Algorithms for Graph Visualization
Flow Methods: Orthogonal Layout

Tamara Mchedlidze
8.01.2019
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
- ...

Flussmethoden: orthogonales Graphenzeichnen
(Planar) Orthogonal Drawings

Three-step approach:  \textit{Topology} – \textit{Shape} – \textit{Metrics}

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

(Planar) Orthogonal Drawings

Three-step approach:  \textit{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: \textit{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

(Planar) Orthogonal Drawings

Three-step approach:  \textit{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
- orthogonal representation
- combinatorial embedding/planarization
- Bend Minimization
- Area-minimization

\[ \text{Planar orthogonal drawing} \]
(Planar) Orthogonal Drawings

Three-step approach: Topology – Shape – Metrics


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

Reducing Crossings

combinatorial embedding/planarization

Bend Minimization

Area-minimization

orthogonal representation

planar orthogonal drawing
Orthogonal Representation

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

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

**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

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

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

for an edge \( \{u, v\} \) shared by faces \( f \) and \( g \) with
\[
((u, v), \delta_1, \alpha_1) \in H(f) \quad \text{and} \quad ((v, u), \delta_2, \alpha_2) \in H(g)
\]
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 $((u, v), \delta_1, \alpha_1) \in H(f)$ 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 $\sum_{r \in H(f_0)} C(r) = -4$
Correctness of an Orthogonal Representation

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

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 \( \delta_2 \)

3) 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
\[ \sum_{r \in H(f_0)} C(r) = -4 \]

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

1) $H(G)$ corresponds to $F, f_0$

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 $\delta_2$

3) 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:

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

4) For each node $v$ the sum of incident angles is $2\pi$

Pair, think and share:

What does the condition (H3) mean intuitively?
Bend Minimization with Given Embedding

**Problem: Geometric Bend Minimization**

Given: • planar Graph $G = (V, E)$ with maximum degree 4
  • combinatorial embedding $F$ 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 $F$ 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 $F$ 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 $F$ 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 $c : A \to \mathbb{R}_0^+$
- Source $s \in V$, Sink $t \in V$

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

\[
0 \leq X(u, v) \leq c(u, v) \quad \forall (u, v) \in A \quad (1)
\]

\[
\sum_{(u, v) \in A} X(u, v) - \sum_{(v, u) \in A} X(v, u) = 0 \quad \forall u \in V \setminus \{s, t\} \quad (2)
\]
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:

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

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

(A) Valid Flow:
Find a Valid Flow \( X : A \to \mathbb{R}_0^+ \), 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 $X : A \rightarrow \mathbb{R}_0^+$, 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: $\text{cost}(X) := \sum_{(u,v) \in A} \text{cost}(u,v) \cdot X(u,v)$

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

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

$\bullet \ A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ incident to } f\} \cup$

$\{(f, g) \in \mathcal{F} \times \mathcal{F} \mid f, g \text{ adjacent through edge } e\}$
Flow Network for Bend Minimization

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

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

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

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

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

$\forall (f, g) \in A, f, g \in \mathcal{F} \quad \ell(f, g) := 0 \leq X(f, g) \leq \infty =: c(f, g)$

$\forall (v, f) \in A, v \in V, f \in \mathcal{F} \quad \ell(v, f) := 1 \leq X(v, f) \leq 4 =: c(v, f)$

$\sum_w b(w) = 0$ (Euler)
Example Flow Network

\( f_0 \)

Flussmethoden: orthogonales Graphenzeichnen
Example Flow Network
Example Flow Network

![Diagram of a flow network with nodes and edges labeled with notation and variable symbols.

Graph consists of nodes labeled $v_1, v_2, v_3, v_4, v_5$ connected by edges $e_1, e_2, e_3, e_4, e_5, e_6$, and flow values $f_0, f_1, f_2$. The notation $V \times \mathcal{F}$ represents the product set of vertices and flows.]
Example Flow Network

\[ V \times F \supseteq \]

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

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

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

\[
\begin{array}{c}
\text{\(f_0\)} & \text{\(-14\)} \\
\text{\(e_1\)} & \text{\(1\)} \\
\text{\(e_2\)} & \text{\(2\)} \\
\text{\(e_3\)} & \text{\(3\)} \\
\text{\(f_1\)} & \text{\(1\)} \\
\text{\(f_2\)} & \text{\(1\)} \\
\text{\(v_1\)} & \text{\(4\)} \\
\text{\(v_2\)} & \text{\(4\)} \\
\text{\(v_3\)} & \text{\(4\)} \\
\text{\(v_4\)} & \text{\(4\)} \\
\text{\(v_5\)} & \text{\(4\)} \\
\end{array}
\]

\[
\begin{align*}
\ell/u/c & = 1/4/0 \\
V \times \mathcal{F} & \supseteq \\
\mathcal{F} \times \mathcal{F} & \supseteq
\end{align*}
\]

\(V \subset \mathcal{F}\)
Example Flow Network

\begin{itemize}
\item $f_0$
\item $f_1$
\item $f_2$
\item $e_1$
\item $e_2$
\item $e_3$
\item $e_4$
\item $e_5$
\item $e_6$
\item $v_1$
\item $v_2$
\item $v_3$
\item $v_4$
\item $v_5$
\end{itemize}

\text{cost} = 1

\textbf{bend!}

\textbf{outside}

\text{\ell/u/c $1/4/0$}

\text{$V \times \mathcal{F} \supseteq$}

\text{\mathcal{F} \times \mathcal{F} \supseteq$}

\begin{align*}
V & \quad \mathcal{F} \\
\circ & \quad \bullet
\end{align*}
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:**

\(\Leftarrow\) Given flow \(X\) in \(N(G)\) 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:

\[\Leftrightarrow \text{Given flow } X \text{ in } N(G) \text{ with cost } k\]

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:**

\(\iff\) 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)

\(\implies\) 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.

[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. [Garg, Tamassia SIAM J. Comput. 2001]
Summary of Bend Minimization

• From Theorem 1 is 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**

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

Planar orthogonal drawing

Reduce Crossings

Bend Minimization

Area-minimization

(Planare) Orthogonale Zeichnungen

Three-step approach: 

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

Topology – Shape – Metrics

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

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

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

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

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

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

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

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

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

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

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

What values of the drawing represent the following?

- $|x_{hor}(t, s)|$ and $|x_{ver}(t, s)|$?
- $\sum_{a \in A_{hor}} x_{hor}(a) + \sum_{a \in A_{ver}} x_{ver}(a)$
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 $O(n^{3/2}) \ast$ 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 minimun 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

- 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
• 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

- 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 guarantee minimum total edge length
- Time complexity $O(n)$

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

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

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

Dummy nodes for bends: •

corner(e)

next(e)
Refinement of \((G, H)\) – Inner Face

\[
\text{dummy nodes for bends: } \bullet
\]

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

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

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

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

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

- **Dummy nodes for bends:**
  - **Turn function:**
    
    \[\text{turn}(e) = \begin{cases} 
    1 & \text{left bend} \\
    0 & \text{no bend} \\
    -1 & \text{right bend}
    \end{cases}\]
Refinement of \((G, H) - \text{Inner Face}\)

Dummy nodes for bends: \(\bigcirc\)

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

\[ f_0 \]
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 $(G, H) - \text{Outer Face}$

- $\text{front}(e)$ 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 \((G, H)\) – 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
- 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
- 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
- when \(\sum \text{turn}(e) < 1\) for the complete turn around \(f_0\), project on \(R\)

all faces are rectangles → apply flow network

\[ e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \]
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

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

Has minimum area? NO!
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]