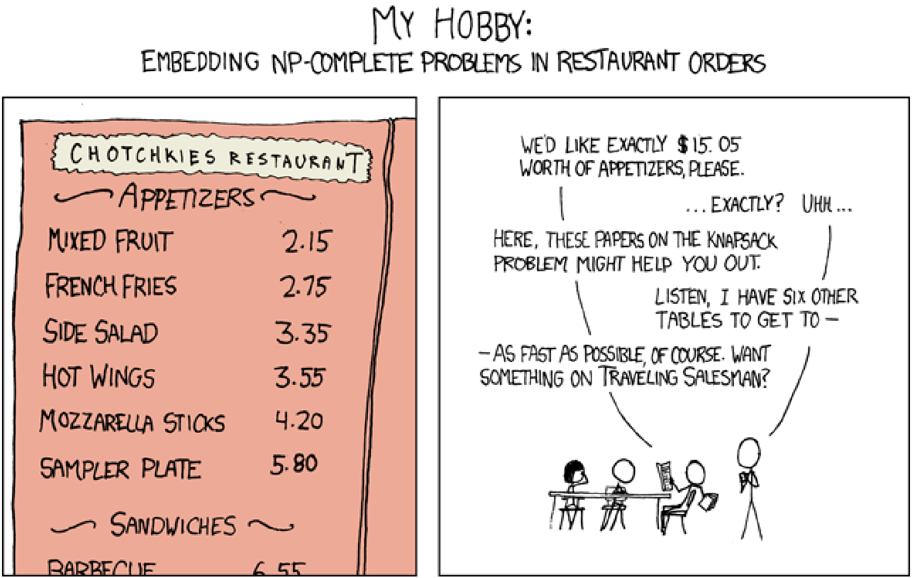
Correlation Clustering and Two-edge-connected Augmentation for Planar Graphs

Philip N. Klein[1], Claire Mathieu[2,3], and Hang Zhou[3]

[1]Brown University, United States

[2]CNRS, France

[3]École Normale Supérieure de Paris, France


MY HOBBY:
EMBEDDING NP-COMPLETE PROBLEMS IN RESTAURANT ORDERS
CHOTCHKIES RESTAURANT
APPETIZERS
MIXED FRUIT 2.15
FRENCH FRIES 2.75
SIDE SALAD 3.35
HOT WINGS 3.55
MOZZARELLA STICKS 4.20
SAMPLER PLATE 5.80
SANDWICHES
WE'D LIKE EXACTLY $15.05 WORTH OF APPETIZERS, PLEASE.
...EXACTLY? UHH...
HERE, THESE PAPERS ON THE KNAPSACK PROBLEM MIGHT HELP YOU OUT.
LISTEN, I HAVE SIX OTHER TABLES TO GET TO –
– AS FAST AS POSSIBLE, OF COURSE. WANT SOMETHING ON TRAVELING SALESMAN?

Approximation Schemes for Planar Graphs

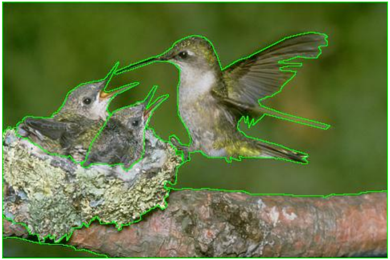
- Local problems (1977-1983):
  independent set, vertex cover, dominating set, etc.
- Connectivity problems (2005-2011):
  TSP, Steiner, 2-edge-connected subgraph, etc.
- Other problems (2012-2014):
  multiway cut, k-center.
- In our work:
  correlation clustering, 2-edge-connected augmentation.

Correlation Clustering

Input: a graph with edge-weights, where every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$ according to *similarity of its endpoints*

Output: a partition of the vertices that disagrees with the edge labels as little as possible

Motivated by **image segmentation**.



(an example from Berkeley Segmentation Dataset)

Correlation Clustering

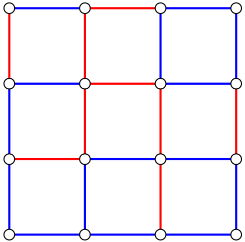
Input: a graph with edge-weights, where every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$ according to *similarity of its endpoints*

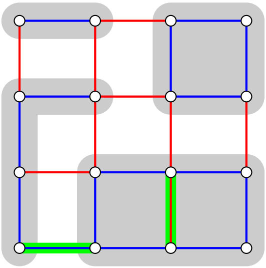
Output: a partition of the vertices that disagrees with the edge labels as little as possible

Correlation Clustering

Input: a graph with edge-weights, where every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$ according to *similarity of its endpoints*

Output: a partition of the vertices that disagrees with the edge labels as little as possible





Correlation Clustering

Input: a graph with edge-weights, where every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$ according to *similarity of its endpoints*

Output: a partition of the vertices that disagrees with the edge labels as little as possible

Our Result

PTAS for correlation clustering in planar graphs

Previous results:

- Constant-factor approximation for minor-excluded graphs [Demaine, Emanuel, Fiat, Immorlica, 2006]
- NP-hardness for planar graphs [Bachrach, Kohli, Kolmogorov, Zadimoghaddam, 2013]
- APX-hardness for general graphs [Bansal, Blum, Chawla, 2004]

Two-Edge-Connected Augmentation

Goal: Augment input subset R of edges of a graph so that the result is a collection of 2-edge-connected components.

Our Result

PTAS for two-edge-connected augmentation in planar graphs

Previous results for general graphs:

- APX-hardness [Kortsarz, Krauthgamer, Lee, 2004]
- 2-approximation [Jain, 2001]

Techniques

Reduction from correlation clustering to 2-edge-connected augmentation

- planar duality

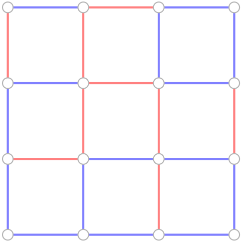
PTAS for 2-edge-connected augmentation

- prize-collecting clustering
- brick decomposition
- sphere-cut decomposition
- dynamic programming

Reduction

Reduction from correlation clustering to 2-edge-connected augmentation:

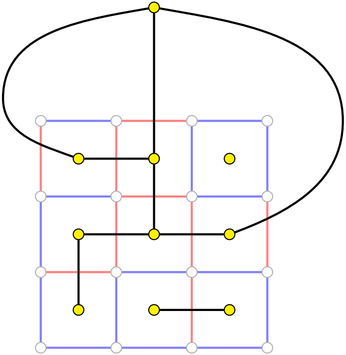
1. Start from an instance of correlation clustering
2. Construct an instance of 2-edge-connected augmentation:
   - the graph is the dual of the correlation clustering graph
   - the set R contains the duals of the $\langle - \rangle$ edges
3. Solve the 2-edge-connected augmentation instance
4. Switch the labels of the dual edges of the augmentation to obtain a partition



Reduction

Reduction from correlation clustering to 2-edge-connected augmentation:

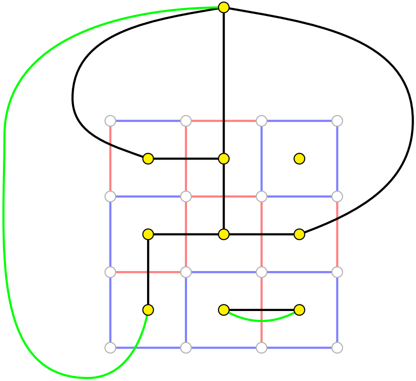
1. Start from an instance of correlation clustering
2. Construct an instance of 2-edge-connected augmentation:
    - the graph is the dual of the correlation clustering graph
    - the set R contains the duals of the $\langle - \rangle$ edges
3. Solve the 2-edge-connected augmentation instance
4. Switch the labels of the dual edges of the augmentation to obtain a partition



Reduction

Reduction from correlation clustering to 2-edge-connected augmentation:

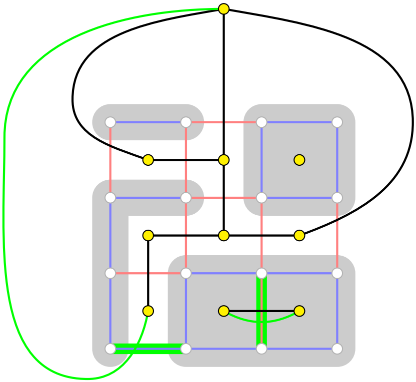
1. Start from an instance of correlation clustering
2. Construct an instance of 2-edge-connected augmentation:
   - the graph is the dual of the correlation clustering graph
   - the set R contains the duals of the $\langle - \rangle$ edges
3. Solve the 2-edge-connected augmentation instance
4. Switch the labels of the dual edges of the augmentation to obtain a partition



Reduction

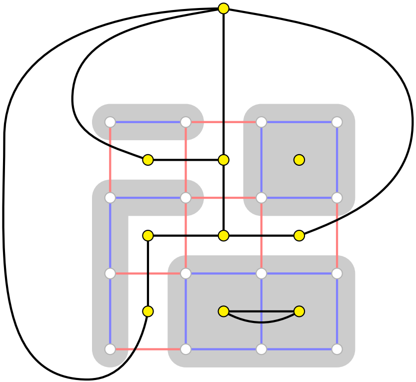
Reduction from correlation clustering to 2-edge-connected augmentation:

1. Start from an instance of correlation clustering
2. Construct an instance of 2-edge-connected augmentation:
   - the graph is the dual of the correlation clustering graph
   - the set R contains the duals of the $\langle - \rangle$ edges
3. Solve the 2-edge-connected augmentation instance
4. Switch the labels of the dual edges of the augmentation to obtain a partition



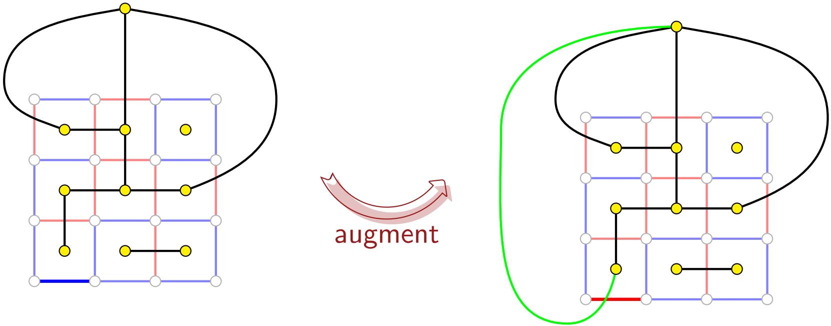
Reduction

Observation: A partition of $V[G]$ has a 1-to-1 correspondence to a collection of 2-edge-connected components of G^*.



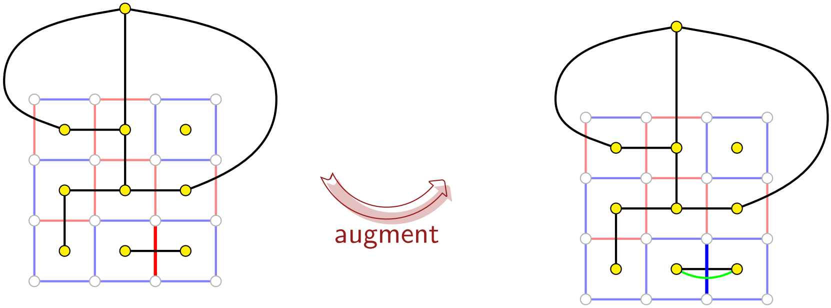
Reduction

- Switch a $\langle + \rangle$ edge to a $\langle - \rangle$ edge $\Leftrightarrow$ add an edge in the dual
- Switch a $\langle - \rangle$ edge to a $\langle + \rangle$ edge $\Leftrightarrow$ remove an edge in the dual $\Leftrightarrow$ double that edge in the dual
- Only need to find an augmentation in the dual so that every component is 2-edge-connected.



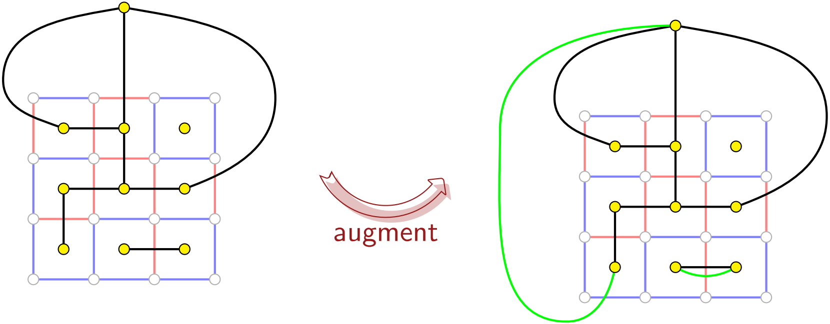
Reduction

- Switch a $\langle + \rangle$ edge to a $\langle - \rangle$ edge $\Leftrightarrow$ add an edge in the dual
- Switch a $\langle - \rangle$ edge to a $\langle + \rangle$ edge $\Leftrightarrow$ remove an edge in the dual $\Leftrightarrow$ double that edge in the dual
- Only need to find an augmentation in the dual so that every component is 2-edge-connected.



Reduction

- Switch a $\langle + \rangle$ edge to a $\langle - \rangle$ edge $\Leftrightarrow$ add an edge in the dual
- Switch a $\langle - \rangle$ edge to a $\langle + \rangle$ edge $\Leftrightarrow$ remove an edge in the dual $\Leftrightarrow$ double that edge in the dual
- Only need to find an augmentation in the dual so that every component is 2-edge-connected.



Reduction

- Switch a $\langle + \rangle$ edge to a $\langle - \rangle$ edge $\Leftrightarrow$ add an edge in the dual
- Switch a $\langle - \rangle$ edge to a $\langle + \rangle$ edge $\Leftrightarrow$ remove an edge in the dual $\Leftrightarrow$ double that edge in the dual
- Only need to find an augmentation in the dual so that every component is 2-edge-connected.

Theorem

There is an approximation-preserving reduction from correlation clustering to 2-edge-connected augmentation.

Approximation Scheme

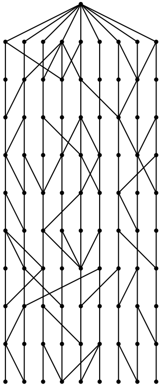
Universal paradigm

1. Reduce to bounded treewidth
2. Solve the problem by dynamic programming

Q: How do we reduce to bounded treewidth?

A: For local problems, Baker's basic framework:

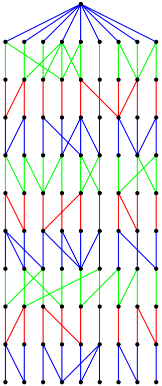
1. Breadth First Search
2. Delete edges on one level every $1/\epsilon$ levels



Approximation Scheme

Universal paradigm

1. Reduce to bounded treewidth
2. Solve the problem by dynamic programming



Q: How do we reduce to bounded treewidth?

A: For local problems, Baker's basic framework:

1. Breadth First Search
2. Delete edges on one level every $1/\epsilon$ levels

Approximation Scheme

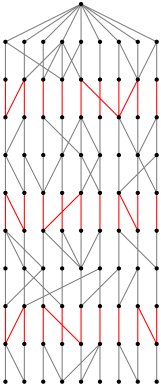
Universal paradigm

1. Reduce to bounded treewidth
2. Solve the problem by dynamic programming

Q: How do we reduce to bounded treewidth?

A: For local problems, Baker's basic framework:

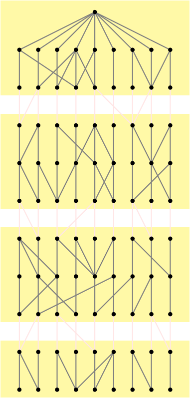
1. Breadth First Search
2. Delete edges on one level every $1/\epsilon$ levels



Approximation Scheme

Universal paradigm

1. Reduce to bounded treewidth
2. Solve the problem by dynamic programming



Q: How do we reduce to bounded treewidth?

A: For local problems, Baker's basic framework:

1. Breadth First Search
2. Delete edges on one level every $1/\epsilon$ levels

Klein's Dual Framework for Non-Local Problems

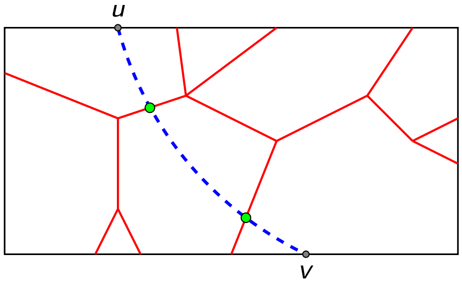
1. Compute a brick decomposition such that
   - The total weight of brick boundaries is $O(OPT)$;
   - Structure Property: A near-optimal solution inside each brick is simple.
2. Breadth First Search on the dual graph of the brick graph
3. Delete or contract edges on one level every $1/\epsilon$ levels

Ensure that the brick graph has bounded treewidth.

Our Contribution: A New Structural Property on Bricks

Structure Property

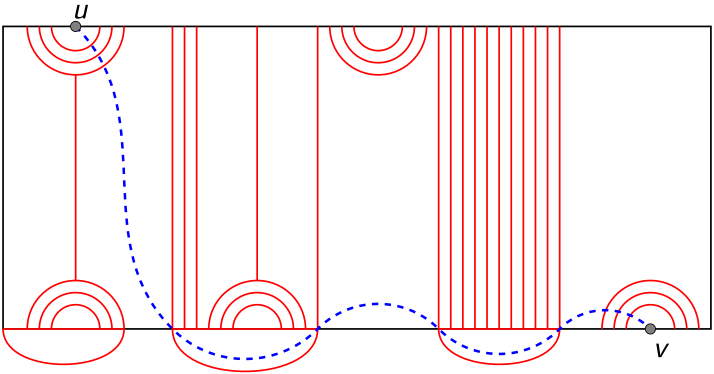
For any two vertices u, v on the boundary of a brick, there exists a u-to-v Jordan curve inside the brick that intersects the near-optimal solution at only a constant number of points.



Our Contribution: A New Structural Property on Bricks

Structure Property

For any two vertices u, v on the boundary of a brick, there exists a u-to-v Jordan curve inside the brick that intersects the near-optimal solution at only a constant number of points.



Our Contribution: A New Structural Property on Bricks

Structure Property

For any two vertices u, v on the boundary of a brick, there exists a u-to-v Jordan curve inside the brick that intersects the near-optimal solution at only a constant number of points.

Theorem

We have a PTAS for 2-edge-connected augmentation in planar graphs.

Conclusion

- Local problems (1977-1983):
  independent set, vertex cover, dominating set, etc.
- Connectivity problems (2005-2011):
  TSP, Steiner, 2-edge-connected subgraph, etc.
- Other problems (2012-2014):
  multiway cut, k-center.
- In our work:
  correlation clustering, 2-edge-connected augmentation.
- Open problem:
  Steiner version of the 2-edge-connected subgraph problem.

Thank you!