(Approximate) Graph Products

Marc Hellmuth

Max Planck Institute for Mathematics in the Sciences
Leipzig, Germany

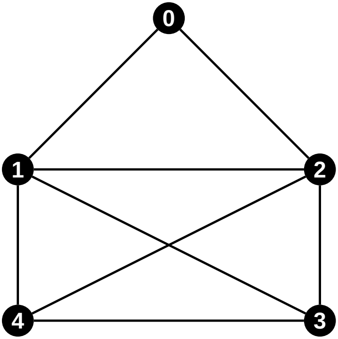
Bioinformatics Group
Science Department of Computer Science, and Interdisciplinary Center for Bioinformatics
University Leipzig, Germany

BLED - 2010

Basics

A graph is a pair $G = (V, E)$ with vertex set $V \neq \emptyset$ and edge set E.

here: simple, connected, undirected graphs

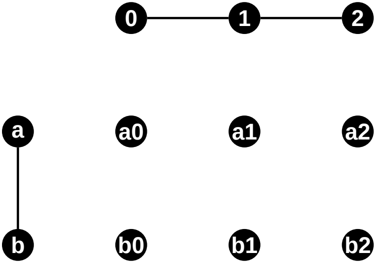


Strong and Cartesian Product

The vertex set of the Cartesian product ($\Box$) and strong product ($\boxtimes$) is defined as follows:

$$V(G_1 \Box G_2) = V(G_1 \boxtimes G_2) = \{(v_1, v_2) \mid v_1 \in V(G_1), v_2 \in V(G_2)\}$$

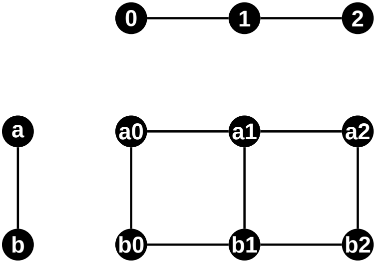
.



Cartesian Product

Two vertices (x_1,x_2), (y_1,y_2) are adjacent in $G_1\Box G_2$ if

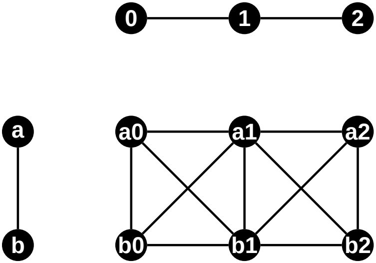
1. $(x_1,y_1)\in E(G_1)$ and $x_2=y_2$ or if
2. $(x_2,y_2)\in E(G_2)$ and $x_1=y_1$.



Strong Product

Two vertices (x_1, x_2), (y_1, y_2) are adjacent in $G_1 \boxtimes G_2$ if

1. $(x_1, y_1) \in E(G_1)$ and $x_2 = y_2$ or if
2. $(x_2, y_2) \in E(G_2)$ and $x_1 = y_1$ or if
3. $(x_1, y_1) \in E(G_1)$ and $(x_2, y_2) \in E(G_2)$.



Decomposition

Definition

G is prime, if $\nexists A * B = G$ with A, B nontrivial, i.e. $|V(A)|, |V(B)| > 1$.
$(* = \Box, \boxtimes)$

Aim: Prime factor decomposition (PFD) of given G.

Prime Factor Decomposition

Theorem (Sabidussi (1959))

PFD of every connected graph w.r.t. the Cartesian product is unique.

Theorem (Dörfler and Imrich (1969), McKenzie (1971))

PFD of every connected graph w.r.t. the strong product is unique.

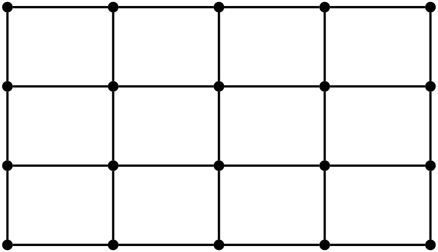
Theorem (Imrich and Peterin (2007))

PFD of every connected graph w.r.t. the Cartesian product can be computed in $O(|E(G)|)$ time.

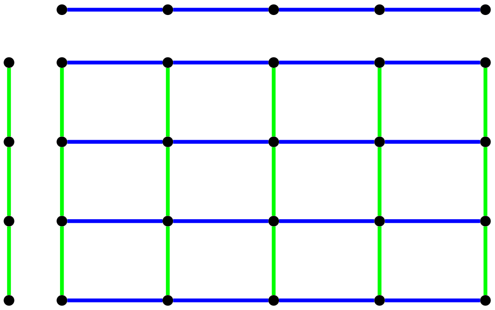
Theorem (Hammack and Imrich (2009))

PFD of every connected graph w.r.t. the strong product can be computed in $O(|E(G)|\Delta^2)$ time.

Decomposition of Cartesian Product



Decomposition of Cartesian Product



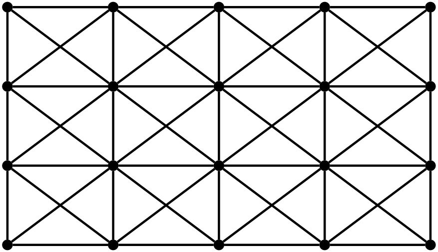
Copies of Factors in a Product are called layer or fiber.

MAIN IDEA: Decomposition strong product

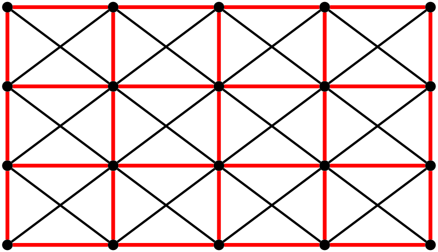
Find a spanning subgraph with special properties in G, the so called cartesian skeleton.

The decomposition of the cartesian skeleton w.r.t. cartesian product together with some additional operations leads to the possible factors of the strong product.

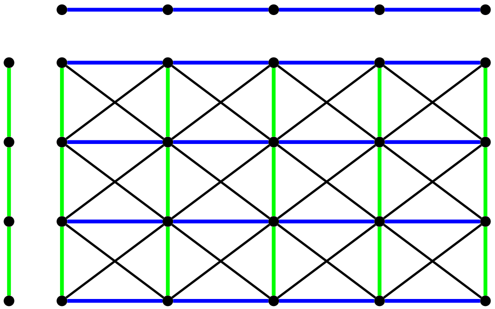
MAIN IDEA: Decomposition strong product



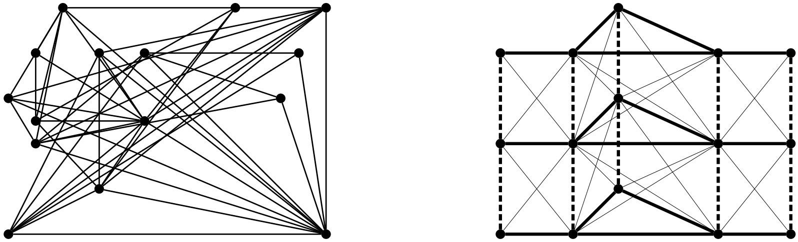
MAIN IDEA: Decomposition strong product



MAIN IDEA: Decomposition strong product



Motivation



Two isomorphic product graphs.

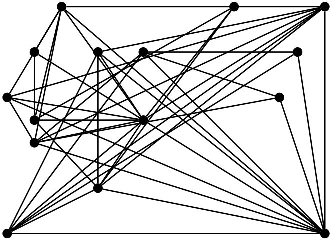
Motivation

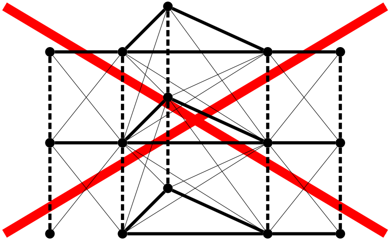
Problem:

Often real data, that is represented by graphs, is disturbed

and thus the corresponding "product graph" is disturbed.

Motivation





Problem:

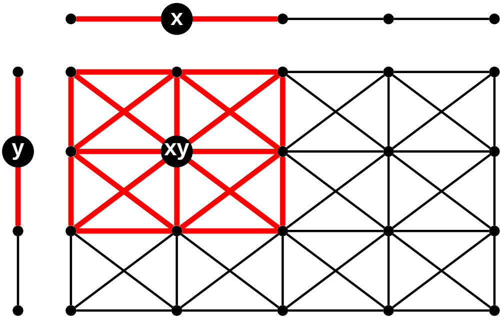
Often real data, that is represented by graphs, is disturbed and thus the corresponding "product graph" is disturbed.

- How can we recognize original factors of disturbed products?
- How can we recognize at least some parts of a disturbed product as a product?

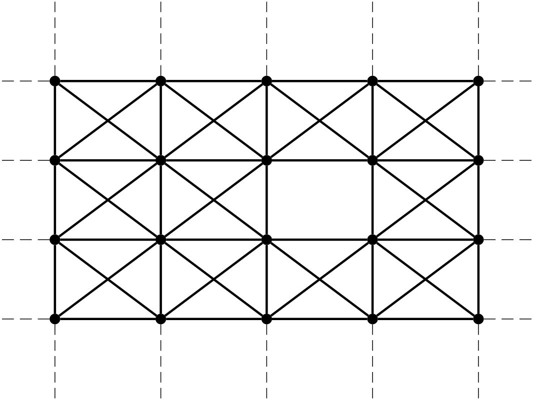
What if prime?

Aim: Get a product of graphs that is "near" a given prime graph (approximate products).

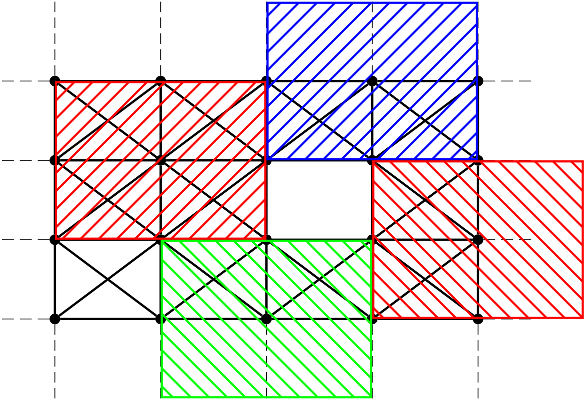
Remark: Induced neighborhoods in products are products.



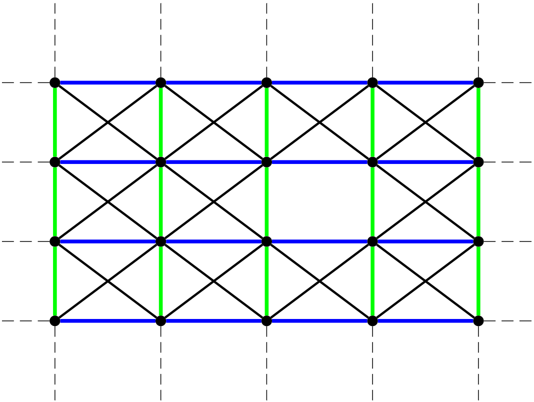
IDEA: Approximate Products



IDEA: Approximate Products



IDEA: Approximate Products



Tools

1. **`S=1-condition`**
2. Backbone $\mathbb{B}(G)$
3. Color-Continuation

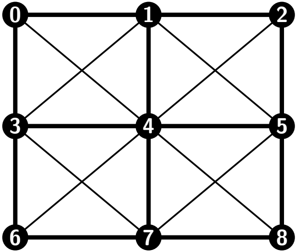
Thinness (**S=1-condition**)

Let G be a graph and $v,w \in V(G)$.

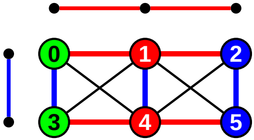
- v,w are in Relation S if N[v] = N[w]
- We call a graph S-thin if no two vertices v,w are in Relation S.

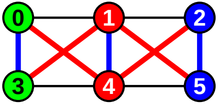
If G is S-thin the Cartesian edges are uniquely determined

Thinness (**S=1-condition**)

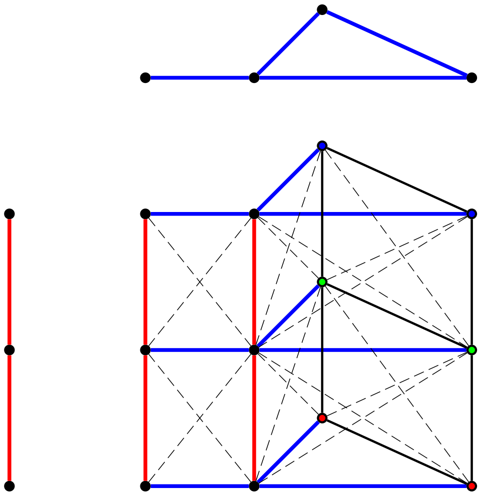


WHAT ARE THE FIBERS ?





Thinness (**S=1-condition**)

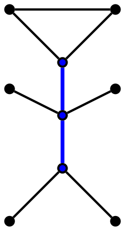


The Backbone $\mathbb{B}(G)$

$$\begin{aligned}\mathbb{B}(G) &:= \{v \in V(G) \mid |S_v(v)| = 1\} \\ &= \{v \in V(G) \mid N[v] \text{ is strictly maximal in } G\}\end{aligned}$$

Theorem

$\mathbb{B}(G)$ is a connected dominating set.



The Backbone $\mathbb{B}(G)$

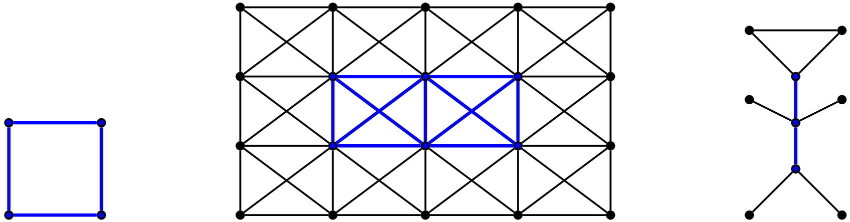


Figure: Examples

The Backbone $\mathbb{B}(G)$

For a local covering we consider neighborhoods of vertices of $\mathbb{B}(G)$ only.

Why?

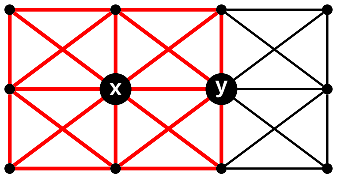
Theorem

*All Cartesian edges that satisfy the **`S=1-condition`** in an arbitrary induced neighborhood also satisfy the **`S=1-condition`** in the induced neighborhood of a vertex of the backbone $\mathbb{B}(G)$.*

Color-Continuation

Color-continuation from H_1 to H_2:

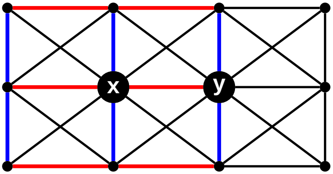
For all newly colored edges with color c in H_2
(**`S=1-condition`** Cartesian edges in H_2),
we have to find a representative edge that
satisfies the **`S=1-condition`** in H_1 and was already colored in H_1.



Color-Continuation

Color-continuation from H_1 to H_2:

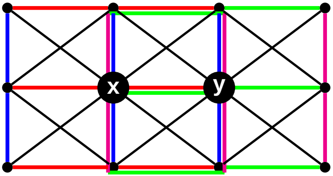
For all newly colored edges with color c in H_2
(**`S=1-condition`** Cartesian edges in H_2),
we have to find a representative edge that
satisfies the **`S=1-condition`** in H_1 and was already colored in H_1.



Color-Continuation

Color-continuation from H_1 to H_2:

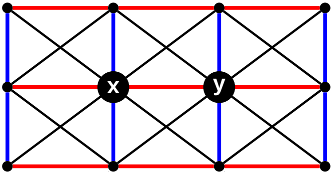
For all newly colored edges with color c in H_2
(**`S=1-condition`** Cartesian edges in H_2),
we have to find a representative edge that
satisfies the **`S=1-condition`** in H_1 and was already colored in H_1.



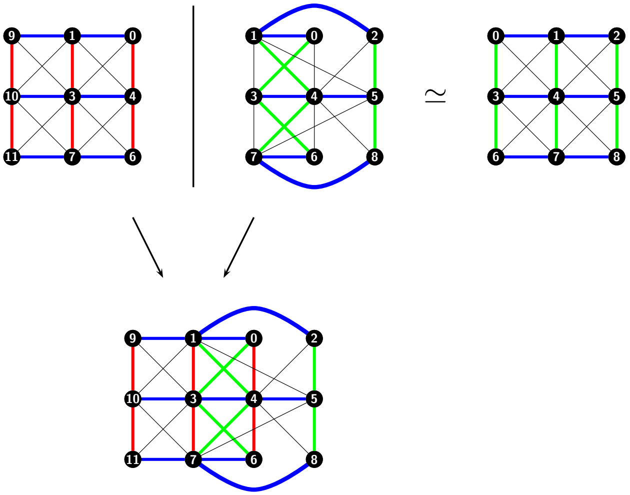
Color-Continuation

Color-continuation from H_1 to H_2:

For all newly colored edges with color c in H_2
(**`S=1-condition`** Cartesian edges in H_2),
we have to find a representative edge that
satisfies the **`S=1-condition`** in H_1 and was already colored in H_1.

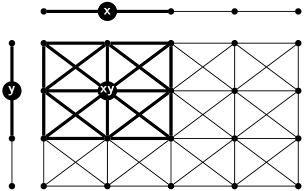


Example: Color-Continuation fails

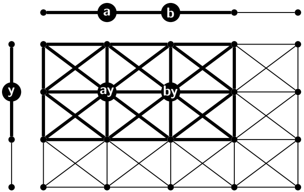


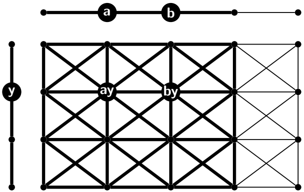
Local Approach

Used Subproducts



1-neighborhood $\langle N[(x,y)]\rangle = \langle N[x]\rangle \boxtimes \langle N[y]\rangle$

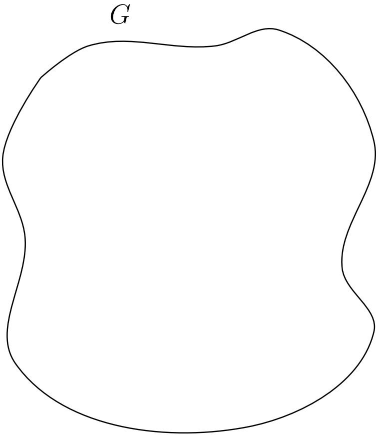




lhs.: The edge-neighborhood $\langle N[(a,y)] \cup N[(b,y)]\rangle$

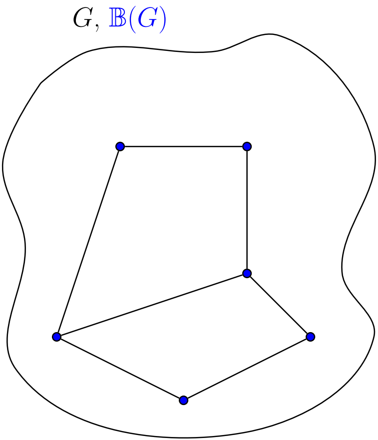
rhs.: The N^*-neighborhood $N^*_{(ay),(by)} = \langle \cup_{z \in N[(ay)] \cap N[by]} N[z]\rangle$

Local Approach



INPUT: thin graph G;

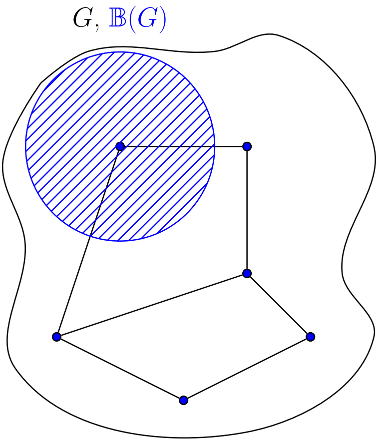
Local Approach



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Local Approach

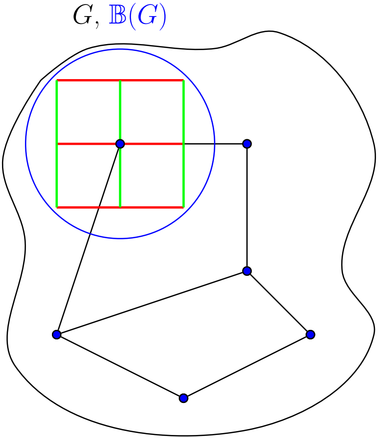


INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Local Approach

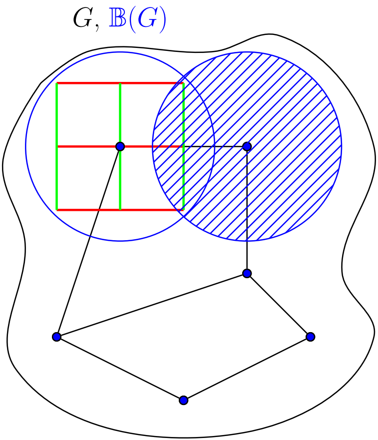


INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Local Approach



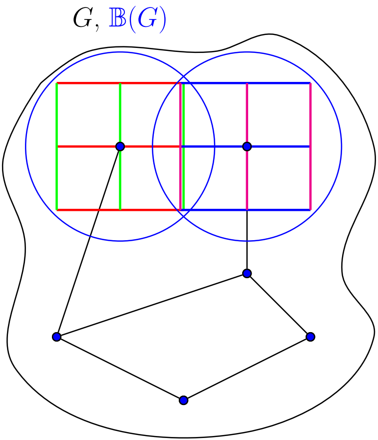
INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$;

Local Approach



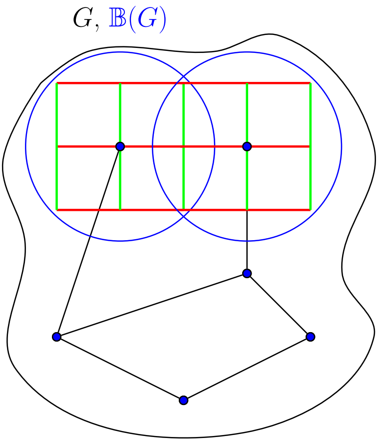
INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

Local Approach



INPUT: thin graph G;

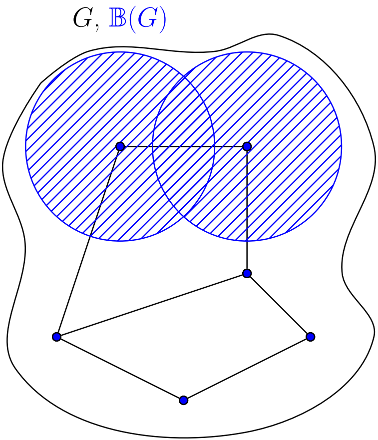
Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x]\rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y]\rangle$);

IF (color-conti works OR
  $\langle N[x]\rangle$ and $\langle N[y]\rangle$ are thin) **THEN** $\surd$;

Local Approach



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

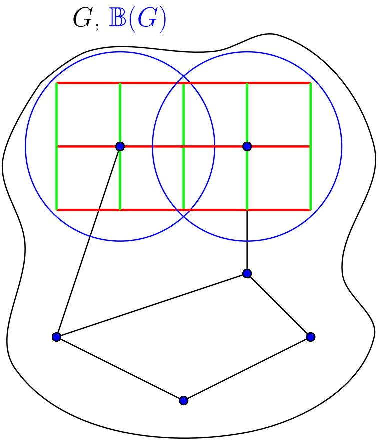
Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

IF (color-conti works OR
$\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin) **THEN** $\surd$;

ELSE IF ((x, y) is Cartesian) **THEN**
PFD($\langle N[x] \cup N[y] \rangle$);

Local Approach



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

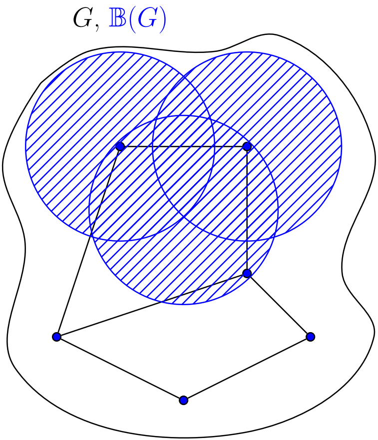
Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

IF (color-conti works OR
  $\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin) **THEN** $\surd$;

ELSE IF ((x, y) is Cartesian) **THEN**
  PFD($\langle N[x] \cup N[y] \rangle$);

Local Approach



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

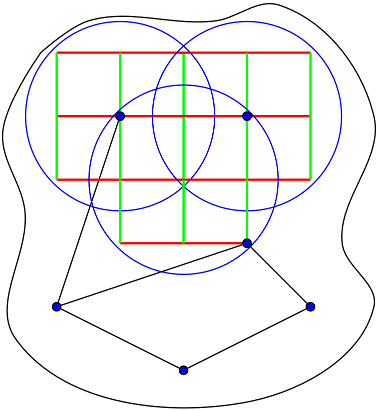
IF (color-conti works OR
$\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin) **THEN** $\surd$;

ELSE IF ((x, y) is Cartesian) **THEN**

PFD($\langle N[x] \cup N[y] \rangle$);

ELSE PFD($\langle N^*_{x,y} \rangle$);

Local Approach



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

IF (color-conti works OR
$\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin) **THEN** $\surd$;

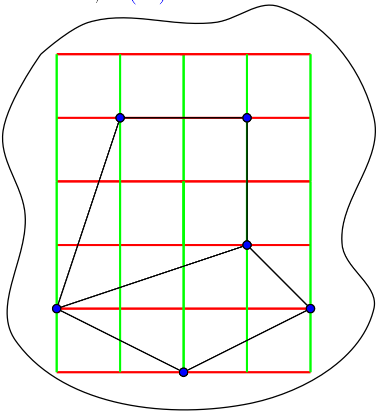
ELSE IF ((x, y) is Cartesian) **THEN**

PFD($\langle N[x] \cup N[y] \rangle$);

ELSE PFD($\langle N^*_{x,y} \rangle$);

Local Approach

G, $\mathbb{B}(G)$



INPUT: thin graph G;

Compute $\mathbb{B}_{BFS}(G)$;

Take first $x \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[x] \rangle$);

Take next $y \in \mathbb{B}_{BFS}(G)$; PFD($\langle N[y] \rangle$);

IF (color-conti works OR
$\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin) **THEN** $\surd$;

ELSE IF ((x, y) is Cartesian) **THEN**

PFD($\langle N[x] \cup N[y] \rangle$);

ELSE PFD($\langle N^*_{x,y} \rangle$);

....

OUTPUT: Product-colored graph G and
Primefactors of G;

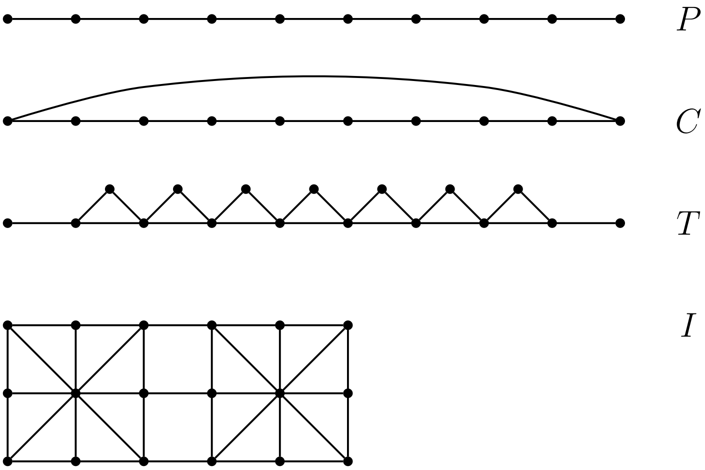
Local Approach

Theorem

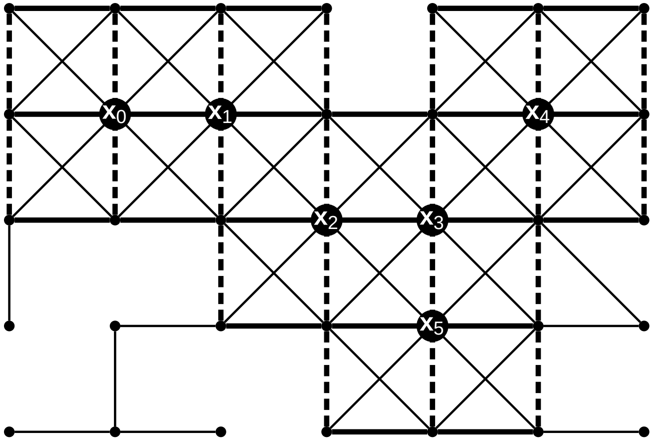
The Local Approach determines the prime factors and the corresponding product coloring of a given graph $G = (V, E)$ with bounded maximum degree in $O(|V|\Delta^6)$ time.

Approximate Products - Results

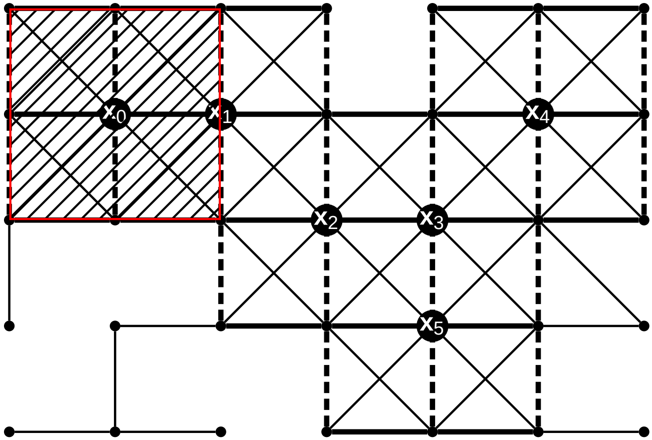
Test DataSet



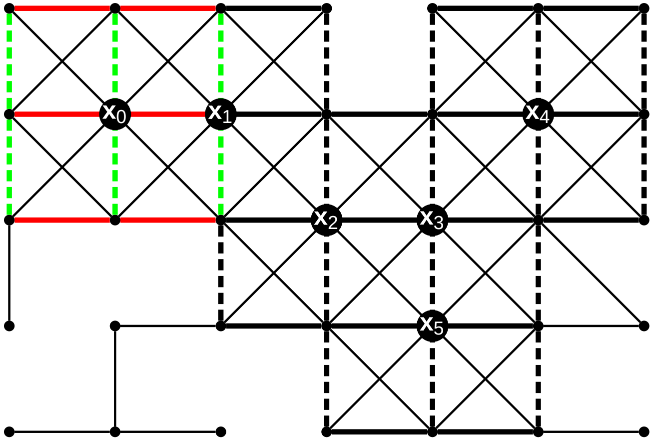
Approximate Products - Approach



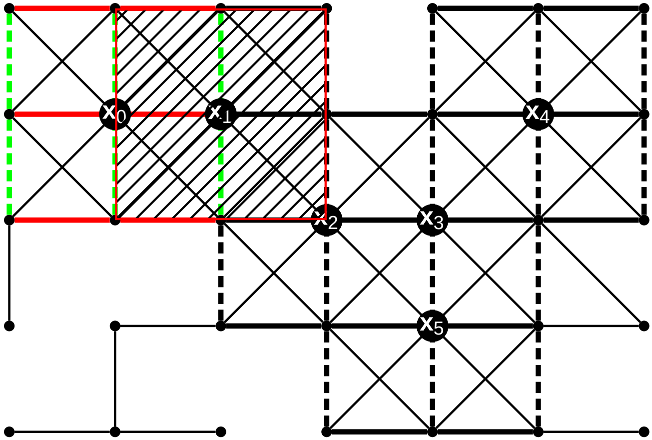
Approximate Products - Approach



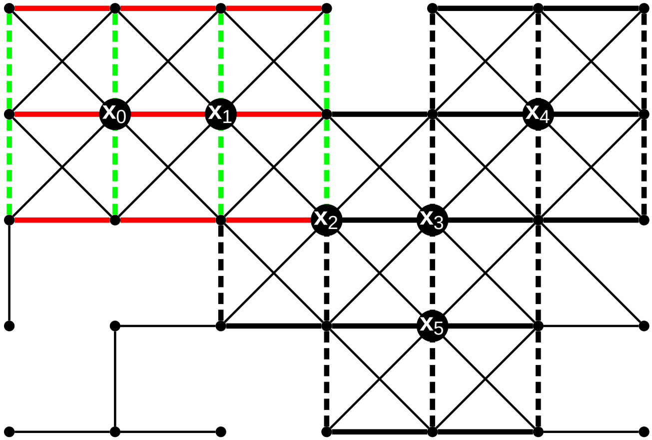
Approximate Products - Approach



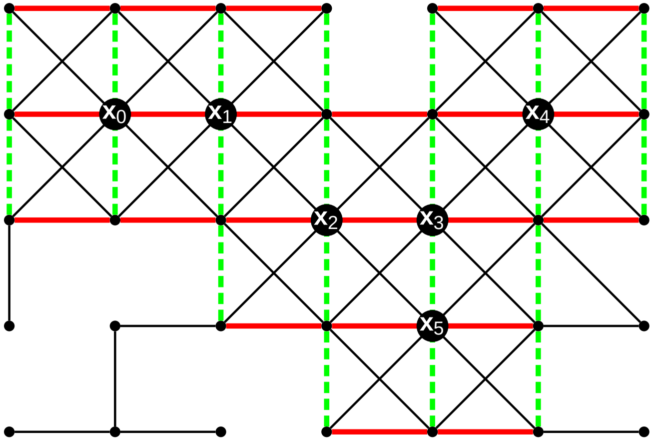
Approximate Products - Approach



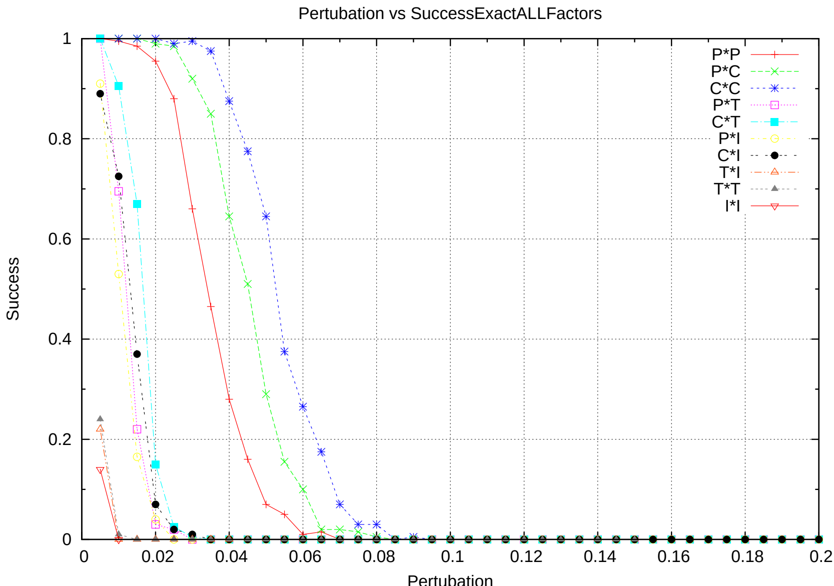
Approximate Products - Approach

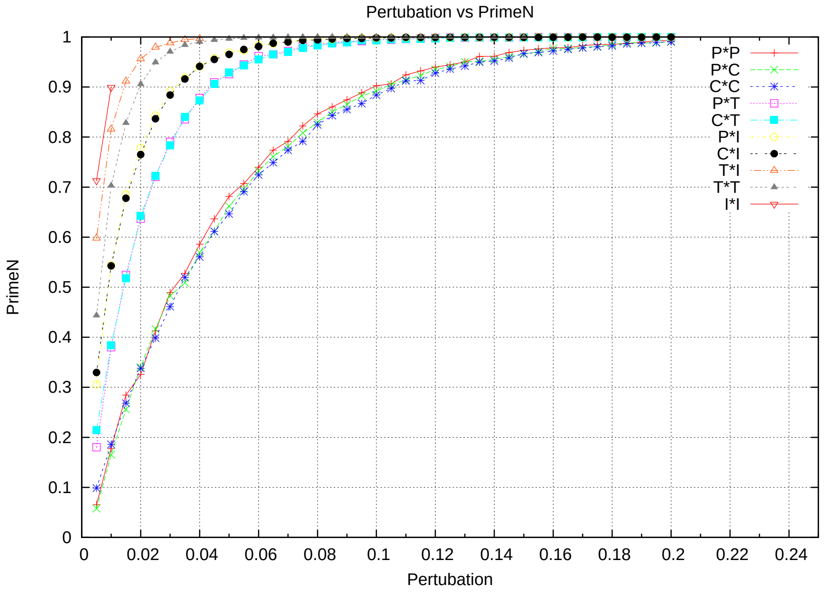


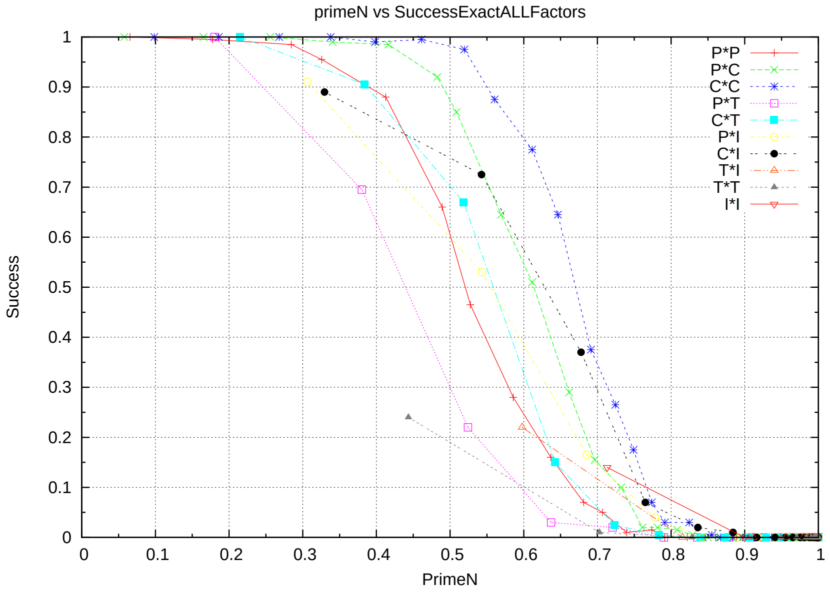
Approximate Products - Approach



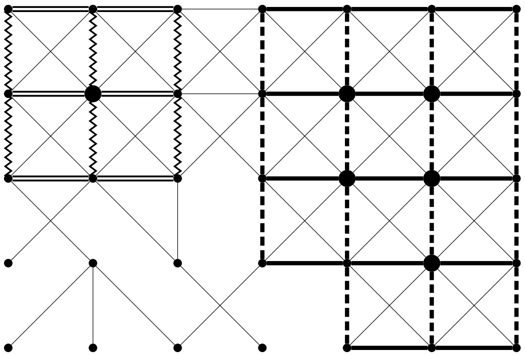
How much perturbation is allowed s.t. we could recover both original factors?



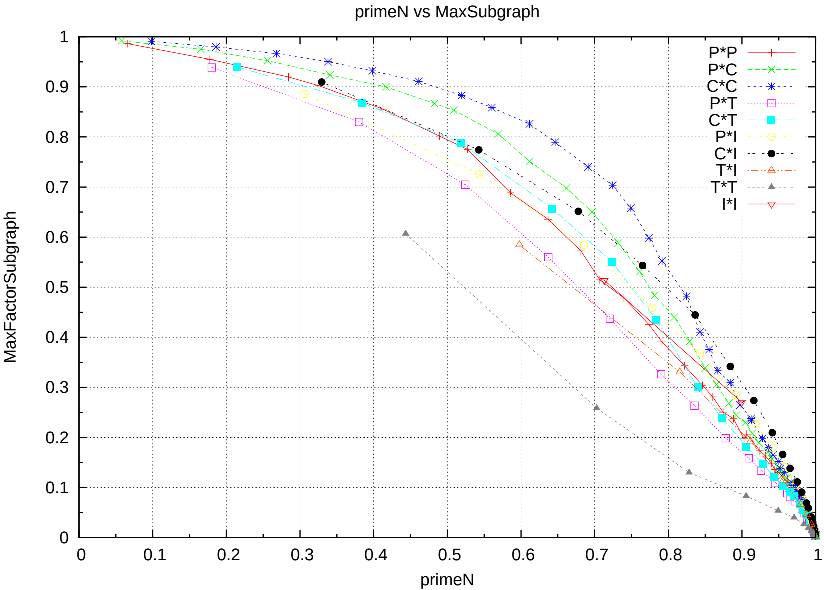

Pertubation vs PrimeN
PrimeN
Pertubation
P*P
P*C
C*C
P*T
C*T
P*I
C*I
T*I
T*T
I*I
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24


primeN vs SuccessExactALLFactors
Success
PrimeN
P*P
P*C
C*C
P*T
C*T
P*I
C*I
T*I
T*T
I*I
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Maximal Factorized Subgraph H



Ratio of $H = \frac{1}{2}\left(\frac{|V(H)|}{|V(G)|} + \frac{|E(H)|}{|E(G)|}\right) = \frac{1}{2}\left(\frac{19}{35} + \frac{51}{106}\right) = \frac{1}{2}(0.54 + 0.48) = 0.51$


primeN vs MaxSubgraph
MaxFactorSubgraph
primeN
P*P
P*C
C*C
P*T
C*T
P*I
C*I
T*I
T*T
I*I
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Summary

- New Local Approach for PFD that runs in $O(|V|\Delta^6)$ time.
- Suitable Results for Approximate Graph Products.

Outlook

- What if the subproducts are approximate products?
- Approximate products w.r.t. other products (Cartesian, direct, ...)
- Generalization (factorization of directed graphs, weighted graphs, hypergraphs) and Recognition of approximate graph products of those graphs.
- Preprocessing step via statistical approaches (degree distributions, shortest paths distributions, ...) that gives us (at least) necessary conditions to decide that a prime graph is either very similar to a product graph or not

Download

`http://www.bioinf.uni-leipzig.de/~marc/download.html`

Thanks to Peter F. Stadler, Wilfried Imrich, Werner Klöckl and Lydia Gringmann!

Thank you for your attention!