Algorithms for Graph Visualization
Flow Methods: Orthogonal Layout

Tamara Mchedlidze
18.01.2018
Orthogonal Layout

- Edges consist of vertical and horizontal segments
- Applied in many areas
Orthogonal Layout

- Edges consist of vertical and horizontal segments
- Applied in many areas

Aesthetic functions?
Orthogonal Layout

- Edges consist of vertical and horizontal segments
- Applied in many areas

Aesthetic functions:
- number of bends
- length of edges
- width, height, area
- monotonicity of edges
- ...
(Planar) Orthogonal Drawings

Three-step approach:  

**Topology – Shape – Metrics**

\[ V = \{v_1, v_2, v_3, v_4\} \]
\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

(Planar) Orthogonal Drawings

Three-step approach: Topology – Shape – Metrics


\[ V = \{v_1, v_2, v_3, v_4\} \]
\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

Reduce Crossings

combinatorial embedding/planarization
(Planar) Orthogonal Drawings

Three-step approach: **Topology – Shape – Metrics**


\[ V = \{v_1, v_2, v_3, v_4\} \]

\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

Reduce Crossings

Bend Minimization

orthogonal representation
(Planar) Orthogonal Drawings

Three-step approach: Topology – Shape – Metrics

$$V = \{v_1, v_2, v_3, v_4\}$$
$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Bend Minimization

Area-minimization

Reduce Crossings

combinatorial embedding/planarization

orthogonal representation

planar orthogonal drawing

Tamassia SIAM J. Comput. 1987
(Planar) Orthogonal Drawings

Three-step approach: Topology – Shape – Metrics

\[ V = \{v_1, v_2, v_3, v_4\} \]
\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

Bend Minimization

orthogonal representation

Reduce Crossings

combinatorial embedding/planarization

planar orthogonal drawing

Area-minimization

Orthogonal Representation

**Given:** planar Graph $G = (V, E)$, set of faces $\mathcal{F}$, outer face $f_0$

**Find:** orthogonale representation $H(G) = \{H(f) | f \in \mathcal{F}\}$

**Face representation** $H(f)$: of $f$ is a clockwise* ordered sequence of edge descriptions $(e, \delta, \alpha)$ with

- $e$ edge of $f$
- $\delta$ is sequence of $\{0, 1\}^*$ ($0 =$ right bend, $1 =$ left bend)
- $\alpha$ is angle $\in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$ between $e$ and next edge $e'$
Orthogonal Representation: Example

\[ H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2})) \]

\[ H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi)) \]

\[ H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2})) \]

Combinatorial “drawing” of \( H(G) \)?
Orthogonal Representation: Example

\[ H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2})) \]
\[ H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi)) \]
\[ H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2})) \]
Orthogonal Representation: Example

\[ H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2})) \]
\[ H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi)) \]
\[ H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2})) \]

is \( f_0 \) listed wrongly!?
Orthogonal Representation: Example

\[ H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2})) \]

\[ H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi)) \]

\[ H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2})) \]
Orthogonal Representation: Example

\( H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2})) \)

\( H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi)) \)

\( H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2})) \)
Orthogonal Representation: Example

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$

Concrete coordinates are not fixed yet!
Correctness of an Orthogonal Representation

I1) $H(G)$ corresponds to $\mathcal{F}, f_0$
Correctness of an Orthogonal Representation

1) \( H(G) \) corresponds to \( \mathcal{F}, f_0 \)

2) for an edge \( \{u, v\} \) shared by faces \( f \) and \( g \) with 
\( ((u, v), \delta_1, \alpha_1) \in H(f) \) and \( ((v, u), \delta_2, \alpha_2) \in H(g) \)
sequence \( \delta_1 \) is reversed and inverted \( \delta_2 \)
Correctness of an Orthogonal Representation

I1) \( H(G) \) corresponds to \( \mathcal{F}, f_0 \)

I2) for an edge \( \{u, v\} \) shared by faces \( f \) and \( g \) with 
\[ ((u, v), \delta_1, \alpha_1) \in H(f) \text{ and } ((v, u), \delta_2, \alpha_2) \in H(g) \]
sequence \( \delta_1 \) is reversed and inverted \( \delta_2 \)

I3) Let \( |\delta|_0 \) (resp. \( |\delta|_1 \)) be the number of zeros (resp. ones) in \( \delta \) and \( r = (e, \delta, \alpha) \). For \( C(r) := |\delta|_0 - |\delta|_1 + 2 - 2\alpha/\pi \) holds that:
\[ \sum_{r \in H(f)} C(r) = 4 \text{ for } f \neq f_0 \text{ and } \sum_{r \in H(f_0)} C(r) = -4 \]
Correctness of an Orthogonal Representation

I1) \( H(G) \) corresponds to \( \mathcal{F}, f_0 \)

I2) for an edge \( \{u, v\} \) shared by faces \( f \) and \( g \) with \((u, v), \delta_1, \alpha_1) \in H(f) \) and \((v, u), \delta_2, \alpha_2) \in H(g)\)

sequence \( \delta_1 \) is reversed and inverted \( \delta_2 \)

I3) Let \( |\delta|_0 \) (resp. \( |\delta|_1 \)) be the number of zeros (resp. ones) in \( \delta \) and \( r = (e, \delta, \alpha) \). For \( C(r) := |\delta|_0 - |\delta|_1 + 2 - 2\alpha/\pi \) holds that: 

\[
\sum_{r \in H(f)} C(r) = 4 \quad \text{for} \quad f \neq f_0 \quad \text{and} \quad \sum_{r \in H(f_0)} C(r) = -4
\]

I4) For each node \( v \) the summ of incident angles is \( 2\pi \)
Correctness of an Orthogonal Representation

I1) \(H(G)\) corresponds to \(F, f_0\)

I2) for an edge \(\{u, v\}\) shared by faces \(f\) and \(g\) with 
\(((u, v), \delta_1, \alpha_1) \in H(f)\) and \(((v, u), \delta_2, \alpha_2) \in H(g)\)
sequence \(\delta_1\) is reversed and inverted \(\delta_2\)

I3) Let \(|\delta|_0\) (resp. \(|\delta|_1\)) be the number of zeros (resp. ones) in \(\delta\)
and \(r = (e, \delta, \alpha)\). For \(C(r) := |\delta|_0 - |\delta|_1 + 2 - 2\alpha/\pi\) holds
that: \(\sum_{r \in H(f)} C(r) = 4\) for \(f \neq f_0\) and \(\sum_{r \in H(f_0)} C(r) = -4\)

I4) For each node \(v\) the summ of incident angles is \(2\pi\)

Pair, think and share:
What does the condition (H3) mean intuitively?
Bend Minization with Given Embedding

**Problem: Geometric Bend Minimization**

Given:
- planar Graph $G = (V, E)$ with maximum degree 4
- combinatorial embedding $\mathcal{F}$ and outer face $f_0$

Find: orthogonal drawing with minimum number of bends that preserves the embedding
Bend Minimization with Given Embedding

**Problem: Geometric Bend Minimization**

Given: • planar Graph $G = (V, E)$ with maximum degree 4
  • combinatorial embedding $\mathcal{F}$ and outer face $f_0$

Find: orthogonal drawing with minimum number of bends that preserves the embedding

compare with the following variation

**Problem Combinatorial Bend Minimization**

Given: • planar Graph $G = (V, E)$ with maximum degree 4
  • combinatorial embedding $\mathcal{F}$ and outer face $f_0$

Find: *orthogonal representation* $H(G)$ with minimum number of bends that preserves the embedding
Combinatorial Bend Minimization

**Problem Combinatorial Bend Minimization**

Given:  
- Graph $G = (V, E)$ with maximum degree 4  
- combinatorial embedding $\mathcal{F}$ and outer face $f_0$

Find: **orthogonal representation** $H(G)$ with minimum number of bends that preserves the embedding
Combinatorial Bend Minimization

Problem Combinatorial Bend Minimization

Given: • Graph $G = (V, E)$ with maximum degree 4
  • combinatorial embedding $\mathcal{F}$ and outer face $f_0$

Find: **orthogonal representation** $H(G)$ with minimum number of bends that preserves the embedding

Idea: formulate as a network flow problem

• a unit of flow represents an angle $\pi/2$
• flow from vertices to faces represents the angles at the vertices
• flow between adjacent faces represent the bends at the edges
Reminder: \( s-t \) Flow Network

**Flow network** \((D = (V, A); s, t; c)\) with
- directed graph \(D = (V, A)\)
- Edge capacity \(c : A \rightarrow \mathbb{R}_0^+\)
- Source \(s \in V\), Sink \(t \in V\)

A function \(X : A \rightarrow \mathbb{R}_0^+\) is called **\( s-t \)-flow**, if:

\[
0 \leq X(u, v) \leq c(u, v) \quad \forall (u, v) \in A
\]

\[
\sum_{(u,v)\in A} X(u,v) - \sum_{(v,u)\in A} X(v,u) = 0 \quad \forall u \in V \setminus \{s, t\}
\]
Reminder: General Flow Network

**Flow network** \((D = (V, A); \ell; u; b)\) with

- **directed graph** \(D = (V, A)\)
- **edge lower bound** \(\ell : A \rightarrow \mathbb{R}_0^+\)
- **edge capacity** \(c : A \rightarrow \mathbb{R}_0^+\)
- **node production/consumption** \(b : V \rightarrow \mathbb{R}\) with
  \[\sum_{i \in V} b(i) = 0\]

An assignment \(X : A \rightarrow \mathbb{R}_0^+\) is called **valid flow**, if:

\[
\ell(u, v) \leq X(u, v) \leq c(u, v) \quad \forall (u, v) \in A \tag{3}
\]

\[
\sum_{(u,v) \in A} X(u,v) - \sum_{(v,u) \in A} X(v,u) = b(u) \quad \forall u \in V \tag{4}
\]
Problems for Flow Networks

(A) Valid Flow:
Find a Valid Flow $X : A \rightarrow \mathbb{R}_0^+$, such that.

- Lower bounds and capacities $\ell(e), u(e)$ are respected (inequalities (3))
- Consumption/production is $b(v)$ satisfied (inequality (4))
Problems for Flow Networks

(A) Valid Flow:
Find a Valid Flow $X : A \rightarrow \mathbb{R}_0^+$, such that.

- Lower bounds and capacities $\ell(e), u(e)$ are respected (inequalities (3))
- Consumption/production is $b(v)$ satisfied (inequality (4))

Additionally provided: Cost function $\text{cost} : A \rightarrow \mathbb{R}_0^+$

Def: $\text{cost}(X) := \sum_{(u,v) \in A} \text{cost}(u,v) \cdot X(u, v)$
Problems for Flow Networks

(A) Valid Flow:
Find a Valid Flow $X : A \to \mathbb{R}_0^+$, such that.
- Lower bounds and capacities $l(e), u(e)$ are respected (inequalities (3))
- Consumption/production is $b(v)$ satisfied (inequality (4))

Additionally provided: **Cost function** $\text{cost} : A \to \mathbb{R}_0^+$

Def: $\text{cost}(X) := \sum_{(u,v) \in A} \text{cost}(u, v) \cdot X(u, v)$

(B) Minimum Cost Flow
Find a valid flow $X : A \to \mathbb{R}_0^+$, that minimizes cost function $\text{cost}(X)$ (over all valid flows)
Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup F, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times F \mid v \text{ incident to } f\} \cup \{(f, g) \in F \times F \mid f, g \text{ adjacent through edge } e\}$
Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
- $b(v) = 4 \quad \forall v \in V$
Flow Network for Bend Minimization

Flow Network \( N(G) = ((V \cup F, A); \ell; c; b; \text{cost}) \)

- \( A = \{(v, f) \in V \times F \mid v \text{ incident to } f\} \cup \{(f, g) \in F \times F \mid f, g \text{ adjacent through edge } e\} \)
- \( b(v) = 4 \quad \forall v \in V \)
- \( b(f) = -2(d_G(f) - 2) \quad \forall f \in F \setminus \{f_0\} \)
- \( b(f_0) = -2(d_G(f_0) + 2) \)
Flow Network for Bend Minimization

Flow Network $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
- $b(v) = 4 \quad \forall v \in V$
- $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
- $b(f_0) = -2(d_G(f_0) + 2)$

$\Rightarrow \sum w b(w) = 0$
Flow Network for Bend Minimization

Flow Network \( N(G) = ((V \cup F, A); \ell; c; b; \text{cost}) \)

- \( A = \{(v, f) \in V \times F \mid v \text{ incident to } f\} \cup \{(f, g) \in F \times F \mid f, g \text{ adjacent through edge } e\} \)
- \( b(v) = 4 \quad \forall v \in V \)
- \( b(f) = -2(d_G(f) - 2) \quad \forall f \in F \setminus \{f_0\} \)
- \( b(f_0) = -2(d_G(f_0) + 2) \)

\[ \Rightarrow \sum_w b(w) = 0 \]  (Euler)
Flow Network for Bend Minimization

Flow Network \( N(G) = ((V \cup F, A); \ell; c; b; \text{cost}) \)

- \( A = \{(v, f) \in V \times F \mid v \text{ incident to } f\} \cup \{(f, g) \in F \times F \mid f, g \text{ adjacent through edge } e\} \)
- \( b(v) = 4 \quad \forall v \in V \)
- \( b(f) = -2(d_G(f) - 2) \quad \forall f \in F \setminus \{f_0\} \)
- \( b(f_0) = -2(d_G(f_0) + 2) \)

\[ \Rightarrow \sum_w b(w) = 0 \quad (\text{Euler}) \]

\( \forall (f, g) \in A, f, g \in F \)
\( \ell(f, g) := 0 \leq X(f, g) \leq \infty =: c(f, g) \)
\( \text{cost}(f, g) = 1 \)

\( \forall (v, f) \in A, v \in V, f \in F \)
\( \ell(v, f) := 1 \leq X(v, f) \leq 4 =: c(v, f) \)
\( \text{cost}(v, f) = 0 \)
Example Flow Network
Example Flow Network

\[ f_0 \]

\[ v_1 \quad v_2 \quad v_3 \]
\[ e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \]

\( e_1 \)

\( e_2 \)

\( e_3 \)

\( e_4 \)

\( e_5 \)

\( e_6 \)

\( v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \)

\( V \)

\( F \)
Example Flow Network

\( f_0 \)

\[ V \times \mathcal{F} \supseteq \]

V
F

\( f_1 \)
\( f_2 \)
\( e_1 \)
\( e_2 \)
\( e_3 \)
\( e_4 \)
\( e_5 \)
\( e_6 \)
Example Flow Network

\[ V \times F \supseteq \]

\[ F \times F \supseteq \]
Example Flow Network
Example Flow Network

\[
\begin{align*}
V & \times F \supseteq \\
\mathcal{F} & \times \mathcal{F} \supseteq
\end{align*}
\]
Example Flow Network

\( V \times \mathcal{F} \supseteq \)

\( \mathcal{F} \times \mathcal{F} \supseteq \)
Example Flow Network
Example Flow Network

cost = 1
bend!
outside

\(f_0\)
-14

\(f_1\)

\(f_2\)

\(e_1\)

\(e_2\)

\(e_3\)

\(e_4\)

\(e_5\)

\(e_6\)

\(v_1\)

\(v_2\)

\(v_3\)

\(v_4\)

\(v_5\)

\(v\)

\(F\)

\(V\)

\(\ell/u/c\)

\(1/4/0\)

\(V \times F \supseteq\)

\(0/\infty/1\)

\(F \times F \supseteq\)
Main Statement

**Thm 1:** A planar embedded graph \((G, \mathcal{F}, f_0)\) has a valid orthogonal description \(H(G)\) with \(k\) bends iff the flow network \(N(G)\) has a valid flow \(X\) with cost \(k\).
Main Statement

Thm 1: A planar embedded graph \((G, \mathcal{F}, f_0)\) has a valid orthogonal description \(H(G)\) with \(k\) bends iff the flow network \(N(G)\) has a valid flow \(X\) with cost \(k\).

Proof:
\[\Leftarrow\text{ Given flow } X \text{ in } N(G) \text{ with cost } k\]
Construct orthogonal representation \(H(G)\) with \(k\) bends
Main Statement

**Thm 1:** A planar embedded graph \((G, \mathcal{F}, f_0)\) has a valid orthogonal description \(H(G)\) with \(k\) bends iff the flow network \(N(G)\) has a valid flow \(X\) with cost \(k\).

**Proof:**

\(\Leftarrow\) Given flow \(X\) in \(N(G)\) with cost \(k\)

- Construct orthogonal representation \(H(G)\) with \(k\) bends
- transform from to orthogonal description
- show properties (H1)–(H4)
Main Statement

**Thm 1:** A planar embedded graph $(G, \mathcal{F}, f_0)$ has a valid orthogonal description $H(G)$ with $k$ bends iff the flow network $N(G)$ has a valid flow $X$ with cost $k$.

**Proof:**

$\Leftarrow$ Given flow $X$ in $N(G)$ with cost $k$

Construct orthogonal representation $H(G)$ with $k$ bends

• transform from to orthogonal description
• show properties (H1)–(H4)

$\Rightarrow$ Given an orthogonal description $H(G)$ with $k$ bends

Construct flow $X$ in $N(G)$ with cost $k$
Main Statement

**Thm 1:** A planar embedded graph \((G, \mathcal{F}, f_0)\) has a valid orthogonal description \(H(G)\) with \(k\) bends iff the flow network \(N(G)\) has a valid flow \(X\) with cost \(k\).

**Proof:**

\(\Leftarrow\) Given flow \(X\) in \(N(G)\) with cost \(k\)
- Construct orthogonal representation \(H(G)\) with \(k\) bends
  - transform from to orthogonal description
  - show properties (H1)–(H4)

\(\Rightarrow\) Given an orthogonal description \(H(G)\) with \(k\) bends
- Construct flow \(X\) in \(N(G)\) with cost \(k\)
  - define assignement \(X : A \rightarrow \mathbb{R}_0^+\)
  - show that \(X\) is a valid flow and has cost \(k\)
Summary of Bend Minimization

• From Theorem 1 it follows that a combinatorial orthogonal bend minimization problem for embedded planar graphs can be solved using an algorithm for Min-Cost-Flow Problem.
Summary of Bend Minimization

• From Theorem 1 it follows that a combinatorial orthogonal bend minimization problem for embedded planar graphs can be solved using an algorithm for Min-Cost-Flow Problem.

• This special flow problem for planar network \( N(G) \) can be solved in \( O(n^{3/2}) \) time.

[Cornešen, Karrenbauer GD 2011]
Summary of Bend Minimization

- From Theorem 1 it follows that a combinatorial orthogonal bend minimization problem for embedded planar graphs can be solved using an algorithm for Min-Cost-Flow Problem.
- This special flow problem for planar network $N(G)$ can be solved in $O(n^{3/2})$ time. [Cornelsen, Karrenbauer GD 2011]
Summary of Bend Minimization

• From Theorem 1 it follows that a combinatorial orthogonal bend minimization problem for embedded planar graphs can be solved using an algorithm for Min-Cost-Flow Problem.

• This special flow problem for planar network \( N(G) \) can be solved in \( O(n^{3/2}) \) time.

  [Cornelsen, Karrenbauer GD 2011]

• Bend minimization without a given combinatorial embedding is an NP-hard problem.

(Planare) Orthogonale Zeichnungen

Three-step approach: Topology – Shape – Metrics

\[ V = \{v_1, v_2, v_3, v_4\} \]
\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

Reduce Crossings

combinatorial embedding/planarization

Bend Minimization

orthogonal representation

Area-minimization

planar orthogonal drawing

(Planare) Orthogonale Zeichnungen

Three-step approach:  

**Topology – Shape – Metrics**

\[ V = \{v_1, v_2, v_3, v_4\} \]
\[ E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\} \]

- Bend Minimization
- Area-minimization
- Reduce Crossings
- Orthogonal representation
- Combinatorial embedding/planarization

Compaction

Problem Compaction

Given: • planar graph $G = (V, E)$ with maximum degree 4
  • orthogonal representation $H(G)$

Find: compact orthogonal layout of $G$ that realizes $H(G)$
Compaction

Problem Compaction

Given: • planar graph $G = (V, E)$ with maximum degree 4
  • orthogonal representation $H(G)$

Find: compact orthogonal layout of $G$ that realizes $H(G)$

Special Case: all faces are rectangles

→ Guarantees possible
  • minimum total edge length
  • minimum area
Compaction

Problem Compaction

Given: • planar graph $G = (V, E)$ with maximum degree 4
  • orthogonal representation $H(G)$

Find: compact orthogonal layout of $G$ that realizes $H(G)$

Special Case: all faces are rectangles

→ Guarantees possible • minimum total edge length
  • minimum area

Properties:
• bends only on the outer face
• opposite sides of a face have the same length
Compaction

Problem Compaction

Given: • planar graph \( G = (V, E) \) with maximum degree 4
  • orthogonal representation \( H(G) \)

Find: compact orthogonal layout of \( G \) that realizes \( H(G) \)

Special Case: all faces are rectangles

→ Guarantees possible • minimum total edge length
  • minimum area

Properties:
  • bends only on the outer face
  • opposite sides of a face have the same length

Think for a minute:
Why the two properties hold?

1 min
Compaction

Problem Compaction

Given: • planar graph $G = (V, E)$ with maximum degree 4
  • orthogonal representation $H(G)$

Find: compact orthogonal layout of $G$ that realizes $H(G)$

Special Case: all faces are rectangles

→ Guarantees possible
  • minimum total edge length
  • minimum area

Properties:
  • bends only on the outer face
  • opposite sides of a face have the same length

We will formulate a flow network for (horizontal) compaction
Flow Network for Edge Length Computation

Def: Flow Network $\mathcal{N}_{\text{hor}} = ((W_{\text{hor}}, A_{\text{hor}}); \ell; u; b; \text{cost})$

- $W_{\text{hor}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{hor}} = \{(f, g) \mid f, g \text{ share a horizontal segment and } f \text{ lies below } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $u(a) = \infty \quad \forall a \in A_{\text{hor}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $b(f) = 0 \quad \forall f \in W_{\text{hor}}$
Flow Network for Edge Length Computation

**Def:** Flow Network $N_{\text{hor}} = ((W_{\text{hor}}, A_{\text{hor}}); \ell; u; b; \text{cost})$

- $W_{\text{hor}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{hor}} = \{(f, g) \mid f, g \text{ share a horizontal segment and } f \text{ lies below } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $u(a) = \infty \quad \forall a \in A_{\text{hor}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $b(f) = 0 \quad \forall f \in W_{\text{hor}}$

$s$ and $t$ represent lower and upper side of $f_0$
Flow Network for Edge Length Computation

Def: Flow Network $\mathcal{N}_{\text{ver}} = ((W_{\text{ver}}, A_{\text{ver}}); \ell; u; b; \text{cost})$

- $W_{\text{ver}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{ver}} = \{(f, g) \mid f, g \text{ share a vertical segment and } f \text{ lies to the left of } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{ver}}$
- $u(a) = \infty \quad \forall a \in A_{\text{ver}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{ver}}$
- $b(f) = 0 \quad \forall f \in W_{\text{ver}}$
Flow Network for Edge Length Computation

**Def:** Flow Network $N_{ver} = ((W_{ver}, A_{ver}); \ell; u; b; cost)$

- $W_{ver} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{ver} = \{(f, g) \mid f, g \text{ share a vertical segment and } f \text{ lies to the left of } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{ver}$
- $u(a) = \infty \quad \forall a \in A_{ver}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{ver}$
- $b(f) = 0 \quad \forall f \in W_{ver}$

What values of the drawing represent the following?

- $|x_{\text{hor}}(t, s)|$ and $|x_{\text{ver}}(t, s)|$?
- $\sum_{a \in A_{\text{hor}}} x_{\text{hor}}(a) + \sum_{a \in A_{\text{ver}}} x_{\text{ver}}(a)$
Optimal Layout

Thm 2: Integer flows $x_{\text{hor}}$ and $x_{\text{ver}}$ in $N_{\text{hor}}$ and $N_{\text{ver}}$ with minimum cost induce valid orthogonal layout with minimum total edge length. The layout can be computed in $O(n^{3/2}) \star$ time.
Faster Flow Computation

- construct the duals $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$ of $N_{\text{hor}}$ and $N_{\text{ver}}$
- topological numbering $T_{\text{hor}}$ and $T_{\text{ver}}$ of $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$
- for edge $(f, g)$ of $N_{\text{hor}}$ set flow
  $x_{\text{hor}}(f, g) = T_{\text{hor}}(b) - T_{\text{hor}}(a)$, where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f, g)$, similar for $x_{\text{ver}}$
- easy to see that the constructed assignments $x_{\text{hor}}$, $x_{\text{ver}}$ have minimum value
Faster Flow Computation

- construct the duals $N_{\text{hor}}^*$ and $N_{\text{ver}}^*$ of $N_{\text{hor}}$ and $N_{\text{ver}}$
- topological numbering $T_{\text{hor}}$ and $T_{\text{ver}}$ of $N_{\text{hor}}^*$ and $N_{\text{ver}}^*$
- for edge $(f,g)$ of $N_{\text{hor}}$ set flow $x_{\text{hor}}(f,g) = T_{\text{hor}}(b) - T_{\text{hor}}(a)$, where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f,g)$, similar for $x_{\text{ver}}$
- easy to see that the constructed assignments $x_{\text{hor}}$, $x_{\text{ver}}$ have minimum value
**Faster Flow Computation**

- construct the duals $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$ of $N_{\text{hor}}$ and $N_{\text{ver}}$
- topological numbering $T_{\text{hor}}$ and $T_{\text{ver}}$ of $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$
- for edge $(f, g)$ of $N_{\text{hor}}$ set flow
  
  $$x_{\text{hor}}(f, g) = T_{\text{hor}}(b) - T_{\text{hor}}(a),$$
  where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f, g)$, similar for $x_{\text{ver}}$
- easy to see that the constructed assignements $x_{\text{hor}}, x_{\text{ver}}$ have minimum value
Faster Flow Computation

- construct the duals $N_{\text{hor}}^*$ and $N_{\text{ver}}^*$ of $N_{\text{hor}}$ and $N_{\text{ver}}$
- topological numbering $T_{\text{hor}}$ and $T_{\text{ver}}$ of $N_{\text{hor}}^*$ and $N_{\text{ver}}^*$
- for edge $(f, g)$ of $N_{\text{hor}}$ set flow $x_{\text{hor}}(f, g) = T_{\text{hor}}(b) - T_{\text{hor}}(a)$, where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f, g)$, similar for $x_{\text{ver}}$
- easy to see that the constructed assignments $x_{\text{hor}}, x_{\text{ver}}$ have minimum value
Faster Flow Computation

- construct the duals $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$ of $N_{\text{hor}}$ and $N_{\text{ver}}$
- topologial numbering $T_{\text{hor}}$ and $T_{\text{ver}}$ of $N^*_{\text{hor}}$ and $N^*_{\text{ver}}$
- for edge $(f, g)$ of $N_{\text{hor}}$ set flow $x_{\text{hor}}(f, g) = T_{\text{hor}}(b) - T_{\text{hor}}(a)$, where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f, g)$, similar for $x_{\text{ver}}$
- easy to see that the constructed assignments $x_{\text{hor}}$, $x_{\text{ver}}$ have minumun value
Faster Flow Computation

- construct the duals $N_{hor}^*$ and $N_{ver}^*$ of $N_{hor}$ and $N_{ver}$
- topological numbering $T_{hor}$ and $T_{ver}$ of $N_{hor}^*$ and $N_{ver}^*$
- for edge $(f, g)$ of $N_{hor}$ set flow $x_{hor}(f, g) = T_{hor}(b) - T_{hor}(a)$, where $b$ is dual vertex on the left and $b$ is dual vertex on the right of $(f, g)$, similar for $x_{ver}$
- easy to see that the constructed assignements $x_{hor}, x_{ver}$ have minimum value
Faster Flow Computation

• This approach finds minimum width, height, area, but does not guarantee minimum total edge length
• Time complexity $O(n)$
Faster Flow Computation

- This approach finds minimum width, height, area, but does not guarantees minimum total edge length
- Time complexity $O(n)$

But what we do if not all faces are rectangles?
Refinement of \((G, H)\) – Inner Face

\[ f \]
Refinement of \((G, H)\) – Inner Face

Dummy nodes for bends: ○
Refinement of \((G, H)\) – Inner Face

Corner node placements:

- \(e\)
- \(e_0\)
- \(e_1\)
- \(e_2\)
- \(e_3\)
- \(e_4\)
- \(e_5\)
- \(e_6\)
- \(e_7\)
- \(e_8\)
- \(e_9\)
- \(e_{10}\)
- \(e_{11}\)
- \(e_{12}\)
- \(e_{13}\)
- \(e_{14}\)
- \(e_{15}\)

Dummy nodes for bends: •
Refinement of \((G, H)\) – Inner Face

\[ \text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend} 
\end{cases} \]
Refinement of $(G, H)$ – Inner Face

- $e_{14}$
- $e_{15}$
- $e_0$, $e_1$, $e_2$, $e_3$, $e_4$, $e_6$, $e_7$, $e_8$, $e_9$, $e_{10}$, $e_{11}$, $e_{12}$, $e_{13}$

$$\sum \text{turn}(e) = 1$$

- Dummy nodes for bends:

$$\text{front}(e_0): \text{edge following } e_0 \text{ such that for the edges in between}$$

$$\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}$$
Refinement of \((G, H)\) – Inner Face

front\((e_0)\): edge following \(e_0\) such that for the edges in between \(\sum \text{turn}(e) = 1\)

Dummy nodes for bends: ○

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of $(G, H)$ – Inner Face

Dummy nodes for bends:

\[ \text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend} 
\end{cases} \]
Refinement of \((G, H) – \text{Inner Face}\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend} 
\end{cases}
\]

Dummy nodes for bends: •
Refinement of $(G, H) –$ Inner Face

$$\text{turn}(e) = \begin{cases} 1 & \text{left bend} \\ 0 & \text{no bend} \\ -1 & \text{right bend} \end{cases}$$

Dummy nodes for bends: ○
Refinement of \((G, H) – \text{Inner Face}\)

Dummy nodes for bends: ○

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Inner Face

Dummy nodes for bends: ○

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H) – \text{Inner Face}\)

Dummy nodes for bends: ○

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H) – \text{Inner Face}\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend} 
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face
Refinement of \((G, H)\) – Outer Face

- front\((e)\) may be undefined

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H) – \text{Outer Face}\)

- \text{front}(e)\ may\ be\ undefined
- \text{when } \sum \text{turn}(e) < 1 \text{ for the complete turn around } f_0,\ project\ on\ \mathbb{R}

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

\[
\begin{aligned}
\text{front}(e) &\text{ may be undefined} \\
\text{when } \sum \text{turn}(e) &< 1 \\
\text{for the complete turn around } f_0, \text{ project on } R
\end{aligned}
\]

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend} 
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\)
for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \text{front}(e)\ may\ be\ undefined
- \text{when } \sum \text{turn}(e) < 1 \text{ for the complete turn around } f_0,\ project\ on\ \mathbb{R}

\text{turn}(e) = \begin{cases} 1 & \text{left bend} \\ 0 & \text{no bend} \\ -1 & \text{right bend} \end{cases}
Refinement of \((G, H)\) – Outer Face

- **front\((e)\)** may be undefined
- when \(\sum_{e} \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

\[
\text{turn}(e) = \begin{cases} 
1 & \text{left bend} \\
0 & \text{no bend} \\
-1 & \text{right bend}
\end{cases}
\]
Refinement of \((G, H)\) – Outer Face

- \(\text{front}(e)\) may be undefined
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

all faces are rectangles → apply flow network
Refinement of \((G, H)\) – Outer Face

Has minimum area?
Refinement of \((G, H)\) – Outer Face

Has minimum area? NO!
Refinement of \((G, H)\) – Outer Face

Has minimum area? NO!

Area Minimization with a given orthogonal representation is an NP-hard problem!
Summary

• An orthogonal representation with minimum number of bends can be found in $O(n^{3/2})$ time
• Given an orthogonal representation a layout with minimum area and total edge length is achievable for the case of rectangular faces
• In case of non rectangular faces, reduce the problem to rectangular case. The resulting area is not minimum.
Summary

• An orthogonal representation with minimum number of bends can be found in \( O(n^{3/2}) \) time
• Given an orthogonal representation a layout with minimum area and total edge length is achievable for the case of rectangular faces
• In case of non rectangular faces, reduce the problem to rectangular case. The resulting area is not minimum.
• Area minimization with a given orthogonal representation is an NP-hard problem.

[Patrignany CGTA 2001]
Summary

• An orthogonal representation with minimum number of bends can be found in $O(n^{3/2})$ time.
• Given an orthogonal representation a layout with minimum area and total edge length is achievable for the case of rectangular faces.
• In case of non-rectangular faces, reduce the problem to rectangular case. The resulting area is not minimum.
• Area minimization with a given orthogonal representation is an NP-hard problem.
• Solvable with an integer linear program (ILP)

[Patrignany CGTA 2001]

[Klau, Mutzel IPCO 1999]
Summary

• An orthogonal representation with minimum number of bends can be found in $O(n^{3/2})$ time.

• Given an orthogonal representation a layout with minimum area and total edge length is achievable for the case of rectangular faces.

• In case of non rectangular faces, reduce the problem to rectangular case. The resulting area is not minimum.

• Area minimization with a given orthogonal representation is an NP-hard problem. [Patrignany CGTA 2001]

• Solvable with an integer linear program (ILP). [Klau, Mutzel IPCO 1999]

• Various heuristics have been implemented and experimentally evaluated wrt running time and quality. [Klau, Klein, Mutzel GD 2001]
Summary

- An orthogonal representation with minimum number of bends can be found in $O(n^{3/2})$ time.
- Given an orthogonal representation a layout with minimum area and total edge length is achievable for the case of rectangular faces.
- In case of non-rectangular faces, reduce the problem to rectangular case. The resulting area is not minimum.
- Area minimization with a given orthogonal representation is an NP-hard problem. [Patrignany CGTA 2001]
- Solvable with an integer linear program (ILP) [Klau, Mutzel IPCO 1999]
- Various heuristics have been implemented and experimentally evaluated wrt running time and quality [Klau, Klein, Mutzel GD 2001]
- for non-planar graphs the area minimization is hard to approximate [Bannister, Eppstein, Simons JGAA 2012]