

# Finding 2-Factors Closer to TSP Tours in Cubic Graphs

Sylvia Boyd (U. Ottawa)

Satoru Iwata (U. Tokyo)

Kenjiro Takazawa (Kyoto U. / Lab. G-SCOP)

18th Aussois Combinatorial Optimization Workshop  
January 6-10, 2014

# Petersen's Theorem

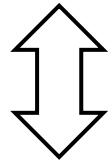
$G=(V,E)$ : Bridgeless Cubic Graph

= 2-edge-connected

$\deg(v) = 3$  for every  $v$  in  $V$

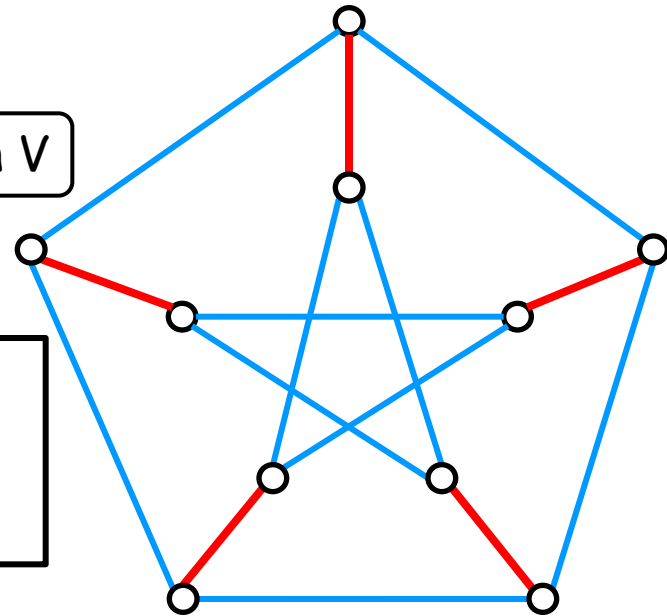
Thm.

Every bridgeless cubic graph has a **perfect matching**



Every bridgeless cubic graph has a **2-factor**

[1891]



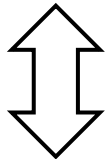
# Schönberger's Theorem

$G=(V,E)$ : Bridgeless Cubic Graph

$e^*$  in  $E$

Thm.

$G$  has a **perfect matching** including  $e^*$

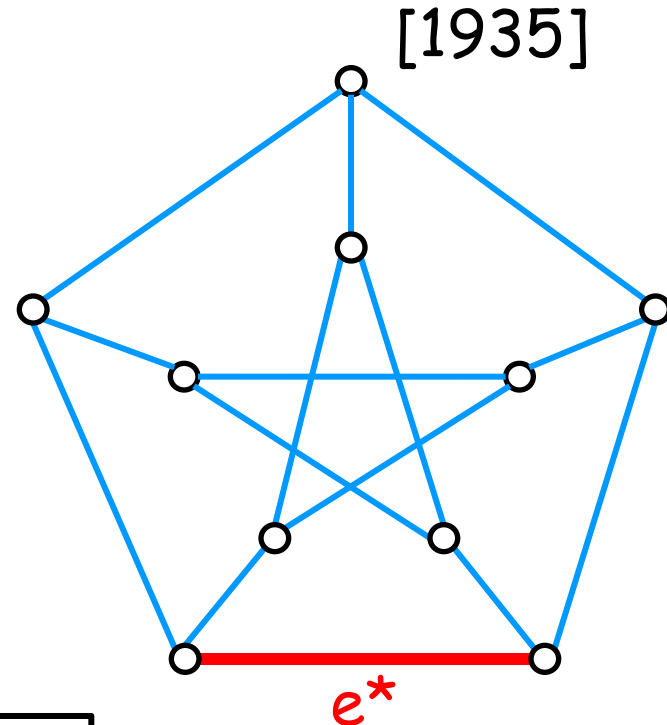


$G$  has a **2-factor** excluding  $e^*$

➤  $O(n \log^4 n)$  algorithm

[Biedl, Bose, Demaine, Lubiw 2001]

$$n = |V|$$



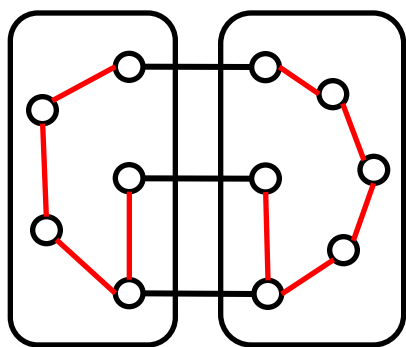
# Kaiser & Škrekovski's Theorem

$G=(V,E)$ : Bridgeless Cubic

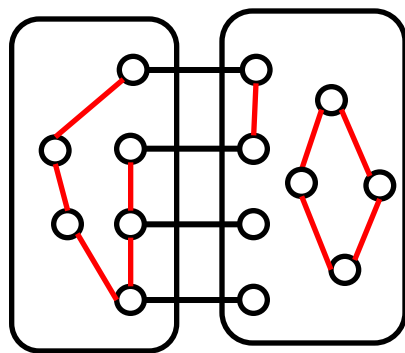
[2008]

$e^*$  in  $E$

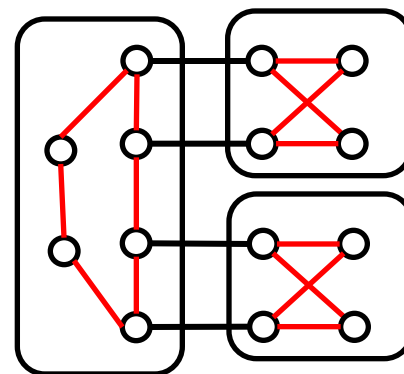
Thm.  $G$  has a **2-factor** excluding  $e^*$  and covering all 3- and 4-edge cuts



3-edge cut



4-edge cut

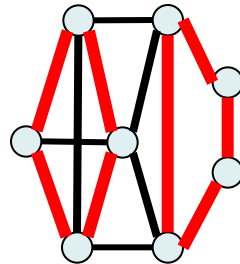
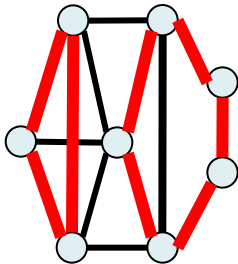


Not a 4-edge cut

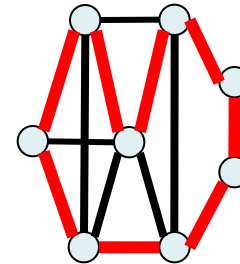
# 2-factors and TSP Tours

- TSP tour = 2-factor of one cycle of length  $n$

→ 2-factor **without cycles of length  $k$  or less** :  
**Relax**  $C_{\leq k}$ -free 2-factor (in simple graphs)



✓  $C_{\leq 3}$ -free



✓  $C_{\leq 4}$ -free

$k = n/2 \rightarrow$  TSP tour

# Complexity of $C_{\leq k}$ -free 2-factors

	Unweighted	Weighted
$k \geq 5$	NP-hard [Papadimitriou '80]	NP-hard
$k = 4$	(a) OPEN	(b) NP-hard [Vornberger '80]
$k = 3$	(c) P [Hartvigsen '84]	(d) OPEN
$k = 2$	P	P

- Subcubic graphs

(a) : P [Bérczi & Végh '10]

(c) : P [Bérczi & Végh '10, Hartvigsen & Li '11]

(d) : P [Vornberger '80, Kobayashi '10, Hartvigsen & Li '13]

- Bipartite graphs

(a) : P [Hartvigsen '06, Pap '07]

(b) : NP-hard for general weight [Király 00]

P if the weight has a special property [Makai '07, T. '09]

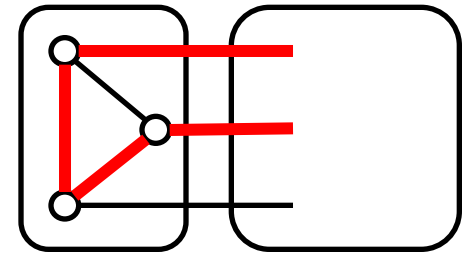
# 2-factors Covering Cuts

- TSP tour = 2-factor covering all edge cuts

→ 2-factor covering **prescribed edge cuts**  
Relax

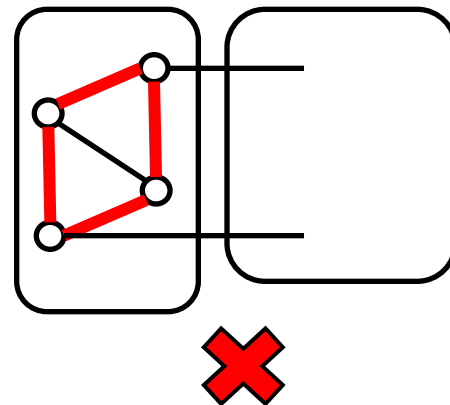
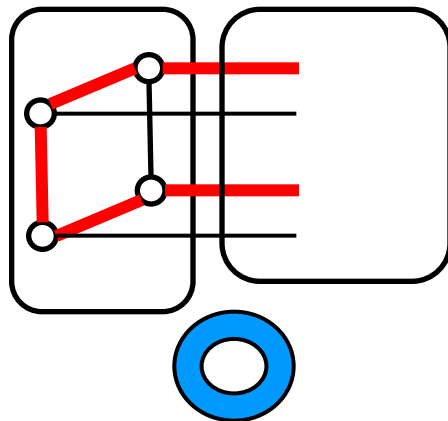
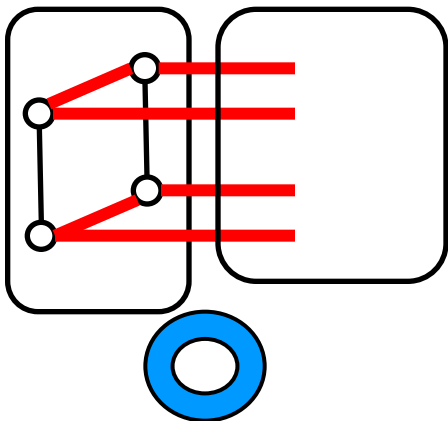
$G$ : Cubic

- 2-factor covering all 3-edge cuts  
→  $C_{\leq 3}$ -free



$G$ : 3-edge-connected cubic

- 2-factor covering all 3,4-edge cuts →  $C_{\leq 4}$ -free



# Our Results

(1) - An  $O(n^3)$ -algorithm for finding a *min.-weight* 2-factor covering all *3-edge cuts* in *bridgeless* cubic graphs  
- Polyhedral description

(2) An  $O(n^3)$ -algorithm for finding a 2-factor covering all *3-, 4-edge cuts* in *bridgeless* cubic graphs

➤ Constructive proof for [Kaiser, Škrekovski 2008]

Application



(3) A *6/5-approx.* algorithm for the *minimum 2-edge-connected subgraph problem* in *3-edge-connected* cubic graphs

➤ Start with the 2-factor found by Algorithm (2)

➤ Previous ratio: *5/4* for 3-edge-connected cubic graphs [Huh 2004]



# Contents

- Introduction

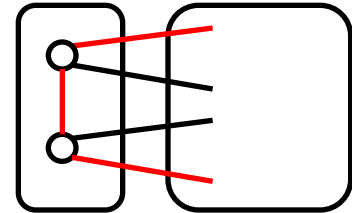
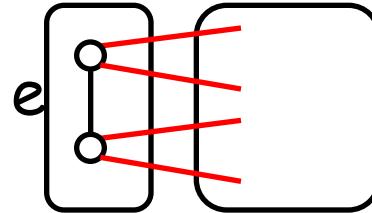
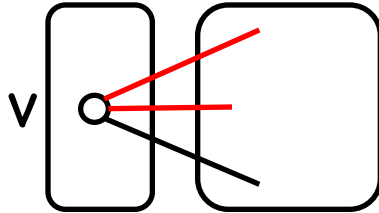
(2) An  $O(n^3)$  algorithm for finding a 2-factor covering all 3-, 4-edge cuts in bridgeless cubic graphs

(3) A  $6/5$ -approx. algorithm for the minimum 2-edge-connected subgraph problem in 3-edge-connected cubic graphs

- Summary

# Proper 3- and 4-Edge Cuts

- 3- and 4-edge cuts covered by *every* 2-factor

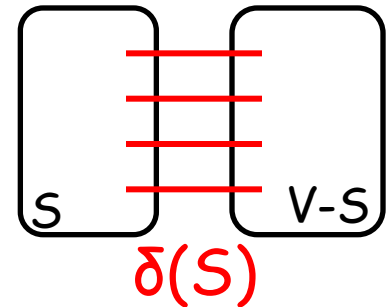
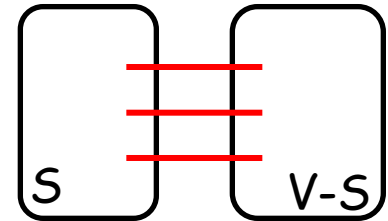


- A 3-edge cut  $\delta(S)$  is **proper**

$$\iff 2 \leq |S| \leq n - 2$$

- A 4-edge cut  $\delta(S)$  is **proper**

$$\iff 3 \leq |S| \leq n - 3$$



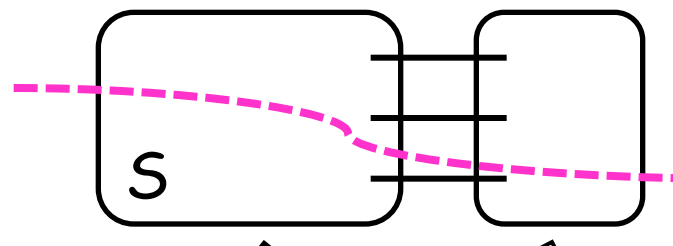
## Goal

Find a 2-factor  $F$  satisfying:

- Covering all **proper** 3- and 4-edge cuts
- Excluding an edge  $e^*$  in  $E$

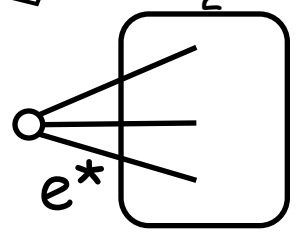
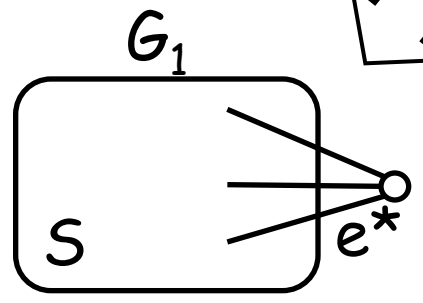
# Covering 3-Edge Cuts Gluing technique in [Cornuéjols, Naddef, Pulleyblank 85]

(1) Find a **proper** 3-edge cut  $\delta(S)$



(2) Contract  $V - S, S$

→ **Smaller**  
bridgeless cubic graphs



(3) Recurse →  $F_1$

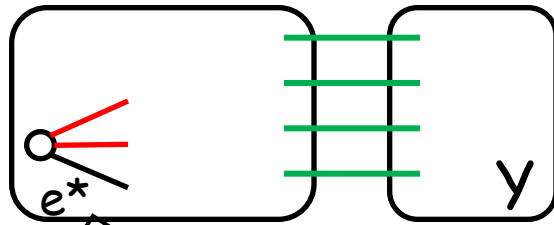
(4) Recurse →  $F_2$   
(Exclude  $e^*$ )

**F** covers all 3- & 4-edge cuts

(5) Return  $F = F_1 + F_2$

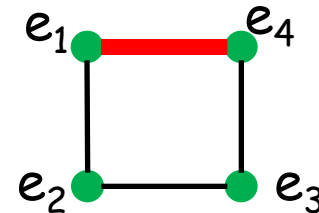
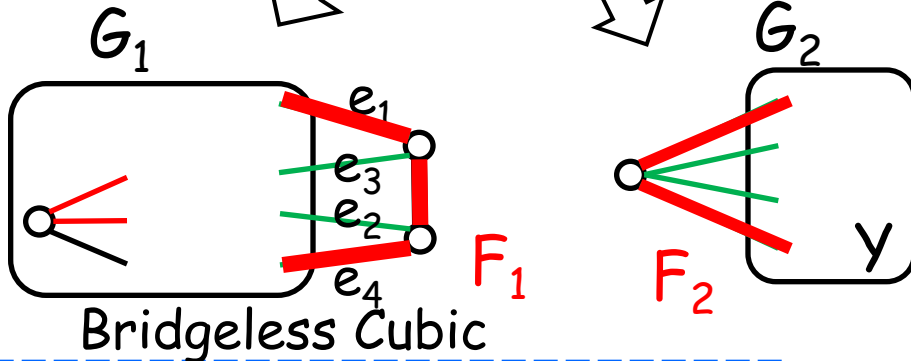
# Covering 4-Edge Cuts

(1) Find a proper 4-edge cut  $\delta(Y) = \{e_1, e_2, e_3, e_4\}$  ( $Y$ : minimal)



(2) Contract  $V - Y$

(3) For any pair  $e_i, e_j$ , check if  $G_2$  has a 2-factor including  $\{e_i, e_j\}$

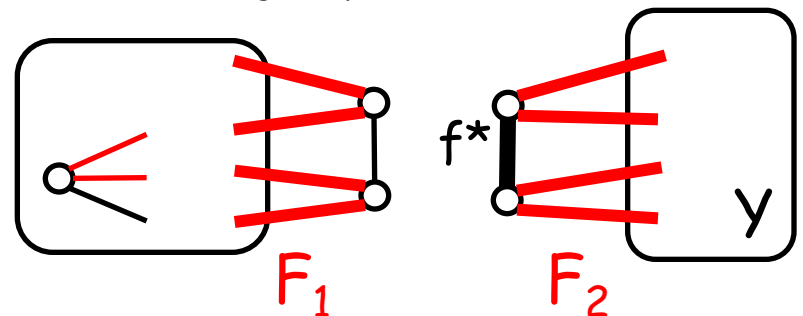


(4) Contract  $Y$  to  $v_y$ , split  $v_y$  according to (3)

(5) Recurse  $\rightarrow F_1$

(6) Return  $F = F_1 + F_2$

$\{e_1, e_2, e_3, e_4\}$  in  $F_1$



# Contents

- Introduction

(2) An  $O(n^3)$  algorithm for finding a 2-factor covering all 3-, 4-edge cuts in bridgeless cubic graphs

(3) A **6/5-approx.** algorithm for the **minimum 2-edge-connected subgraph problem** in **3-edge-connected** cubic graphs

- Summary

# The Minimum 2-Edge-Connected Subgraph Problem

Input: Graph  $G = (V, E)$

Goal: 2-edge-connected subgraph  $(V, E')$  minimizing  $|E'|$

- Hamilton cycle → Optimal solution
- $n$ : lower bound

## ● General graphs

Khuller, Vishkin ('94), Cheriyan, Sebő, Szigeti ('98)

Vempala, Vetta ('00), Jothi, Raghavachari, Varadarajan ('03)

Sebő, Vygen ('13): **4/3-approx.**

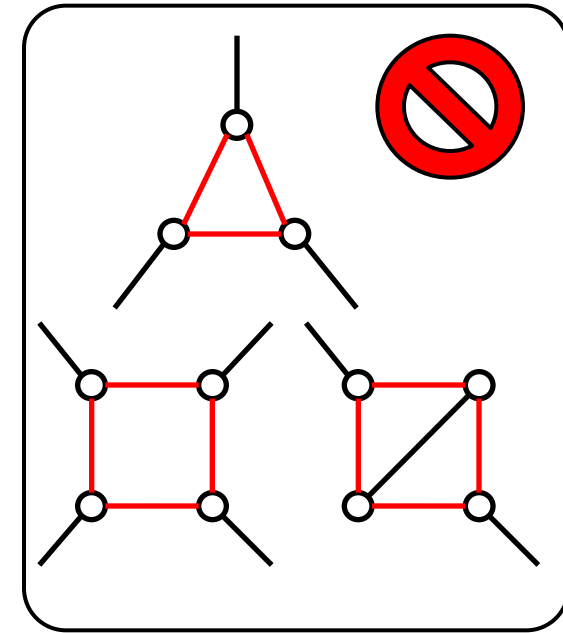
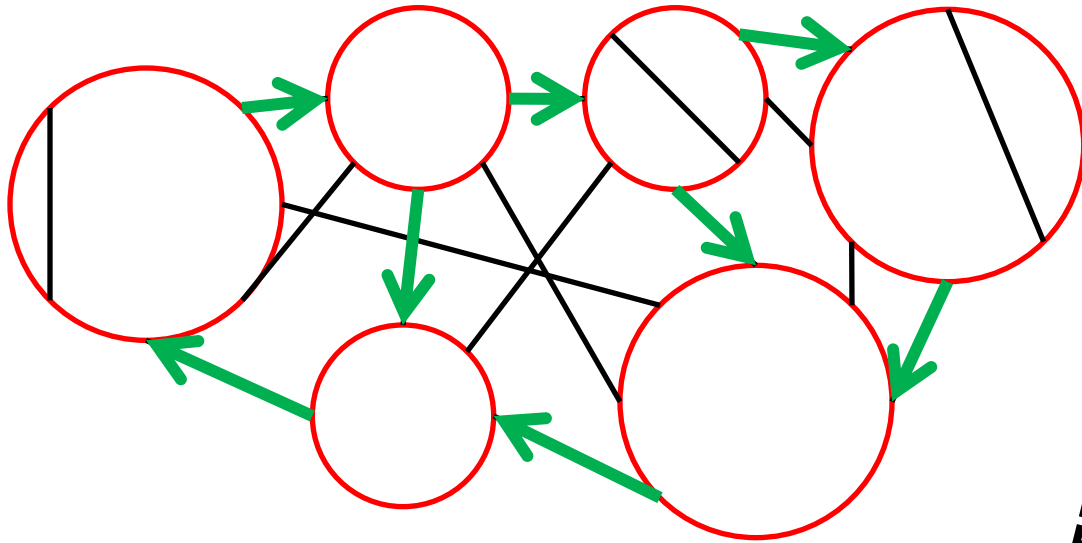
## ● 3-edge-connected cubic graphs

Huh ('04): **5/4-approx.**

**This talk: 6/5-approx**

# Rough Idea

**F**: 2-factor covering all 3- and 4-edge cuts  
→ Cycles of length  $\geq 5$

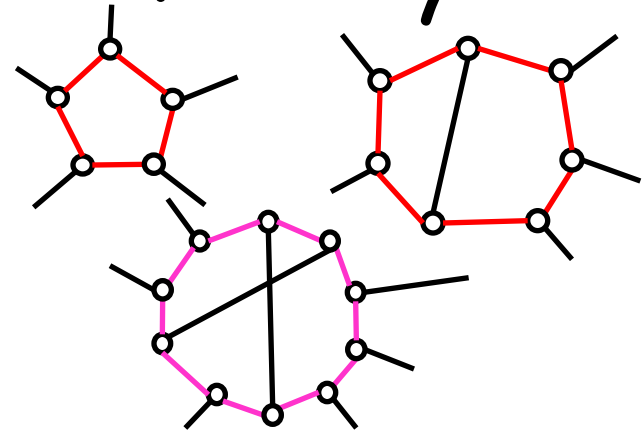


2 extra edges for each cycle  
→  $7/5$ -approx. 😞

# Saving 1 Edge in a Small Cycle

Cycles in  $F$  :

- **Small cycle**: Size 5--9
- **Large cycle**: Size  $\geq 10$

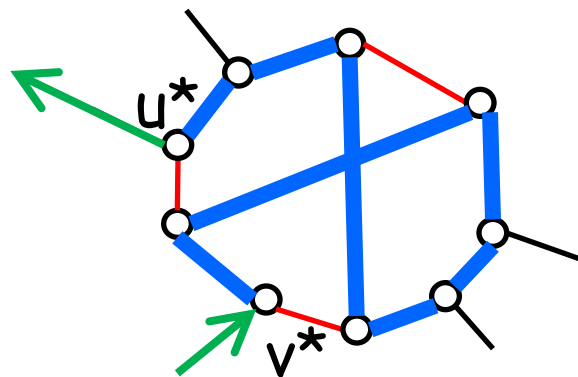
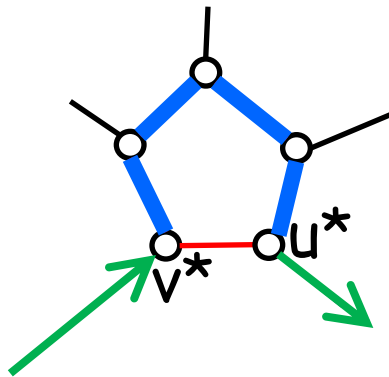


## Lemma

[  $C$ : Small cycle in  $F$

[ We have reached at  $v^*$  in  $V(C)$

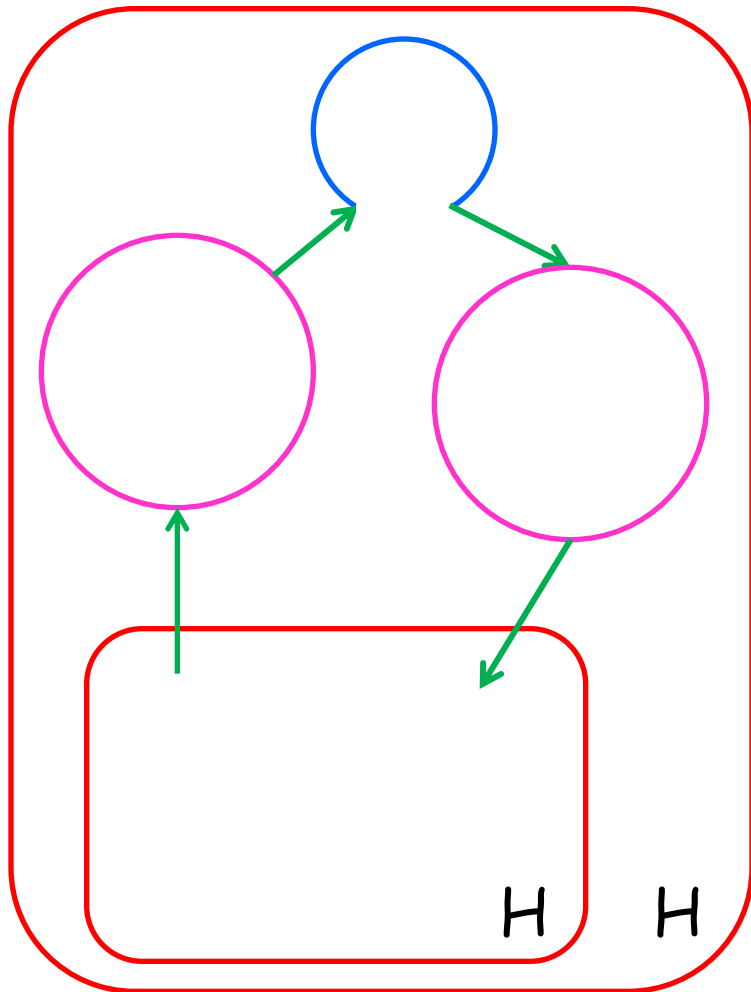
➔  $G[V(C)]$  has a **Hamilton path** from  $v^*$  to  $u^*$ , and we can leave for another cycle from  $u^*$



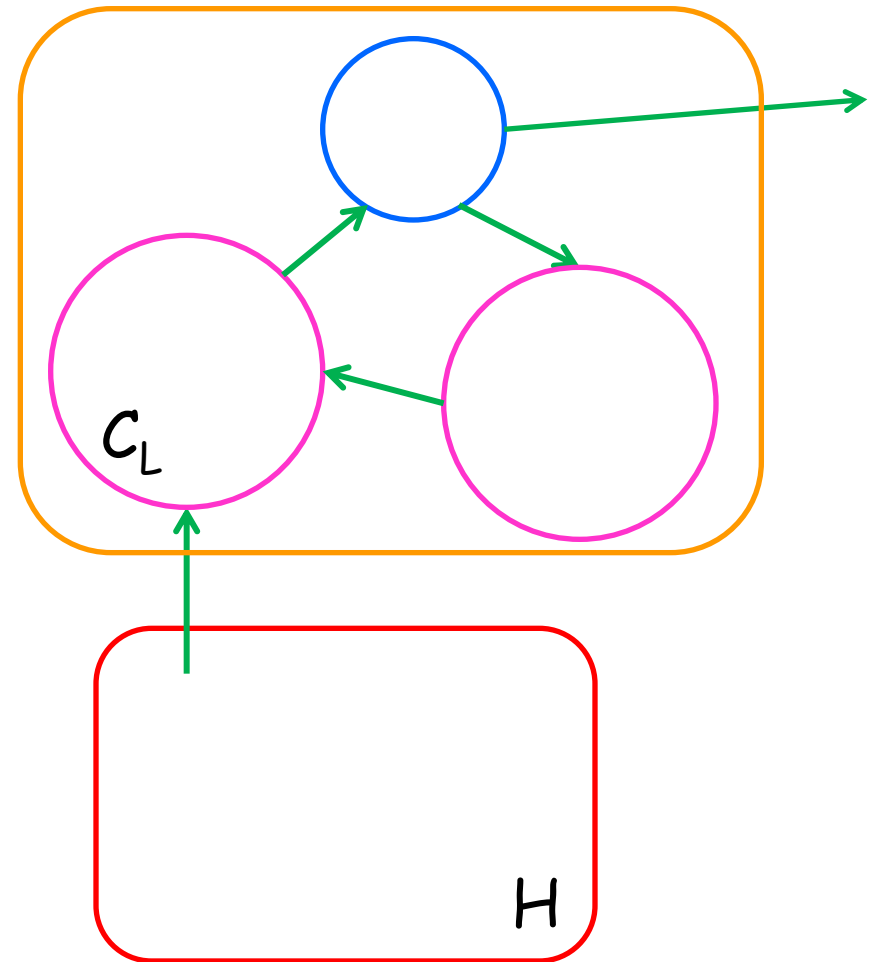


# Algorithm Sketch

➤ Back to  $H \rightarrow$  Update  $H$

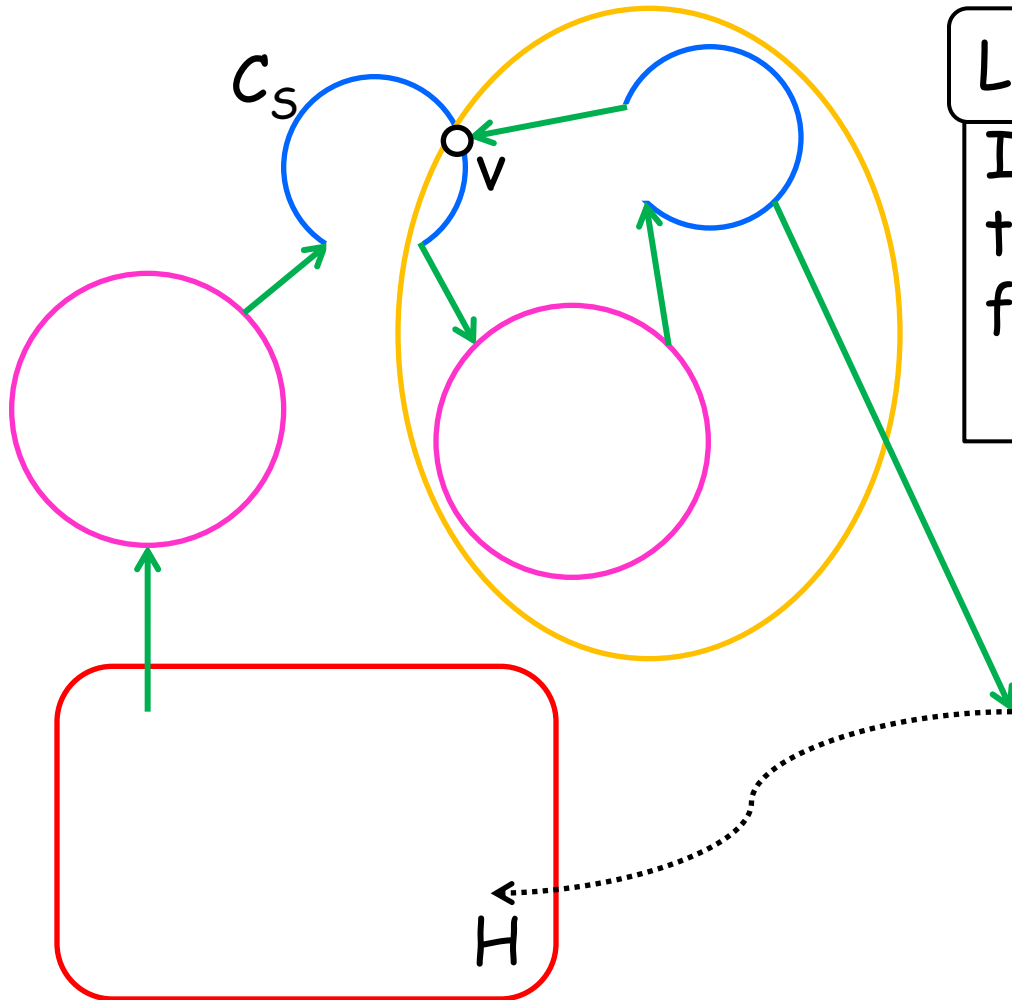


➤ Back to a large cycle  $C_L$   
➔ Compound  $C_L--C_L$

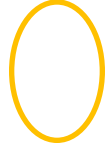


# Algorithm Sketch

- Back to a small cycle  $C_S$  (at  $v$  in  $V(C_S)$ )  $\rightarrow$  Compound  $v--v$



## Lemma

If  $G$  is 3-edge-connected, there exists an edge from  to another cycle.

# Approximation Ratio

Thm.

$$|E(H)| \leq 6n/5 - 1$$

(Pf.)  $x = \#$  **small cycles** in the initial 2-factor  $F$   
 $y = \#$  **large cycles** in the initial 2-factor  $F$

2 extra edges  
for each cycle

Save 1 edge for each small cycle

$$|E(H)| \leq n + 2(x + y - 1) - (x - 1)$$

$$= n + x + 2y - 1$$

$$\leq 6n/5 - 1$$

$$5x + 10y \leq n$$

# Contents

- Introduction

(2) An  $O(n^3)$ -algorithm for finding a 2-factor covering all 3-, 4-edge cuts in bridgeless cubic graphs

(3) A  $6/5$ -approx. algorithm for the minimum 2-edge-connected subgraph problem in 3-edge-connected cubic graphs

- Summary

# Summary

- For bridgeless cubic graphs:
  - A 2-factor covering all 3- and 4-edge cuts: Algorithm
  - A min-weight 2-factor covering all 3-edge cuts:
    - ✓ Algorithm
    - ✓ Polyhedral description

- For 3-edge-connected cubic graphs
  - 6/5-approx. algorithm for the min. 2-edge-connected subgraph problem

## Open Problems

*Min-weight* 2-factor covering all 3- and 4-edge cuts in bridgeless cubic graphs

6/5-approx. algorithm for the min. 2-edge-connected subgraph problem in *bridgeless cubic graphs*



# 2-Factors Covering 3-Edge Cuts [Weighted]

2-factor polytope [Edmonds 1965]

$$x(\delta(v)) = 2 \quad v \text{ in } V$$

$$x(Y) - x(\delta(S) - Y) \leq |Y| - 1 \quad \begin{array}{l} S \subset V, \\ Y \subseteq \delta(S), Y: \text{ matching, } |Y|: \text{ odd} \end{array}$$

$$0 \leq x(e) \leq 1 \quad e \text{ in } E$$

Additional constraint

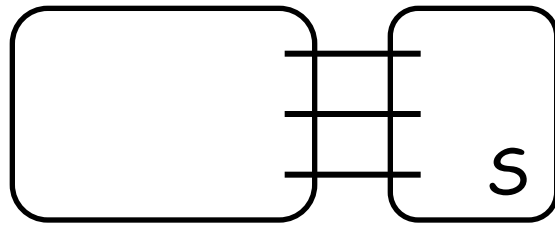
$$x(\delta(S)) = 2 \quad S \subset V, \delta(S) \text{ is a 3-edge cut}$$

Thm.

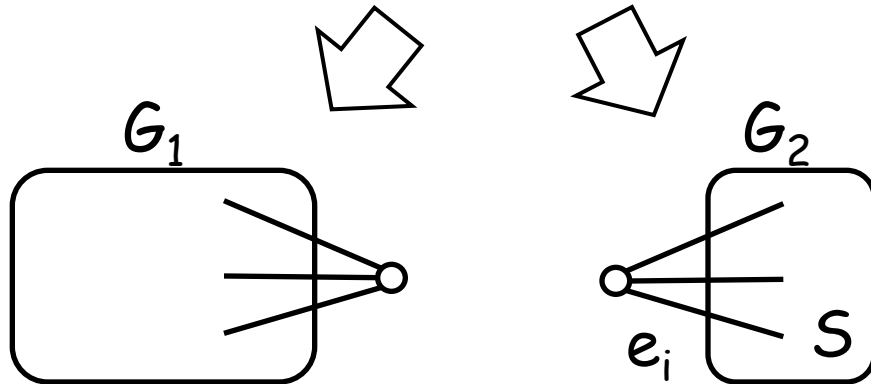
The above constraints determine the polytope of the 2-factors covering all 3-edge cuts

# Algorithm Sketch Gluing technique in [Cornuéjols, Naddef, Pulleyblank 85]

(1) Find a **proper** 3-edge cut  $\delta(S) = \{e_1, e_2, e_3\}$  ( $S$ : minimal)



(2) Contract  $V - S, S$



(3) In  $G_2$ , find a min. weight 2-factor  $F_i$  excluding  $e_i$  ( $i=1,2,3$ )

➤  $L_i = w(F_i \cap E[S])$

(3) In  $G_1$ , add extra weight  $x_i$  for  $e_i$ , where  
 $x_1 + x_2 = L_3, x_2 + x_3 = L_1, x_3 + x_1 = L_2$

(4) Recurse in  $G_1$