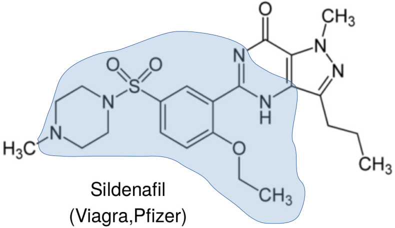
Finding Largest Common Substructures of Molecules in Quadratic Time

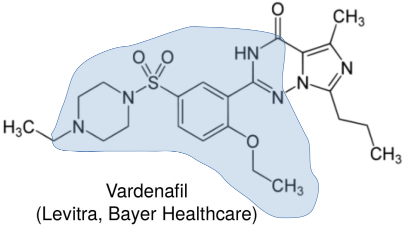
Andre Droschinsky Nils Kriege Petra Mutzel

Dept. of Computer Science, TU Dortmund University, Dortmund, Germany

43rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-FOCS 2017)
January 16 – 20, 2017



Sildenafil
(Viagra,Pfizer)



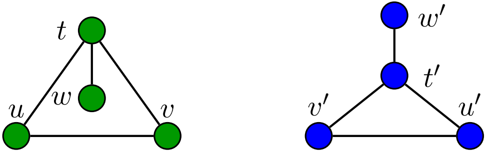
Vardenafil
(Levitra, Bayer Healthcare)

Motivation

- Common Substructure is a natural measurement of similarity
  - Useful for prediction of biological activity and reaction site modeling [Raymond, Willett 2002]

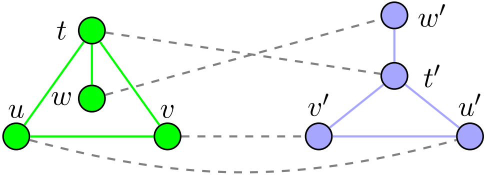
Graph isomorphism

Input: Graphs G and H



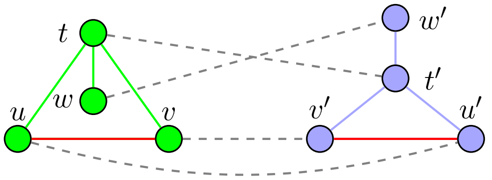
Graph isomorphism

Input: Graphs G and H
Output: A Bijection $\varphi : V_G \to V_H$ with
$\forall x, y \in V_G$: $xy \in E_G \Leftrightarrow \varphi(x)\varphi(y) \in E_H$



Graph isomorphism

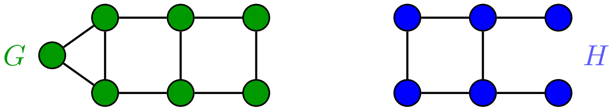
Input: Graphs G and H
Output: A Bijection $\varphi : V_G \to V_H$ with
$\forall x, y \in V_G$: $xy \in E_G \Leftrightarrow \varphi(x)\varphi(y) \in E_H$



Example: $uv \in E_G$ and $u'v' \in E_H$
$uw \notin E_G$ and $u'w' \notin E_H$

Maximum Common *Edge* Subgraph (MCES)

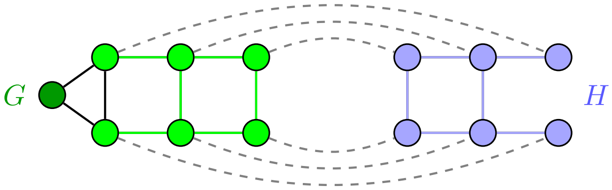
Input: Graphs G and H



Maximum Common *Edge* Subgraph (MCES)

Input: Graphs G and H

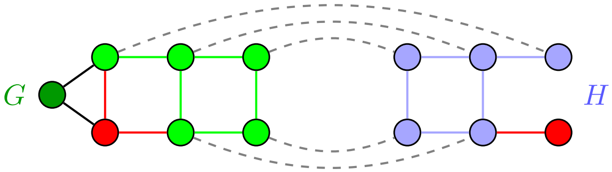
Output: An isomorphism between connected subgraphs of G and H with the maximum possible number of edges



Maximum Common *Induced* Subgraph (MCIS)

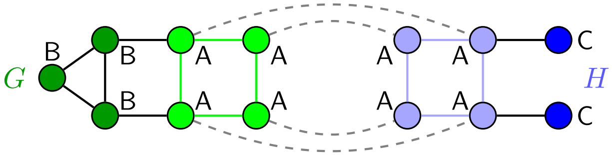
Input: Graphs G and H

Output: An isomorphism between connected *induced* subgraphs of G and H with the maximum possible number of *vertices*

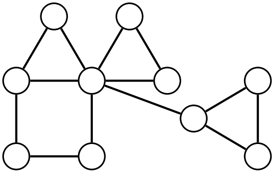


Maximum Common *Induced* Subgraph; labeled

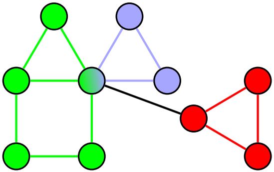
Input: Labeled graphs G and H
Output: A maximum common induced subgraph with respect to the labels



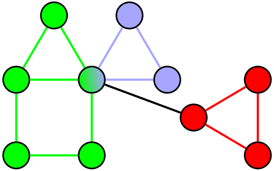
Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face



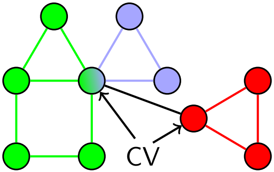
Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face
Block: A maximal biconnected subgraph



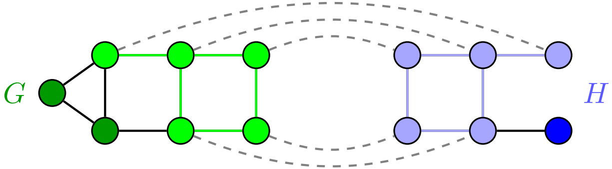
Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face
Block: A maximal biconnected subgraph
Bridge: Each remaining edge with its incident vertices



Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face
Block: A maximal biconnected subgraph
Bridge: Each remaining edge with its incident vertices
Cut vertex: Splits the graph if removed

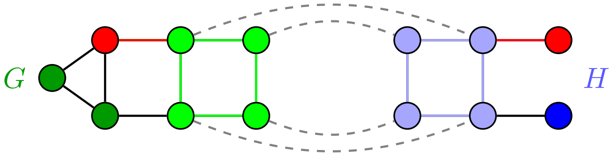


Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face
Block: A maximal biconnected subgraph
Bridge: Each remaining edge with its incident vertices
Cut vertex: Splits the graph if removed



MCIS

Outerplanar graph: A graph admitting a planar embedding with each vertex on the outer face
Block: A maximal biconnected subgraph
Bridge: Each remaining edge with its incident vertices
Cut vertex: Splits the graph if removed



MCIS with Block and Bridge Preserving (BBP)

BBP1: Blocks of common subgraph ↦ blocks of G and H
BBP2: Bridges of common subgraph ↦ bridges of G and H

Complexity of Maximum Common Subgraph

NP-hard on the following graph classes

- General graphs
- Outerplanar graphs [Syslo 1982]
- Trees, if we want to find a common forest [Brandenburg 2000]

Complexity

Polynomial time results on the following graph classes

- Trees: $\mathcal{O}(|G||H|\Delta)$ [D., K., M. 2016]
- Outerplanar graphs, MCS is biconnected
  - MCES: $\mathcal{O}(|G||H|)$ [Schietgat, Ramon, Bruynooghe 2013]
  - MCIS: $\mathcal{O}(|G||H|)$ [Kriege 2015]

Complexity

Polynomial time results on the following graph classes

- Trees: $\mathcal{O}(|G||H|\Delta)$ [D., K., M. 2016]
- Outerplanar graphs, MCS is biconnected
  - MCES: $\mathcal{O}(|G||H|)$ [Schietgat, Ramon, Bruynooghe 2013]
  - MCIS: $\mathcal{O}(|G||H|)$ [Kriege 2015]
- Outerplanar graphs, MCS with BBP property
  - MCES: $\mathcal{O}(n^4)$ [Schietgat, Ramon, Bruynooghe 2013]
  - MCIS: $\mathcal{O}(|G||H|\Delta)$ [This contribution]

Complexity

Polynomial time results on the following graph classes

- Trees: $\mathcal{O}(|G||H|\Delta)$ [D., K., M. 2016]
- Outerplanar graphs, MCS is biconnected
  - MCES: $\mathcal{O}(|G||H|)$ [Schietgat, Ramon, Bruynooghe 2013]
  - MCIS: $\mathcal{O}(|G||H|)$ [Kriege 2015]
- Outerplanar graphs, MCS with BBP property
  - MCES: $\mathcal{O}(n^4)$ [Schietgat, Ramon, Bruynooghe 2013]
  - MCIS: $\mathcal{O}(|G||H|\Delta)$ [This contribution]

BBP in molecular graphs

- Ring structures are kept intact
- Computation time of few ms

Computing a BBP-MCIS between two outerplanar graphs

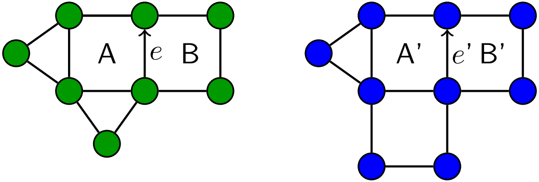
1. Biconnected MCIS between biconnected outerplanar graphs

Computing a BBP-MCIS between two outerplanar graphs

1. Biconnected MCIS between biconnected outerplanar graphs
2. Connect the blocks and bridges → BBP-MCIS

Properties of biconnected outerplanar graphs

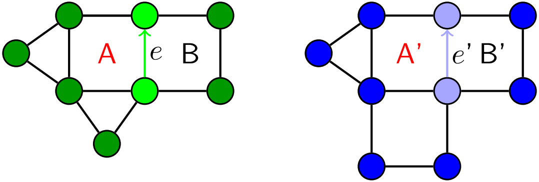
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

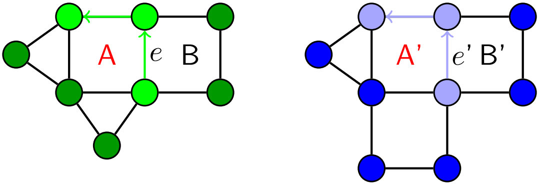
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

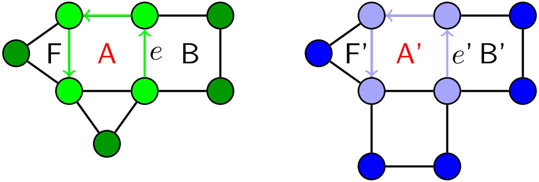
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

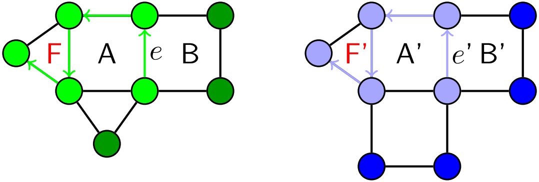
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

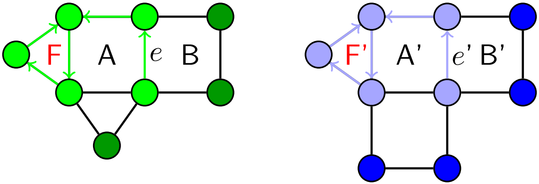
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

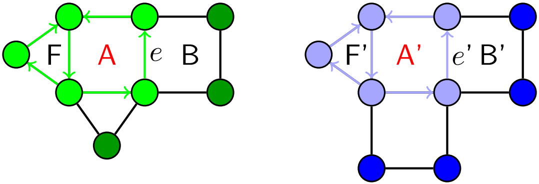
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

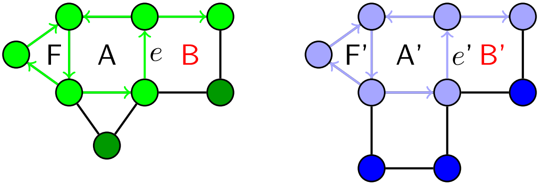
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

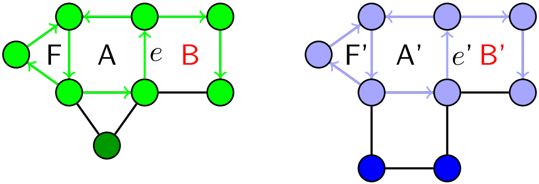
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

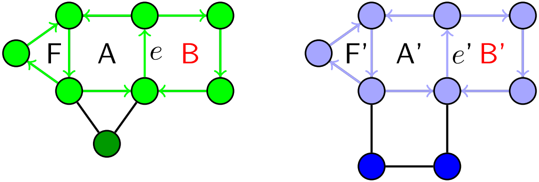
- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Properties of biconnected outerplanar graphs

- Outerplanar embedding is unique
- Each edge is incident to exactly two uniquely defined faces



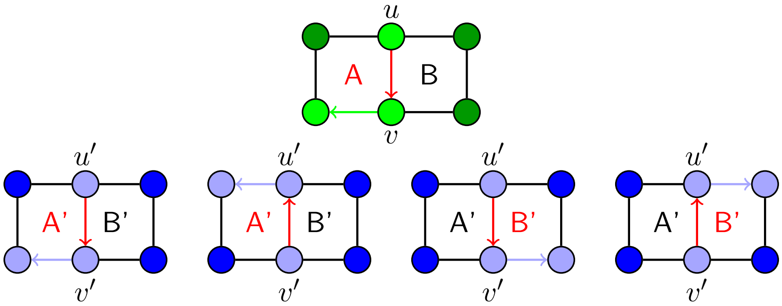
Lemma: Given an arc and face mapping, there is exactly one maximal isomorphism fulfilling that mapping.

Algorithm and running time

- Compute all maximal solutions to obtain maximum

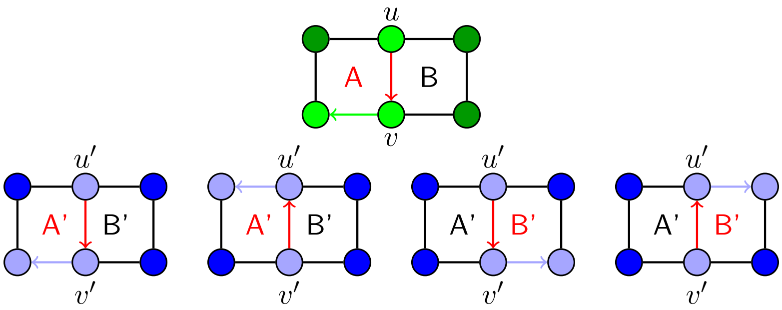
Algorithm and running time

- Compute all maximal solutions to obtain maximum
- 4 possible types of mappings for each edge pair



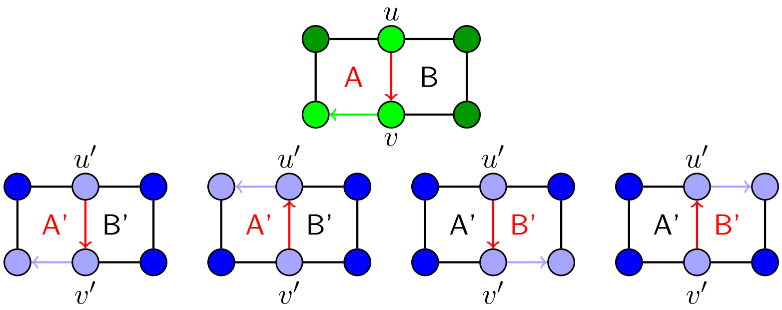
Algorithm and running time

- Compute all maximal solutions to obtain maximum
- 4 possible types of mappings for each edge pair
- Table of size $4|E_G||E_H|$ to store the sizes



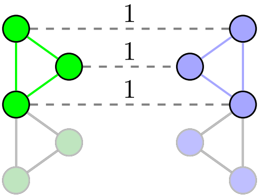
Algorithm and running time

- Compute all maximal solutions to obtain maximum
- 4 possible types of mappings for each edge pair
- Table of size $4|E_G||E_H|$ to store the sizes
- Time per cell $\mathcal{O}(1) \to$ quadratic total time



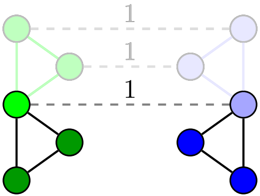
BBP-MCIS between outerplanar graphs – Example 1

1) Compute a maximal isomorphisms between two blocks



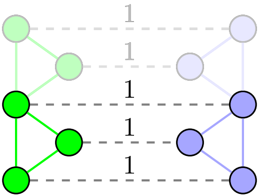
BBP-MCIS between outerplanar graphs – Example 1

1) Compute a maximal isomorphisms between two blocks
2) Recursively extend it along cut vertices; consider all possibilities



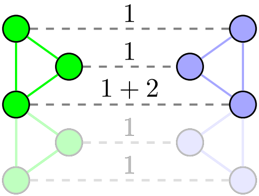
BBP-MCIS between outerplanar graphs – Example 1

1) Compute a maximal isomorphisms between two blocks
2) Recursively extend it along cut vertices; consider all possibilities



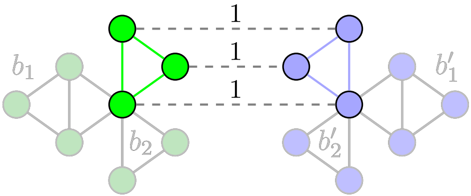
BBP-MCIS between outerplanar graphs – Example 1

1) Compute a maximal isomorphisms between two blocks
2) Recursively extend it along cut vertices; consider all possibilities
3) Add size of extension to cut vertices → Total size 5.



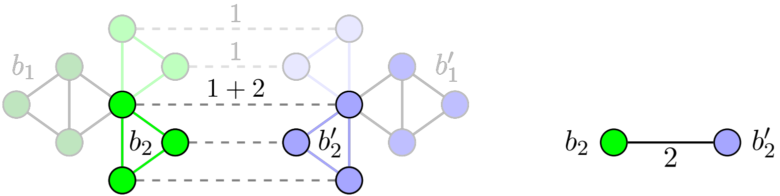
BBP-MCIS between outerplanar graphs – Example 2

1) Compute a maximal isomorphisms between two blocks



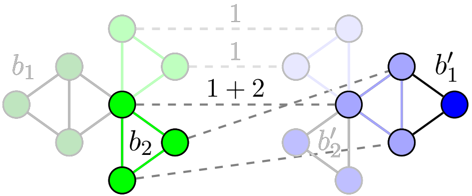
BBP-MCIS between outerplanar graphs – Example 2

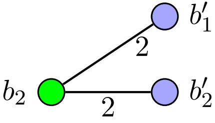
1) Compute a maximal isomorphisms between two blocks
2) Try extensions separately for each pair of adjacent blocks



BBP-MCIS between outerplanar graphs – Example 2

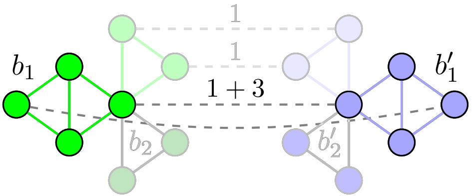
1) Compute a maximal isomorphisms between two blocks
2) Try extensions separately for each pair of adjacent blocks

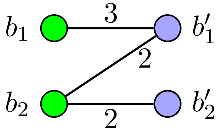




BBP-MCIS between outerplanar graphs – Example 2

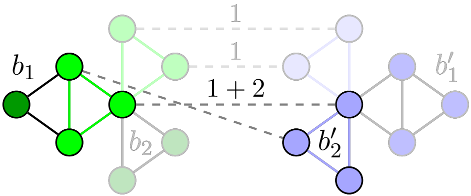
1) Compute a maximal isomorphisms between two blocks
2) Try extensions separately for each pair of adjacent blocks

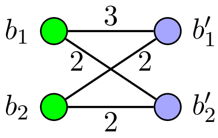




BBP-MCIS between outerplanar graphs – Example 2

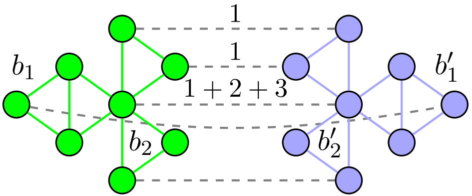
1) Compute a maximal isomorphisms between two blocks
2) Try extensions separately for each pair of adjacent blocks

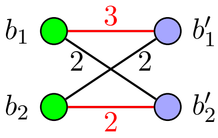




BBP-MCIS between outerplanar graphs – Example 2

1) Compute a maximal isomorphisms between two blocks
2) Try extensions separately for each pair of adjacent blocks
3) Compute maximum weight matching for block to block mapping





Theorem (Main result)

BBP-MCIS between two outerplanar graphs G and H can be solved in time $\mathcal{O}(|G||H|\Delta)$.

$\Delta =$ Maximum degree of all cut vertices (or 1, if none present)

Corollary

The time complexity of BBP-MCIS between outerplanar molecular graphs G and H is $\Theta(|G||H|)$.

Test setup

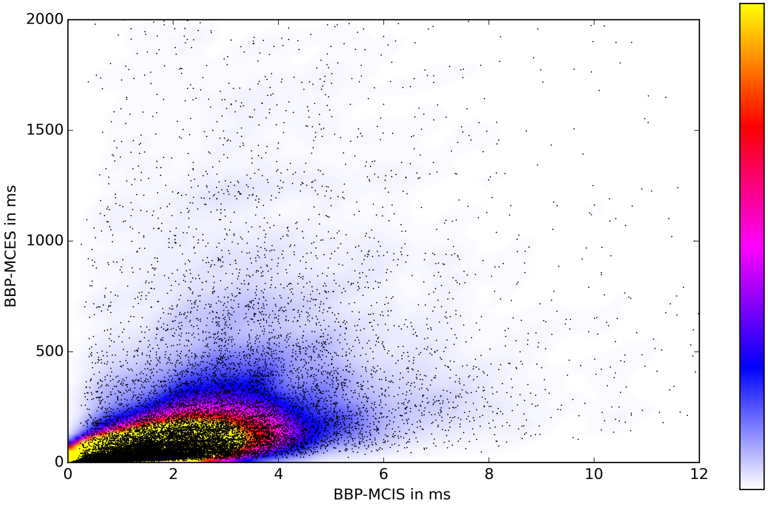
- Molecular graphs from the NCI Open Database GI50
  - 29 000 randomly chosen pairs of outerplanar graphs
  - Up to 104 vertices; average size 22 vertices
- Comparison to BBP-MCES from Schietgat et al.
  - No other BBP-MCIS algorithm available
  - Source kindly provided by Leander Schietgat
  - Goal: Maximize number of edges+vertices
- Both sources compiled with GCC; run on Intel i7-3770 CPU

Results

- BBP-MCES/MCIS differ in only 0.40% of 29 000 randomly chosen pairs of outerplanar molecular graphs.
- (fastest) BBP-MCIS – BBP-MCES – general MCS (slowest)

Table : Running times in ms on randomly chosen molecular graphs

Algorithm	Average	Median	95% less than	Maximum
MCIS	1.97 ms	1.51 ms	5.28 ms	40.35 ms
MCES	207.08 ms	41.43 ms	871.48 ms	26 353.68 ms



(601 of 29 000 BBP-MCS computations did not fit into the borders.)

Conclusion

- First efficient BBP-MCIS computation in theory and practice
  - Supports labels with nonnegative weights attached
- Much faster than BBP-MCES; identical results in 99.6%

Future work

- BBP-MCIS for non outerplanar molecular graphs
- Negative weights; bounded integer weights

Conclusion

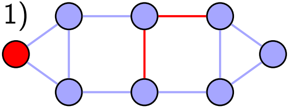
- First efficient BBP-MCIS computation in theory and practice
  - Supports labels with nonnegative weights attached
- Much faster than BBP-MCES; identical results in 99.6%

Future work

- BBP-MCIS for non outerplanar molecular graphs
- Negative weights; bounded integer weights

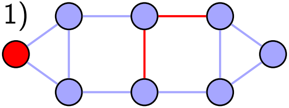
Labeled graphs, post processing

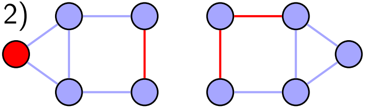
1) Biconnected CIS with different labels; colored red



Labeled graphs, post processing

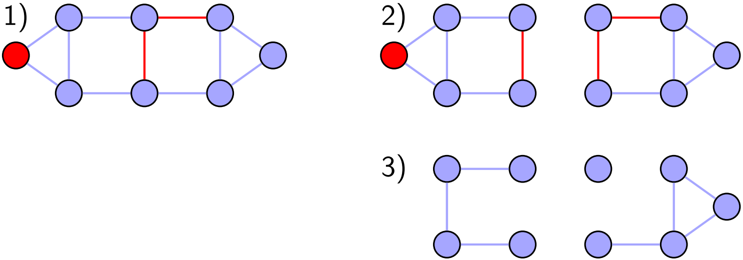
1) Biconnected CIS with different labels; colored red
2) Separate graph along inner edges with different labels.





Labeled graphs, post processing

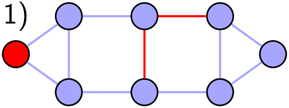
1) Biconnected CIS with different labels; colored red
2) Separate graph along inner edges with different labels.
3) Remove edges and vertices with different labels.



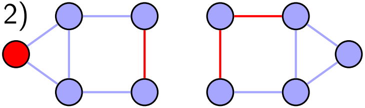
Labeled graphs, post processing

1) Biconnected CIS with different labels; colored red
2) Separate graph along inner edges with different labels.
3) Remove edges and vertices with different labels.
4) Strip non-block parts; store size for each component.

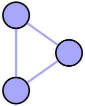
1)



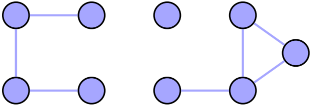
2)



4)



3)



Labeled graphs, post processing

1) Biconnected CIS with different labels; colored red
2) Separate graph along inner edges with different labels.
3) Remove edges and vertices with different labels.
4) Strip non-block parts; store size for each component.
5) Store $-\infty$ for removed edge mappings.