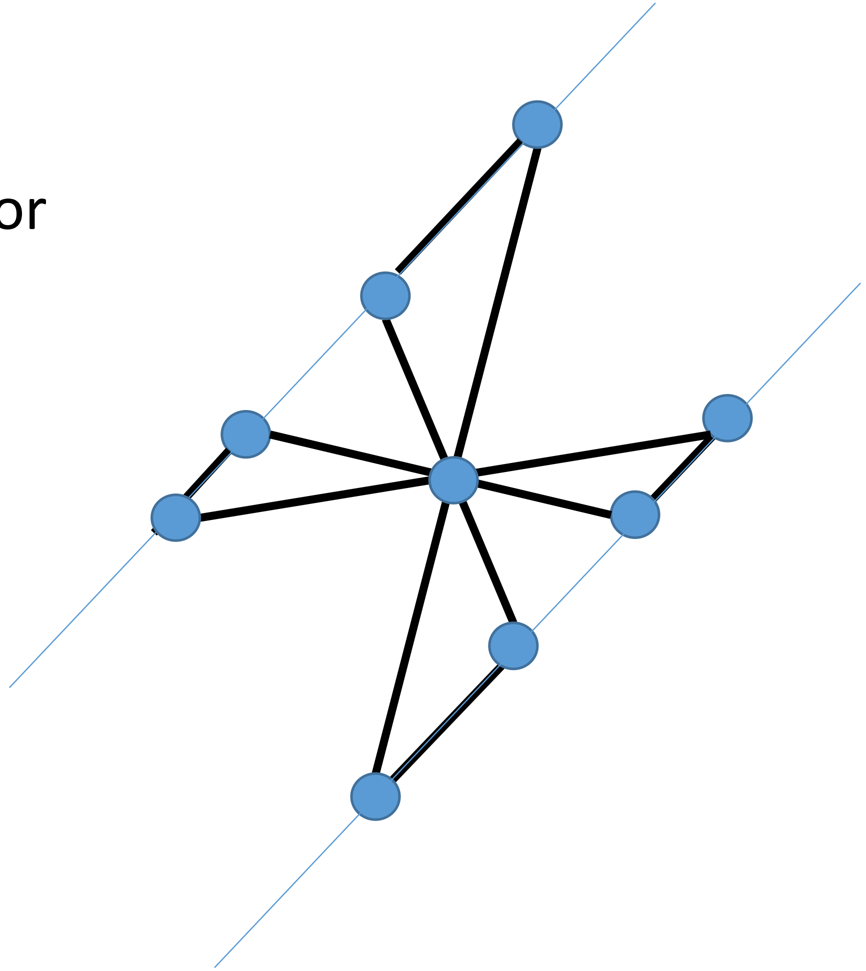
Dujmovic, Eppstein, Suderman, Wood 2007:

seg(G) can be computed efficiently for trees and for 3-connected cubic planar graphs

Okamoto, Ravsky, Wolff GD 2019

Determining seg(G) is ∃R-complete (and hence NP-hard)

Our concern - FPT



FPT – Fixed Parameter Tractability

Input is structured – An instance of size n and a parameter k (positive integer)

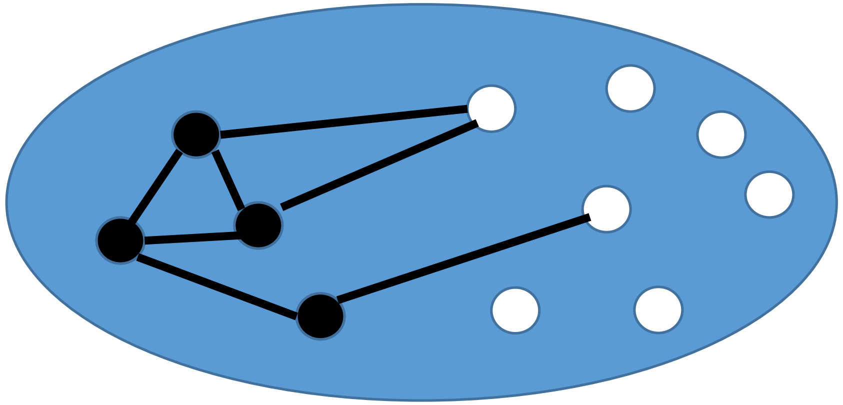
FPT algorithm – running time $f(k)\, n^c$ for a constant c independent of k and n, also written as $f(k)\, O^*(n)$

FPT – Fixed Parameter Tractability

Input is structured – An instance of size n and a parameter k (positive integer)

FPT algorithm – running time $f(k)\, n^c$ for a constant c independent of k and n, also written as $f(k)\, O^*(n)$

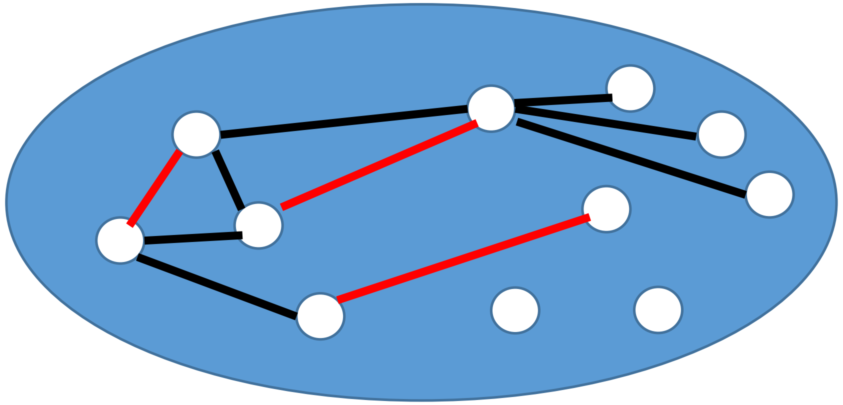
Classical example – vertex cover = complement of independent set



Is there a vertex cover of size $\leq k$?

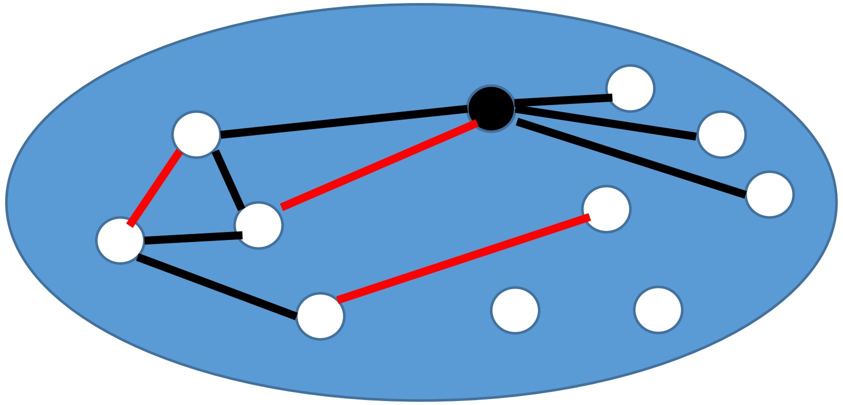
Vertex Cover is FPT

1. Find a maximum matching M. If $|M| > k$, then output NO and stop.



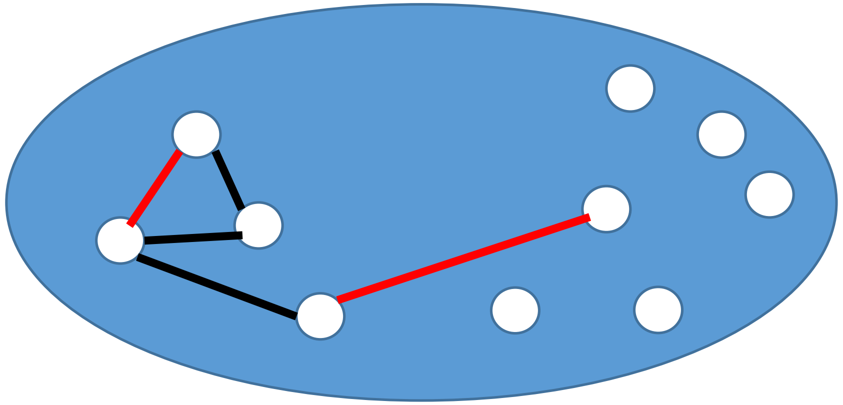
Vertex Cover is FPT

1. Find a maximum matching M. If $|M| > k$, then output NO and stop.
2. If a vertex u of M has $> k$ neighbors outside of M, put u in the vertex cover and delete it. Set $k \leftarrow k$-1. Repeat.



Vertex Cover is FPT

1. Find a maximum matching M. If $|M| > k$, then output "NO" and stop.
2. If a vertex u of M has $\geq k$ neighbors outside of M, put u in the vertex cover and delete it. Set $k \leftarrow k$-1. Repeat.
3. We are left with a graph which has at most $2k^2$ vertices of degree > 0. Solve by brute force.
4. Total running time is $\binom{2k^2}{k} O(n^2) O(nm)$

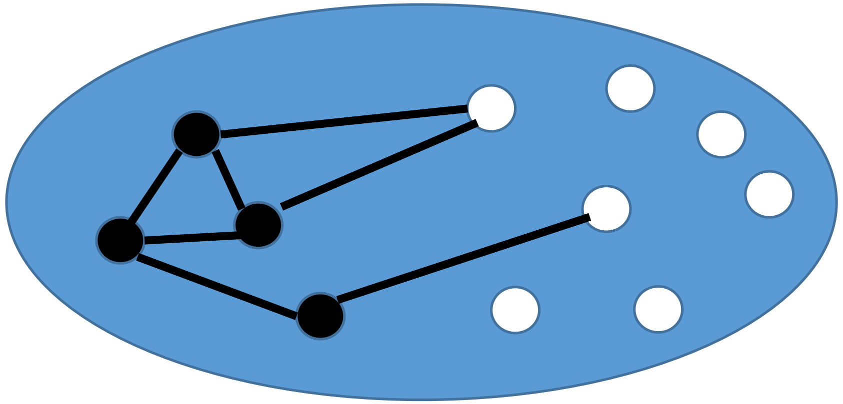


Vertex Cover is FPT

Classical example – vertex cover = complement of independent set

A graph has a vertex cover of size $\leq k$ iff it has an independent set of size $\geq n$-k.

"Is there an independent set of size $\geq k$?" is not FTP, it is W[1]-hard.



Our results

Thm: **Seg(*G*)** $\leq k$ is FPT when parameterized by

- *k* (natural parameter)
- line(*G*)
- vertex cover number of *G*

Thm: **List incidence line cover number** and **list incidence segment number** are FPT when parameterized by the natural parameter.

Determining seg(*G*) is FPT when parameterized by line(*G*)

Determining seg(*G*) is FPT when parameterized by line(*G*)

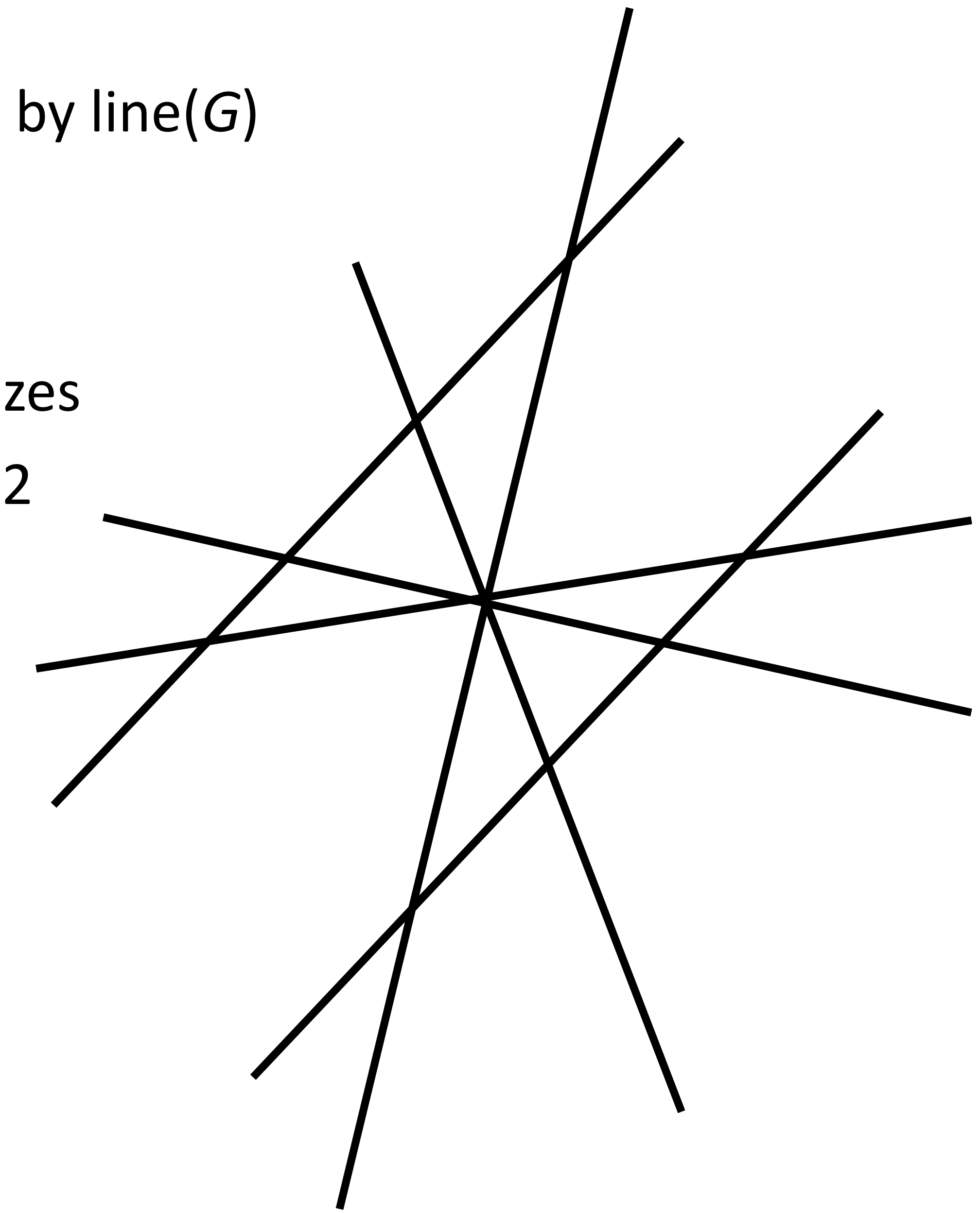
line(*G*) $\leq$ seg(*G*),

hence this implies that seg(*G*) is FPT also when parameterized by seg(*G*)

Determining seg(*G*) is FPT when parameterized by line(*G*)

line(*G*) = *k*

Consider a line arrangement of *k* lines that realizes a drawing of *G*. Vertices of degree greater than 2 must be placed in crossing points of the lines, hence G can only have $\leq \binom{k}{2}$ such vertices.

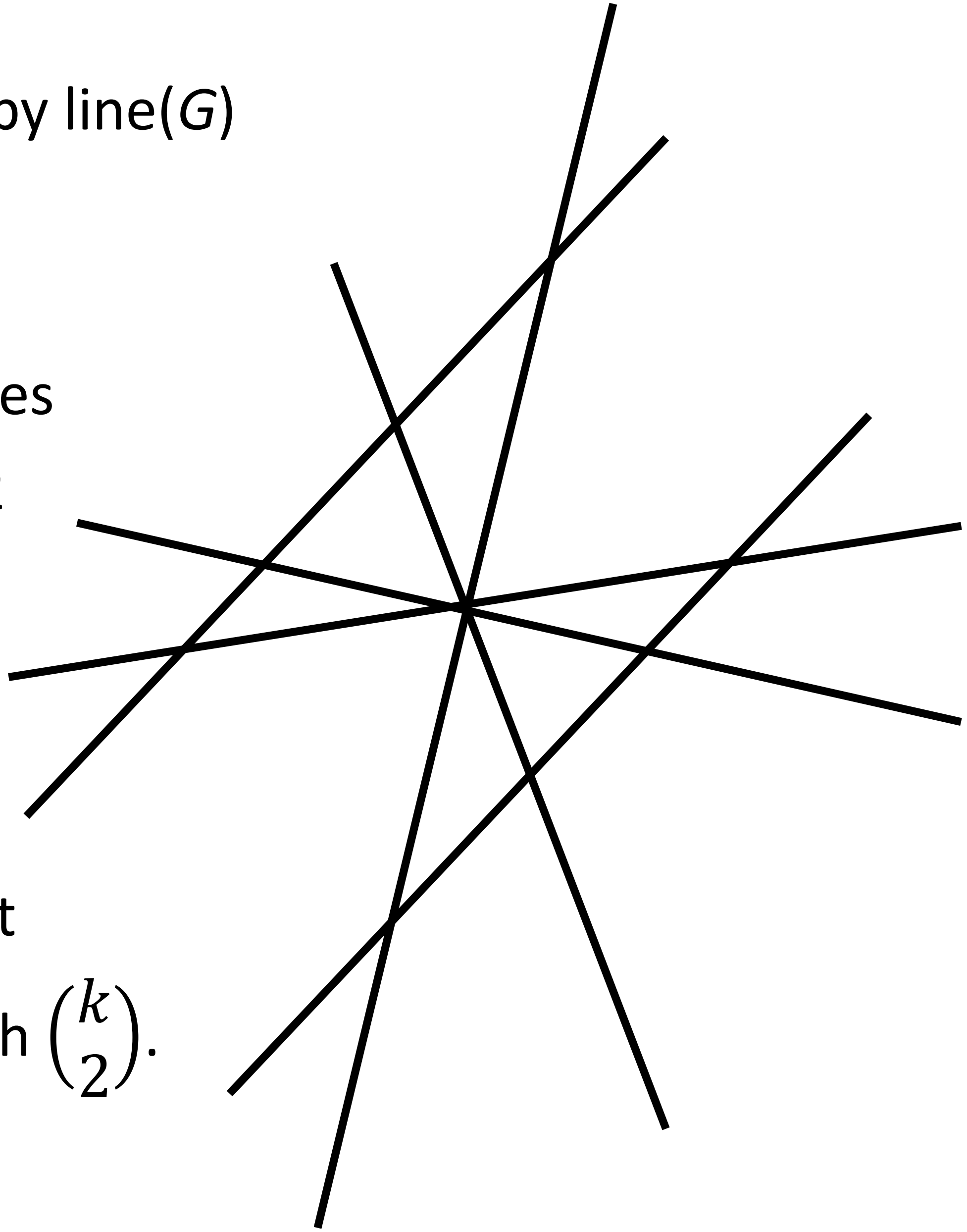


Determining seg(*G*) is FPT when parameterized by line(*G*)

line(*G*) = *k*

Consider a line arrangement of *k* lines that realizes a drawing of *G*. Vertices of degree greater than 2 must be placed in crossing points of the lines, hence G can only have $\leq \binom{k}{2}$ such vertices.

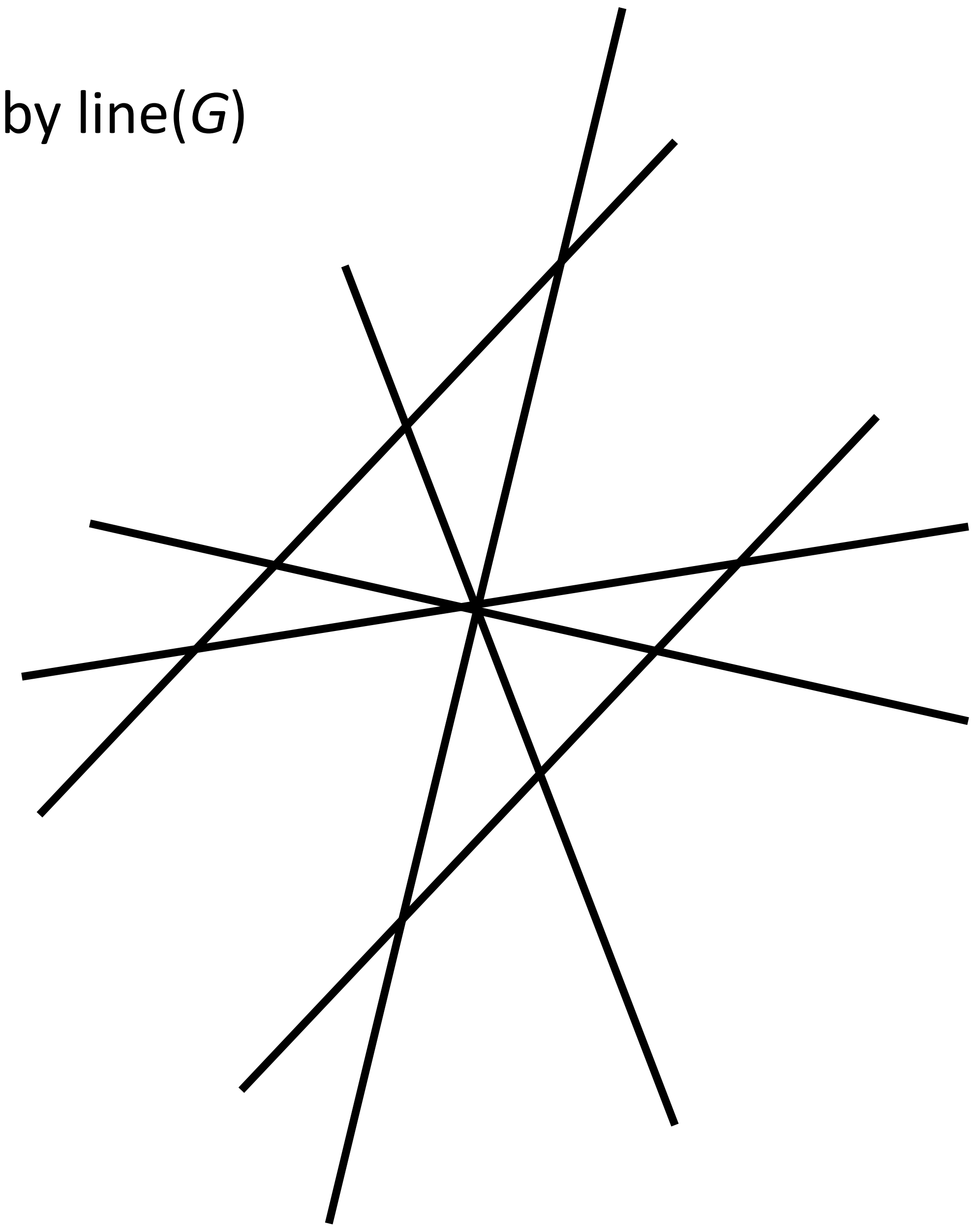
Paths consisting of vertices of degree 2 can only bend in crossing points, and so have at most $\binom{k}{2}$ bends. Replace long paths by paths of length $\binom{k}{2}$.



Determining seg(G) is FPT when parameterized by line(G)

line(G) = k

Max degree is $\leq 2k$. Thus the input is reduced to a graph with at most $\binom{k}{2} + k\binom{k}{2}$ vertices.

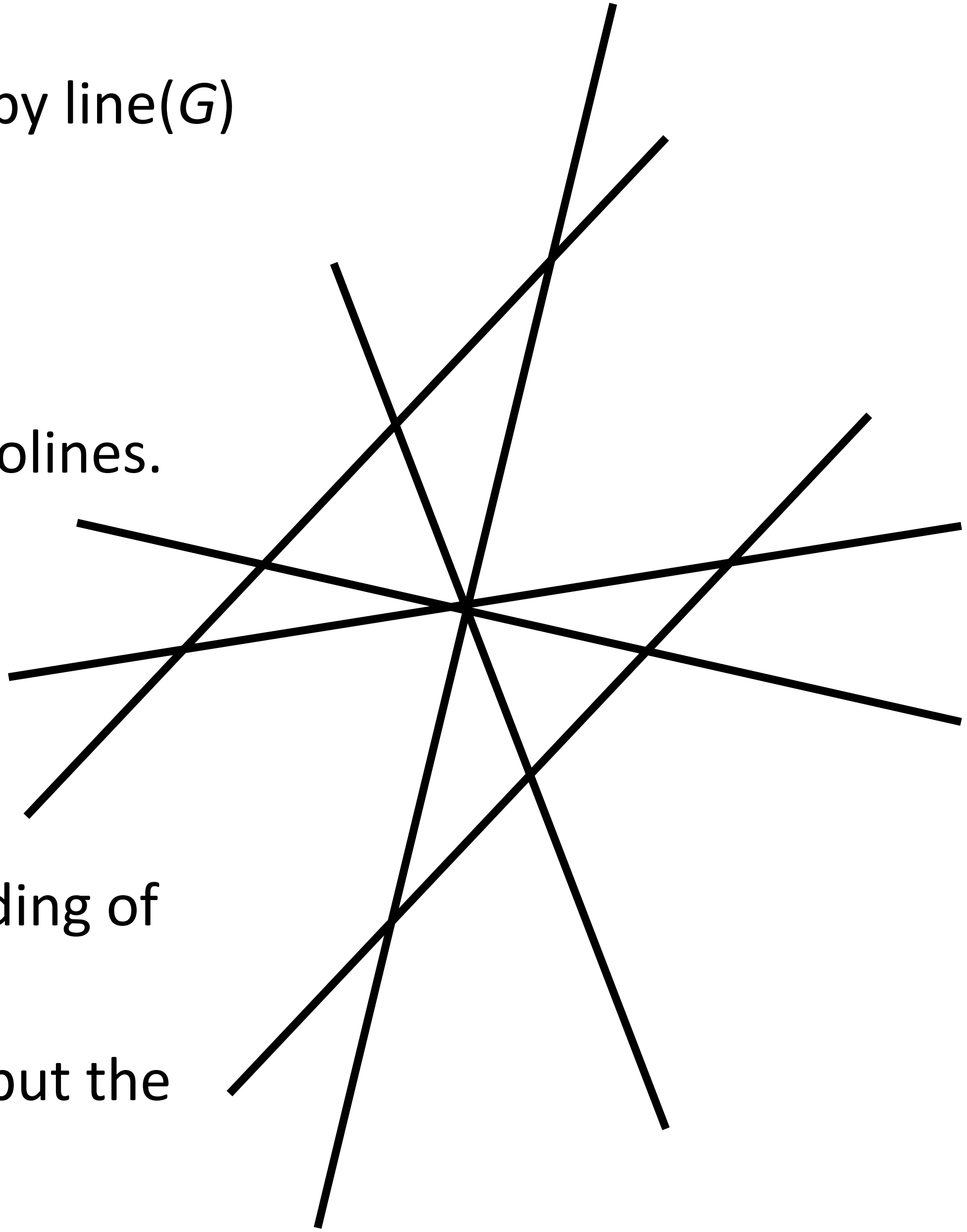


Determining seg(G) is FPT when parameterized by line(G)

line(G) = k

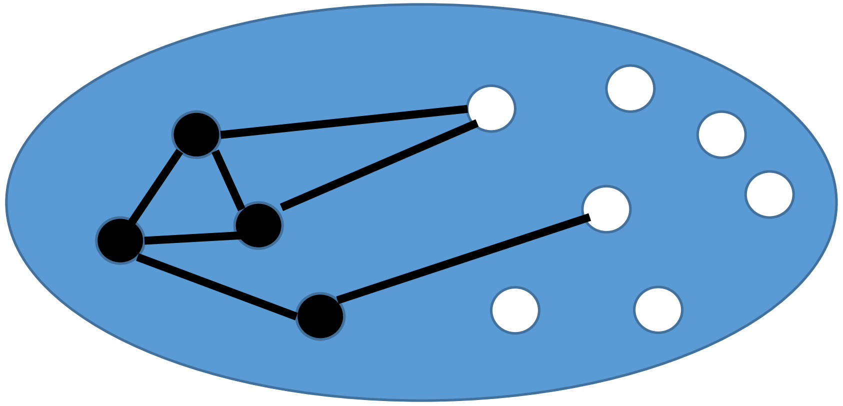
Algorithm:

1. Try all combinatorial arrangements of k pseudolines. There are at most $(k!\binom{2k}{k})^k$ of them.
2. For each of them, check if it is stretchable (use Renegar [1992]).
3. If stretchable, try all possible assignments of high degree vertices to crossing points, and bending of paths with vertices of degree 2. In $F(k)$ ways.
4. Count the number of segments, and then output the minimum one.



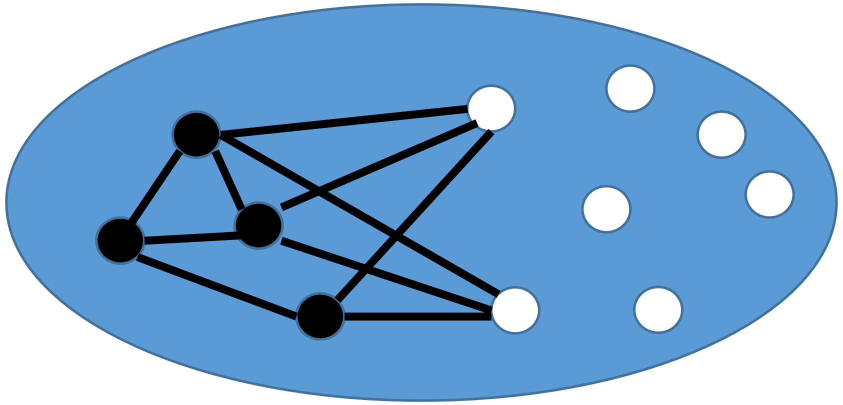
Determining seg(*G*) is FPT when parameterized by the vertex cover number of *G*

$A \subseteq V(G)$ a vertex cover, $|A|=k$.



Determining seg(G) is FPT when parameterized by the vertex cover number of G

$A \subseteq V(G)$ a vertex cover, $|A|=k$.

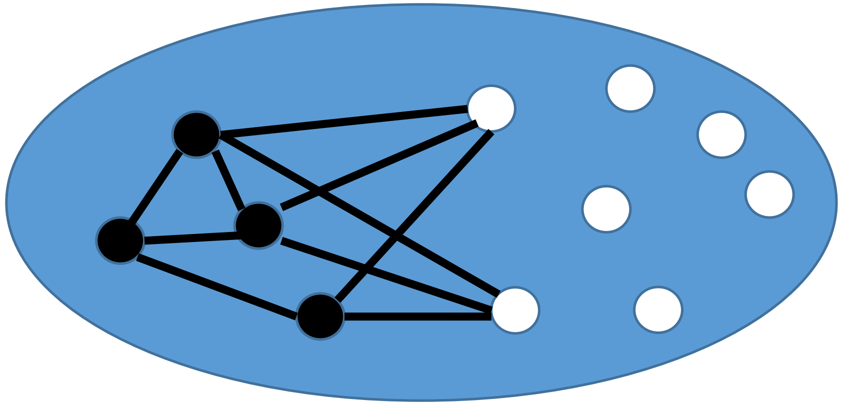


1. Bounding the number of high degree vertices.

Call two vertices from V-A equivalent if they have the same set of neighbors in A. A j-vertex has exactly j neighbors in A. Observation: For every $j>2$, there are at most 2 equivalent j-vertices. Otherwise, G contains $K_{3,3}$ and is non-planar. Hence G has at most 2^{k+1} vertices of degree >2.

Determining seg(*G*) is FPT when parameterized by the vertex cover number of *G*

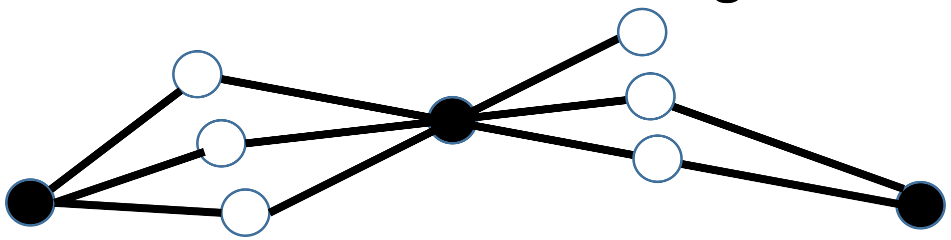
$A \subseteq V(G)$ a vertex cover, $|A|=k$.



1. Bounding the number of high degree vertices.

Call two vertices from V-A equivalent if they have the same set of neighbors in A. A j-vertex has exactly j neighbors in A. Observation: For every $j>2$, there are at most 2 equivalent j-vertices. Otherwise, G contains $K_{3,3}$ and is non-planar. Hence G has at most 2^{k+1} vertices of degree >2.

2. Try to pair degree 2 vertices to maximize "alignments". Details omitted.



List incidence segment number and list incidence line cover number of *G*

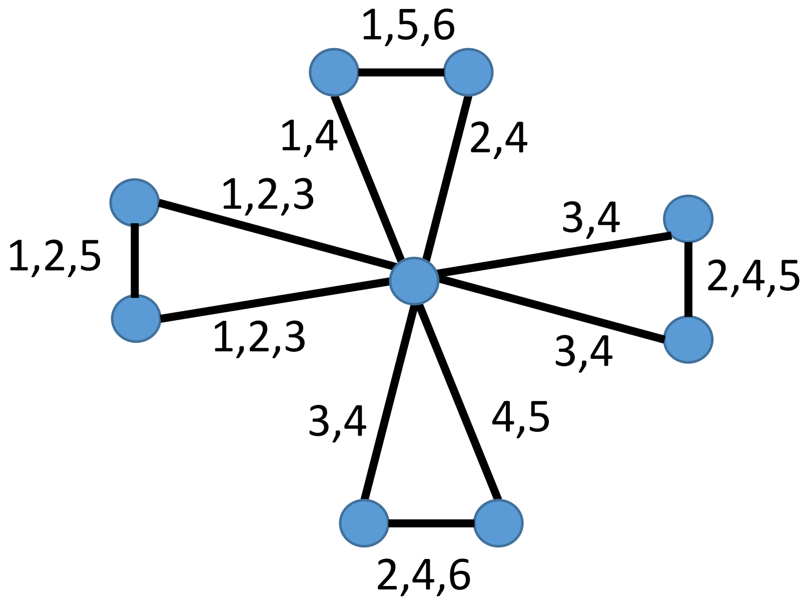
Input: A planar graph *G* and for every edge $e \in E(G)$, a set $S(e) \subseteq \{1,2,\dots,k\}$.

Question: Is there a planar drawing of *G* with *k* segments/lines and a numbering of these segments/lines by $1,2,\dots,k$ such that for every edge *e,* the following holds true: If *e* is drawn on segment/line *j*, then $j \in S(e)$?

List incidence segment number and list incidence line cover number of *G*

Input: A planar graph G and for every edge $e \in E(G)$, a set $S(e) \subseteq \{1,2,\ldots,k\}$.

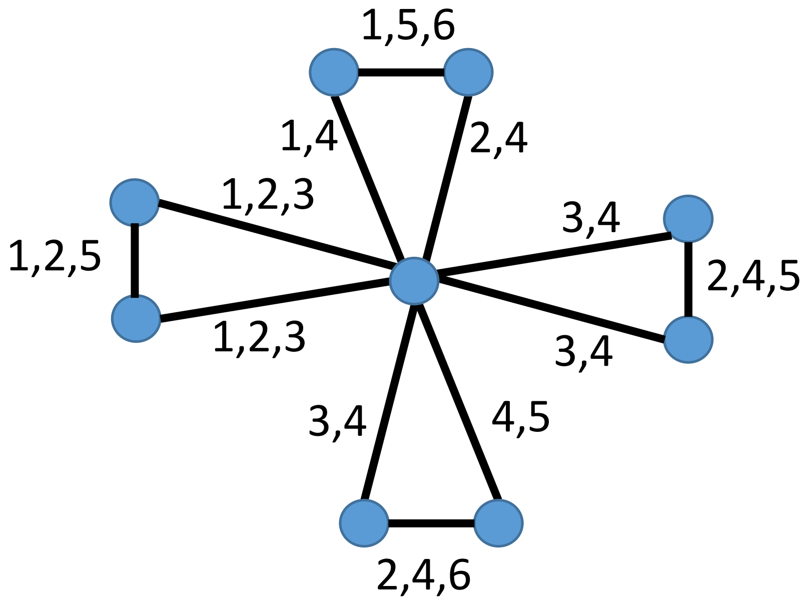
Question: Is there a planar drawing of G with k segments/lines and a numbering of these segments/lines by $1,2,\ldots,k$ such that for every edge e, the following holds true: If e is drawn on segment/line j, then $j \in S(e)$?

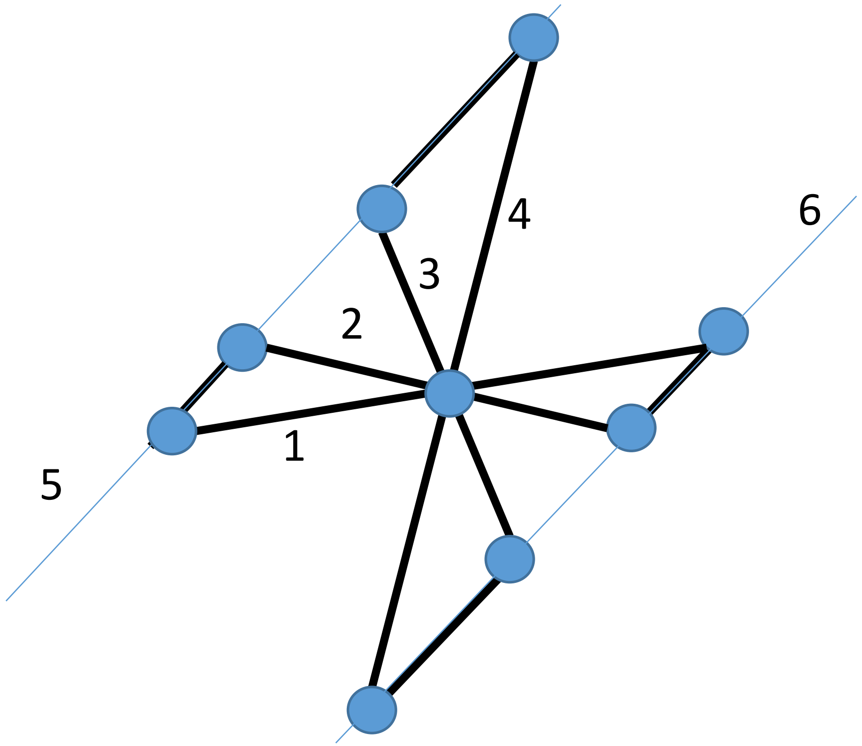


List incidence segment number and list incidence line cover number of *G*

Input: A planar graph *G* and for every edge $e \in E(G)$, a set $S(e) \subseteq \{1,2,\ldots,k\}$.

Question: Is there a planar drawing of *G* with *k* segments/lines and a numbering of these segments/lines by 1,2,…,*k* such that for every edge *e*, the following holds true: If *e* is drawn on segment/line *j*, then $j \in S(e)$?





List incidence segment number and list incidence line cover number of *G* are FPT when parameterized by *k*.

List incidence segment number and list incidence line cover number of *G* are FPT when parameterized by *k*.

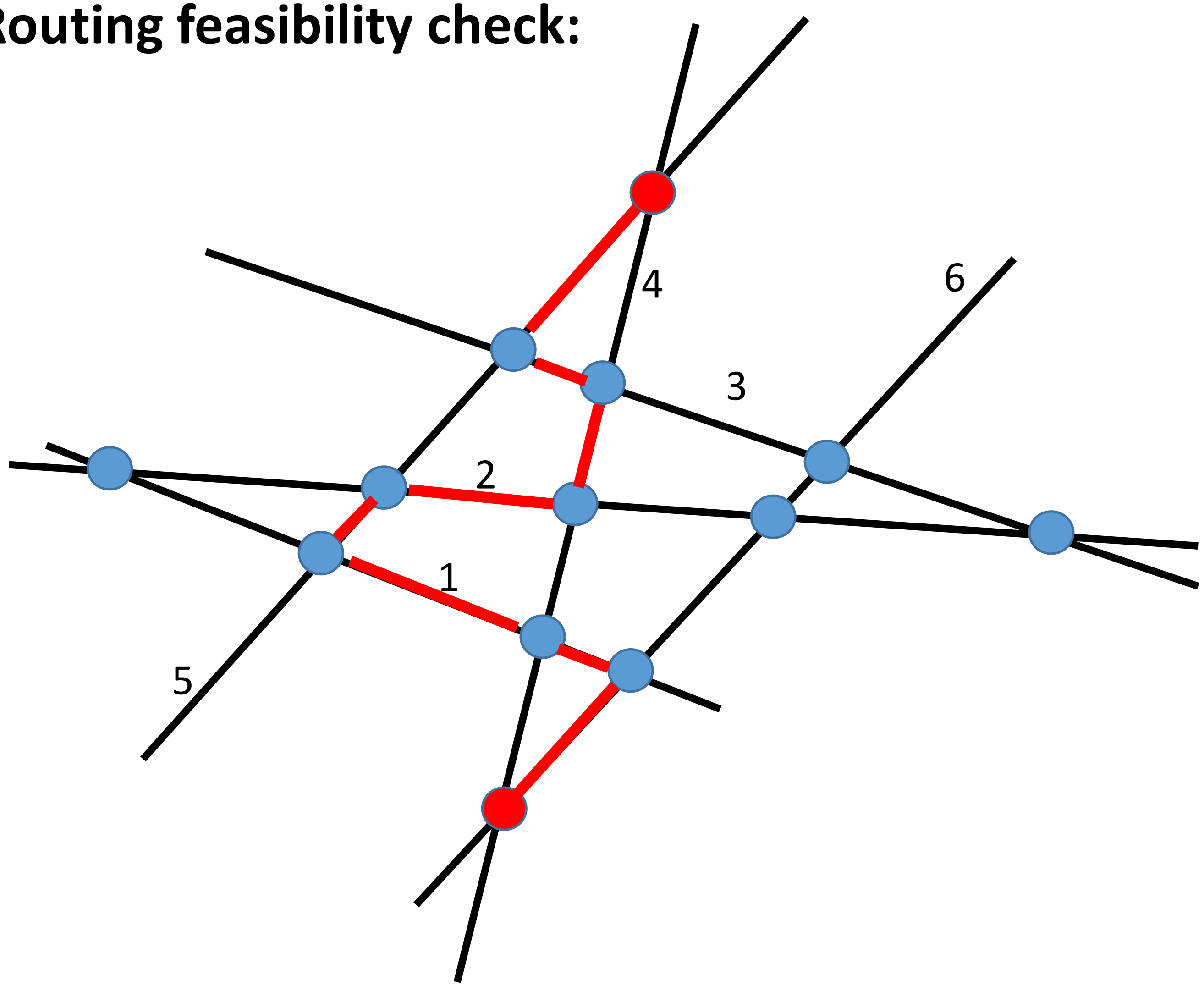
Algorithm:

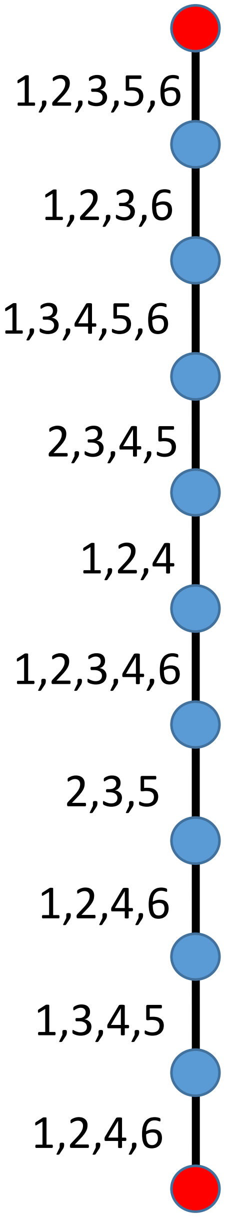
1. Bound the number of vertices of degree >2
2. Try all possible line arrangements and assignments of high degree vertices to crossing points.
3. Try all possible routings of the paths with internal vertices of degree 2.

Comment: So far the number of possibilities is only a function of *k*.

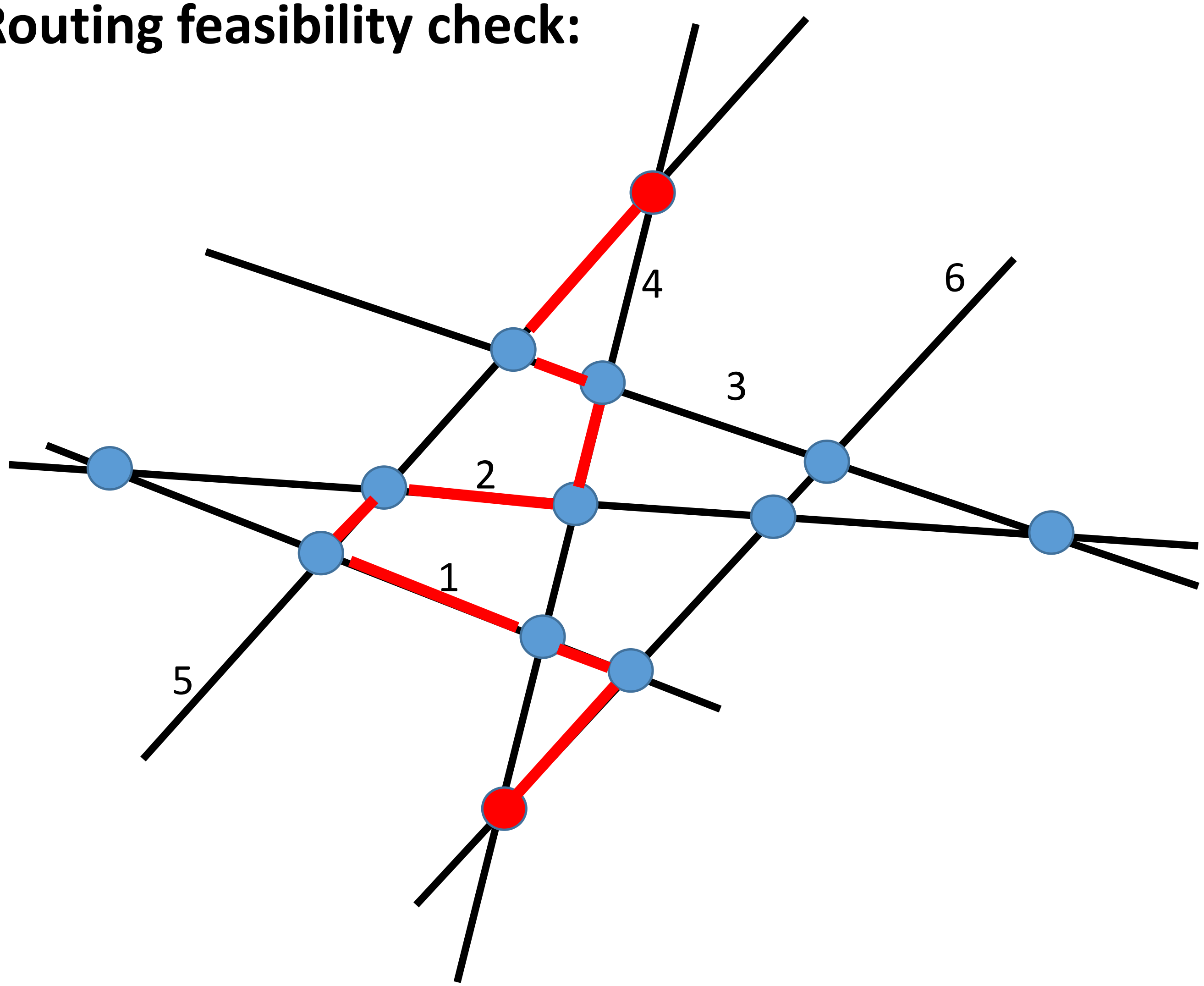
4. For each path, check by dynamic programming if the prescribed routing is compatible with the lists at its edges.

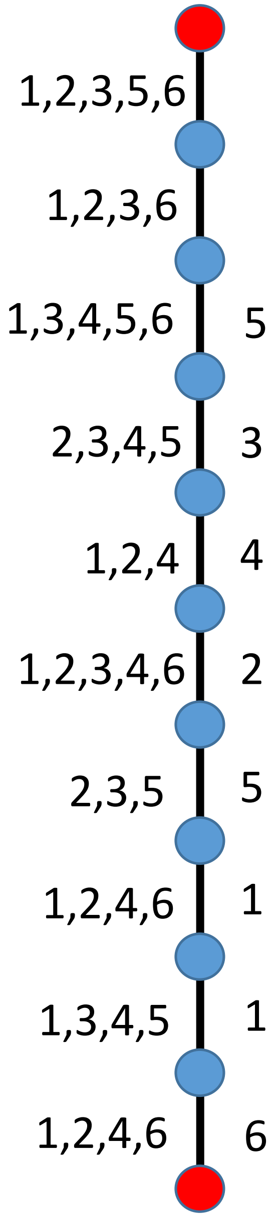
Routing feasibility check:



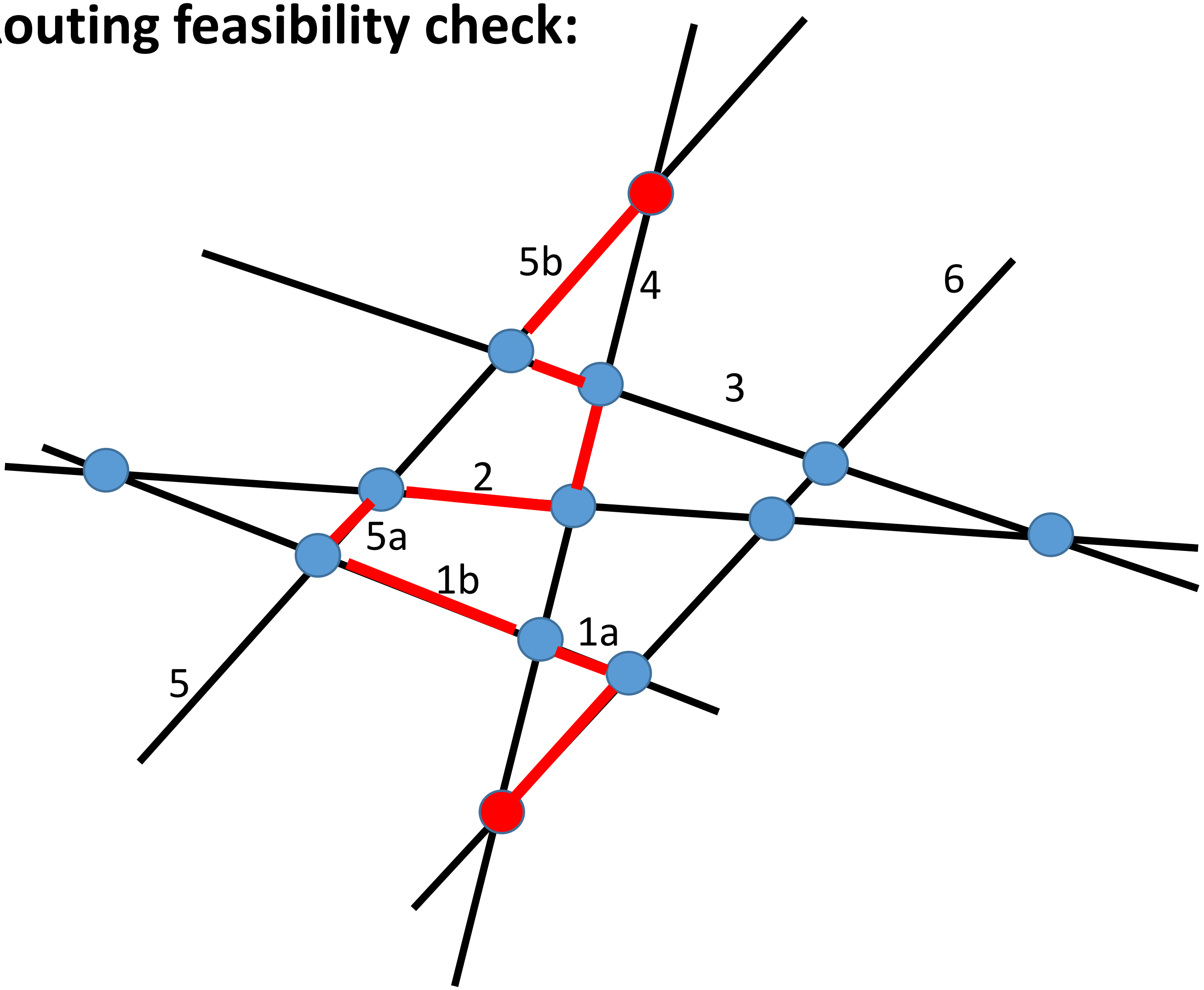


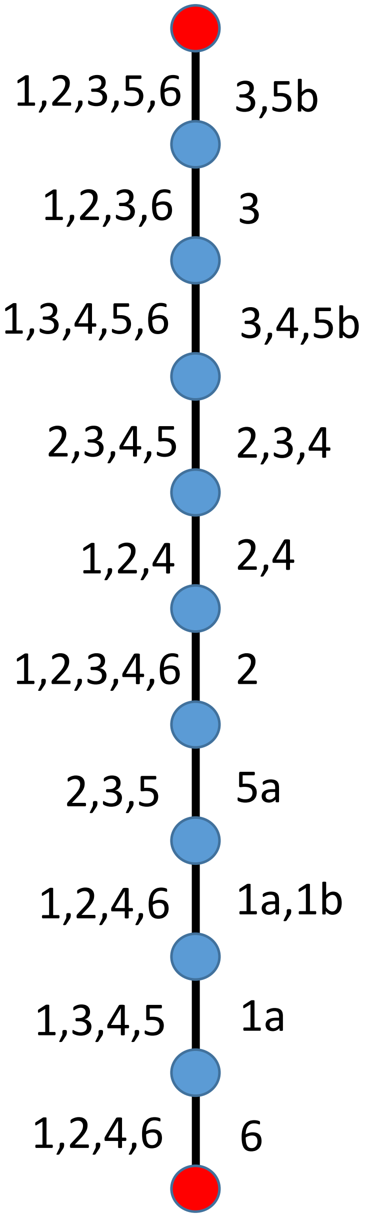
Routing feasibility check:



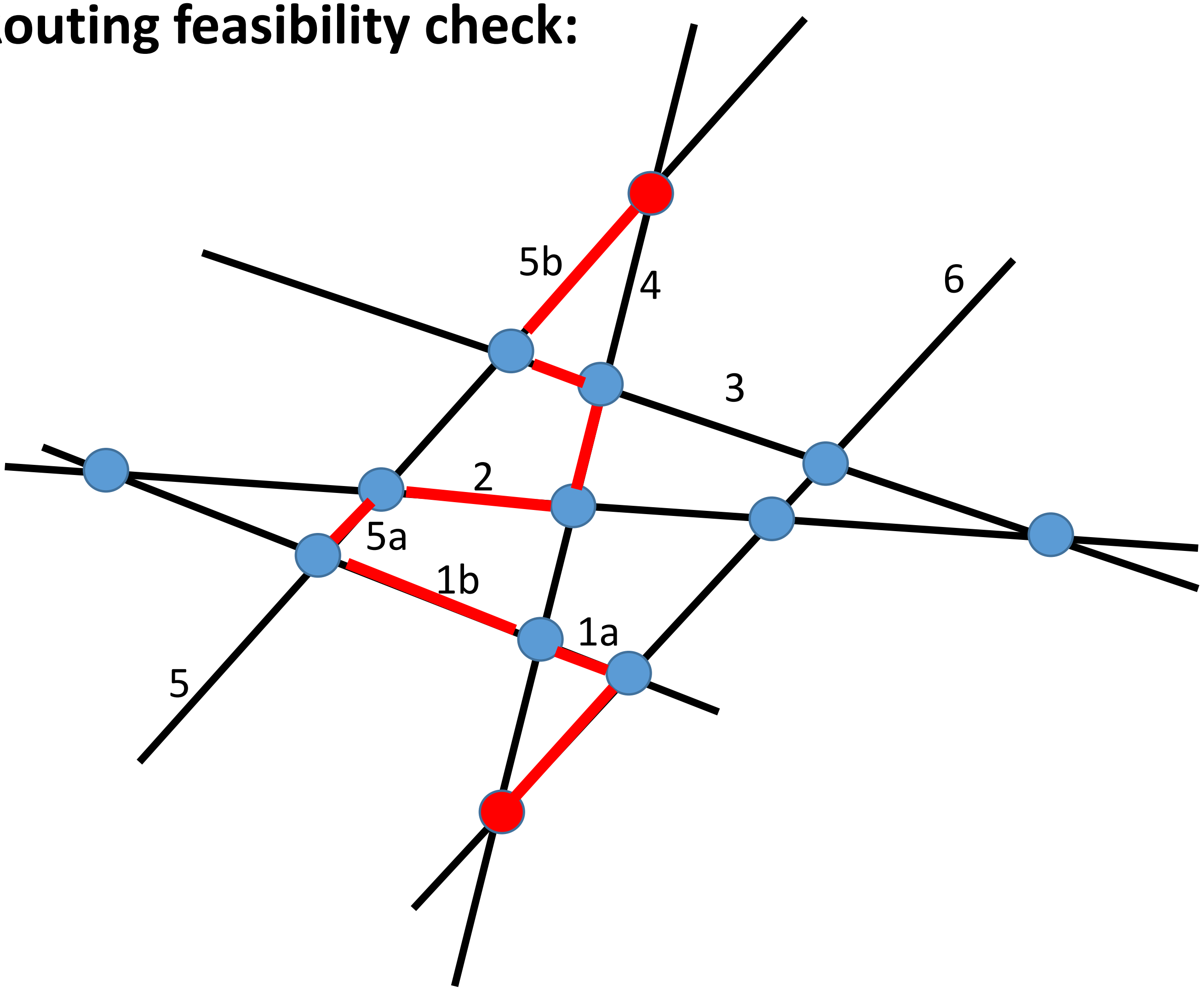


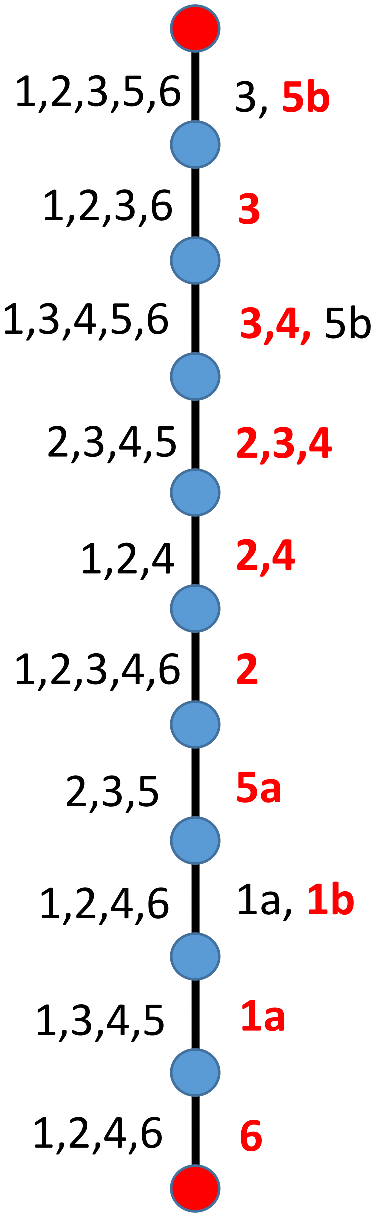
Routing feasibility check:



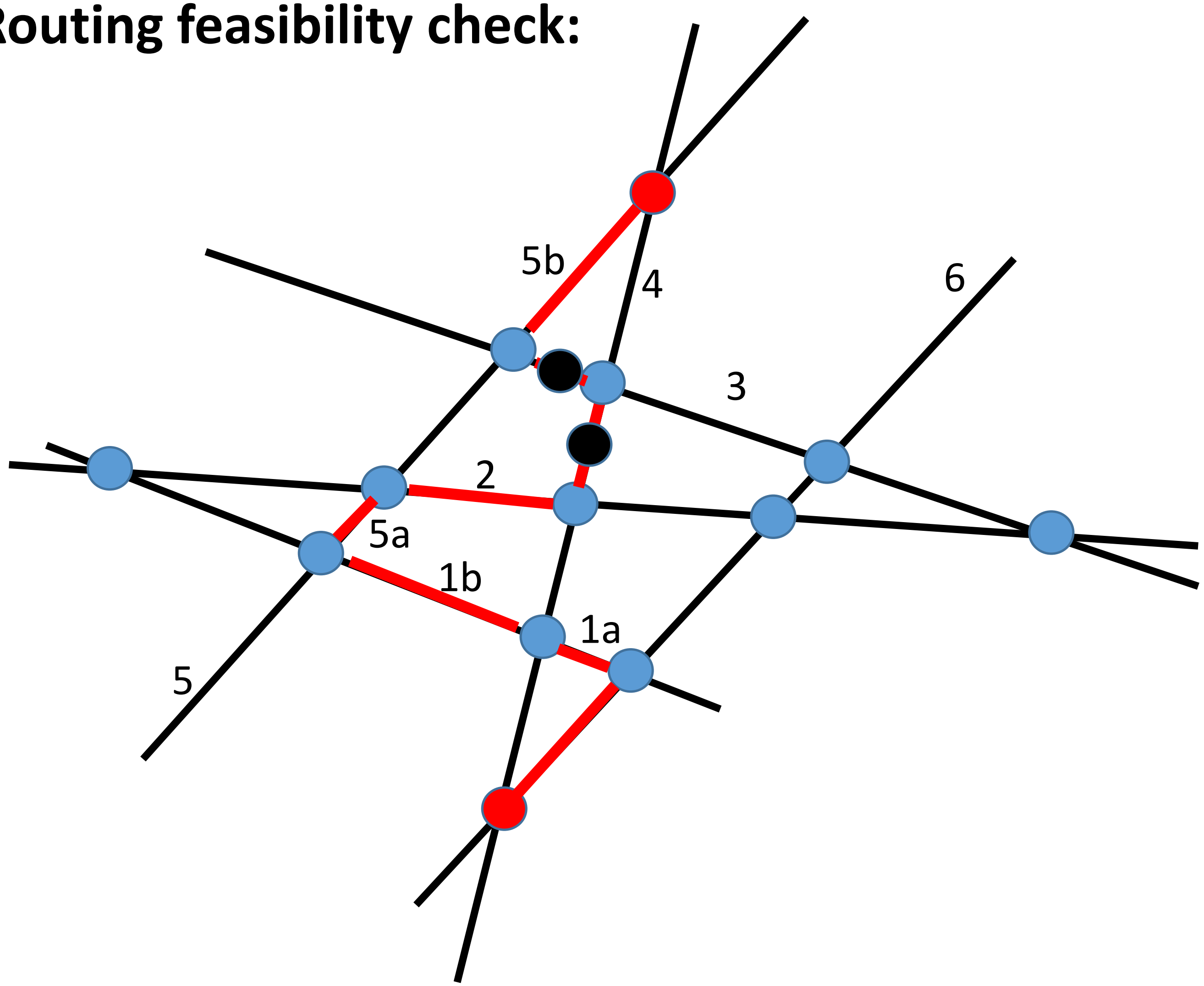


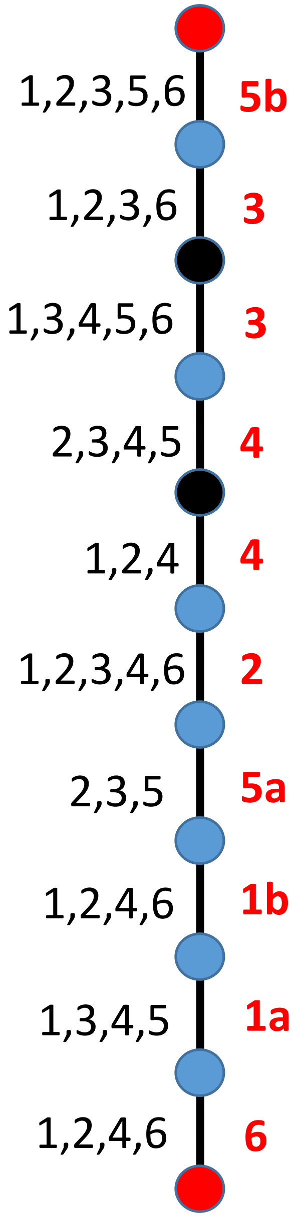
Routing feasibility check:





Routing feasibility check:





List incidence segment number and list incidence line cover number of *G* are FPT when parameterized by *k*.

Algorithm:

1. Bound the number of vertices of degree >2
2. Try all possible line arrangements and assignments of high degree vertices to crossing points.
3. Try all possible routings of the paths with internal vertices of degree 2.

Comment: So far the number of possibilities is only a function of *k*.

4. For each path, check by dynamic programming if the prescribed routing is compatible with the lists at its edges.

Running time: Extra $O(n)$ factor for the dynamic programming.

Final Comments and Open Problem

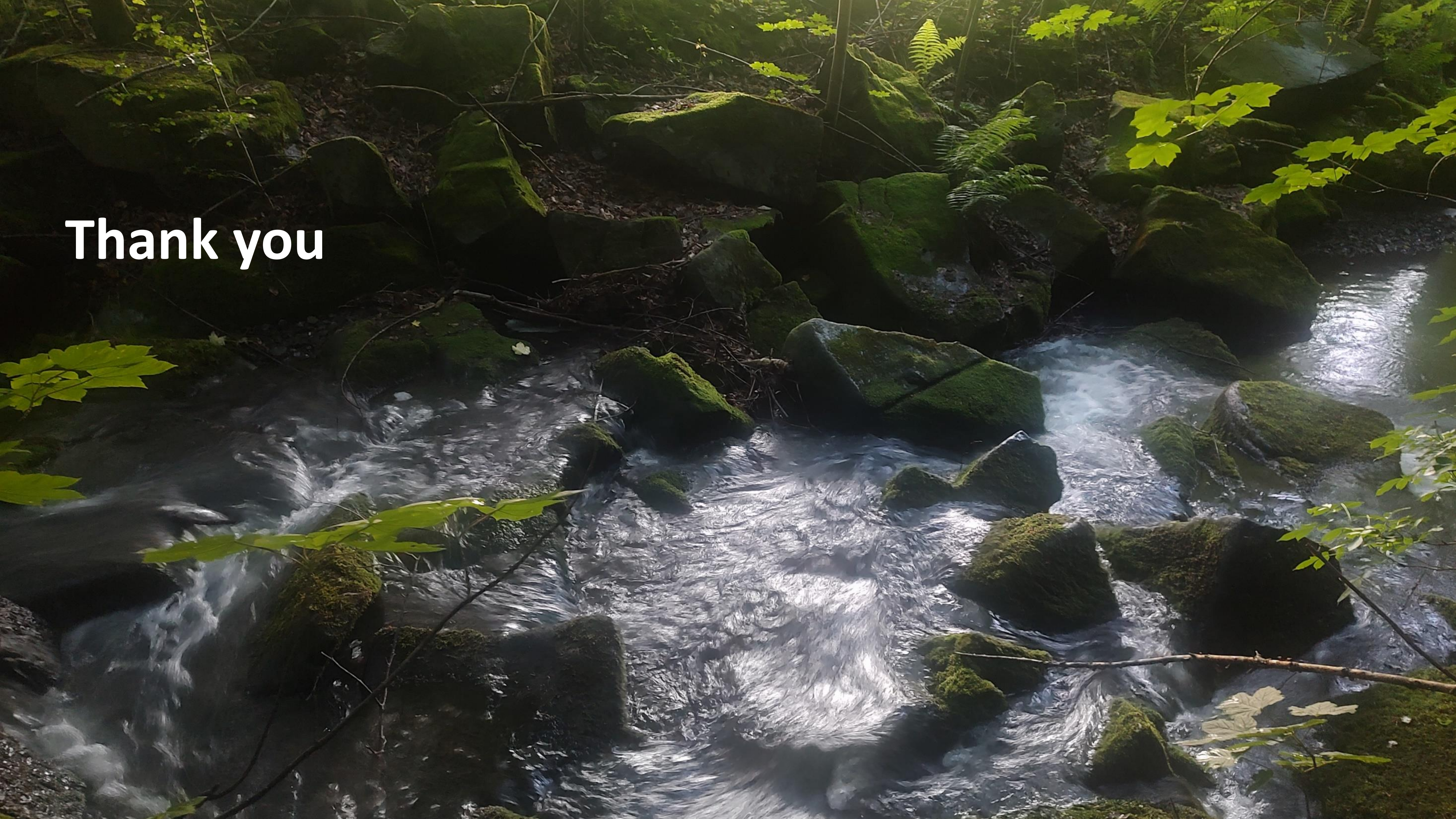
Final Comments and Open Problem

Comment: I was cheating. There is no **w.l.o.g we may assume that *G* is connected**. Components that are paths or cycles need extra care. But they are doable.

Final Comments and Open Problem

Comment: I was cheating. There is no **w.l.o.g we may assume that *G* is connected**. Components that are paths or cycles need extra care. But they are doable.

Open problem: Parameterization by **cluster deletion number** of *G*. (Minimum number of vertices whose deletion yields a disjoint union of cliques. Obviously, $\mathrm{cdn}(G) \leq \mathrm{vcn}(G)$.)


Thank you