Width-Optimal Visibility Representations of Plane Graphs

Speaker: Chun-Cheng Lin

Coauthors: Jia-Hao Fan   Hsueh-I Lu   Hsu-Chun Yen

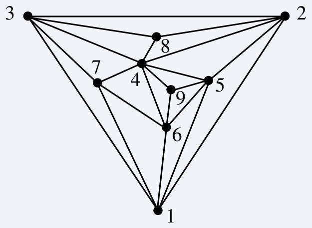
National Taiwan University, Taipei, Taiwan

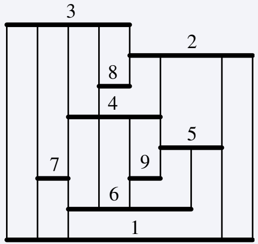
(Presented at the 18th International Symposium on Algorithms and Computation (ISAAC 2007))

Outline

Visibility Representation (a.k.a., Visibility Drawing)

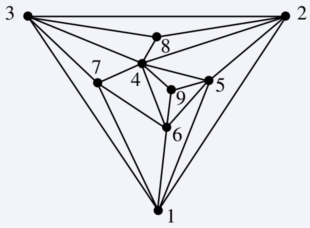
- Visibility Representation
- Node segment
- Edge segment
- Measuring the drawing area in a grid

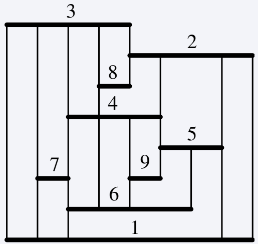




Visibility Representation (a.k.a., Visibility Drawing)

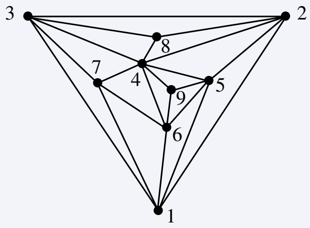
- Visibility Representation
- Node segment
- Edge segment
- Measuring the drawing area in a grid

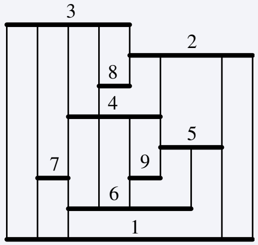




Visibility Representation (a.k.a., Visibility Drawing)

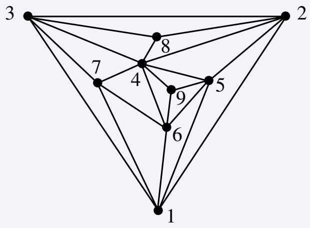
- Visibility Representation
- Node segment
- Edge segment
- Measuring the drawing area in a grid

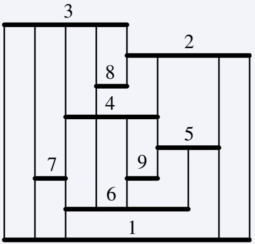




Visibility Representation (a.k.a., Visibility Drawing)

- Visibility Representation
- Node segment
- Edge segment
- Measuring the drawing area in a grid





Compactness of Visibility Representation

- Otten and van Wijk (1978)
  - first known algorithm for visibility drawings;
  - no bound for the compactness of the output.

worst-case upper bound			
required height		required width	
$n-1$	(Rosenstiehl & Tarjan, 1986; Tamassia & Tollis, 1986)	$2n-5$	(Rosenstiehl & Tarjan,1986; Tamassia & Tollis, 1986; Nummenmaa, 1992)
$\lfloor \frac{15n}{16} \rfloor$	(Zhang & He, 2003)		
$\lfloor \frac{5n}{6} \rfloor$	(Zhang & He, 2005)	$\lfloor \frac{22n-42}{15} \rfloor$	(Lin, Lu, and Sun, 2004)
$\lfloor \frac{4n-1}{5} \rfloor$	(Zhang & He, 2006)	$\lfloor \frac{13n-24}{9} \rfloor$	(Zhang & He, 2005)
$\frac{2n}{3} + 2\lceil \sqrt{n/2} \rceil$	(He & Zhang, 2006)	$\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$	(He & Zhang, 2006)
lower bound			
The size of the required area is at least $\lfloor \frac{2n}{3} \rfloor \times (\lfloor \frac{4n}{3} \rfloor - 3)$ (Zhang & He, 2005)			

Compactness of Visibility Representation

- Otten and van Wijk (1978)
  - first known algorithm for visibility drawings;
  - no bound for the compactness of the output.

worst-case upper bound			
required height		required width	
$n-1$	(Rosenstiehl & Tarjan, 1986; Tamassia & Tollis, 1986)	$2n-5$	(Rosenstiehl & Tarjan,1986; Tamassia & Tollis, 1986; Nummenmaa, 1992)
$\lfloor \frac{15n}{16} \rfloor$	(Zhang & He, 2003)		
$\lfloor \frac{5n}{6} \rfloor$	(Zhang & He, 2005)	$\lfloor \frac{22n-42}{15} \rfloor$	(Lin, Lu, and Sun, 2004)
$\lfloor \frac{4n-1}{5} \rfloor$	(Zhang & He, 2006)	$\lfloor \frac{13n-24}{9} \rfloor$	(Zhang & He, 2005)
$\frac{2n}{3}+2\lceil\sqrt{n/2}\rceil$	(He & Zhang, 2006)	$\frac{4n}{3}+2\lceil\sqrt{n}\rceil$	(He & Zhang, 2006)

lower bound

The size of the required area is at least $\lfloor \frac{2n}{3} \rfloor \times (\lfloor \frac{4n}{3} \rfloor - 3)$ (Zhang & He, 2005)

Compactness of Visibility Representation

- Otten and van Wijk (1978)
  - first known algorithm for visibility drawings;
  - no bound for the compactness of the output.

worst-case upper bound			
required height		required width	
$n-1$	(Rosenstiehl & Tarjan, 1986; Tamassia & Tollis, 1986)	$2n-5$	(Rosenstiehl & Tarjan,1986; Tamassia & Tollis, 1986; Nummenmaa, 1992)
$\lfloor \frac{15n}{16} \rfloor$	(Zhang & He, 2003)		
$\lfloor \frac{5n}{6} \rfloor$	(Zhang & He, 2005)	$\lfloor \frac{22n-42}{15} \rfloor$	(Lin, Lu, and Sun, 2004)
$\lfloor \frac{4n-1}{5} \rfloor$	(Zhang & He, 2006)	$\lfloor \frac{13n-24}{9} \rfloor$	(Zhang & He, 2005)
$\frac{2n}{3} + 2\lceil \sqrt{n/2} \rceil$	(He & Zhang, 2006)	$\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$	(He & Zhang, 2006)
lower bound			
The size of the required area is at least $\lfloor \frac{2n}{3} \rfloor \times (\lfloor \frac{4n}{3} \rfloor - 3)$ (Zhang & He, 2005)			

- Lin, Lu, and Sun (2004) conjectured ...
  - no wider than $\frac{4n}{3} + O(1)$.

Compactness of Visibility Representation

- Otten and van Wijk (1978)
  - first known algorithm for visibility drawings;
  - no bound for the compactness of the output.

worst-case upper bound			
required height		required width	
$n-1$	(Rosenstiehl & Tarjan, 1986; Tamassia & Tollis, 1986)	$2n-5$	(Rosenstiehl & Tarjan,1986; Tamassia & Tollis, 1986; Nummenmaa, 1992)
$\lfloor \frac{15n}{16} \rfloor$	(Zhang & He, 2003)		
$\lfloor \frac{5n}{6} \rfloor$	(Zhang & He, 2005)	$\lfloor \frac{22n-42}{15} \rfloor$	(Lin, Lu, and Sun, 2004)
$\lfloor \frac{4n-1}{5} \rfloor$	(Zhang & He, 2006)	$\lfloor \frac{13n-24}{9} \rfloor$	(Zhang & He, 2005)
$\frac{2n}{3}+2\lceil\sqrt{n/2}\rceil$	(He & Zhang, 2006)	$\frac{4n}{3}+2\lceil\sqrt{n}\rceil$	(He & Zhang, 2006)
lower bound			
The size of the required area is at least $\lfloor \frac{2n}{3} \rfloor \times (\lfloor \frac{4n}{3} \rfloor - 3)$ (Zhang & He, 2005)			

- Lin, Lu, and Sun (2004) conjectured ...
  - no wider than $\frac{4n}{3}+O(1)$.

Our Main Result

Theorem

Given an n-node plane triangulation G, a visibility drawing of G with its width bounded by $\lfloor \frac{4n}{3} \rfloor - 2$ can be obtained in time $O(n)$.

- Our bound is the optimal
  because our bound differs the previously known lower bound $\frac{4n}{3} - 3$ (Zhang and He, 2005) only by a unit.
- Answering in the affirmative a conjecture of [Lin, Lu, Sun, 2004] about whether any visibility drawing no wider than $\frac{4n}{3} + O(1)$ can be obtained in polynomial time.
- Rather than conventionally using canonical ordering, st-numbering, or Schnyder's realizer as the initial input, our algorithm applies a new kind of ordering, called **constructive ordering**, of G to constructing the visibility drawing.

Our Main Result

Theorem

Given an n-node plane triangulation G, a visibility drawing of G with its width bounded by $\lfloor \frac{4n}{3} \rfloor - 2$ can be obtained in time $O(n)$.

- Our bound is the optimal
  because our bound differs the previously known lower bound $\frac{4n}{3} - 3$ (Zhang and He, 2005) only by a unit.
- Answering in the affirmative a conjecture of [Lin, Lu, Sun, 2004] about whether any visibility drawing no wider than $\frac{4n}{3} + O(1)$ can be obtained in polynomial time.
- Rather than conventionally using canonical ordering, st-numbering, or Schnyder's realizer as the initial input, our algorithm applies a new kind of ordering, called **constructive ordering**, of G to constructing the visibility drawing.

Our Main Result

Theorem

Given an n-node plane triangulation G, a visibility drawing of G with its width bounded by $\lfloor \frac{4n}{3} \rfloor - 2$ can be obtained in time $O(n)$.

- Our bound is the optimal because our bound differs the previously known lower bound $\frac{4n}{3} - 3$ (Zhang and He, 2005) only by a unit.
- Answering in the affirmative a conjecture of [Lin, Lu, Sun, 2004] about whether any visibility drawing no wider than $\frac{4n}{3} + O(1)$ can be obtained in polynomial time.
- Rather than conventionally using canonical ordering, st-numbering, or Schnyder's realizer as the initial input, our algorithm applies a new kind of ordering, called **constructive ordering**, of G to constructing the visibility drawing.

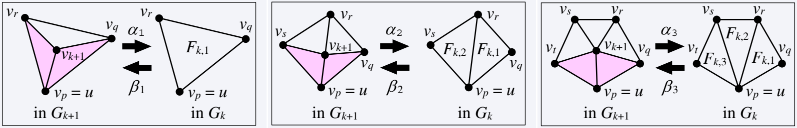
Our Main Result

Theorem

Given an n-node plane triangulation G, a visibility drawing of G with its width bounded by $\lfloor \frac{4n}{3} \rfloor - 2$ can be obtained in time $O(n)$.

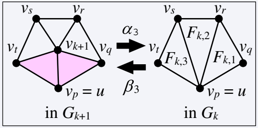
- Our bound is the optimal
  because our bound differs the previously known lower bound $\frac{4n}{3} - 3$ (Zhang and He, 2005) only by a unit.
- Answering in the affirmative a conjecture of [Lin, Lu, Sun, 2004] about whether any visibility drawing no wider than $\frac{4n}{3} + O(1)$ can be obtained in polynomial time.
- Rather than conventionally using canonical ordering, st-numbering, or Schnyder's realizer as the initial input, our algorithm applies a new kind of ordering, called **constructive ordering**, of G to constructing the visibility drawing.

Coalescing and Splitting Operations



(a) $deg(v_{k+1}) = 3$ in G_{k+1} (b) $deg(v_{k+1}) = 4$ in G_{k+1} (c) $deg(v_{k+1}) = 5$ in G_{k+1}

Coalescing and Splitting Operations

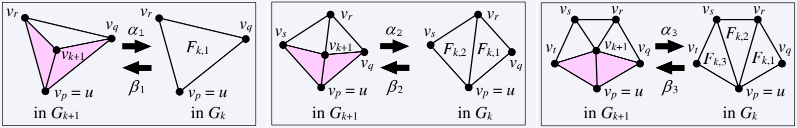


(c) $deg(v_{k+1}) = 5$ in G_{k+1}

When $deg(v_{k+1}) = 5$,

- $\alpha_3(v_{k+1}, u) =$ coalescing two nodes v_{k+1} and u
- $\beta_3(v_{k+1}, F_{k,1}, F_{k,2}, F_{k,3}) =$ splitting node v_{k+1} at faces $F_{k,1}, F_{k,2}, F_{k,3}$

Coalescing and Splitting Operations

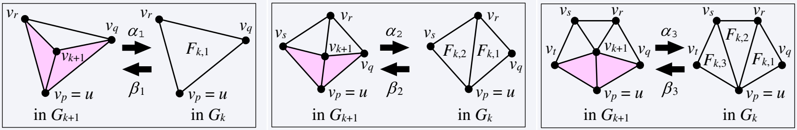


(a) $deg(v_{k+1}) = 3$ in G_{k+1} (b) $deg(v_{k+1}) = 4$ in G_{k+1} (c) $deg(v_{k+1}) = 5$ in G_{k+1}

When $deg(v_{k+1}) = 5$,

- $\alpha_3(v_{k+1}, u) =$ coalescing two nodes v_{k+1} and u
- $\beta_3(v_{k+1}, F_{k,1}, F_{k,2}, F_{k,3}) =$ splitting node v_{k+1} at faces $F_{k,1}, F_{k,2}, F_{k,3}$

Coalescing and Splitting Operations

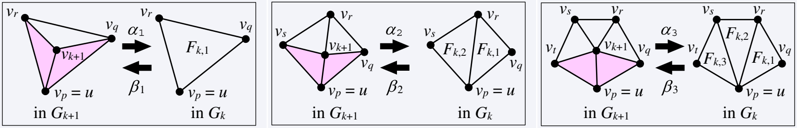


(a) $deg(v_{k+1}) = 3$ in G_{k+1} (b) $deg(v_{k+1}) = 4$ in G_{k+1} (c) $deg(v_{k+1}) = 5$ in G_{k+1}

When $deg(v_{k+1}) = 5$,

- $\alpha_3(v_{k+1}, u) =$ coalescing two nodes v_{k+1} and u
  $\equiv$ detaching two faces
- $\beta_3(v_{k+1}, F_{k,1}, F_{k,2}, F_{k,3}) =$ splitting node v_{k+1} at faces $F_{k,1}, F_{k,2}, F_{k,3}$
  $\equiv$ attaching two new faces

Coalescing and Splitting Operations



(a) $deg(v_{k+1}) = 3$ in G_{k+1} (b) $deg(v_{k+1}) = 4$ in G_{k+1} (c) $deg(v_{k+1}) = 5$ in G_{k+1}

When $deg(v_{k+1}) = 5$,

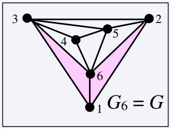
- $\alpha_3(v_{k+1}, u) =$ coalescing two nodes v_{k+1} and u
  $\equiv$ detaching two faces
- $\beta_3(v_{k+1}, F_{k,1}, F_{k,2}, F_{k,3}) =$ splitting node v_{k+1} at faces $F_{k,1}, F_{k,2}, F_{k,3}$
  $\equiv$ attaching two new faces
  $\equiv$ inserting a node at faces $F_{k,1}, F_{k,2}, F_{k,3}$

Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \dots, v_k$. We call $\pi = (v_1, v_2, \dots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \le k \le n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \le k \le n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.

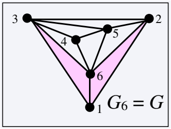


Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \ldots, v_k$. We call $\pi = (v_1, v_2, \ldots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \leq k \leq n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \leq k \leq n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.



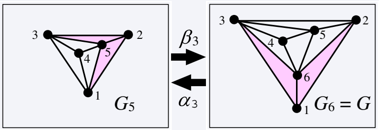
Note that there always exists a node with degree 3, 4, or 5 in any plane triangulation.

Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \ldots, v_k$. We call $\pi = (v_1, v_2, \ldots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \le k \le n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \le k \le n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.



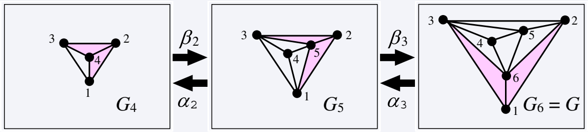
Note that there always exists a node with degree 3, 4, or 5 in any plane triangulation.

Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \ldots, v_k$. We call $\pi = (v_1, v_2, \ldots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \leq k \leq n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \leq k \leq n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.



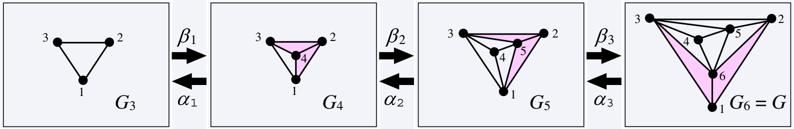
Note that there always exists a node with degree 3, 4, or 5 in any plane triangulation.

Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \ldots, v_k$. We call $\pi = (v_1, v_2, \ldots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \leq k \leq n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \leq k \leq n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.



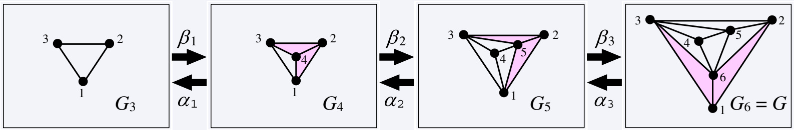
Note that there always exists a node with degree 3, 4, or 5 in any plane triangulation.

Constructive Ordering of a Plane Triangulation

Definition

G_k involves the k nodes $v_1, v_2, \ldots, v_k$. We call $\pi = (v_1, v_2, \ldots, v_n)$ a **constructive ordering** of a plane triangulation G if the following conditions hold for each k, $3 \leq k \leq n$:

1. G_k is a plane triangulation with outer edges v_1v_2, v_2v_3, v_3v_1;
2. if $3 \leq k \leq n-1$, then node v_{k+1} is split from a node in G_k, and the degree of node v_{k+1} is three, four, or five in G_{k+1}.



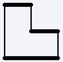
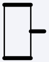
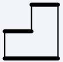
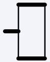
Note that there always exists a node with degree 3, 4, or 5 in any plane triangulation.

Lemma

Every n-node plane triangulation G has a constructive ordering, which can be found in $O(n)$ time.

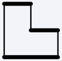
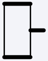
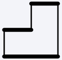
L-Shape and the β_1 Operation

- L-shape

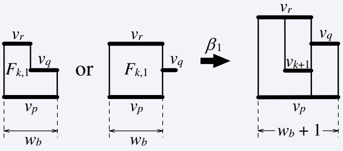
	Regular L-shape	Degenerated L-shape
right L-shape		
left L-shape		

L-Shape and the β_1 Operation

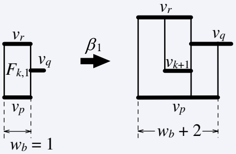
- L-shape

	Regular L-shape	Degenerated L-shape
right L-shape		
left L-shape		

- The β_1 operation (the degree-three splitting operation)



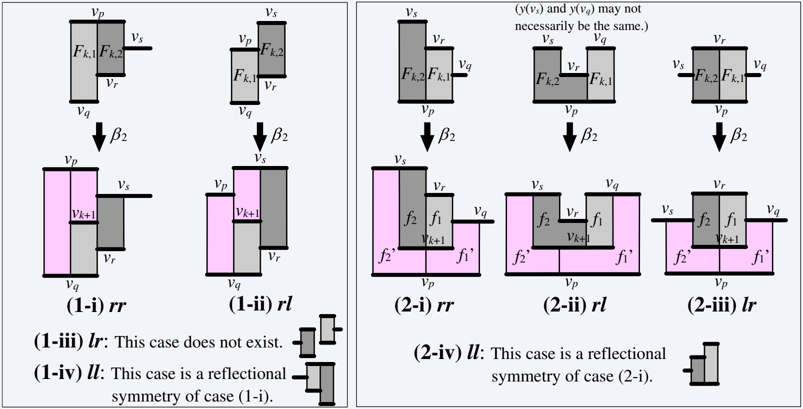
(i) For L-sahpes with $w_b > 1$



(ii) For L-shapes with $w_b = 1$

The β_2 Operation acting at two narrowest L-shapes

- The β_2 operation (the degree-four splitting operation)

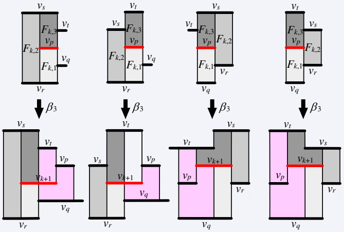


(a) The two input L-shapes do not share the same bottom node segment.

(b) The two input L-shapes share the same bottom node segment.

The β_3 Operation acting at three narrowest L-shapes (1/3)

- The degree-five splitting operation (1/3)

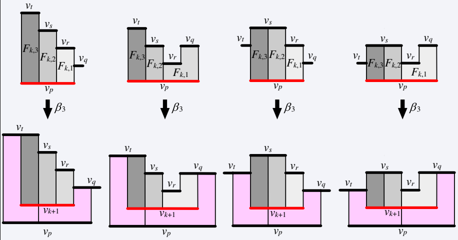


(2v-i-a) (2v-i-b) (2v-ii-a) (2v-ii-b)

(2v-i) ***rr*:** This case has the above two possible subcases.
(2v-ii) ***rl*:** This case has the above two possible subcases.

(2v-iii) ***lr***: This case is a reflectional symmetry of the case (2v-ii).
(2v-iv) ***ll***: This case is a reflectional symmetry of the case (2v-i).

(a) Only two input L-shapes are visible from the down side.



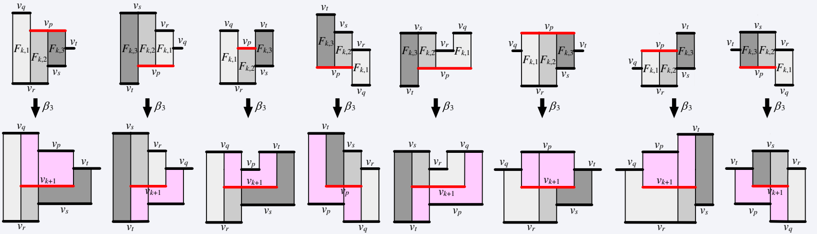
(3-i) ***rrr*** **(3-ii)** ***rrl*** **(3-v)** ***lrr*** **(3-vi)** ***lrl***

(3-iii) ***rlr***: This case is a reflectional symmetry of the case (3-vi).
(3-iv) ***rll***: This case is a reflectional symmetry of the case (3-ii).
(3-vii) ***llr***: This case is a reflectional symmetry of the case (3-v).
(3-viii) ***lll***: This case is a reflectional symmetry of the case (3-i).

(b) All of the three input L-shapes share the same bottom node segment.

The β_3 Operation acting at three narrowest L-shapes (2/3)

- The degree-five splitting operation (2/3)



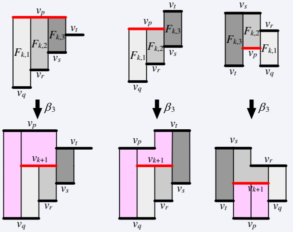
(2-i-a) **(2-i-b)** **(2-ii-a)** **(2-ii-b)** **(2-ii-c)** **(2-v)** ***lrr*** **(2-vi-a)** **(2-vi-b)**

(2-iii) ***rlr***: This case is a reflectional symmetry of the case (2-vi). **(2-iv)** ***rll***: This case is a reflectional symmetry of the case (2-ii). **(2-vii)** ***llr***: This case is a reflectional symmetry of the case (2-v). **(2-viii)** ***lll***: This case is a reflectional symmetry of the case (2-i).

(c) Two of the three input L-shapes share the same bottom node segment.

The β_3 Operation acting at three narrowest L-shapes (3/3)

- The degree-five splitting operation (3/3)



(1-i) ***rrr*** **(1-ii-a)** ***rrl*** **(1-ii-b)** ***rrl***

(1-iii) ***rlr***: This case does not exist.

(1-iv) ***rll***: This case is a reflectional symmetry of case (1-ii).

(1-v) ***lrr***: This case does not exist.

(1-vi) ***lrl***: This case does not exist.

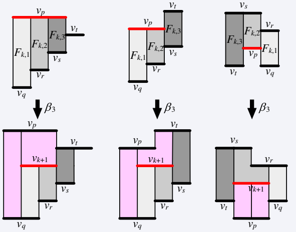
(1-vii) ***llr***: This case does not exist.

(1-viii) ***lll***: This case is a reflectional symmetry of case (1-i).

(d) None of the three input L-shapes shares the same bottom node segment.

The β_3 Operation acting at three narrowest L-shapes (3/3)

- The degree-five splitting operation (3/3)



(1-i) ***rrr*** **(1-ii-a)** ***rrl*** **(1-ii-b)** ***rrl***

(1-iii) ***rlr***: This case does not exist.

(1-iv) ***rll***: This case is a reflectional symmetry of case (1-ii).

(1-v) ***lrr***: This case does not exist.

(1-vi) ***lrl***: This case does not exist.

(1-vii) ***llr***: This case does not exist.

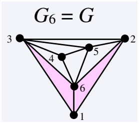
(1-viii) ***lll***: This case is a reflectional symmetry of case (1-i).

(d) None of the three input L-shapes shares the same bottom node segment.

Observation

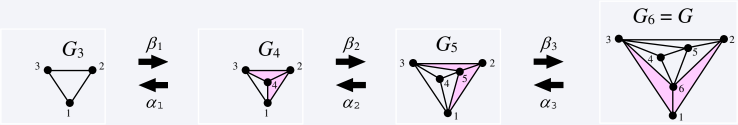
If every inner face in G_k is drawn as an L-shape, then we consider all the possible cases of the β_1, β_2, and β_3 operations.
Furthermore, the bottom node segment of any L-shape rather than input L-shapes is not modified in executing the operation.

Our Drawing Algorithm



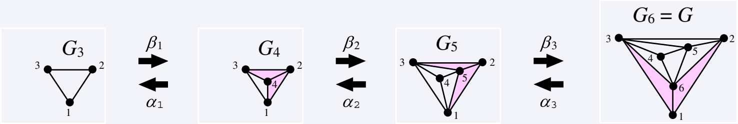
Our Drawing Algorithm

1. Find a constructive ordering of G.

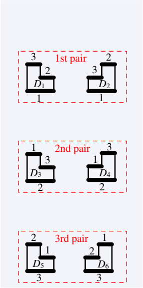


Our Drawing Algorithm

1. Find a constructive ordering of G.

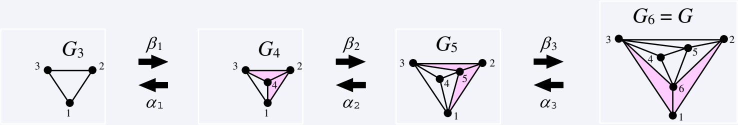


2. Initially, generate all the (six) possible visibility drawings of G_3.

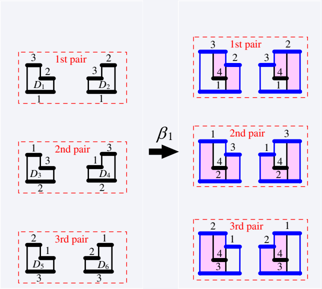


Our Drawing Algorithm

1. Find a constructive ordering of G.

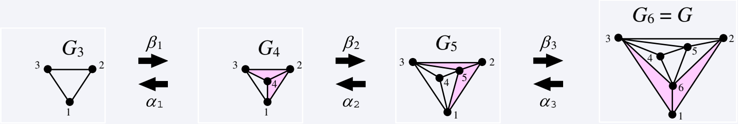


2. Initially, generate all the (six) possible visibility drawings of G_3.
3. For each insertion, use our splitting operations to maintain 6 drawings of G_k; appropriately adjust the drawings of the other faces.

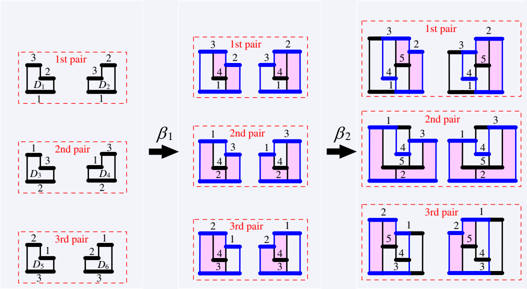


Our Drawing Algorithm

1. Find a constructive ordering of G.

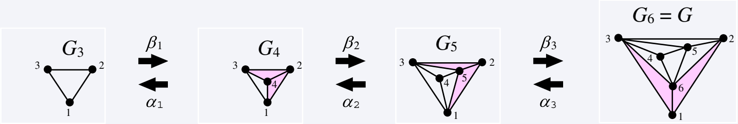


2. Initially, generate all the (six) possible visibility drawings of G_3.
3. For each insertion, use our splitting operations to maintain 6 drawings of G_k; appropriately adjust the drawings of the other faces.

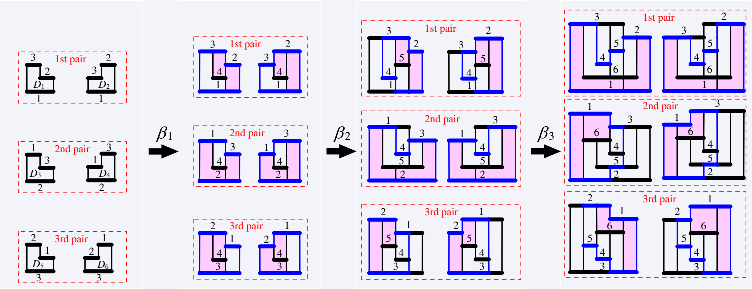


Our Drawing Algorithm

1. Find a constructive ordering of G.

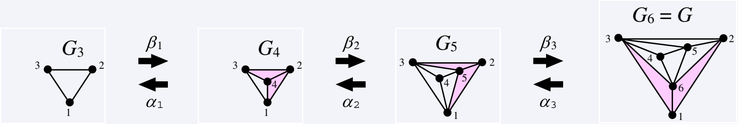


2. Initially, generate all the (six) possible visibility drawings of G_3.
3. For each insertion, use our splitting operations to maintain 6 drawings of G_k; appropriately adjust the drawings of the other faces.

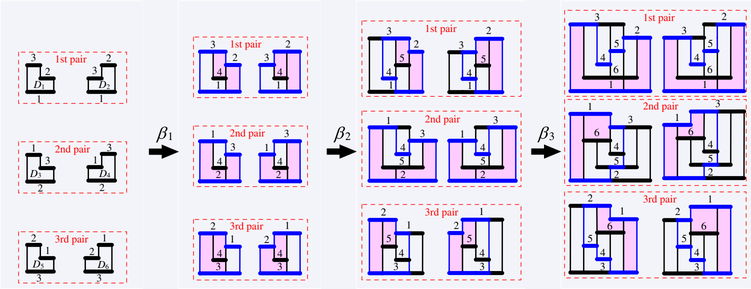


Our Drawing Algorithm

1. Find a constructive ordering of G.



2. Initially, generate all the (six) possible visibility drawings of G_3.
3. For each insertion, use our splitting operations to maintain 6 drawings of G_k; appropriately adjust the drawings of the other faces.

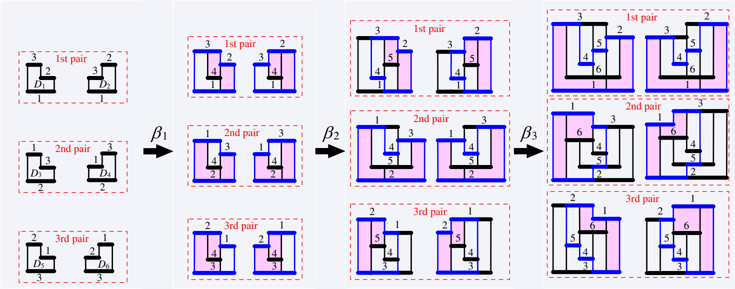


4. Compress the width of every of the six drawings of G_n as much as possible. Output the drawing with the narrowest width.

Six Drawings $\Longrightarrow$ Three Pairs

Observation

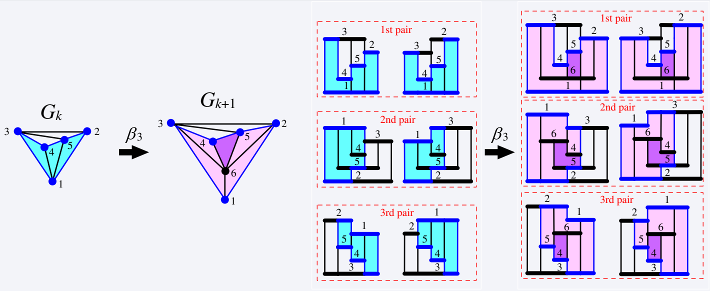
The six drawings of every face F can be classified into three pairs, where each node of F serves as a bottom node segment in each pair.
Furthermore, the input L-shapes of every splitting operation can be classified into three pairs.



Six Drawings $\Longrightarrow$ Three Pairs

Observation

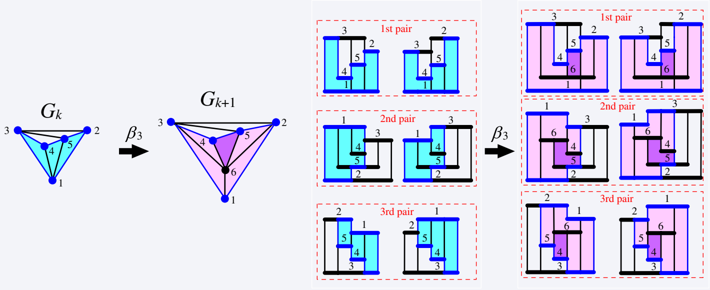
The six drawings of every face F can be classified into three pairs,
where each node of F serves as a bottom node segment in each pair.
Furthermore, the input L-shapes of every splitting operation can be classified into three pairs.



Six Drawings $\Longrightarrow$ Three Pairs

Observation

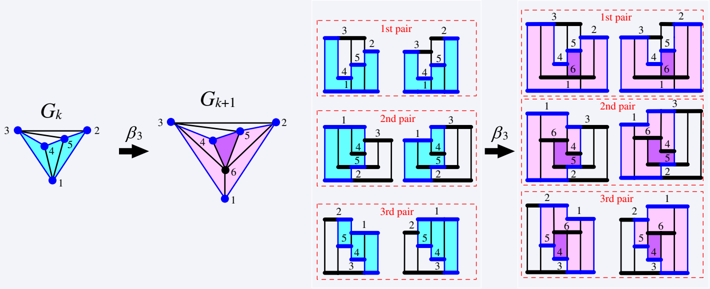
The six drawings of every face F can be classified into three pairs, where each node of F serves as a bottom node segment in each pair. Furthermore, the input L-shapes of every splitting operation can be classified into three pairs.



Six Drawings $\Longrightarrow$ Three Pairs

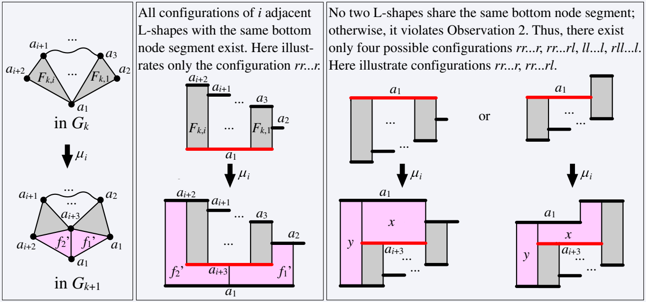
Observation

The six drawings of every face F can be classified into three pairs, where each node of F serves as a bottom node segment in each pair. Furthermore, the input L-shapes of every splitting operation can be classified into three pairs.



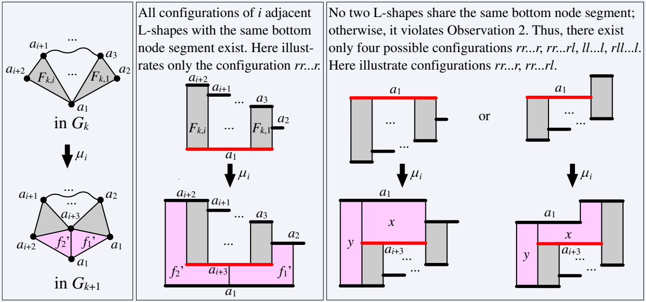
- Note that the two drawings in a pair are almost the same except for the positions of the topmost two node segments, so it suffices to concern one of the two drawings.

U-Shaped Insertion



(a) (b) 1st pair (c) 2nd pair or 3rd pair

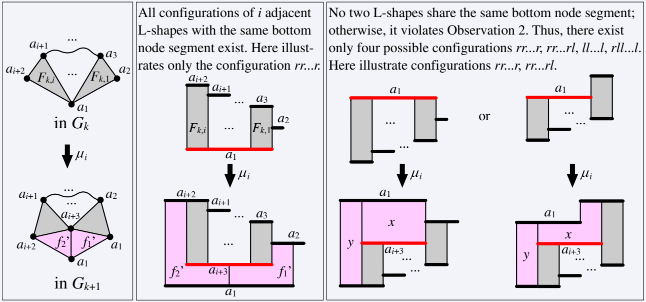
U-Shaped Insertion



(a) (b) 1st pair (c) 2nd pair or 3rd pair

(width + 2) (width + 1) (width + 1)

U-Shaped Insertion

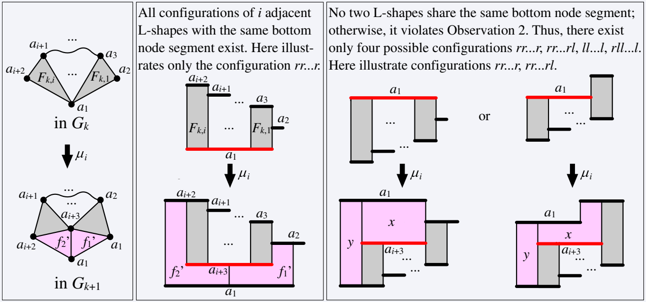


(a) (b) 1st pair (c) 2nd pair or 3rd pair

(width + 2) (width + 1) (width + 1)

$\Longrightarrow$ The sum of widths of six drawings is increased by $2 \times (+2+1+1) = +8$ units.

U-Shaped Insertion

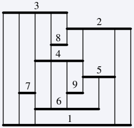


(a) | (b) 1st pair (width + 2) | (c) 2nd pair or 3rd pair (width + 1) (width + 1)

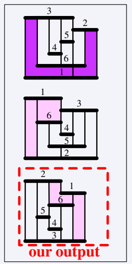
$\Longrightarrow$ The sum of widths of six drawings is increased by $2 \times (+2+1+1) = +8$ units.

Observation

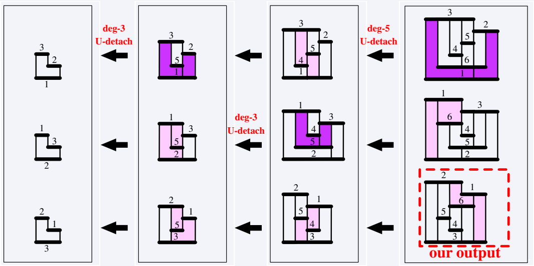
There exists a U-shaped constructive ordering for the drawings produced by our algorithm.



Example



Example



Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

12

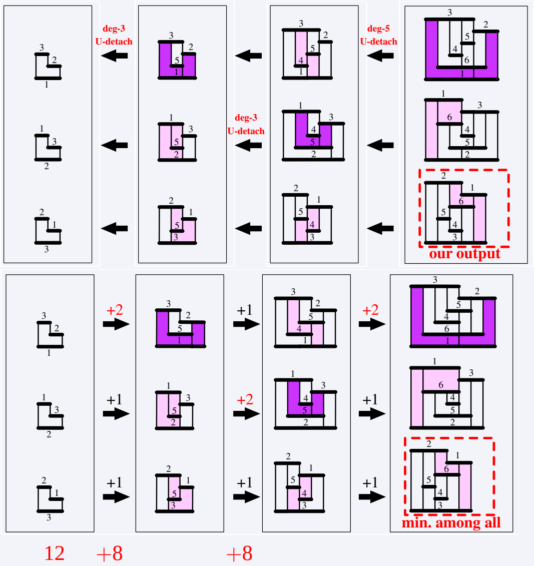
Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

12 +8

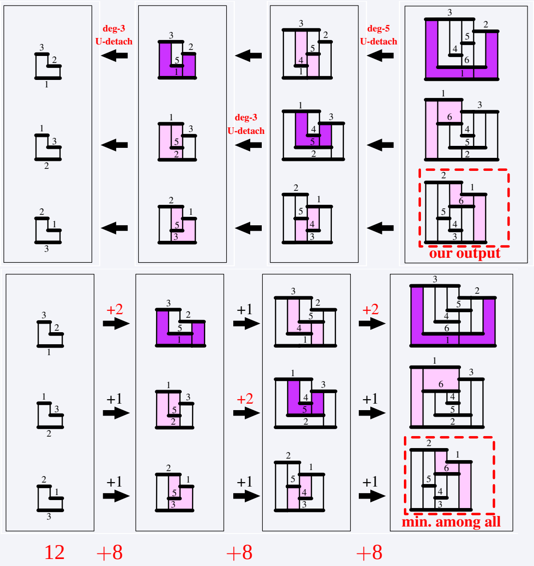
Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.



Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.



Example

- There exists a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

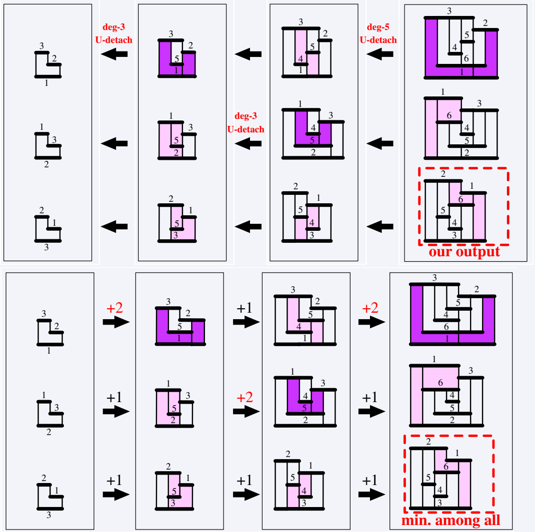
$12 \quad +8 \quad +8 \quad +8$

$= 12 + (n-3) \times 8 = 8n - 12$

Example

- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

- Hence, the drawing with the minimum width must be no wider than the average of $8n-12$, i.e., $\lfloor \frac{8n-12}{6} \rfloor = \lfloor \frac{4n}{3} \rfloor - 2$.



$12 \quad +8 \quad +8 \quad +8$

$= 12 + (n-3) \times 8 = 8n - 12$

Example

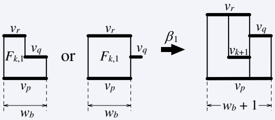
- There **exists** a **certain** U-shaped construct. order. of the final six drawings such that we rebuild visibility drawings according to the ordering in which the final drawing with the same visibility drawing embedding of our output has the minimum width.

- Hence, the drawing with the minimum width must be no wider than the average of $8n-12$, i.e., $\lfloor \frac{8n-12}{6} \rfloor = \lfloor \frac{4n}{3} \rfloor - 2$.

- So our output must be no wider than $\lfloor \frac{4n}{3} \rfloor - 2$.

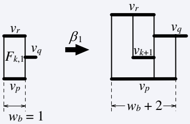
$$12 \quad +8 \quad +8 \quad +8$$
$$= 12 + (n-3) \times 8 = 8n - 12$$

Main Problem: Consecutive Degree-3 U-Shaped Insertions



(i) For L-sahpes with $w_b > 1$

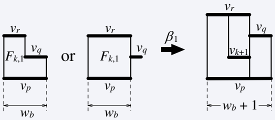
(width + 1)



(ii) For L-shapes with $w_b = 1$

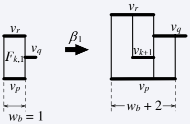
(width + 2)

Main Problem: Consecutive Degree-3 U-Shaped Insertions



(i) For L-sahpes with $w_b > 1$

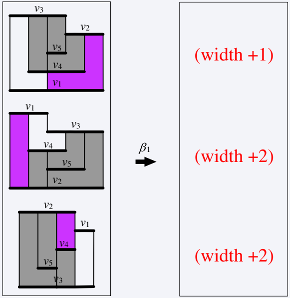
(width + 1)



(ii) For L-shapes with $w_b = 1$

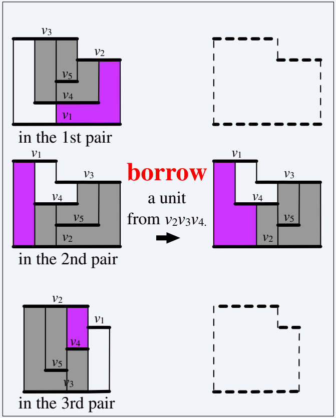
(width + 2)

- For example, if we insert a degree-3 node into the following purple drawings ...

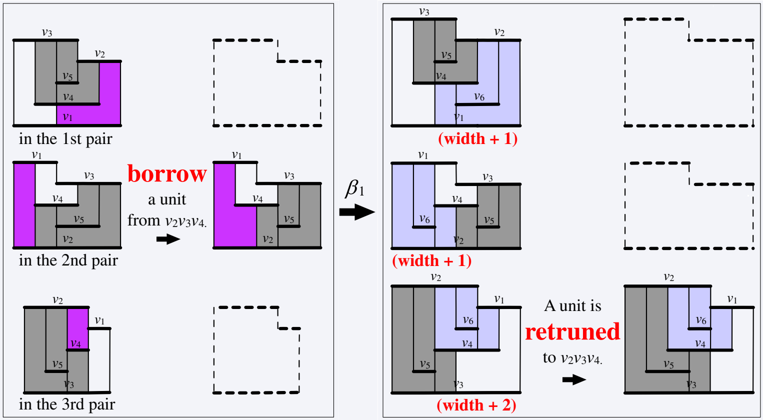


The sum of the widths of the 6 drawings is increased by at least $2 \times (+1 + 2 + 2) = +10$

Borrowing and Returning Widths



Borrowing and Returning Widths



Conclusion

- A linear-time algorithm to find a visibility drawing of a plane triangulation no wider than $\lfloor \frac{4n}{3} \rfloor - 2$ has been proposed in this work.
- Our result improves upon the previously known upper bound $\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$, providing a positive answer to a conjecture about whether an upper bound $\frac{4n}{3} + O(1)$ on the required width can be achieved for an arbitrary plane graph.
- Our result achieves optimality in the upper bound of width because the bound differs from the previously known lower bound $\lfloor \frac{4n}{3} \rfloor - 3$ only by one unit.
- A line of future work is to try to use our technique to find the height-optimal visibility drawing.

Conclusion

- A linear-time algorithm to find a visibility drawing of a plane triangulation no wider than $\lfloor \frac{4n}{3} \rfloor - 2$ has been proposed in this work.
- Our result improves upon the previously known upper bound $\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$, providing a positive answer to a conjecture about whether an upper bound $\frac{4n}{3} + O(1)$ on the required width can be achieved for an arbitrary plane graph.
- Our result achieves optimality in the upper bound of width because the bound differs from the previously known lower bound $\lfloor \frac{4n}{3} \rfloor - 3$ only by one unit.
- A line of future work is to try to use our technique to find the height-optimal visibility drawing.

Conclusion

- A linear-time algorithm to find a visibility drawing of a plane triangulation no wider than $\lfloor \frac{4n}{3} \rfloor - 2$ has been proposed in this work.
- Our result improves upon the previously known upper bound $\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$, providing a positive answer to a conjecture about whether an upper bound $\frac{4n}{3} + O(1)$ on the required width can be achieved for an arbitrary plane graph.
- Our result achieves optimality in the upper bound of width because the bound differs from the previously known lower bound $\lfloor \frac{4n}{3} \rfloor - 3$ only by one unit.
- A line of future work is to try to use our technique to find the height-optimal visibility drawing.

Conclusion

- A linear-time algorithm to find a visibility drawing of a plane triangulation no wider than $\lfloor \frac{4n}{3} \rfloor - 2$ has been proposed in this work.
- Our result improves upon the previously known upper bound $\frac{4n}{3} + 2\lceil \sqrt{n} \rceil$, providing a positive answer to a conjecture about whether an upper bound $\frac{4n}{3} + O(1)$ on the required width can be achieved for an arbitrary plane graph.
- Our result achieves optimality in the upper bound of width because the bound differs from the previously known lower bound $\lfloor \frac{4n}{3} \rfloor - 3$ only by one unit.
- A line of future work is to try to use our technique to find the height-optimal visibility drawing.