FROM GENOMES TO SUPERGENOMES

HOW TO DEAL WITH BETWEENNESS

FABIAN EXTERNBRINK

SCADS & BIOINFORMATIK, LEIPZIG

FABIAN@BIOINF.UNI-LEIPZIG.DE

www.scads.de

WHAT IS A SUPERGENOME?

BIOINFORMATICS

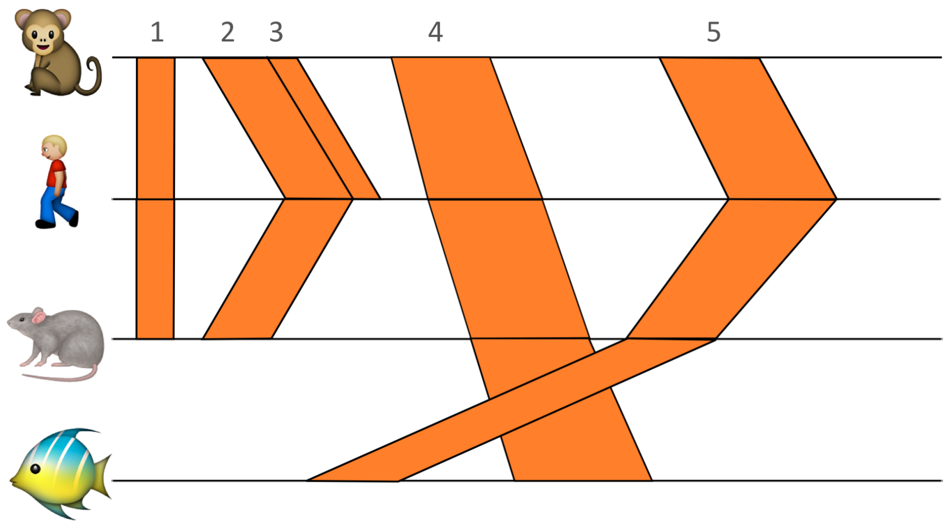
Vol. 28 ISMB 2012, pages i7–i15
doi:10.1093/bioinformatics/bts217

GenomeRing: alignment visualization based on SuperGenome coordinates

A. Herbig[†], G. Jäger[†], F. Battke[†] and K. Nieselt*
Center for Bioinformatics Tübingen, Faculty of Science, University of Tübingen, Sand 14, 72076 Tübingen, Germany

A Supergenome is a common coordinate system for all genomes in a multiple alignment.

SUPERGENOME PROBLEM



- Multiple alignments
- Alignment blocks i.e. local best alignments
- Evolutionary events change the order
- Task:
  - Order the Blocks to create a common coordinate system

WHAT IS BETWEENNESS?

SIAM J. COMPUT.
Vol. 8, No. 1, February 1979

0097-5397/79/0801-0008 $01.00/0

TOTAL ORDERING PROBLEM*

J. OPATRNY†

Abstract. The problem of finding a total ordering of a finite set satisfying a given set of in-between restrictions is considered. It is shown that the problem is *NP*-complete.

Key words. algorithms, computational complexity, total ordering, *NP*-completeness

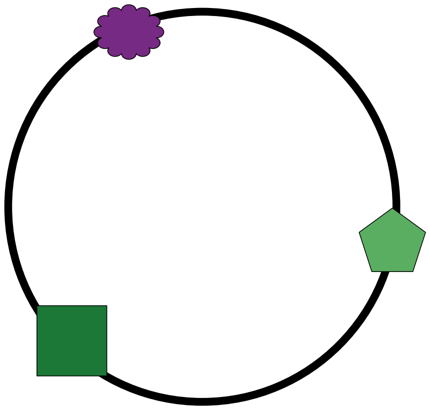
Deciding Problem

Given a finite set X and a collection $C \subseteq X^3$, is there a total order on X such that $(i, j, k) \in C$ either $i < j < k$ or $i > j > k$?

Optimization Problem

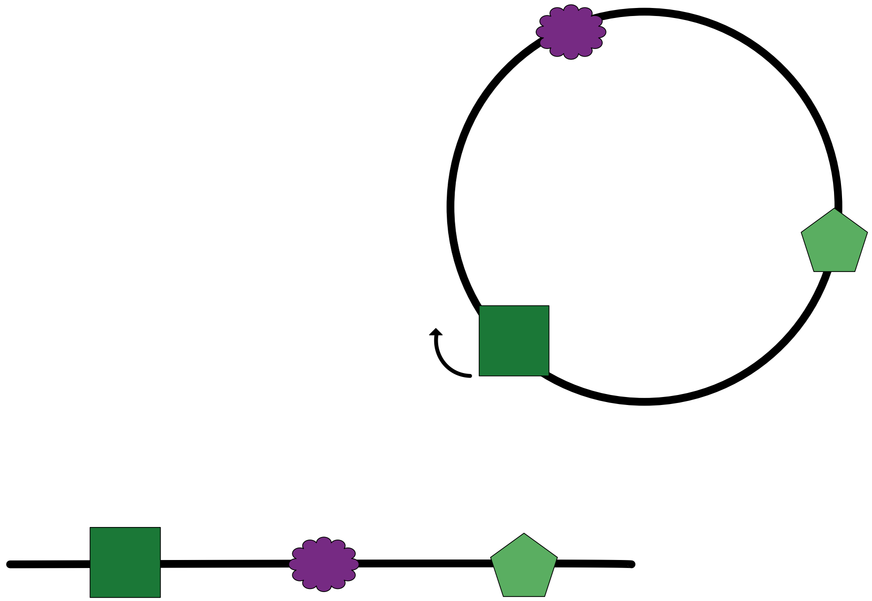
Given a finite set X and a collection $C \subseteq X^3$, find a maximal subset S from C, for which the decision problem w.r.t. S is true.

TOTALLY UNRELATED
RNA EXAMPLE OF BETWEENNESS

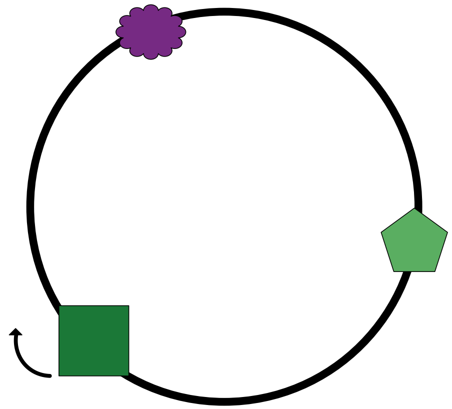


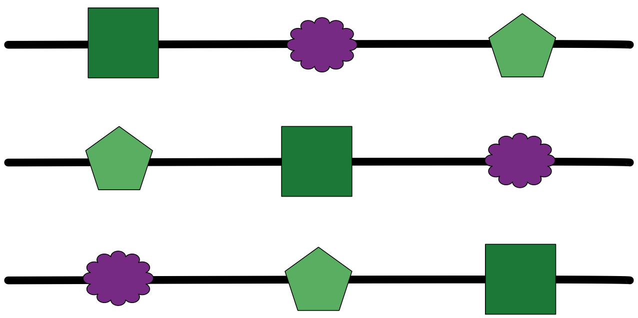
- Given a circular RNA molecule
- Different marker may exist on the RNA molecule
- Question:
  - What was the linear transcript?
    - Linear order of the marker?

TOTALLY UNRELATED
RNA EXAMPLE OF BETWEENNESS

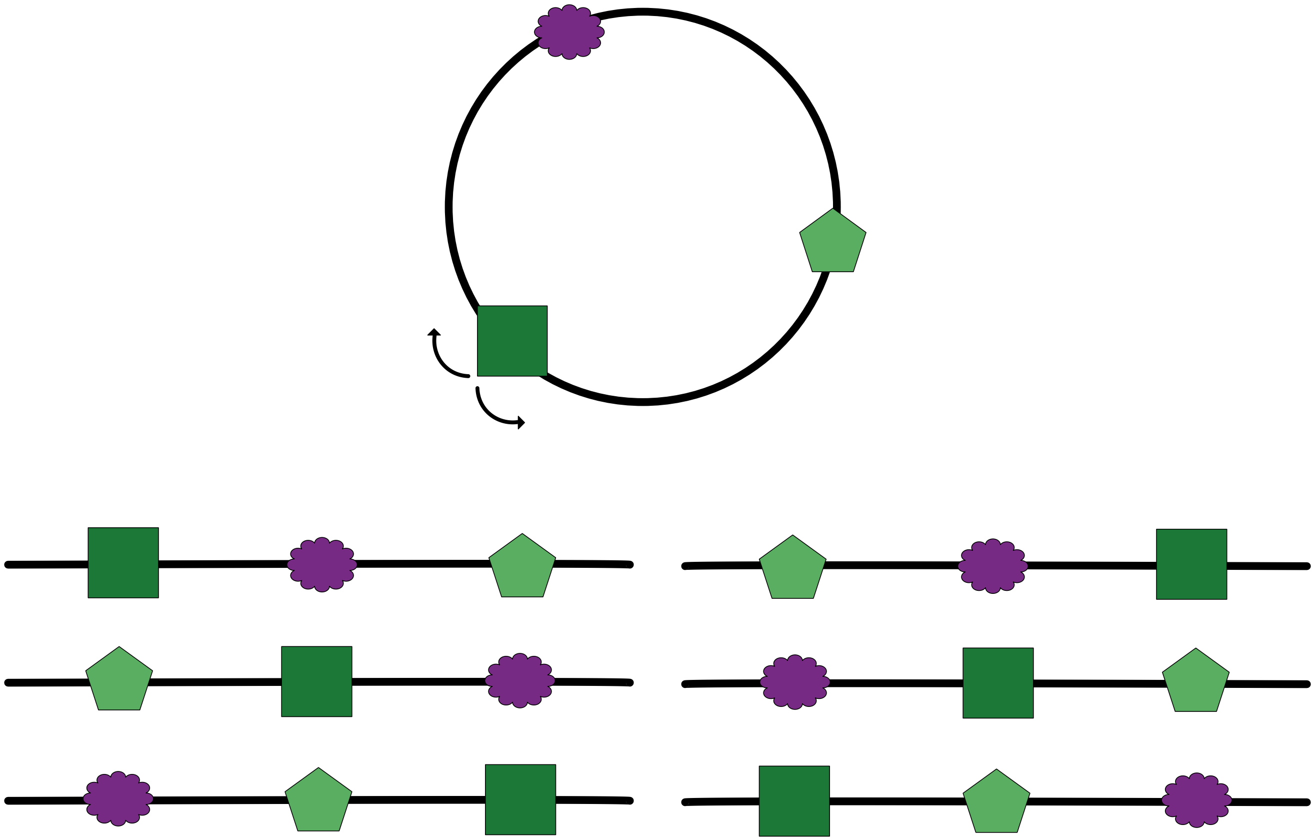


TOTALLY UNRELATED RNA EXAMPLE OF BETWEENNESS

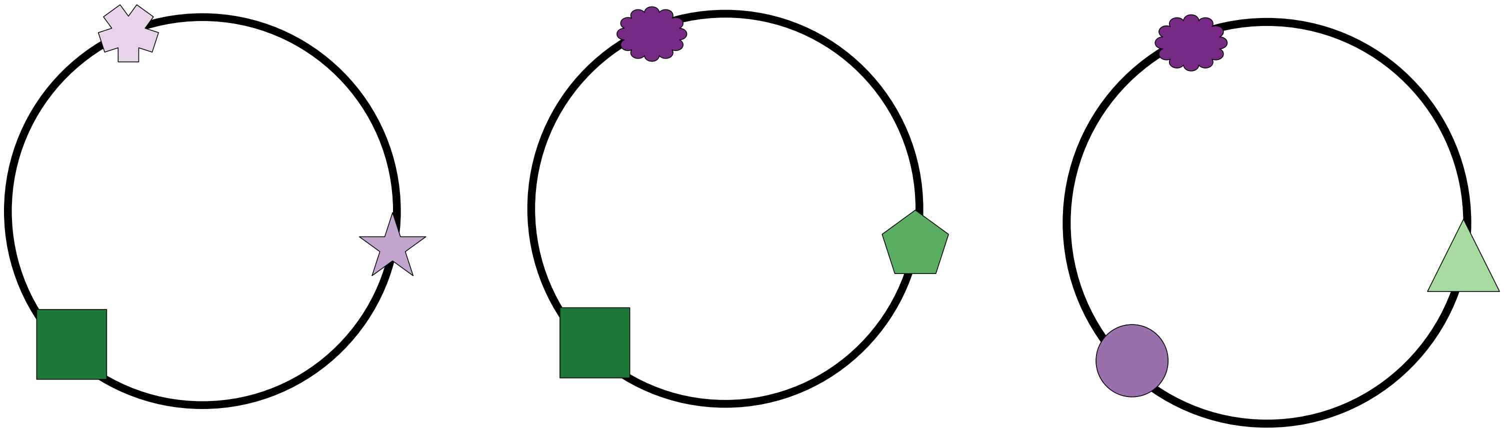




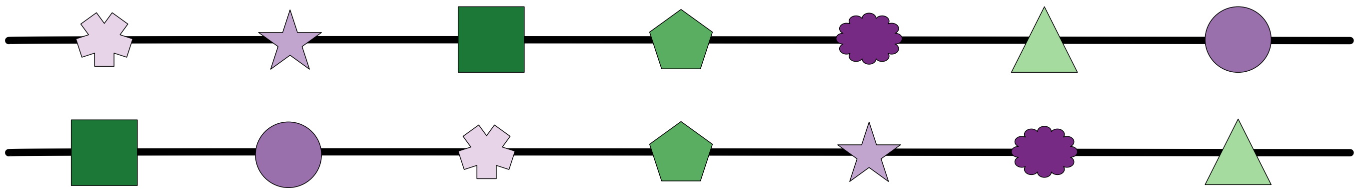
TOTALLY UNRELATED
RNA EXAMPLE OF BETWEENNESS



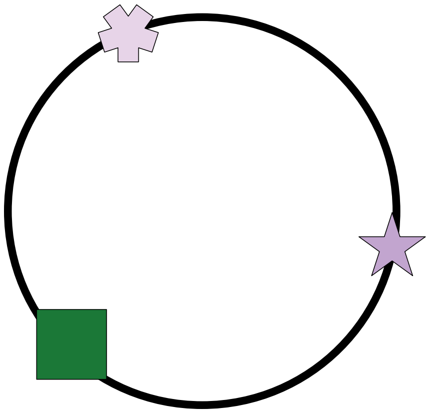
TOTALLY UNRELATED
RNA EXAMPLE OF BETWEENNESS

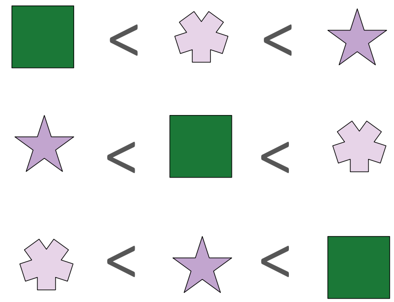


- More then one RNA molecule from one transcript
- Reading directions of the RNA molecules are independent
- Splicing allows deletions of markers



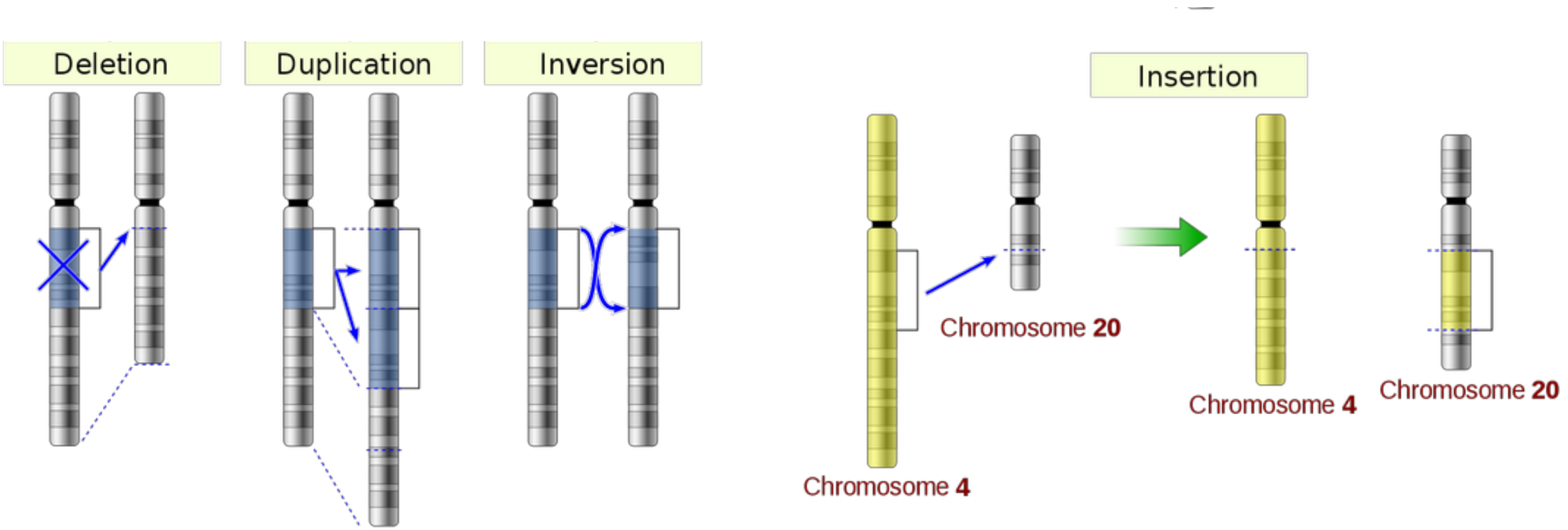
TOTALLY UNRELATED
RNA EXAMPLE OF BETWEENNESS





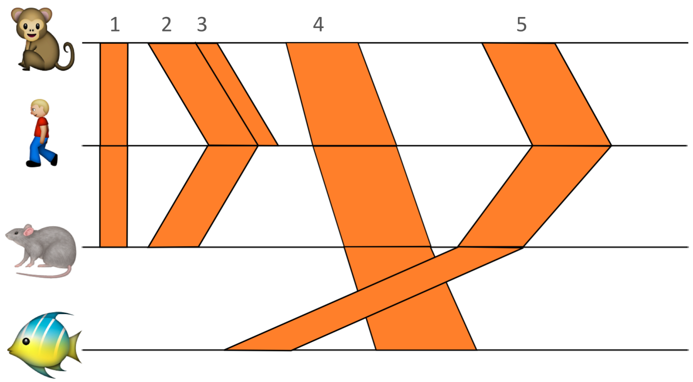
- It is a Betweenness Optimization Problem
  - The direction is not clear
  - Because of circularity, not all triples can be fulfilled
- Find largest subset of Triples that can be fulfilled by a linear order
- The linear order is the most likely linear transcript

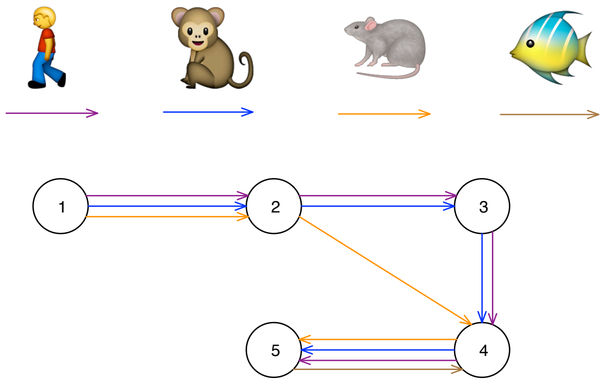
WHERE IS THE CONNECTION?



- Supergenome Problem is a Betweenness Optimization Problem
- Chromosome mutations
  - Direction of blocks is not clear
  - Not all triples can be fulfilled

SUPERGENOME GRAPH

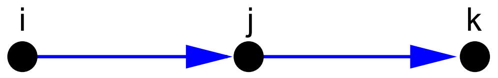




- Create graph from alignment
- One block is one vertex
- A edge from block **v** to **w** in color <u>x</u> is added if block **w** is successor of **v** in a genome <u>x</u>

BETWEENNESS AND GRAPH?

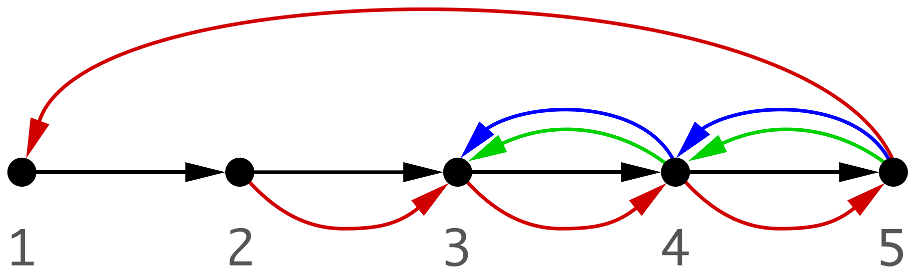
- Extends betweenness problem to a graph.
- Colored Multigraph Betweenness Problem
  - Find a maximal subset of colored edges E’ of the multigraph such that the set of triples C(E’) has a total order, where (i,j,k) $\in$ C(E’) if and only if there are two edges{i,j} and {j, k} with the same color.

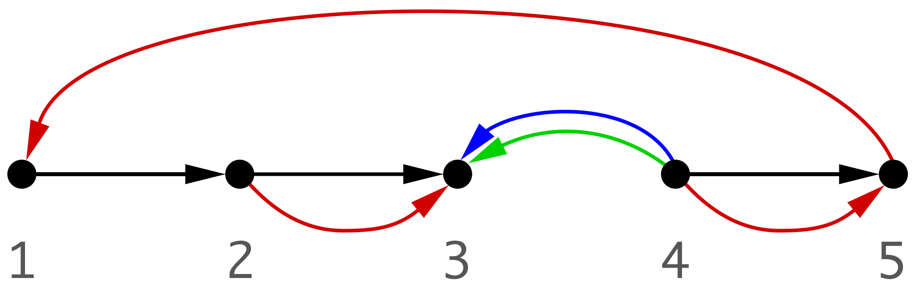


- Idea to solve this is:
  - Create a order of the vertices of the graph.
  - Then calculate which edges are in the subset.

FAS AND TOPOLOGICAL SORTING

- Topological Sorting
  - Create order out of a DAG
  - If there is an edge from **v** to **w**, than **v** is before **w** in the order
- Feedback Arc Sets (FAS)
  - Create a DAG
  - Remove as less edges as possible

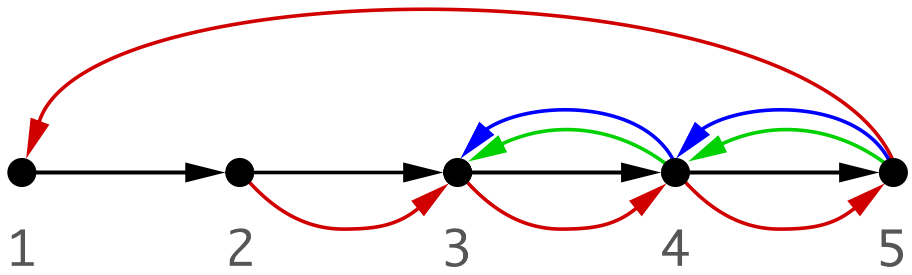




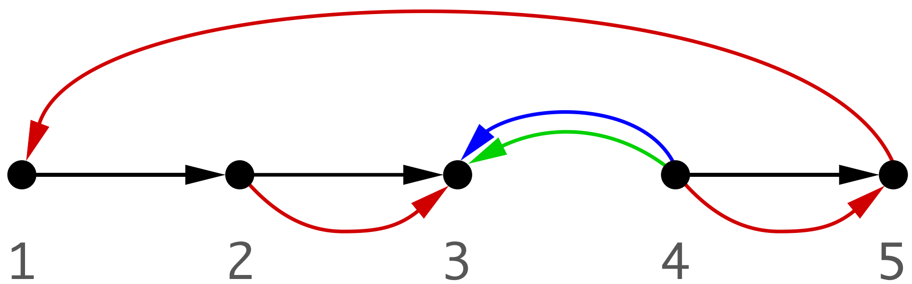
`Order:4,5,1,2,3`

FAS AND TOPOLOGICAL SORTING

- The resulting order destroys many betweenness information
- FAS does not fit betweenness problem well
  - Create artificial sinks and sources

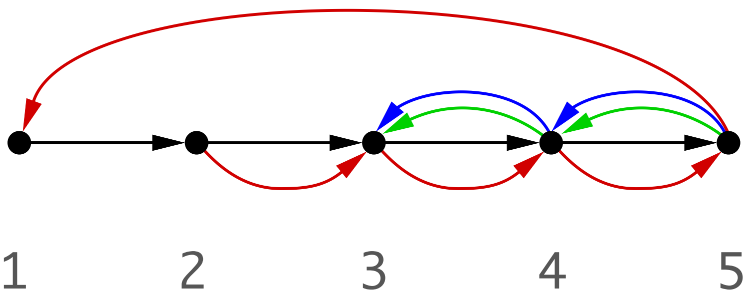


(1,2,3) (4,5,1)



Order:4,5,1,2,3

BETWEENNESS PROBLEMS



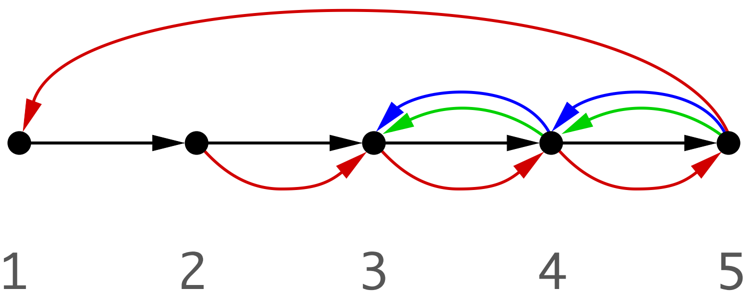
(1,2,3) (2,3,4) (3,4,5)

(2,3,4) (3,4,5) (4,5,1)

(5,4,3)

(5,4,3)

BETWEENNESS PROBLEMS



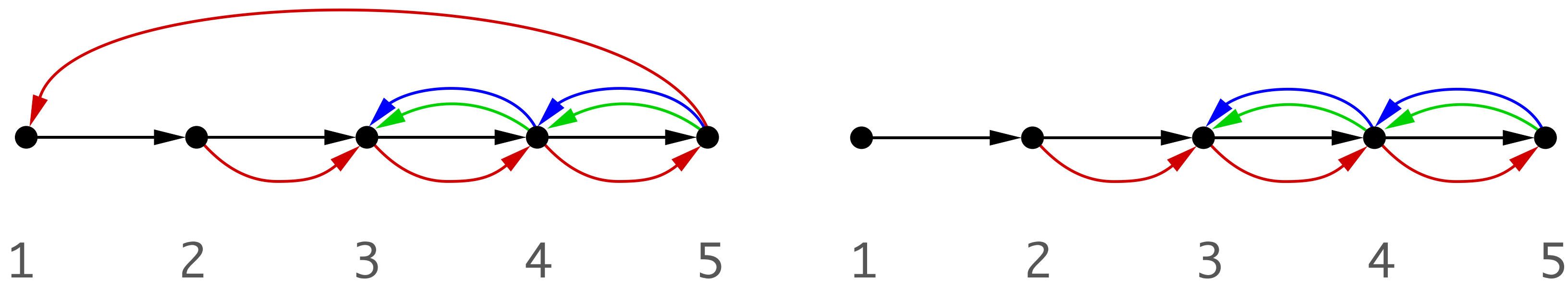
(1,2,3) (2,3,4) (3,4,5)

(2,3,4) (3,4,5) (4,5,1)

(5,4,3)

(5,4,3)

OPTIMAL BETWEENNESS SOLUTION



(1,2,3) (2,3,4) (3,4,5)

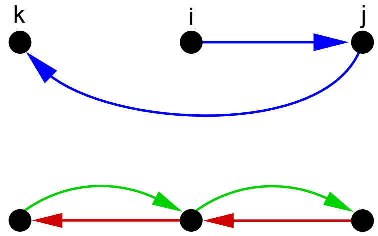
(2,3,4) (3,4,5)

(5,4,3)

(5,4,3)

WHY IS FAS BAD?

- Betweenness allow some cycles.
- Remove of all cycles is too much
- Two types of cycles:
  - Inconsistent cycles
  - Undirected cycles
- If only inconsistent cycles in the graph
  - Solution to FAS and Betweenness Problem is the same

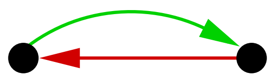


PREPROCCESING GRAPH

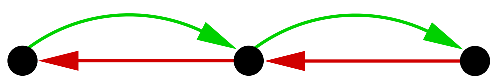
- Remove (most of) the undirected cycles
  - Mini-cycle Remover
- FAS is NP-complete
  - Use heuristic
- Noise reduction by simplifying collinear parts of the graph.
  - Sink/source simplifier
  - Closed-DAG simplifier

MINI-CYCLE REMOVER

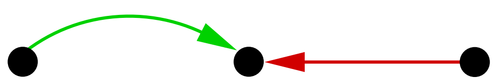
- Type of the cycle depends on used total order
- No order is given at this point
  - Use heuristic to find undirected cycles
- A mini-cycle with only two vertices



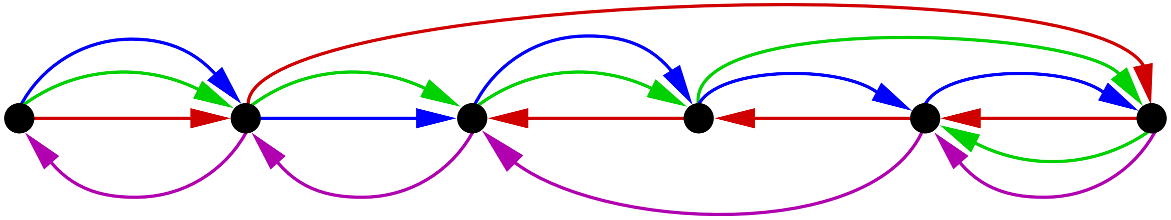
  - Very likely a undirected cycles
- Remove all mini-cycles in a intelligent way
  - If two mini-cycles share a vertex remove them together



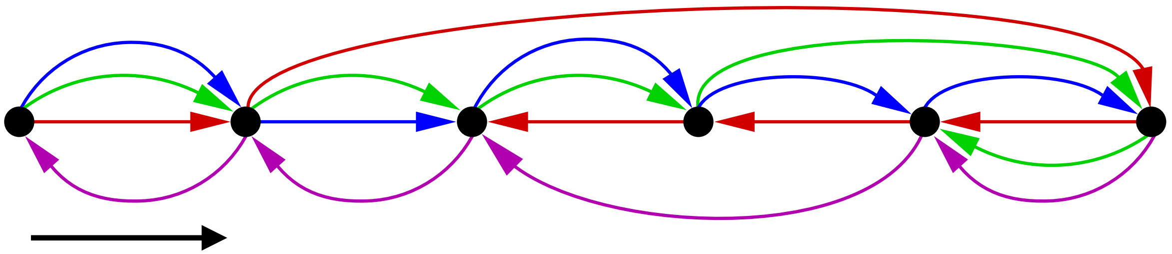
  - Avoids generation of artificial sinks and sources



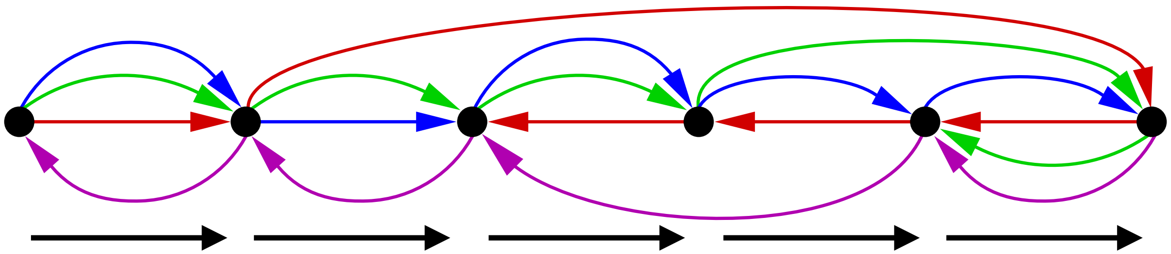
MINI-CYCLE REMOVER EXAMPLE



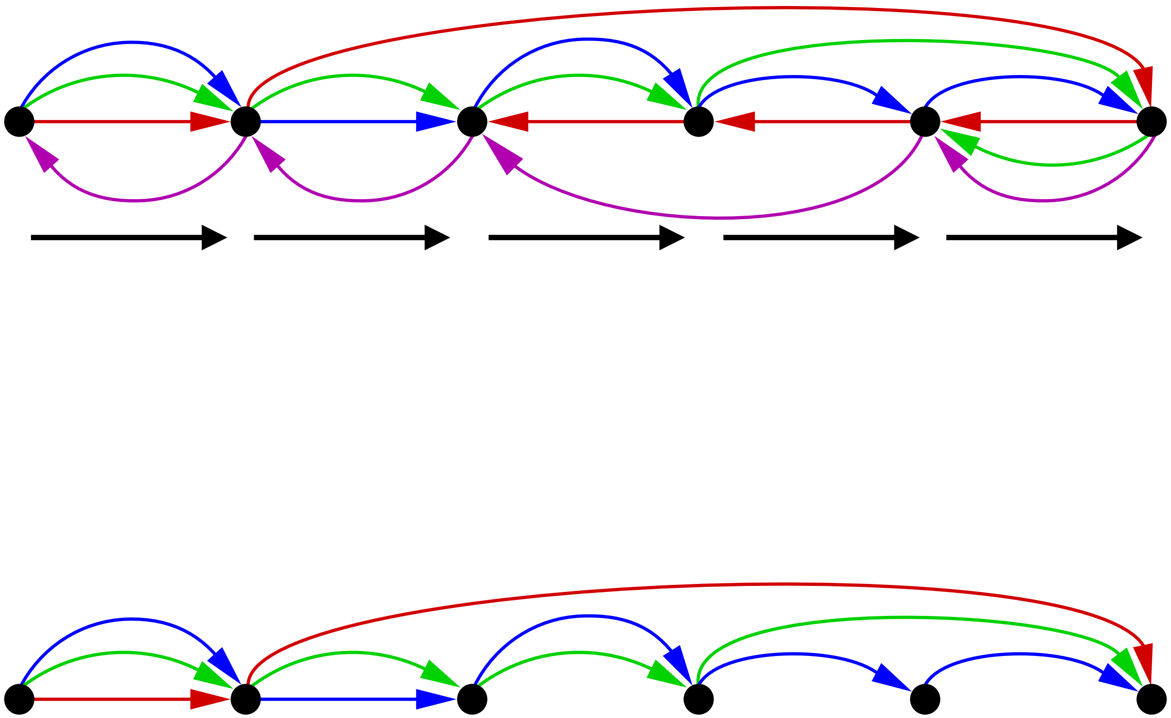
MINI-CYCLE REMOVER EXAMPLE



MINI-CYCLE REMOVER EXAMPLE

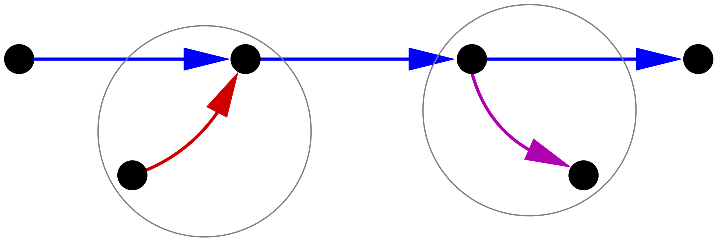


MINI-CYCLE REMOVER EXAMPLE

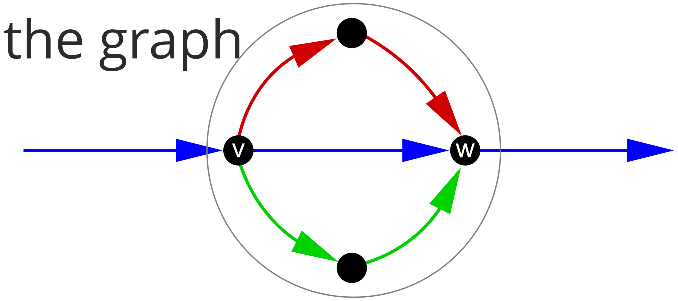


SINK/SOURCE SIMPLIFIER

- A sink/source with only one predecessor/successor
- The position in the order is only influenced by the predecessor/successor
- It can be placed directly behind/before this predecessor/successor
- This is a collinear part in the graph
  - Simplified to one vertex

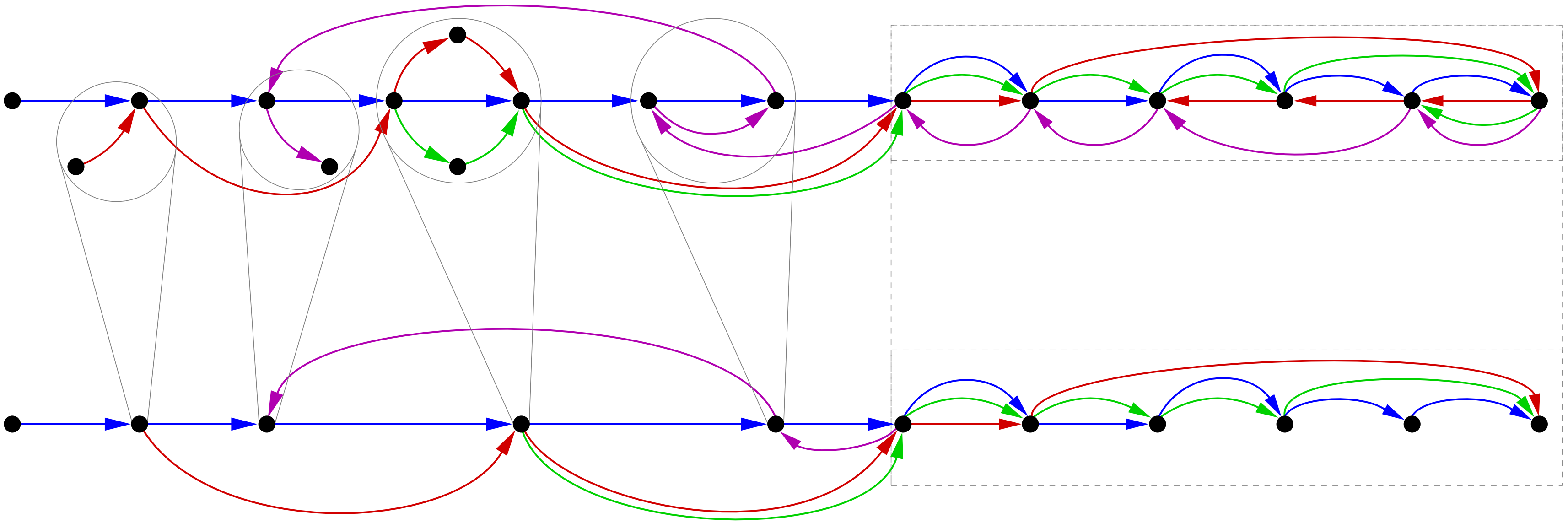


CLOSED-DAG SIMPLIFIER



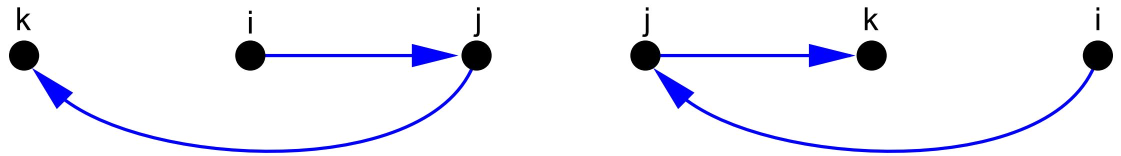
- A Closed-DAG is a collinear part in the graph
- It has this features:
  - It is a directed acyclic graph
  - It is connected to the rest of the graph by a single source vertex **v** and a single sink vertex **w**
  - All direct successors of **v** and all direct predecessors of **w** are contained in it
  - All vertices in it are successors of **v** and predecessors of **w**
- The Closed-DAG is an atomic unit in the order.

THE BIG PICTURE



MEASURE RESULTING ORDER

- Directed acyclic graph
  - Remove all edges that go from a vertex on position i to a vertex on position j if j<i
- Betweenness graph
  - Add invers Edges
  - Readd all edges that do not create a bad triple (i,j,k)



- Number of edges and triples can be counted
- No gold standard!
  - Can be compared with the start graph

DATASETS

- Two UCSC Datasets.
  - Created with a Reference Species
- Yeast
  - 7 species
  - 43495 vertices
  - 203275 edges, 197043 triples
- Insects
  - 27 species
  - 1451433 vertices
  - 25549792 edges, 25540919 triples

RESULT

Edges % (triples %)	*Yeast DAG*	*Insect DAG*	*Yeast Betweenness*	*Insect Betweenness*
Simple FAS				
No mini-cycle Remover				
No simplifier				
All				

RESULT

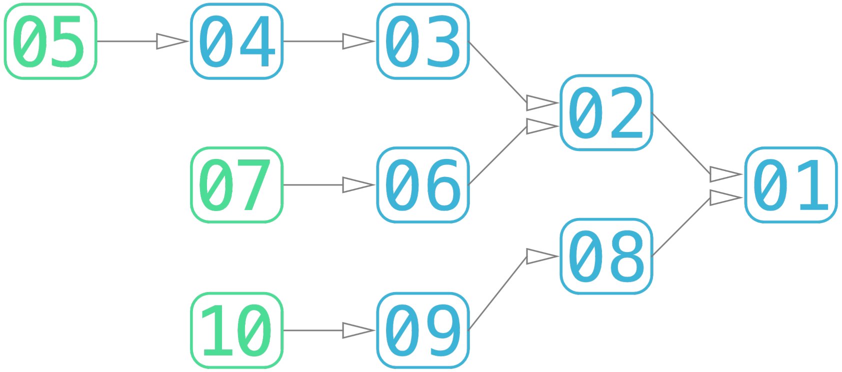
Edges % (triples %)	*Yeast DAG*	*Insect DAG*	*Yeast Betweenness*	*Insect Betweenness*
Simple FAS	*66.87 (53.86)*	*61.54 (52.49)*	*82.65 (66.56)*	*86.97 (75.24)*
No mini-cycle Remover	*66.86 (53.87)*	*61.55 (52.50)*	*82.67 (66.60)*	*86.98 (75.26)*
No simplifier	*60.99 (59.29)*	*56.75 (56.37)*	*96.96 (94.04)*	*99.39 (98.80)*
All	*61.04 (59.31)*	*56.75 (56.37)*	***96.91 (93.94)***	***99.39 (98.80)***

SUMMARY

- Betweenness is everywhere!
- Solve optimization problem
- New graph based solution
  - Maximal subset of Edges
- Well studied approaches does not fit well
  - Can be fixed by a preprocessing
- Results can be measured
- Results look very promising

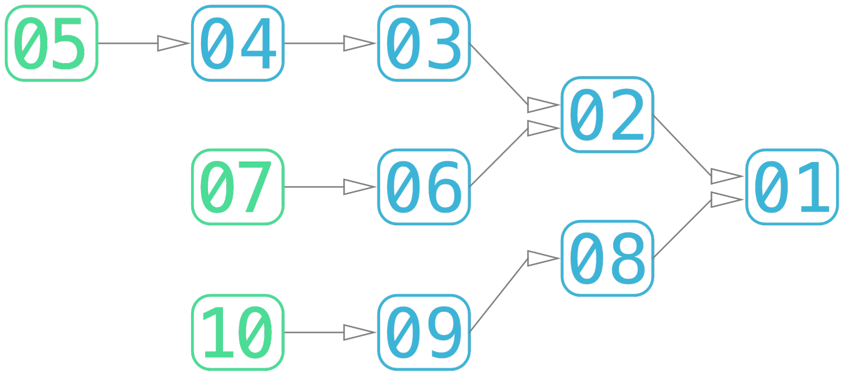
THANK YOU FOR YOUR ATTENTION

TOPOLOGICAL SORTING



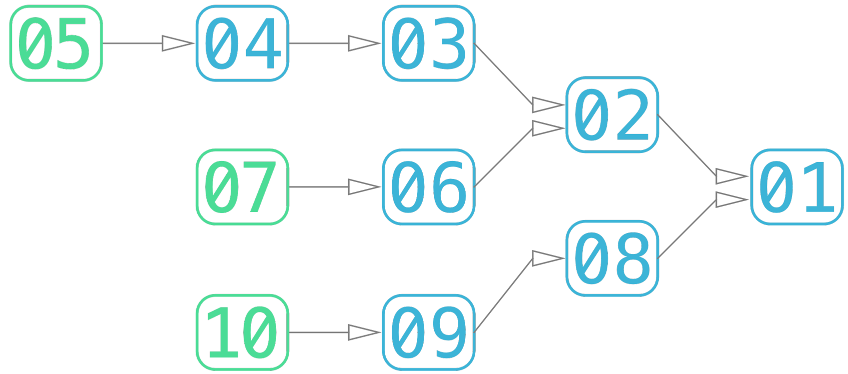
- Topological sorting is not unambiguous
- Valid orders e.g.:
  - 5,4,3,7,6,2,10,9,8,1
  - 5,7,10,4,9,6,3,8,2,1
  - 10,7,5,6,4,3,2,9,8,1
  - 7.10,5,4,9,3,6,8,2,1

DISTANCE TOPOLOGICAL SORTING



- Use Distance information
- Next vertex in order is chosen by distance
- Not optimal for betweenness
- Valid orders e.g.:
  - 5,4,3,7,6,2,10,9,8,1
  - 7,6,5,4,3,2,10,9,8,1
  - 10,9,8,76,5,4,3,2,1

BETWEENNESS SORTING



- Optimize minimal number of violation of the Robinson rule (1951):

$$\max(d(i,j), d(j,k)) \leq d(i,k)$$

$$\underbrace{\underbrace{i < j}\underbrace{< k}}_{\leq}$$

- Change the order to an other valide topological sorting
- Check if number of violation is lowered
- Reaped until no further optimization is found
- **7,6**,5,4,3,2,10,9,8,1 ⟶ 5,4,3,**7,6**,2,10,9,8,1

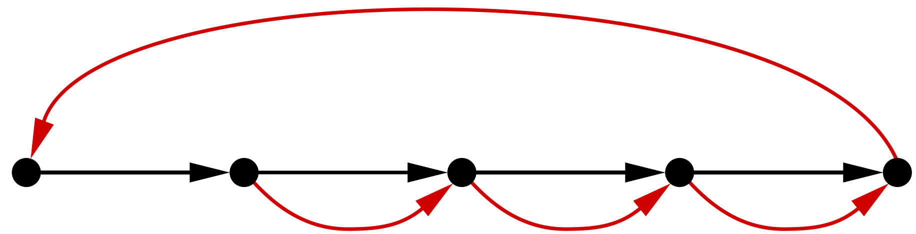
GUTTENPLAG

- 1218 plagiarism fragments
- 135 sources
- 63% of the work
- Sources widely distributed
- Possible questions:
  - Is basic structure from a source?
  - Which source is dominant in which part?

GUTTENPLAG

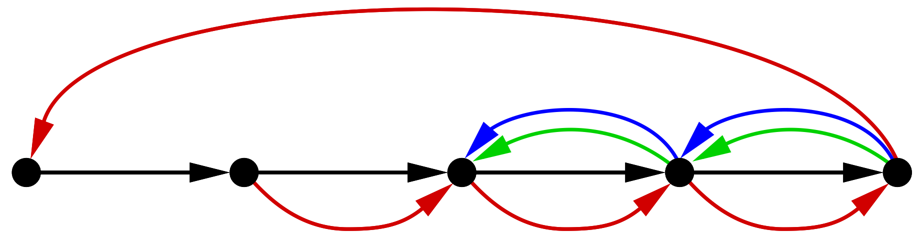
- 1218 plagiarism fragments
- 135 sources
- 63% of the work
- Sources widely distributed
- Possible questions:
  - Is basic structure from a source?
  - Which source is dominant in which part?

GUTTENPLAG GRAPH



- Cites, pages, or sections as vertices
- Edges in the order of the dissertation and in order of the cites.

GUTTENPLAG GRAPH



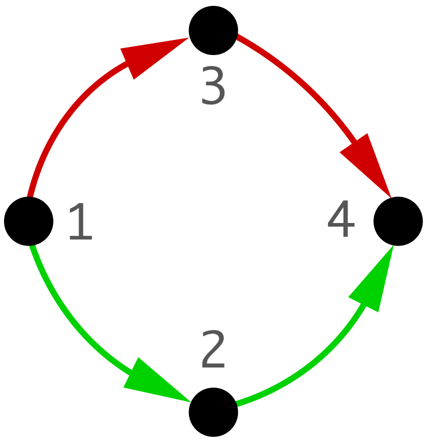
- Cites, pages, or sections as vertices
- Edges in the order of the dissertation and in order of the cites.

HAMILTONIAN PATH

- A Hamiltonian path is a path that visits each vertex exactly once.
- The graph is connected
- Ignoring the direction of edges
  - Betweenness has no direction

1,2,4,3

1,3,4,2

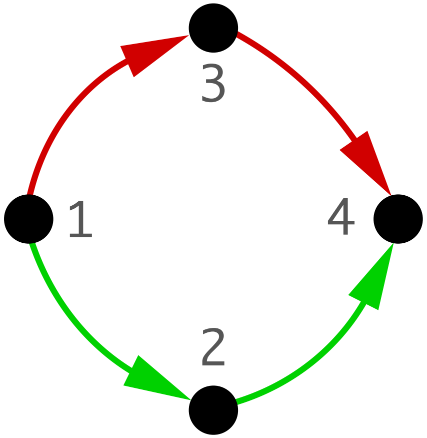


HAMILTONIAN PATH

- Violated betweenness when two parts parallel
- Does not fit betweenness problem well

1,2,4,3

1,3,4,2



Betweenness solution:

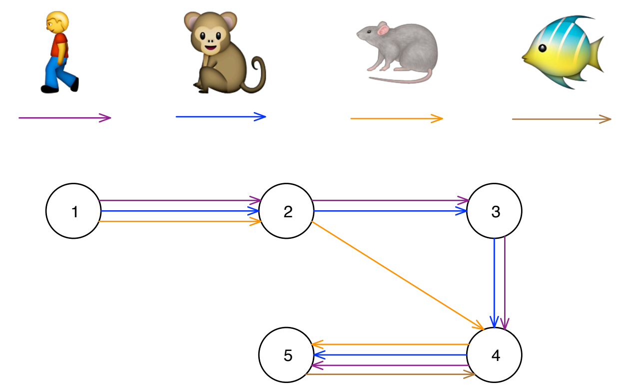
1,2,3,4

1,3,2,4

ScaDS
DRESDEN LEIPZIG

SIMULTANEOUS CONSECUTIVE ONES

	1	2	3	4	5
1-2	1	1	0	0	0
2-3	0	1	1	0	0
2-4	0	1	0	1	0
3-4	0	0	1	1	0
4-5	0	0	0	1	1
5-4	0	0	0	1	1



- Matrix with vertices as columns and adjacencies as rows
- Sort both the rows and columns of the matrix independently
- In such a way that rows and columns show all non-zero entries consecutively

SIMULTANEOUS CONSECUTIVE ONES

	1	2	3	4	5
1-2	1	1	0	0	0
2-3	0	1	1	0	0
2-4	0	1	0	1	0
3-4	0	0	1	1	0
4-5	0	0	0	1	1
5-4	0	0	0	1	1

	1	2	3	5	4
3-4	0	0	1	0	1
2-3	0	1	1	0	0
1-2	1	1	0	0	0
2-4	0	1	0	0	1
4-5	0	0	0	1	1
5-4	0	0	0	1	1

	1	3	2	4	5
1-2	1	0	1	0	0
2-3	0	1	1	0	0
3-4	0	1	0	1	0
2-4	0	0	1	1	0
4-5	0	0	0	1	1
5-4	0	0	0	1	1

- Consecutive ones property is violated even when betweenness is intact
- Bad adjacencies have huge impact
- Does not fit betweenness problem well