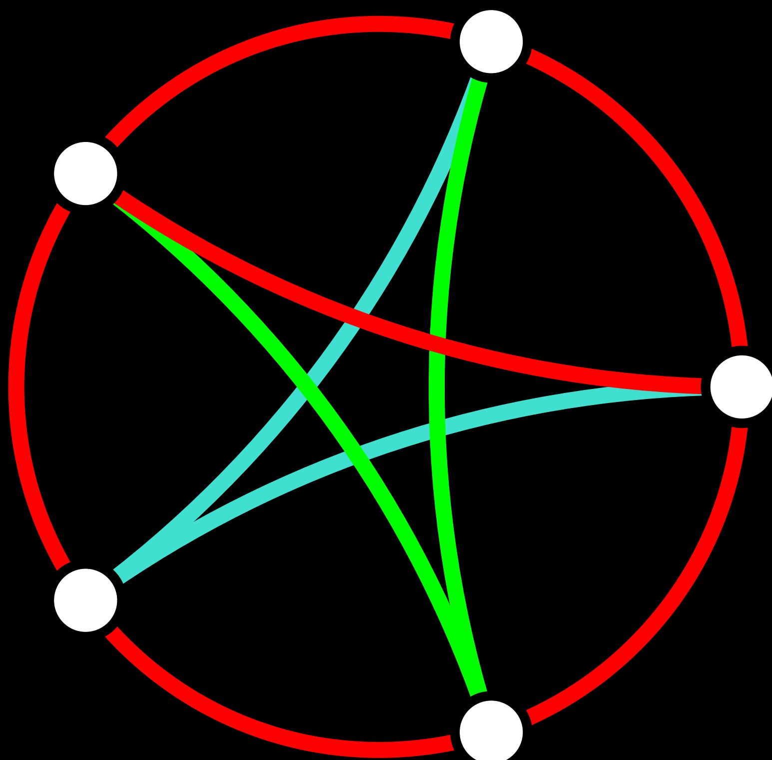
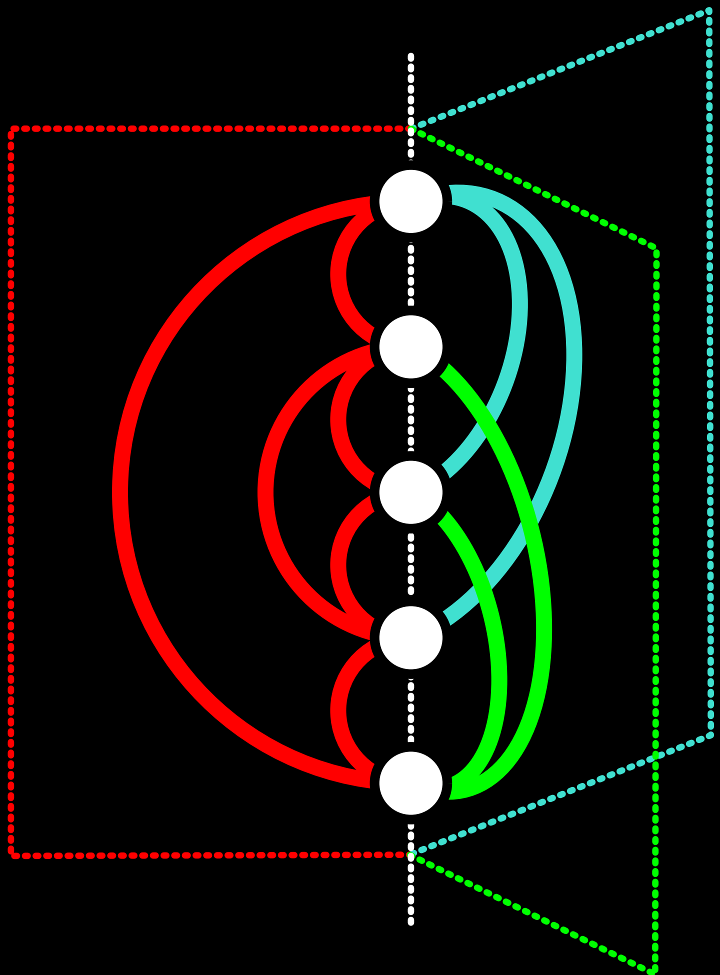
Fixed-Parameter Tractable Algorithms for Crossing Minimization in Book Embeddings

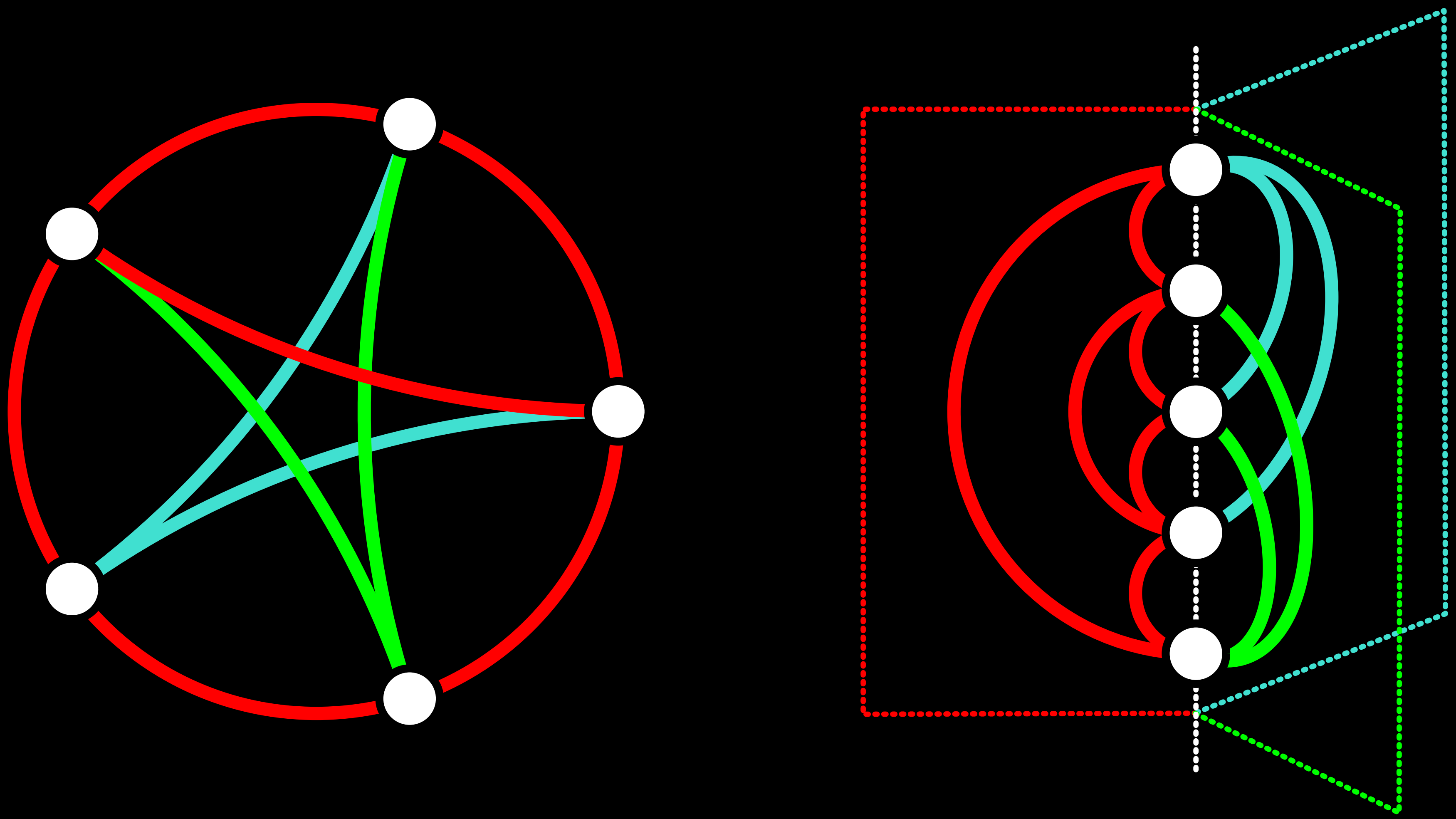




Presented by: Michael J Bannister

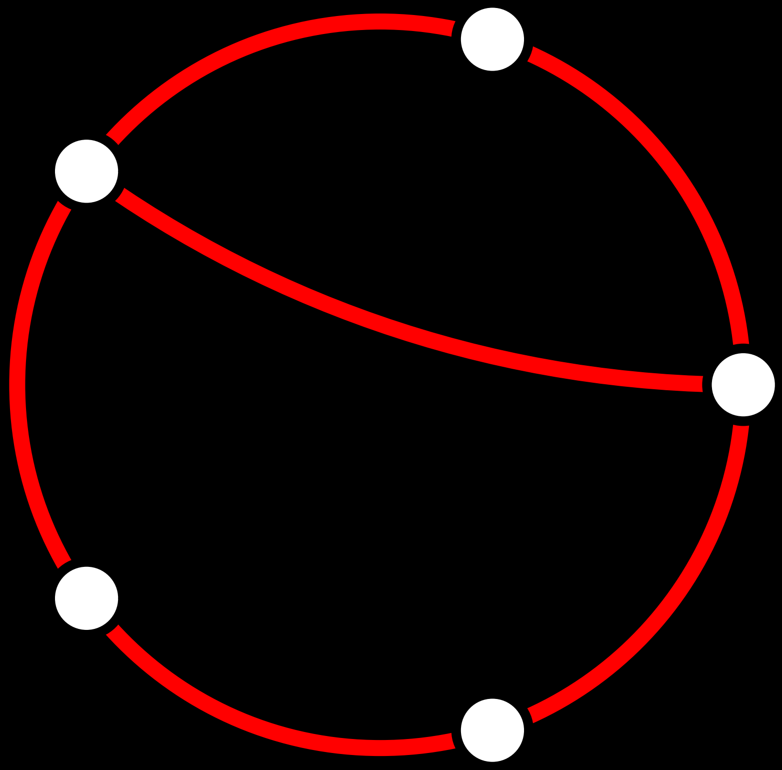
Joint work with: David Eppstein and Joseph A. Simons

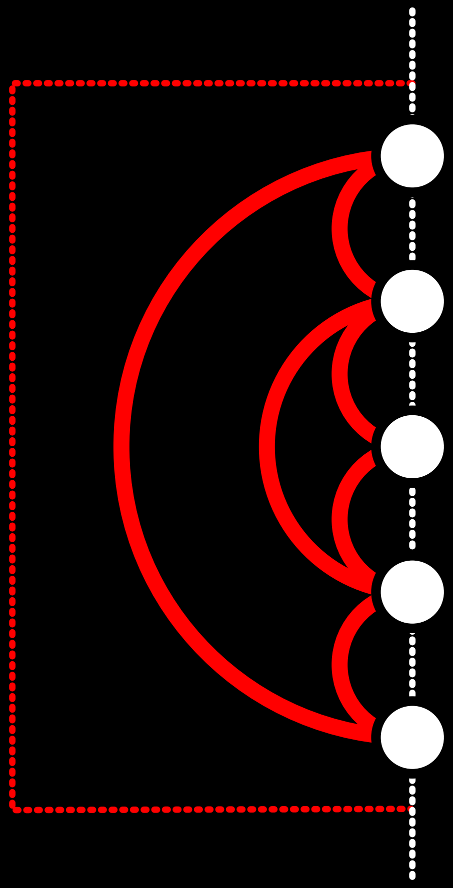
Book Embedding



An embedding of K_5 in 3-pages

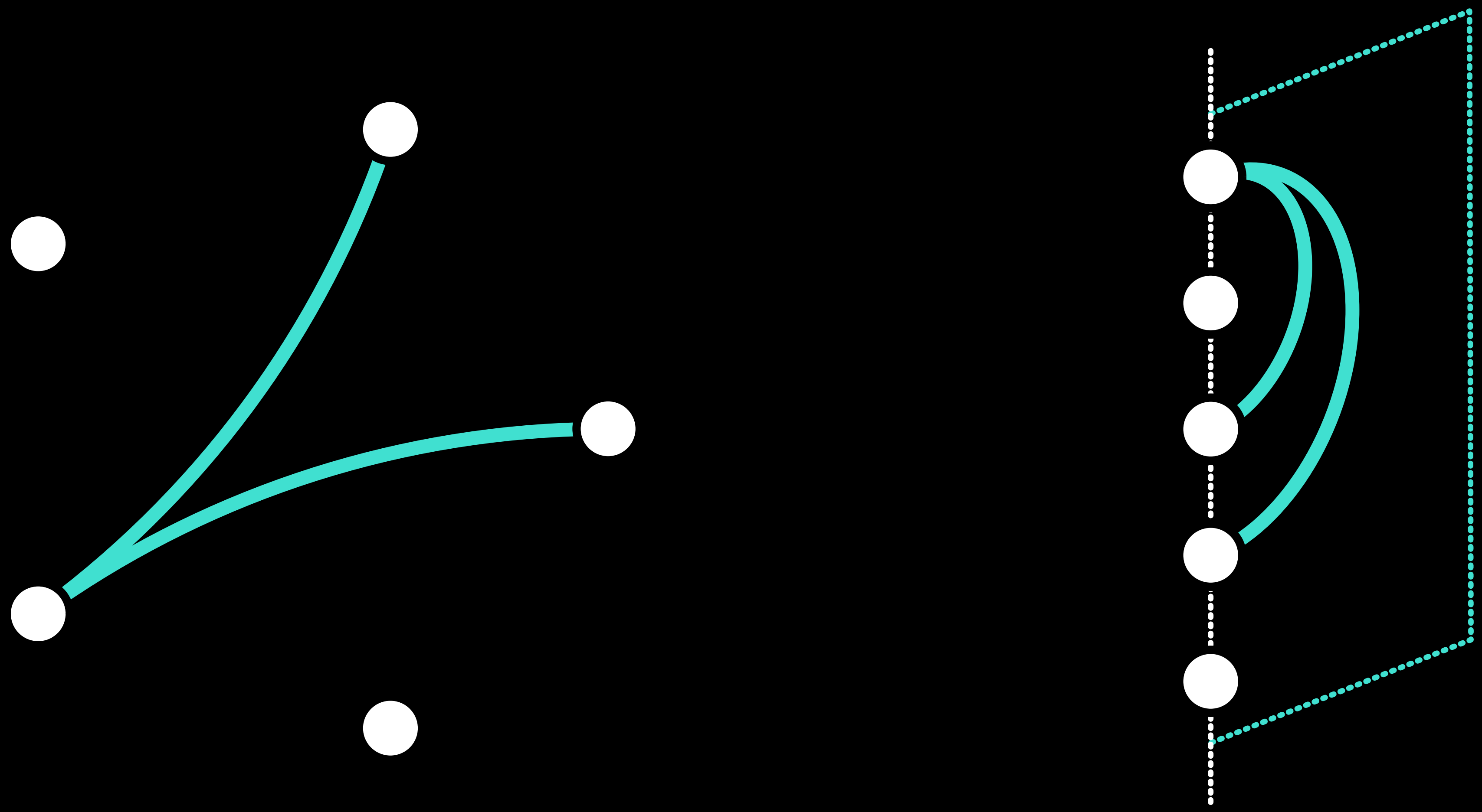
Book Embedding





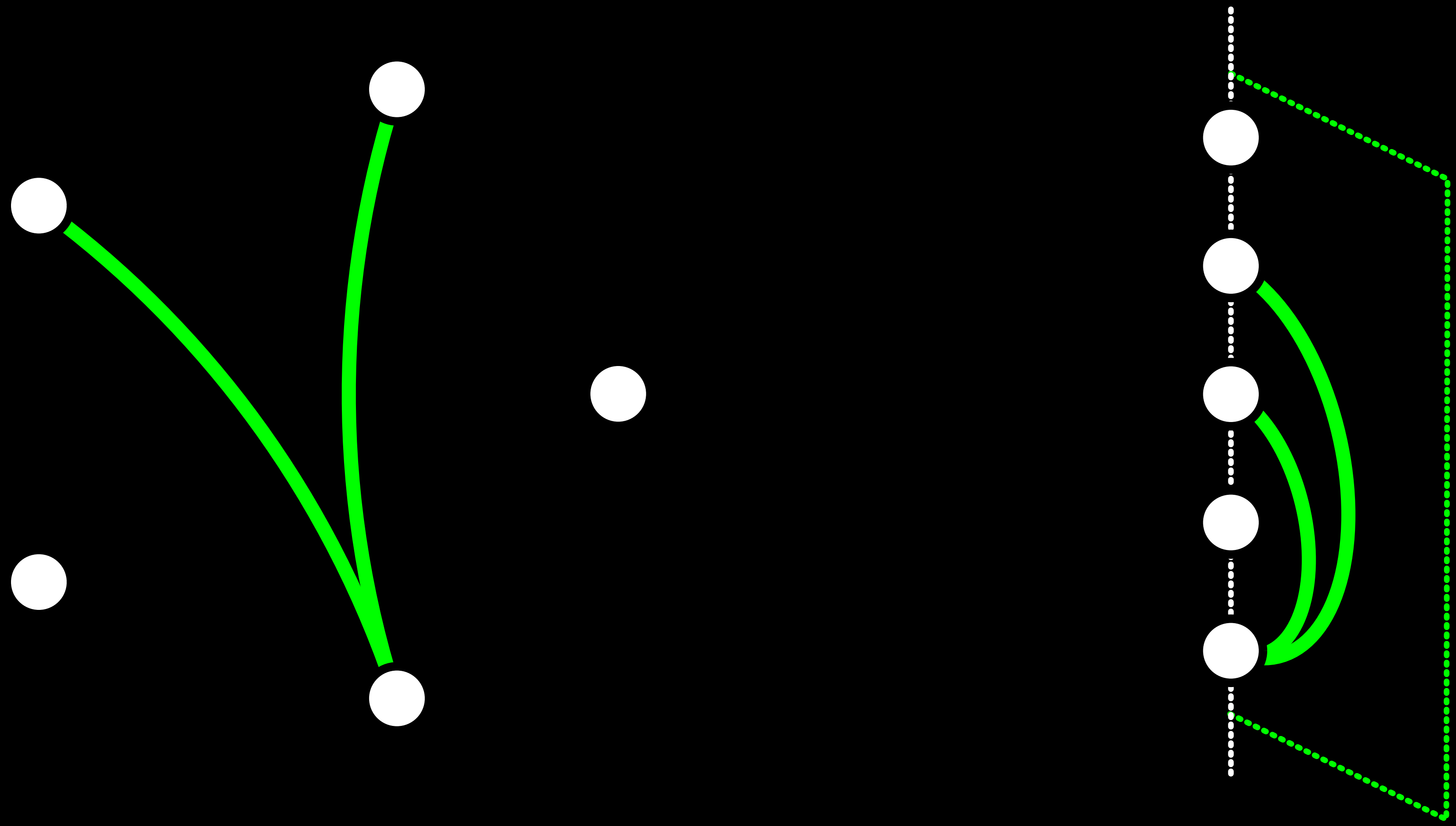
An embedding of K_5 in 3-pages

Book Embedding



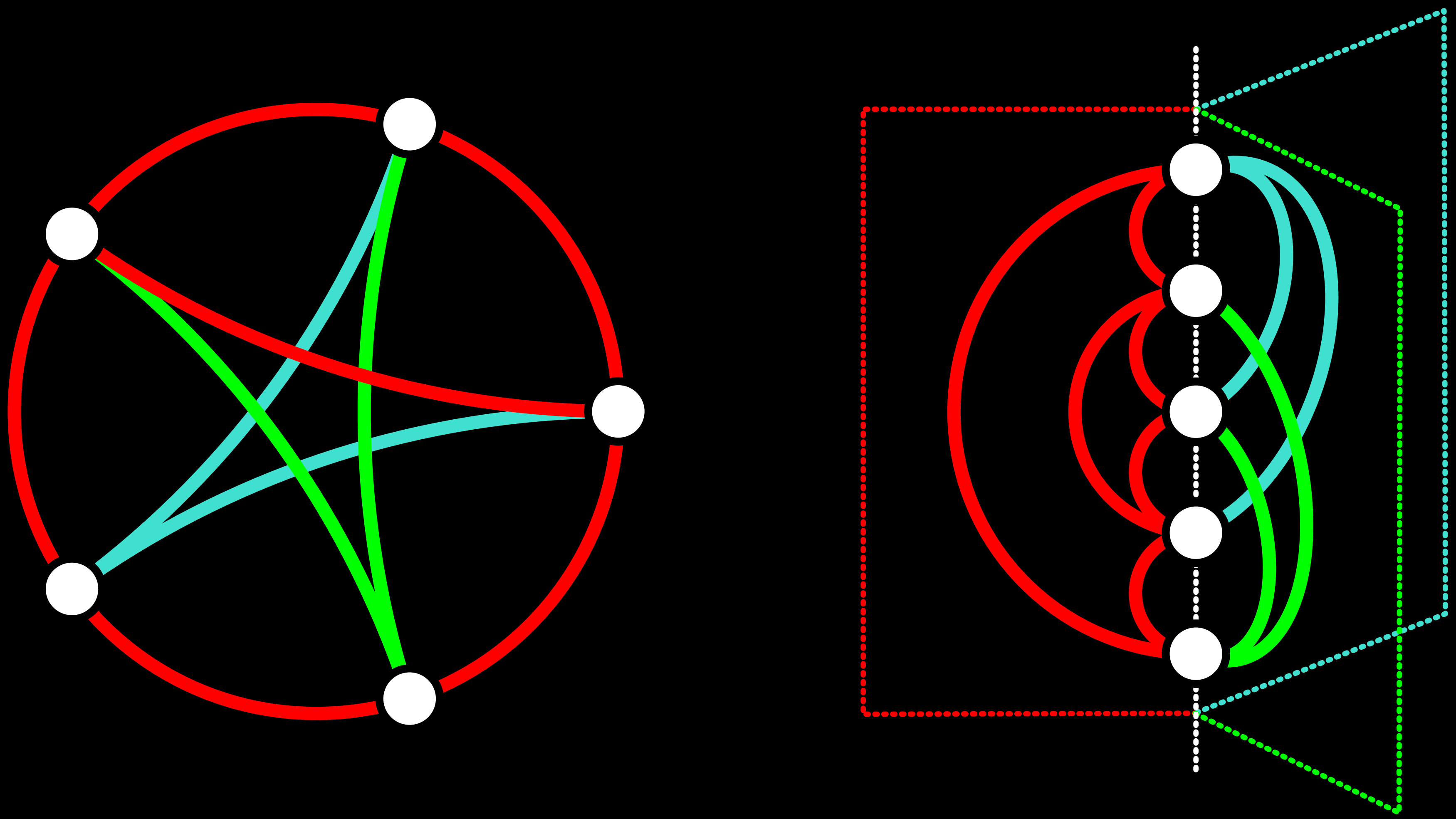
An embedding of K_5 in 3-pages

Book Embedding



An embedding of K_5 in 3-pages

Book Embedding



An embedding of K_5 in 3-pages

Book Embedding

Book crossing number of a drawing D

= sum of the # crossings in each page

k-Page crossing number of a graph G

= min crossing # of a k-page drawing of G

1-Page crossing minimization

Known complexity results:

Planar: $O(n)$

Minimization: NP-hard

Exact minmization: $O(n!) = 2^{O(n \log n)}$

Approximation ratio: $O(\log^2 n)$

Heuristics work well in practice

2-Page crossing minimization

Known complexity results:

Planar: NP-complete

Minimization: NP-hard

Exact minimization: $O(n!2^m) = 2^{O(n \log n + m)}$

Approximation ratio: $O(\log^2 n)$

Heuristics do not work well in practice

Parameterized Complexity

NP-Hard $\Rightarrow$ likely (at least) exponential

Is the problem easier for:
- graphs we few crossings?
- graphs that are "tree like"?
- social networks?

WE FEEL THE OFFICE SCAPEGOAT IS A KEY COMPONENT OF TEAM-BUILDING, AND YOU'RE A GREAT FIT FOR THE JOB.

Fixed-parameter tractable if
- given a parameter p of the input
- $\exists$ a computable function f
- $\exists$ an exact algorithm running in
  $O(f(p)n^c)$ with $c = O(1)$

Parameterized Complexity

NP-Hard $\Rightarrow$ likely (at least) exponential

Is the problem easier for:
    graphs we few crossings?
    graphs that are "tree like"?
    social networks?

WE FEEL THE OFFICE SCAPEGOAT IS A KEY COMPONENT OF TEAM-BUILDING, AND YOU'RE A GREAT FIT FOR THE JOB.

Fixed-parameter tractable if
    given a parameter p of the input
    $\exists$ a computable function f
    $\exists$ an exact algorithm running in
        $O(f(p)n^c)$ with $c = O(1)$

Blame the difficulty of the problem on p not n!

Parameters

k-Almost tree parameter

  Distance from a tree by edge count

  Max of $m - n + 1$ over all biconnected components

Treewidth parameter

  Common FPT parameter

  Measure of large scale tree structure

Natural parameter

  Number of crossings

  Not possible for 2-page

Methods

Kernelization

Step 1: Reduce the size of the input
Step 2: Solve the smaller problem with exact algorithms

Logical expressibility

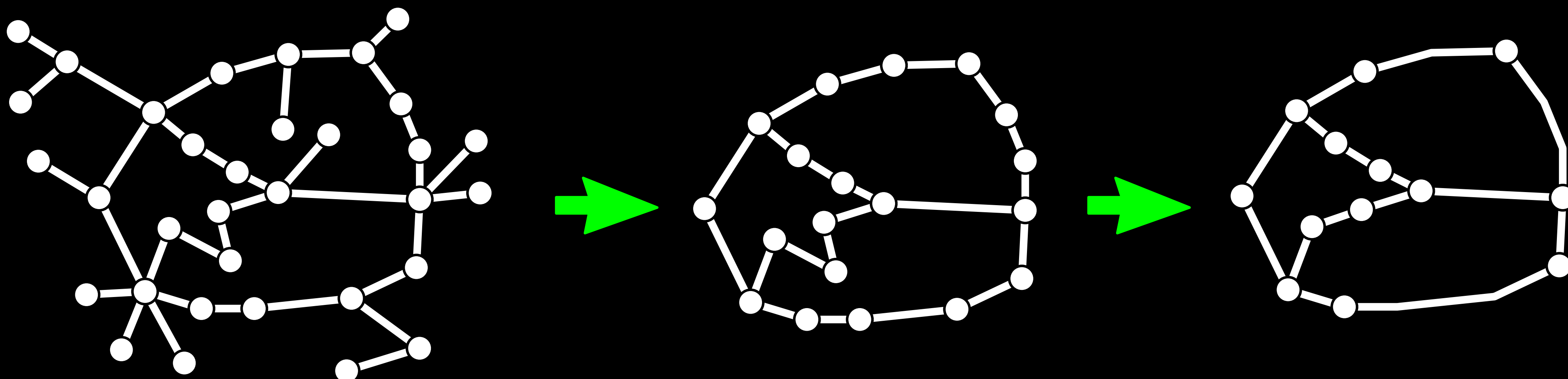
Step 1: Describe the problem in MSO ($\subseteq$ 2nd order logic)
Step 2: Use powerful meta-theorems to produce algorithms

k-Almost tree kernel

Step 1: Iteratively remove degree 1 vertices

Step 2: Reduce paths of degree 2 vertices to $\leq \ell(k)$

To be determined later



Kernelization Results

Crossing minimization:

1-Page $O((5k)!n)$

2-Page $O(2^{6k^3}(6k^3)!n)$

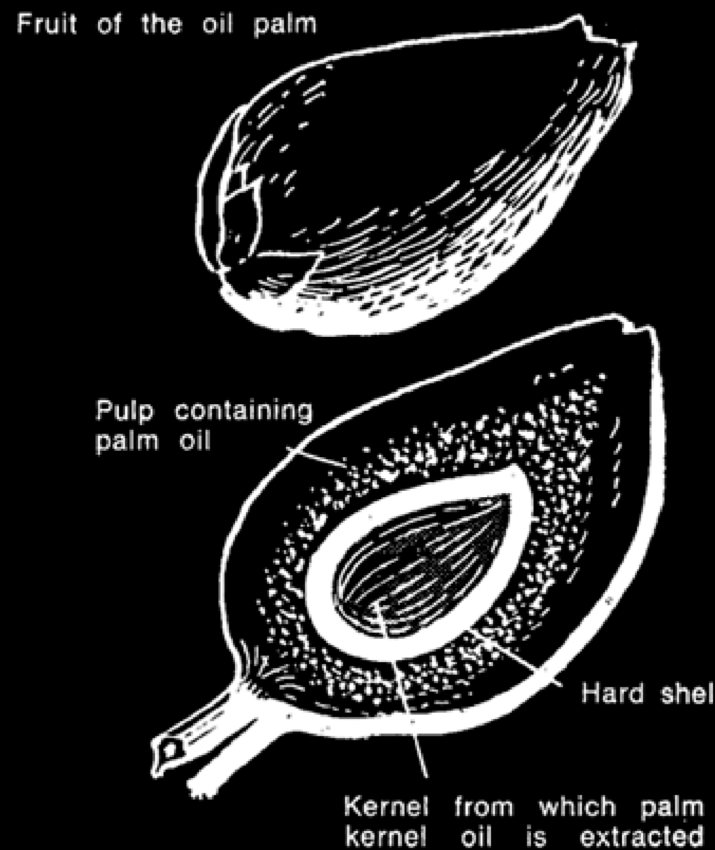
Crossed edge minimization:

1-Page $O((5k)!n)$

2-Page $O(2^{6k^2}(6k^2)!n)$

Additional Result:

1-Planar recognition is FPT



MSO_2 and Courcelle's theorem

Monadic second-order logic:

- Vertex and edge variables: $v_0, v_1, \ldots$, $e_0, e_1, \ldots$
- Vertex and edge set variables: $V_0, V_1, \ldots, E_0, E_1, \ldots$
- Binary relations: $=$, $\in$, I
- Propositional logic operations: $\neg, \wedge, \vee, \rightarrow$
- Quantifiers: $\forall$, $\exists$

Examples of properties expressible in MSO_2

k-coloring	minor containment
connectedness	planarity
hamiltonicity	outerplanarity

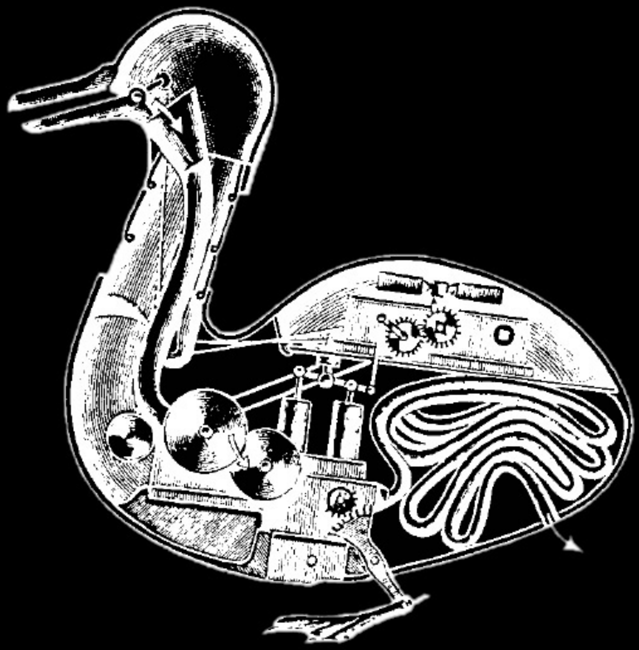
MSO_2 and Courcelle's theorem

Input: A graph G and an MSO_2-formula ϕ

Parameter: treewidth(G) + length(ϕ)

Output: Does G satisfy ϕ

Runtime: $f(k, \ell)n$



Application:

Crossing minimization is FPT in the # crossings

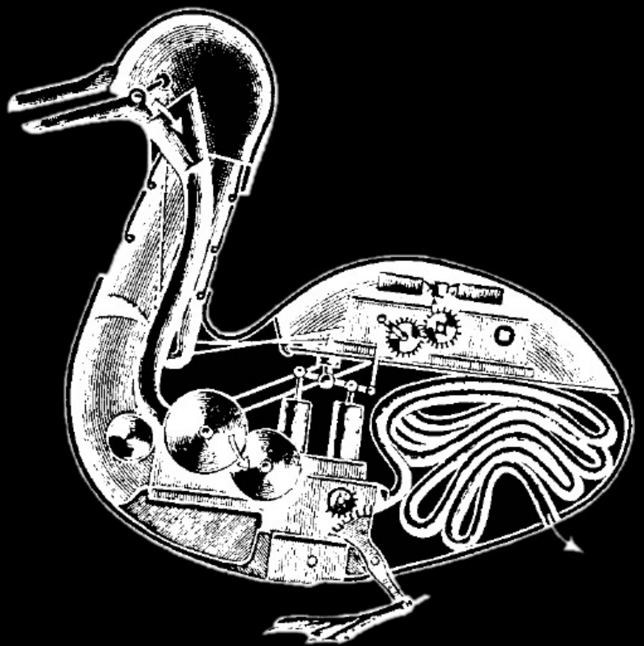
Grohe (2001), Kawarabayashi & Reed (2007)

Logical expressibility results

1-Page crossing minimization is FPT
parameter: crossing number

2-Page planarity is FPT
parameter: treewidth

2-Page crossing minimization is FPT
parameter: treewidth + crossing number



Open Problems

- Extend results to k-pages with $k > 2$
- Faster exact algorithms
- Close the approximation gap
- Practical FPT algorithms

Open Problems

- Extend results to k-pages with $k > 2$
- Faster exact algorithms
- Close the approximation gap
- Practical FPT algorithms

Thank You !