Isomorph-free generation of 2-connected graphs with applications

Derrick Stolee
University of Nebraska–Lincoln
s-dstolee1@math.unl.edu

March 19, 2011

Computer Search

Computers are extremely useful to graph theorists:

Computer Search

Computers are extremely useful to graph theorists:

- Find examples/counterexamples.

Computer Search

Computers are extremely useful to graph theorists:

- Find examples/counterexamples.
- Verify conjectures (on small examples).

Computer Search

Computers are extremely useful to graph theorists:

- Find examples/counterexamples.
- Verify conjectures (on small examples).
- Generate theorems.

2-connected graphs

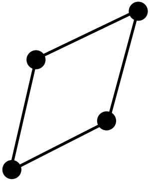
A graph is *2-connected* if there are no cut vertices.

2-connected graphs

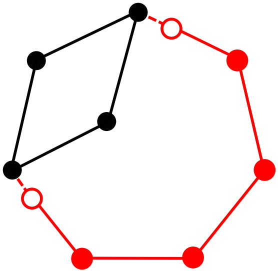
A graph is *2-connected* if there are no cut vertices.

2-connected graphs are exactly the graphs with ear decompositions.

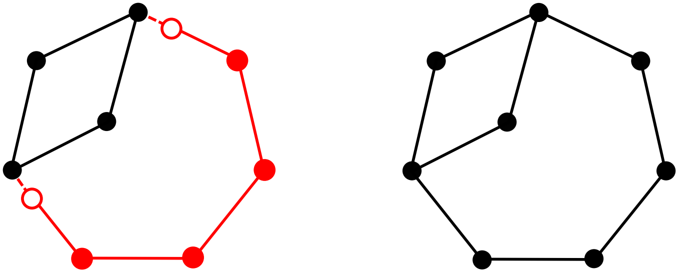
2-connected graphs and ear augmentations



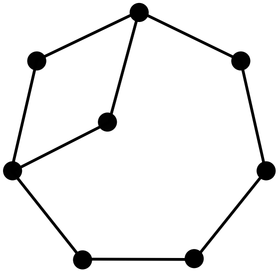
2-connected graphs and ear augmentations



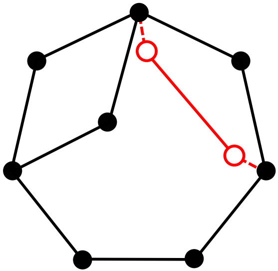
2-connected graphs and ear augmentations



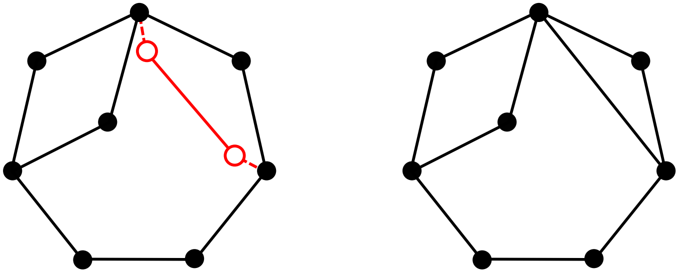
2-connected graphs and ear augmentations



2-connected graphs and ear augmentations



2-connected graphs and ear augmentations



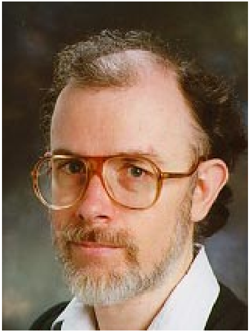
Generating by Ear Augmentations

Starting at each cycle, adding all possible ear augmentations will generate all 2-connected graphs.

Generating by Ear Augmentations

Starting at each cycle, adding all possible ear augmentations will generate all 2-connected graphs.

LOTS of redundancy!



Brendan McKay

"Isomorph-free Exhaustive Generation"

Isomorph-free Exhaustive Generation

The goal:

Generate all graphs of a given type with each isomorphism class represented exactly once.

Isomorph-free Exhaustive Generation

The goal:

Generate all graphs of a given type with each isomorphism class represented exactly once.

The recipe:

Isomorph-free Exhaustive Generation

The goal:

Generate all graphs of a given type with each isomorphism class represented exactly once.

The recipe:

1. An augmentation (vertex, edge, leaf, **ear**).

Isomorph-free Exhaustive Generation

The goal:

Generate all graphs of a given type with each isomorphism class represented exactly once.

The recipe:

1. An augmentation (vertex, edge, leaf, **ear**).
2. A canonical deletion.

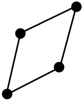
Isomorph-free Exhaustive Generation

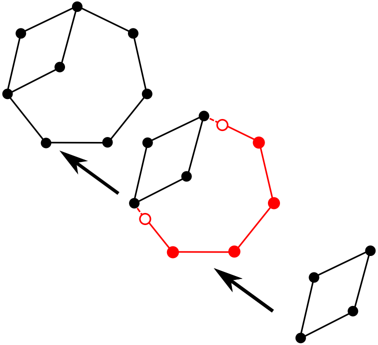
The goal:

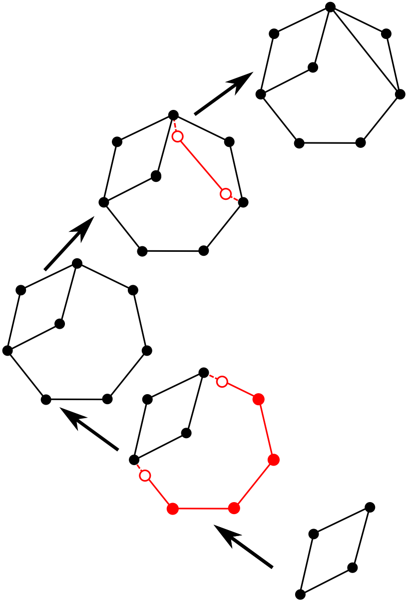
Generate all graphs of a given type with each isomorphism class represented exactly once.

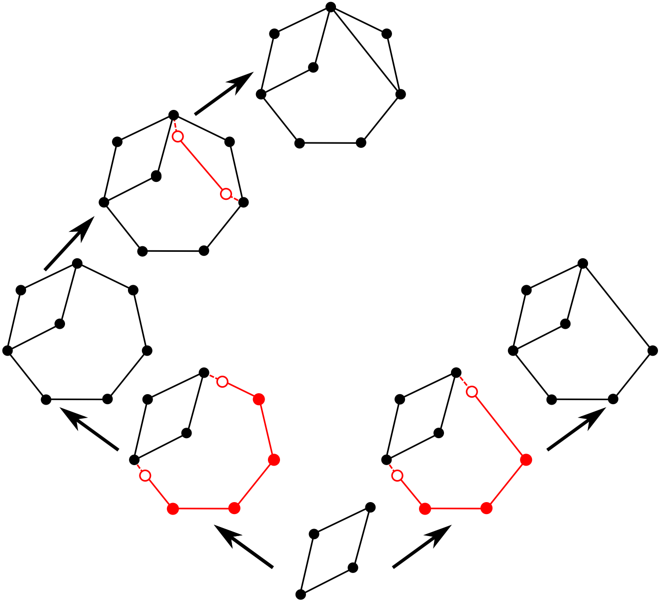
The recipe:

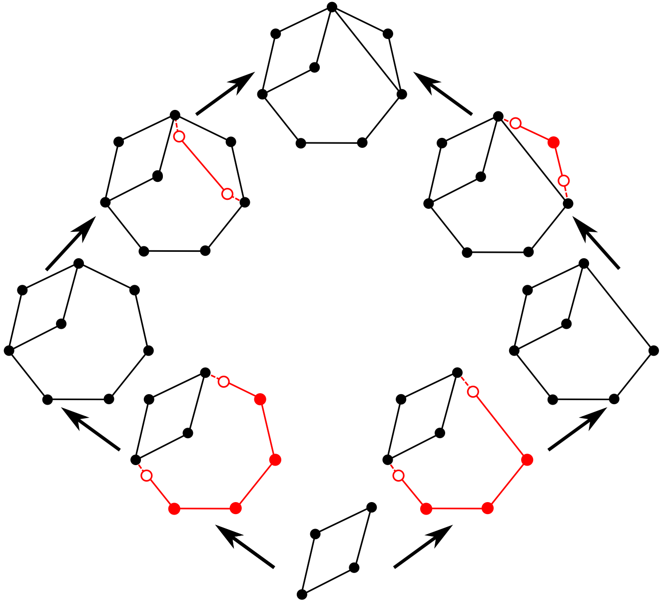
1. An augmentation (vertex, edge, leaf, **ear**).
2. A canonical deletion.
3. A pruning procedure.

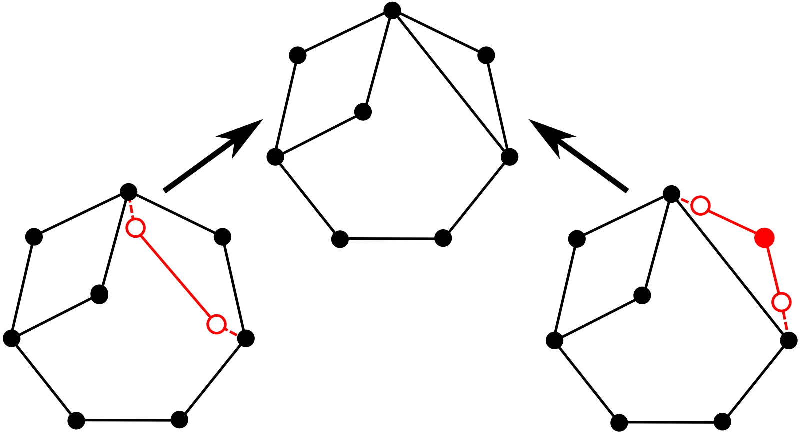


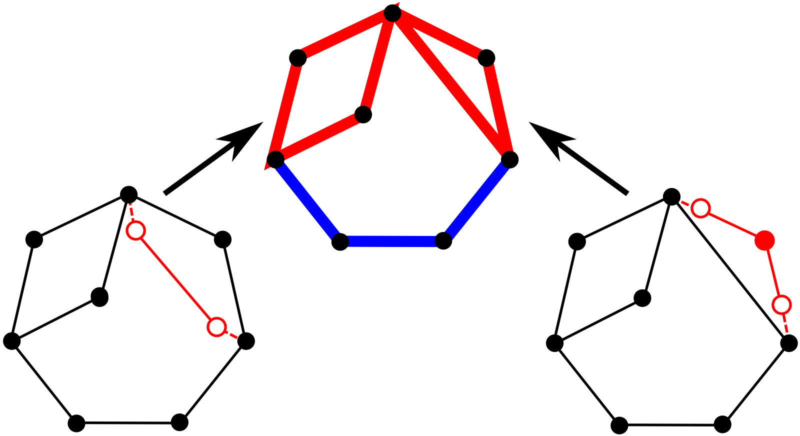


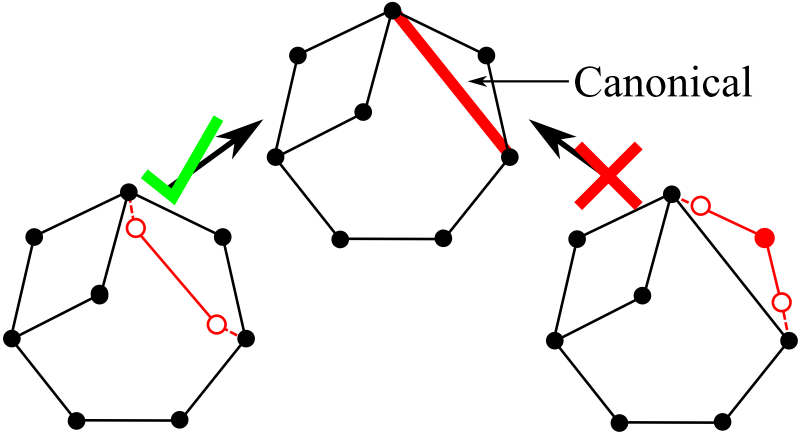


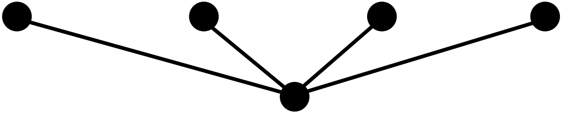


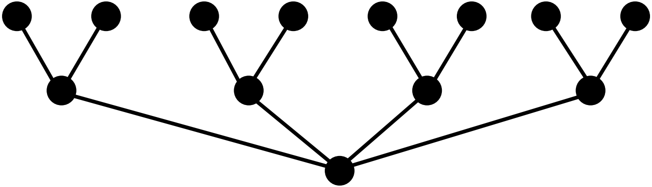


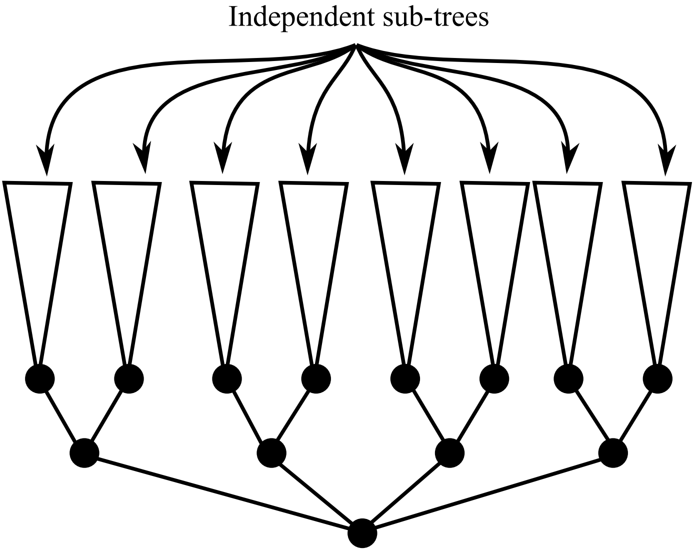





Canonical






Independent sub-trees

Implementation

Implemented in the TreeSearch library for parallelization in the Condor scheduler.

Executed on the Open Science Grid, a collection of supercomputers around the country.

Open Science Grid

Generating 2-connected Graphs

N	C_N	CPU time
5	10	0.01s
6	56	0.11s
7	468	0.26s
8	7123	10.15s
9	194066	5m 17.27s
10	9743542	7h 39m 28.47s
11	900969091	71d 22h 22m 49.12s

Generating 2-connected Graphs

N	C_N	CPU time
5	10	0.01s
6	56	0.11s
7	468	0.26s
8	7123	10.15s
9	194066	5m 17.27s
10	9743542	7h 39m 28.47s
11	900969091	71d 22h 22m 49.12s

Slower than vertex-augmentations, faster than edge-augmentations.

Three Applications

Three Applications

1. Uniquely K_r-Saturated Graphs
    1. Strength: ear-monotone constraints and sparse family.

Three Applications

1. Uniquely K_r-Saturated Graphs
   1. Strength: ear-monotone constraints and sparse family.
2. Edge Reconstruction Conjecture
   1. Strength: sparse family *and* structure of search.

Three Applications

1. Uniquely K_r-Saturated Graphs
    1. Strength: ear-monotone constraints and sparse family.
2. Edge Reconstruction Conjecture
    1. Strength: sparse family *and* structure of search.
3. p-Extremal Graphs
    1. Sparse family.
    2. Ear-monotone constraints.
    3. Structure of search.

Application 3: p-Extremal Graphs

A *perfect matching* (or *1-factor*) is a set of edges which cover each vertex exactly once.

Application 3: p-Extremal Graphs

A *perfect matching* (or *1-factor*) is a set of edges which cover each vertex exactly once.

Let $\Phi(G)$ denote the number of perfect matchings in a graph G.

Application 3: p-Extremal Graphs

Definition

Let n and p be integers. $f(n,p)$ is the maximum number of edges in a graph with n vertices and exactly p perfect matchings.

Application 3: p-Extremal Graphs

Definition

Let n and p be integers. $f(n, p)$ is the maximum number of edges in a graph with n vertices and exactly p perfect matchings.

Definition

A graph on n vertices is p-extremal if it has p perfect matchings and $f(n, p)$ edges.

Application 3: p-Extremal Graphs

Definition

Let n and p be integers. $f(n, p)$ is the maximum number of edges in a graph with n vertices and exactly p perfect matchings.

Definition

A graph on n vertices is p-extremal if it has p perfect matchings and $f(n, p)$ edges.

Question

How does $f(n, p)$ behave, and which graphs are p-extremal?

Andrzej Dudek John Schmitt

“On the Size and Structure of Graphs with a Constant Number of 1-Factors”

Dudek & Schmitt

Dudek & Schmitt

1. If G has p perfect matchings, n_0 vertices and $\frac{n_0^2}{4} + c$ edges, then for all $n \geq n_0$, $f(n,p) \geq \frac{n^2}{4} + c$.

Dudek & Schmitt

1. If G has p perfect matchings, n_0 vertices and $\frac{n_0^2}{4} + c$ edges, then for all $n \geq n_0$, $f(n, p) \geq \frac{n^2}{4} + c$.

Definition

The *excess* of a graph is the value $c(G) = |E(G)| - \frac{n(G)^2}{4}$.

Dudek & Schmitt

1. If G has p perfect matchings, n_0 vertices and $\frac{n_0^2}{4} + c$ edges, then for all $n \geq n_0$, $f(n,p) \geq \frac{n^2}{4} + c$.

Definition

The *excess* of a graph is the value $c(G) = |E(G)| - \frac{n(G)^2}{4}$.

2. For each p, there exist constants n_p, c_p so that for all even $n \geq n_p$,

$$f(n,p) = \frac{n^2}{4} + c_p.$$

Dudek & Schmitt

1. If G has p perfect matchings, n_0 vertices and $\frac{n_0^2}{4} + c$ edges, then for all $n \geq n_0$, $f(n,p) \geq \frac{n^2}{4} + c$.

Definition

The *excess* of a graph is the value $c(G) = |E(G)| - \frac{n(G)^2}{4}$.

2. For each p, there exist constants n_p, c_p so that for all even $n \geq n_p$,

$$f(n,p) = \frac{n^2}{4} + c_p.$$

3. Computed c_p for $p \in \{1, \dots, 6\}$.

Dudek & Schmitt

1. If G has p perfect matchings, n_0 vertices and $\frac{n_0^2}{4} + c$ edges, then for all $n \geq n_0$, $f(n, p) \geq \frac{n^2}{4} + c$.

Definition

The *excess* of a graph is the value $c(G) = |E(G)| - \frac{n(G)^2}{4}$.

2. For each p, there exist constants n_p, c_p so that for all even $n \geq n_p$,

$$f(n, p) = \frac{n^2}{4} + c_p.$$

3. Computed c_p for $p \in \{1, \ldots, 6\}$.
4. Found structure for p-extremal graphs with $p \in \{1, 2, 3\}$.

Stephen G. Hartke

Derrick Stolee

Douglas B. West

Matthew Yancey

“On extremal graphs with a given number of perfect matchings”

Hartke, Stolee, West, & Yancey

Hartke, Stolee, West, & Yancey

1. $c_p \geq 1$ for all $p \geq 2$.

Hartke, Stolee, West, & Yancey

1. $c_p \geq 1$ for all $p \geq 2$.
2. Bounded $n_p = O(\sqrt{p})$.

Hartke, Stolee, West, & Yancey

1. $c_p \geq 1$ for all $p \geq 2$.
2. Bounded $n_p = O(\sqrt{p})$.
3. Used naive search to find c_p and structure of p-extremal graphs for $p \leq 10$.

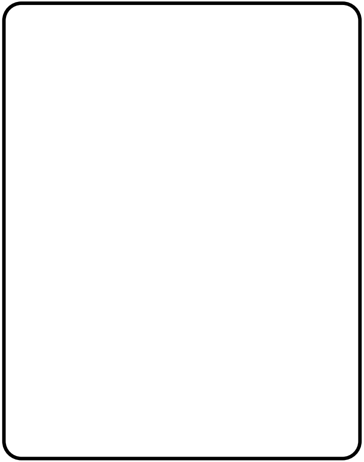
Hartke, Stolee, West, & Yancey

1. $c_p \geq 1$ for all $p \geq 2$.
2. Bounded $n_p = O(\sqrt{p})$.
3. Used naive search to find c_p and structure of p-extremal graphs for $p \leq 10$.

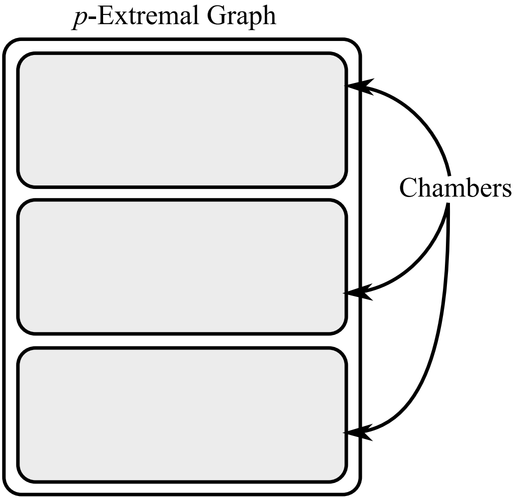
p	1	2	3	4	5	6	7	8	9	10
c_p	0	1	2	2	2	3	3	3	4	4
n_p	2	4	4	6	6	6	6	6	6	6
	[DS10]						[HSWY11+]			

The Structure of p-Extremal Graphs

p-Extremal Graph



The Structure of p-Extremal Graphs



The Structure of p-Extremal Graphs

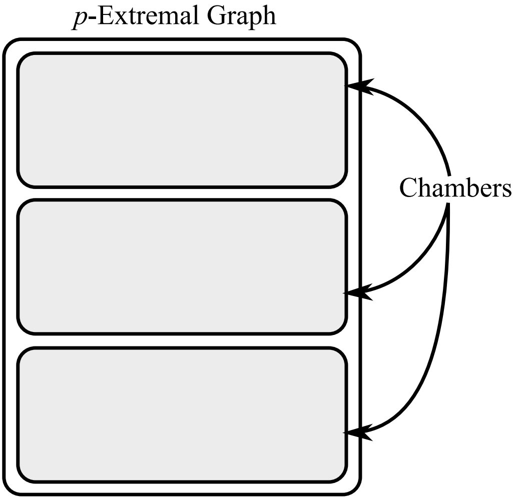
Chambers are the connected components in the subgraph of edges appearing in perfect matchings (*allowable* edges).

For G a graph with chambers $G_1, \dots, G_k$,

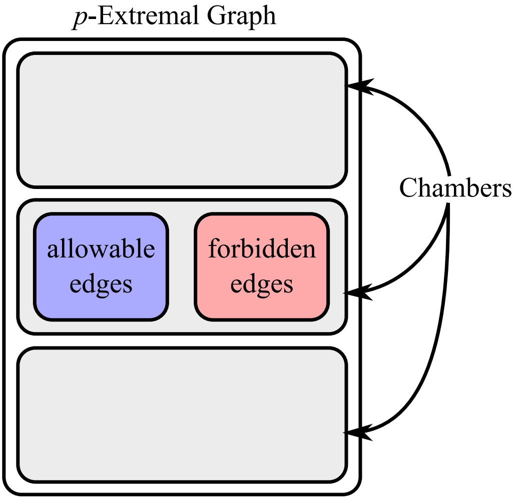
$$\Phi(G) = \prod_{i=1}^{k} \Phi(G_i).$$

$$c(G) \leq \sum_{i=1}^{k} c(G_i).$$

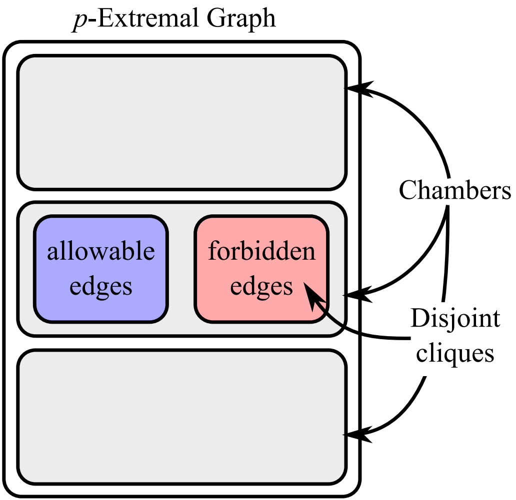
The Structure of p-Extremal Graphs



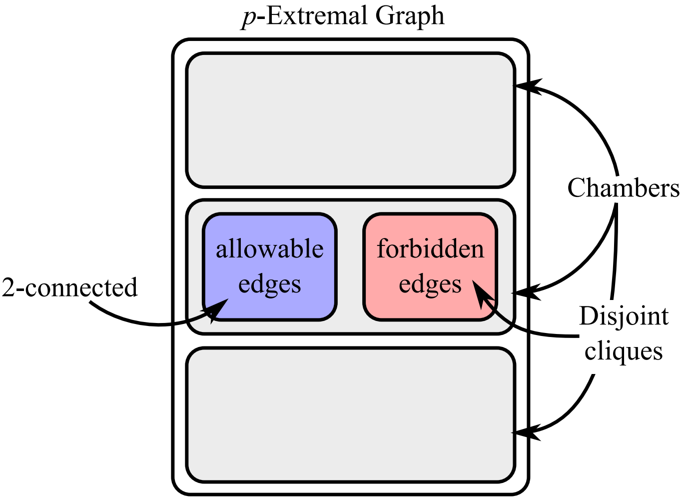
The Structure of p-Extremal Graphs



The Structure of *p*-Extremal Graphs



The Structure of *p*-Extremal Graphs



The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

Theorem (Lovász Two-Ears Theorem)

If H is a 1-extendable graph, there is a graded ear decomposition $H_0 \subset H_1 \subset H_2 \subset \cdots \subset H_k$ *So that*

The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

Theorem (Lovász Two-Ears Theorem)

If H is a 1-extendable graph, there is a graded ear decomposition $H_0 \subset H_1 \subset H_2 \subset \cdots \subset H_k$ So that

1. $H_0 \cong C_{2\ell}$ *for some* ℓ *and* $H_k = H$.

The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

Theorem (Lovász Two-Ears Theorem)

If H is a 1-extendable graph, there is a graded ear decomposition $H_0 \subset H_1 \subset H_2 \subset \cdots \subset H_k$ So that

1. $H_0 \cong C_{2\ell}$ *for some* ℓ *and* $H_k = H$.
2. *Each* H_i *is 1-extendable.*

The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

Theorem (Lovász Two-Ears Theorem)

If H is a 1-extendable graph, there is a graded ear decomposition $H_0 \subset H_1 \subset H_2 \subset \cdots \subset H_k$ So that

1. $H_0 \cong C_{2\ell}$ *for some* ℓ *and* $H_k = H$.
2. *Each* H_i *is 1-extendable.*
3. *Each ear augmentation* $H_i \subset H_{i+1}$ *uses one or two ears.*

The Structure of Allowable Edges

A connected graph with all edges allowable is *1-extendable*.

Theorem (Lovász Two-Ears Theorem)

If H is a 1-extendable graph, there is a graded ear decomposition $H_0 \subset H_1 \subset H_2 \subset \cdots \subset H_k$ So that

1. $H_0 \cong C_{2\ell}$ *for some* ℓ *and* $H_k = H$.
2. *Each* H_i *is 1-extendable.*
3. *Each ear augmentation* $H_i \subset H_{i+1}$ *uses one or two ears.*

Graphs which appear "between" two 1-extendable graphs in a two-ear augmentation are *almost 1-extendable* graphs.

The Search Space

Input: p, N, c.

The Search Space

Input: p, N, c.

Graphs: 1-extendable and almost 1-extendable graphs H with

The Search Space

Input: p, N, c.

Graphs: 1-extendable and almost 1-extendable graphs H with

- At most N vertices.

The Search Space

Input: p, N, c.

Graphs: 1-extendable and almost 1-extendable graphs H with

- At most N vertices.
- At most p perfect matchings.

The Search Space

Input: p, N, c.

Graphs: 1-extendable and almost 1-extendable graphs H with

- At most N vertices.
- At most p perfect matchings.

Solutions: Chambers G with p perfect matchings and $c(G) \geq c$.

Canonical Deletion

Find an ear in H to delete by priority:

Canonical Deletion

Find an ear in H to delete by priority:

Almost 1-extendable

↓

1-extendable

Unique such choice

Canonical Deletion

Find an ear in H to delete by priority:

Almost 1-extendable → 1-extendable	1-extendable → 1-extendable
Unique such choice	Minimum length ear
	Minimum canonical label of endpoints

Canonical Deletion

Find an ear in H to delete by priority:

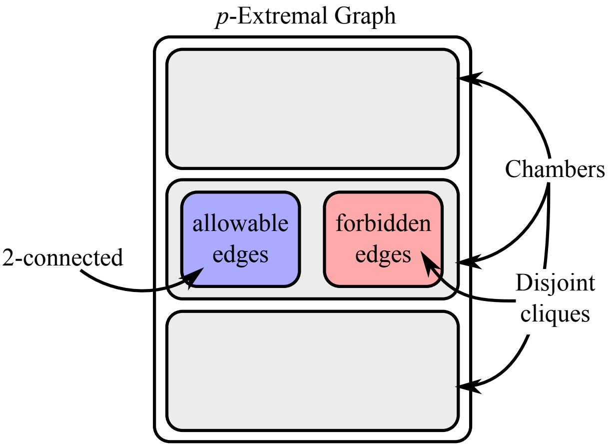
Almost 1-extendable	1-extendable	1-extendable
↓	↓	↓
1-extendable	1-extendable	Almost 1-extendable
Unique such choice	Minimum length ear	Minimum length ear
	Minimum canonical label of endpoints	Minimum canonical label of endpoints

Finding Solutions

Question

How do we transition from 1-extendable graphs to extremal chambers?

Finding Solutions



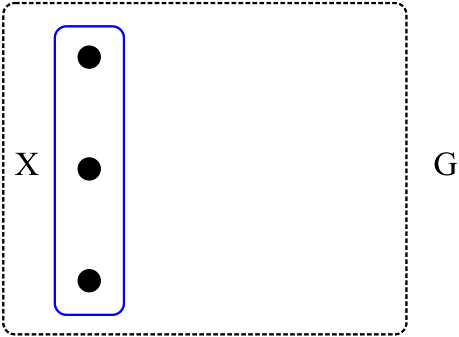
Finding Solutions

Use *barriers*:

G

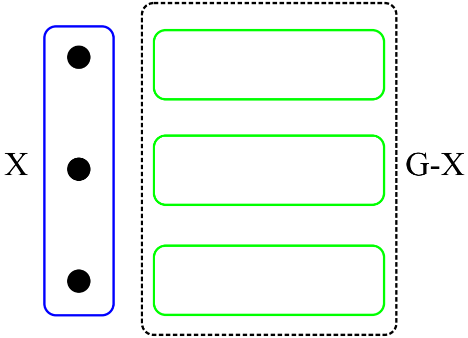
Finding Solutions

Use *barriers*:



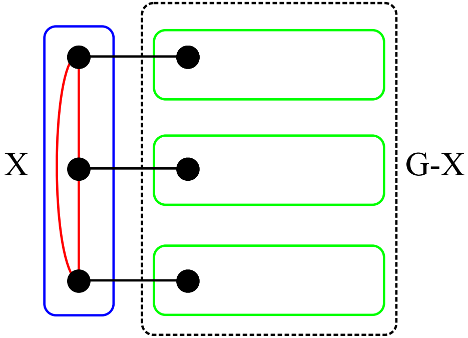
Finding Solutions

Use *barriers*:



Finding Solutions

Use *barriers*:



Finding Solutions

Definition

A *barrier* in a graph G with $\Phi(G) > 0$ is a set $X \subset V(G)$ so that $c_o(G - X) = |X|$.

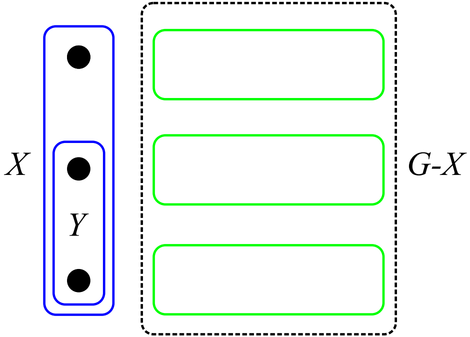
In a p-extremal chamber, every barrier is a clique of forbidden edges.

Conflicting Barriers

Two barriers X, Y *conflict* if:

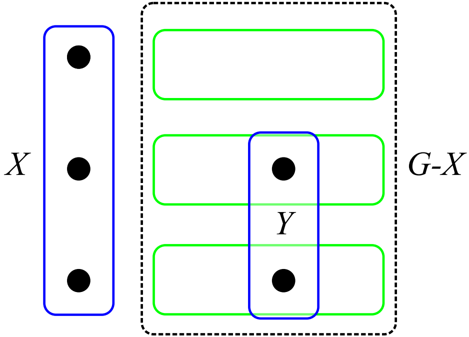
Conflicting Barriers

Two barriers X, Y *conflict* if:



Conflicting Barriers

Two barriers X, Y *conflict* if:



Finding Solutions

$$\{\text{Maximal chamber supergraphs of } H\}$$

$$\Longleftrightarrow$$

$$\{\text{Maximal sets of non-conflicting barriers in } H\}$$

Pruning

We want to prune when the excess can never reach c, no matter what augmentations we use.

Pruning

We want to prune when the excess can never reach c, no matter what augmentations we use.

$$
\begin{array}{lccc}
\textbf{1-extendable:} & H & \xrightarrow{\text{ear-augmentations}} & H' \\
& \text{best filling} \downarrow & & \downarrow \text{best filling} \\
\textbf{Chambers:} & G & & G'
\end{array}
$$

Pruning

We want to prune when the excess can never reach c, no matter what augmentations we use.

$$\begin{array}{lccc} \textbf{1-extendable:} & H & \xrightarrow{\text{ear-augmentations}} & H' \\ & \Big\downarrow{\scriptstyle\text{best filling}} & & \Big\downarrow{\scriptstyle\text{best filling}} \\ \textbf{Chambers:} & G & & G' \end{array}$$

$$\begin{aligned} c(G') \leq c(G) & \\ & + 2(\Phi(H') - \Phi(H)) \\ & - \frac{1}{4}(n(H') - n(H))(n(H) - 2). \end{aligned}$$

Pruning and Optimizations

Let H be the current 1-extendable graph and G a best filling.

Pruning and Optimizations

Let H be the current 1-extendable graph and G a best filling.

1. If $c(G) + 2(p - \Phi(H)) < c$, then prune.

Pruning and Optimizations

Let H be the current 1-extendable graph and G a best filling.

1. If $c(G) + 2(p - \Phi(H)) < c$, then prune.
2. Let N be the maximum so that

$$c(G) + 2(p - \Phi(H)) - \frac{1}{4}(N - n(H))(n(H) - 2) \geq c.$$

   We do not need to augment beyond N vertices.

Pruning and Optimizations

Let H be the current 1-extendable graph and G a best filling.

1. If $c(G) + 2(p - \Phi(H)) < c$, then prune.
2. Let N be the maximum so that

$$c(G) + 2(p - \Phi(H)) - \frac{1}{4}(N - n(H))(n(H) - 2) \geq c.$$

   We do not need to augment beyond N vertices.
3. If adding an ear at endpoints x, y increases $\Phi(H)$ beyond p, never augment on that pair again.

Results

p	1	2	3	4	5	6	7	8	9	10
c_p	**0**	**1**	**2**	**2**	**2**	**3**	**3**	**3**	**4**	**4**
n_p	2	4	4	6	6	6	6	6	6	6

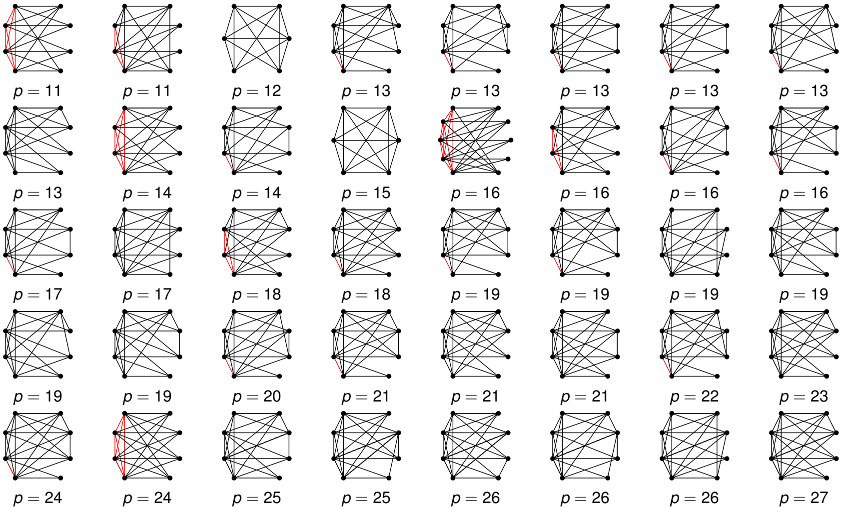
p	11	12	13	14	15	16	17	18	19	20
c_p	3	**5**	3	4	**6**	4	4	5	4	5
n_p	8	6	8	8	6	8	8	8	8	8

p	21	22	23	24	25	26	27			
c_p	5	5	5	**6**	5	5	**6**			
n_p	8	8	8	8	8	8	8			

Timing

p	N_p	c_p	Total CPU Time				
11	14	3					43.29s
12	14	5					44.01s
13	14	3				6m	39.80s
14	16	4				12m	10.40s
15	16	6				12m	42.72s
16	16	4			2h	07m	58.60s
17	16	4			6h	46m	07.72s
18	18	5			11h	45m	01.95s
19	18	4		2d	17h	12m	31.85s
20	18	5		4d	05h	28m	11.79s
21	18	5		13d	17h	29m	12.45s
22	20	5		42d	20h	40m	30.41s
23	20	5		118d	07h	38m	36.84s
24	20	6		209d	10h	09m	54.98s
25	20	5	2y	187d	21h	48m	46.31s
26	20	5	7y	75d	13h	55m	10.27s
27	22	6	10y	247d	21h	03m	13.94s

p-Extremal Chambers



Future Work

For p-extremal graphs:

1. Find a “strong” upper bound on c_p for an infinite family of values of p.
2. A start: prove the complete graphs on $2t$ vertices are p-extremal for $p = (2t-1)!!$.
3. A lower bound on c_p which grows in the limit.

Future Work

For the technique:

1. More applications!
2. More optimizations?

Future Work

For the tools:

1. Implement vertex/edge/leaf augmentations.
2. Apply to new problems.
3. Compare to current applications.

Isomorph-free generation of 2-connected graphs with applications

Derrick Stolee[1]
University of Nebraska–Lincoln
s-dstolee1@math.unl.edu

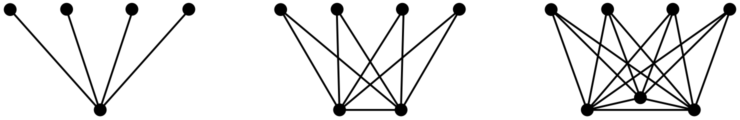
March 19, 2011

[1] Supported by NSF grants CCF-0916525 and DMS-0914815.

Application 1: Uniquely K_r-Saturated Graphs

Definition

A graph G is *uniquely K_r-saturated* if G contains no K_r and for every edge $e \in \overline{G}$ admits exactly one copy of K_r in $G + e$.



(a) 1-book (b) 2-book (c) 3-book

Figure: The $(r-2)$-books are uniquely K_r saturated.

Joshua Cooper

Paul Wenger

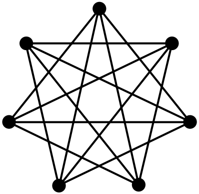
Two Conjectures:

1. For each r, there are a finite number of uniquely K_r-saturated graphs with no dominating vertex.
2. For each r, every uniquely K_r-saturated graph with no dominating vertex is regular.

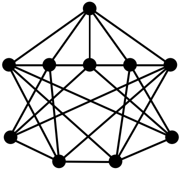
Previously verified to 9 vertices.

Application 1: Uniquely K_r-Saturated Graphs

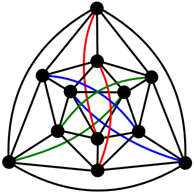
1. Uniquely K_r-saturated graphs have diameter 2 (and are 2-connected).
2. Strength: K_4-free is a sparse, monotone property.
3. Verified for $r = 4$ and $n \leq 12$.
4. Verified for $r \in \{5, 6\}$ and $n \leq 11$.



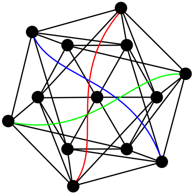
(a)



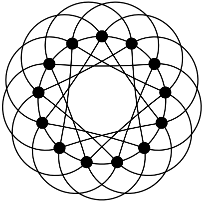
(b)



(c)



(d)



(e)

New! Joint with Stephen G. Hartke