Using the Metro Map Metaphor for Drawing Hypergraphs

SOFSEM 2021, Bozen/Bolzano, Italy

Fabian Frank, Michael Kaufmann, Stephen Kobourov, Tamara Mchedlidze, Sergey Pupyrev, Torsten Ueckerdt and Alexander Wolff

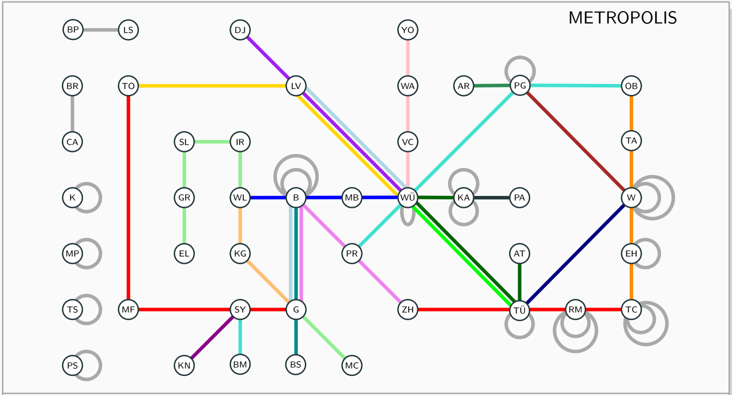
April 11, 2021

Visualizations of a Hypergraph as a Metro Map

Visualization style

- Given hypergraph $H = (V, \mathcal{E})$,
- transform H into a metromap with V being the stations and $\mathcal{E}$ being the metrolines

GD 2019 as a Metro Map



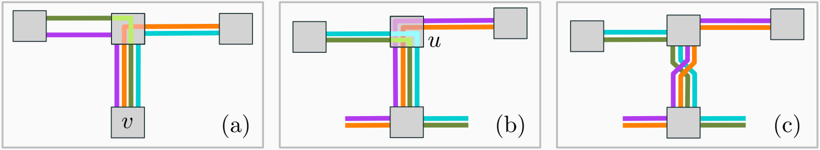
AR	Arezzo	K	Cologne	KG	Klosterneuburg			SY	Sydney
AT	Athens	DJ	Daejeon	KN	Konstanz	OS	Osnabrück	TA	Tampere
BM	Belmont	EL	East Lansing	LS	Lausanne	PS	Palaiseau	TO	Tokyo
B	Berlin	EH	Eindhoven	LV	Lviv	PA	Passau	TS	Tsukuba
BR	Brno	G	Graz	MF	Medfort	PG	Perugia	TÜ	Tübingen
BS	Brunswick	GR	Grenoble	MC	Mexico City	PR	Prague	TC	Tucson
BP	Budapest	IR	Irvine	MB	Middelburg	RM	Rome	VC	Vancouver
CA	Cambridge	KA	Karlsruhe	MP	Montpellier	SL	St. Louis	W	Vienna

WA	Warsaw
WL	Waterloo
WÜ	Würzburg
YO	Yongin
ZH	Zürich

Outlook

- Definitions
- Hardness result
- Algorithm
- Summary of the results

Crossings



- a vertex crossing in a) and b)
- a line crossing in c)

Crossing Vertex Minimization

Crossing Vertex Minimization

Given a pair (G, Π) or a pair $(\mathcal{G}, \Pi)$, we seek for a metro-map embedding that minimizes the *number of crossing vertices*, that is, the number of vertices containing vertex crossings—under the restriction that line crossings are not allowed.

NP-Hardness

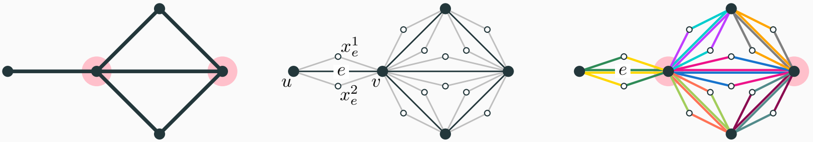
Theorem 1

Crossing Vertex Minimization is NP-complete, both with fixed and variable embedding.

NP-Hardness

Theorem 1

Crossing Vertex Minimization is NP-complete, both with fixed and variable embedding.



By reduction from PLANAR VERTEX COVER.

Algorithms

Theorem 2

Given a planar graph G and a set Π of paths in G,
we can decide efficiently whether (G, Π) admits a metro-map embedding without crossings.

Algorithms

Clustered Planarity

- planar graph H
- set $\mathcal{C}$ of subsets of vertices (clusters)
- any pair $C_1, C_2 \in \mathcal{C}$ of clusters is either disjoint or comparable by inclusion, i.e., $C_1 \cap C_2 \in \{\emptyset, C_1, C_2\}$

Algorithms

Clustered Planarity

- planar graph H
- set $\mathcal{C}$ of subsets of vertices (clusters)
- any pair $C_1, C_2 \in \mathcal{C}$ of clusters is either disjoint or comparable by inclusion, i.e., $C_1 \cap C_2 \in \{\emptyset, C_1, C_2\}$

Is there a crossing-free drawing of H together with a set of crossing-free closed Jordan curves, one for each cluster, such that each curve γ_C for cluster C contains exactly the vertices of C in its interior, and each curve crosses each edge at most once?

Algorithms

Clustered Planarity

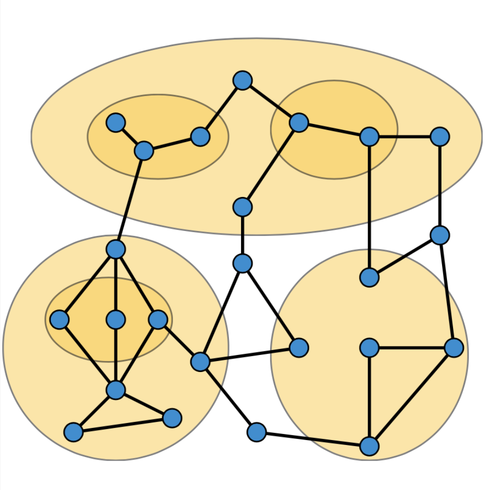
- planar graph H
- set $\mathcal{C}$ of subsets of vertices (clusters)
- any pair $C_1, C_2 \in \mathcal{C}$ of clusters is either disjoint or comparable by inclusion, i.e., $C_1 \cap C_2 \in \{\emptyset, C_1, C_2\}$

Is there a crossing-free drawing of H together with a set of crossing-free closed Jordan curves, one for each cluster, such that each curve γ_C for cluster C contains exactly the vertices of C in its interior, and each curve crosses each edge at most once?

Can be solved efficiently [Fulek & Tóth, SODA 2020].

Even in quadratic time [Bläsius, Fink, Rutter, arxiv 2020].

Clustered Planarity Example



Source: Eppstein, David (2015, September 14):

`https://en.wikipedia.org/wiki/Clustered_planarity`

Algorithms

Given (G, Π), construct $(G' = (V, E'), \Pi')$ and then Clustered Planarity instance $(H, \mathcal{C})$

- adding single edge paths to Π

Algorithms

Given (G, Π), construct $(G' = (V, E'), \Pi')$ and then Clustered Planarity instance $(H, \mathcal{C})$

- adding single edge paths to Π
- for each edge e add parallel edge $\bar{e}$ and a new path $P(\bar{e})$

Algorithms

Given (G, Π), construct $(G' = (V, E'), \Pi')$ and then Clustered Planarity instance $(H, \mathcal{C})$

- adding single edge paths to Π
- for each edge e add parallel edge $\bar{e}$ and a new path $P(\bar{e})$
- $(G' = (V, E'), \Pi')$ admits a crossing-free metro-map embedding if and only if (G, Π) does

Transformation

Construct H

vertex set contains the following vertices

Transformation

Construct H

vertex set contains the following vertices

- vertex–edge incidences $(v, e) \in V \times E'$ and $v \in e$

Transformation

Construct H

vertex set contains the following vertices

- vertex–edge incidences $(v, e) \in V \times E'$ and $v \in e$
- vertex–edge–path incidences

Transformation

Construct H

vertex set contains the following vertices

- vertex–edge incidences $(v, e) \in V \times E'$ and $v \in e$
- vertex–edge–path incidences $(v, e, P) \in V \times E' \times \Pi'$ with $v \in e \in E(P)$

Transformation

Construct H

vertex set contains the following vertices

- vertex–edge incidences $(v, e) \in V \times E'$ and $v \in e$
- vertex–edge–path incidences $(v, e, P) \in V \times E' \times \Pi'$ with $v \in e \in E(P)$
- vertex–path incidences $(v, P) \in V \times \Pi'$ with $v \in P$

Transformation

Construct H

edge set contains the following edges:

Transformation

Construct H

edge set contains the following edges:

- $P = [v_1, \ldots, v_p]$ in Π' with $v_i \in V$ $(i = 1, \ldots, p)$, $e_i = v_i v_{i+1} \in E'$ $(i = 1, \ldots, p-1)$ graph H contains a path $[(v_1, P), (v_1, e_1, P), (v_2, e_1, P), (v_2, P), (v_2, e_2, P), \ldots, (v_{p+1}, e_p, P)]$ on all incidences of P as they appear along the path

Transformation

Construct H

edge set contains the following edges:

- $P = [v_1, \ldots, v_p]$ in Π' with $v_i \in V$ $(i = 1, \ldots, p)$, $e_i = v_i v_{i+1} \in E'$ $(i = 1, \ldots, p-1)$ graph H contains a path $[(v_1, P), (v_1, e_1, P), (v_2, e_1, P), (v_2, P), (v_2, e_2, P), \ldots, (v_{p+1}, e_p, P)]$ on all incidences of P as they appear along the path
- $e = uv$ of the original graph G with parallel edge $\bar{e}$ in G', we put two paths $[(u, e), (u, \bar{e})]$ and $[(v, e), (v, \bar{e})]$ into H

Transformation

Construct $\mathcal{C}$

$\mathcal{C}$ contains the following clusters:

Transformation

Construct $\mathcal{C}$

$\mathcal{C}$ contains the following clusters:

- for each vertex–edge incidence (v, e), add a cluster $C(v, e) = \{(v, e)\} \cup \{(v, e, P) \in V(H) \mid P \in \Pi'\}$.

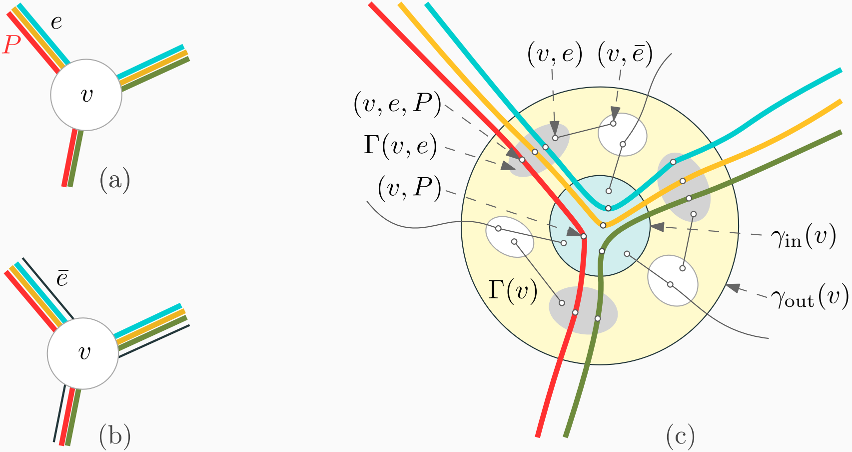
Transformation

Construct $\mathcal{C}$

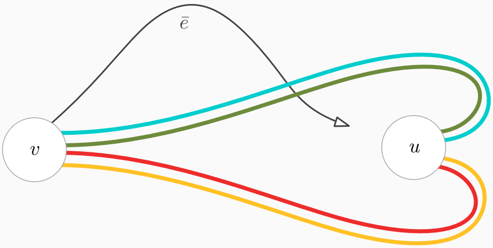
$\mathcal{C}$ contains the following clusters:

- for each vertex–edge incidence (v, e), add a cluster $C(v, e) = \{(v, e)\} \cup \{(v, e, P) \in V(H) \mid P \in \Pi'\}$.
- for each vertex v in G', we define an inner cluster $C_{\mathrm{in}}(v) = \{(v, P) \in V(H) \mid P \in \Pi'\}$ and an outer cluster $C_{\mathrm{out}}(v) = C_{\mathrm{in}}(v) \cup \bigcup_{(v,e) \in V(H)} C(v, e)$

Transformation



Algorithms



avoiding line crossings on the edge *uv* with the path $\bar{e}$

Algorithms

Corollary 3

For any fixed k, one can decide in polynomial time whether there is a metro-map embedding with at most k crossing vertices.

Summary of the Results

Problem Type	Embedding	Graph	Result
k part of input	fixed or variable	planar	NP-complete
k fixed	fixed or variable	planar	polynomial
optimization	fixed or variable	tree	polynomial