Small-Area Orthogonal Drawings of 3-connected graphs

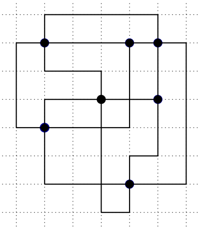
Therese Biedl [1] Jens Schmidt [2]

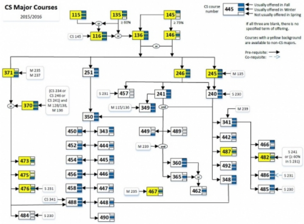
[1]University of Waterloo, Canada. *biedl@uwaterloo.ca*

[2]TU Ilmenau, Germany. *jens.schmidt@tu-ilmenau.de*

September 25, 2015

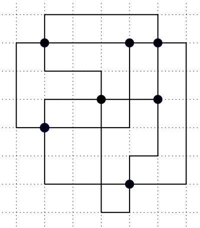
Orthogonal graph drawing

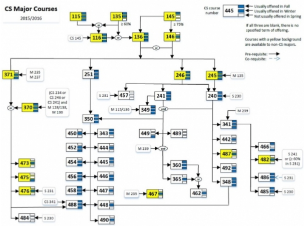




- Vertices of graph → points on an (integer) grid.
- Edges of graph → paths of the grid.
- No overlap or touching. Crossings ok.

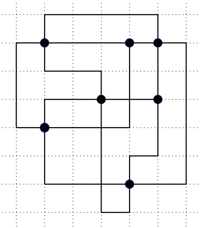
Orthogonal graph drawing

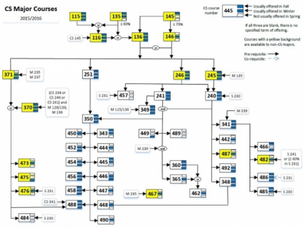




- Vertices of graph → points on an (integer) grid.
- Edges of graph → paths of the grid.
- No overlap or touching. Crossings ok.
- Requirement: At most 4 edges at each vertex (*4-graph*)

Orthogonal graph drawing

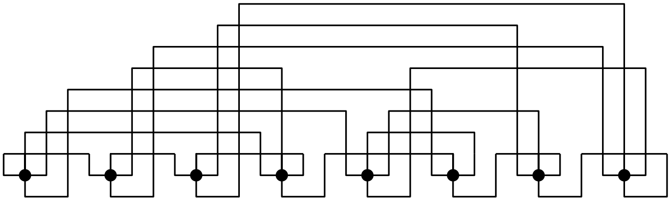




- Vertices of graph → points on an (integer) grid.
- Edges of graph → paths of the grid.
- No overlap or touching. Crossings ok.
- Requirement: At most 4 edges at each vertex (*4-graph*)
- Goal: Small area if drawn on grid.
  - width $W = \#$ columns, height $H = \#$ rows
  - area $A := H \cdot W = \#$ grid points.

Orthogonal drawing basics

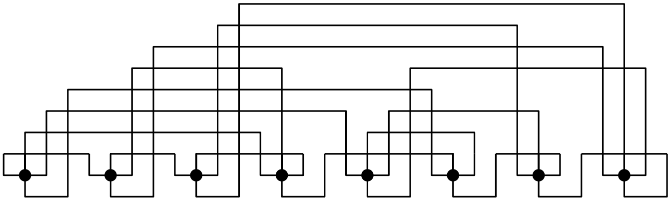
- Easy to see: every 4-graph has an orthogonal drawing



- $\approx 4n \times 2n$-grid $\Rightarrow$ area $\leq 8n^2 + O(n)$.
- Can we do better?

Orthogonal drawing basics

- Easy to see: every 4-graph has an orthogonal drawing



- $\approx 4n \times 2n$-grid $\Rightarrow$ area $\leq 8n^2 + O(n)$.
- Can we do better?
- Valiant'81: Some 4-graphs need $\Omega(n^2)$ area.
  - Some 4-graphs need $\Omega(n^2)$ crossings.
  - Each crossing needs a grid-point.
- But constant is tiny! (area $\geq 10^{-6}n^2$ or so...)

Orthogonal drawing timeline

Orthogonal drawings with area $\leq c \cdot n^2 + o(n^2)$:

$\downarrow$ $8n^2$ area

Orthogonal drawing timeline

Orthogonal drawings with area $\leq c \cdot n^2 + o(n^2)$:

↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area

Orthogonal drawing timeline

Orthogonal drawings with area $\leq c \cdot n^2 + o(n^2)$:

↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
↓	n^2 area

Orthogonal drawing timeline

Orthogonal drawings with area $\leq c \cdot n^2 + o(n^2)$:

$8n^2$ area

1992 Schäffter (published in 1995)

$4n^2$ area

1993 B. (published as B.& Kant 1994/1998)

n^2 area (Let's review this, we'll build on top)

BK-algorithm revisited

- Assume G is 2-connected 4-graph.

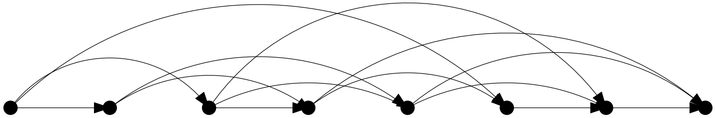
  (Handle 2-connected components separately, then merge.)

BK-algorithm revisited

- Assume G is 2-connected 4-graph.

  (Handle 2-connected components separately, then merge.)

- Find a vertex order $v_1, \ldots, v_n$ that is an *st-order*:



  - $indeg(v_i) \geq 1$ for $i > 1$.
  - $outdeg(v_i) \geq 1$ for $i < n$.

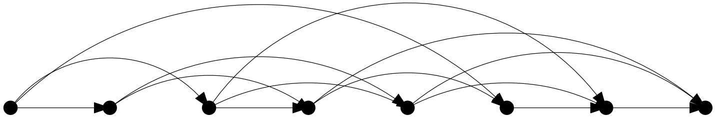
(This exists [LEC'64] and can be found in linear time [ET'76].)

BK-algorithm revisited

- Assume G is 2-connected 4-graph.

  (Handle 2-connected components separately, then merge.)

- Find a vertex order $v_1, \ldots, v_n$ that is an *st-order*:



  - $indeg(v_i) \geq 1$ for $i > 1$.
  - $outdeg(v_i) \geq 1$ for $i < n$.

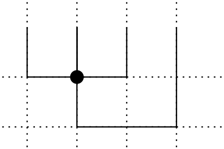
  (This exists [LEC'64] and can be found in linear time [ET'76].)

- For 4-graph: $indeg(v) \in \{1, 2, 3\}$ except for v_1, v_n.

BK-algorithm revisited

Add vertices to drawing in order.

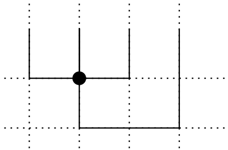
Place v_1:



BK-algorithm revisited

Add vertices to drawing in order.

Place v_1:

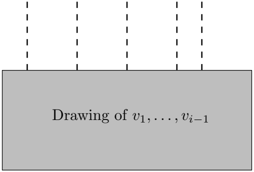


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 3$:

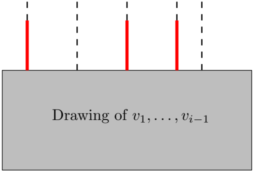


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 3$:

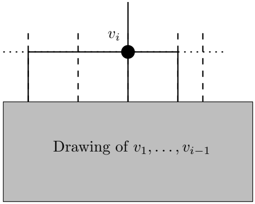


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 3$:

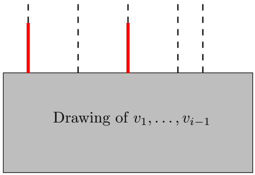


- Invariant: All unfinished edges end in upward ray.
- Rows: added 1
- Columns: added $0 \leq outdeg(v_i) - 1$

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 2$:

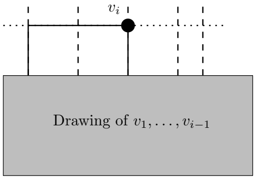


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 2$:

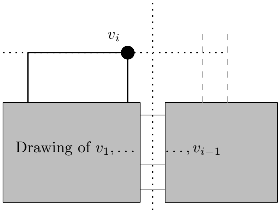


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 2$:

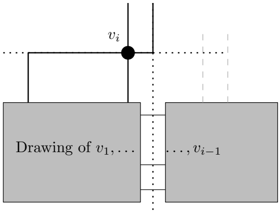


- Invariant: All unfinished edges end in upward ray.

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 2$:

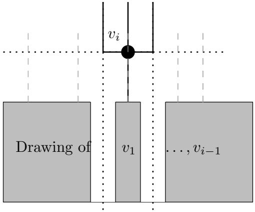


- Invariant: All unfinished edges end in upward ray.
- Rows: added 1
- Columns: added $1 \leq outdeg(v_i) - 1$

BK-algorithm revisited

Add vertices to drawing in order.

Place v_i, $indeg(v_i) = 1$:

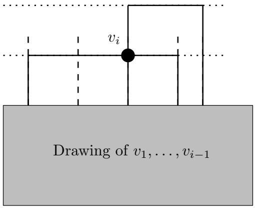


- Invariant: All unfinished edges end in upward ray.
- Rows: added 1
- Columns: added $2 \leq outdeg(v_i) - 1$

BK-algorithm revisited

Add vertices to drawing in order.

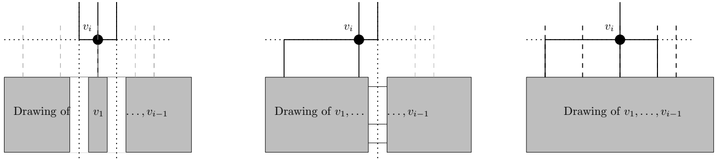
Place v_i, $indeg(v_i) = 4$:



- Invariant: All unfinished edges end in upward ray.
- Only happens for $i = n$.

BK-algorithm revisited

Add vertices to drawing in order.



- Invariant: All unfinished edges end in upward ray.
- Rows: added 1 (if $i \neq 1, n$)
- Columns: added $outdeg(v_i) - 1$ (if $i \neq 1, n$)

Theorem (B.,Kant)

Every 4-graph has an orthogonal drawing in an $n \times n$-grid.

Orthogonal drawing timeline

Orthogonal drawings with area $c \cdot n^2 + o(n^2)$:

↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
↓	n^2 area

Orthogonal drawing timeline

Orthogonal drawings with area $c \cdot n^2 + o(n^2)$:

	$8n^2$ area
1992	Schäffter (published in 1995)
	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
	n^2 area
1994	Papakostas & Tollis, 1996/1998
	$0.76n^2$ area

Orthogonal drawing timeline

Orthogonal drawings with area $c \cdot n^2 + o(n^2)$:

$8n^2$ area

1992 Schäffter (published in 1995)

$4n^2$ area

1993 B. (published as B.& Kant 1994/1998)

n^2 area

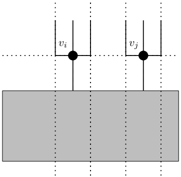
1994 Papakostas & Tollis, 1996/1998

$0.76n^2$ area

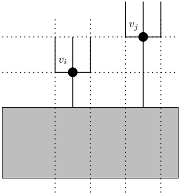
(Let's review this, we'll build on top)

PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.

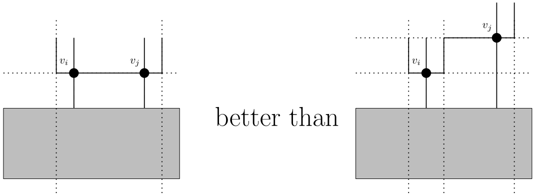


better than



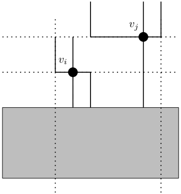
PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.

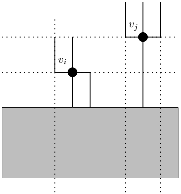


PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.

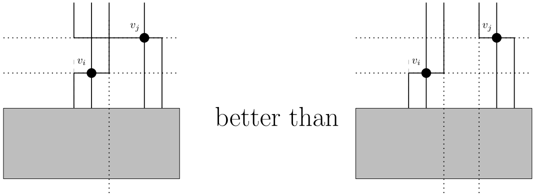


better than



PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.

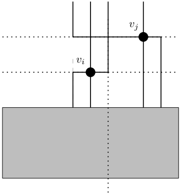


- Half-perimeter $HP = \#\text{rows} + \#\text{columns} \le 2n - \#\text{pairs}$

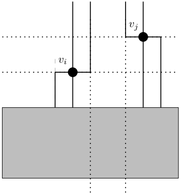
(ignoring lower-order terms)

PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.



better than



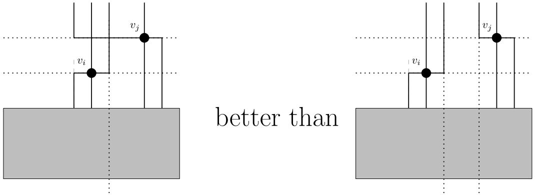
- Half-perimeter $HP = \#\text{rows} + \#\text{columns} \leq 2n - \#\text{pairs}$

(ignoring lower-order terms)

- Show: $\#\text{pairs} \geq \frac{1}{4}n \quad \Rightarrow HP \leq \frac{7}{4}n$

PT-algorithm: improvement by pairing

- Basic approach same as BK-algorithm.
- But: define some vertex-pairs $\{v_i, v_j\}$.
- Draw v_i and v_j together and save a row and/or column.



- Half-perimeter $HP = \#\text{rows} + \#\text{columns} \leq 2n - \#\text{pairs}$

(ignoring lower-order terms)

- Show: $\#\text{pairs} \geq \frac{1}{4}n \quad \Rightarrow HP \leq \frac{7}{4}n$
- Area $\leq \left(\frac{HP}{2}\right)^2 \leq \left(\frac{7}{8}n\right)^2 \approx 0.76n^2$

Theorem (Papakostas & Tollis)

Every 4-graph has an orthogonal drawing with area at most $0.76n^2$.

Orthogonal drawing timeline

Orthogonal drawings with area $c \cdot n^2 + o(n^2)$:

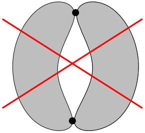
↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
↓	n^2 area
1994	Papakostas & Tollis, 1996/1998
↓	$0.76n^2$ area

Orthogonal drawing timeline

Orthogonal drawings with area $c \cdot n^2 + o(n^2)$:

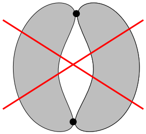
↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
↓	n^2 area
1994	Papakostas & Tollis, 1996/1998
↓	$0.76n^2$ area
2015	This talk: **$\mathbf{0.56n^2}$ area for 3-connected graphs**

- *3-connected*: Cannot disconnect by removing two vertices.



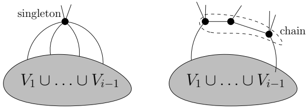
3-connected graphs

- *3-connected*: Cannot disconnect by removing two vertices.
- Many properties (3 vertex-disjoint paths, obtain from K_4 with BG-rules, . . .)
- Needed here: *canonical order*.

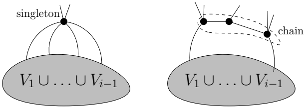


3-connected graphs

- *3-connected*: Cannot disconnect by removing two vertices.
- Many properties (3 vertex-disjoint paths, obtain from K_4 with BG-rules, ...)
- Needed here: *canonical order*.
  - Well-known for 3-connected **planar** graphs (Kant 1992)
  - Useful for lots of planar GD-algorithms:
    - Straight-line convex grid-drawings
    - Orthogonal planar drawings with area $\approx 0.44n^2$
    - Lots more ...

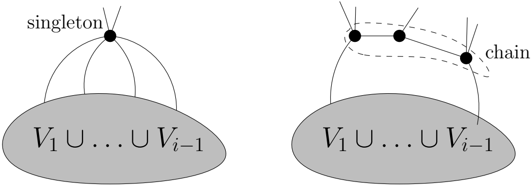


- *3-connected*: Cannot disconnect by removing two vertices.
- Many properties (3 vertex-disjoint paths, obtain from K_4 with BG-rules, ...)



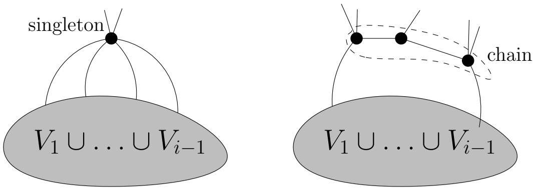
- Needed here: *canonical order*.
  - Well-known for 3-connected **planar** graphs (Kant 1992)
  - Useful for lots of planar GD-algorithms:
    - Straight-line convex grid-drawings
    - Orthogonal planar drawings with area $\approx 0.44n^2$
    - Lots more ...
  - But does this exist for non-planar graphs?

Canonical order for non-planar 3-connected graphs



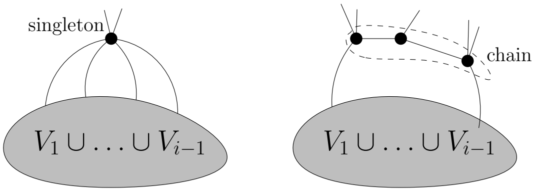
- de Fraysseix & Ossona de Mendez, 1994:
  "*Remark.* Such a canonical ordering can be easily generalized to triconnected non-planar graphs."

Canonical order for non-planar 3-connected graphs



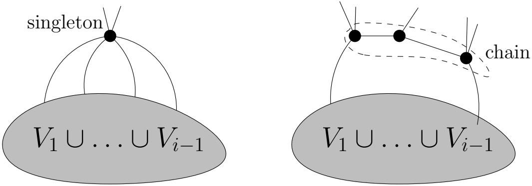
- de Fraysseix & Ossona de Mendez, 1994:
  "*Remark.* Such a canonical ordering can be easily generalized to triconnected non-planar graphs."
- Cheriyan & Maheshwari, 1988:
  Non-separating ear decomposition (implies canonical ordering).

Canonical order for non-planar 3-connected graphs



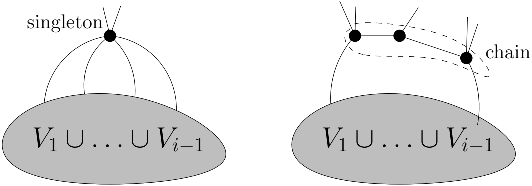
- de Fraysseix & Ossona de Mendez, 1994:
  "*Remark.* Such a canonical ordering can be easily generalized to triconnected non-planar graphs."
- Cheriyan & Maheshwari, 1988:
  Non-separating ear decomposition (implies canonical ordering).
- Mondshein, 1971:
  (2,1) edge-partition (implies canonical ordering).

Canonical order for non-planar 3-connected graphs



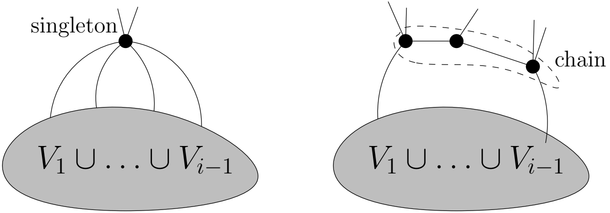
- de Fraysseix & Ossona de Mendez, 1994:
  "*Remark.* Such a canonical ordering can be easily generalized to triconnected non-planar graphs."
- Cheriyan & Maheshwari, 1988:
  Non-separating ear decomposition (implies canonical ordering).
- Mondshein, 1971:
  (2,1) edge-partition (implies canonical ordering).
- None of these discuss time to find it (quadratic?)

Canonical order for non-planar 3-connected graphs



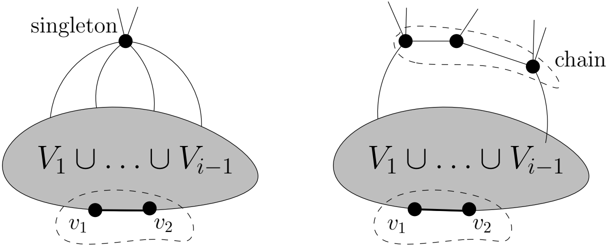
- de Fraysseix & Ossona de Mendez, 1994:
  "*Remark.* Such a canonical ordering can be easily generalized to triconnected non-planar graphs."
- Cheriyan & Maheshwari, 1988:
  Non-separating ear decomposition (implies canonical ordering).
- Mondshein, 1971:
  (2,1) edge-partition (implies canonical ordering).
- None of these discuss time to find it (quadratic?)
- Schmidt, 2014: Mondshein sequence (hence canonical order) can be found in linear time.

Canonical order for non-planar 3-connected graphs



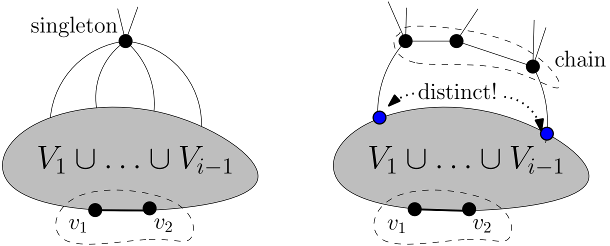
- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$

Canonical order for non-planar 3-connected graphs



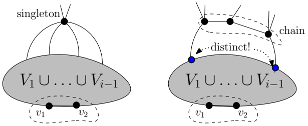
- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).

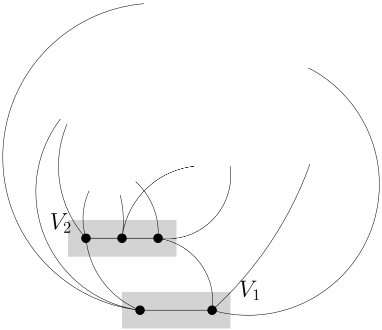
Canonical order for non-planar 3-connected graphs



- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path,
    exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.

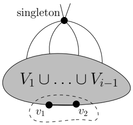
Canonical order for non-planar 3-connected graphs

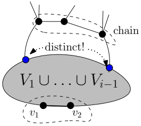


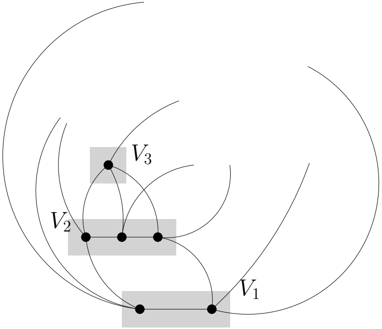


- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path, exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.
- V_k is a singleton without successor, adjacent to v_1.

Canonical order for non-planar 3-connected graphs

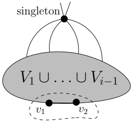


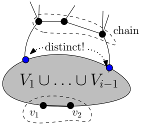


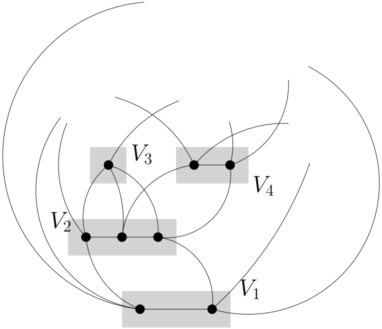


- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path, exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.
- V_k is a singleton without successor, adjacent to v_1.

Canonical order for non-planar 3-connected graphs

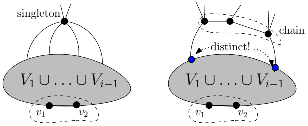


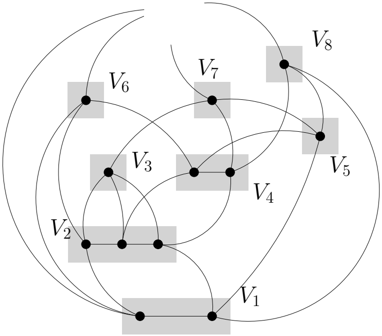




- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path, exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.
- V_k is a singleton without successor, adjacent to v_1.

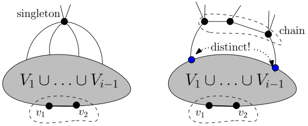
Canonical order for non-planar 3-connected graphs

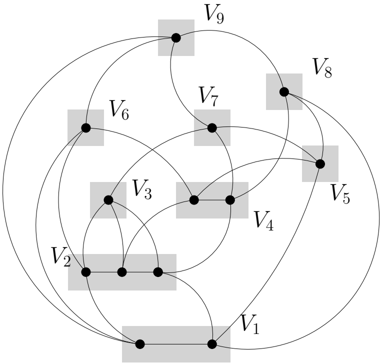




- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path, exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.
- V_k is a singleton without successor, adjacent to v_1.

Canonical order for non-planar 3-connected graphs

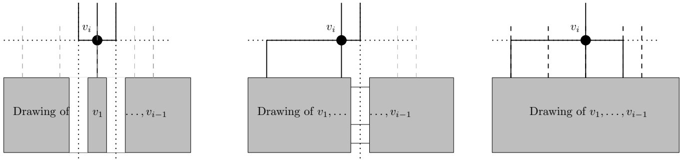




- Sequence of disjoint vertex sets $V_1, V_2, \ldots, V_k$
- V_1 is an edge (v_1, v_2).
- V_i for $1 < i < k$ is:
  - a *singleton*: $V_i = \{z\}$, ≥ 2 predecessors, ≥ 1 successor –OR–
  - a *chain*: $V_i = \{z_1, \ldots, z_\ell\}$ induces path,
    exactly 2 predecessors at z_1, z_ℓ, ≥ 1 successor at each z_j.
- V_k is a singleton without successor, adjacent to v_1.

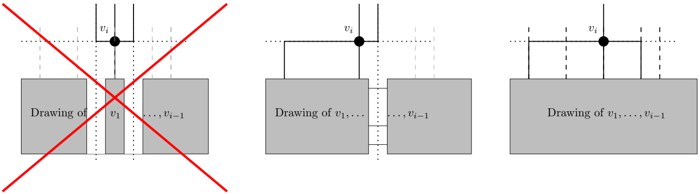
Orthogonal drawing with 3-canonical order

- Approach same as BK-algorithm, but use 3-canonical order.
- Draw singleton exactly as before:



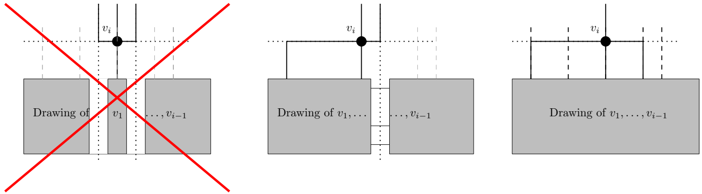
Orthogonal drawing with 3-canonical order

- Approach same as BK-algorithm, but use 3-canonical order.
- Draw singleton exactly as before:

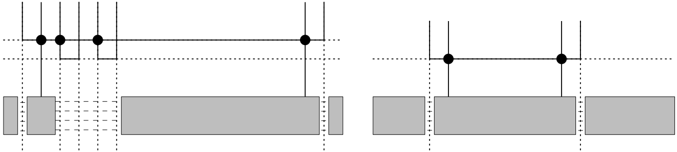


Orthogonal drawing with 3-canonical order

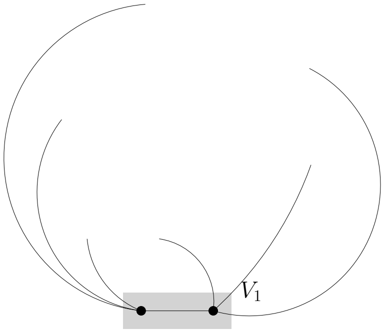
- Approach same as BK-algorithm, but use 3-canonical order.
- Draw singleton exactly as before:

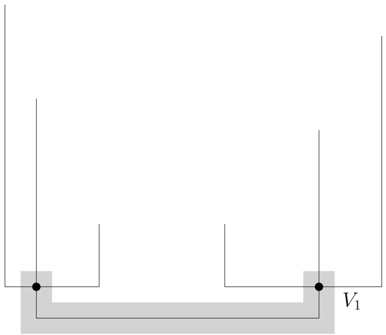


- Draw a (long or short) chain using ≤ 2 rows:

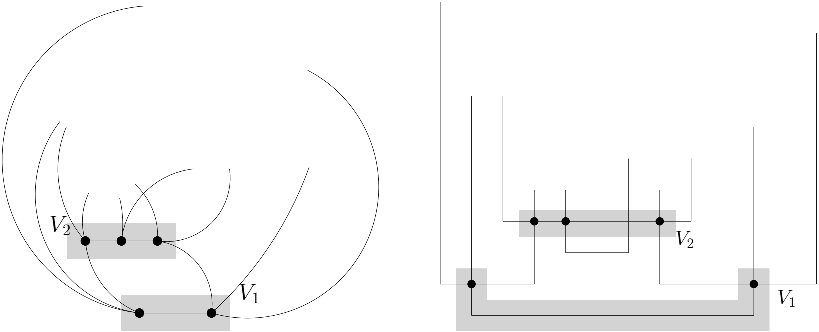


Orthogonal drawing with 3-canonical order

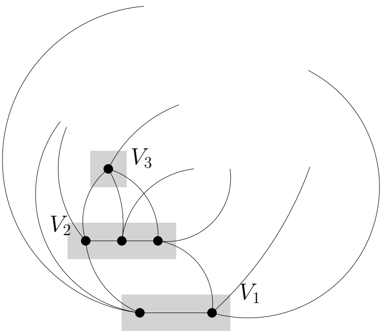


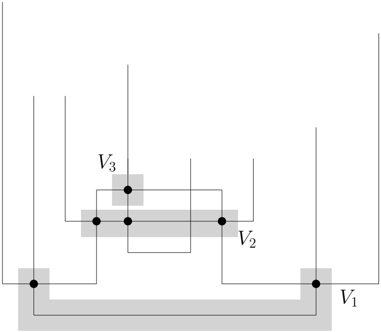


Orthogonal drawing with 3-canonical order

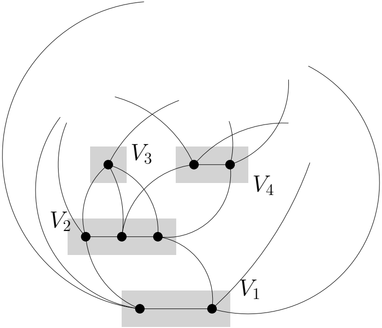


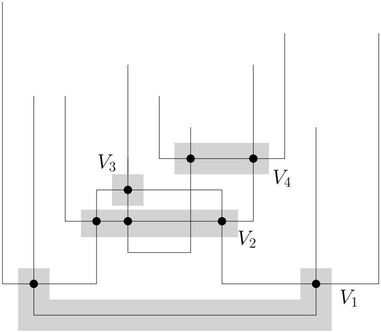
Orthogonal drawing with 3-canonical order



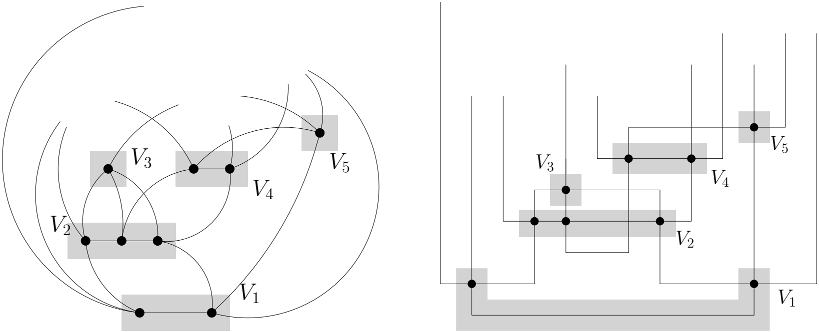


Orthogonal drawing with 3-canonical order

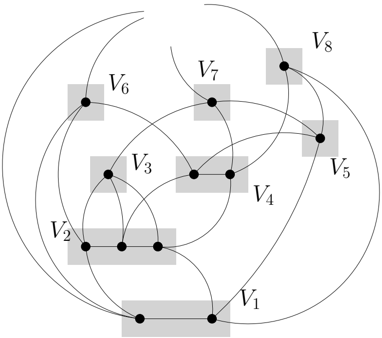


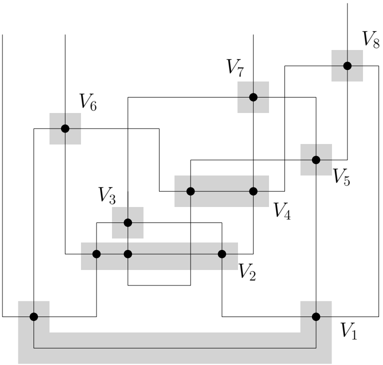


Orthogonal drawing with 3-canonical order

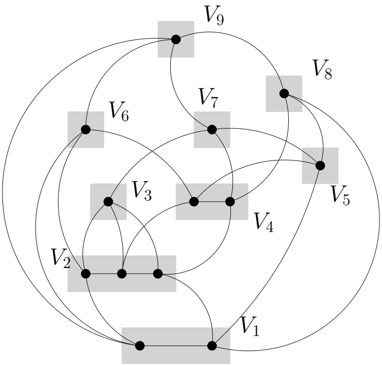


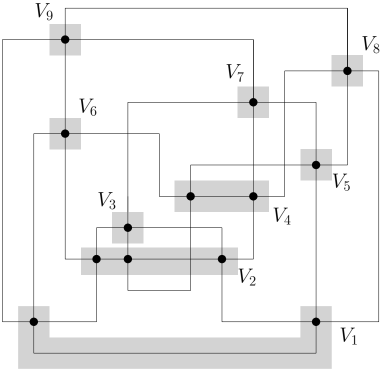
Orthogonal drawing with 3-canonical order



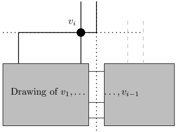


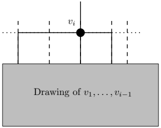
Orthogonal drawing with 3-canonical order



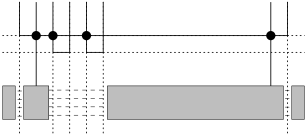


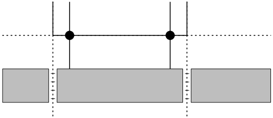
- Draw singleton exactly as before:





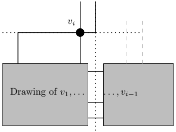
- Draw a (long or short) chain using ≤ 2 rows:

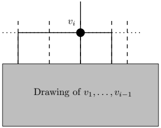




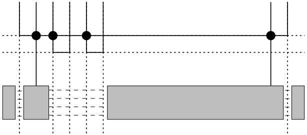
Orthogonal drawing with 3-canonical order

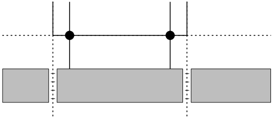
- Draw singleton exactly as before:





- Draw a (long or short) chain using ≤ 2 rows:



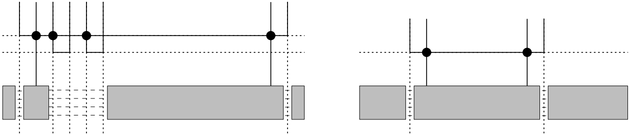


- Chains save rows—do we have many?

Orthogonal drawing with 3-canonical order

- Draw singleton exactly as before:

- Draw a (long or short) chain using ≤ 2 rows:

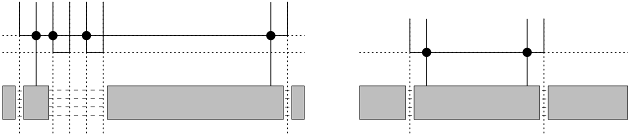


- Chains save rows—do we have many?
- For every 3-singleton, must have a vertex in a chain.

Orthogonal drawing with 3-canonical order

- Draw singleton exactly as before:

- Draw a (long or short) chain using ≤ 2 rows:

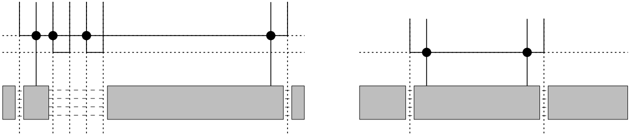


- Chains save rows—do we have many?
- For every 3-singleton, must have a vertex in a chain.
- Some equation fiddling:

$$HP \leq \frac{3}{2}n + \frac{1}{2}\#\{\textit{indeg}\text{-2-vertices}\}$$

- Draw singleton exactly as before:

- Draw a (long or short) chain using ≤ 2 rows:



- Chains save rows—do we have many?
- For every 3-singleton, must have a vertex in a chain.
- Some equation fiddling:

$$HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-}2\text{-vertices}\}$$

- But that could still be $\approx 2n$.

Orthogonal drawing with 3-canonical order

- We get: $\boxed{HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\}}$

Orthogonal drawing with 3-canonical order

- We get: $\boxed{HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\}}$
- 3-canonical order is an *st*-order: could apply PT-algorithm.
  $HP \leq 2n - \#\{pairs\}$

Orthogonal drawing with 3-canonical order

- We get: $$\boxed{HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\}}$$
- 3-canonical order is an *st*-order: could apply PT-algorithm.
$$HP \leq 2n - \#\{pairs\}$$
- Closer inspection of PT-algorithm:
$$\#\{pairs\} \geq \frac{1}{4}n - \frac{1}{4}\#\{indeg\text{-2-vertices}\}$$

Orthogonal drawing with 3-canonical order

- We get: $\boxed{HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\}}$
- 3-canonical order is an *st*-order: could apply PT-algorithm.
  $$HP \leq 2n - \#\{pairs\}$$
- Closer inspection of PT-algorithm:
  $$\#\{pairs\} \geq \frac{1}{4}n - \frac{1}{4}\#\{indeg\text{-2-vertices}\}$$
- PT-algorithm gets $\boxed{HP \leq \frac{7}{4}n - \frac{1}{4}\#\{indeg\text{-2-vertices}\}}$

Orthogonal drawing with 3-canonical order

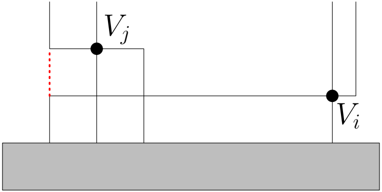
- We get: $\boxed{HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\}}$
- 3-canonical order is an *st*-order: could apply PT-algorithm.
  $$HP \leq 2n - \#\{pairs\}$$
- Closer inspection of PT-algorithm:
  $$\#\{pairs\} \geq \frac{1}{4}n - \frac{1}{4}\#\{indeg\text{-2-vertices}\}$$
- PT-algorithm gets $\boxed{HP \leq \frac{7}{4}n - \frac{1}{4}\#\{indeg\text{-2-vertices}\}}$
- Not both algorithms can be in worst-case.

Theorem

One of the above two algorithms achieves half-perimeter $\leq \frac{5}{3}n$ *(hence area* $\approx 0.69n^2$*).*

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

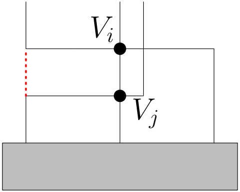


(2-singleton + 2-singleton)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

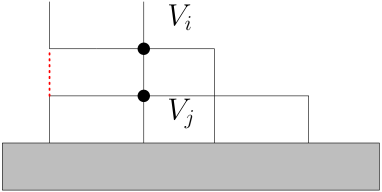


(2-singleton + 2-singleton, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-}2\text{-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

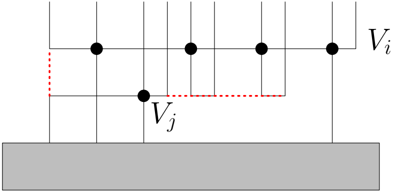


(3-singleton + 2-singleton, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

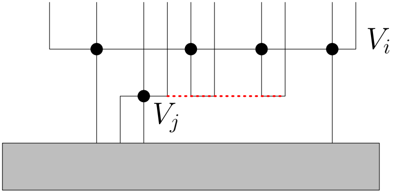


(2-singleton + long chain, no edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

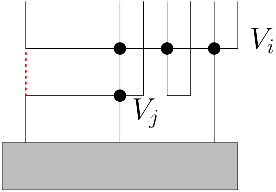


(2-singleton + long chain, no edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

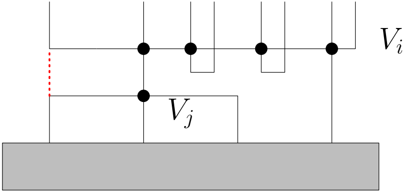


(2-singleton + long chain, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

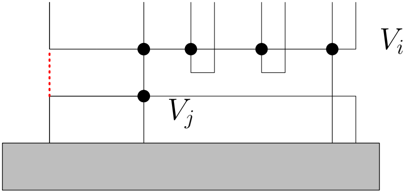


(3-singleton + long chain, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

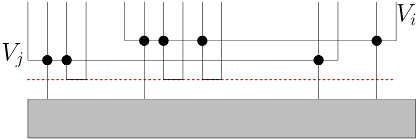
- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.



(3-singleton + long chain, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

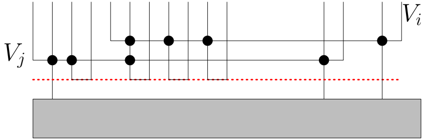


(long chain + long chain, no edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

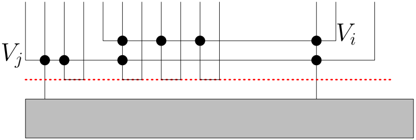


(long chain + long chain, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

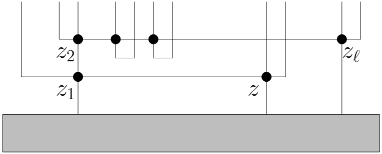


(long chain + long chain, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

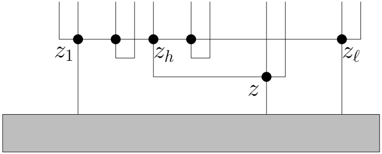


(long chain + 2-singleton, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

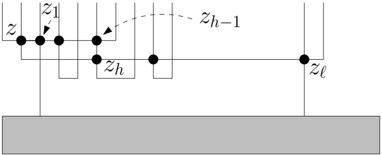


(long chain + 2-singleton, one edge)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

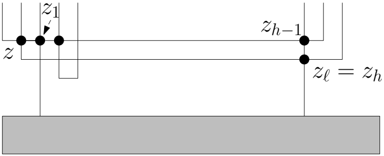


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

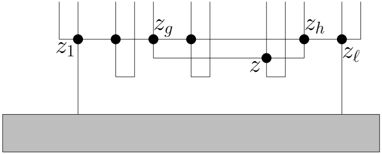
- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.



(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

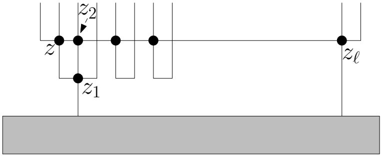


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

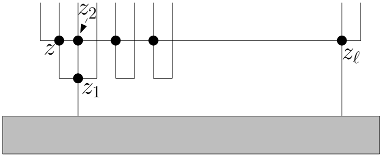


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

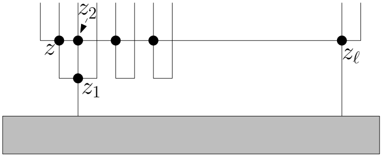


(long chain + 2-singleton, two edges)(whew!)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

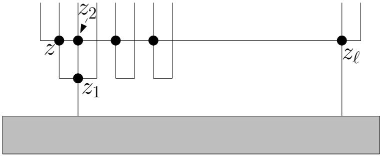


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$
- Every 2-singleton and every chain in a pair.

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

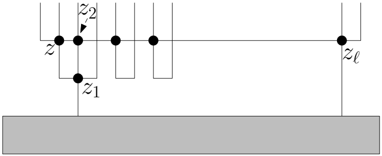


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{\mathit{indeg}\text{-2-vertices}\} - \#\{\text{pairs}\}$
- Every 2-singleton and every chain in a pair.
- $\Rightarrow$ Every *indeg*-2-vertex in a pair.

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.

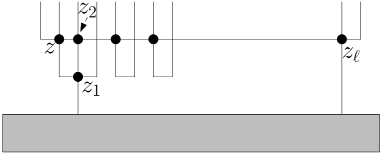


(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{indeg\text{-2-vertices}\} - \#\{\text{pairs}\}$
- Every 2-singleton and every chain in a pair.
- $\Rightarrow$ Every *indeg*-2-vertex in a pair.
- $\#\{\text{pairs}\} \geq \frac{1}{2}\#\{indeg\text{-2-vertices}\} \quad \Rightarrow HP \leq \frac{3}{2}n$

Orthogonal drawing with 3-canonical order

- Now: Use pairing-idea for vertex groups.
- Define vertex-group-pairs $\{V_i, V_j\}$ in a special way
- Draw V_i and V_j together and save a row and/or column.



(long chain + 2-singleton, two edges)

- $HP \leq \frac{3}{2}n + \frac{1}{2}\#\{\textit{indeg}\text{-2-vertices}\} - \#\{\text{pairs}\}$
- Every 2-singleton and every chain in a pair.
- $\Rightarrow$ Every *indeg*-2-vertex in a pair.
- $\#\{\text{pairs}\} \geq \frac{1}{2}\#\{\textit{indeg}\text{-2-vertices}\} \quad \Rightarrow HP \leq \frac{3}{2}n$

Theorem

Every 3-connected 4-graph has an orthogonal drawing with area at most $(\frac{3}{4}n)^2 \approx 0.56n^2$.

Conclusion

↓	$8n^2$ area
1992	Schäffter (published in 1995)
↓	$4n^2$ area
1993	B. (published as B.& Kant 1994/1998)
↓	n^2 area
1994	Papakostas & Tollis, 1996/1998
↓	$0.76n^2$ area
2015	This talk:
	$0.56n^2$ area for 3-connected graphs

Conclusion

↓ $8n^2$ area

1992 Schäffter (published in 1995)

↓ $4n^2$ area

1993 B. (published as B.& Kant 1994/1998)

↓ n^2 area

1994 Papakostas & Tollis, 1996/1998

↓ $0.76n^2$ area

2015 This talk:

↓ **$\mathbf{0.56n^2}$ area for 3-connected graphs**

20??

Can you do better?

Conclusion and Open Problems

- Can we achieve $0.56n^2$ area for 2-connected 4-graphs?

  (Merge 3-connected components?)

Conclusion and Open Problems

- Can we achieve $0.56n^2$ area for 2-connected 4-graphs?

  (Merge 3-connected components?)

- Can we achieve $(2-\varepsilon)n$ bends?

  (We can do $2n - \#\{\text{2-singletons}\}$.)

Conclusion and Open Problems

- Can we achieve $0.56n^2$ area for 2-connected 4-graphs?

  (Merge 3-connected components?)

- Can we achieve $(2-\varepsilon)n$ bends?

  (We can do $2n - \#\{\text{2-singletons}\}$.)

- Use non-planar 3-canonical ordering for other graph drawing algorithms?

Conclusion and Open Problems

- Can we achieve $0.56n^2$ area for 2-connected 4-graphs?

  (Merge 3-connected components?)

- Can we achieve $(2-\varepsilon)n$ bends?

  (We can do $2n - \#\{2\text{-singletons}\}$.)

- Use non-planar 3-canonical ordering for other graph drawing algorithms?