Algorithms for Graph Visualization

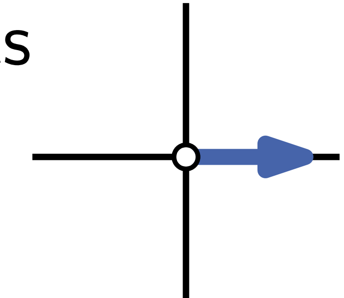
Flow Methods: Orthogonal Layout

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Tamara Mchedlidze

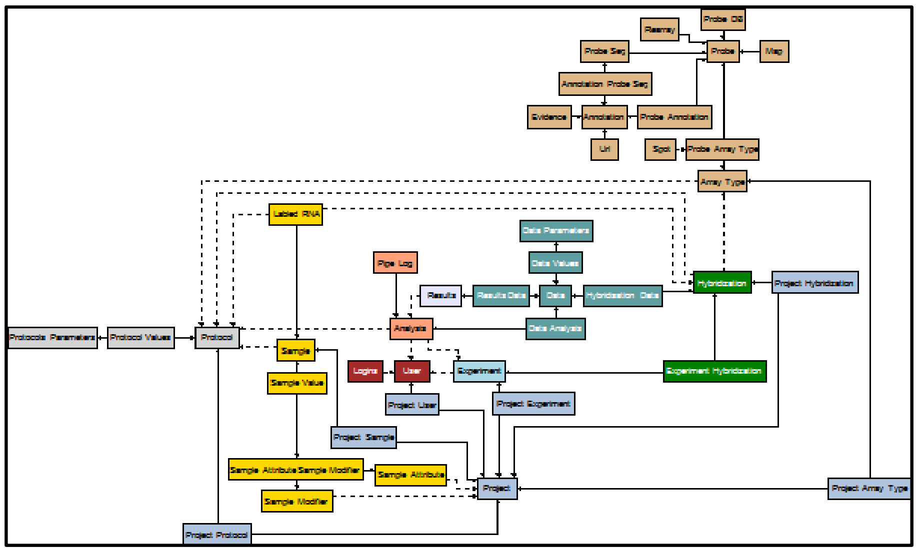
8.01.2019

Orthogonal Layout

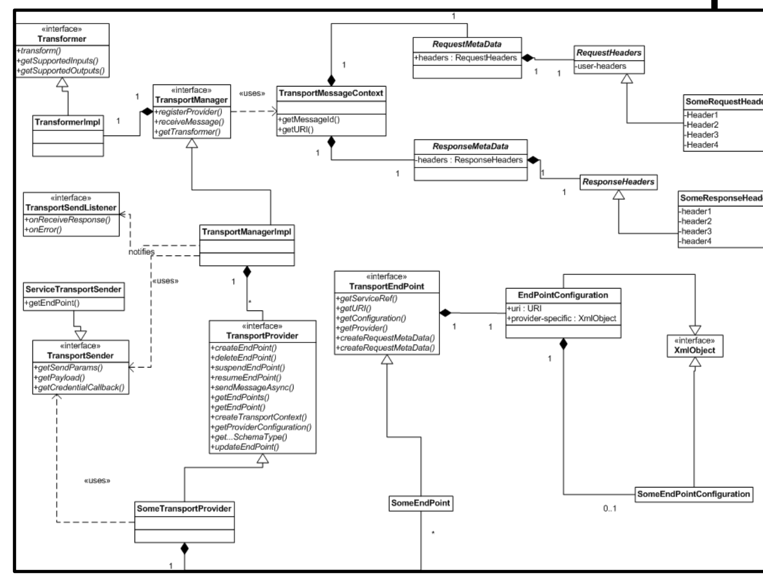


- Edges consist of vertical and horizontal segments
- Applied in many areas

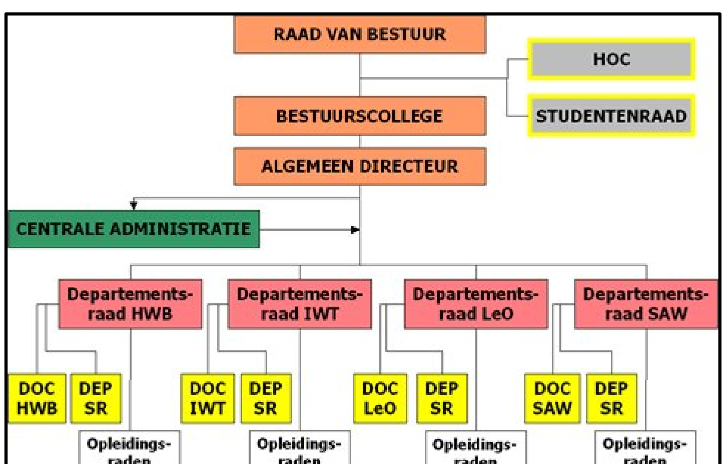
ER diagram in OGDF



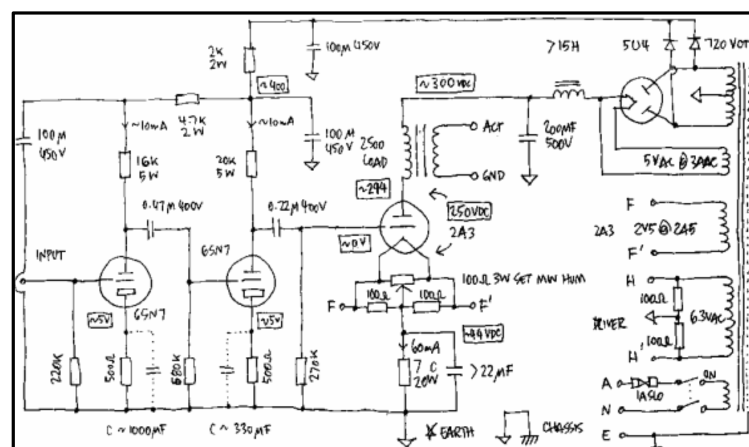
UML diagram by Oracle



Organigram of HS Limburg

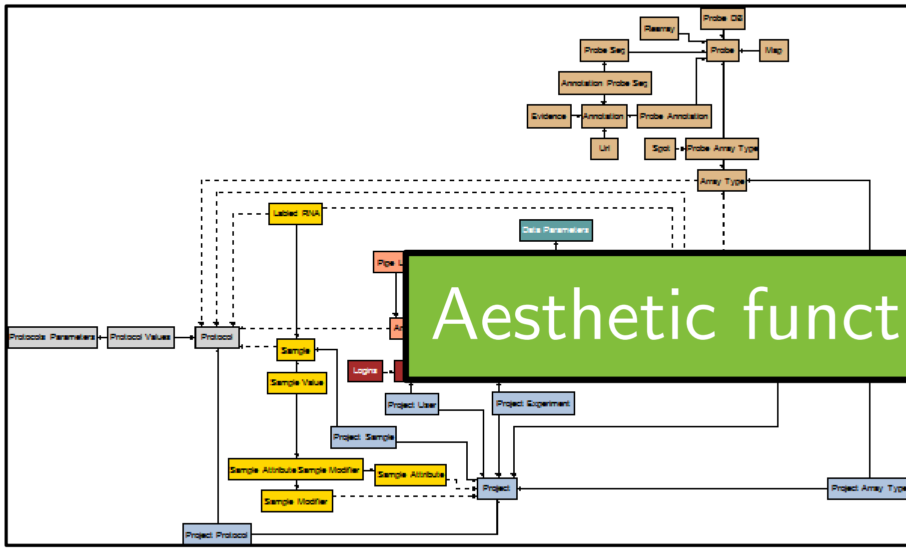


Circuit diagram by Jeff Atwood

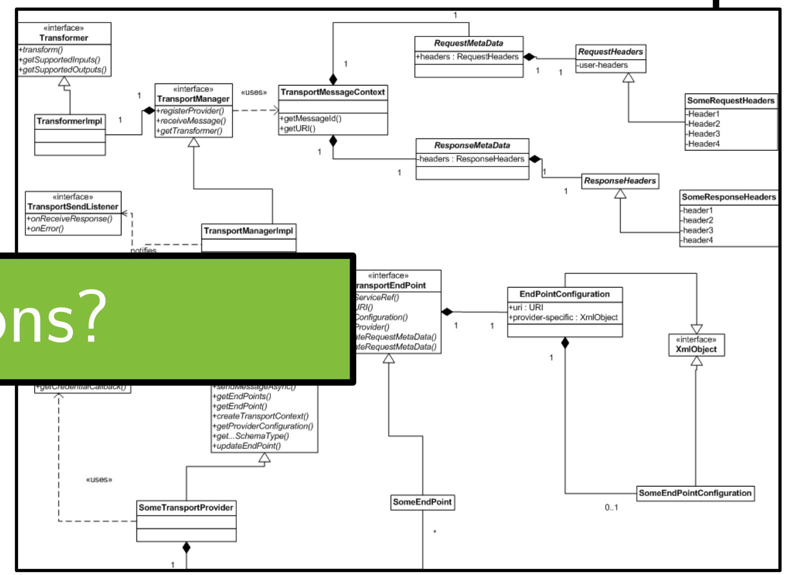


Orthogonal Layout

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

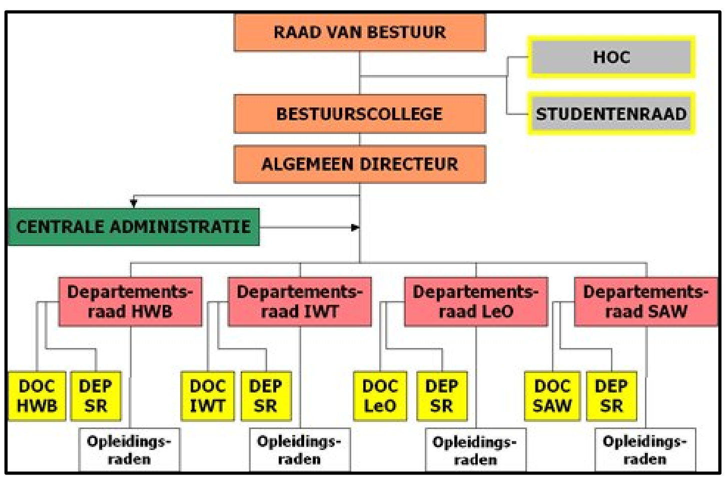


ER diagram in OGDF

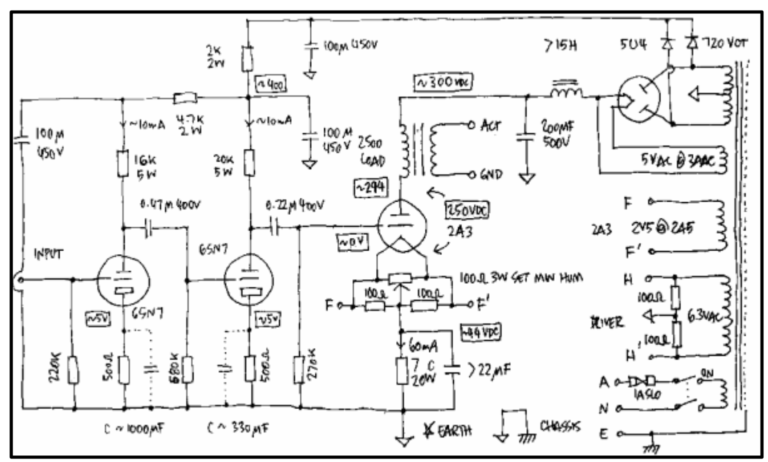


UML diagram by Oracle

Aesthetic functions?



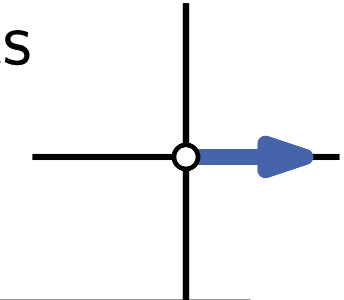
Organigram of HS Limburg

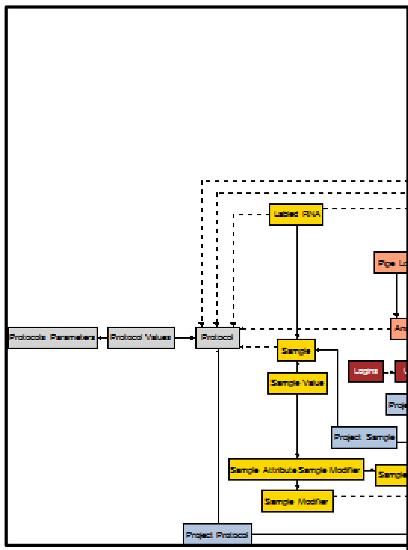


Circuit diagram by Jeff Atwood

Orthogonal Layout

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

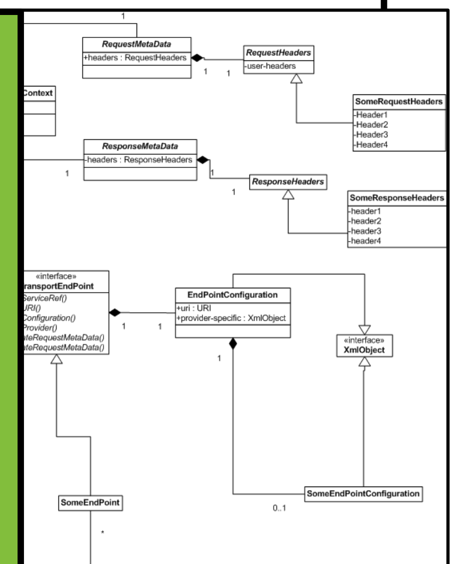




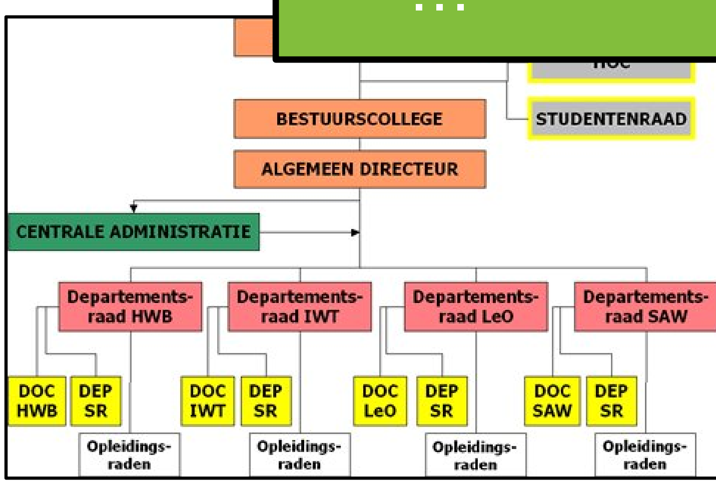
ER diagram in OGDF

Aesthetic functions:

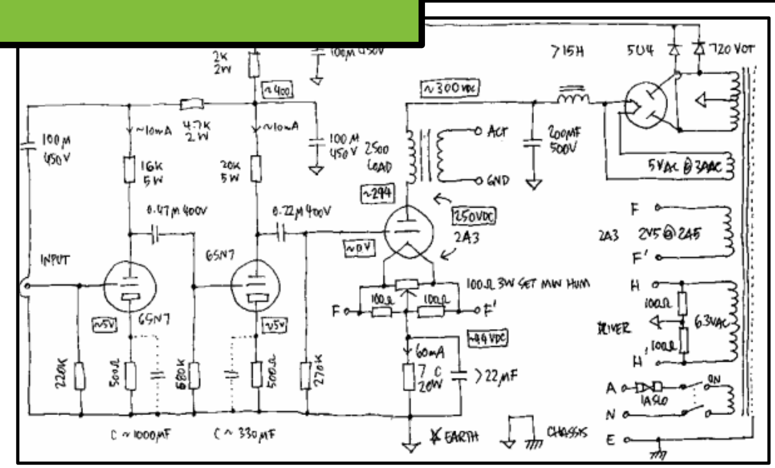
- number of bends
- length of edges
- width, height, area
- monotonicity of edges
- ...



UML diagram by Oracle



Organigram of HS Limburg



Circuit diagram by Jeff Atwood

(Planar) Orthogonal Drawings

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

[Tamassia SIAM J. Comput. 1987]

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

(Planar) Orthogonal Drawings

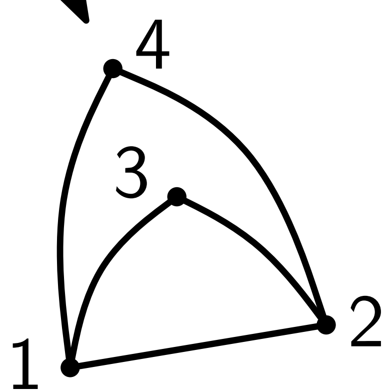
Three-step approach: *Topology – Shape – Metrics*

[Tamassia SIAM J. Comput. 1987]

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

Reduce Crossings

combinatorial embedding/ planarization

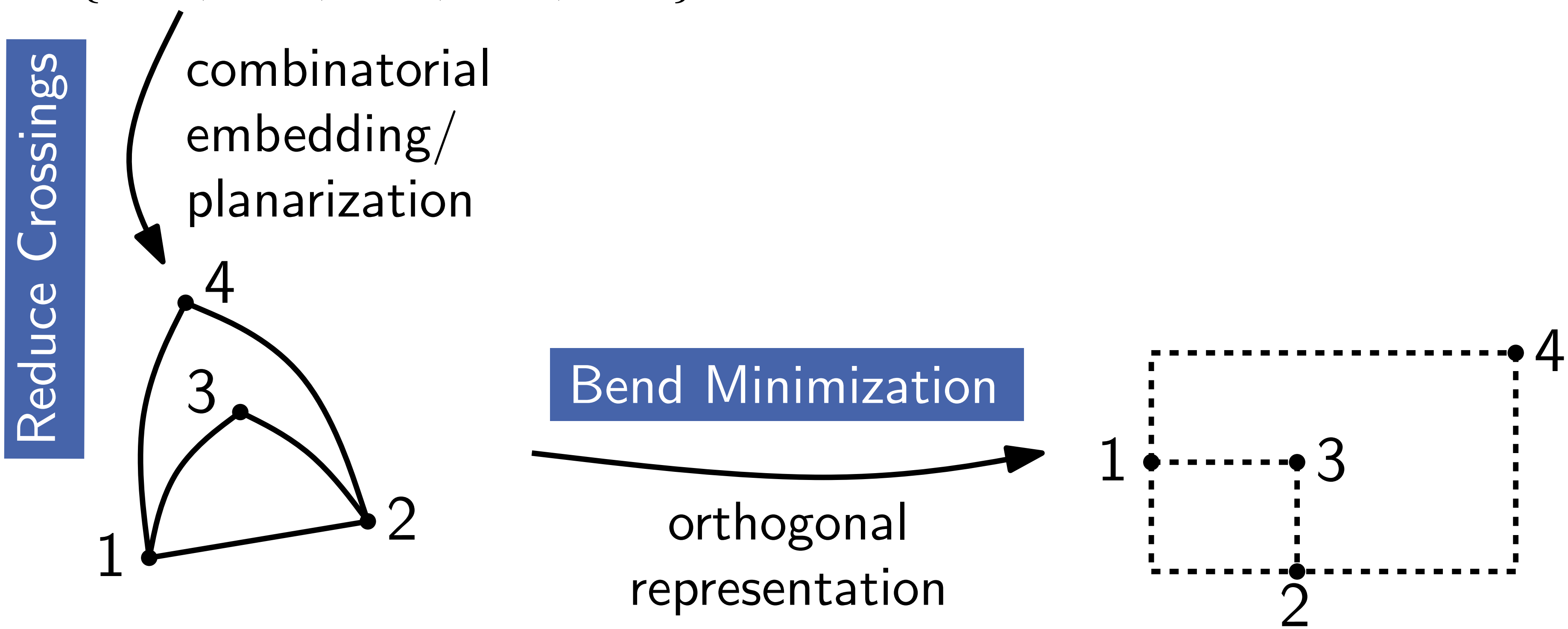


(Planar) Orthogonal Drawings

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

[Tamassia SIAM J. Comput. 1987]

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

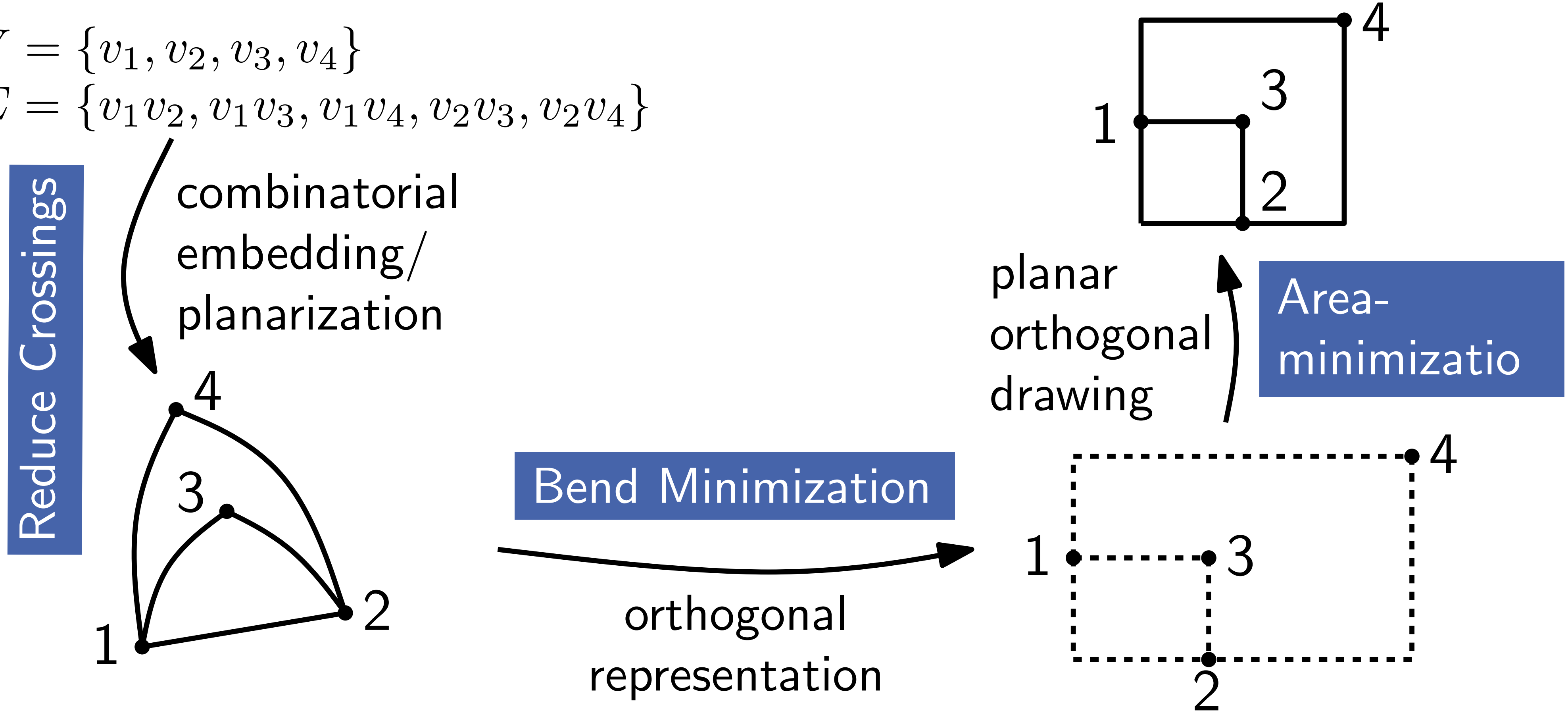


(Planar) Orthogonal Drawings

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

[Tamassia SIAM J. Comput. 1987]

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

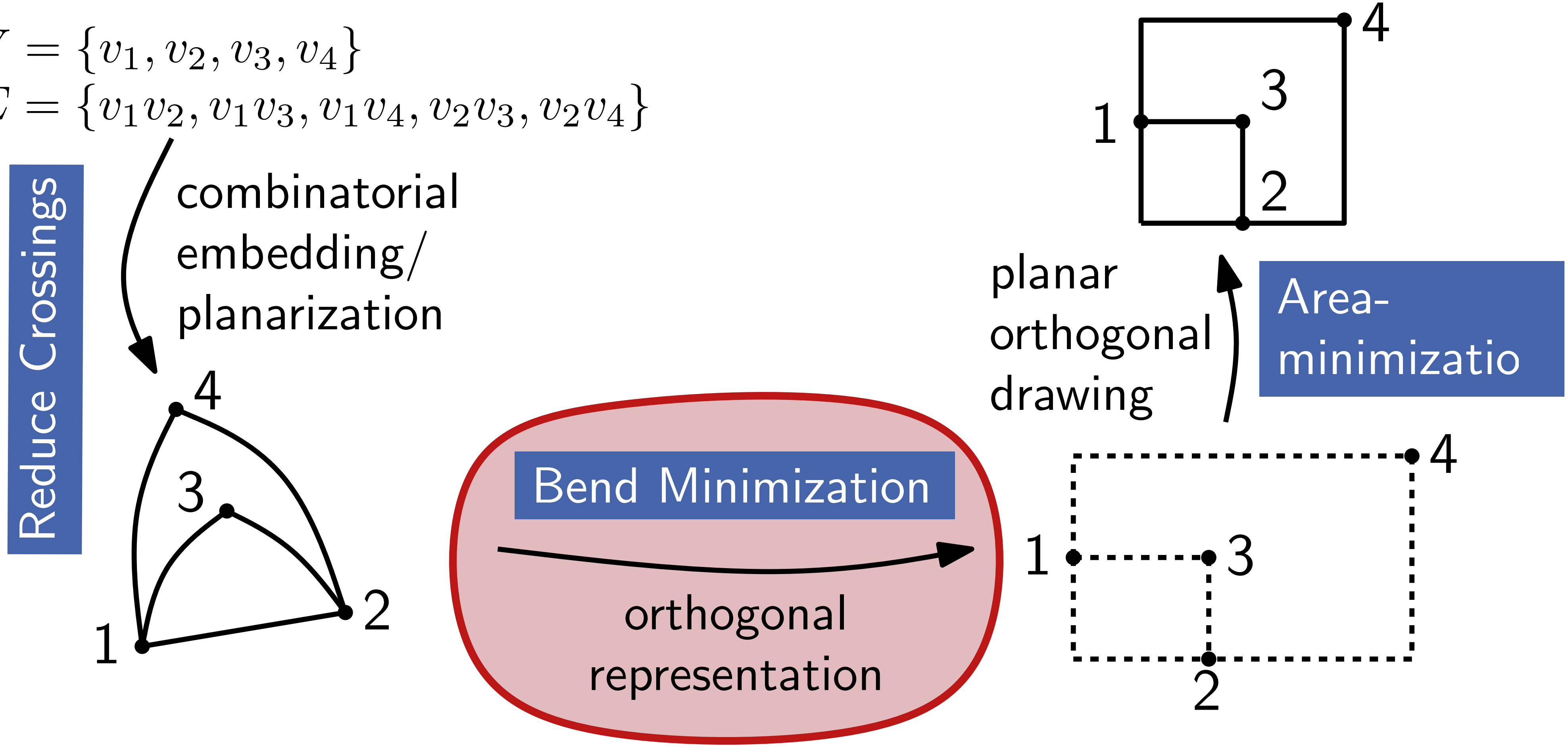


(Planar) Orthogonal Drawings

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

[Tamassia SIAM J. Comput. 1987]

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



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, δ, α) with

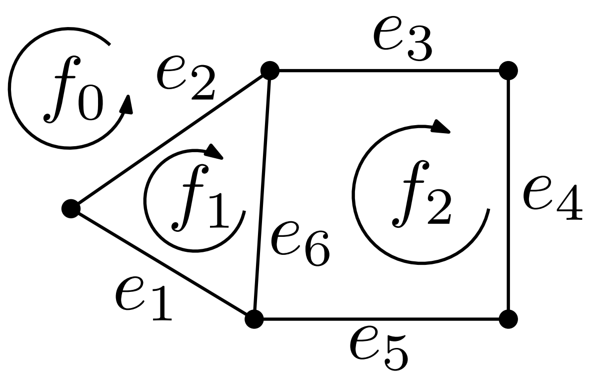
- e edge of f
- δ is sequence of $\{0, 1\}^*$ (0 = right bend, 1 = left bend)
- α is angle $\in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$ between e and next edge e'

Orthogonal Representation: Example

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

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

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



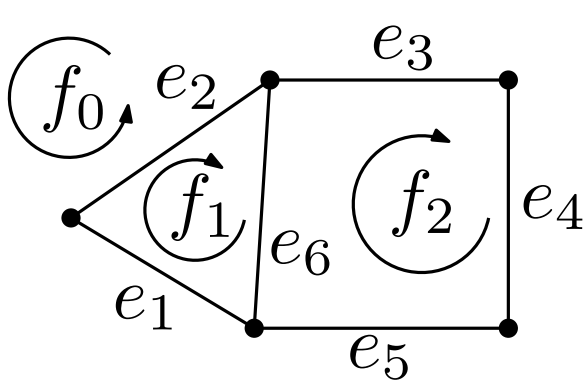
Combinatorial “drawing” of $H(G)$?

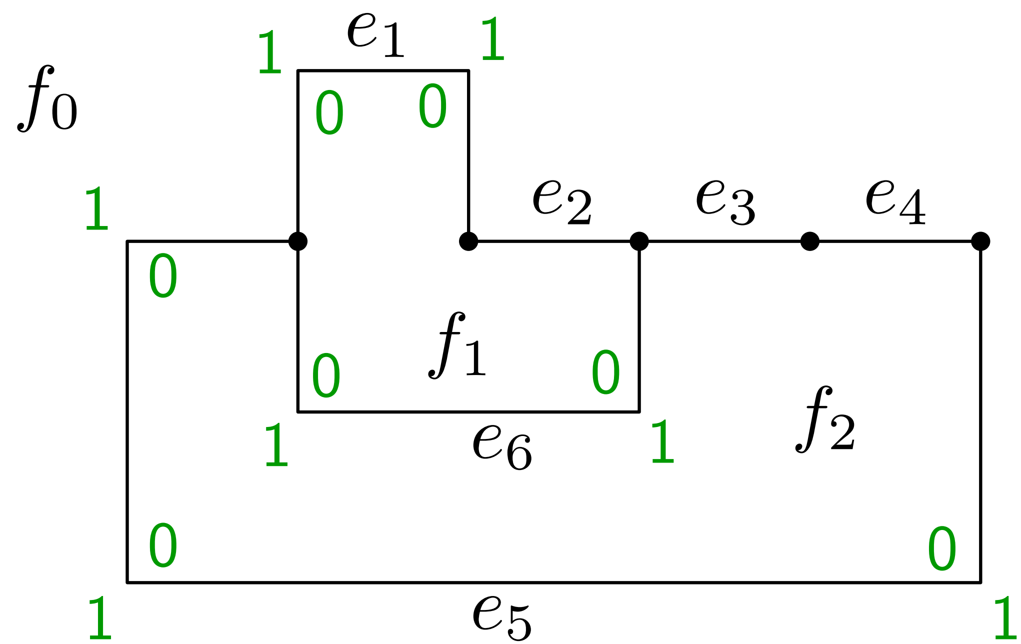
Orthogonal Representation: Example

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

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

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





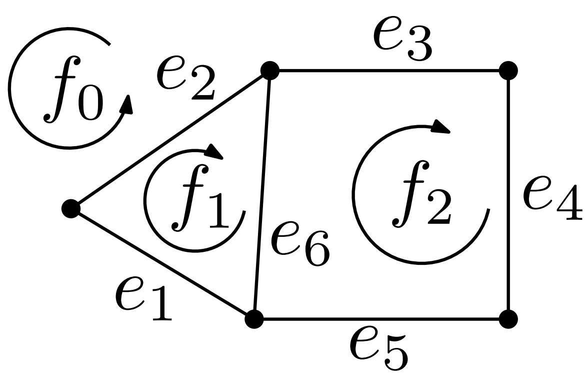
Orthogonal Representation: Example

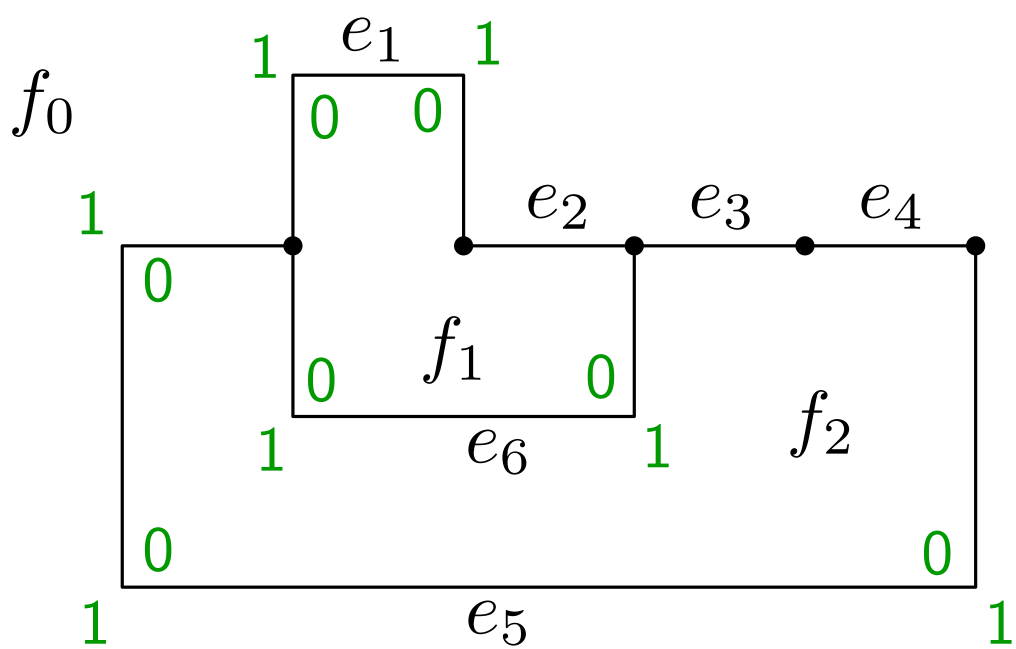
$$H(f_0) = ((e_1, 11, \tfrac{\pi}{2}), (e_5, 111, \tfrac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \tfrac{\pi}{2}))$$

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

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

is f_0 listed wrongly!?



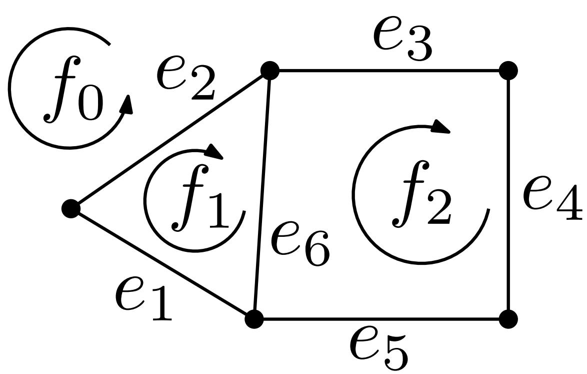


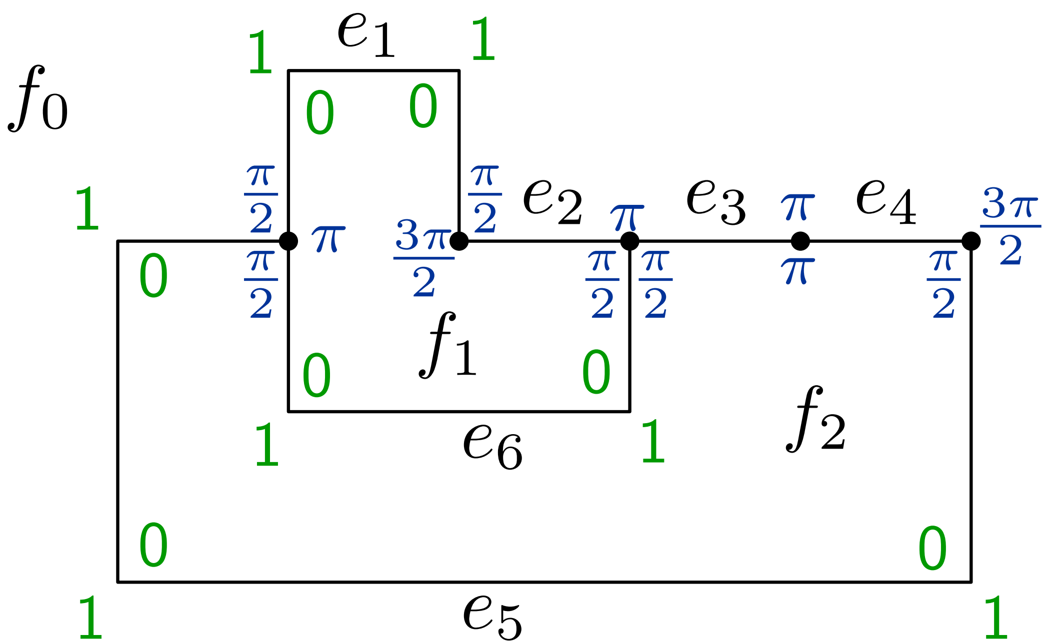
Orthogonal Representation: Example

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

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

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



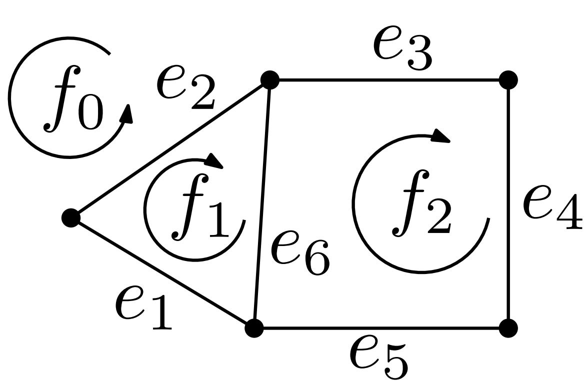


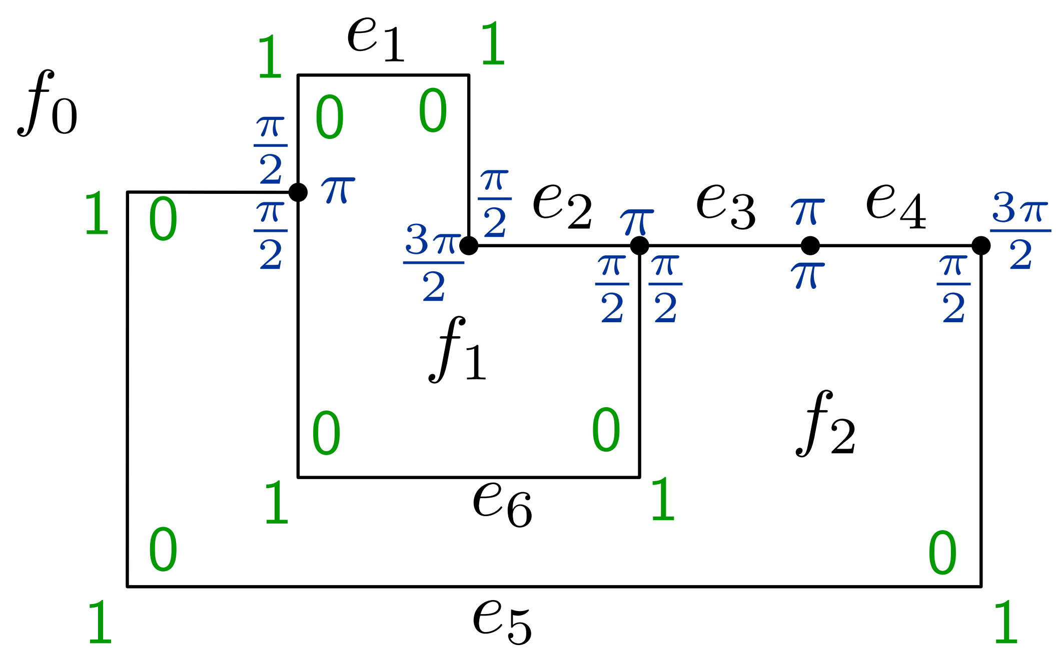
Orthogonal Representation: Example

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

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

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



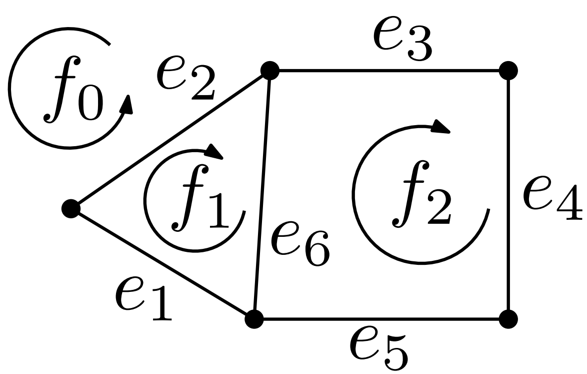


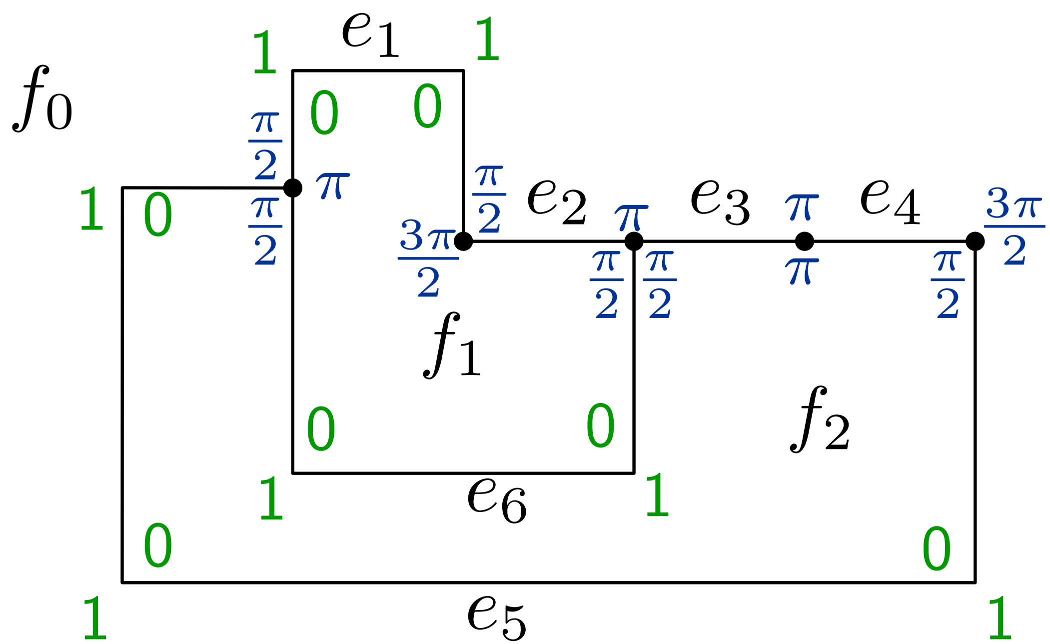
Orthogonal Representation: Example

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

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

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





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

H1) $H(G)$ corresponds to $\mathcal{F}, f_0$

H2) 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 δ_1 is reverced and inverted δ_2

Correctness of an Orthogonal Representation

H1) $H(G)$ corresponds to $\mathcal{F}, f_0$

H2) 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 δ_1 is reverced and inverted δ_2

H3) Let $|\delta|_0$ (resp. $|\delta|_1$) be the number of zeros (resp. ones) in δ 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

H1) $H(G)$ corresponds to $\mathcal{F}, f_0$

H2) 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 δ_1 is reverced and inverted δ_2

H3) Let $|\delta|_0$ (resp. $|\delta|_1$) be the number of zeros (resp. ones) in δ 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 $\sum_{r \in H(f_0)} C(r) = -4$

H4) For each node v the summ of incident angles is 2π

Correctness of an Orthogonal Representation

H1) $H(G)$ corresponds to $\mathcal{F}, f_0$

H2) 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 δ_1 is reverced and inverted δ_2

H3) Let $|\delta|_0$ (resp. $|\delta|_1$) be the number of zeros (resp. ones) in δ 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 $\sum_{r \in H(f_0)} C(r) = -4$

H4) For each node v the summ of incident angles is 2π

Pair, think and share:

What does the condition (H3) mean intuitively?

5 min

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 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

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 $c : A \to \mathbb{R}_0^+$
- Source $s \in V$, Sink $t \in V$

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

$$0 \leq X(u, v) \leq c(u, v) \qquad \forall (u, v) \in A \tag{1}$$

$$\sum_{(u,v) \in A} X(u, v) - \sum_{(v,u) \in A} X(v, u) = 0 \qquad \forall u \in V \setminus \{s, t\} \tag{2}$$

Reminder: General Flow Network

Karlsruhe Institute of Technology

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

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

An assignement $X : A \to \mathbb{R}_0^+$ is called **valid flow**, if:

$$\ell(u, v) \leq X(u, v) \leq c(u, v) \qquad \forall (u, v) \in A \qquad (3)$$

$$\sum_{(u,v) \in A} X(u, v) - \sum_{(v,u) \in A} X(v, u) = b(u) \qquad \forall u \in V \qquad (4)$$

Problems for Flow Networks

Karlsruhe Institute of Technology

(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 $X : A \to \mathbb{R}_0^+$, such that.

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

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

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

Problems for Flow Networks

(A) Valid Flow:

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

- Lower bounds and capacities $\ell(e), u(e)$ 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) Miminum 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 $N(G) = ((V \cup \mathcal{F}, A); \ell; c; b; \text{cost})$

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

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

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

Flow Network for Bend Minimization

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

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

$\Rightarrow \sum_w b(w) \stackrel{?}{=} 0$

Flow Network for Bend Minimization

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

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

$\Rightarrow \sum_w b(w) = 0$ (Euler)

Flow Network for Bend Minimization

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

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

$\Rightarrow \sum_w b(w) = 0$ (Euler)

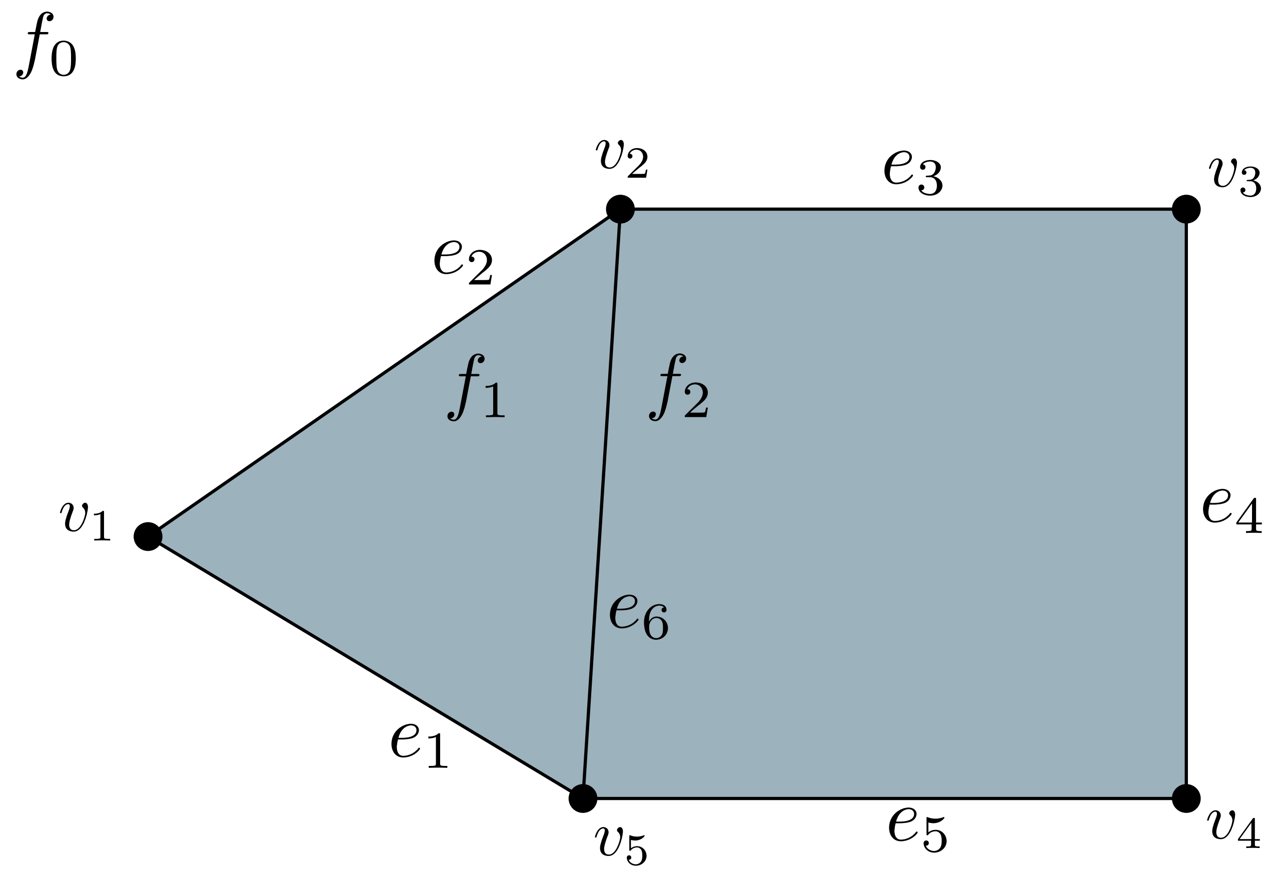
$\forall (f, g) \in A, f, g \in \mathcal{F} \qquad \ell(f, g) := 0 \leq X(f, g) \leq \infty =: c(f, g)$

$\text{cost}(f, g) = 1$

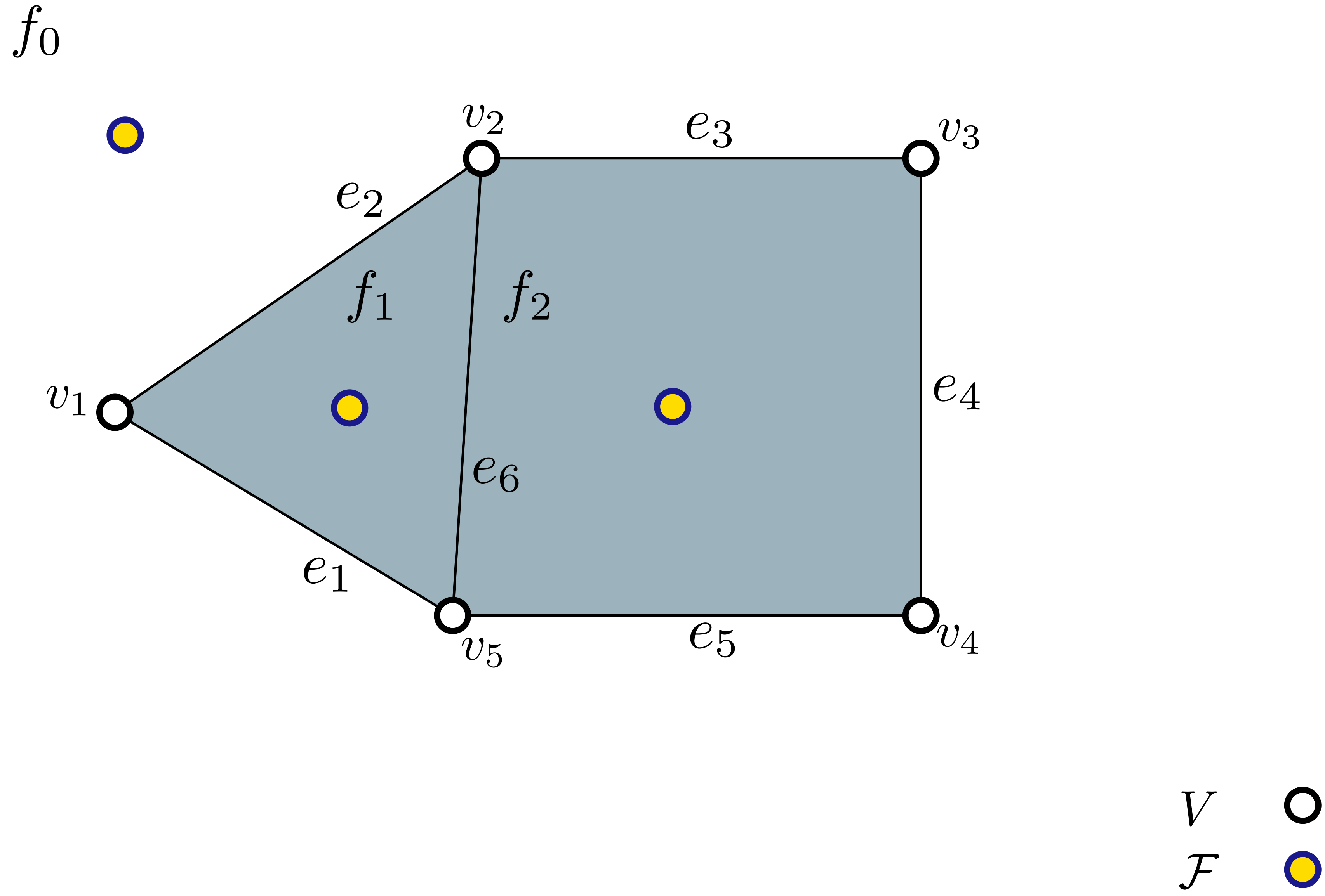
$\forall (v, f) \in A, v \in V, f \in \mathcal{F} \qquad \ell(v, f) := 1 \leq X(v, f) \leq 4 =: c(v, f)$

$\text{cost}(v, f) = 0$

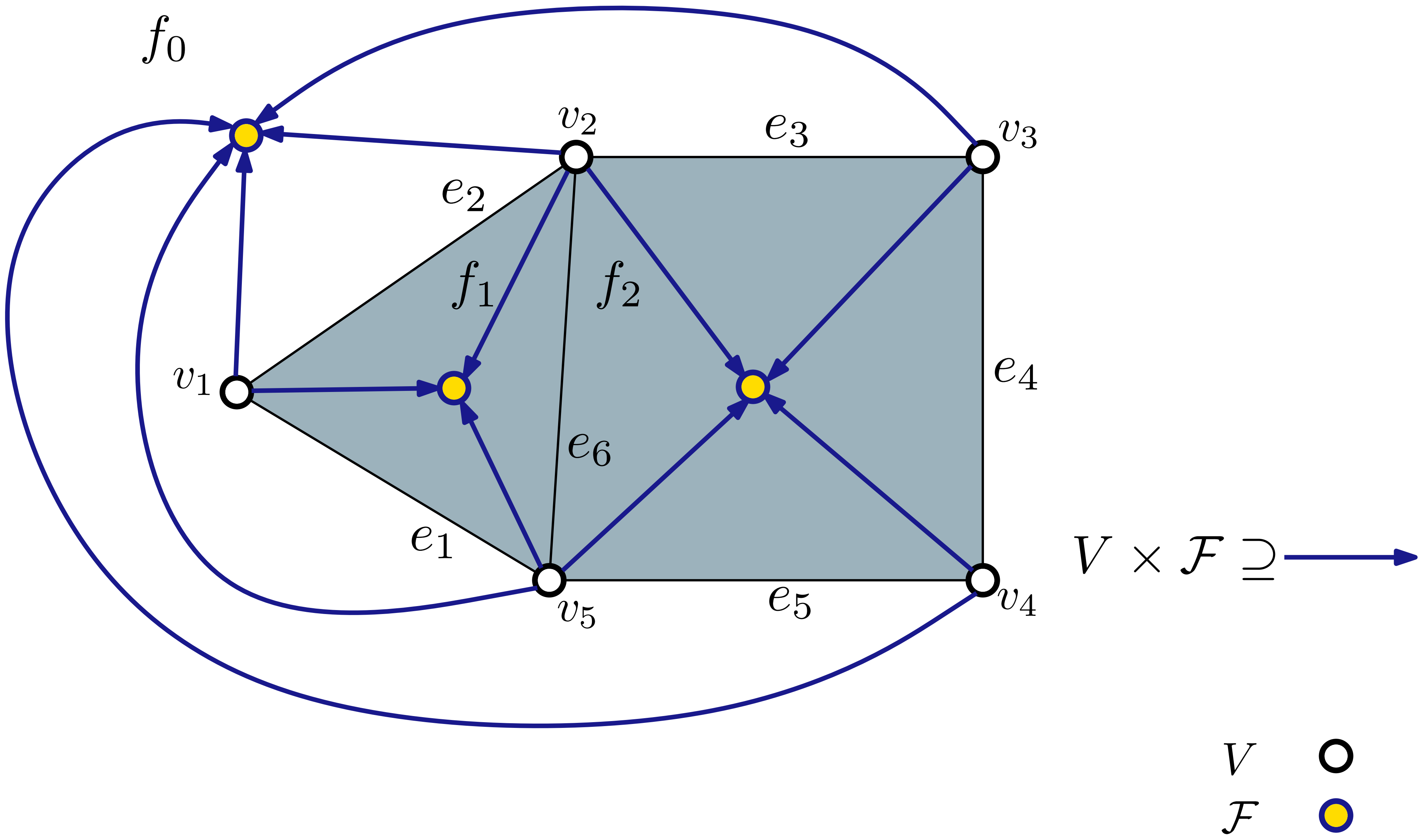
Example Flow Network



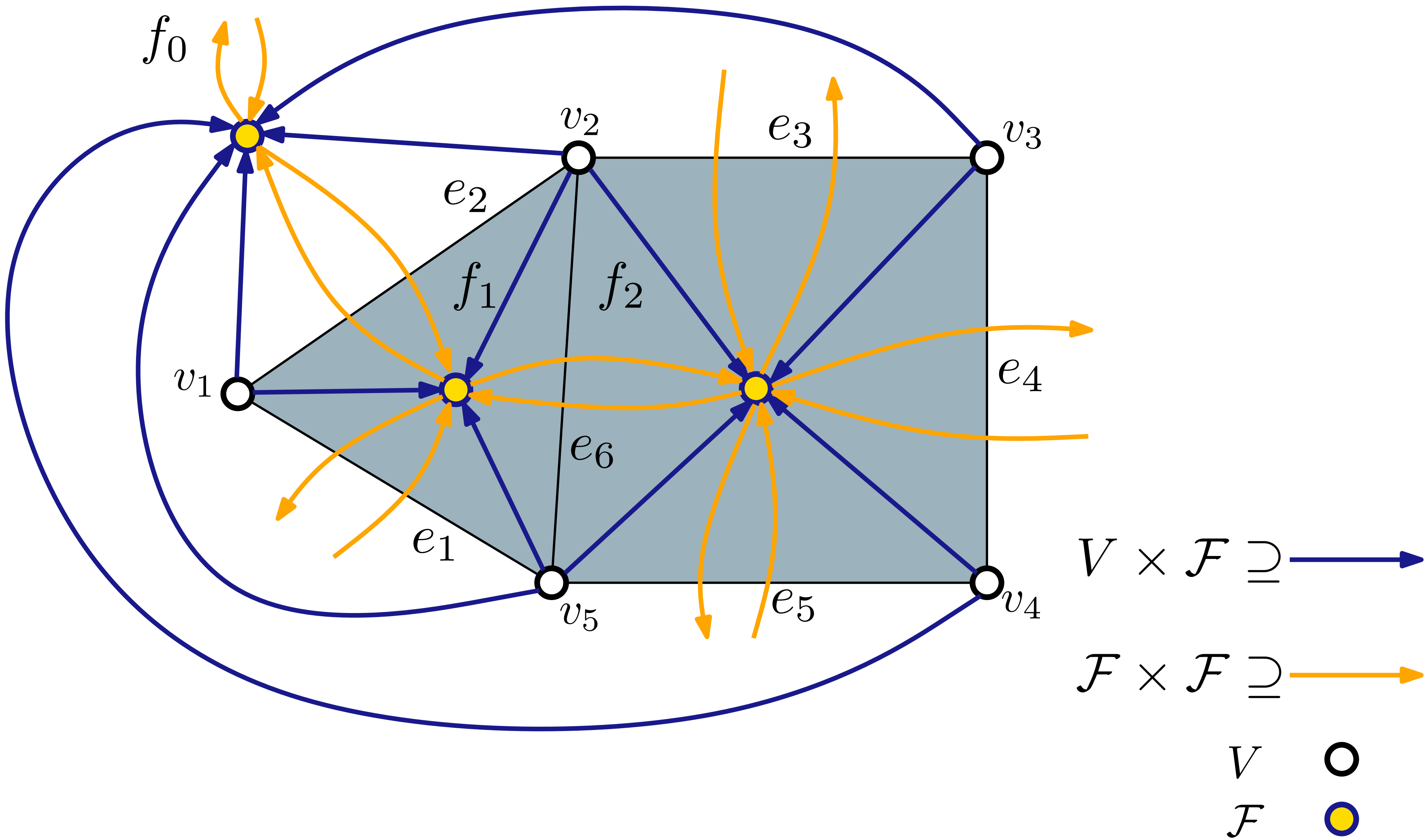
Example Flow Network



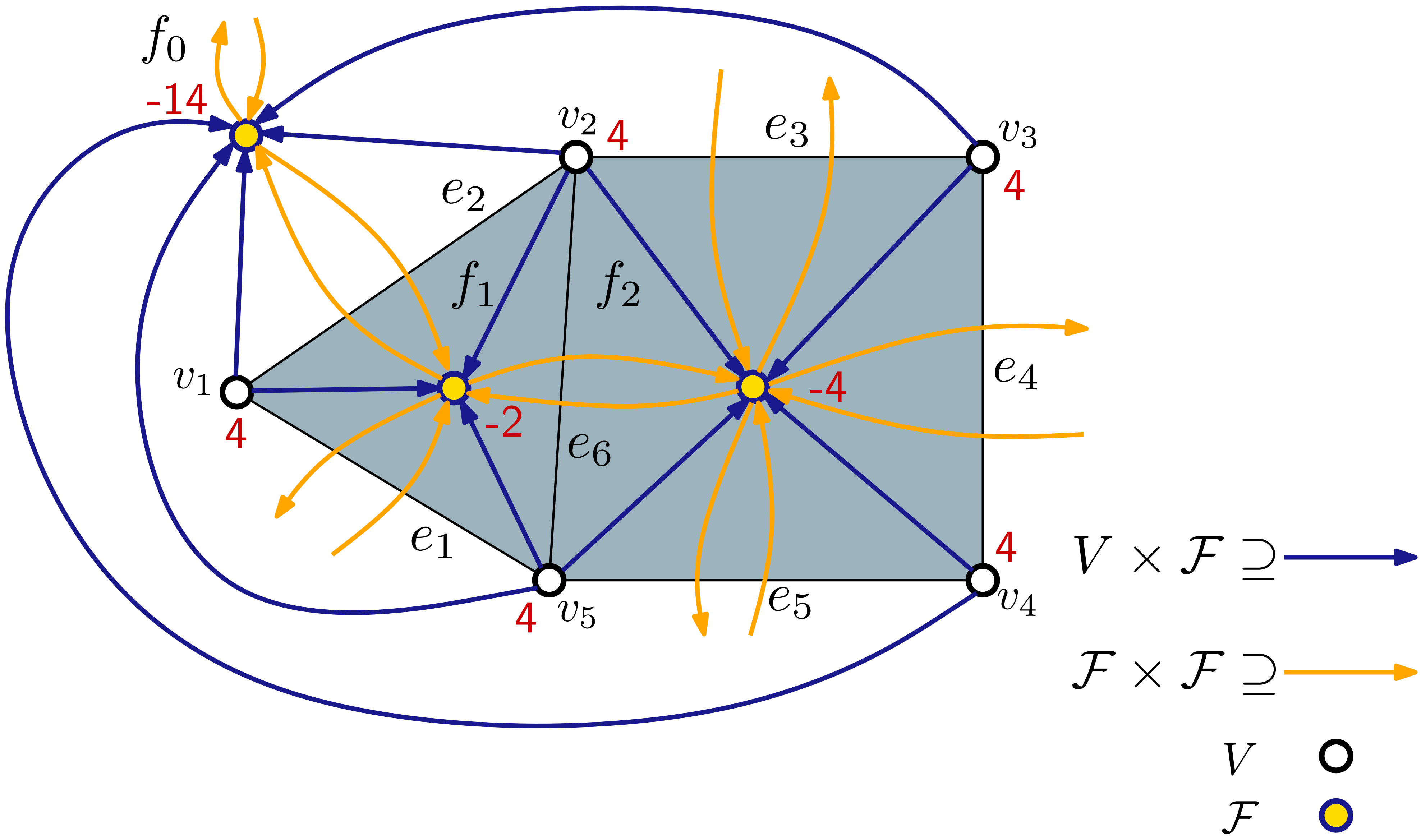
Example Flow Network



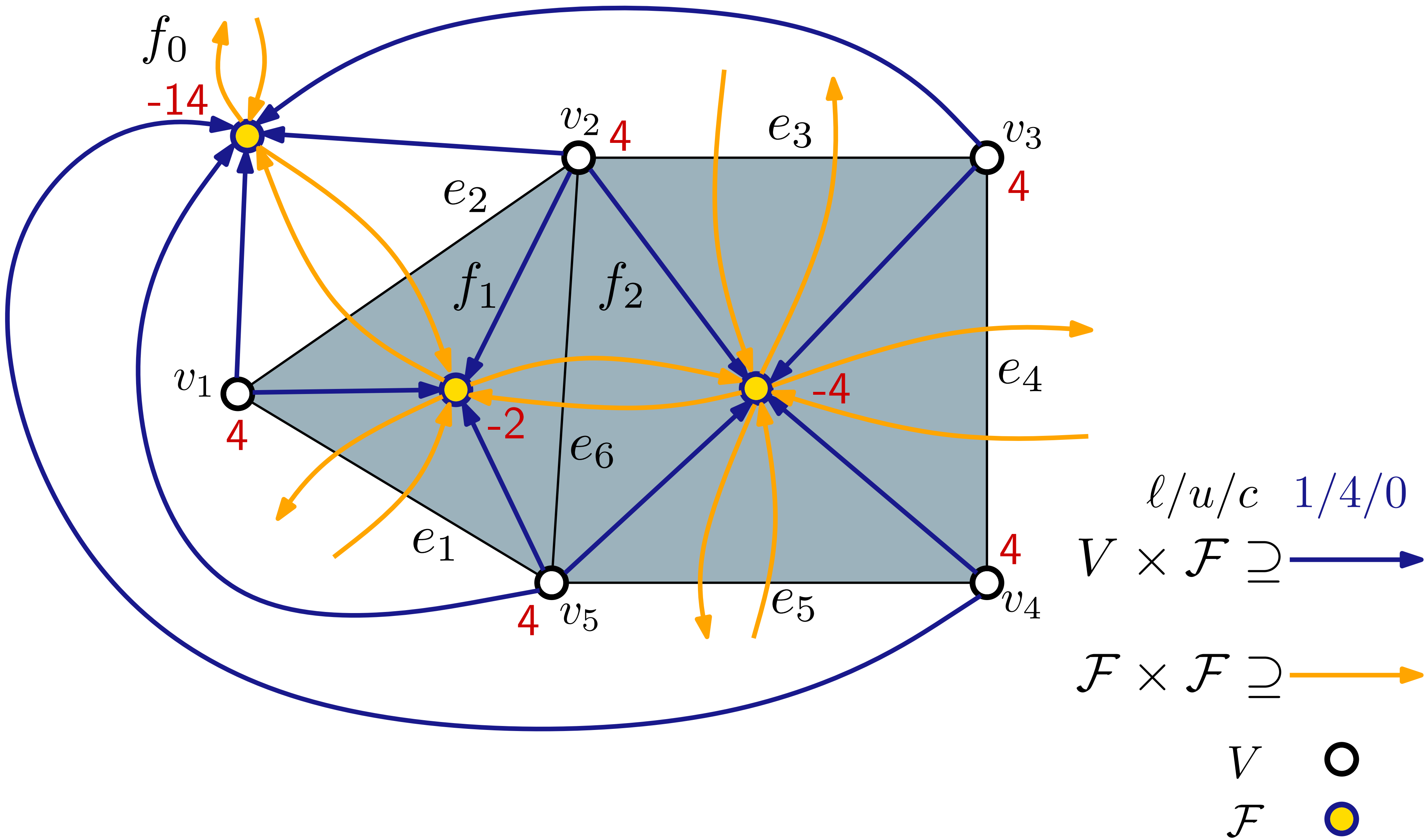
Example Flow Network



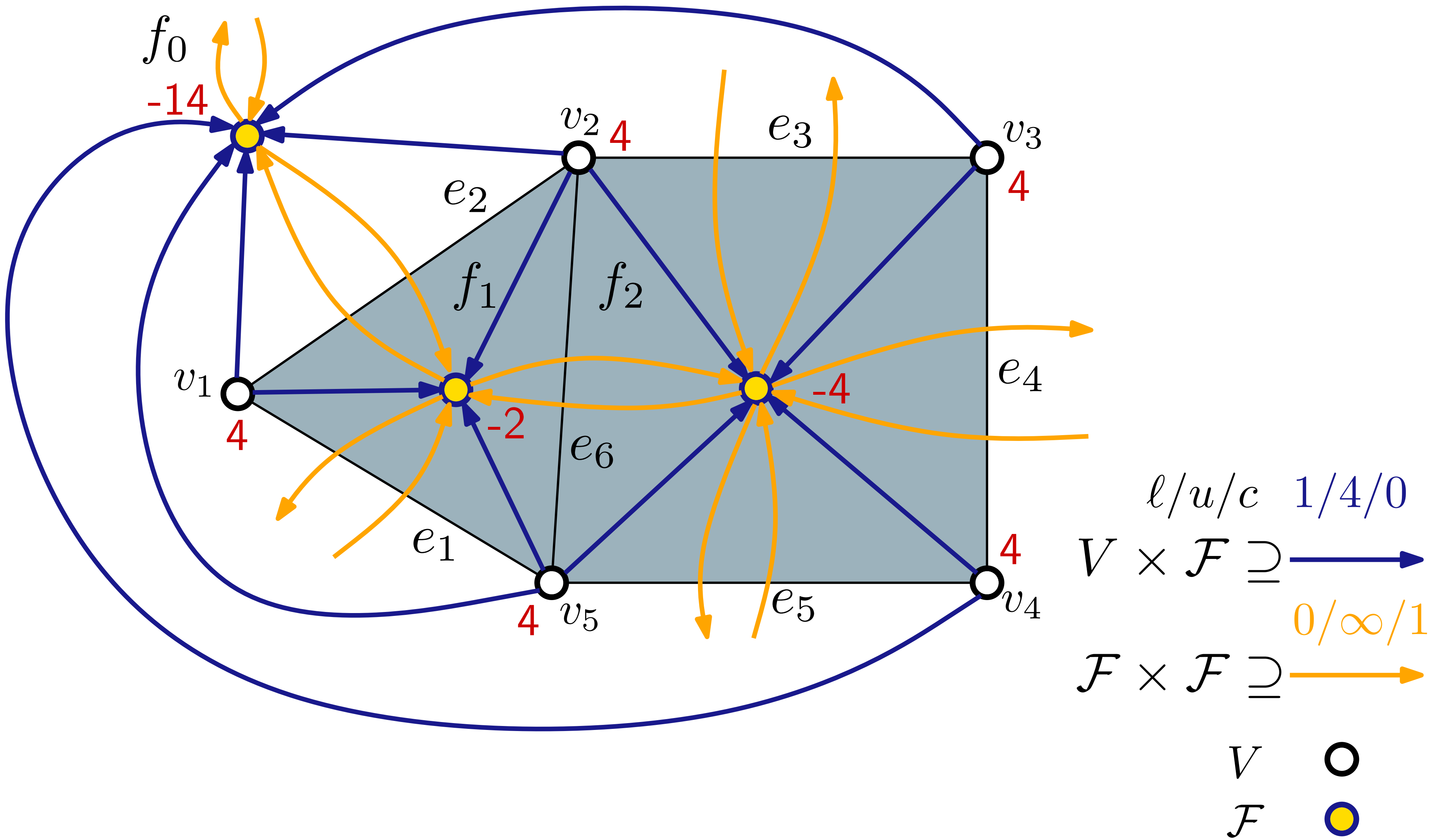
Example Flow Network



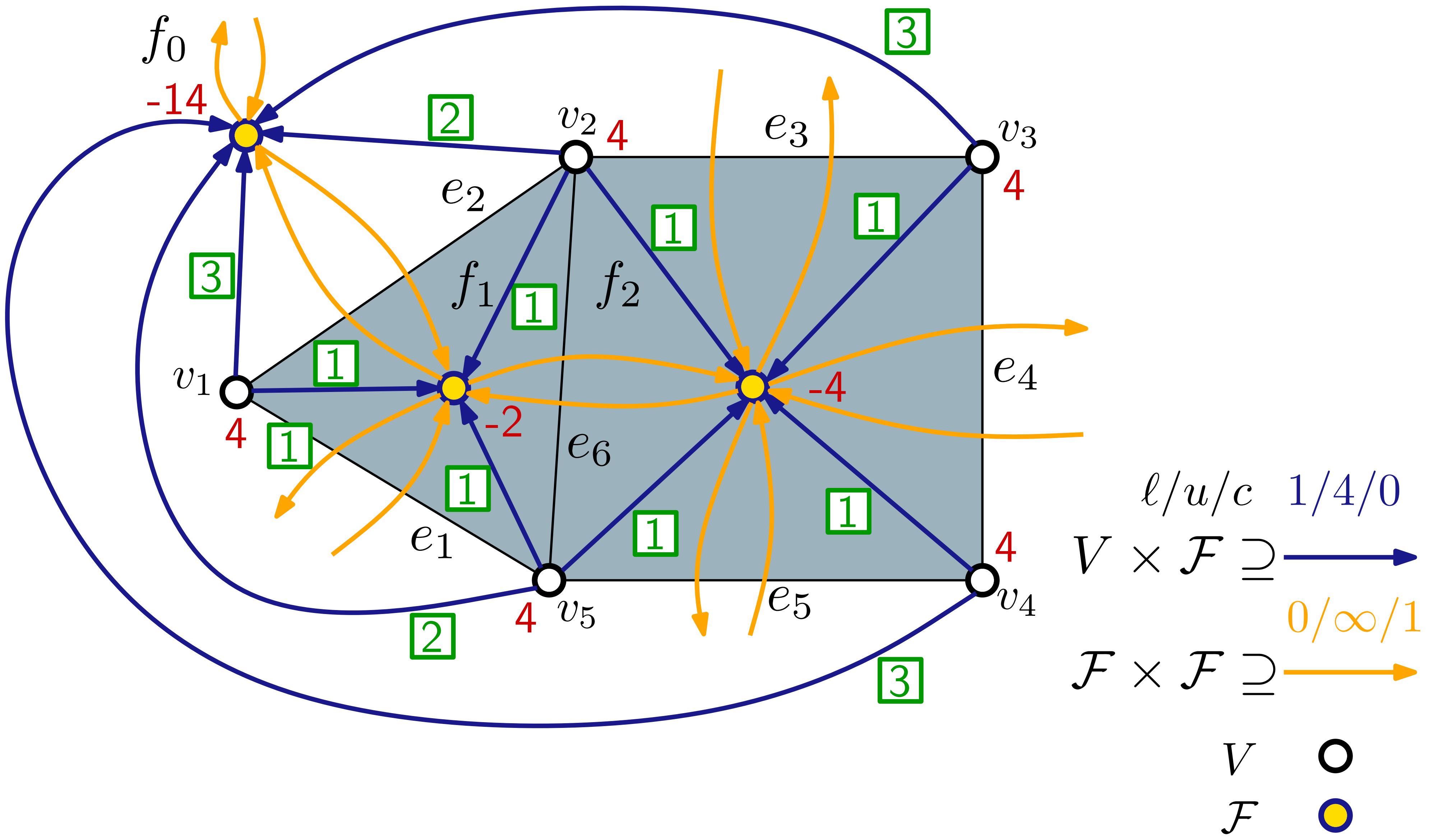
Example Flow Network



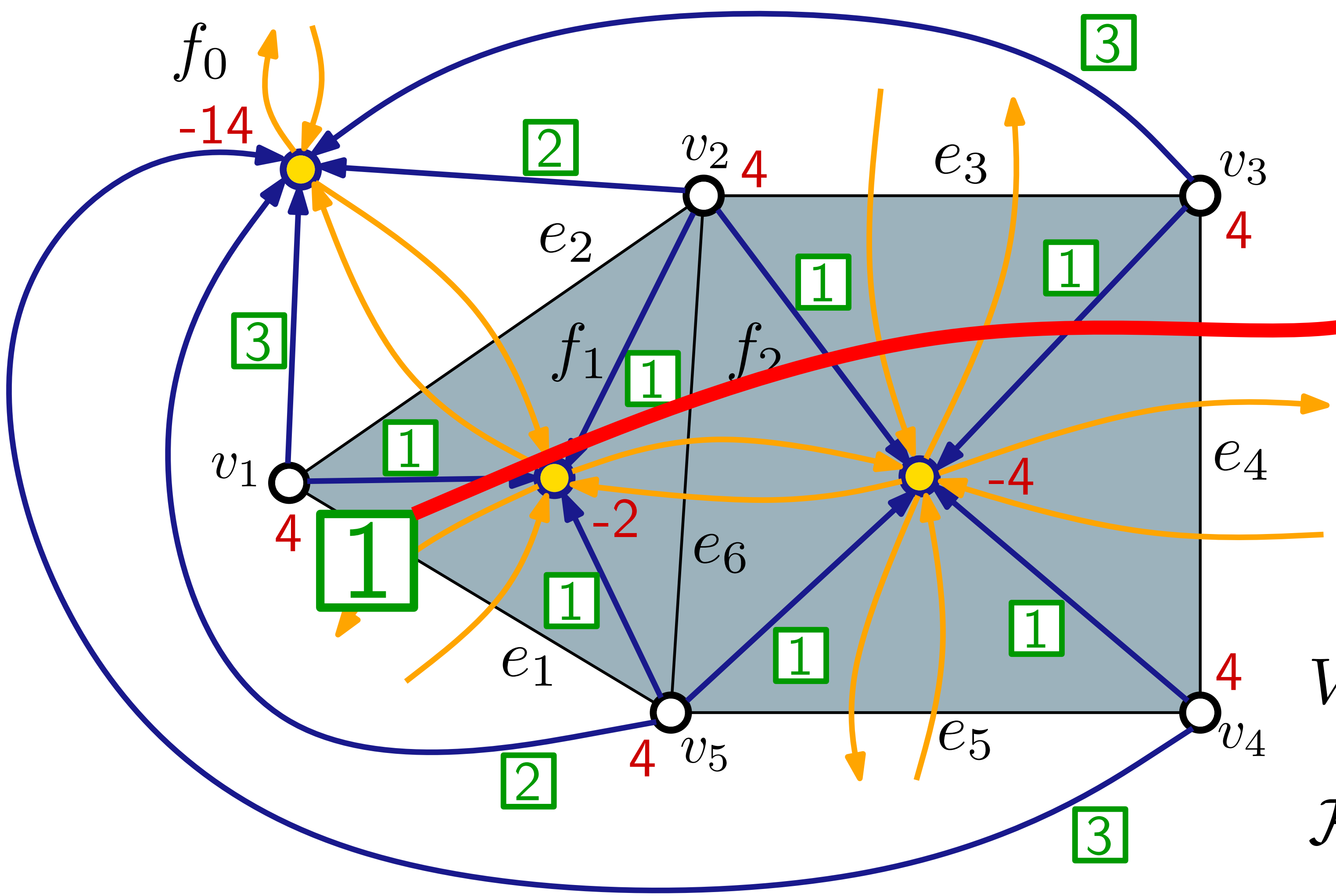
Example Flow Network



Example Flow Network



Example Flow Network



cost = 1
bend!
outside

$\ell/u/c$ $1/4/0$
$V \times \mathcal{F} \supseteq$
$0/\infty/1$
$\mathcal{F} \times \mathcal{F} \supseteq$
V
$\mathcal{F}$

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:

$\Leftarrow$ 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

Karlsruhe Institute of Technology

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 $X : A \to \mathbb{R}_0^+$
- show that X is a valid flow and has cost k

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.

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 slolved in $O(n^{3/2})$ time.

[Cornelsen, Karrenbauer GD 2011]

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 slolved 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 slolved in $O(n^{3/2})$ time.

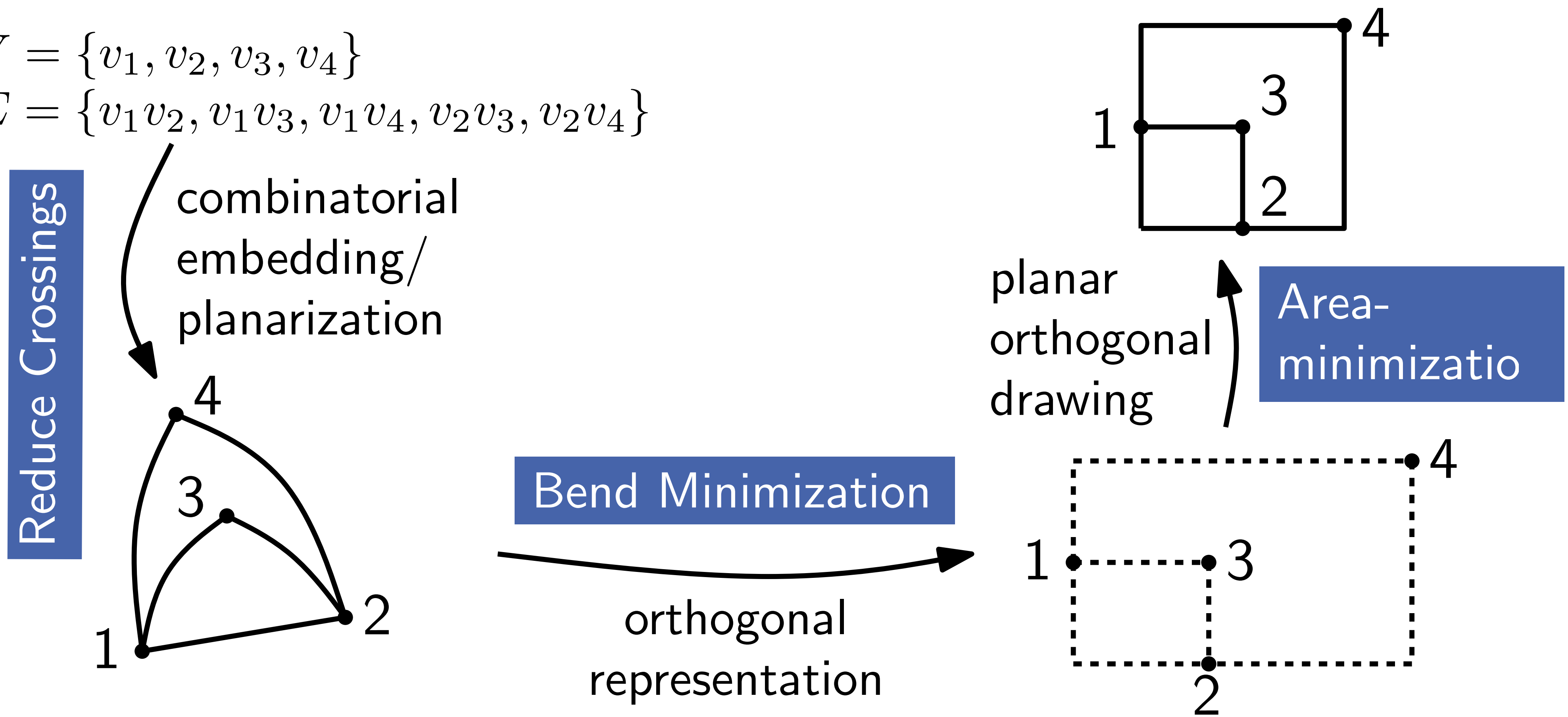
  [Cornelsen, Karrenbauer GD 2011]
- Bend minimization without a given combinatorial embedding is an NP-hard problem. [Garg, Tamassia SIAM J. Comput. 2001]

(Planare) Orthogonale Zeichnungen

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

[Tamassia SIAM J. Comput. 1987]

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



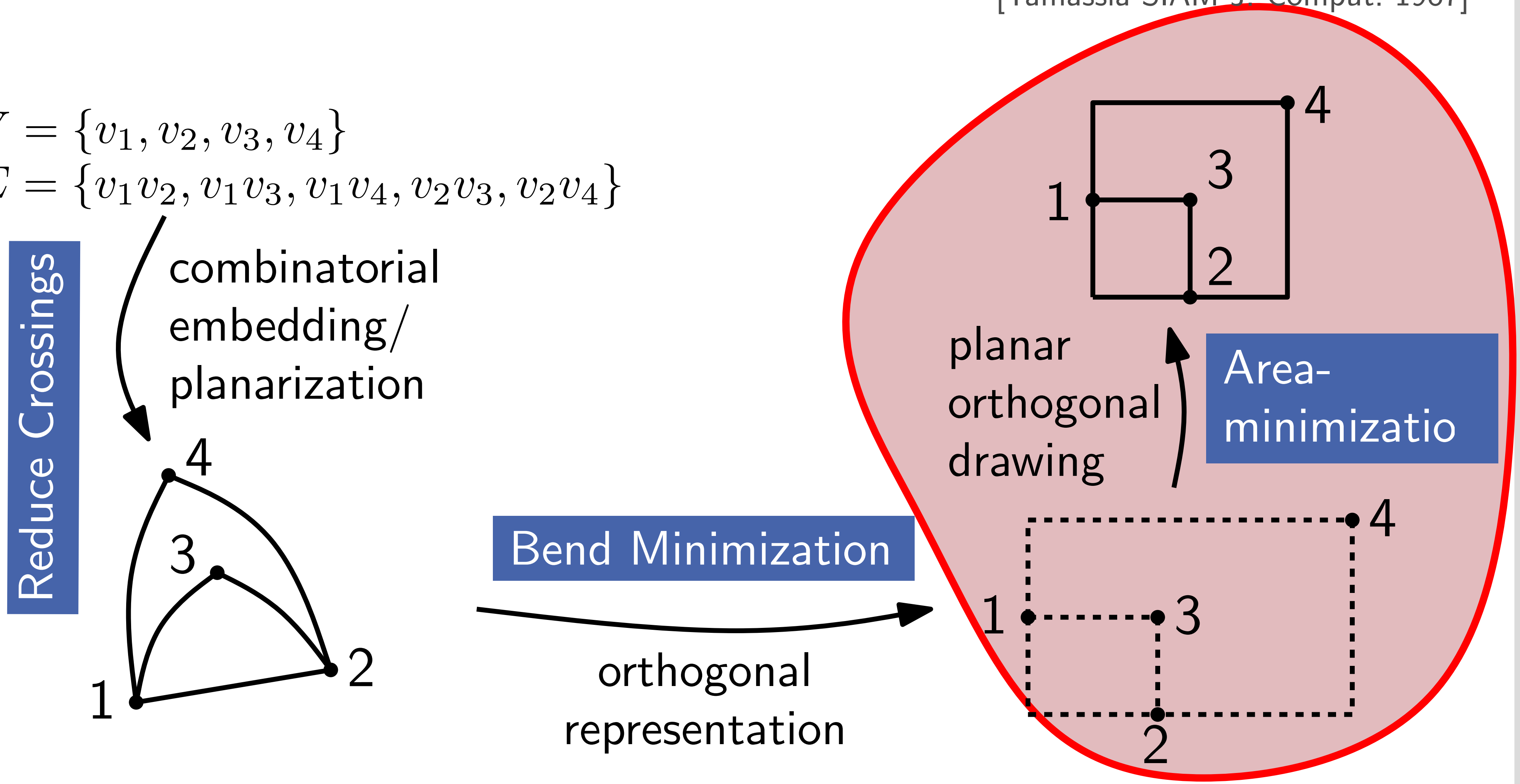
(Planare) Orthogonale Zeichnungen

Karlsruhe Institute of Technology

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

[Tamassia SIAM J. Comput. 1987]

$V = \{v_1, v_2, v_3, v_4\}$
$E = \{v_1 v_2, v_1 v_3, v_1 v_4, v_2 v_3, v_2 v_4\}$



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

$\rightarrow$ Guaratiees 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

$\rightarrow$ Guaratiees 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

$\rightarrow$ Guaratiees possible
- minimum total edge length
- minimum area

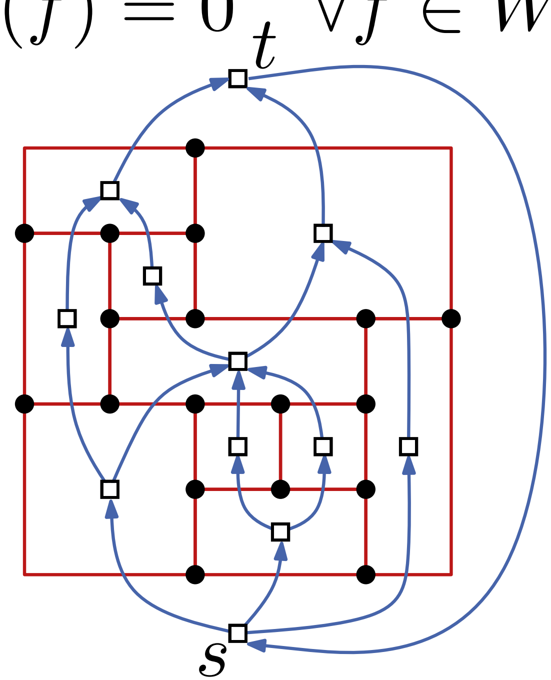
Properties:
- bends only on the outer face
- opposite sides of a face have the same length

We will formulate a flow network for (horizontal) compaction

Flow Network for Edge Length Computation

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

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

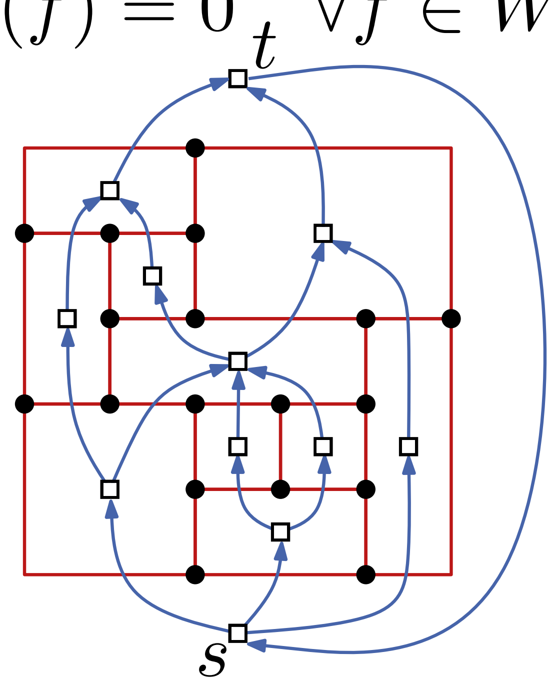


Flow Network for Edge Length Computation

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

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

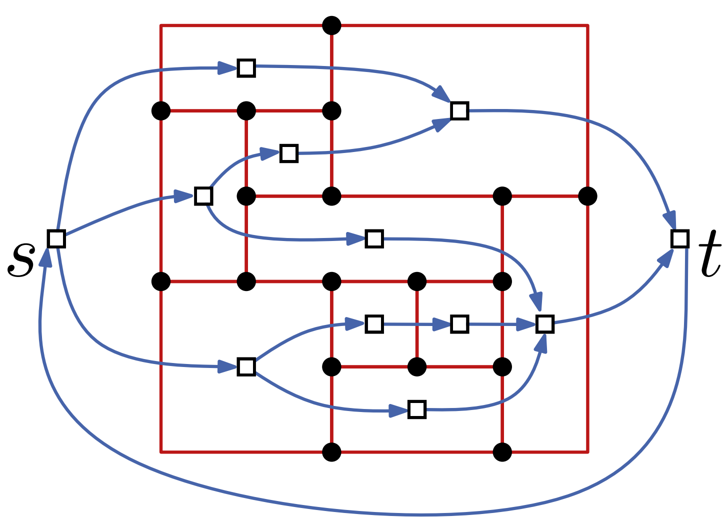
s and t represent lower and upper side of f_0



Flow Network for Edge Length Computation

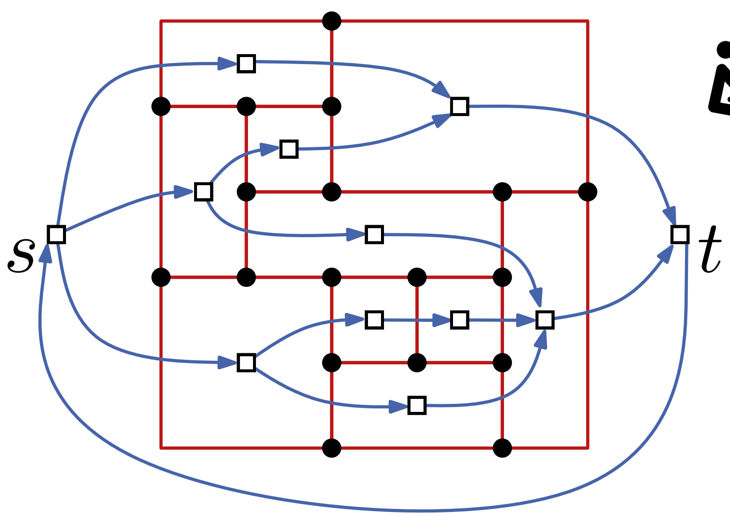
Def: Flow Network $N_{\text{ver}} = ((W_{\text{ver}}, A_{\text{ver}}); \ell; u; b; \text{cost})$

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



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

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

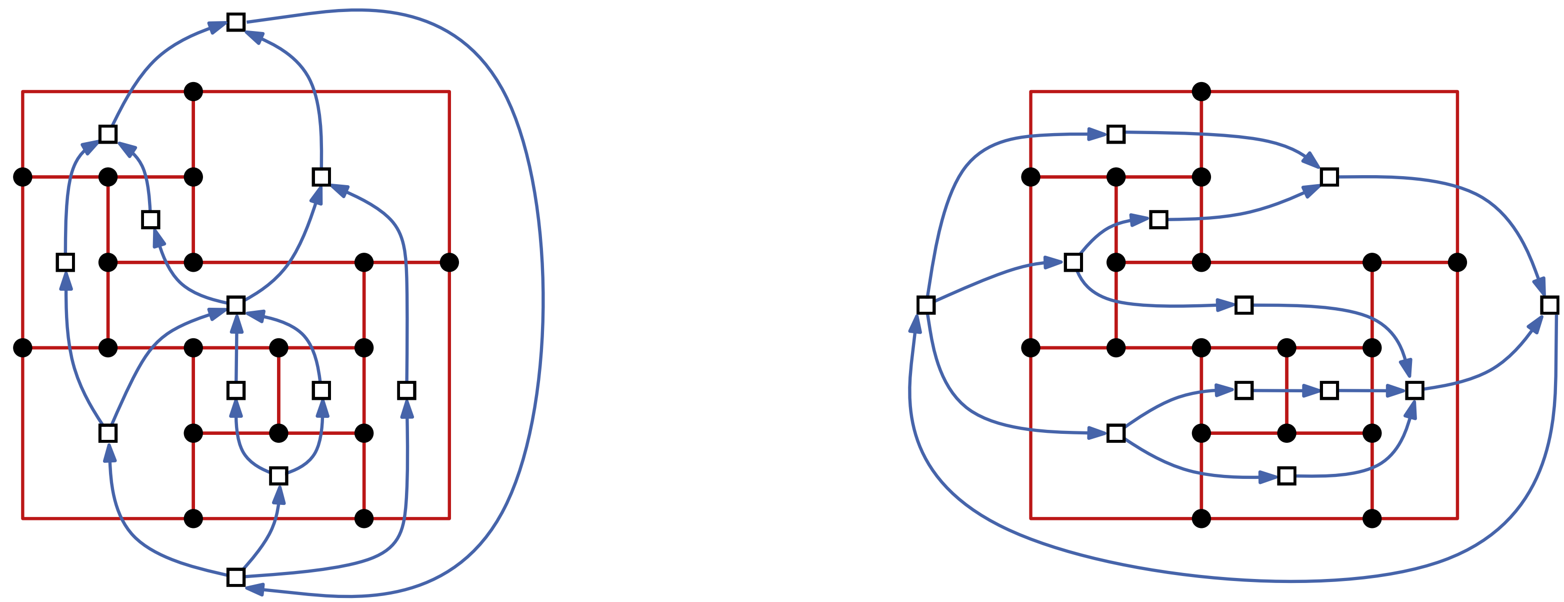


Pair, think, share: **3 min**

What values of the drawing represent the following?

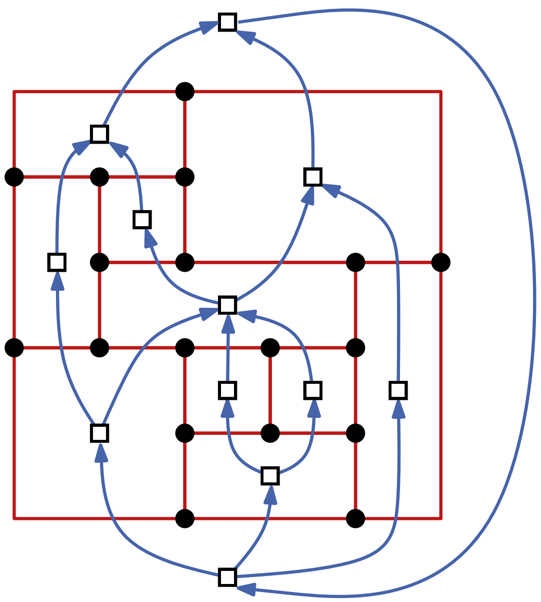
- $|x_{\text{hor}}(t, s)|$ and $|x_{\text{ver}}(t, s)|$?
- $\sum_{a \in A_{\text{hor}}} x_{\text{hor}}(a) + \sum_{a \in A_{\text{ver}}} x_{\text{ver}}(a)$

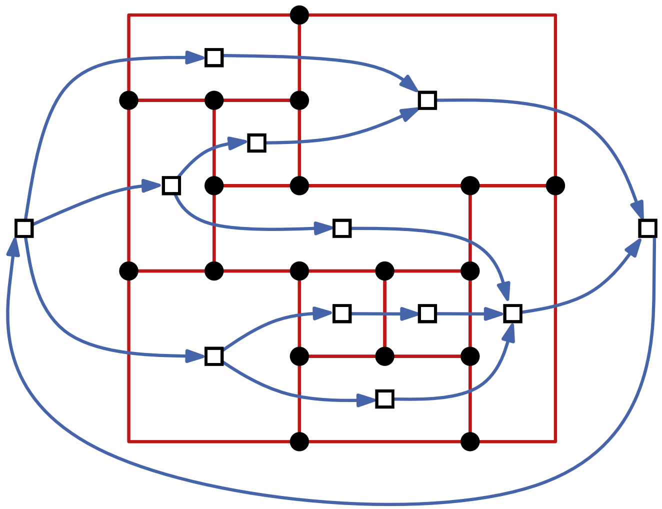
Optimal Layout



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

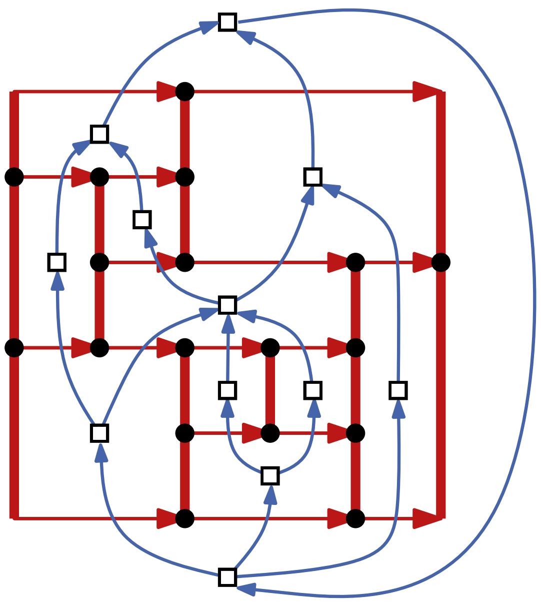
Faster Flow Computation

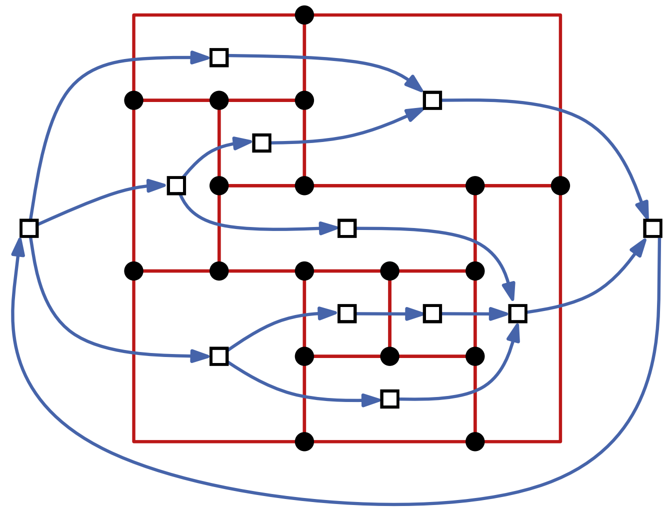




- construct the duals $N^{\star}_{hor}$ and $N^{\star}_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^{\star}_{hor}$ and $N^{\star}_{ver}$
- for edge (f, g) of N_{hor} set flow $x_{hor}(f, g) = T_{hor}(b) - T_{hor}(a)$, where b is dual vertex on the left and b is dual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

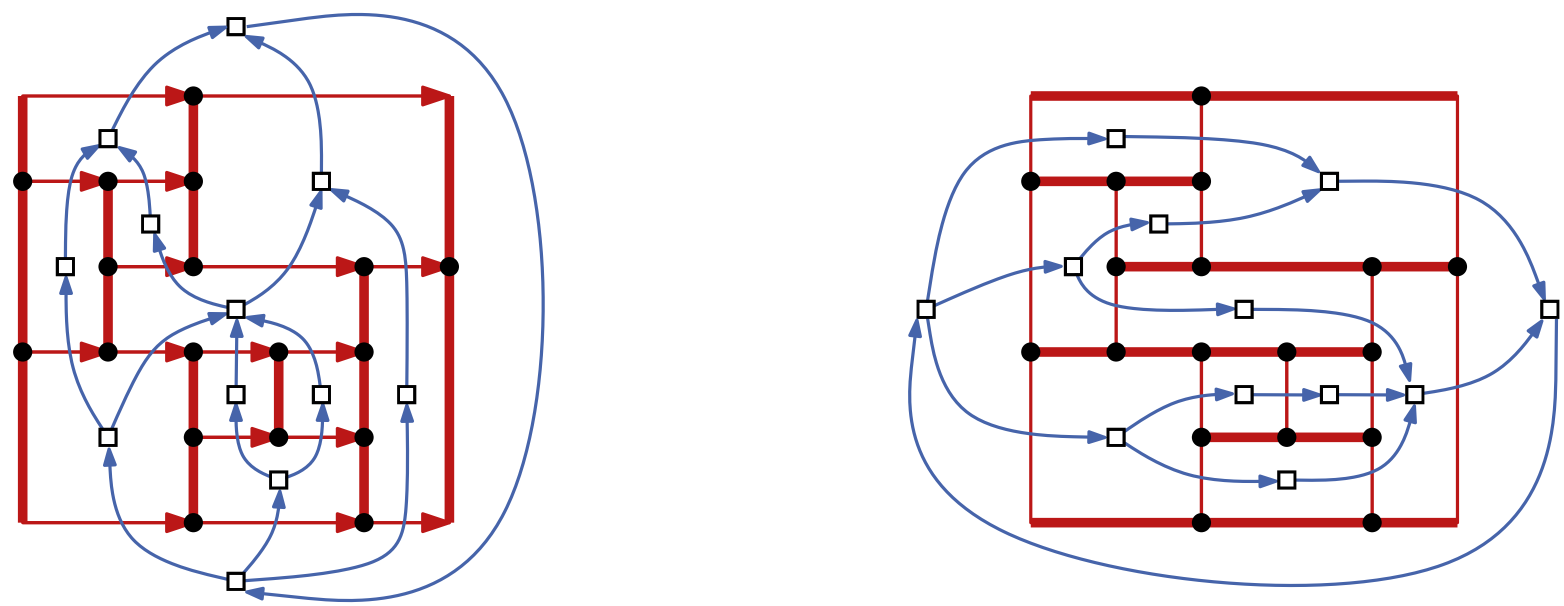
Faster Flow Computation





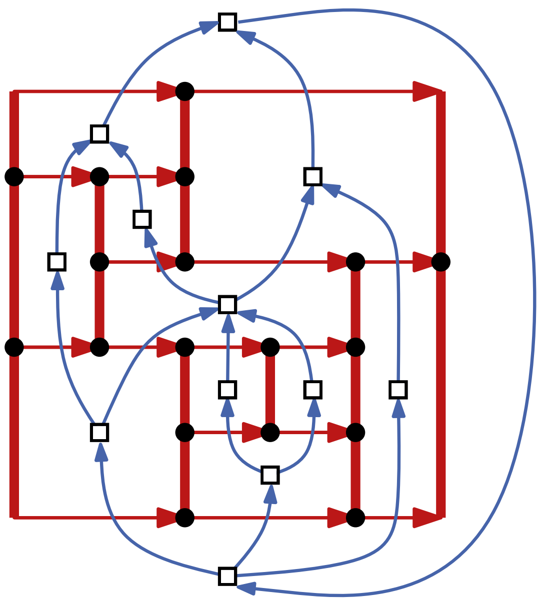
- construct the duals $N^{\star}_{hor}$ and $N^{\star}_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^{\star}_{hor}$ and $N^{\star}_{ver}$
- for edge (f, g) of N_{hor} set flow $x_{hor}(f, g) = T_{hor}(b) - T_{hor}(a)$, where b is dual vertex on the left and b is dual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

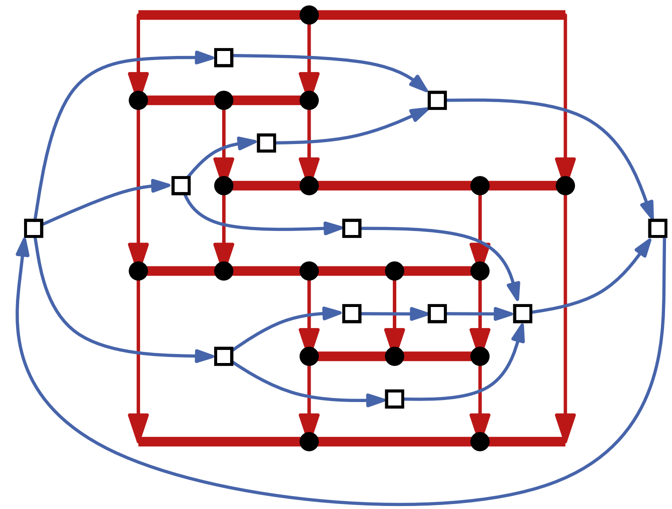
Faster Flow Computation



- construct the duals $N^\star_{hor}$ and $N^\star_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^\star_{hor}$ and $N^\star_{ver}$
- 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 minumun value

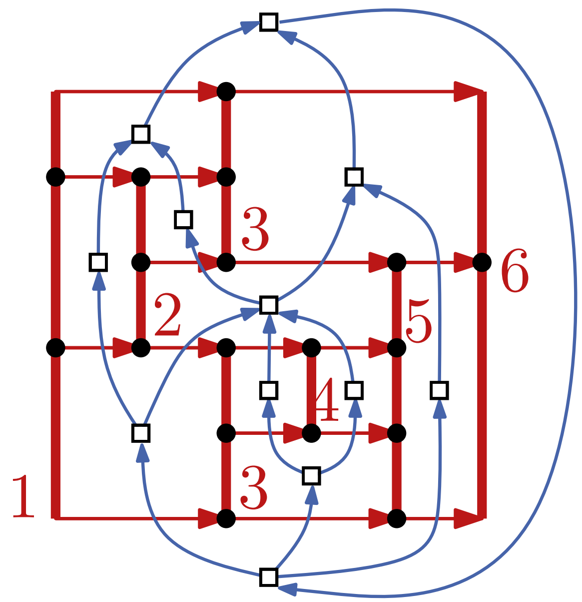
Faster Flow Computation

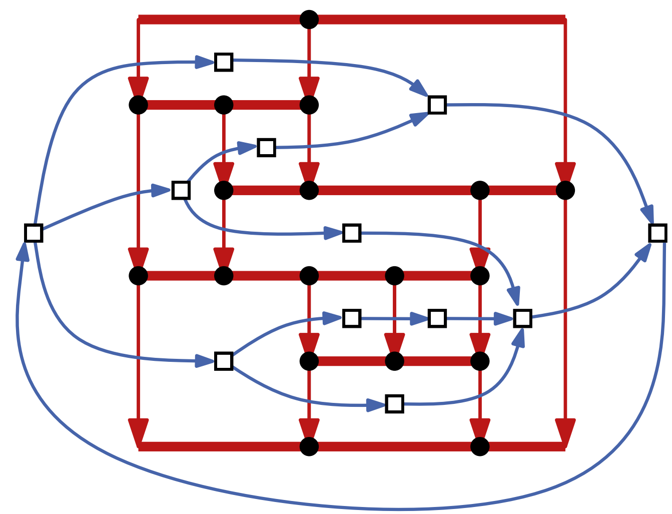




- construct the duals $N^{\star}_{hor}$ and $N^{\star}_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^{\star}_{hor}$ and $N^{\star}_{ver}$
- for edge (f, g) of N_{hor} set flow $x_{hor}(f, g) = T_{hor}(b) - T_{hor}(a)$, where b is dual vertex on the left and b is dual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

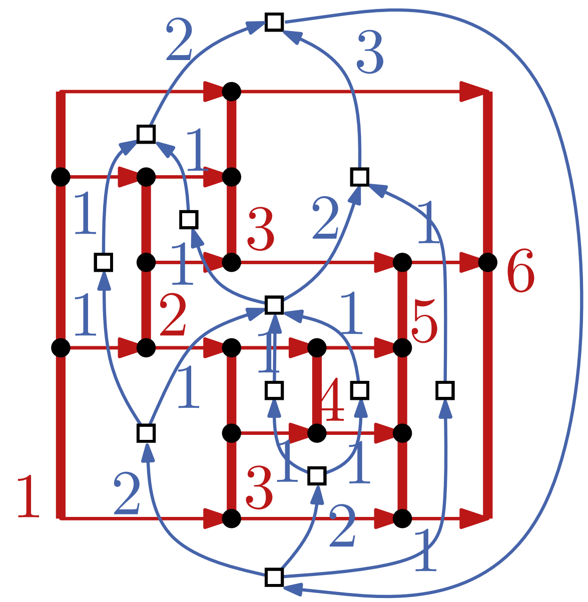
Faster Flow Computation

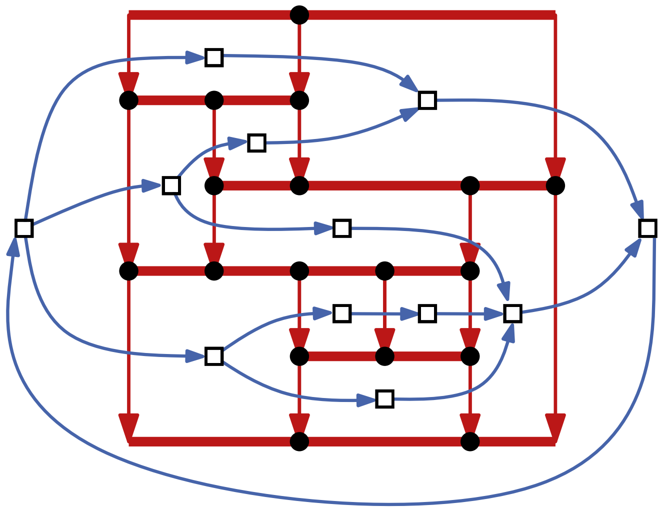




- construct the duals $N^\star_{hor}$ and $N^\star_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^\star_{hor}$ and $N^\star_{ver}$
- for edge (f,g) of N_{hor} set flow $x_{hor}(f,g) = T_{hor}(b) - T_{hor}(a)$, where b is dual vertex on the left and b is dual vertex on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

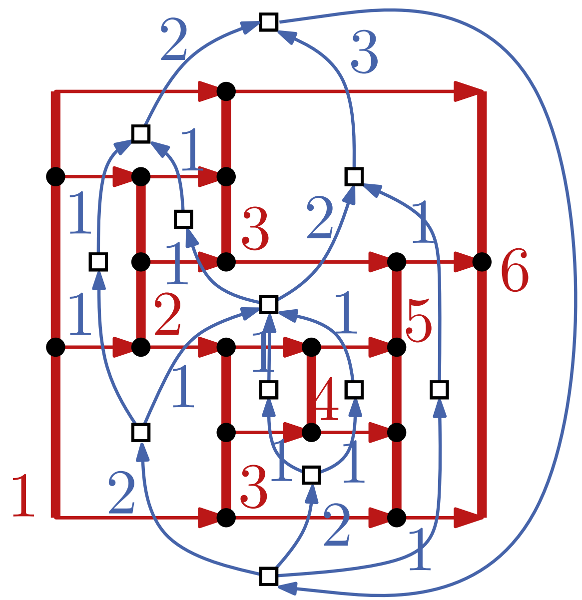
Faster Flow Computation

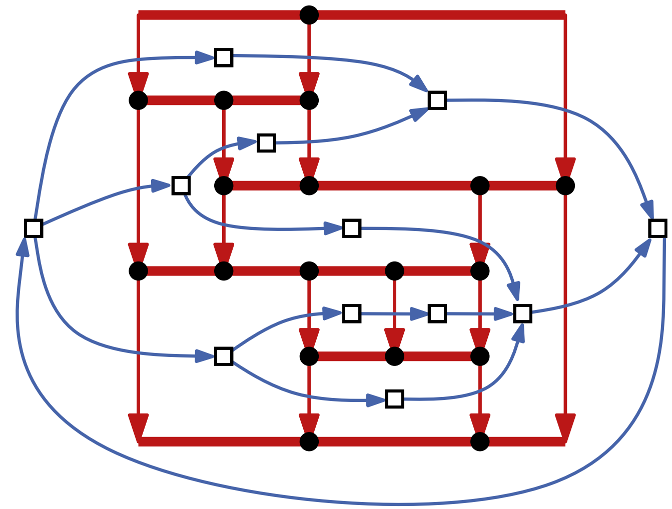




- construct the duals $N^{\star}_{hor}$ and $N^{\star}_{ver}$ of N_{hor} and N_{ver}
- topologial numbering T_{hor} and T_{ver} of $N^{\star}_{hor}$ and $N^{\star}_{ver}$
- for edge (f, g) of N_{hor} set flow $x_{hor}(f, g) = T_{hor}(b) - T_{hor}(a)$, where b is dual vertex on the left and b is dual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

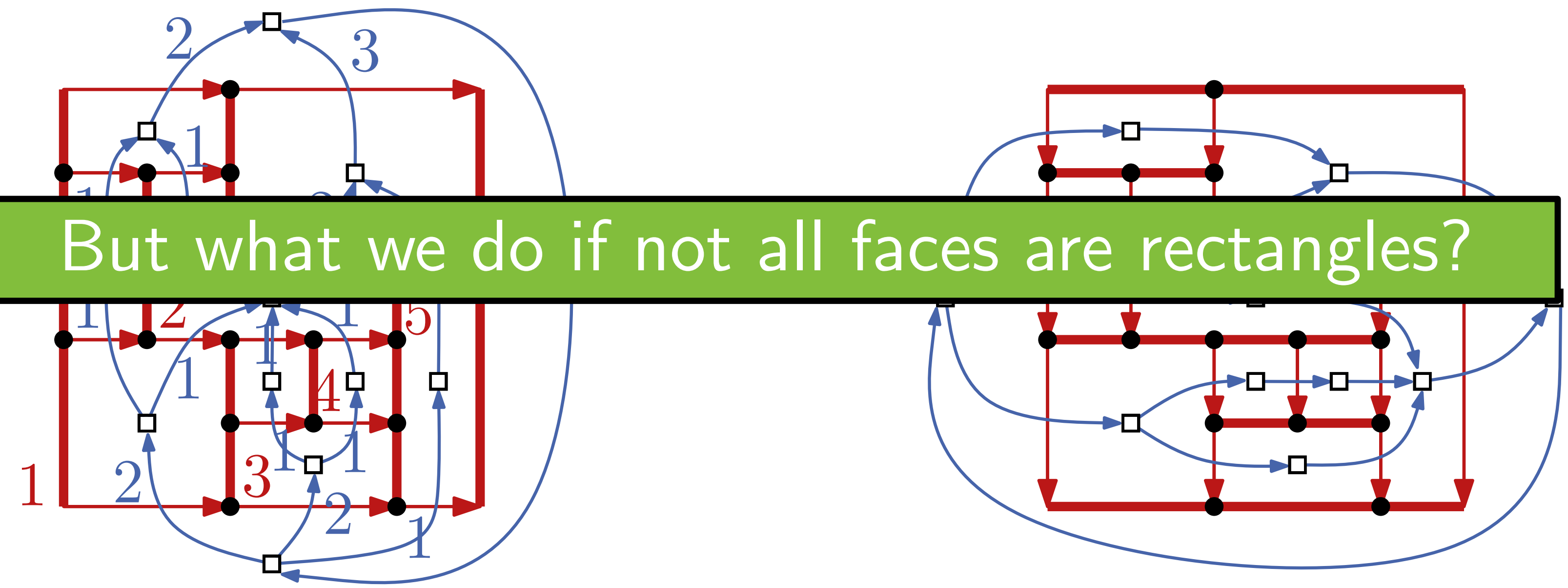
Faster Flow Computation





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

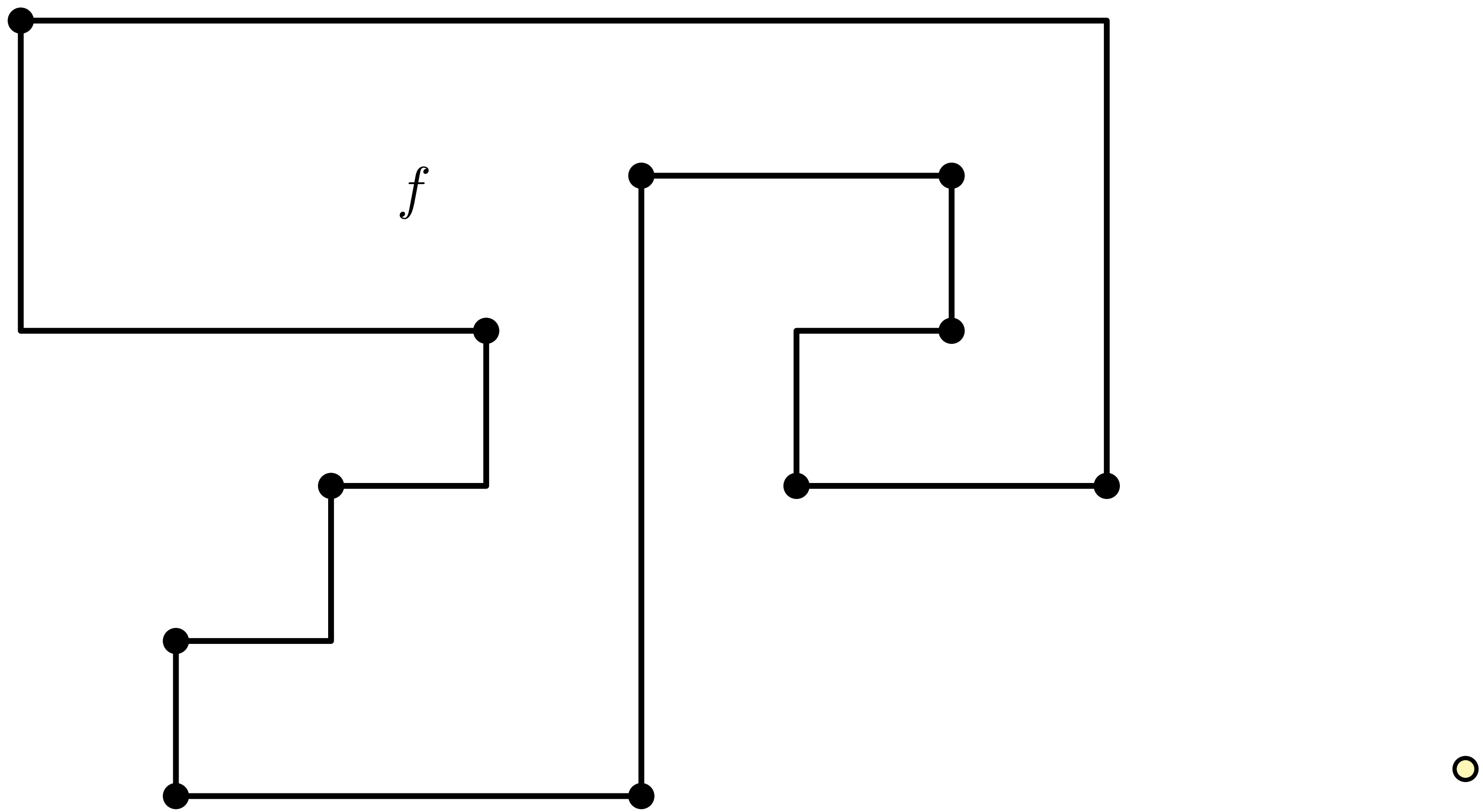
Faster Flow Computation



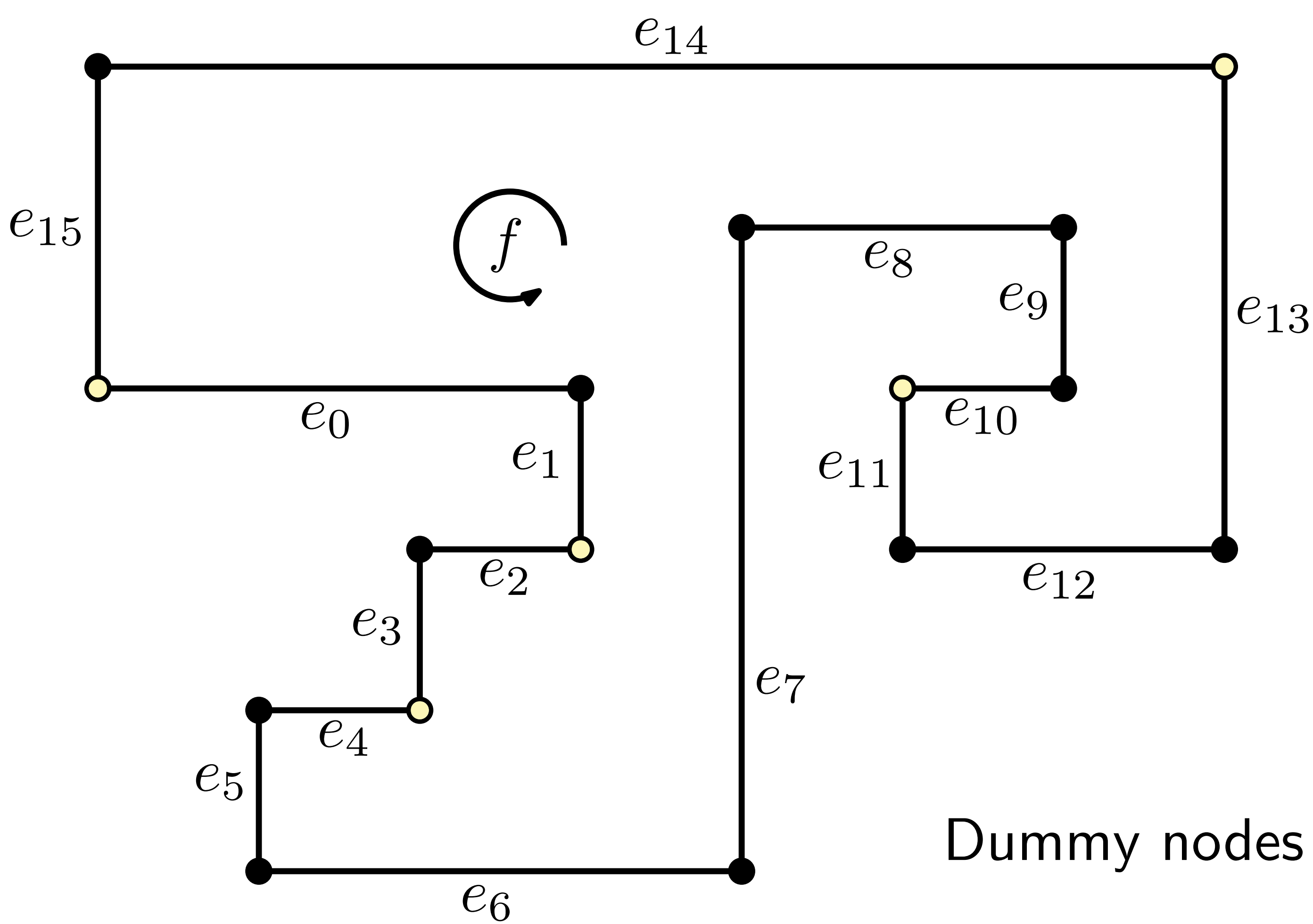
But what we do if not all faces are rectangles?

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

Refinement of (G, H) – Innner Face

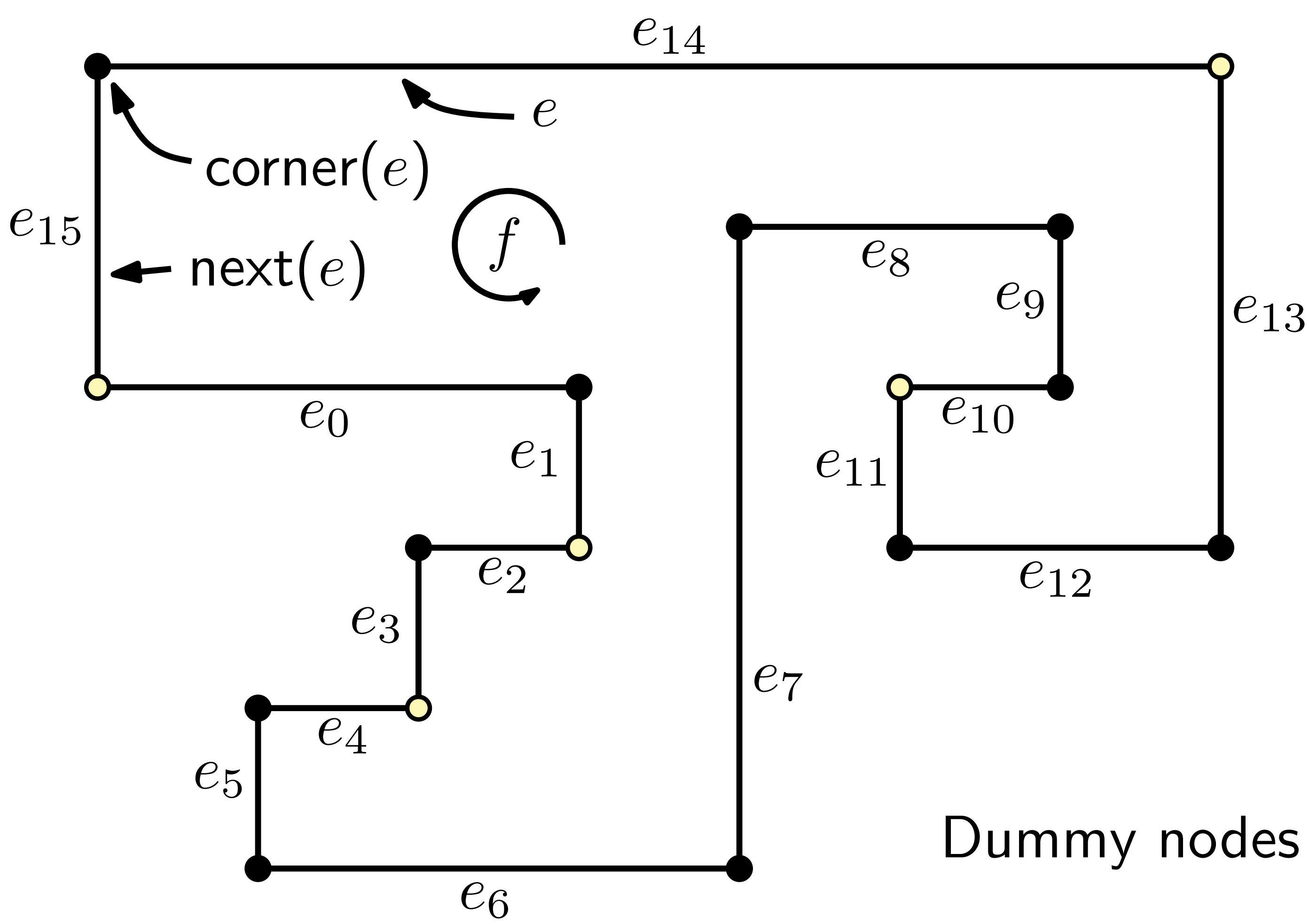


Refinement of (G, H) – Innner Face



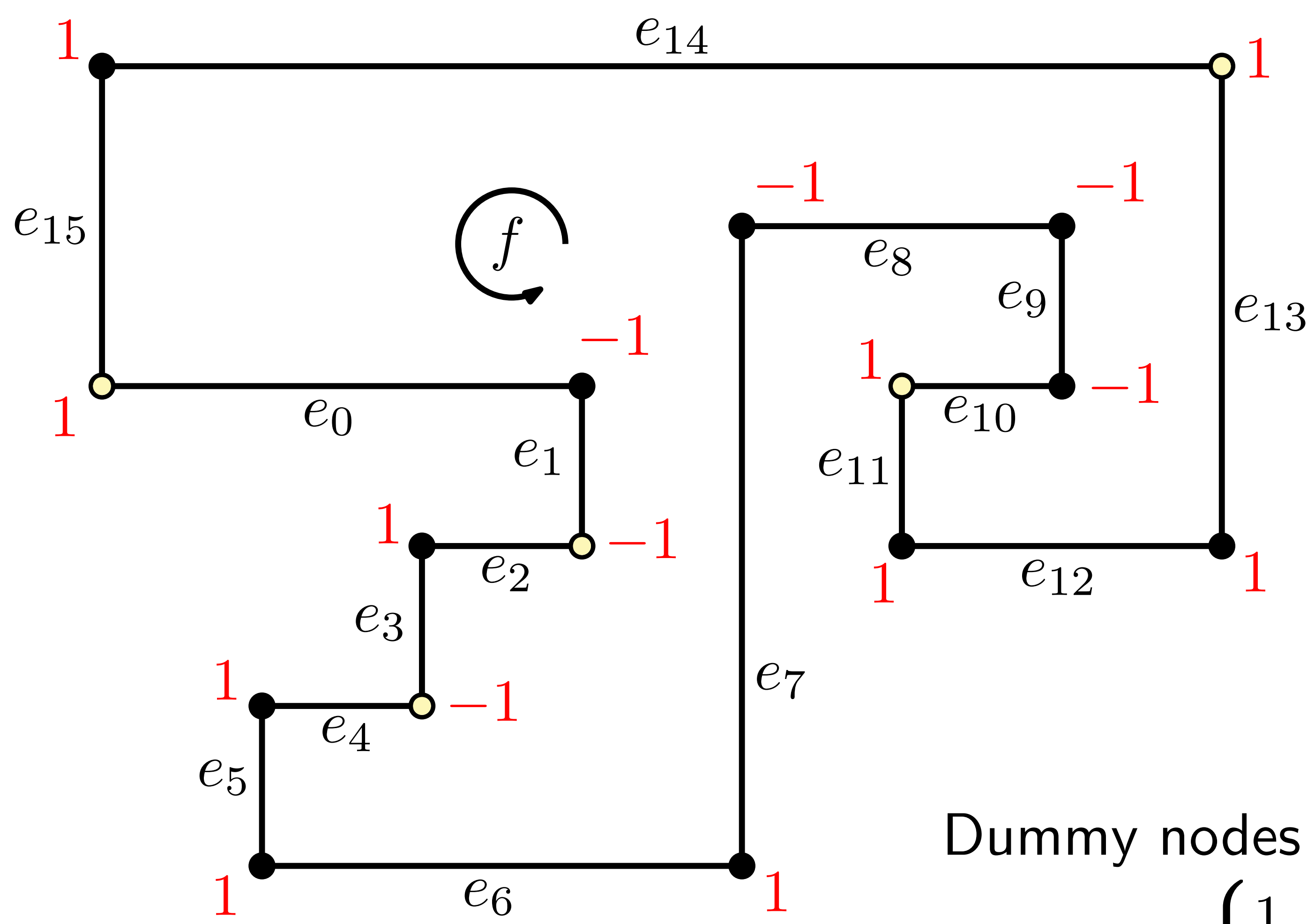
Dummy nodes for bends:

Refinement of (G, H) – Innner Face



Dummy nodes for bends:

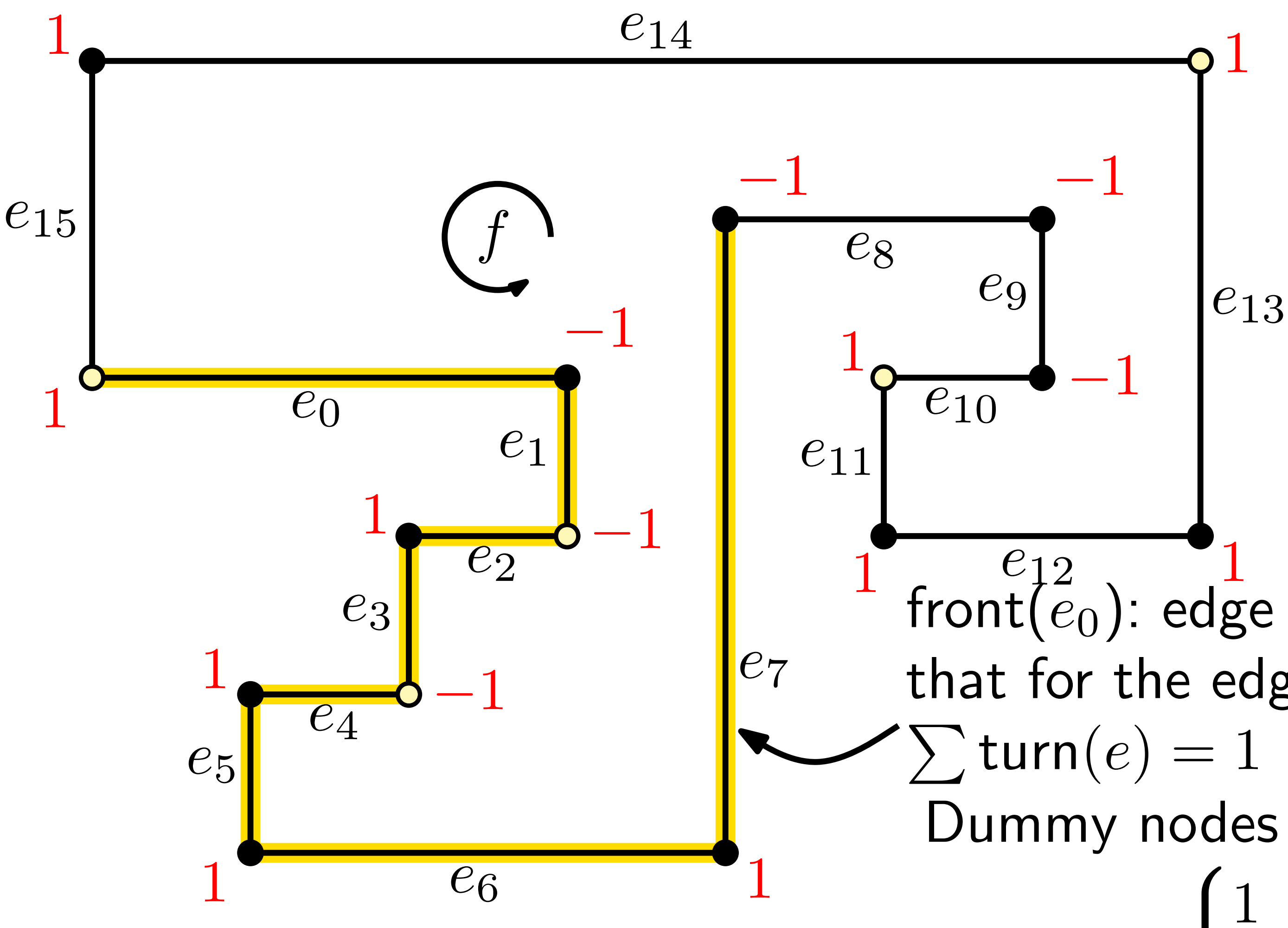
Refinement of (G, H) – Innner Face



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) – Innner Face

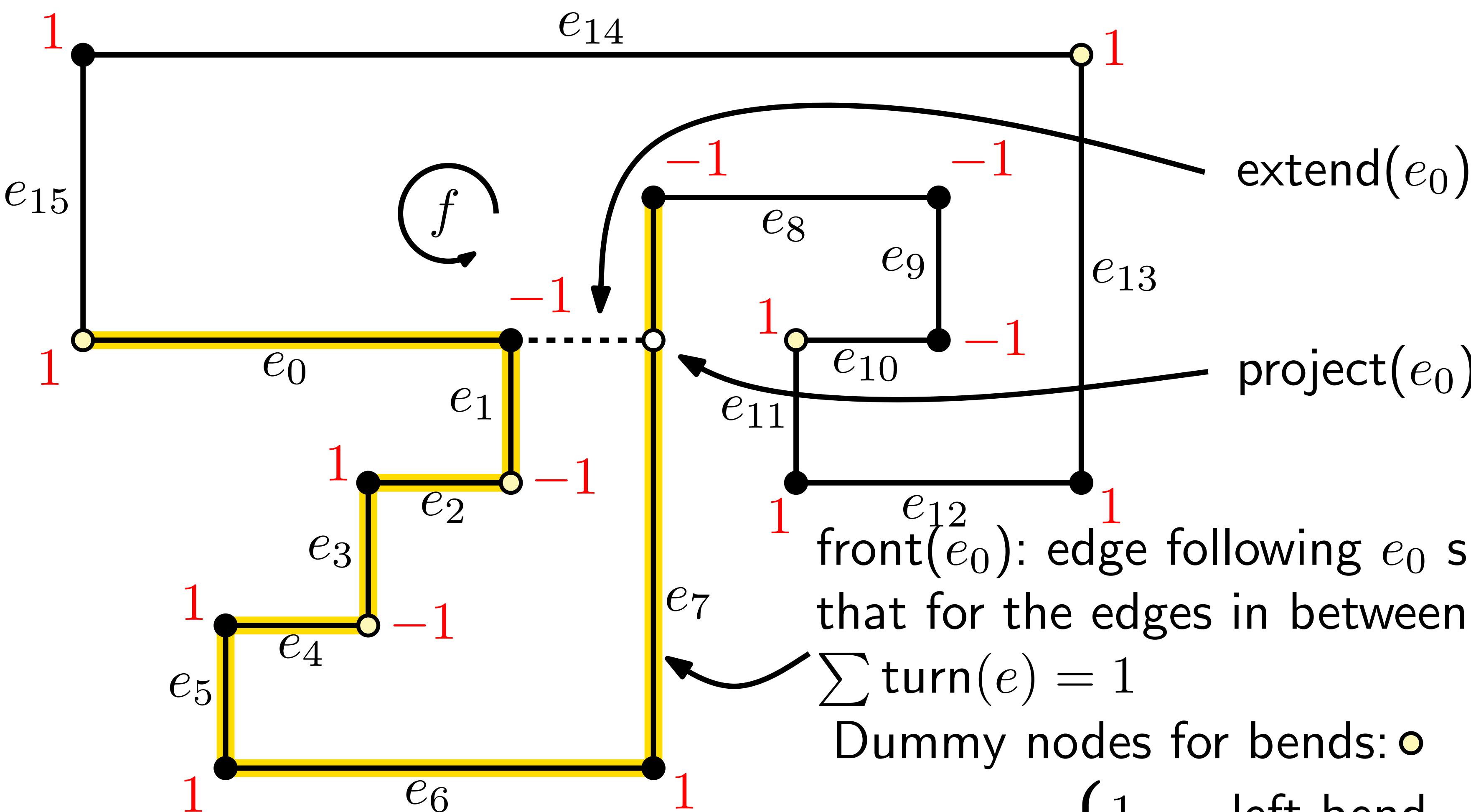


front(e_0): edge following e_0 such that for the edges in between $\sum \text{turn}(e) = 1$

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) – Innner Face

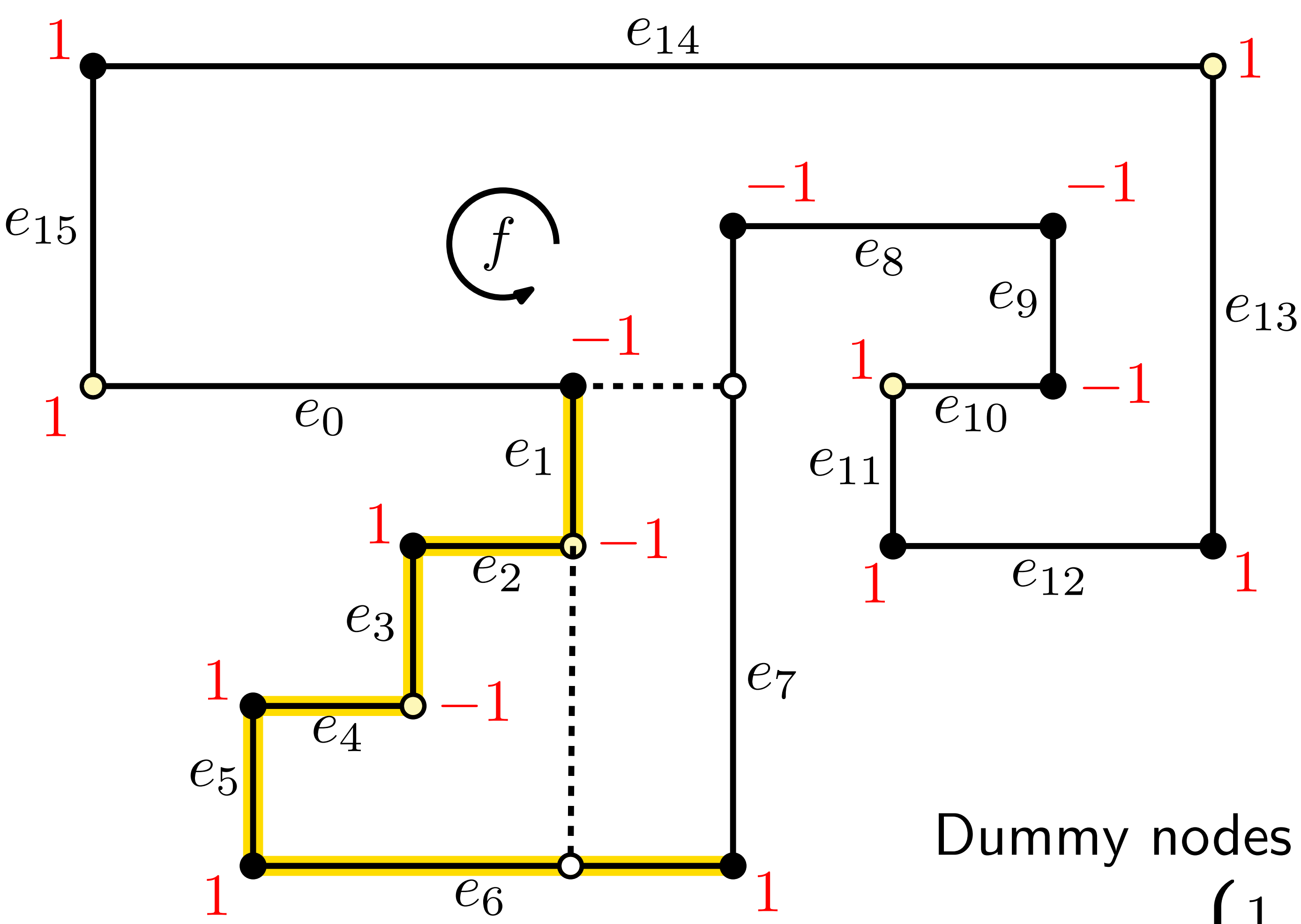


front(e_0): edge following e_0 such that for the edges in between $\sum \text{turn}(e) = 1$

Dummy nodes for bends: ○

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

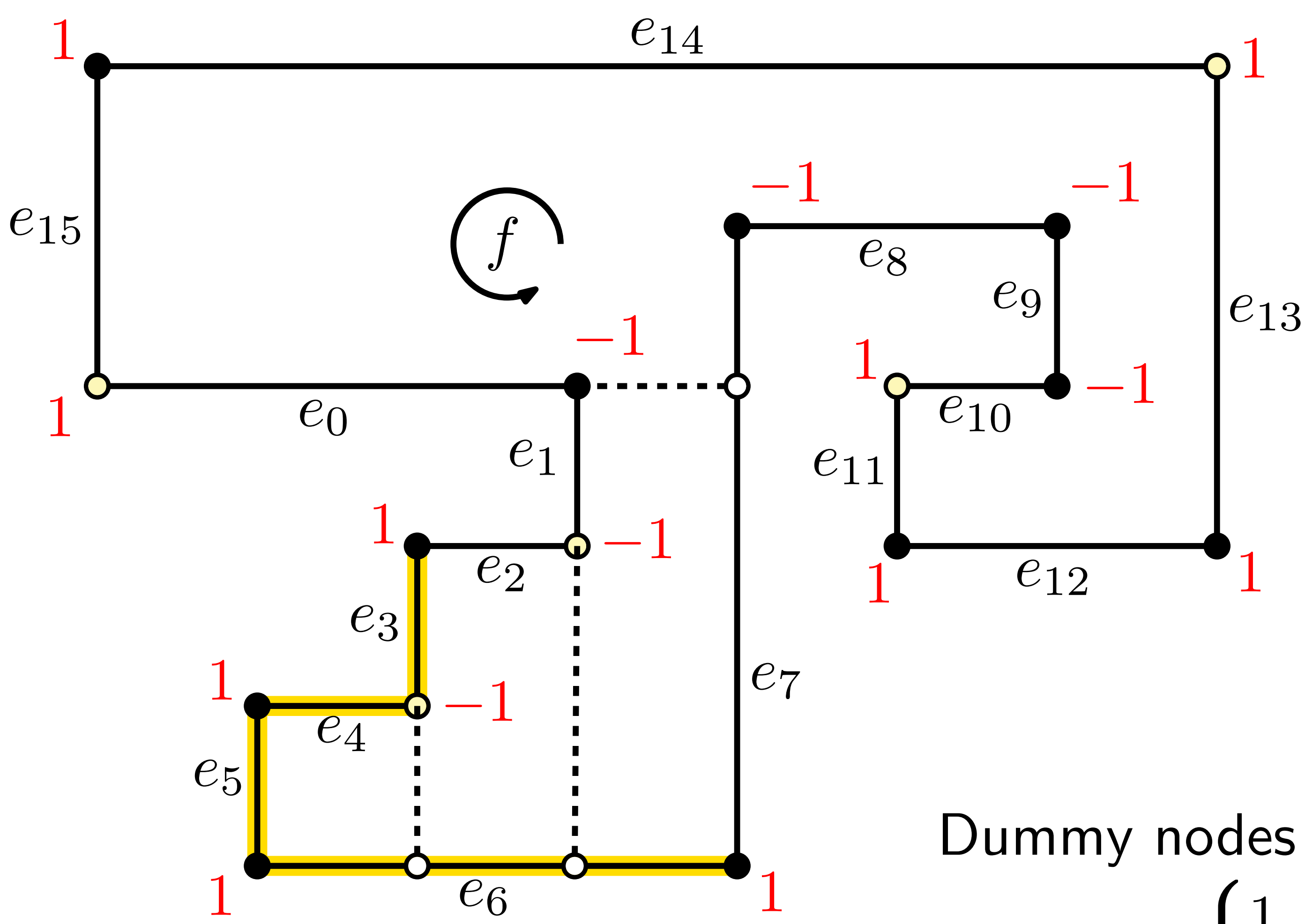
Refinement of (G, H) – Innner Face



Dummy nodes for bends:

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

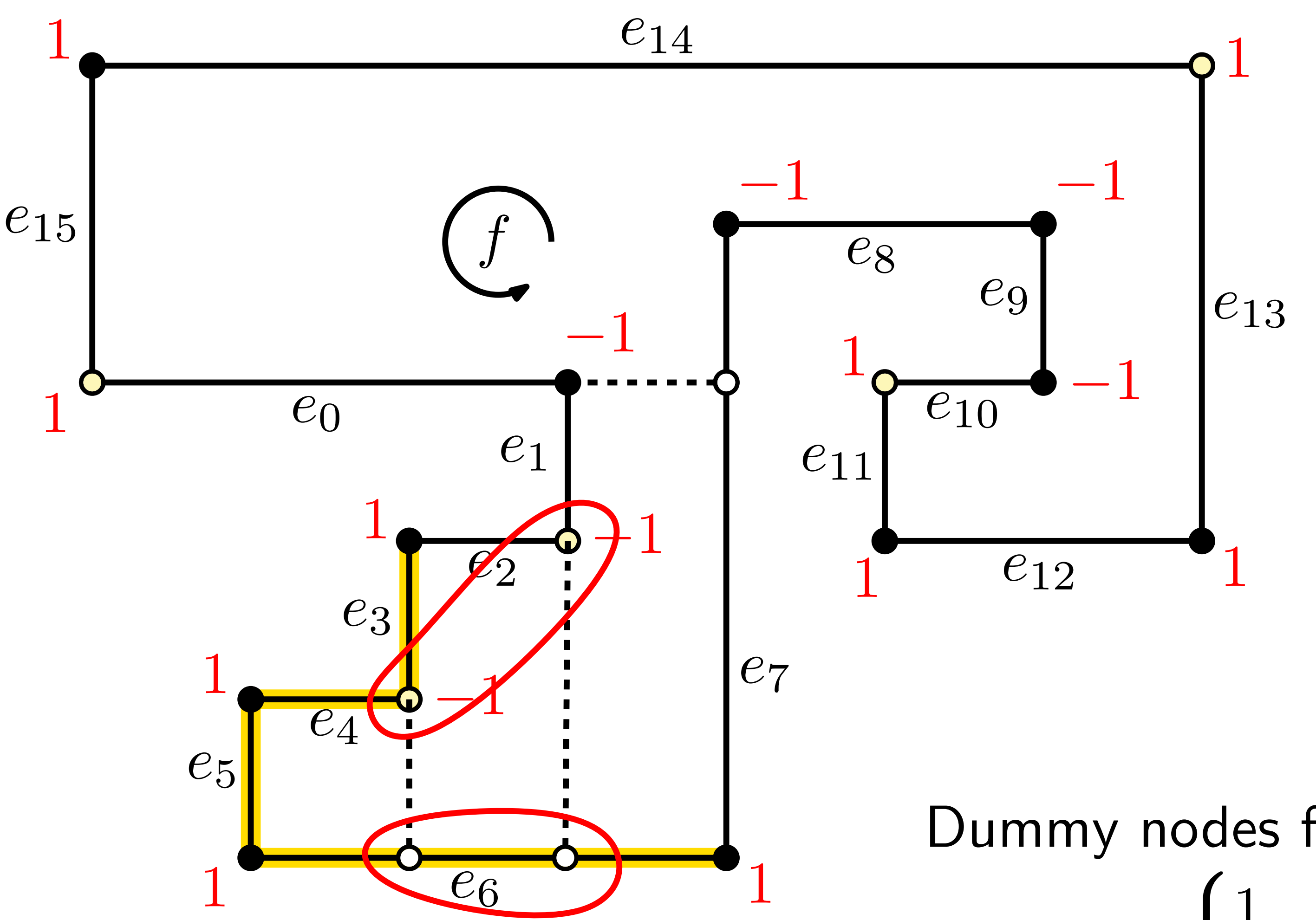
Refinement of (G, H) – Innner Face



Dummy nodes for bends:

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

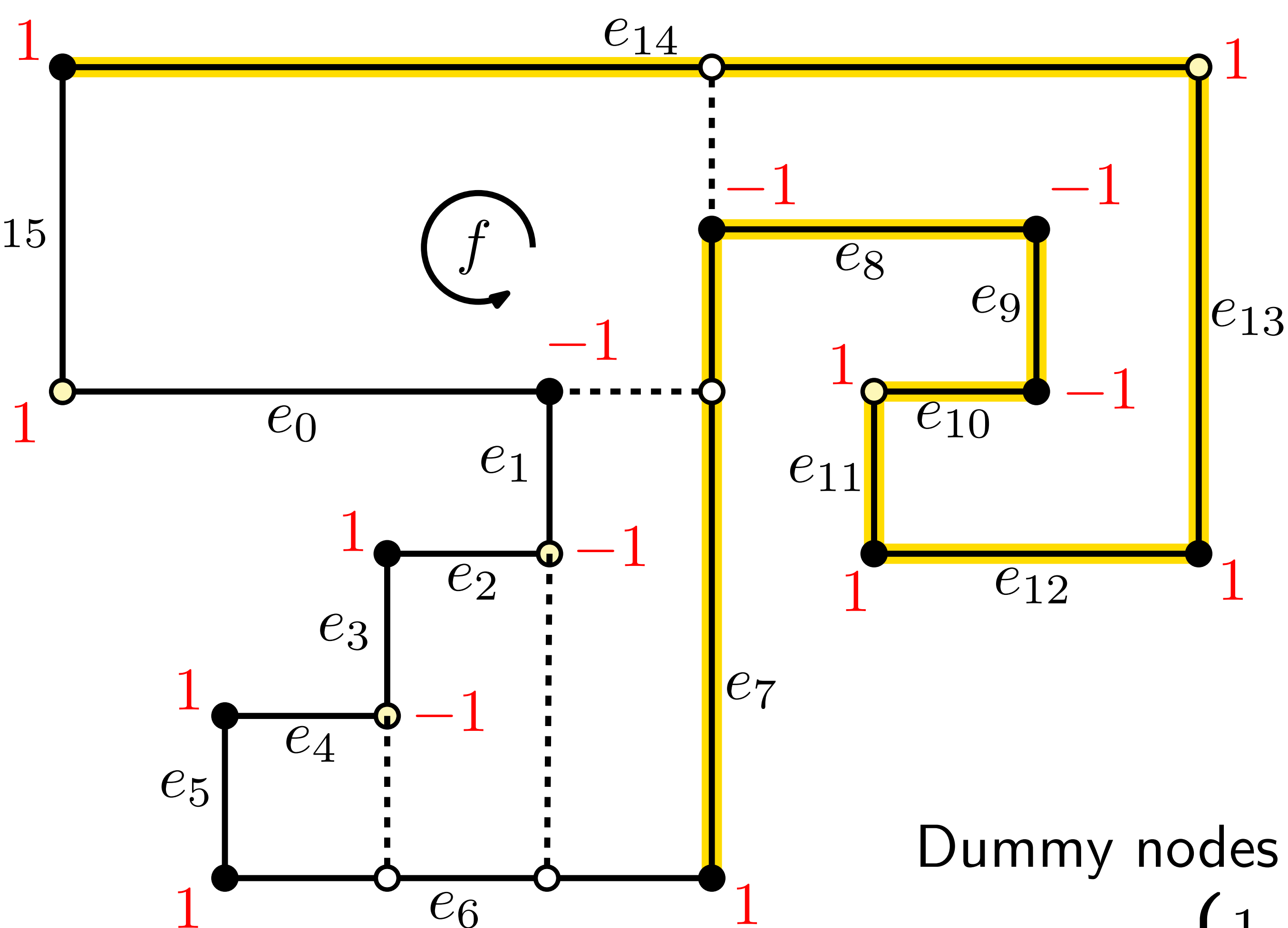
Refinement of (G, H) – Innner Face



Dummy nodes for bends: ○

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

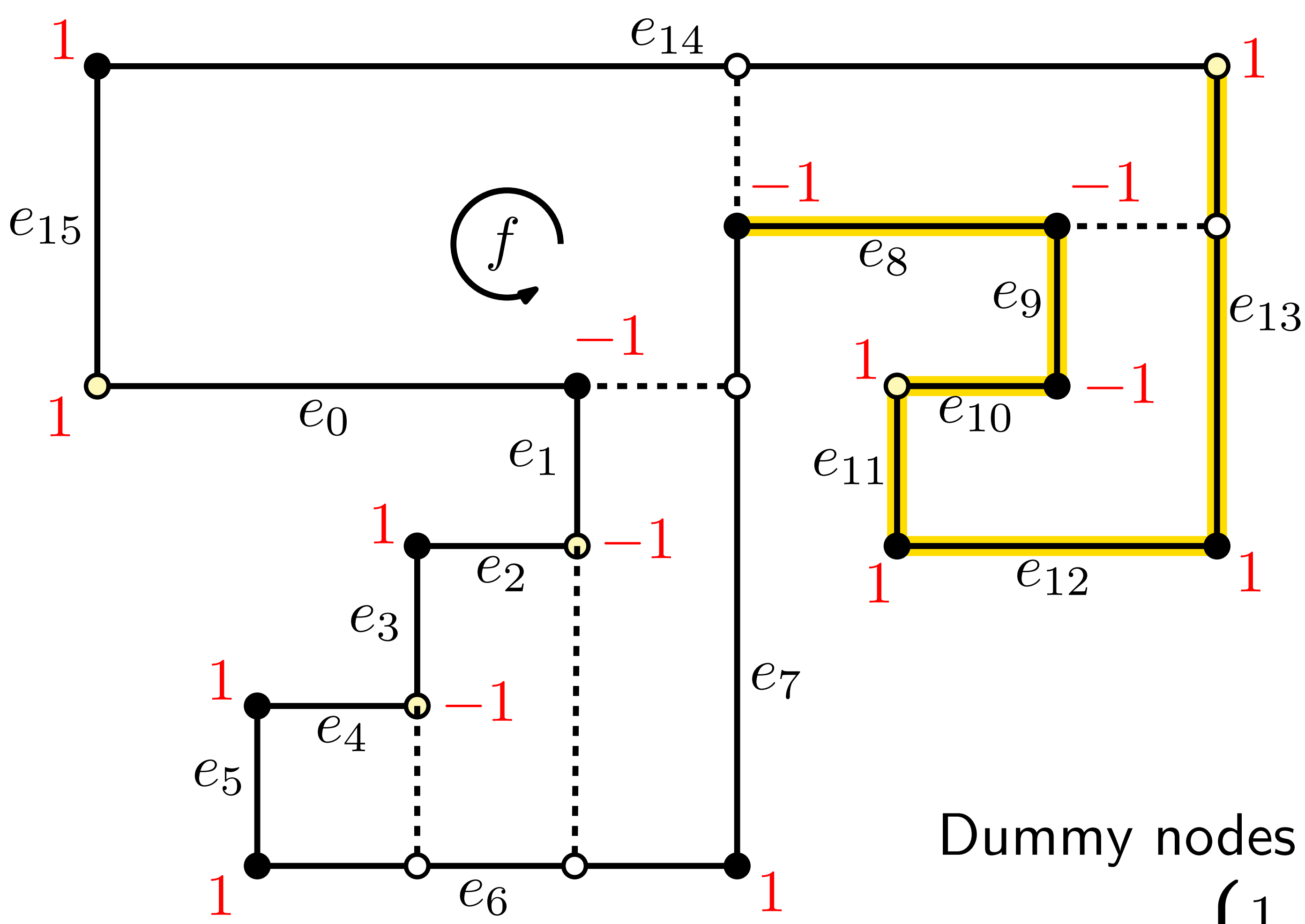
Refinement of (G, H) – Innner Face



Dummy nodes for bends:

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

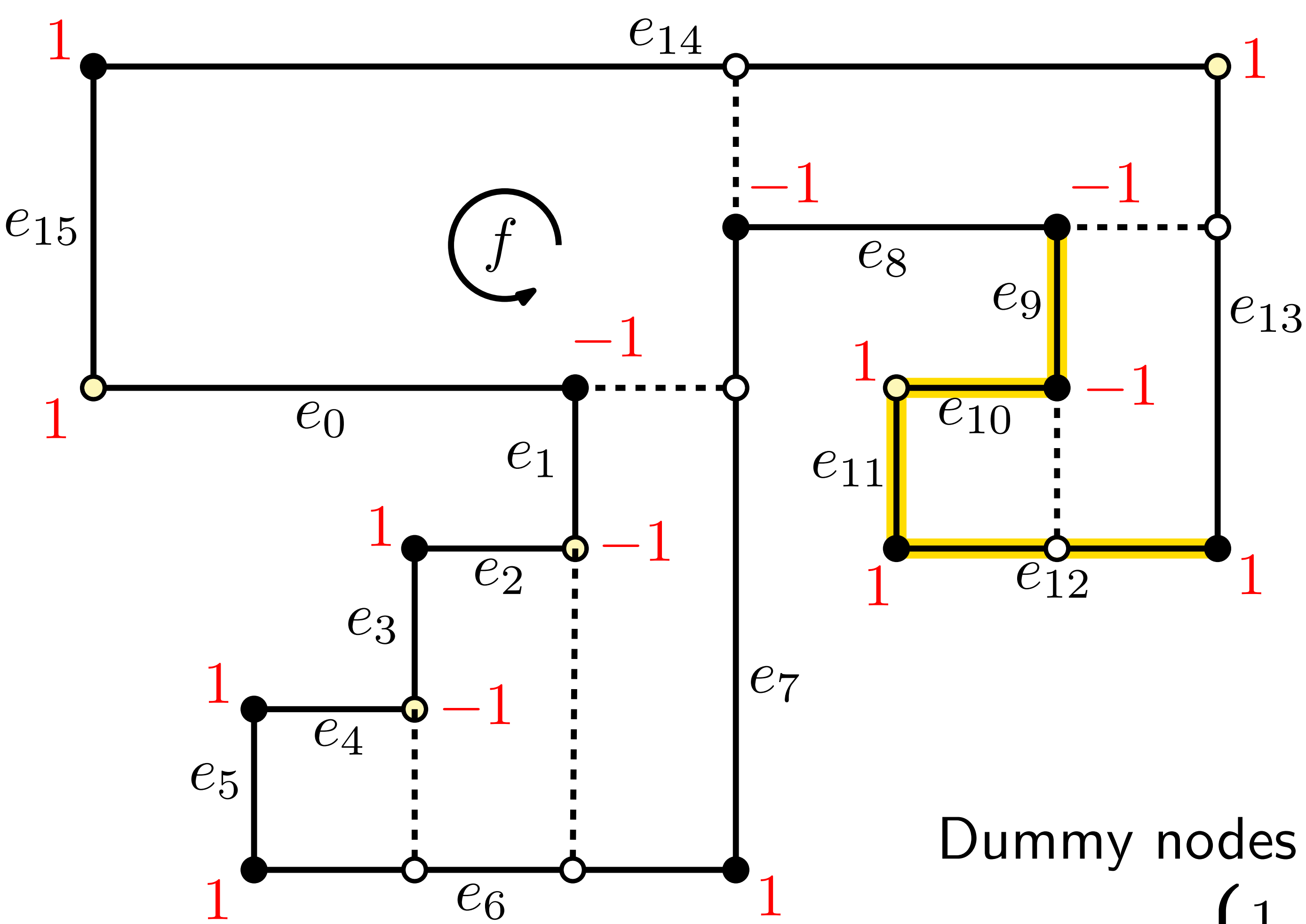
Refinement of (G, H) – Innner Face



Dummy nodes for bends: ○

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

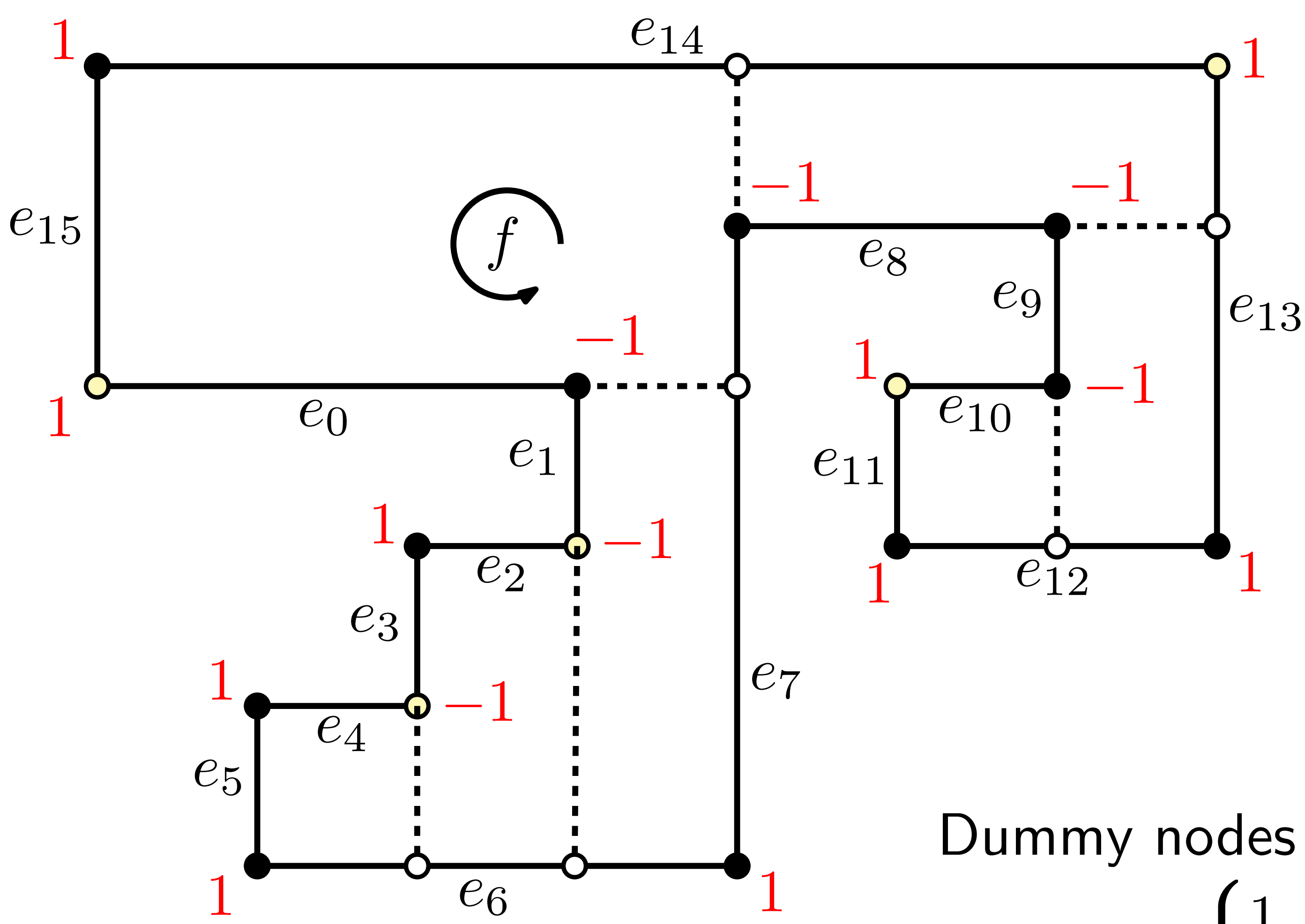
Refinement of (G, H) – Innner Face



Dummy nodes for bends: ○

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

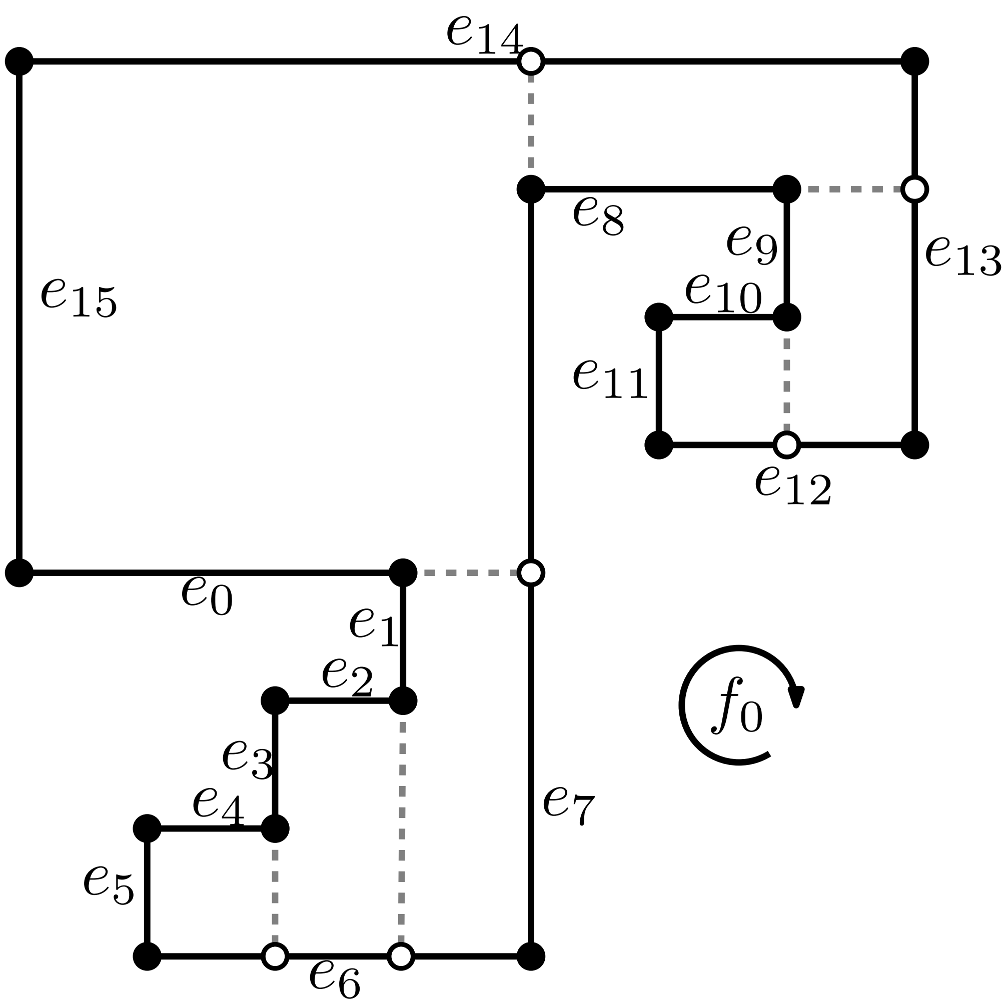
Refinement of (G, H) – Innner Face



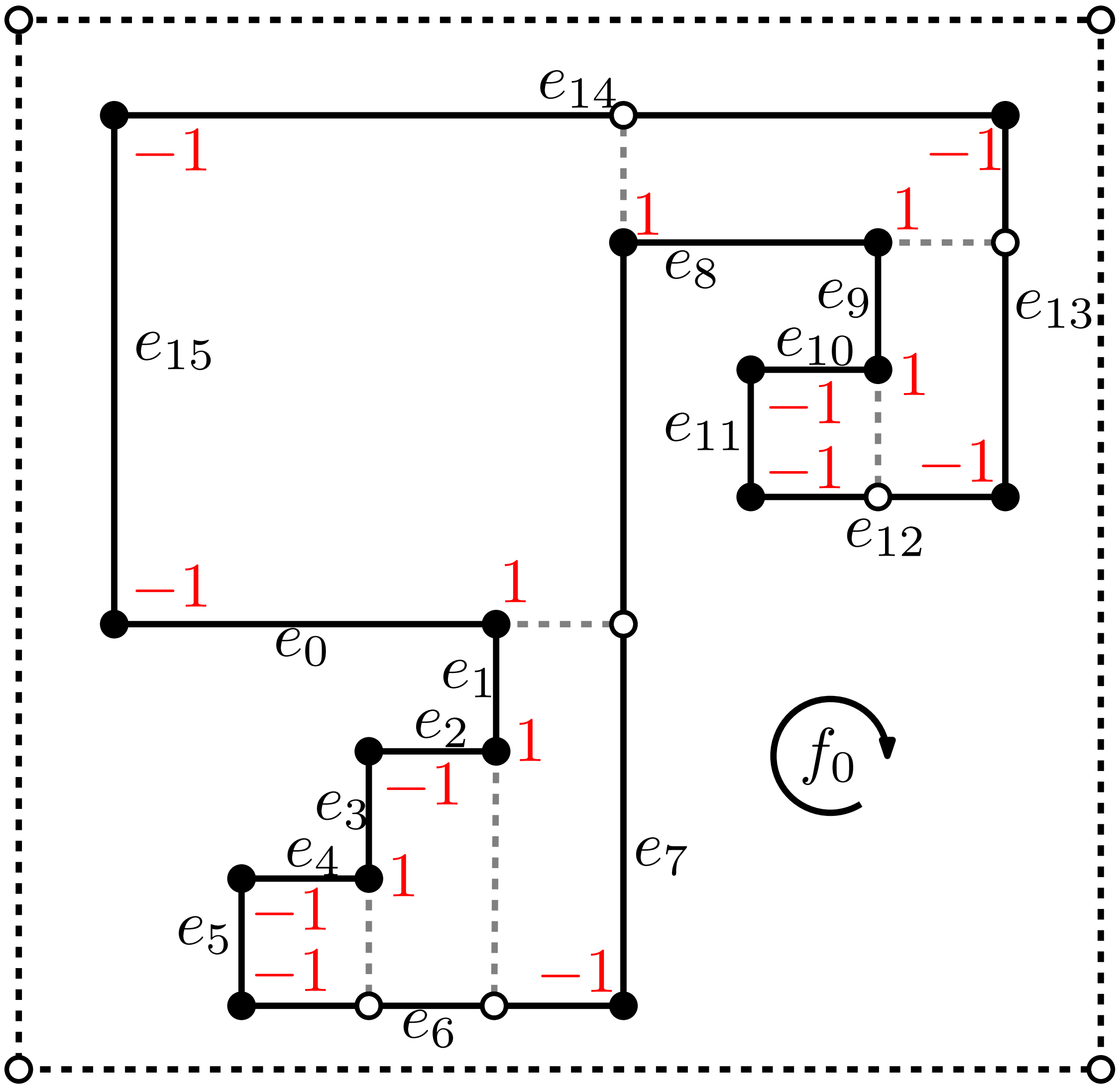
Dummy nodes for bends: ○

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

Refinement of (G, H) – Outer Face



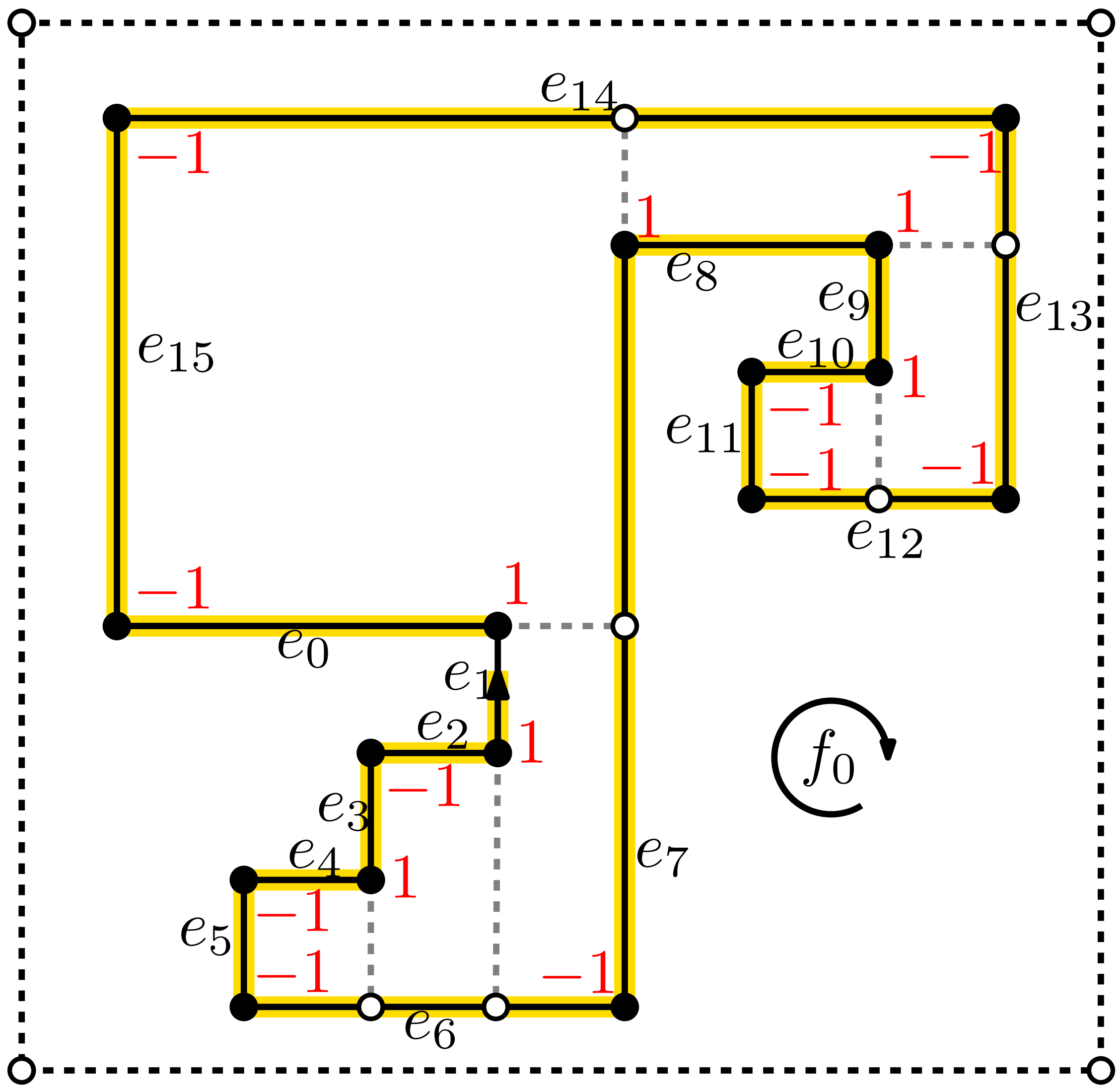
Refinement of (G, H) – Outer Face



- front(e) may be undefined

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

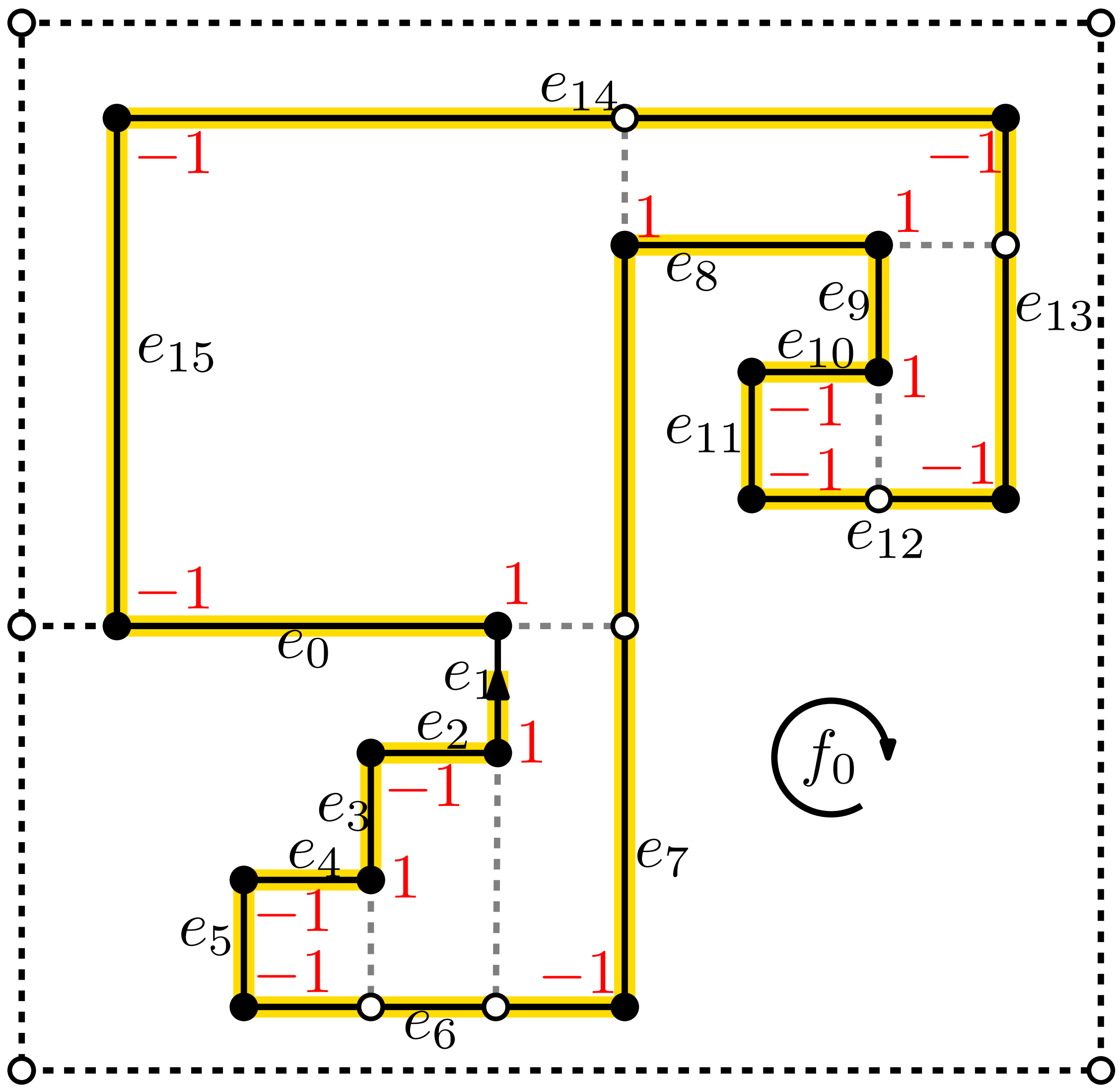
Refinement of (G, H) – Outer Face



- front(e) may be undefined

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

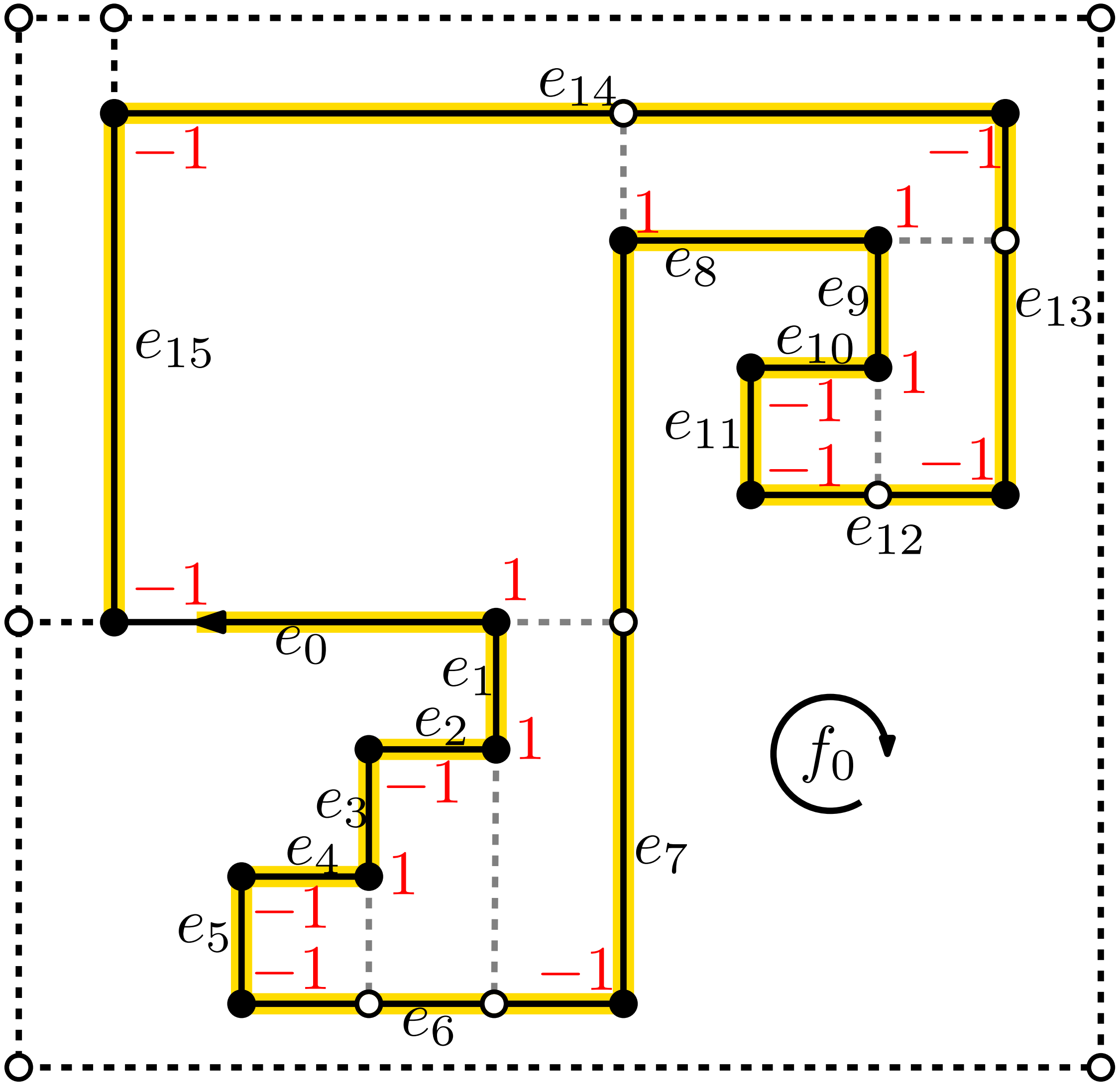
Refinement of (G, H) – Outer Face



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

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

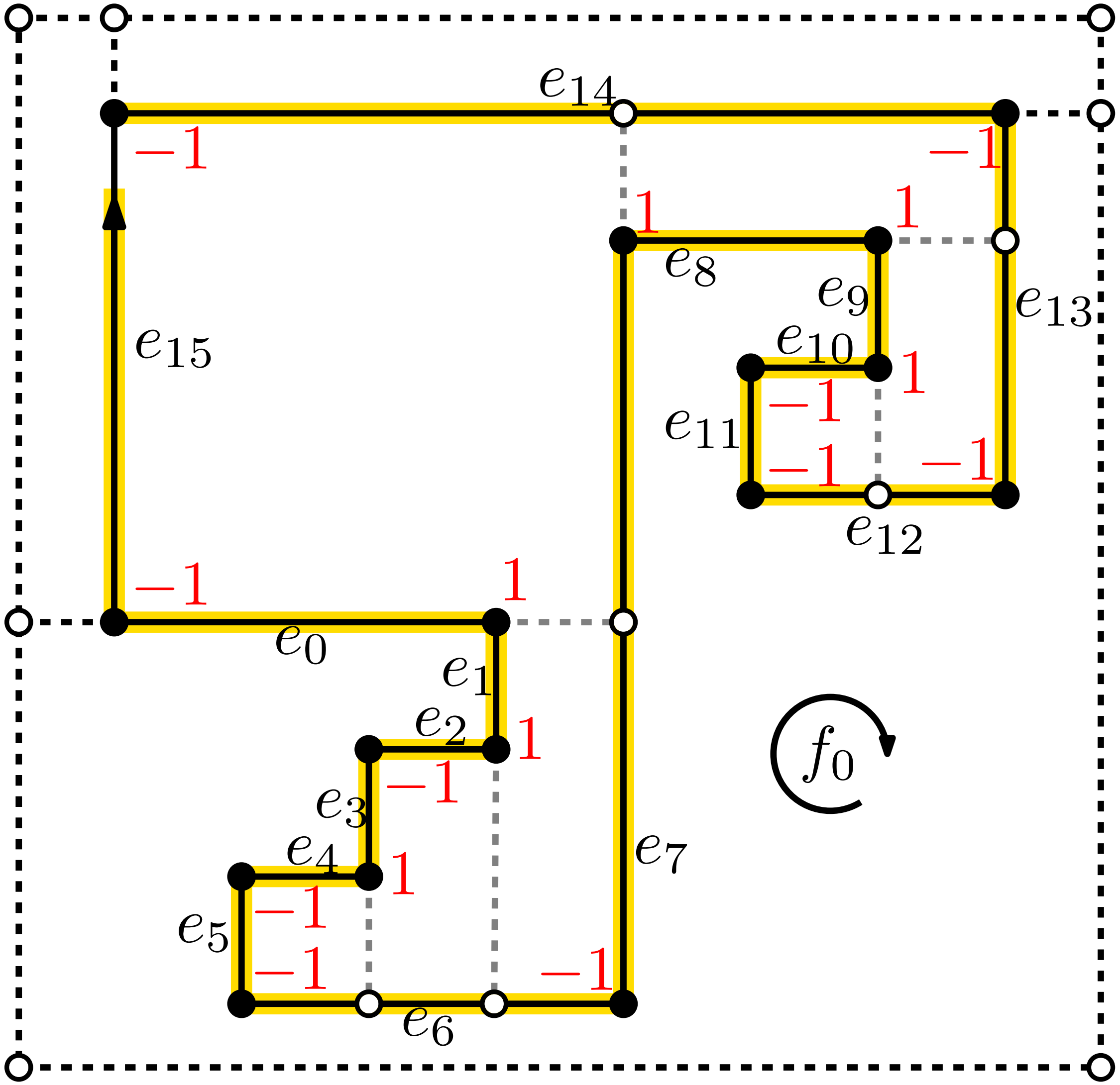
Refinement of (G, H) – Outer Face



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

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

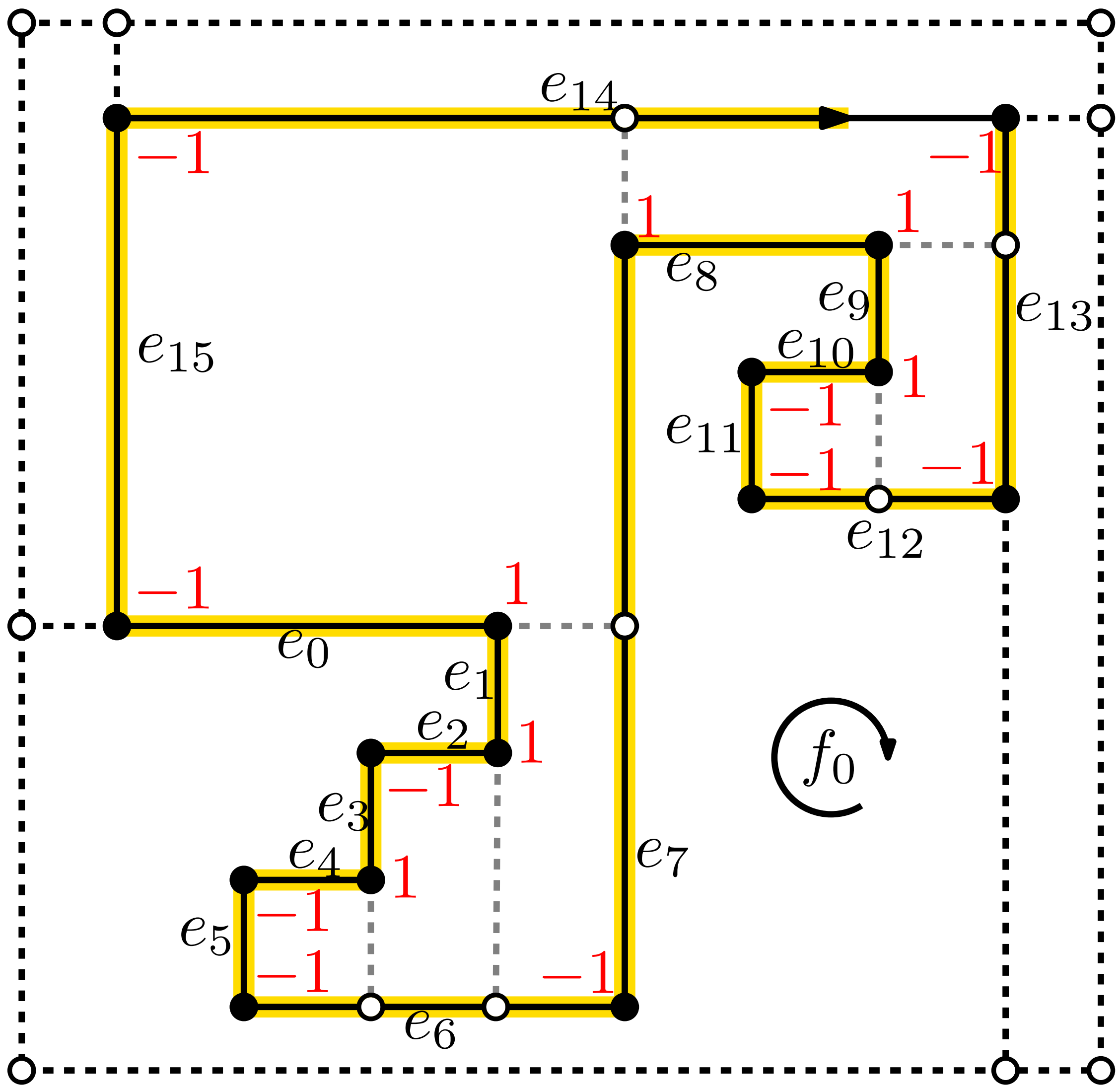
Refinement of (G, H) – Outer Face



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

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

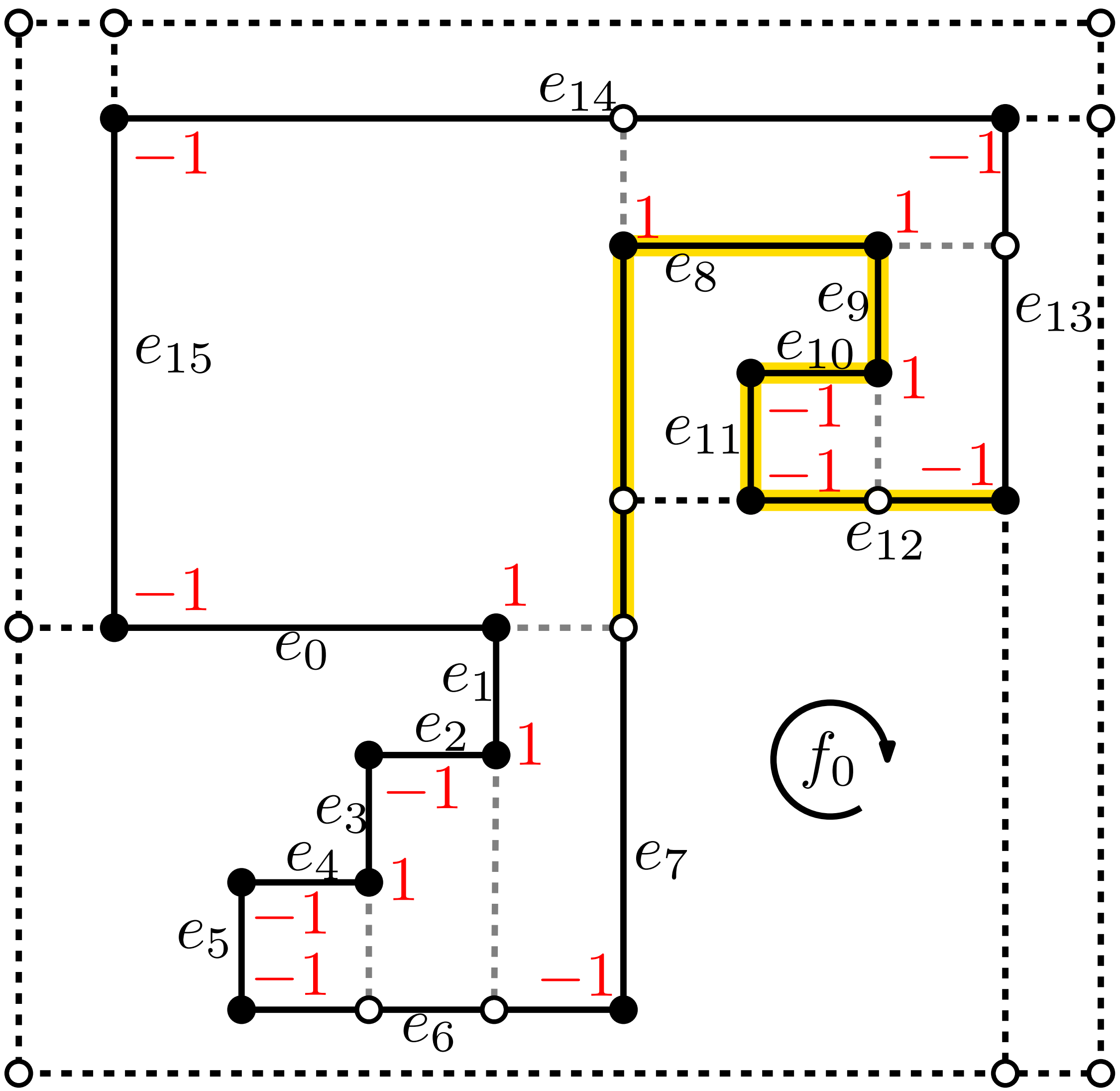
Refinement of (G, H) – Outer Face



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

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

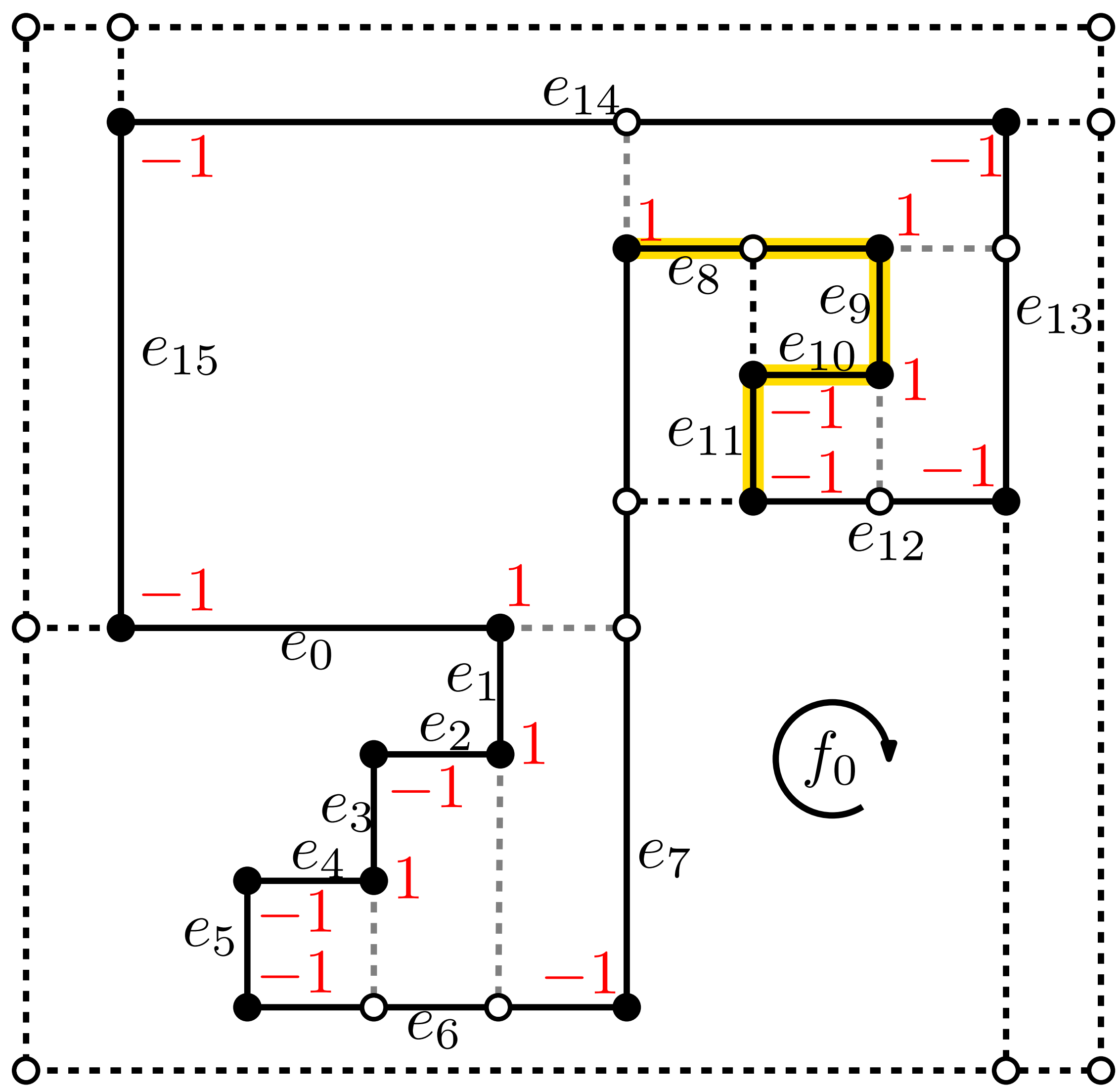
Refinement of (G, H) – Outer Face



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

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

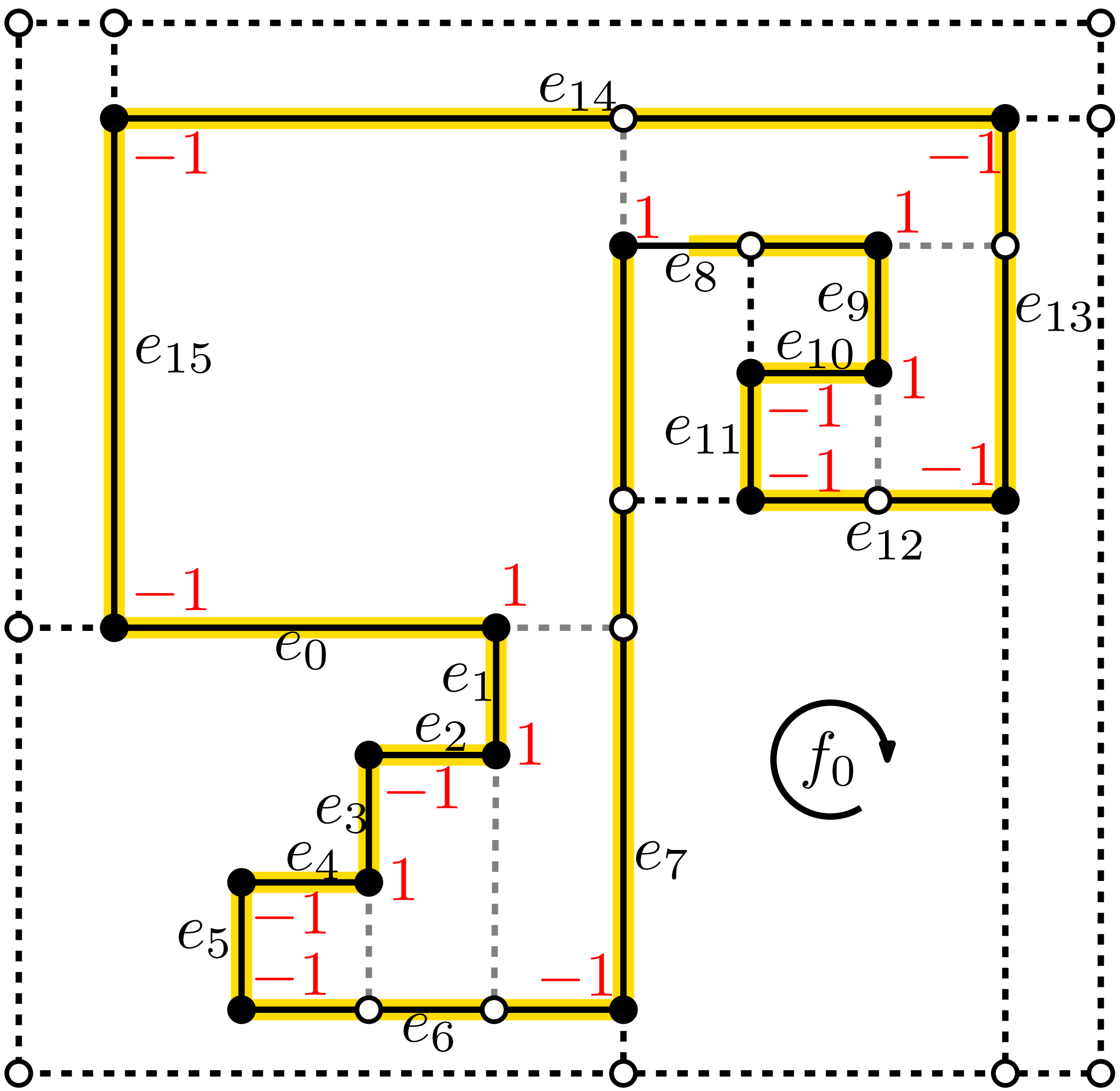
Refinement of (G, H) – Outer Face



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

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

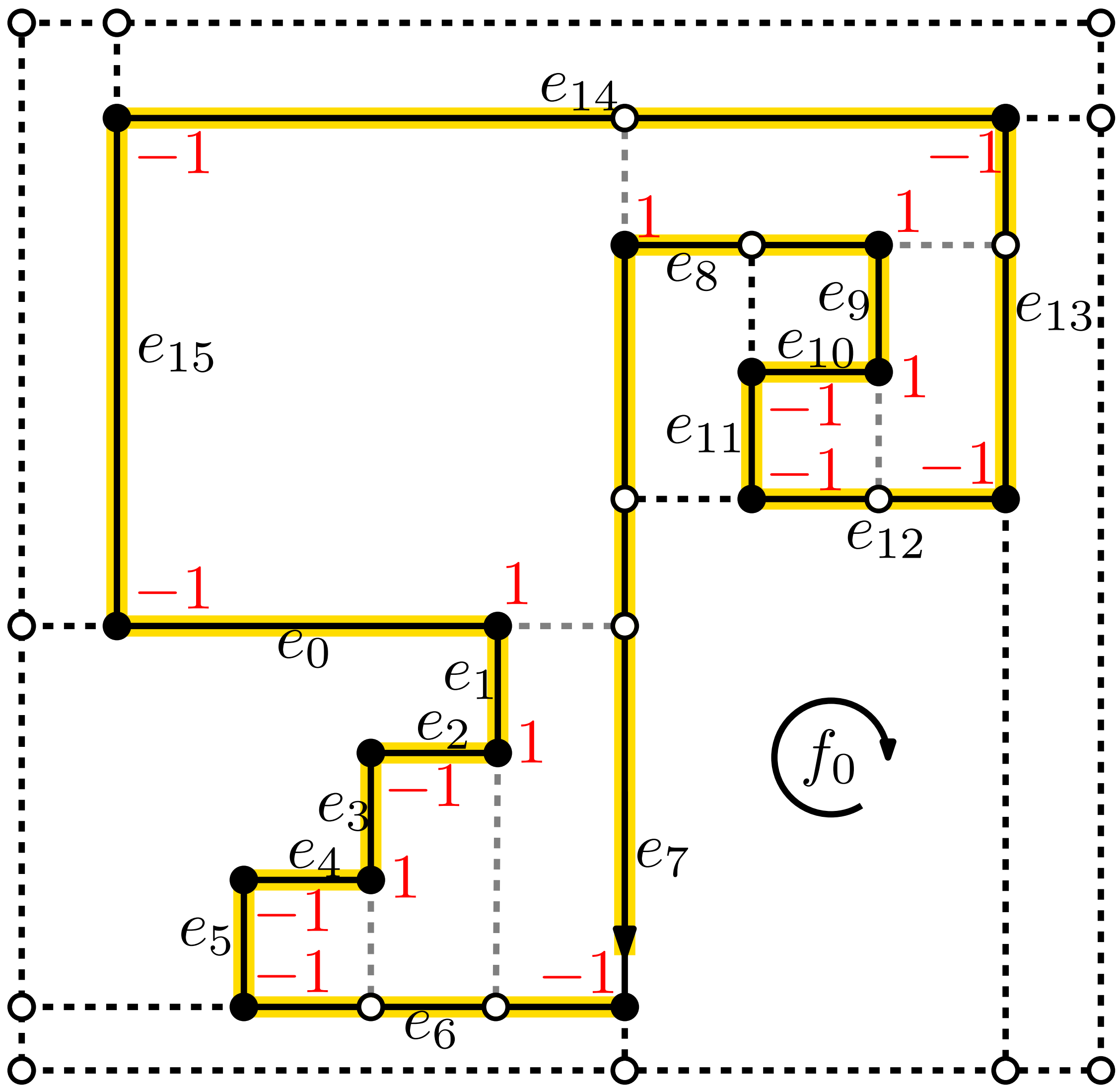
Refinement of (G, H) – Outer Face



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

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

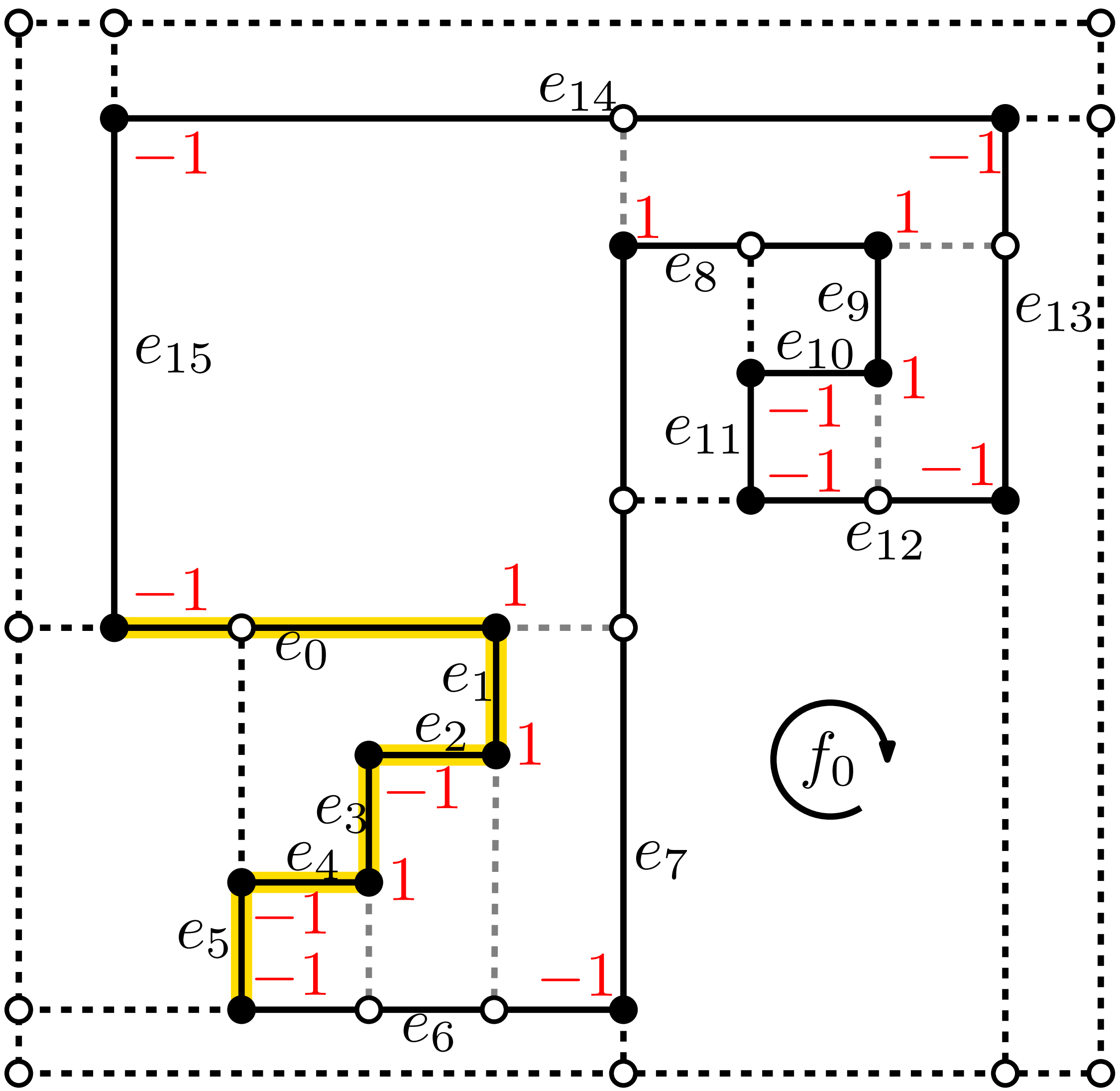
Refinement of (G, H) – Outer Face



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

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

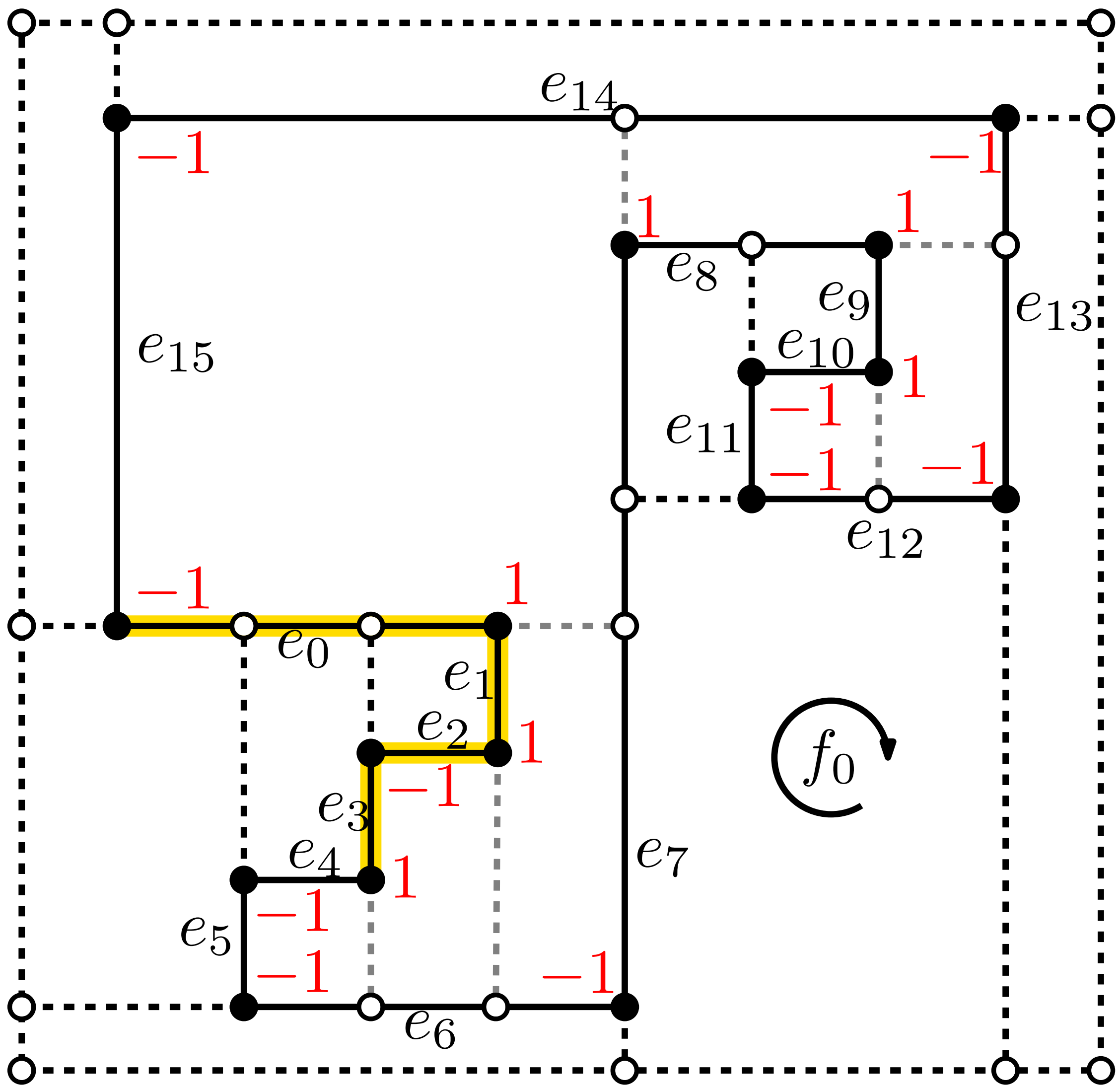
Refinement of (G, H) – Outer Face



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

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

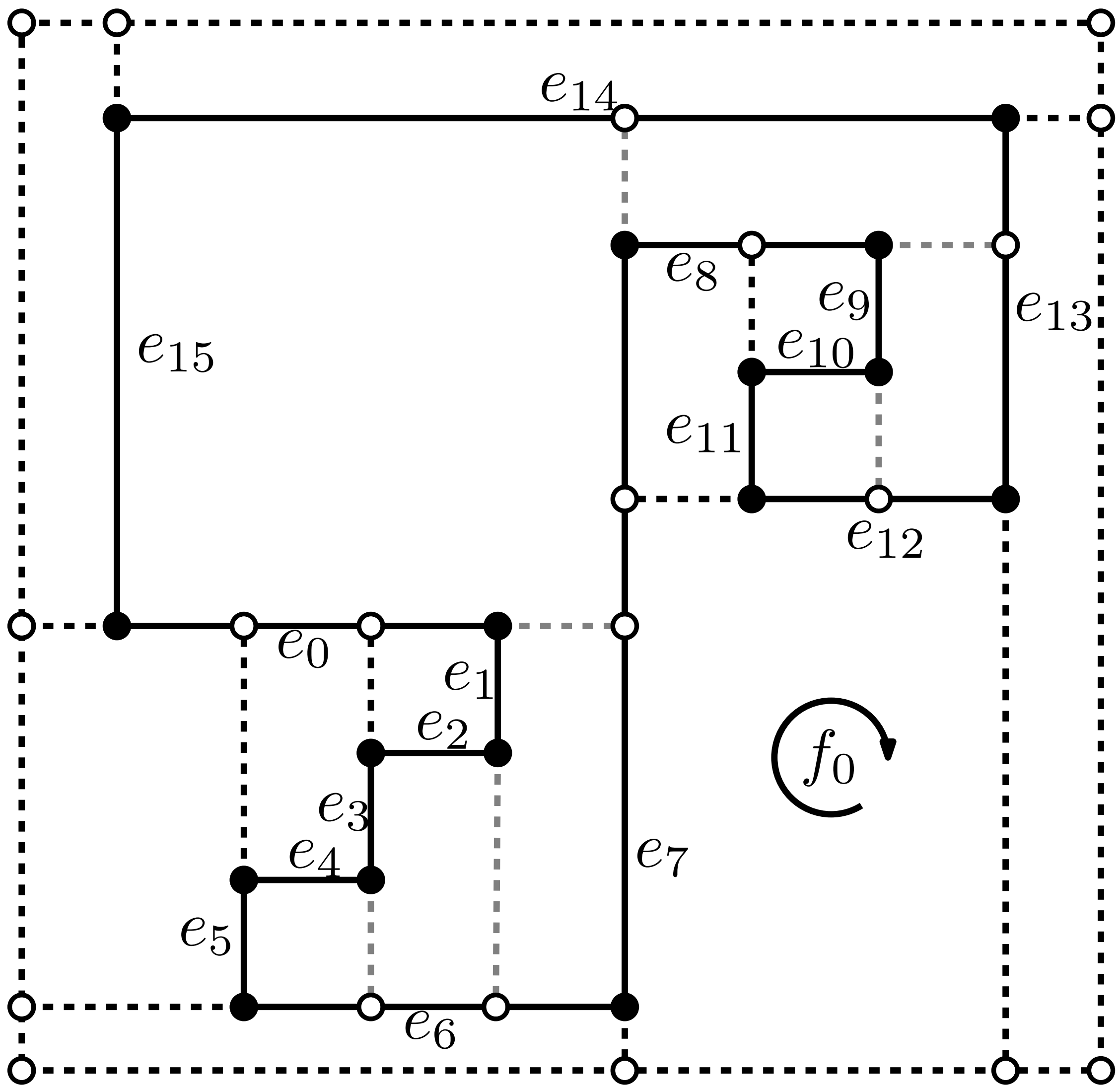
Refinement of (G, H) – Outer Face



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

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

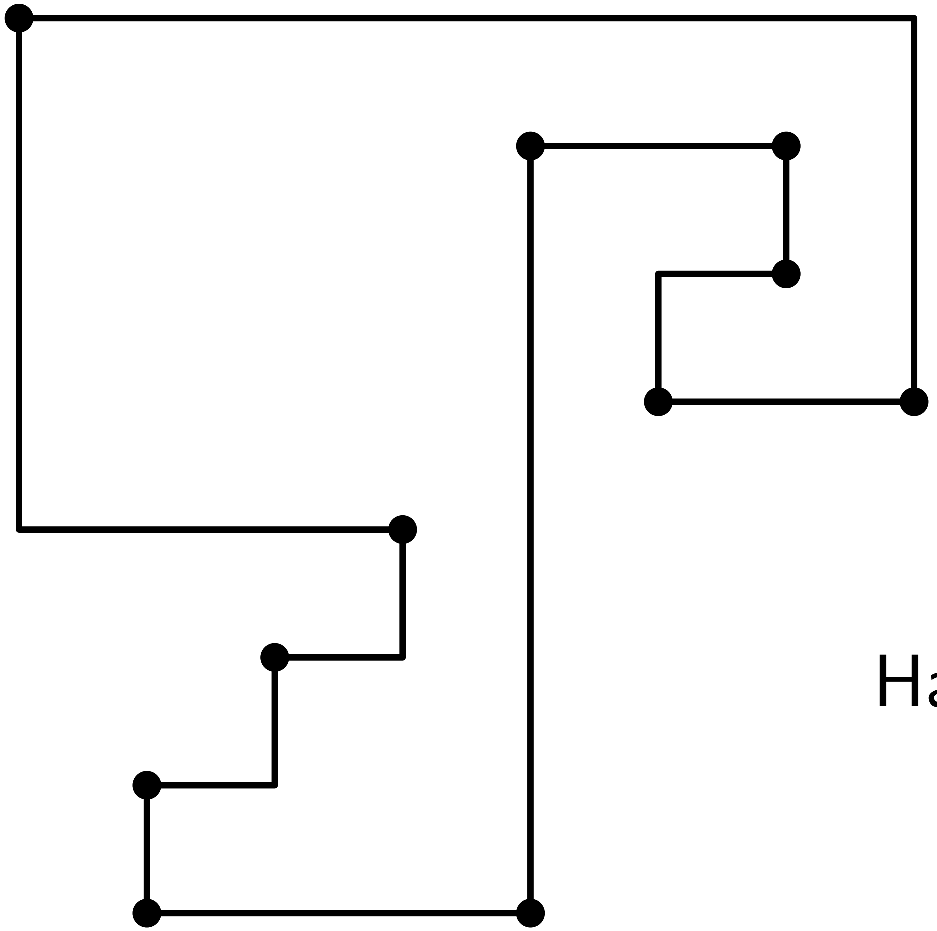
Refinement of (G, H) – Outer Face



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

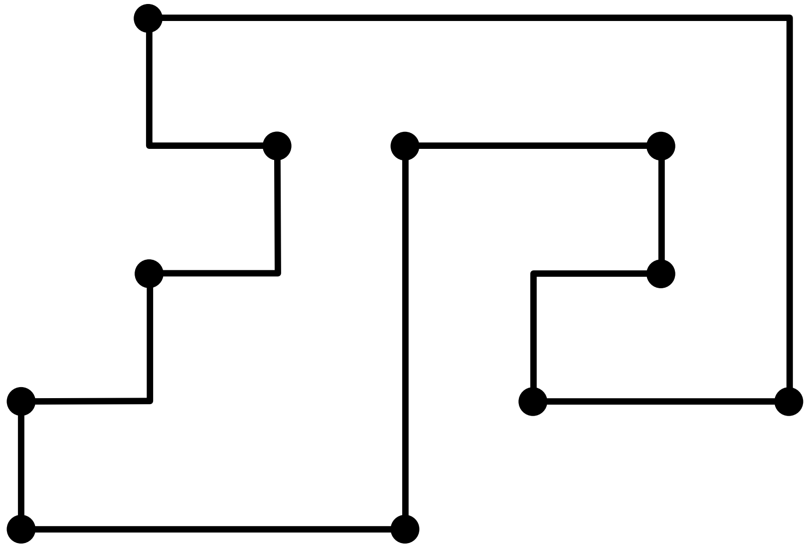
all faces are rectangles $\rightarrow$ apply flow network

Refinement of (G, H) – Outer Face



Has minimum area?

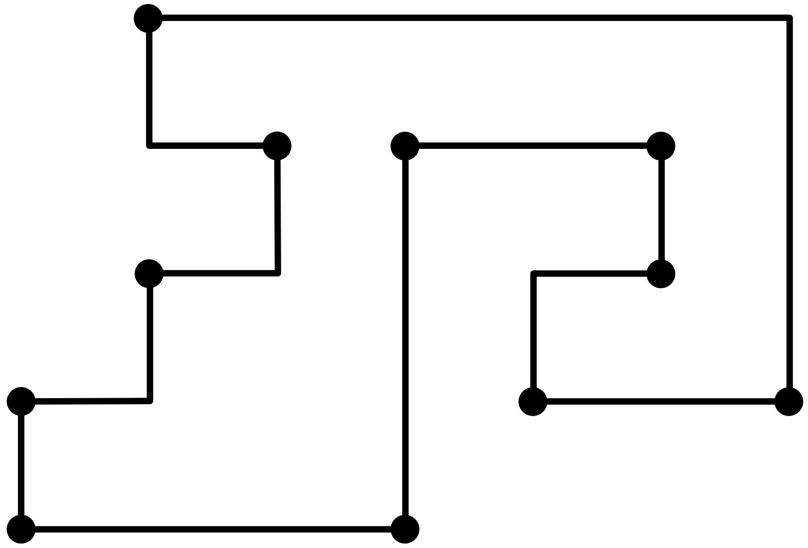
Refinement of (G, H) – Outer Face



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. [Patrignany CGTA 2001]
- Solvable with an integer linear program (ILP) [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]