Computing Hive Plots

A Combinatorial Framework

Martin Nöllenburg · **Markus Wallinger**

22.09.2023 · Graph Drawing and Network Visualization 2023

ALGORITHMS AND
COMPLEXITY GROUP

1 This research has been funded by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

Introduction & Model

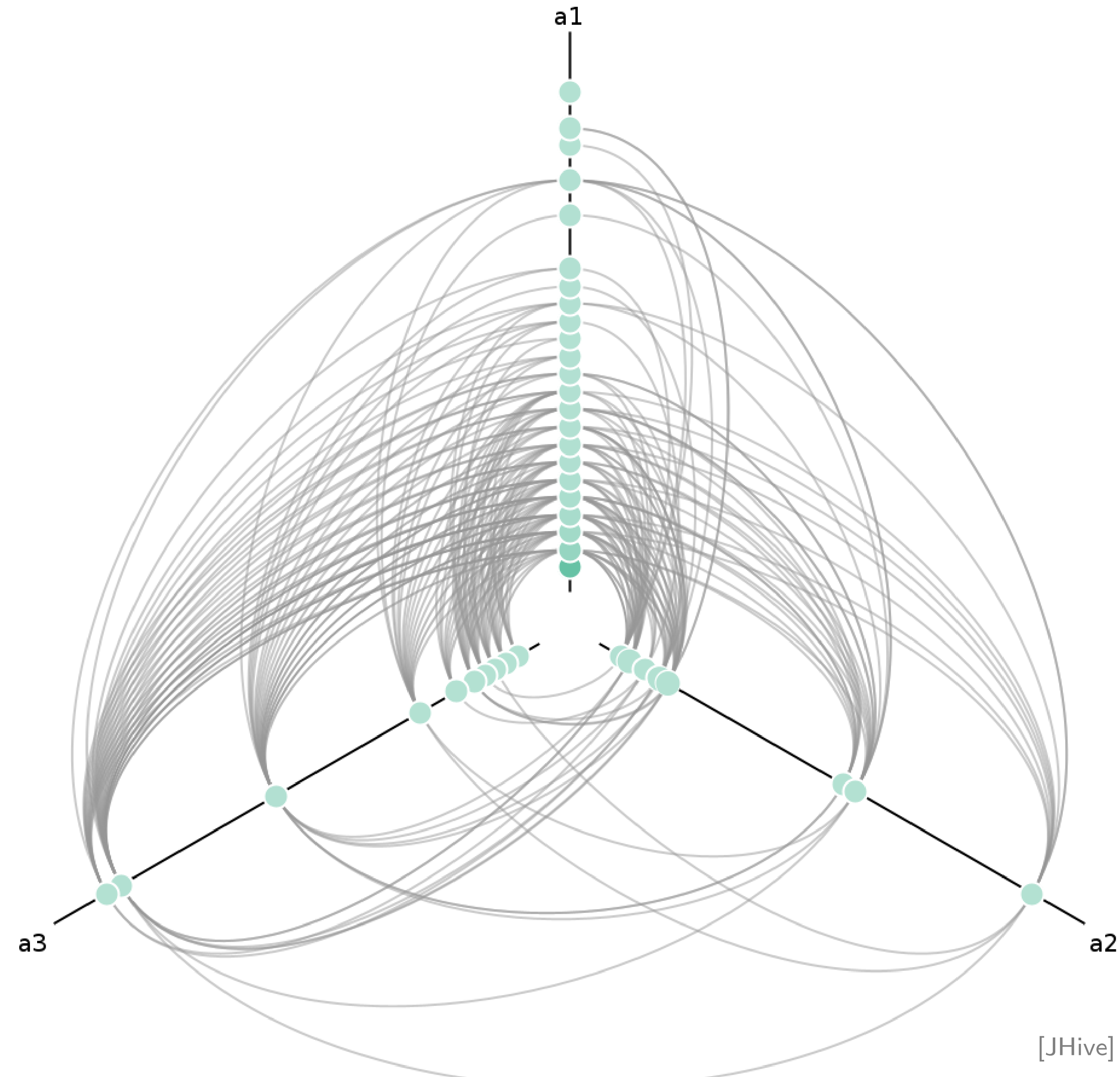
Framework

Evaluation

Introduction

Hive plots first introduced **2012***

Deterministic graph layout procedure



[JHive]

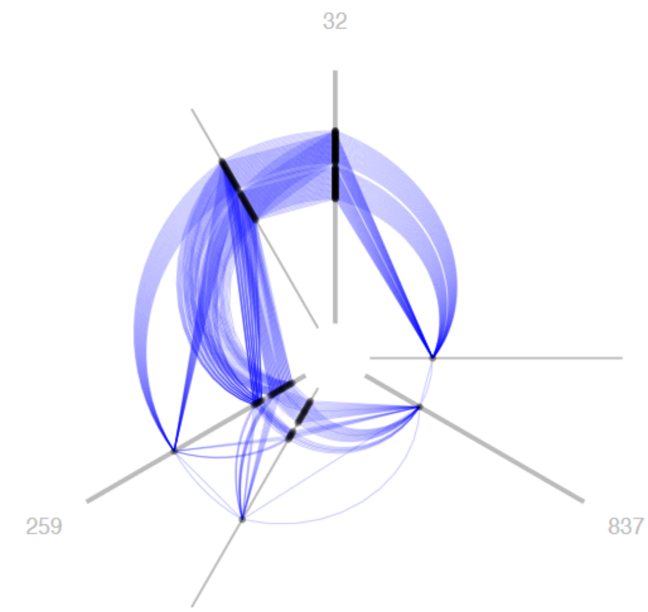
* [Krzywinski et al., 2012]

Introduction

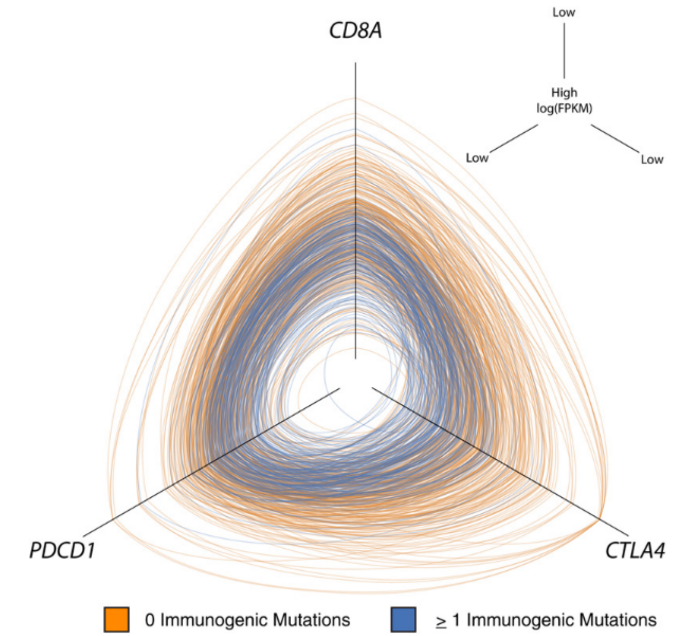
Hive plots first introduced **2012***

Deterministic graph layout procedure

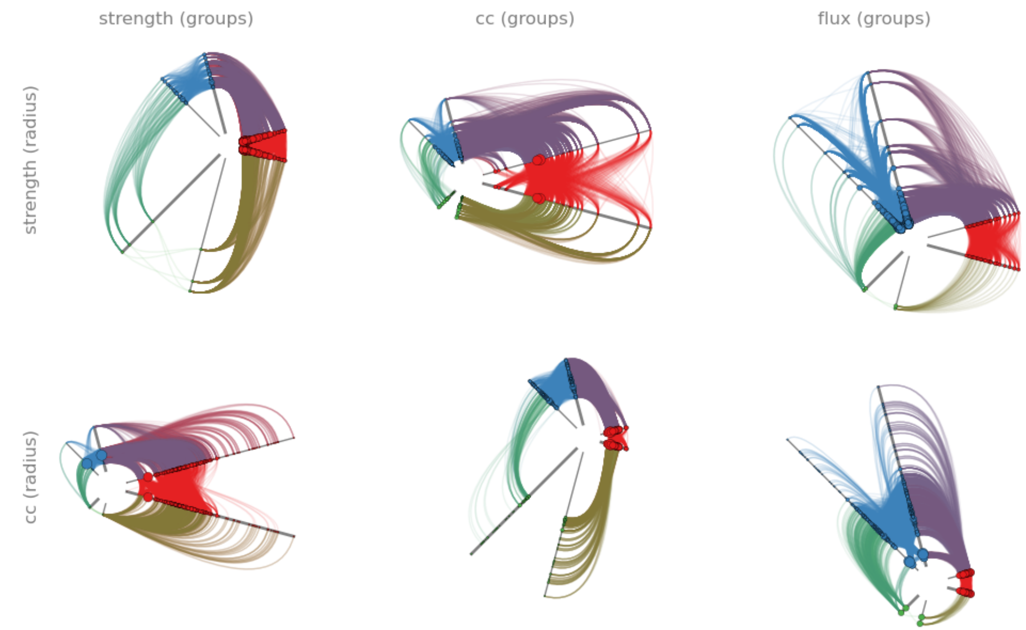
Various use cases



[Engle and Whalen, 2012]



[Brown et al., 2014]



[NNGT]

* [Krzywinski et al., 2012]

Introduction

Hive plots first introduced **2012***

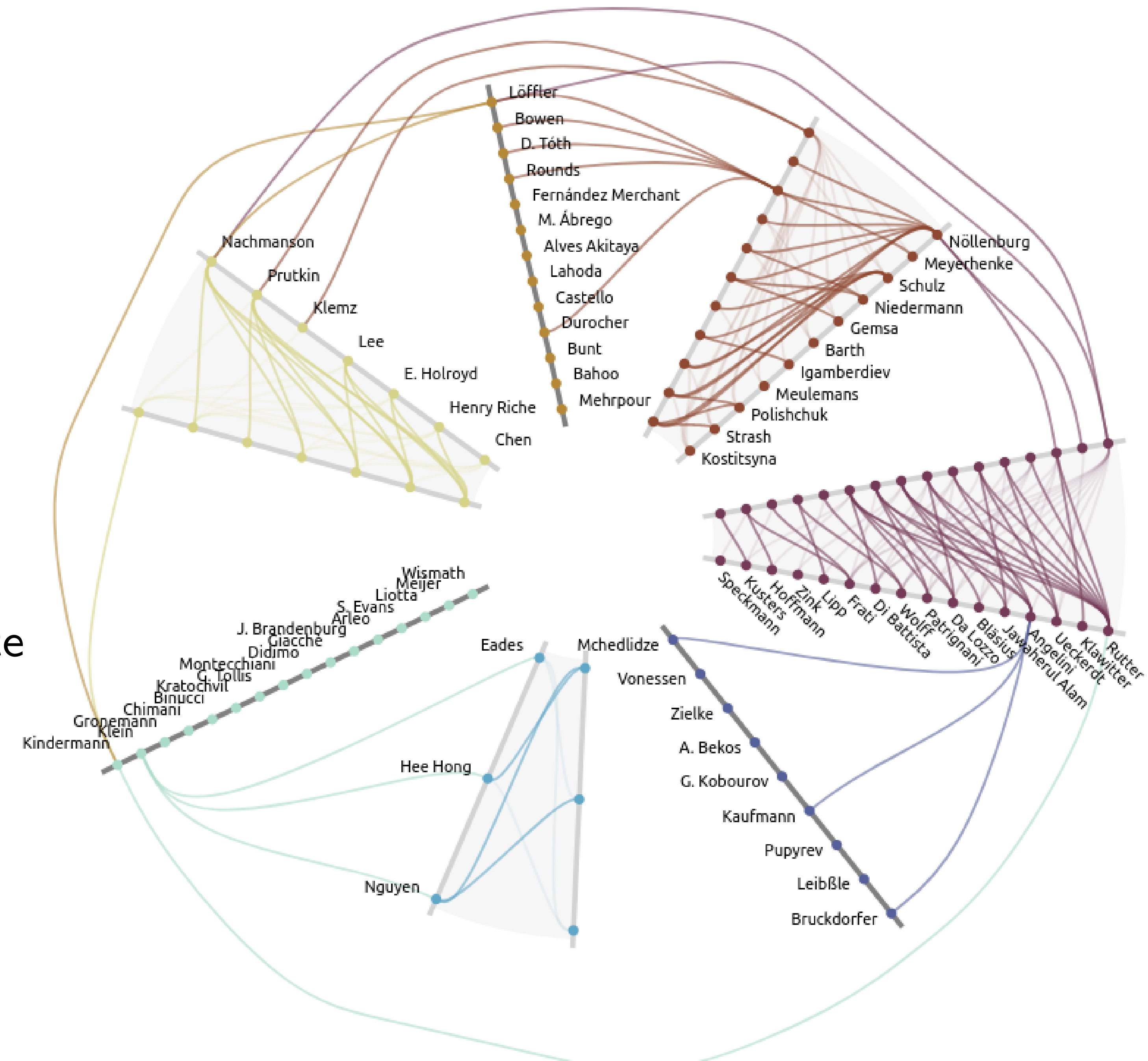
Deterministic graph layout procedure

Various use cases

No graph drawing **investigation**

Contribution: Framework to compute a **combinatorial hive plot layout**

- Introduce degrees of freedom
- Optimize typical graph drawing properties



* [Krzywinski et al., 2012]

What is a Hive Plot?

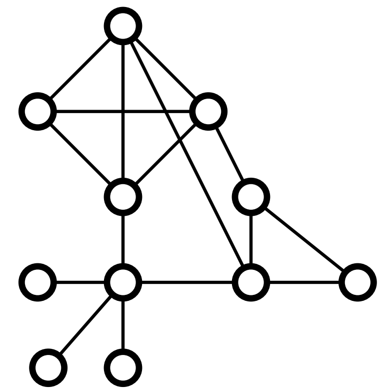
Hive Plot*

Combinatorial Model

* [Krzywinski et al., 2012]

What is a Hive Plot?

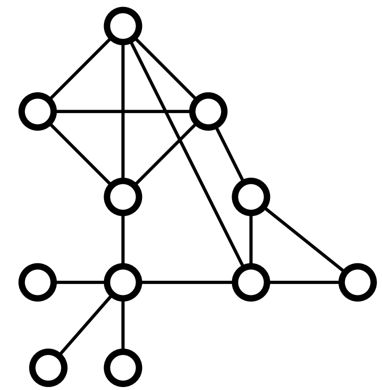
Hive Plot*



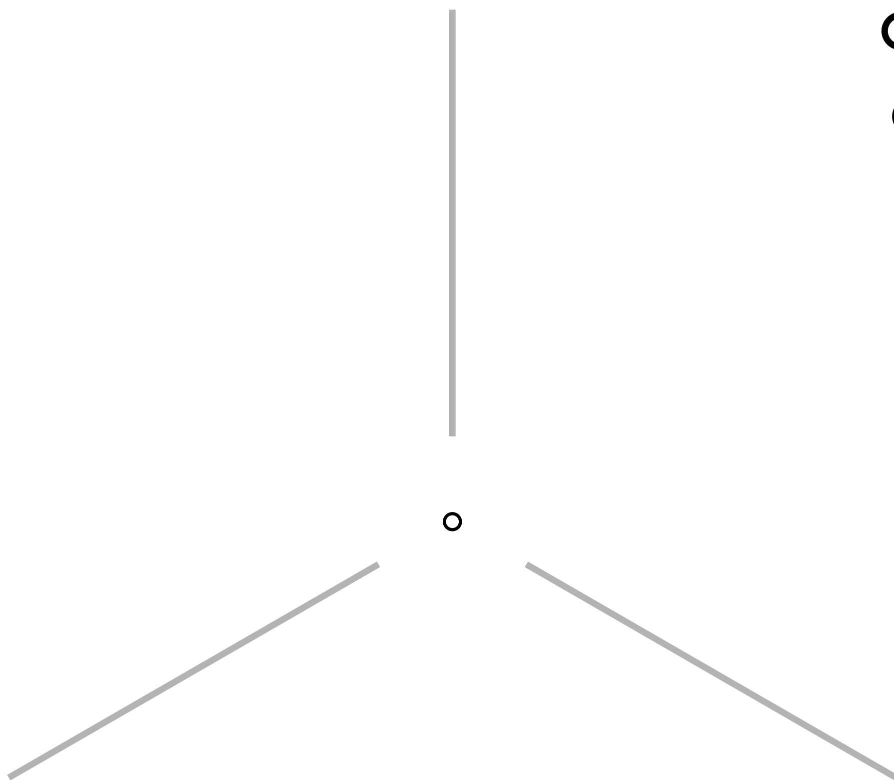
Combinatorial Model

* [Krzywinski et al., 2012]

What is a Hive Plot?



Hive Plot*

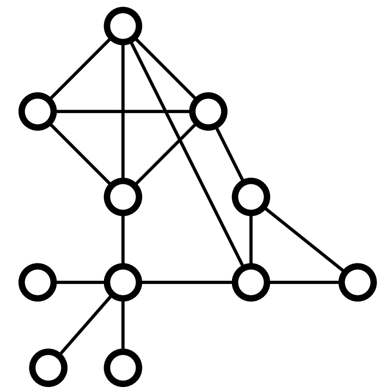


Combinatorial Model

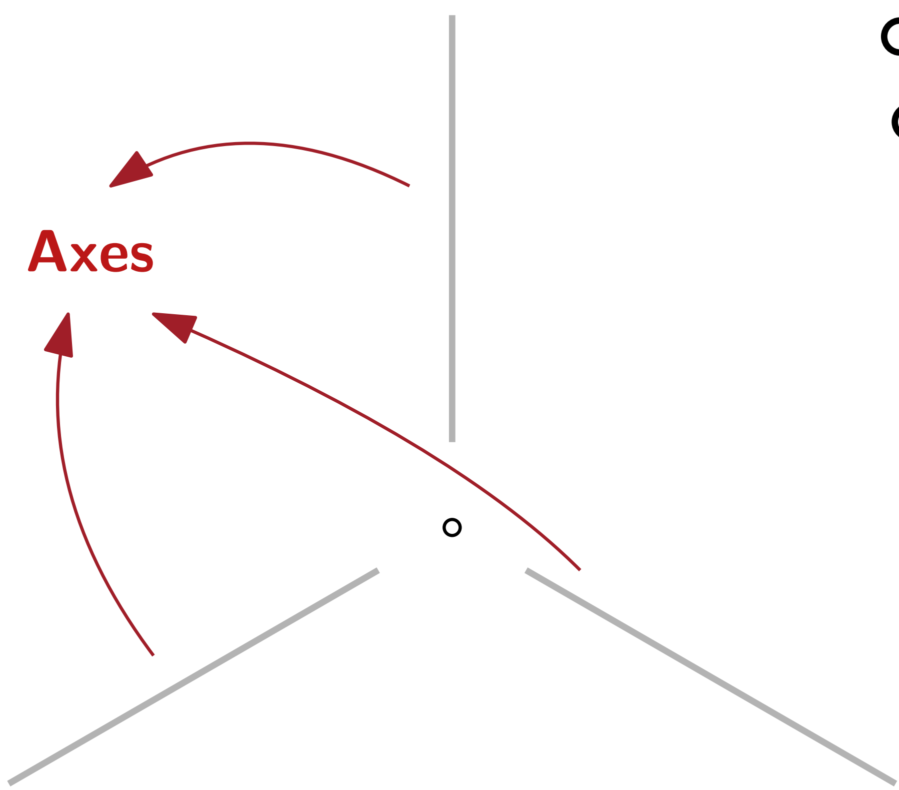
* [Krzywinski et al., 2012]

What is a Hive Plot?

Hive Plot*



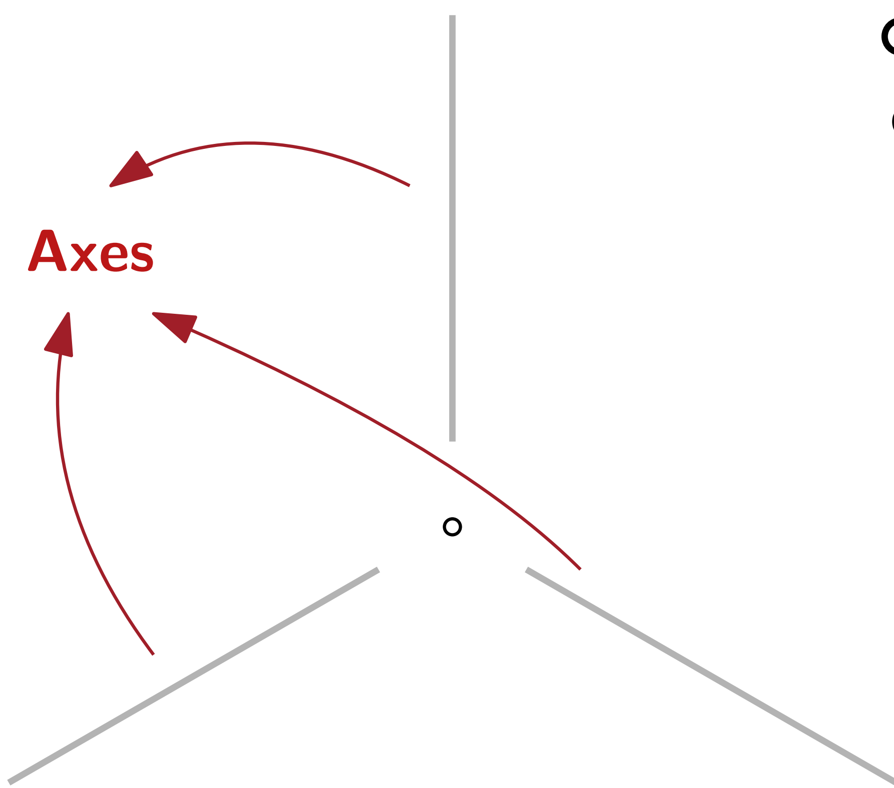
Combinatorial Model

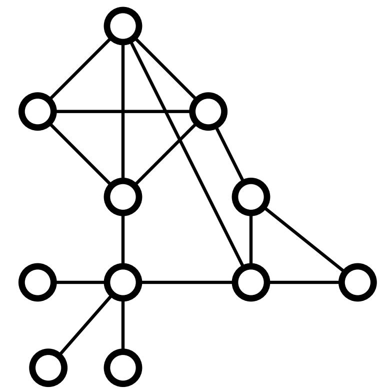


* [Krzywinski et al., 2012]

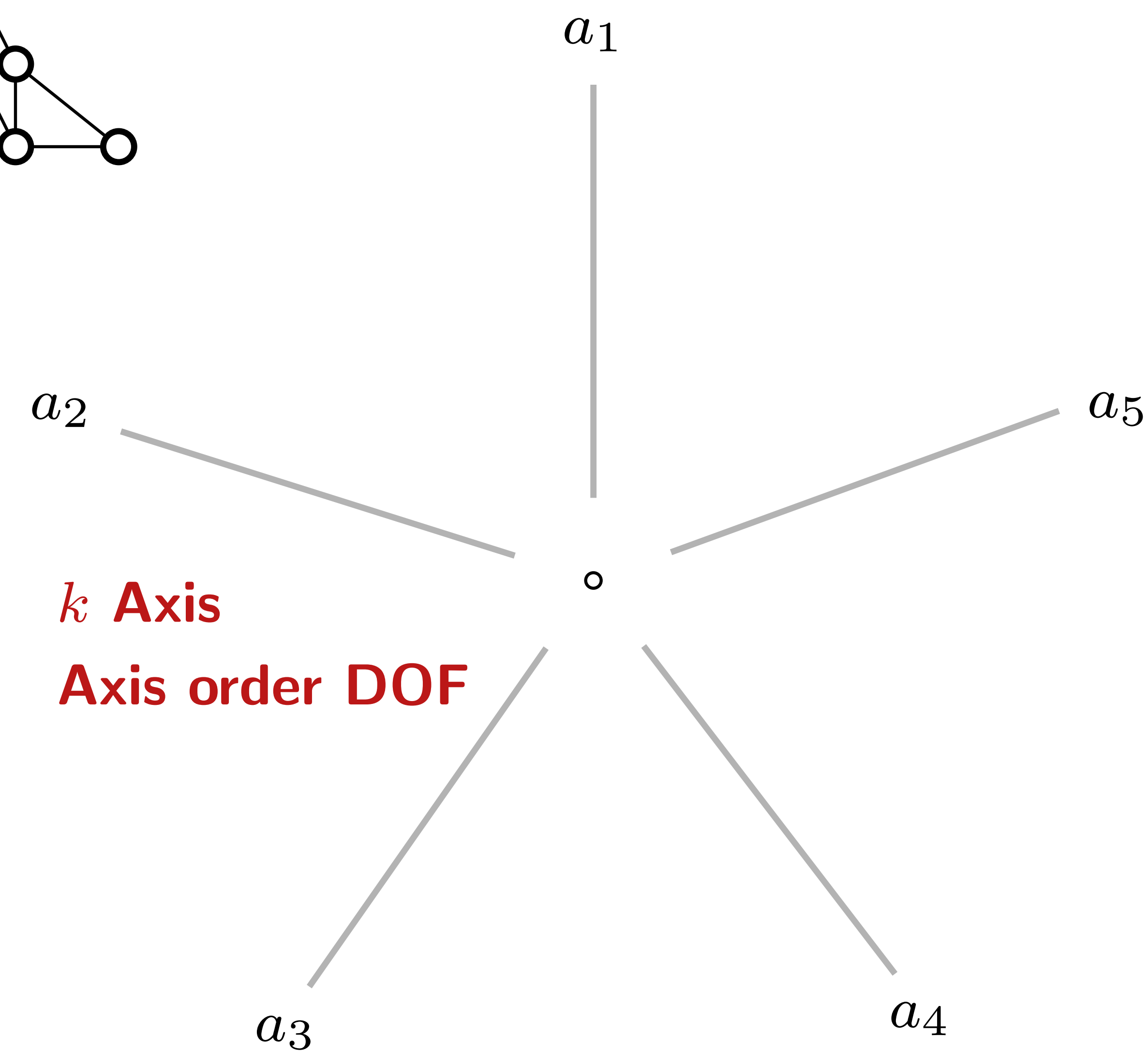
What is a Hive Plot?

Hive Plot*



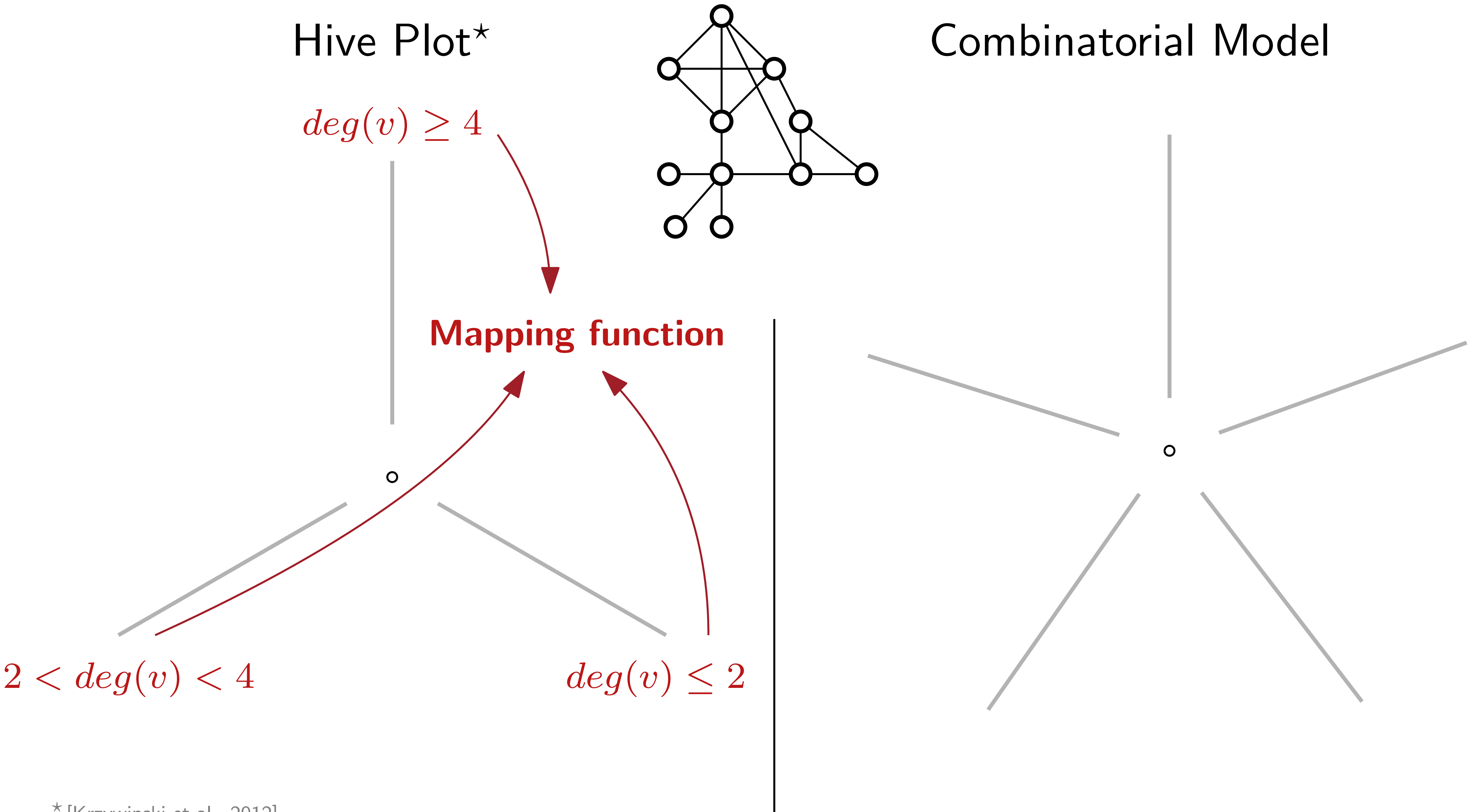


Combinatorial Model



* [Krzywinski et al., 2012]

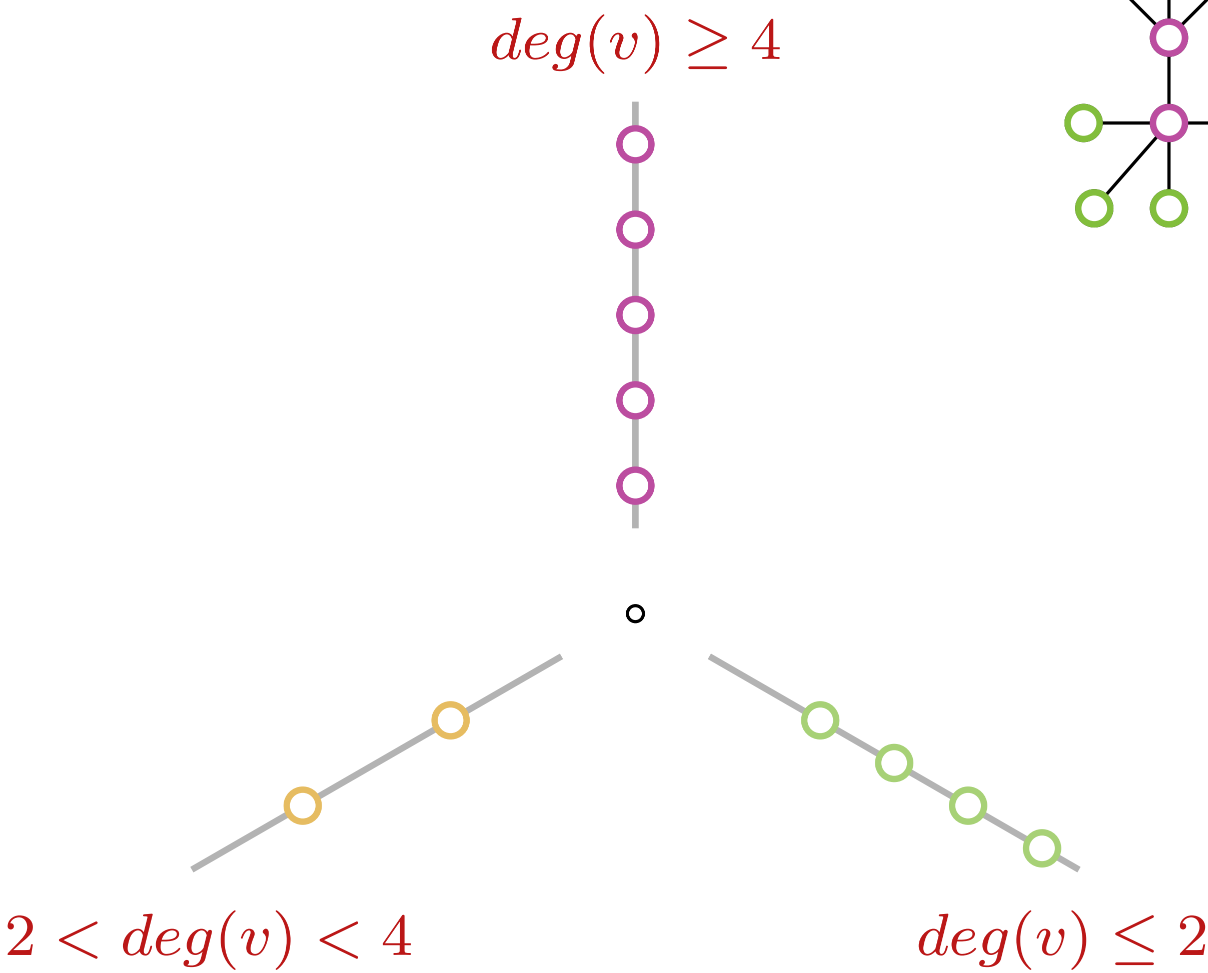
What is a Hive Plot?

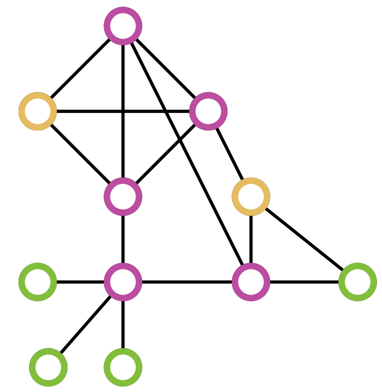


* [Krzywinski et al., 2012]

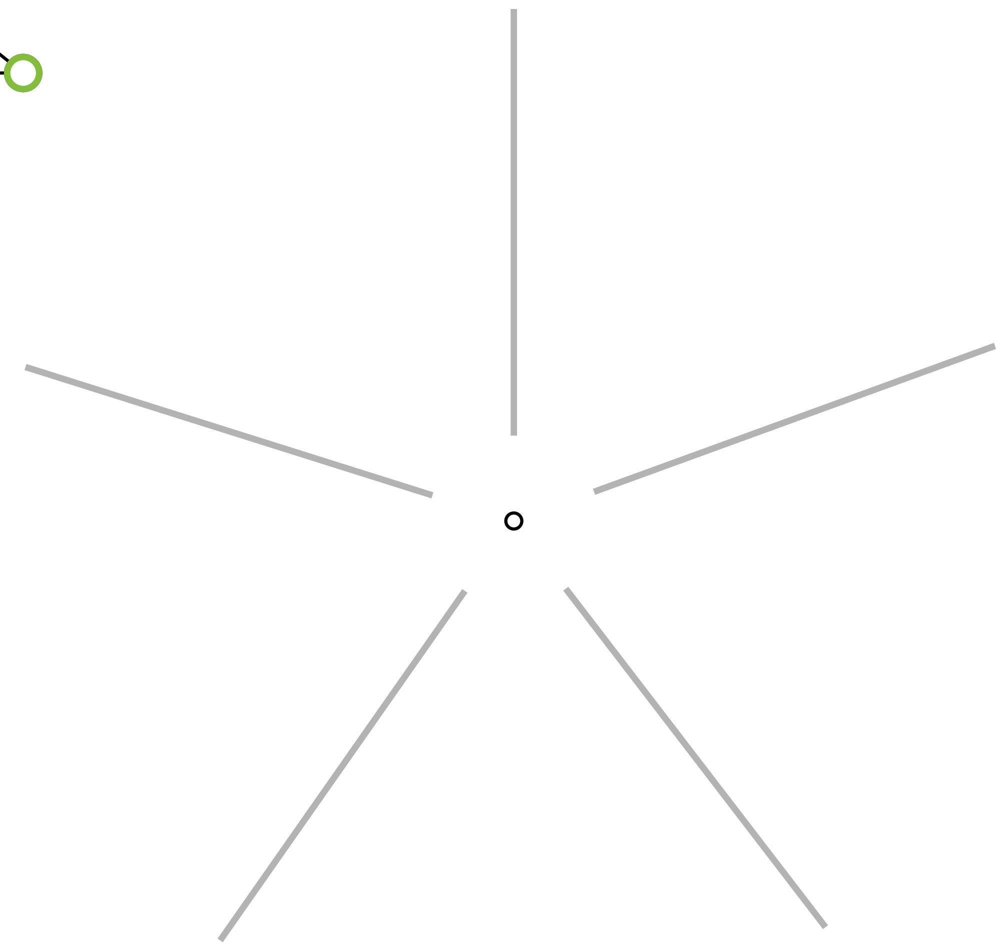
What is a Hive Plot?

Hive Plot*





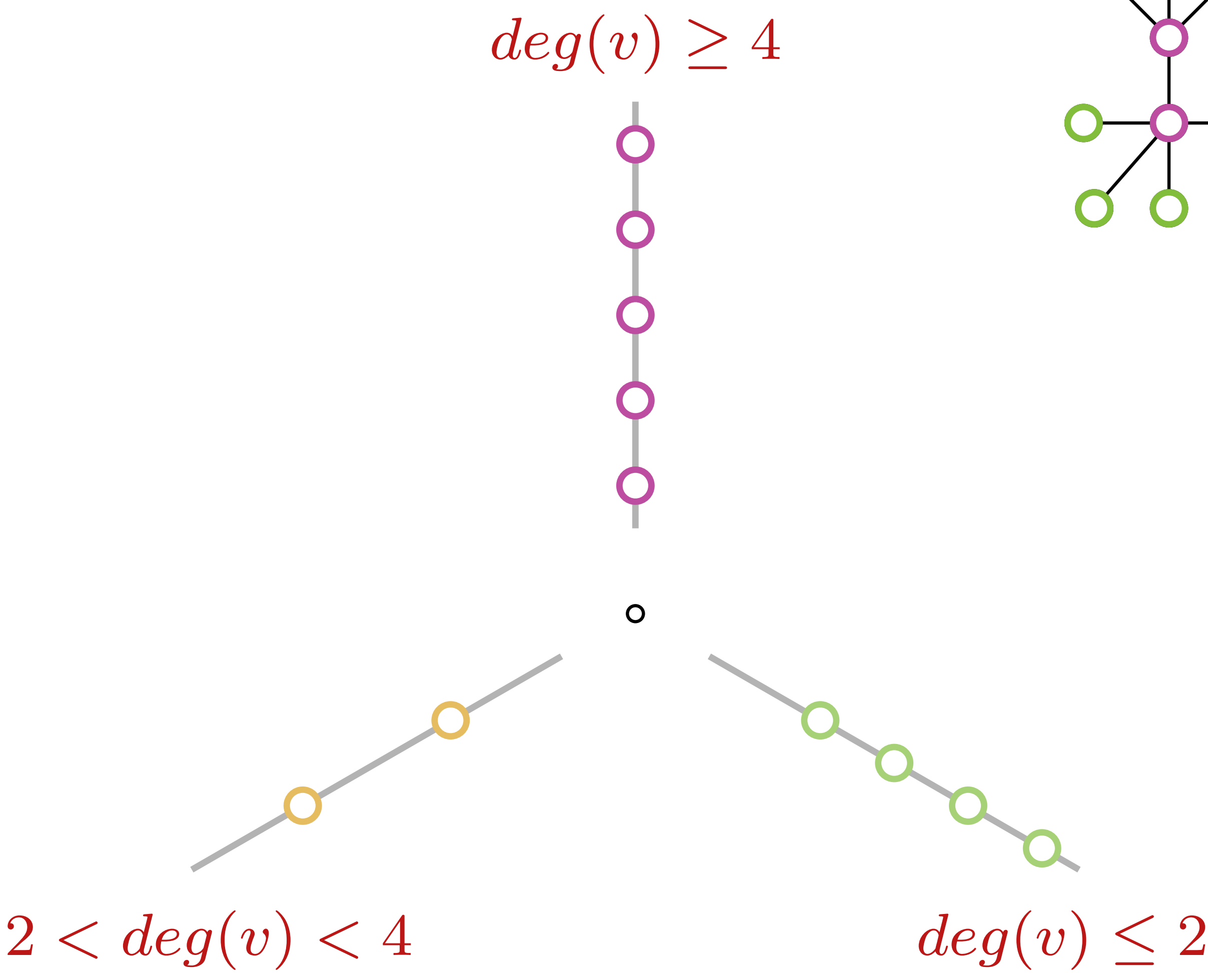
Combinatorial Model

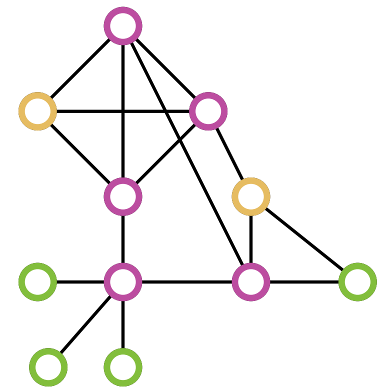


* [Krzywinski et al., 2012]

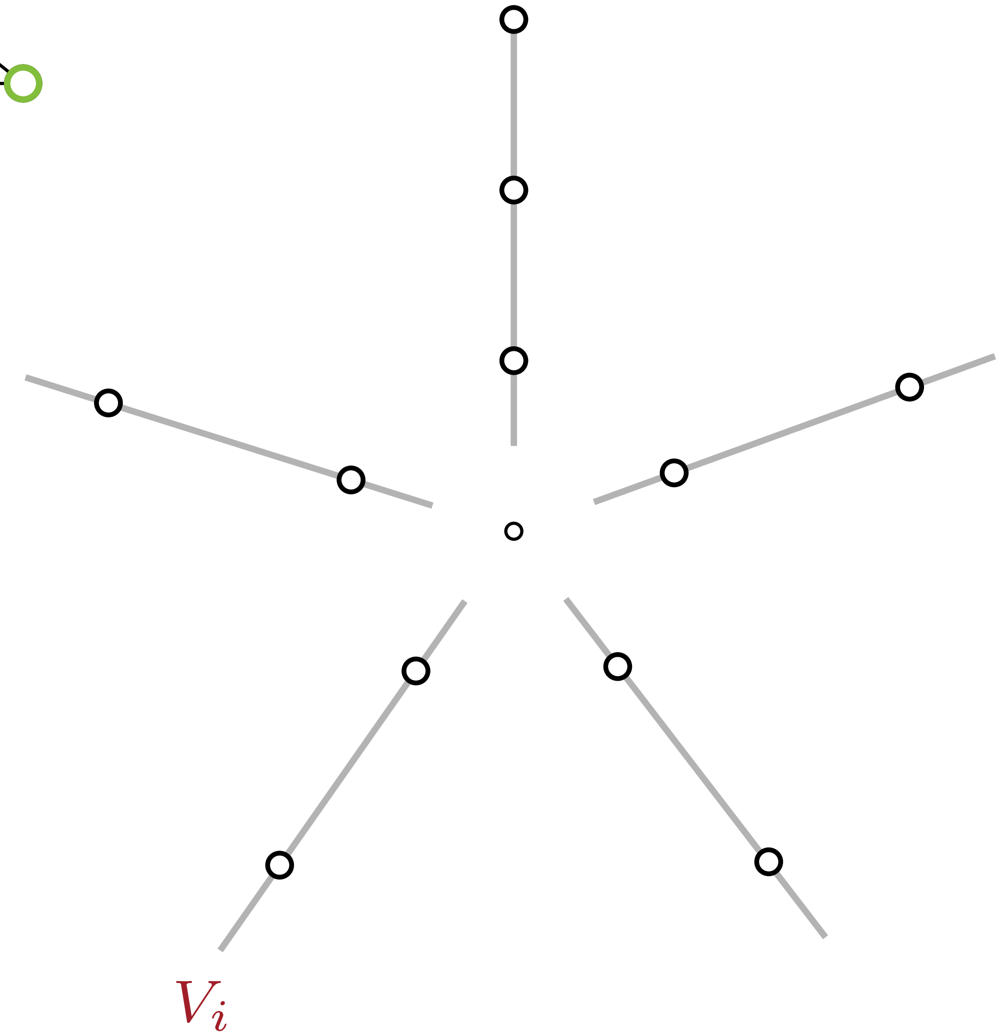
What is a Hive Plot?

Hive Plot*





Combinatorial Model

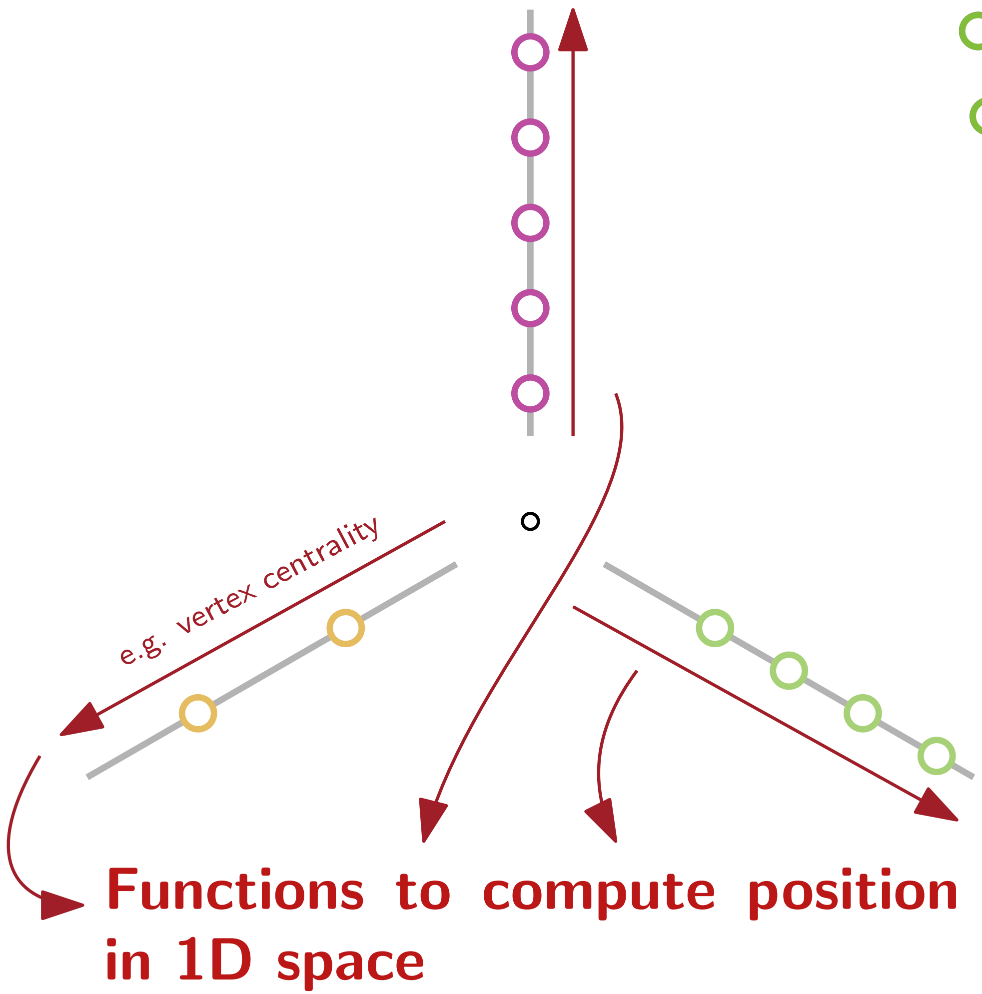


Vertex mapping DOF

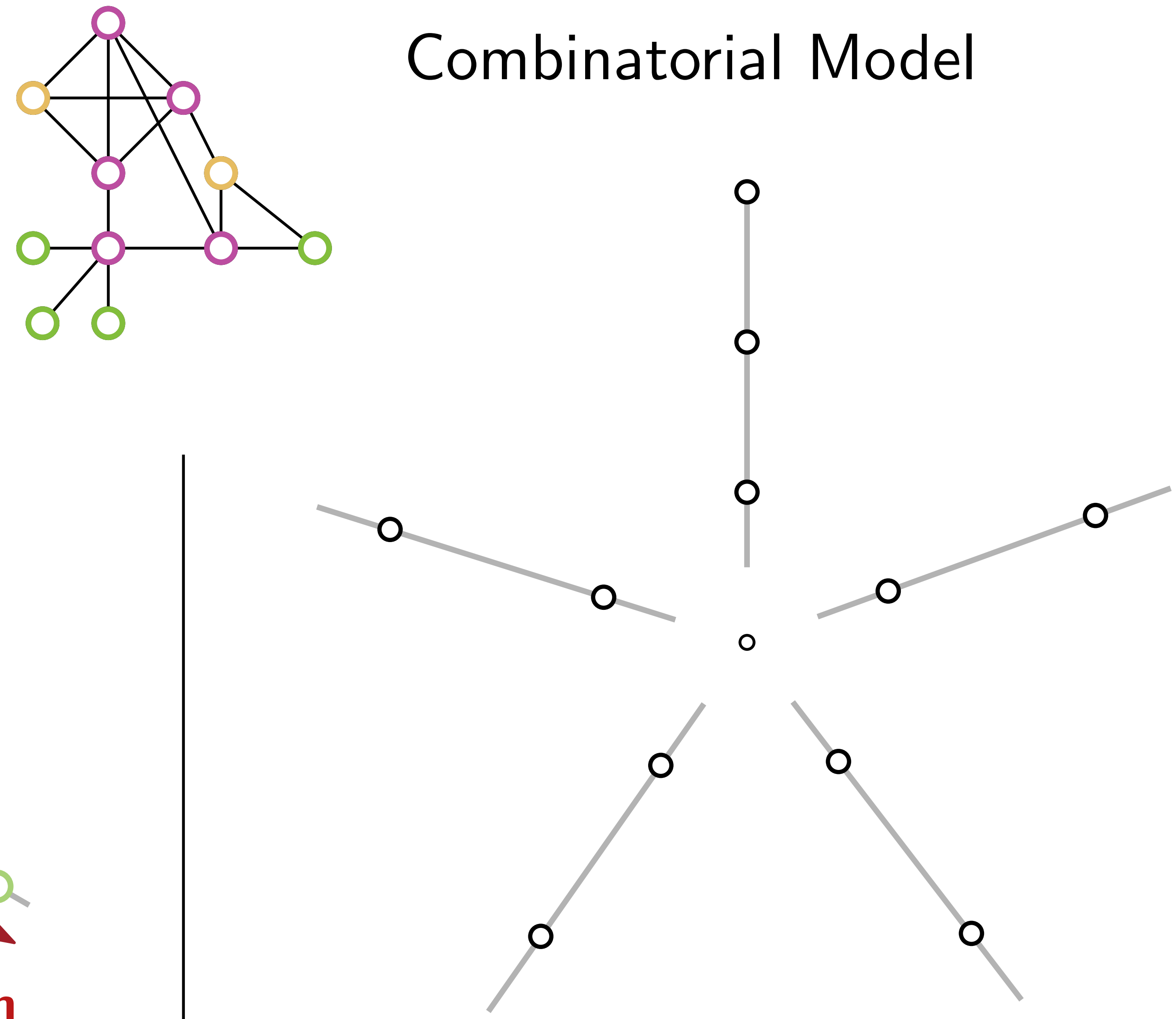
* [Krzywinski et al., 2012]

What is a Hive Plot?

Hive Plot*



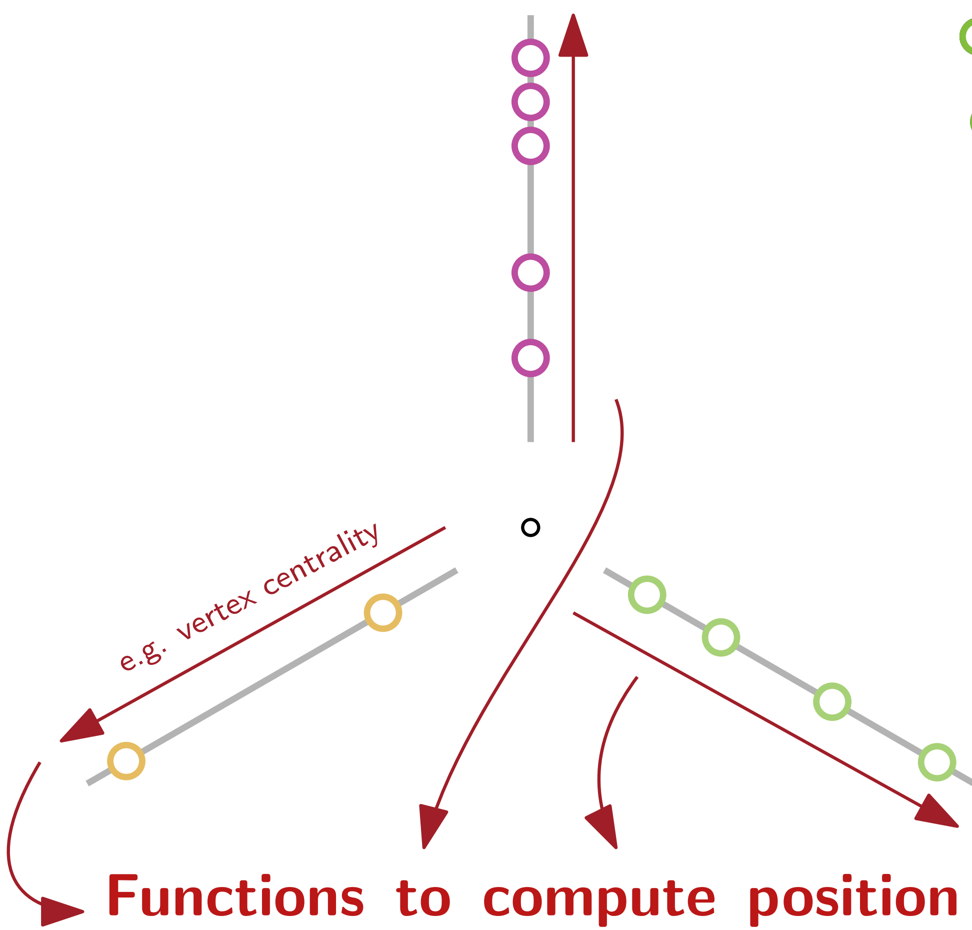
Combinatorial Model



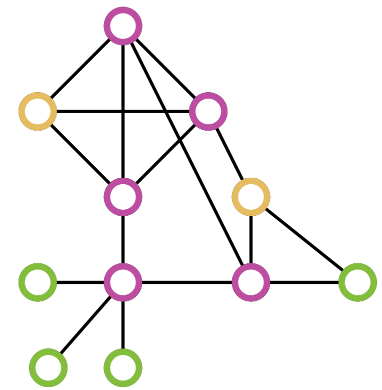
* [Krzywinski et al., 2012]

What is a Hive Plot?

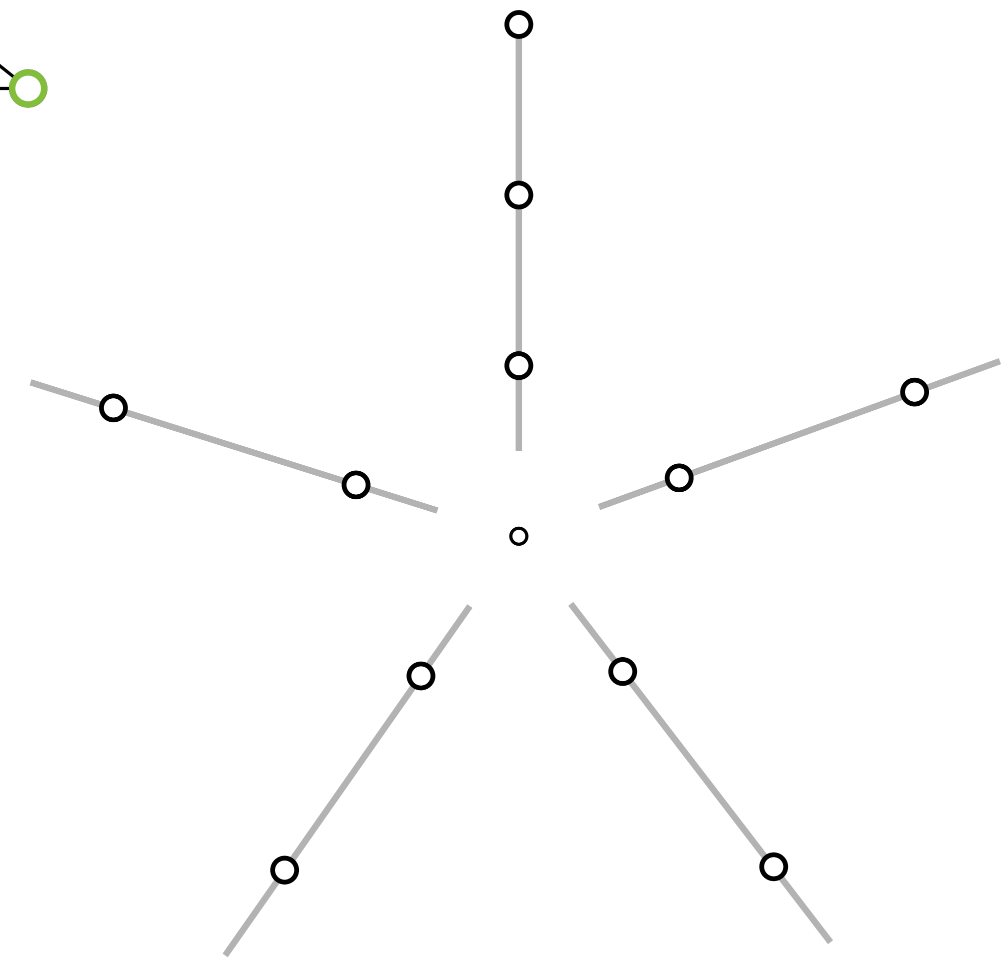
Hive Plot*



Functions to compute position in 1D space



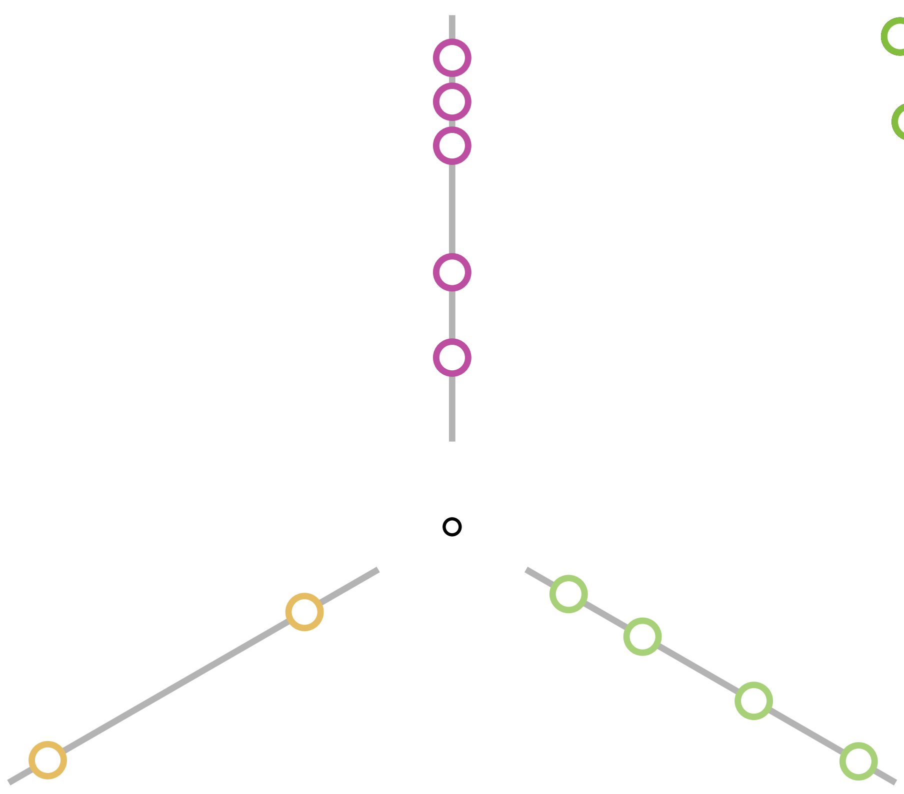
Combinatorial Model

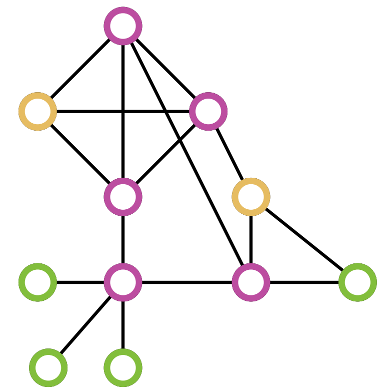


* [Krzywinski et al., 2012]

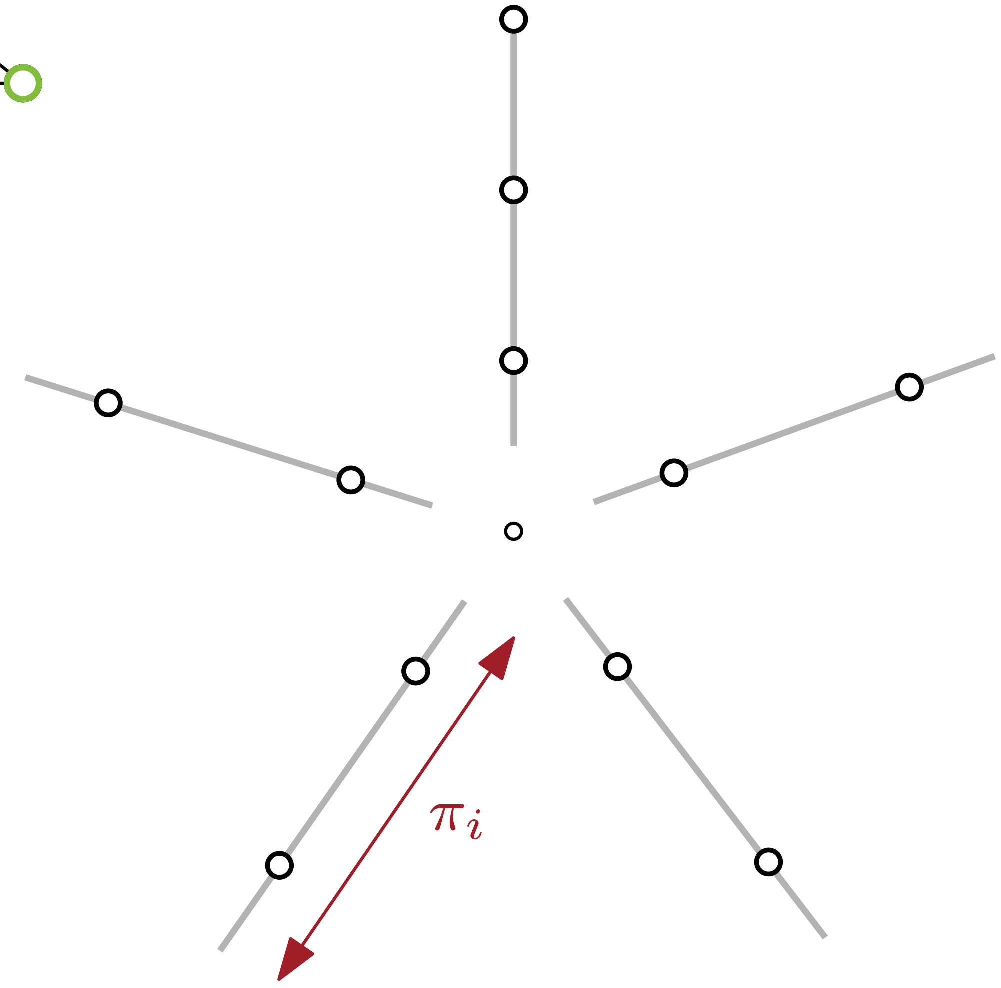
What is a Hive Plot?

Hive Plot*





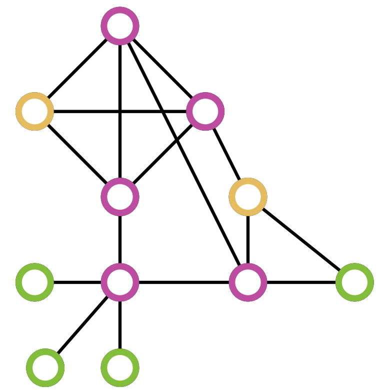
Combinatorial Model



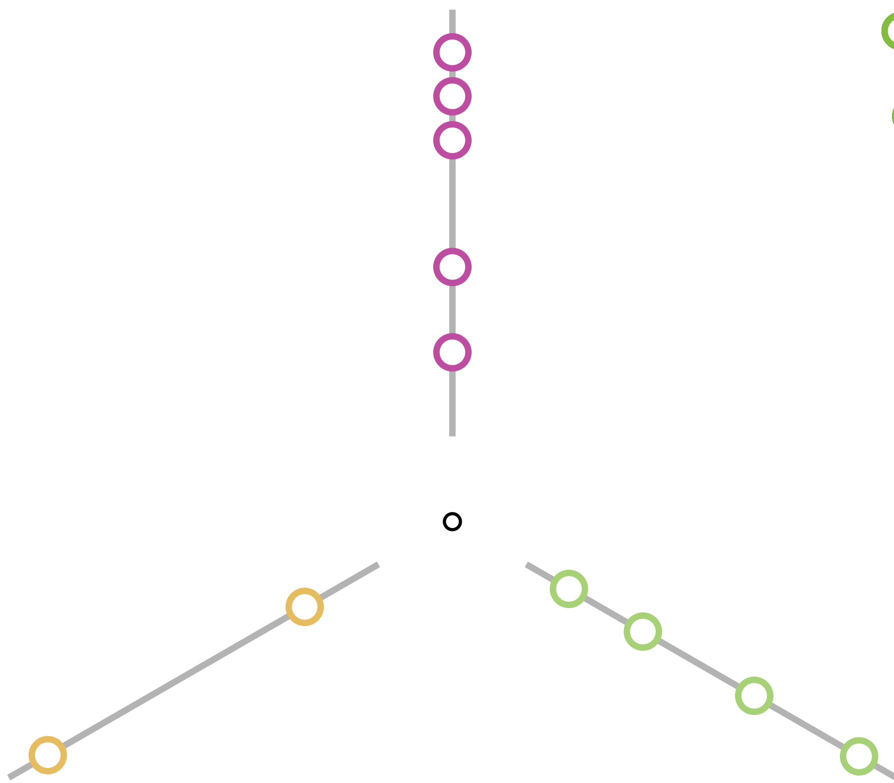
Vertex position DOF

* [Krzywinski et al., 2012]

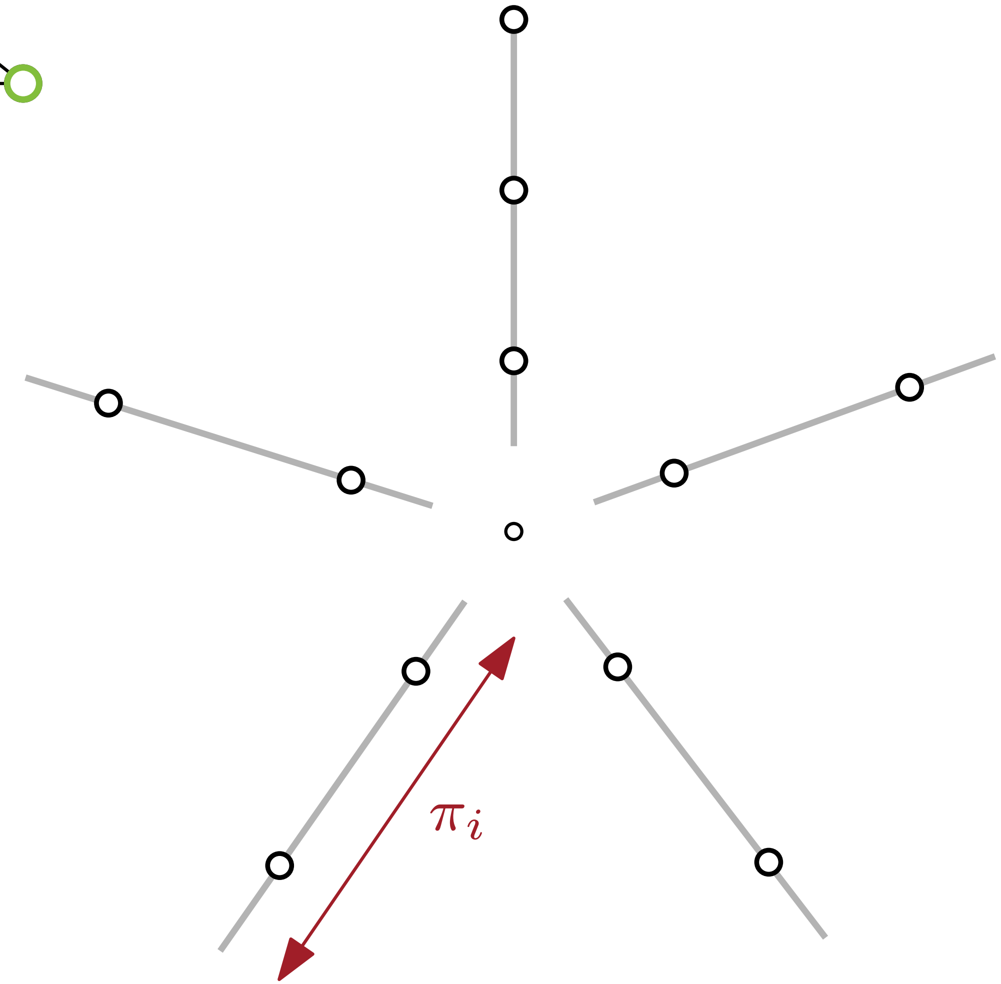
What is a Hive Plot?



Hive Plot*



Combinatorial Model

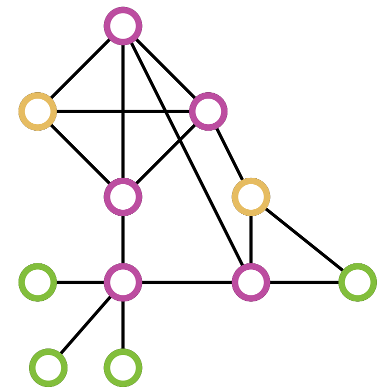


Vertex position DOF

* [Krzywinski et al., 2012]

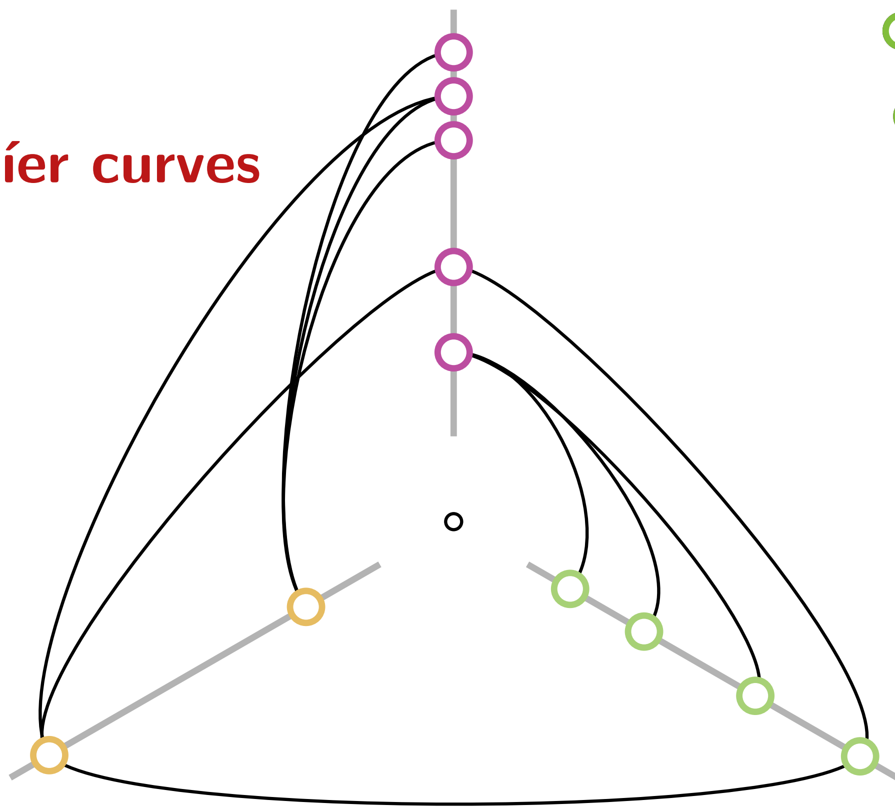
What is a Hive Plot?

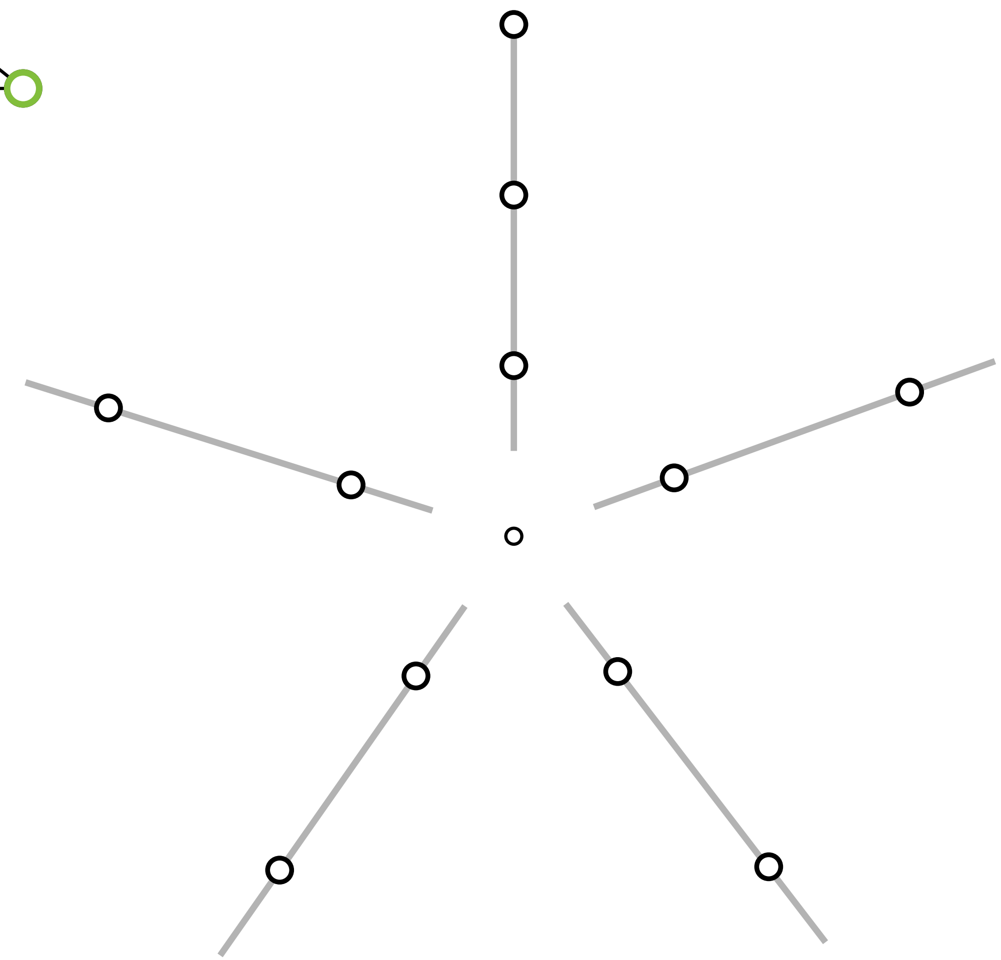
Hive Plot*



Combinatorial Model

Bezíer curves

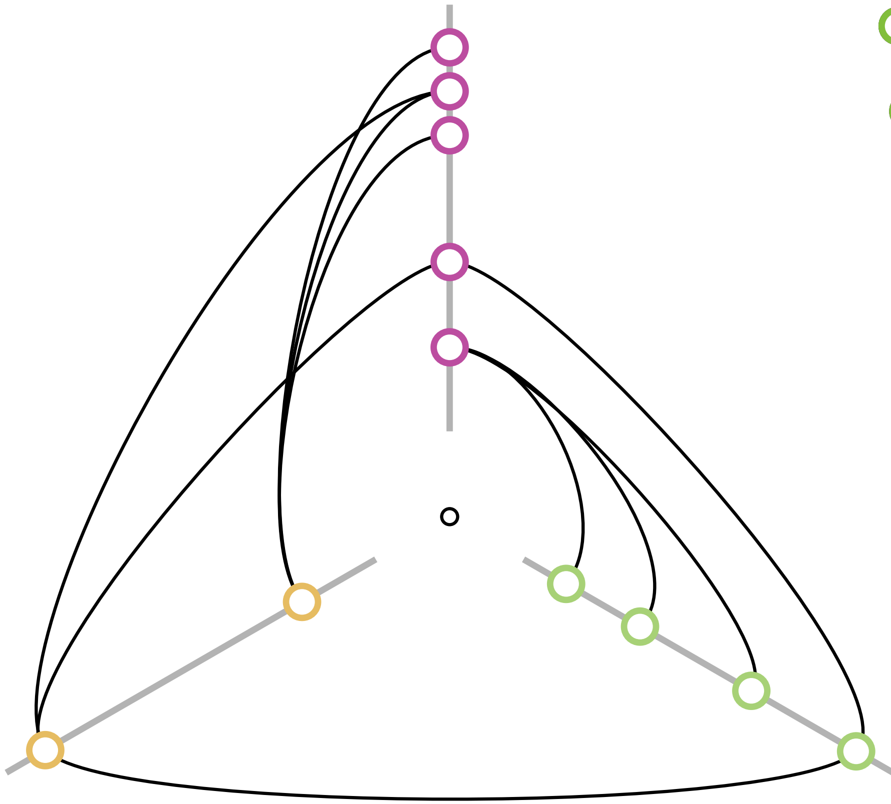


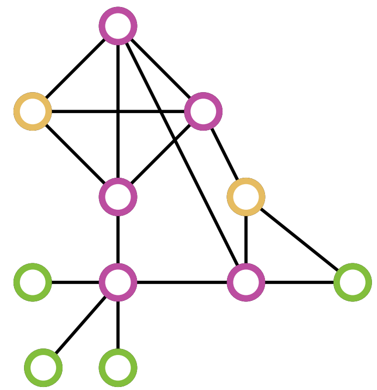


* [Krzywinski et al., 2012]

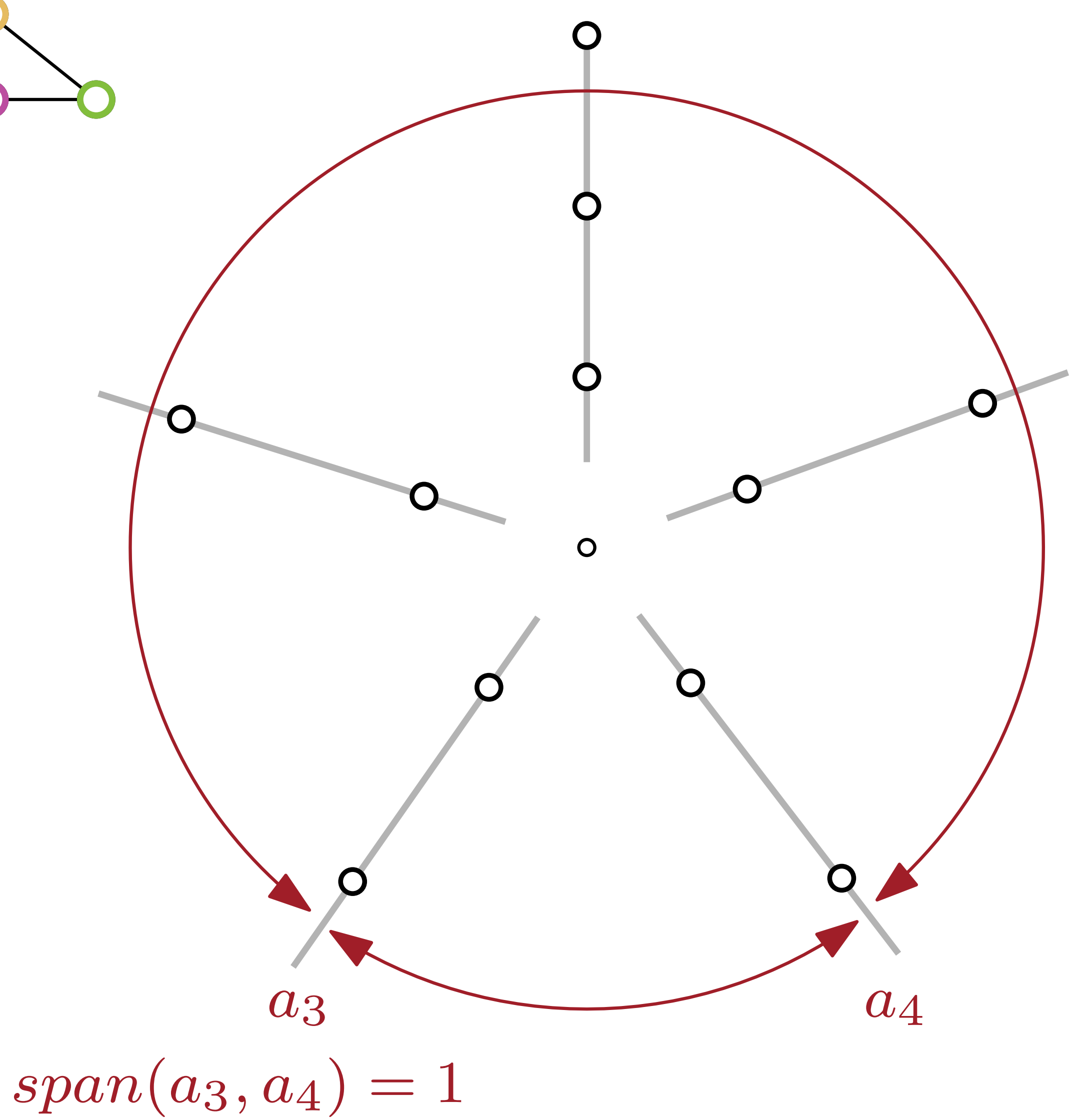
What is a Hive Plot?

Hive Plot*





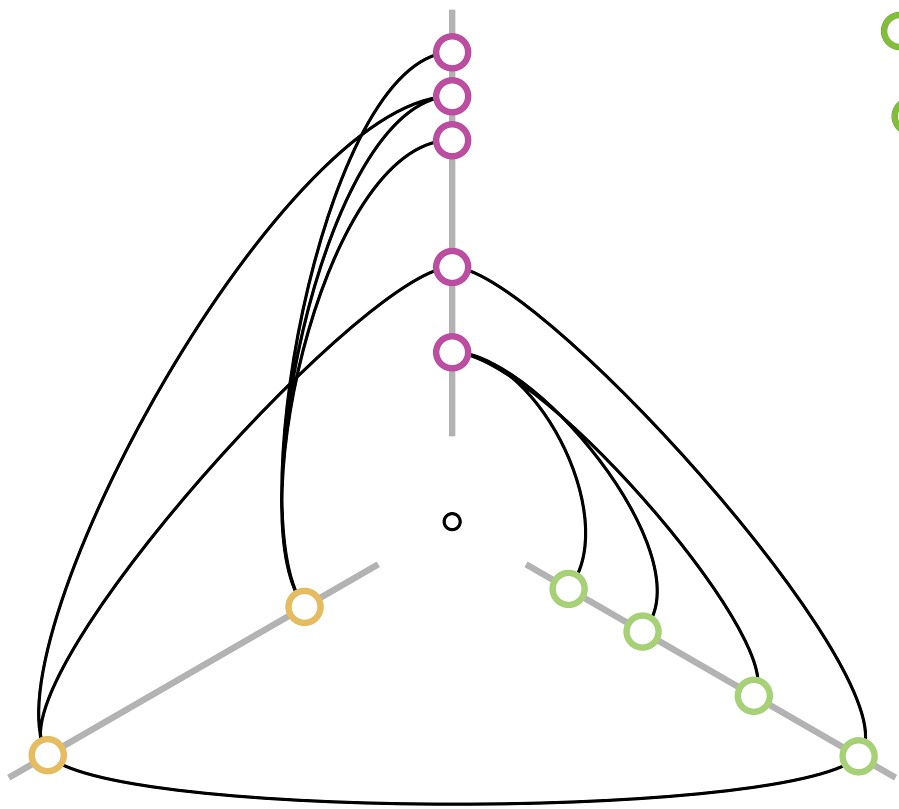
Combinatorial Model

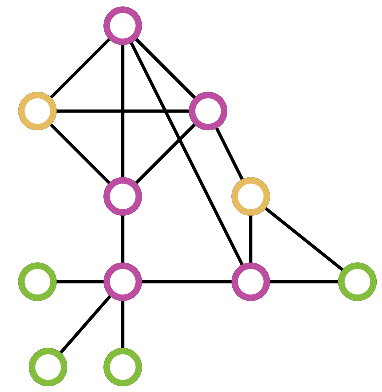


* [Krzywinski et al., 2012]

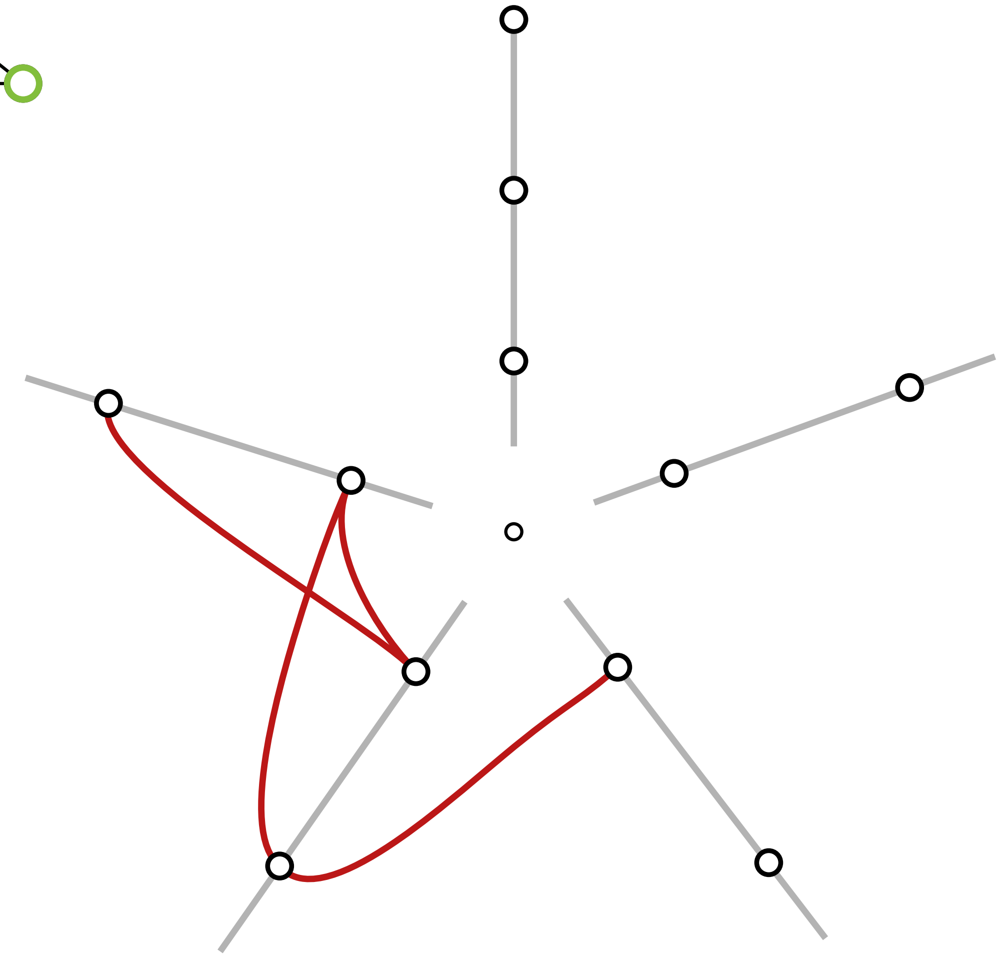
What is a Hive Plot?

Hive Plot*





Combinatorial Model



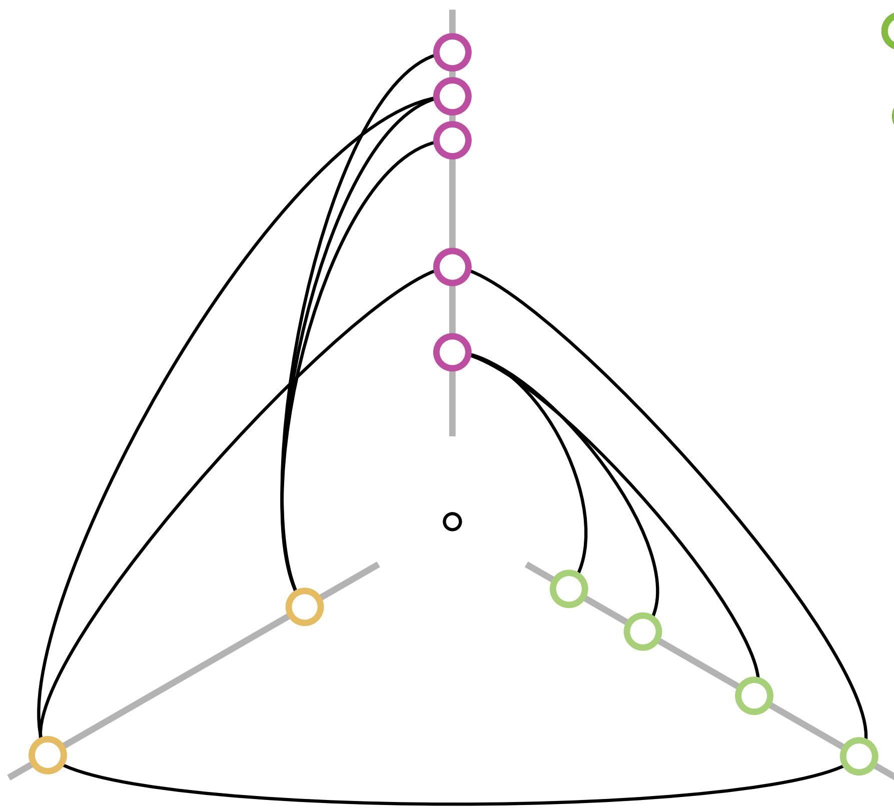
$span(u, v) = 1$

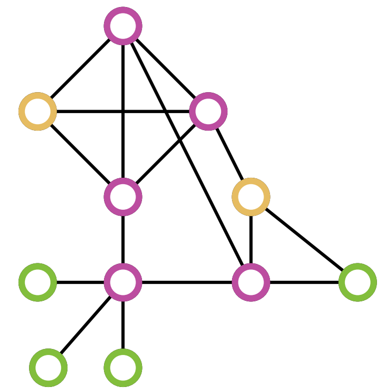
Proper edge

* [Krzywinski et al., 2012]

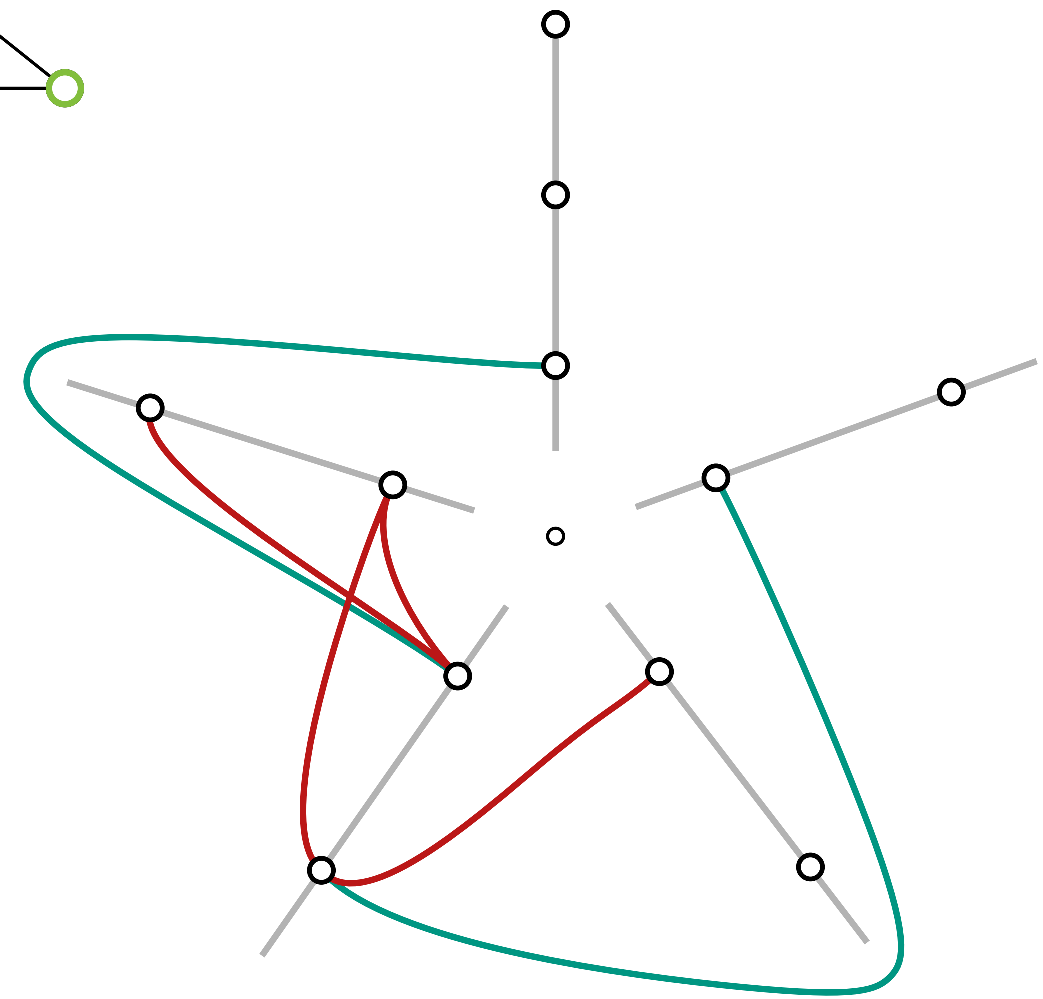
What is a Hive Plot?

Hive Plot*





Combinatorial Model

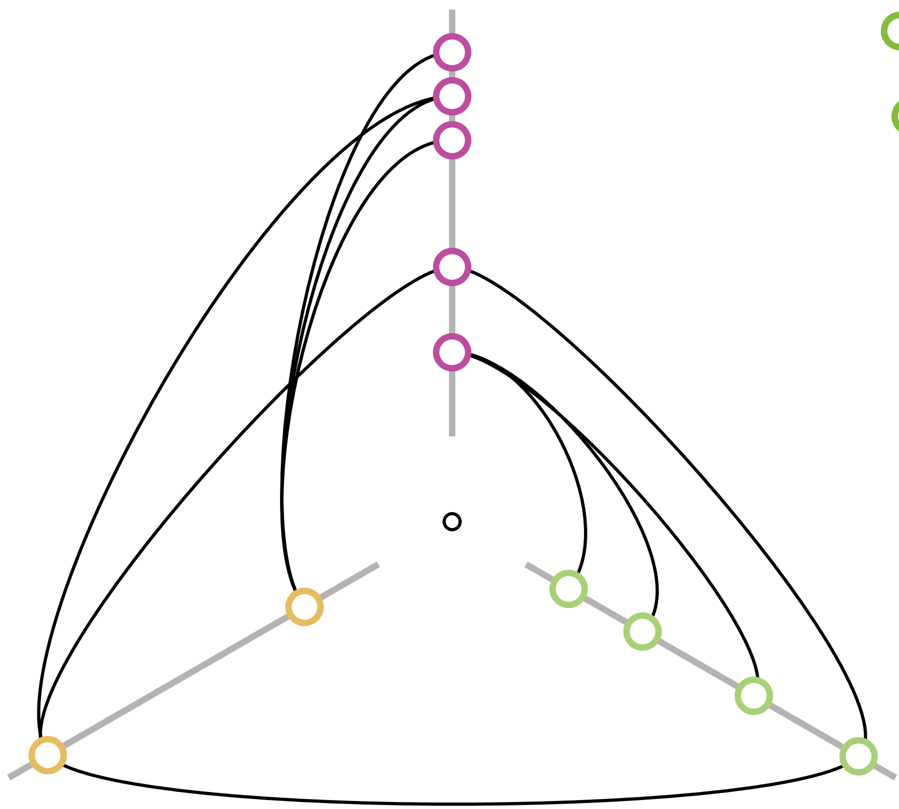


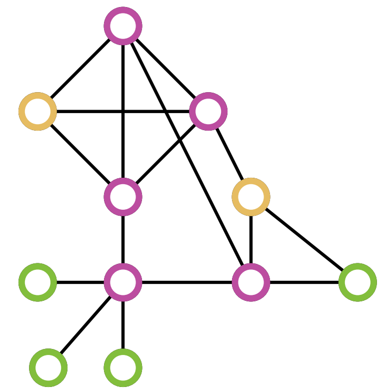
$span(u, v) = 1$ Proper edge
$span(u, v) > 1$ Long edge
(Inter-axis edge)

* [Krzywinski et al., 2012]

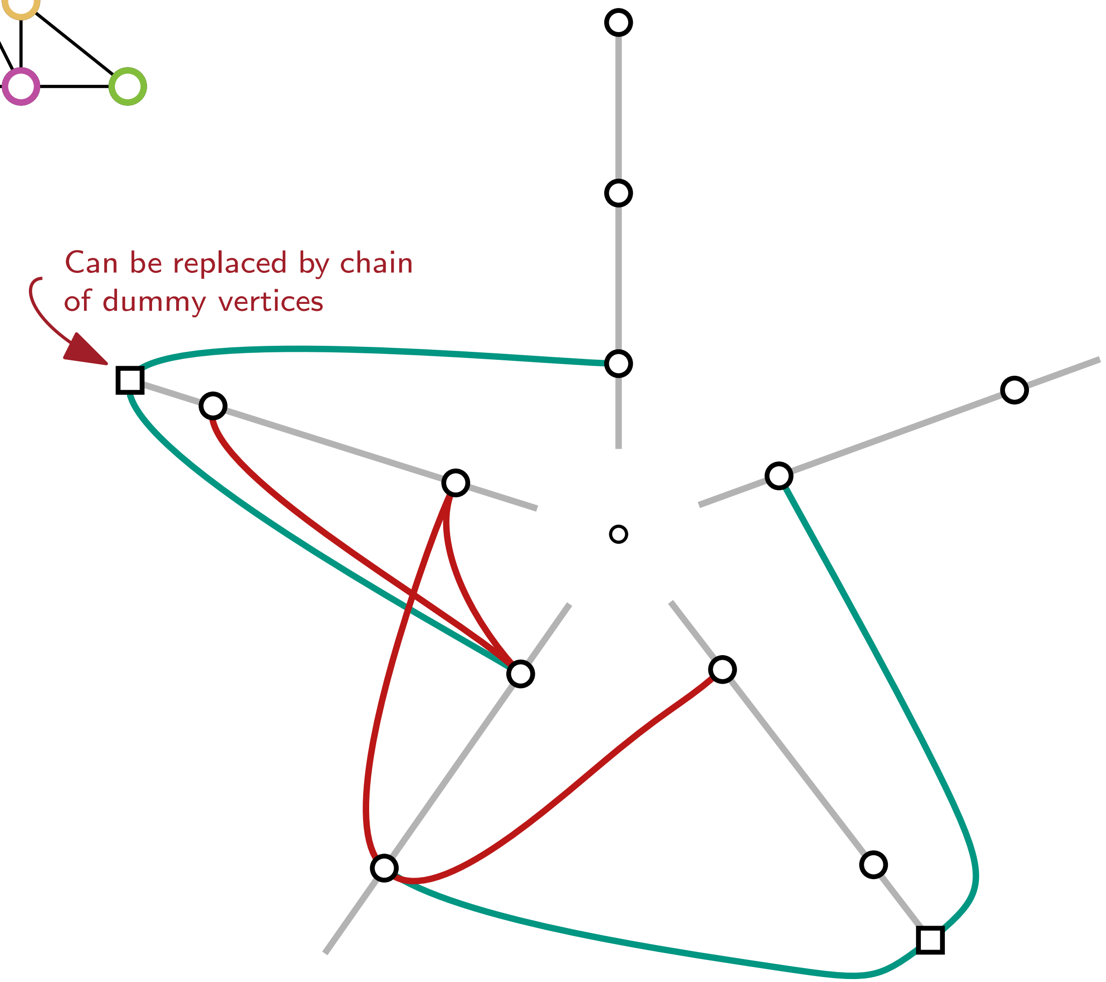
What is a Hive Plot?

Hive Plot*





Combinatorial Model



$span(u, v) = 1$ Proper edge

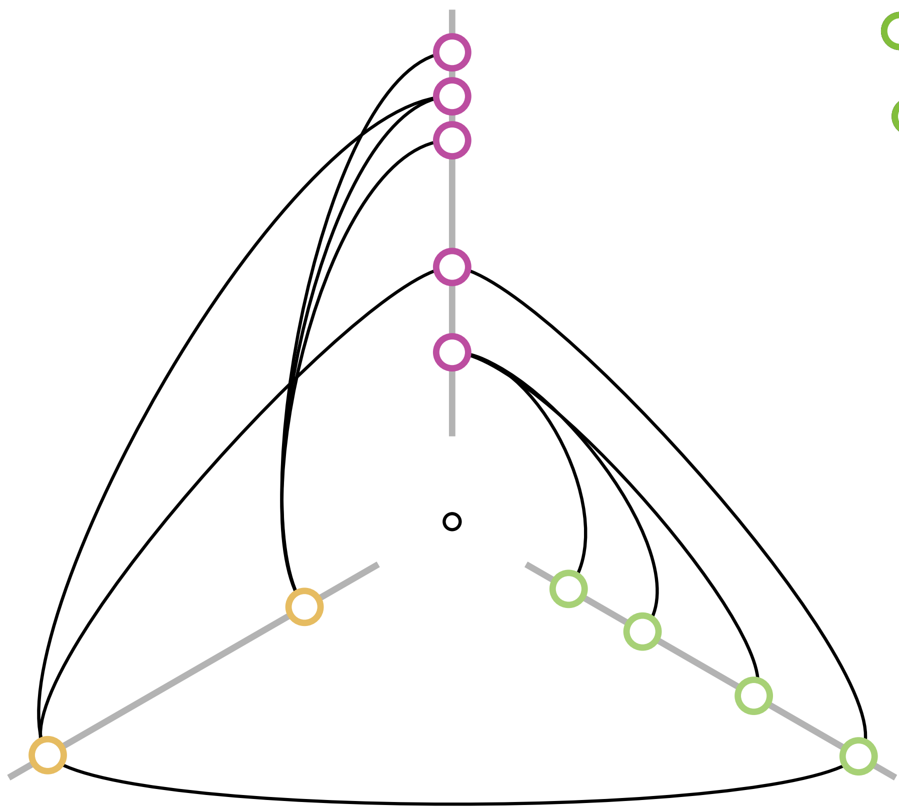
$span(u, v) > 1$ Long edge

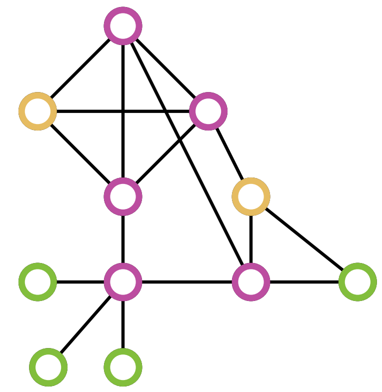
(Inter-axis edge)

* [Krzywinski et al., 2012]

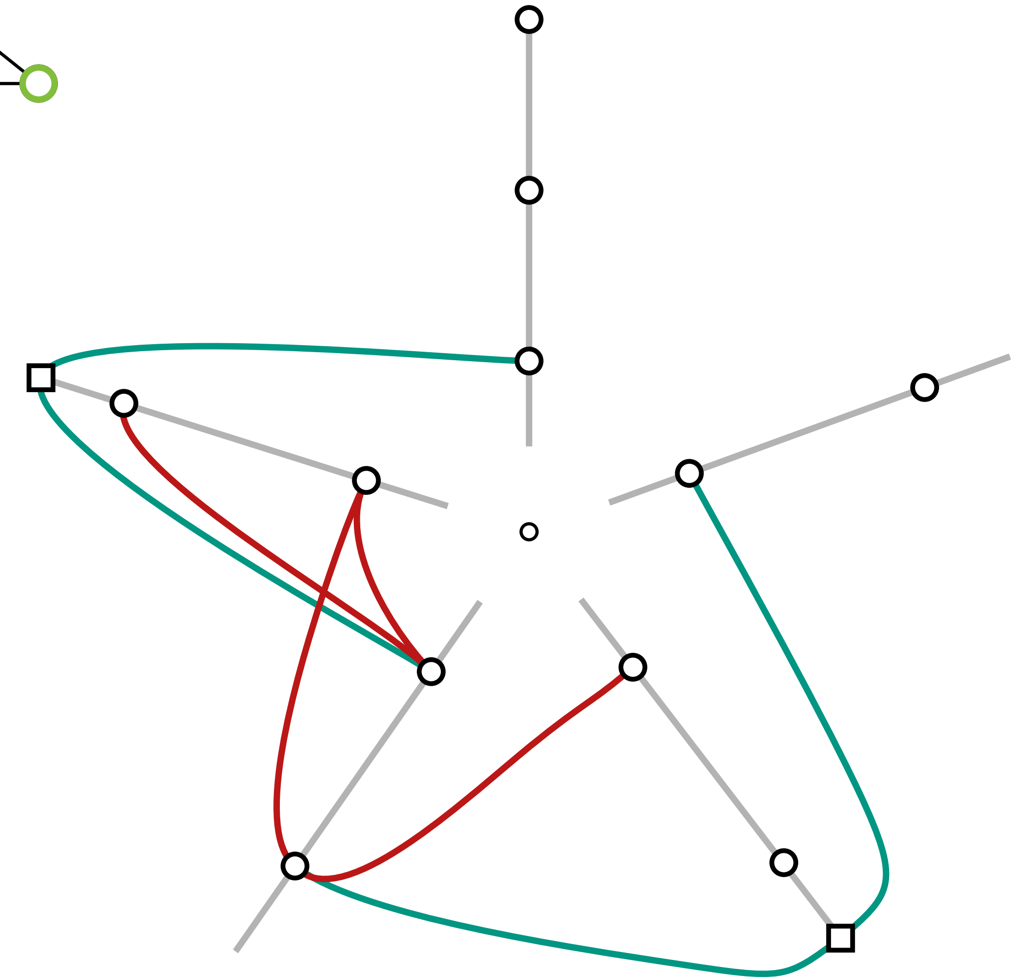
What is a Hive Plot?

Hive Plot*





Combinatorial Model



What about the remaining edges?

Duplicate axes and vertices

* [Krzywinski et al., 2012]

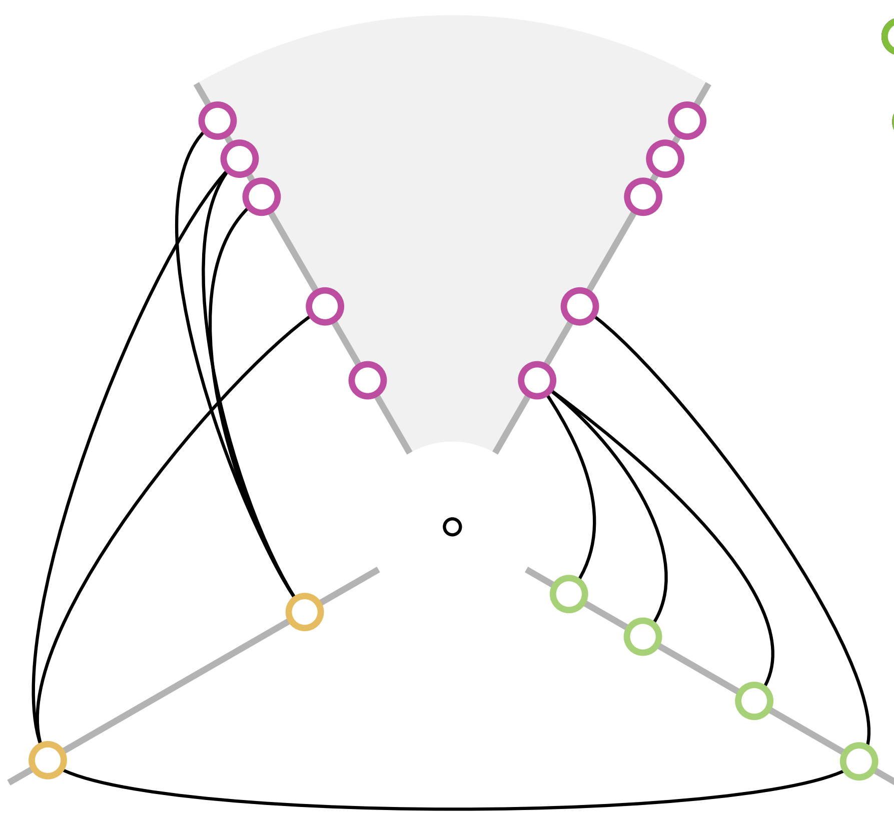
$span(u, v) = 1$ Proper edge

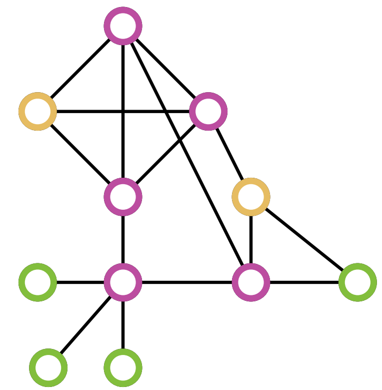
$span(u, v) > 1$ Long edge

(Inter-axis edge)

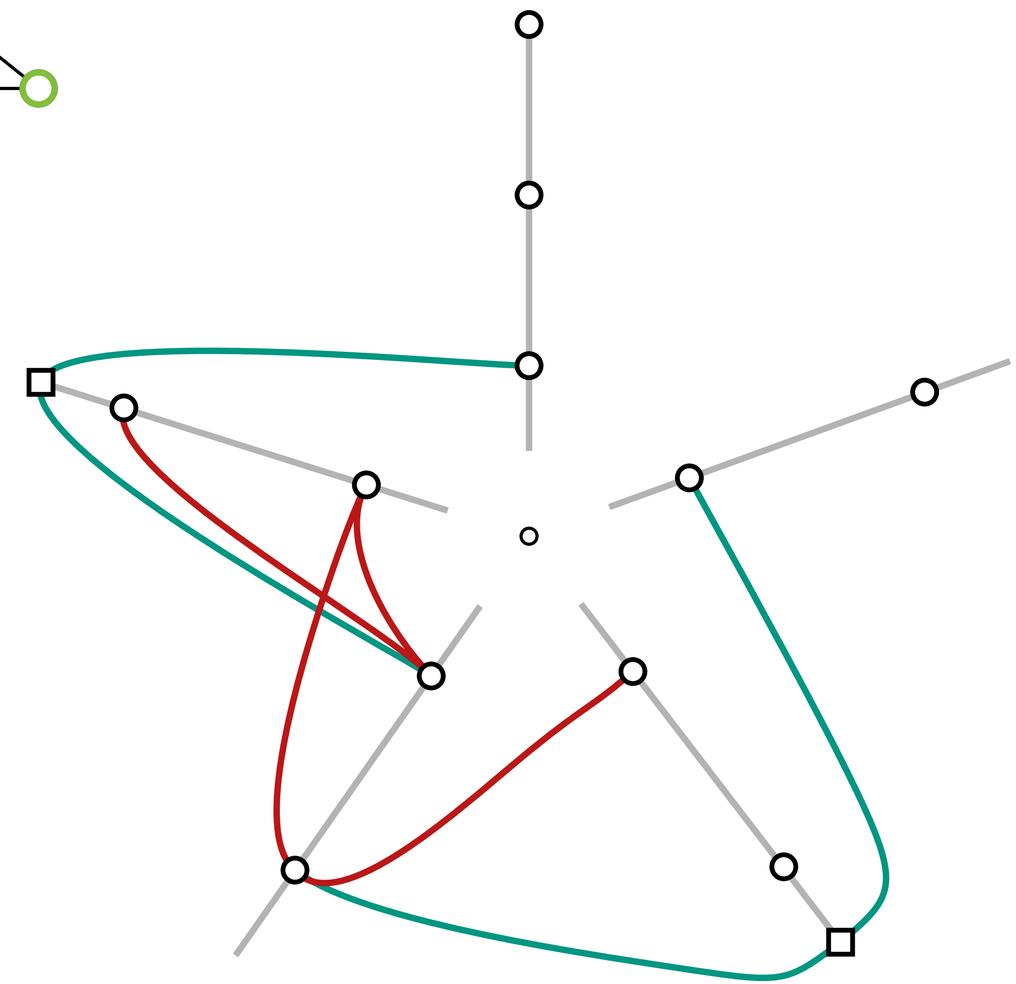
What is a Hive Plot?

Hive Plot*





Combinatorial Model



$span(u, v) = 1$ Proper edge
$span(u, v) > 1$ Long edge
(Inter-axis edge)

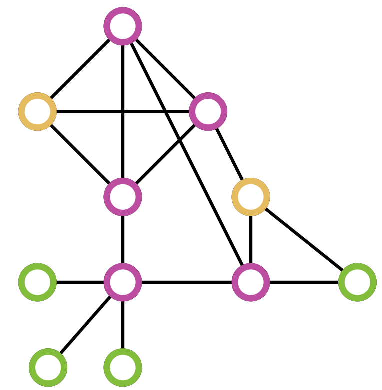
What about the remaining edges?

Duplicate axes and vertices

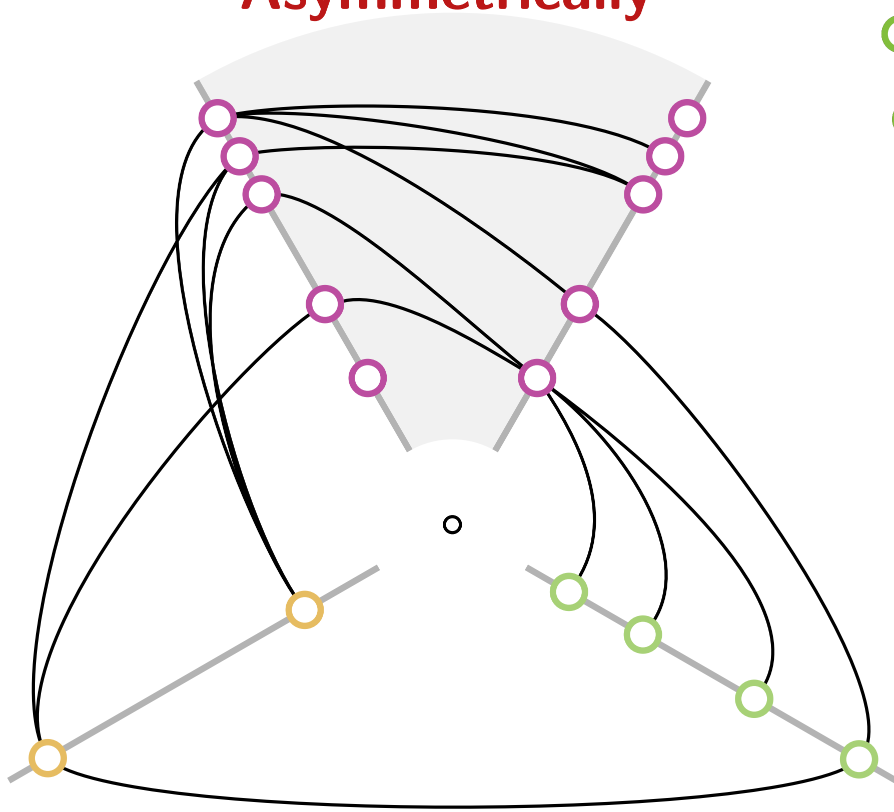
* [Krzywinski et al., 2012]

What is a Hive Plot?

Hive Plot*



Asymmetrically

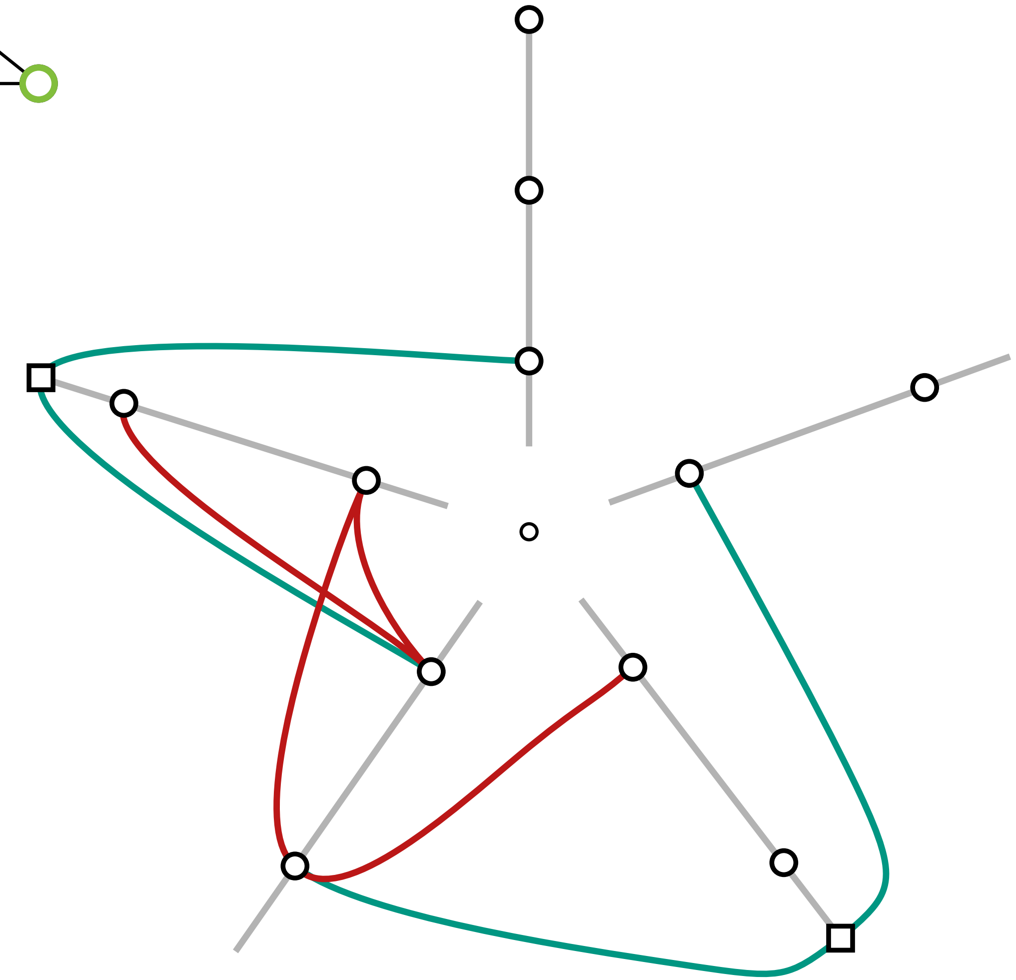


What about the remaining edges?

Duplicate axes and vertices

* [Krzywinski et al., 2012]

Combinatorial Model



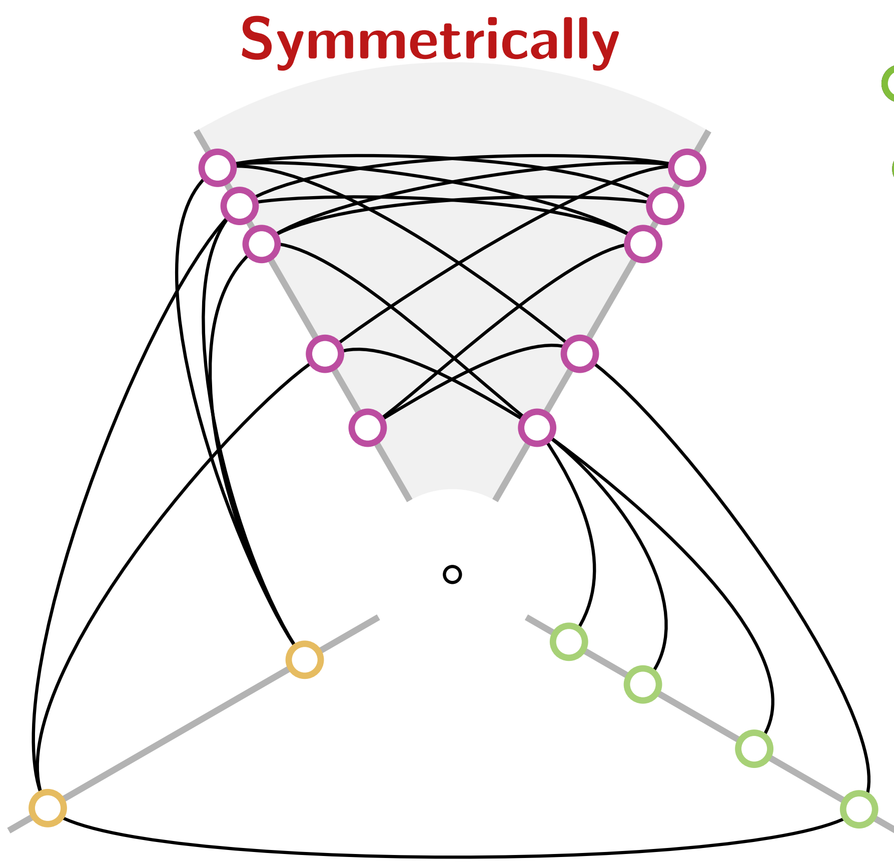
$span(u, v) = 1$ Proper edge

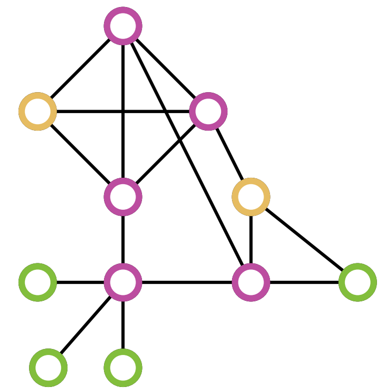
$span(u, v) > 1$ Long edge

(Inter-axis edge)

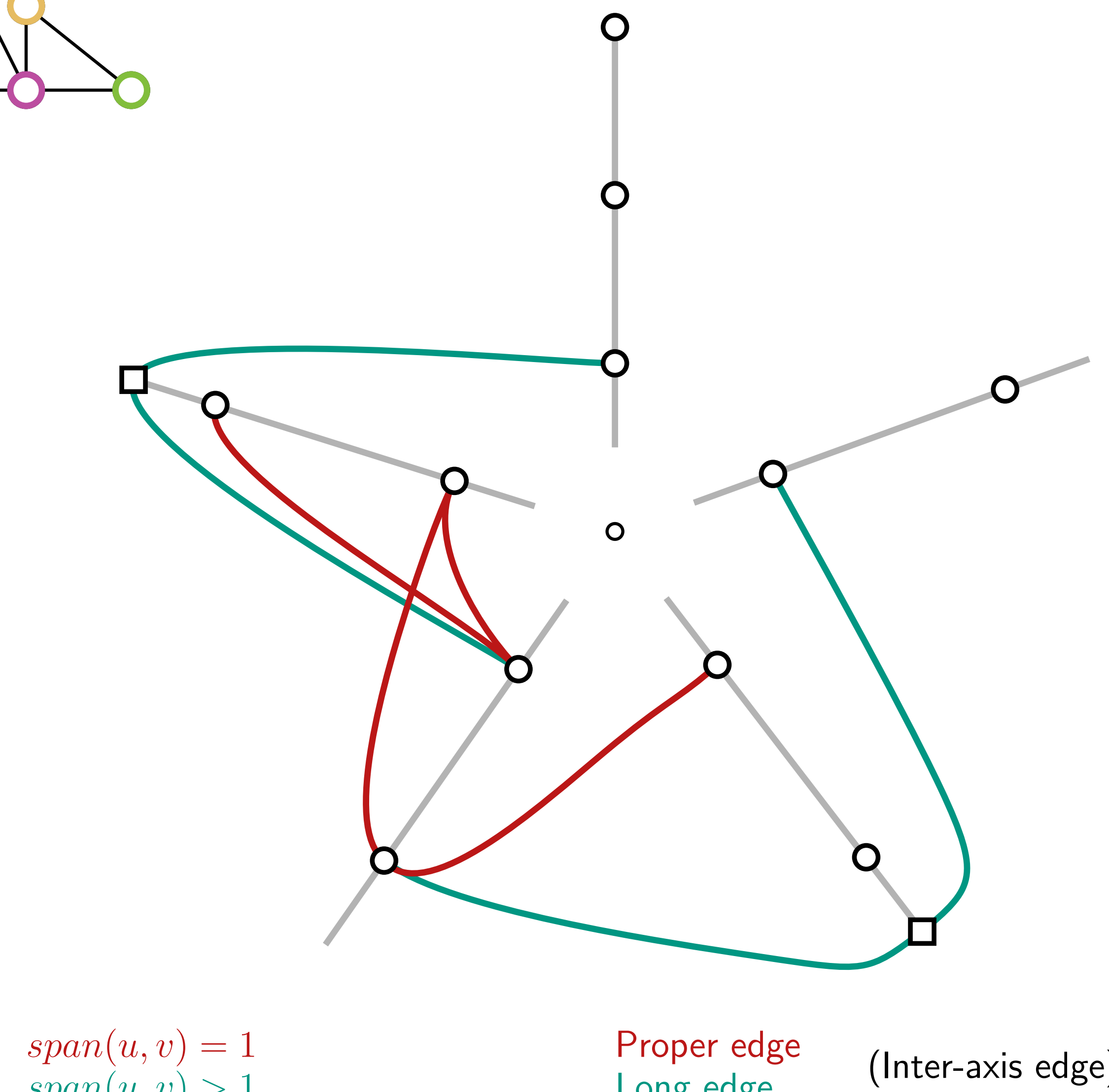
What is a Hive Plot?

Hive Plot*





Combinatorial Model



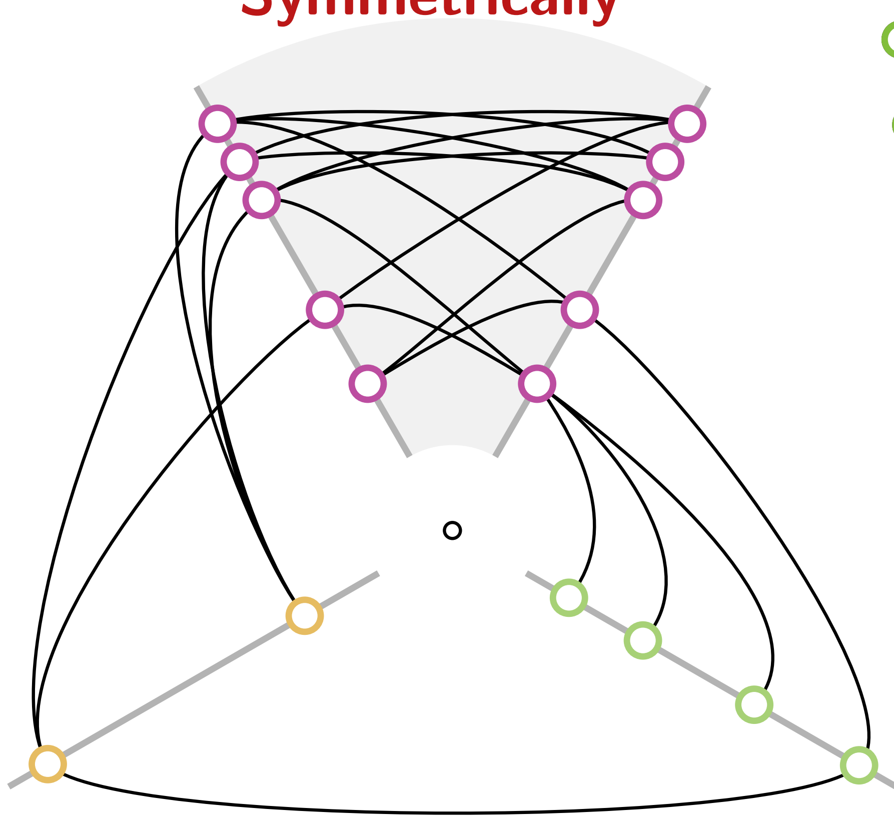
What about the remaining edges?
Duplicate axes and vertices

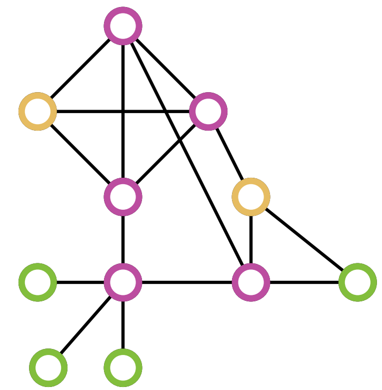
* [Krzywinski et al., 2012]

What is a Hive Plot?

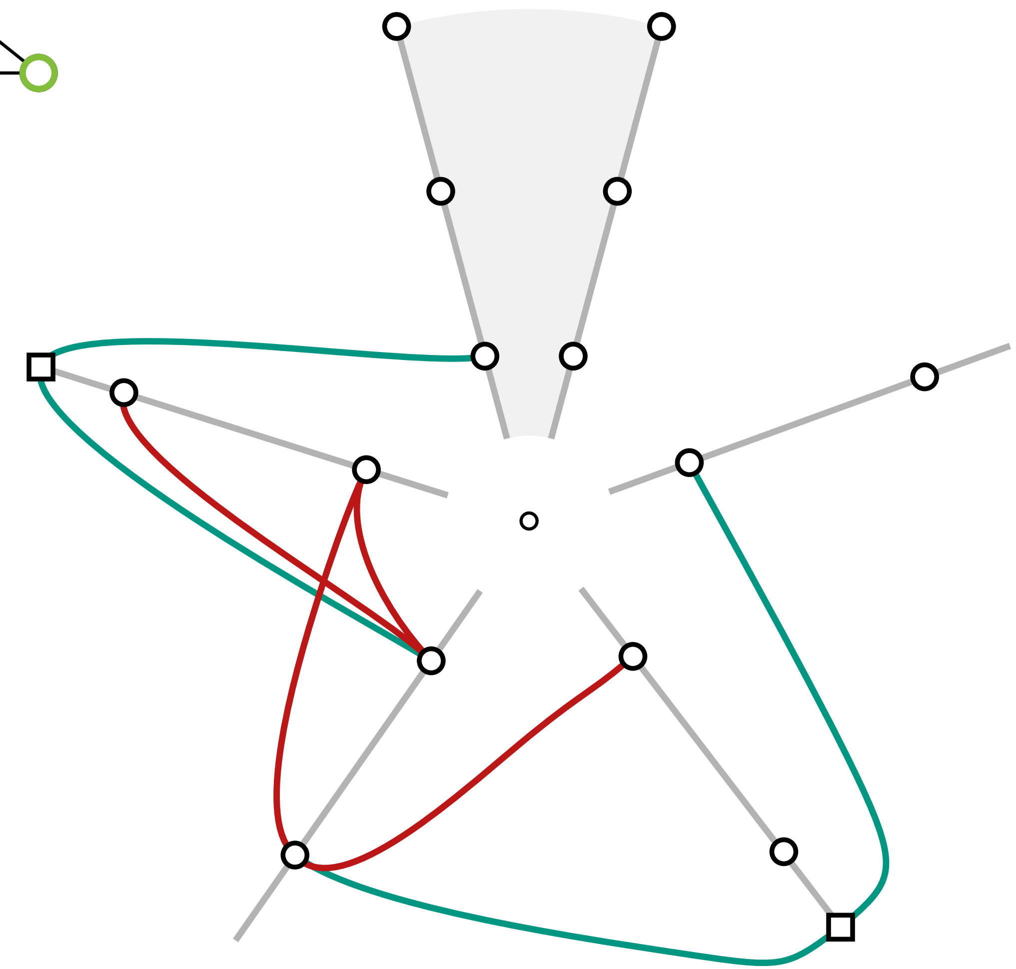
Hive Plot*

Symmetrically





Combinatorial Model



What about the remaining edges?

Duplicate axes and vertices

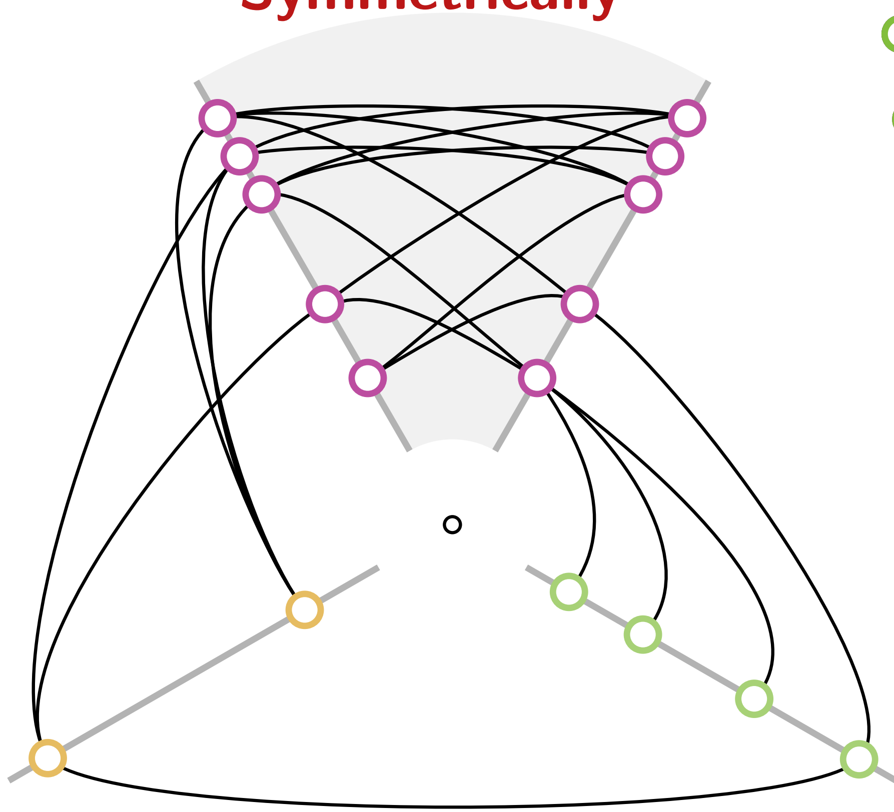
* [Krzywinski et al., 2012]

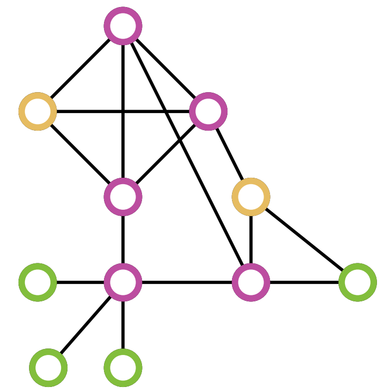
$span(u, v) = 1$ Proper edge
$span(u, v) > 1$ Long edge
(Inter-axis edge)

What is a Hive Plot?

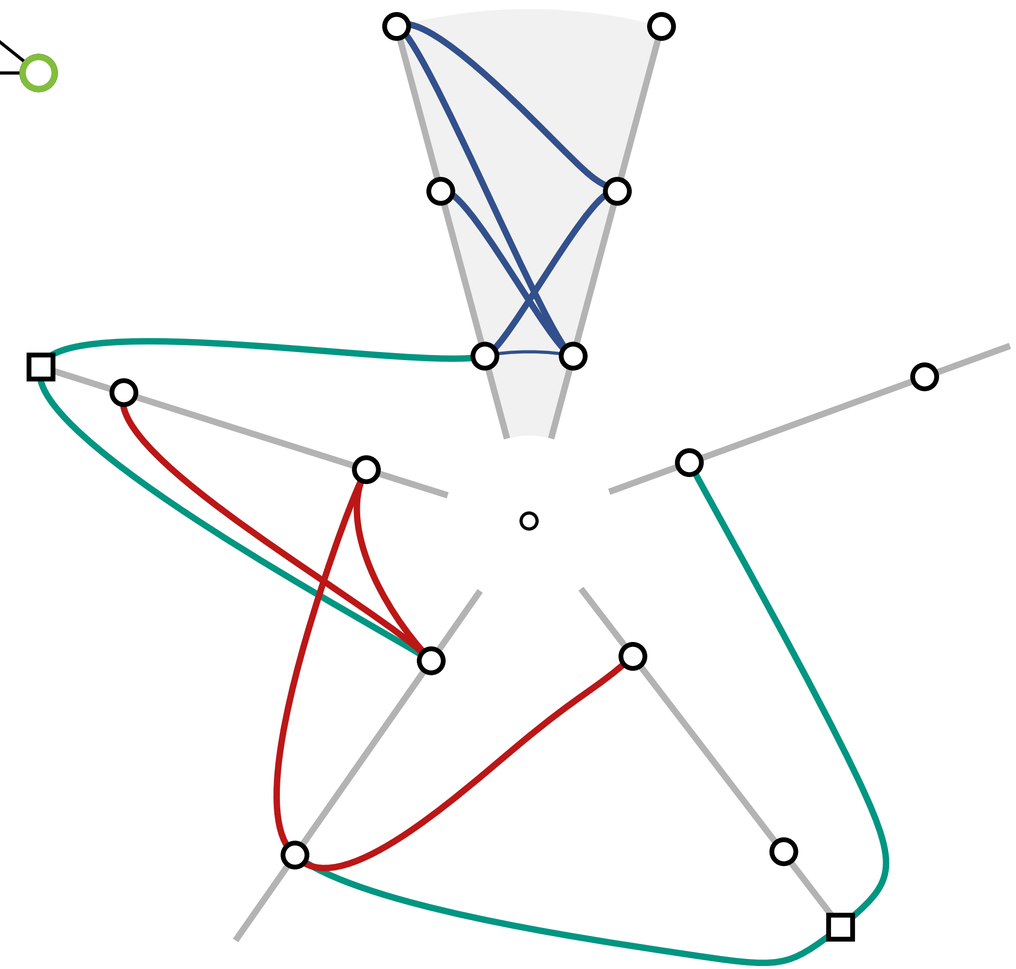
Hive Plot*

Symmetrically





Combinatorial Model



What about the remaining edges?

Duplicate axes and vertices

$span(u, v) = 0$ Intra-axis edge

$span(u, v) = 1$ Proper edge

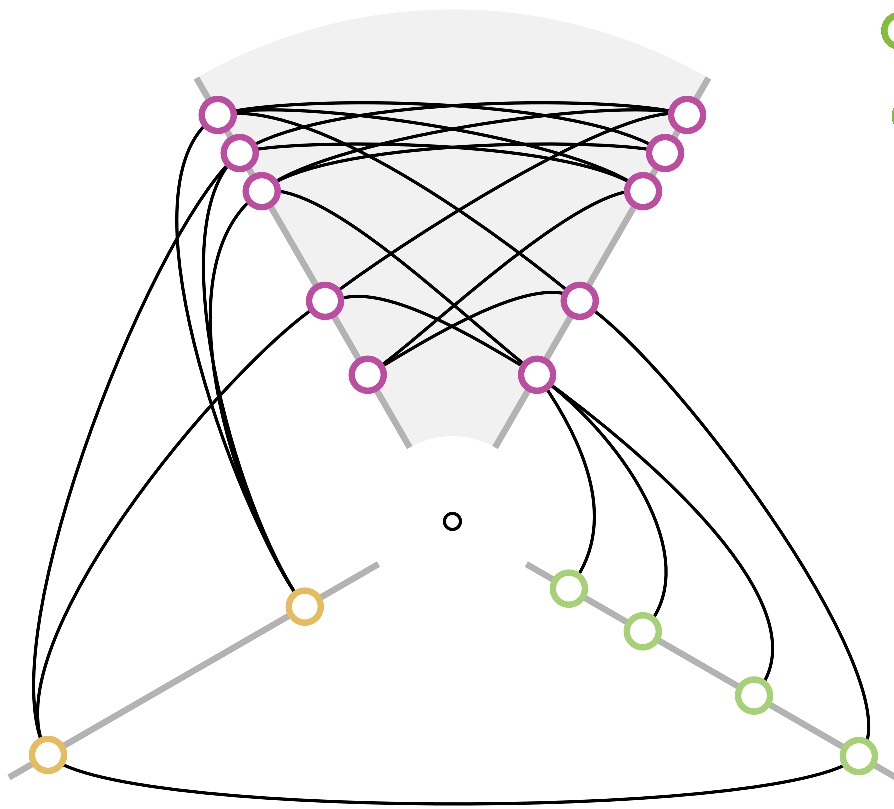
$span(u, v) > 1$ Long edge

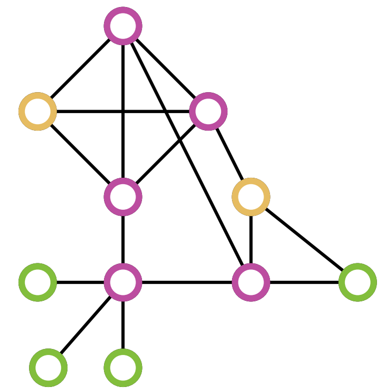
(Inter-axis edge)

* [Krzywinski et al., 2012]

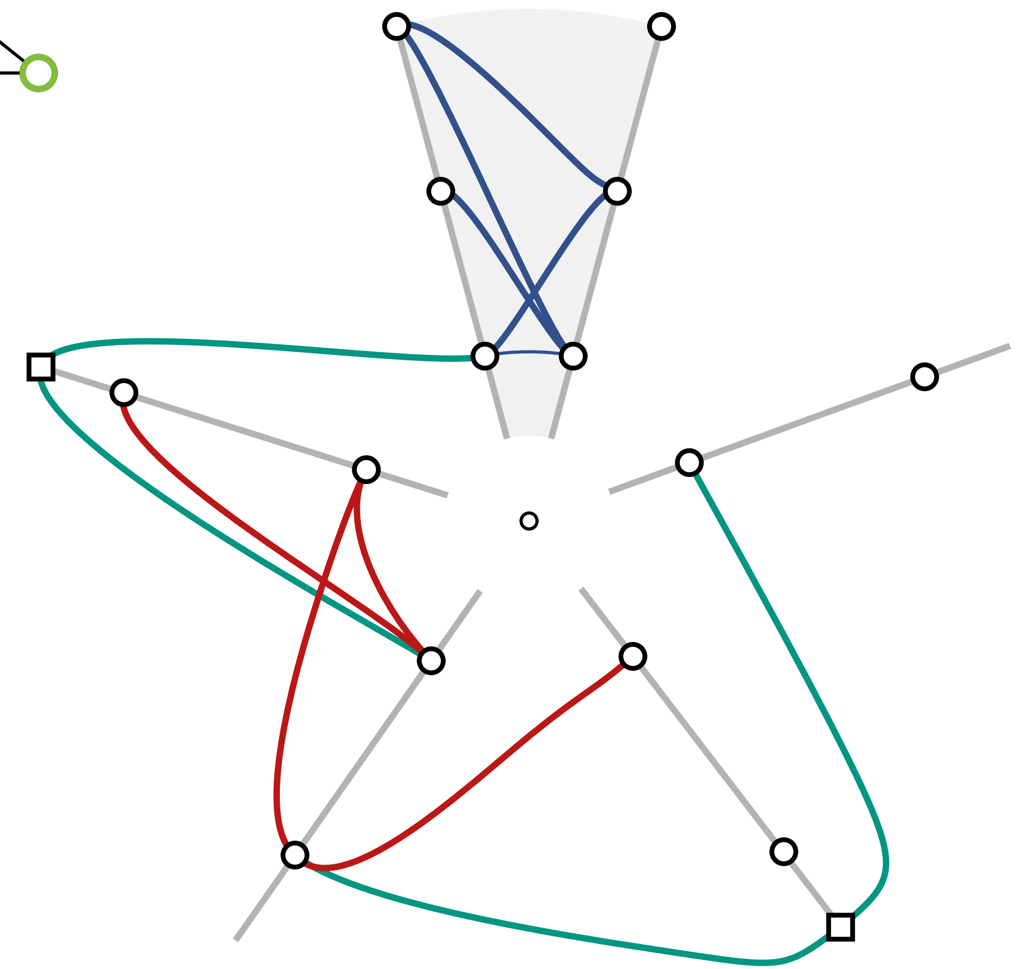
What is a Hive Plot?

Hive Plot*





Combinatorial Model

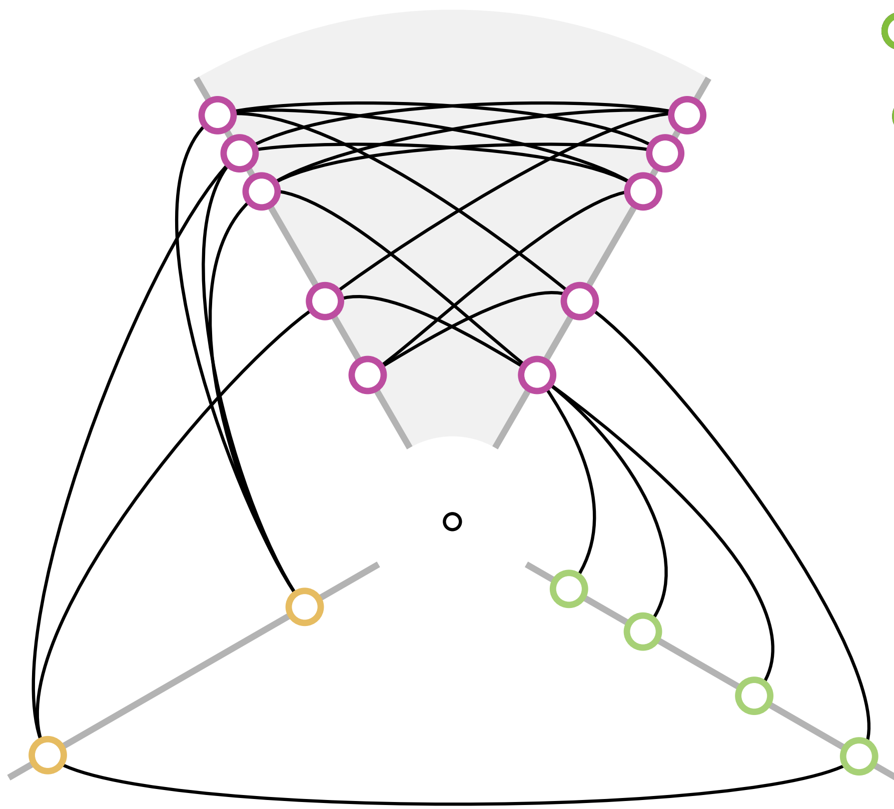


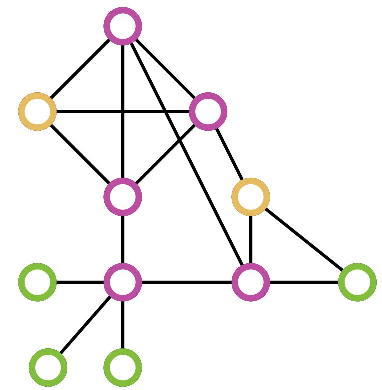
What about **routing edges** in the drawing?

* [Krzywinski et al., 2012]

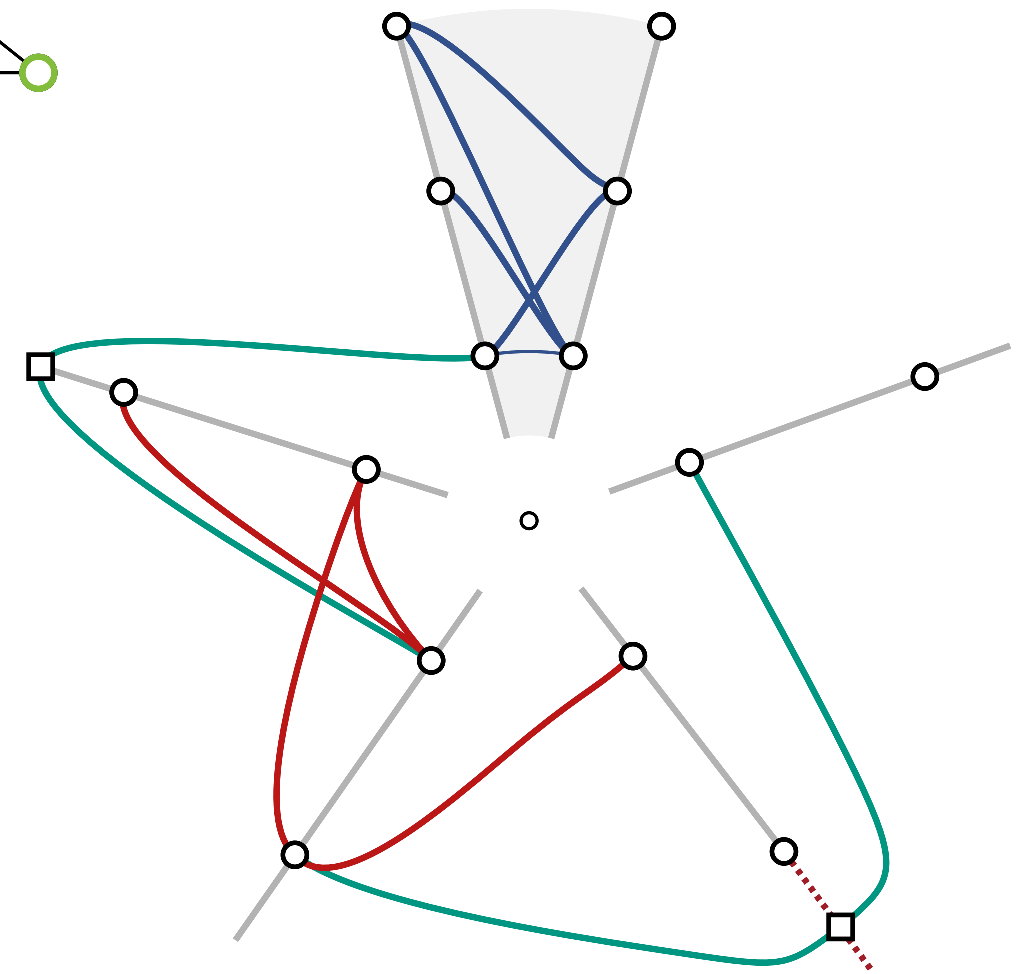
What is a Hive Plot?

Hive Plot*





Combinatorial Model



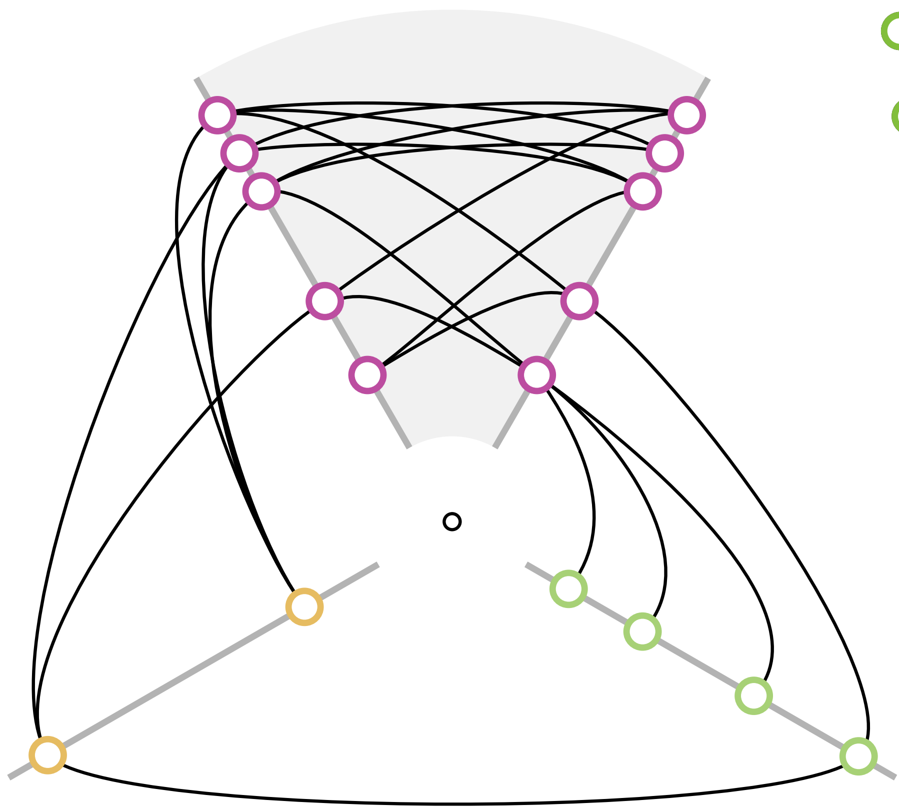
Introduce g **gaps**

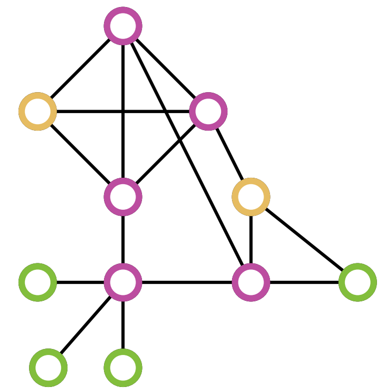
$g = 1$

* [Krzywinski et al., 2012]

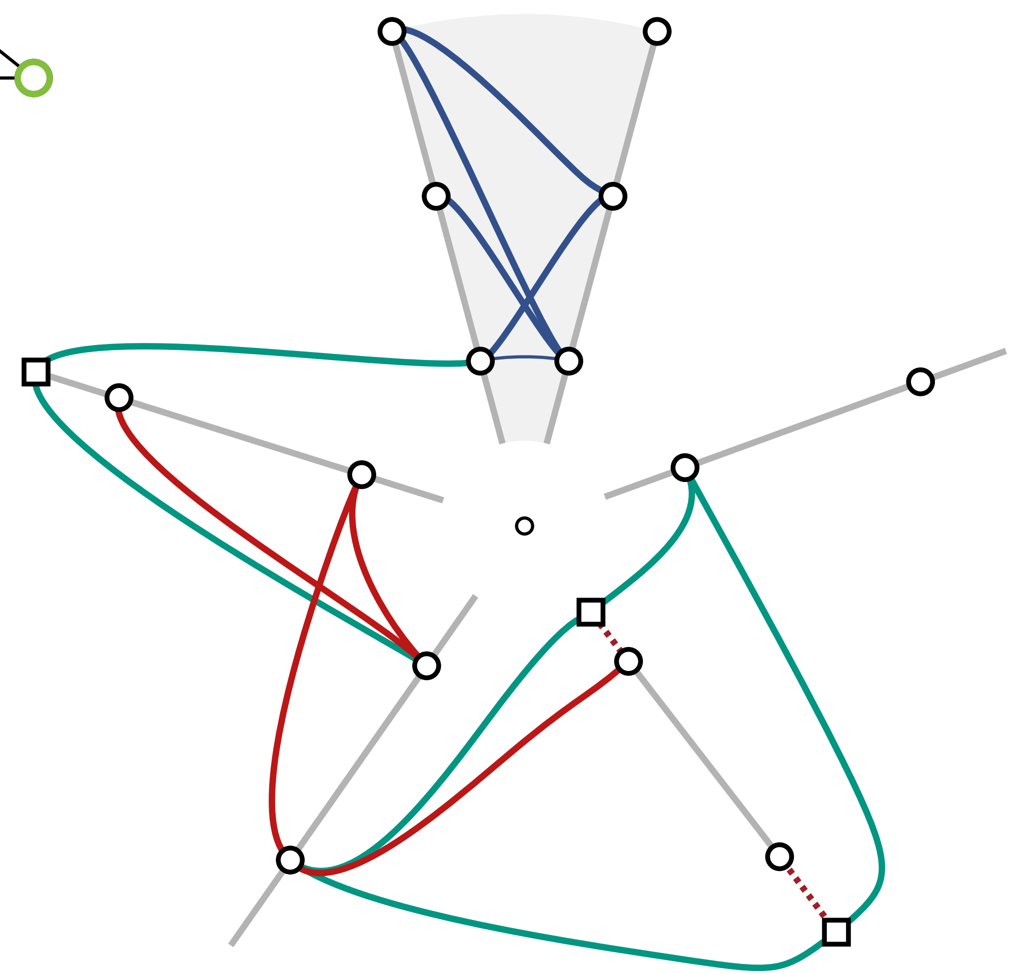
What is a Hive Plot?

Hive Plot*





Combinatorial Model



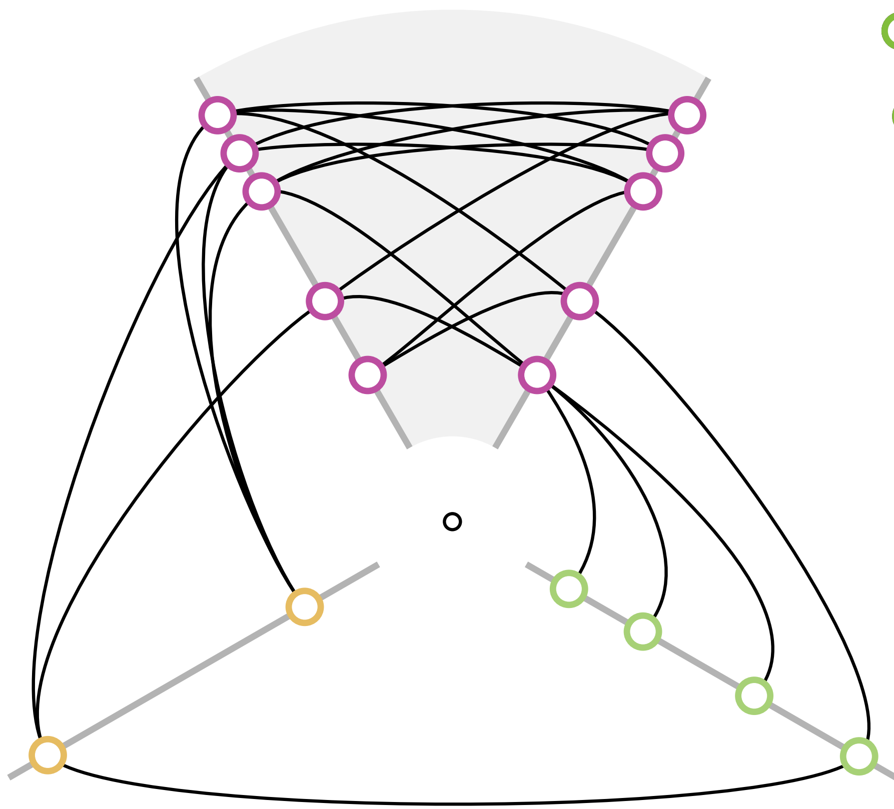
Introduce g **gaps**

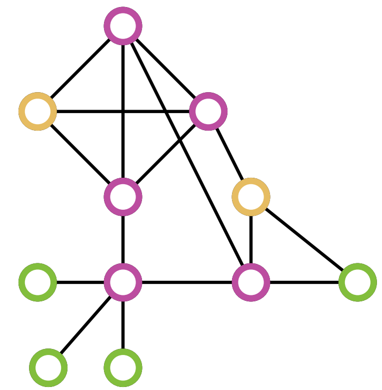
$g = 1$ $g = 2$

* [Krzywinski et al., 2012]

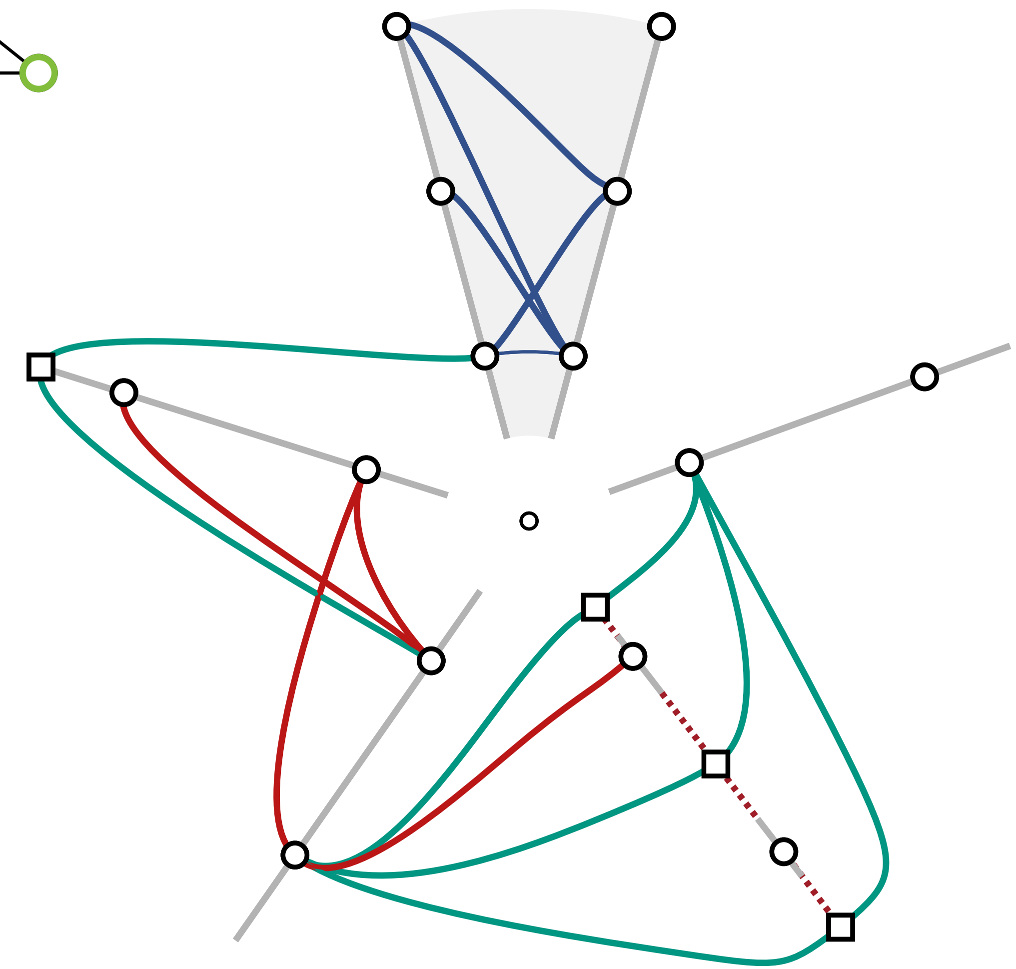
What is a Hive Plot?

Hive Plot*





Combinatorial Model



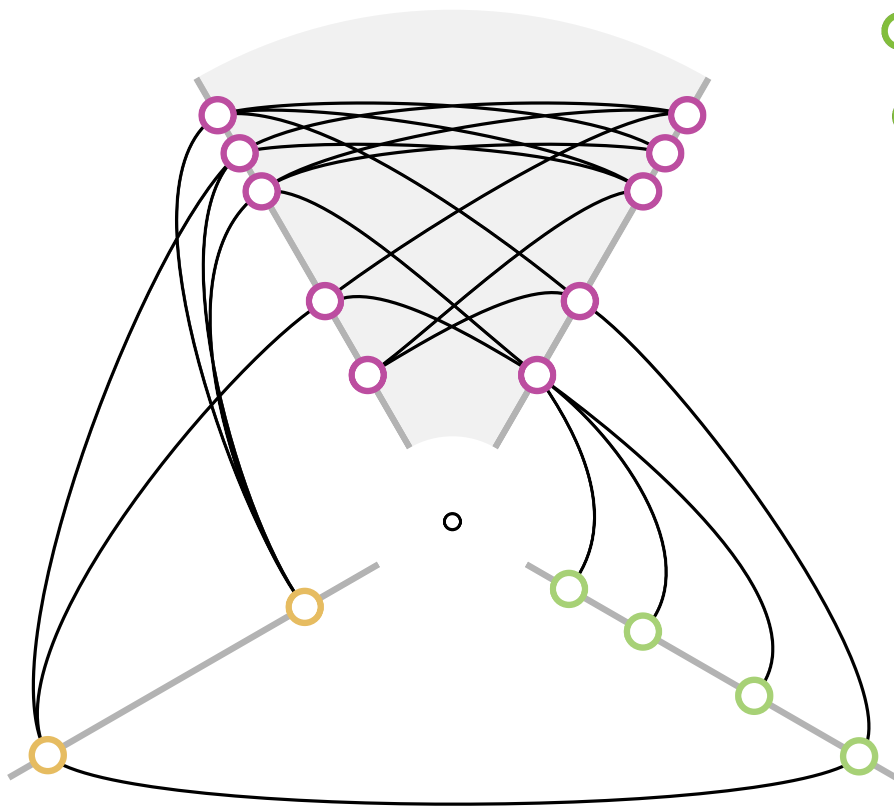
Introduce g **gaps**

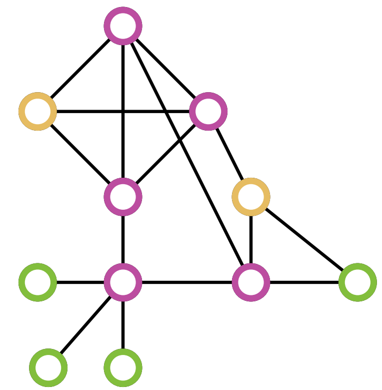
$g = 1$ $g = 2$ $g \geq 3$

* [Krzywinski et al., 2012]

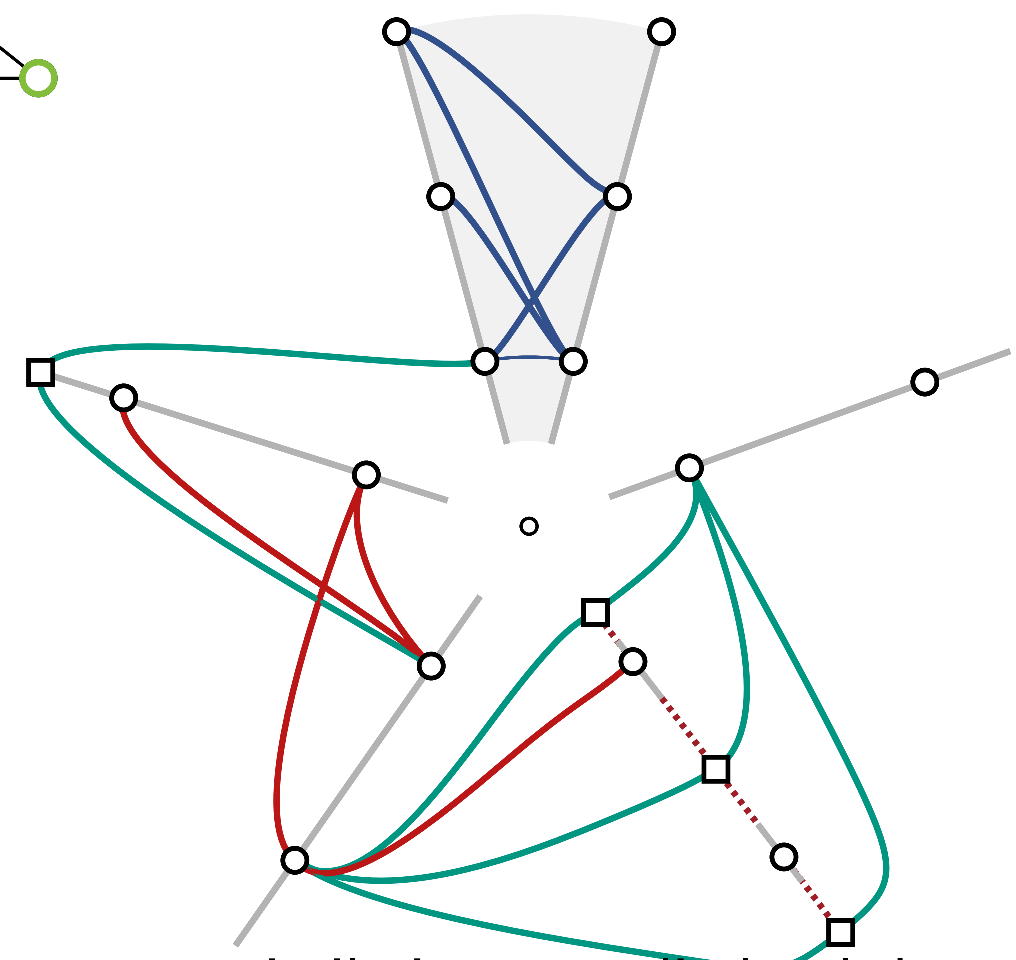
What is a Hive Plot?

Hive Plot*





Combinatorial Model



Note: some similarity to cyclic level drawings
[Bachmair et al., 2008, 2009, 2010]

* [Krzywinski et al., 2012]

Framework – Optimization Targets

Degree of Freedom	Idea and potential benefit of optimization
Vertex assignment	Vertices assigned to the same axis should represent dense subgraphs
# of axes	Show intra-axis edges on demand
	Focus on showing inter-axis edges The strength of weak ties [Granovetter, 1973]

Framework – Optimization Targets

Degree of Freedom	Idea and potential benefit of optimization
Vertex assignment	Vertices assigned to the same axis should represent dense subgraphs
# of axes	Show intra-axis edges on demand
	Focus on showing inter-axis edges The strength of weak ties [Granovetter, 1973]
Axis order	**Minimize total edge length**

Framework – Optimization Targets

Degree of Freedom	Idea and potential benefit of optimization
Vertex assignment	Vertices assigned to the same axis should represent dense subgraphs
# of axes	Show intra-axis edges on demand
	Focus on showing inter-axis edges The strength of weak ties [Granovetter, 1973]
Axis order	**Minimize total edge length**
Vertex position	Minimize total number of crossings
	Priority on minimizing inter-axis crossings

Framework – Optimization Targets

Degree of Freedom	Idea and potential benefit of optimization
Vertex assignment # of axes	Vertices assigned to the same axis should represent dense subgraphs Show intra-axis edges on demand **Focus on showing inter-axis edges** The strength of weak ties [Granovetter, 1973]
Axis order	**Minimize total edge length**
Vertex position	Minimize total number of crossings **Priority on minimizing inter-axis crossings**
# of gaps	Determine edge routing in the drawing Assume as input parameter

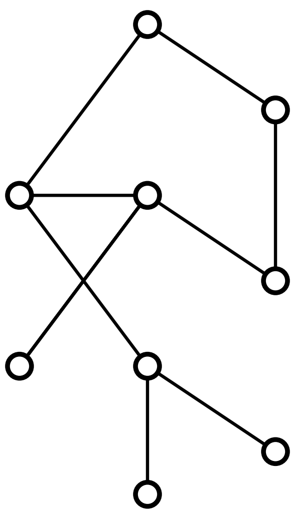
Introduction & Model

Framework

Evaluation

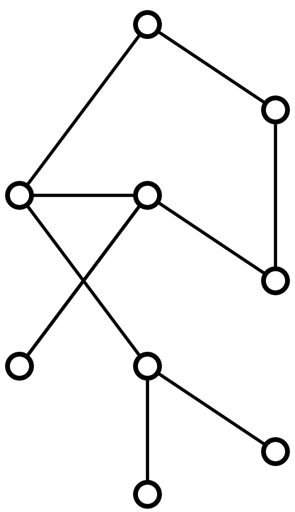
The Framework – Overview

Input



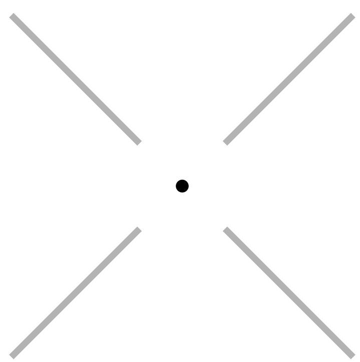
The Framework – Overview

Input



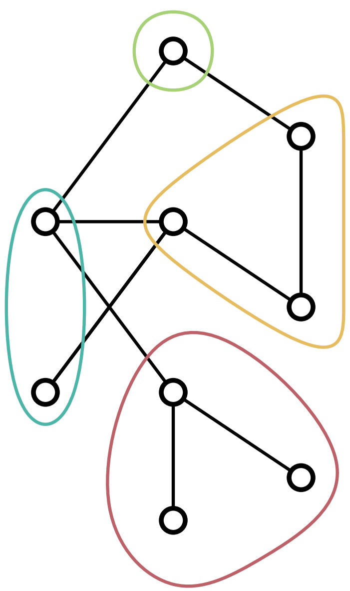
Step I

Vertex Partition



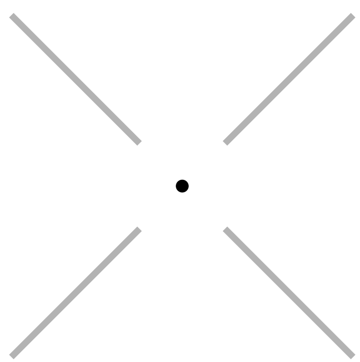
The Framework – Overview

Input



Step I

Vertex Partition



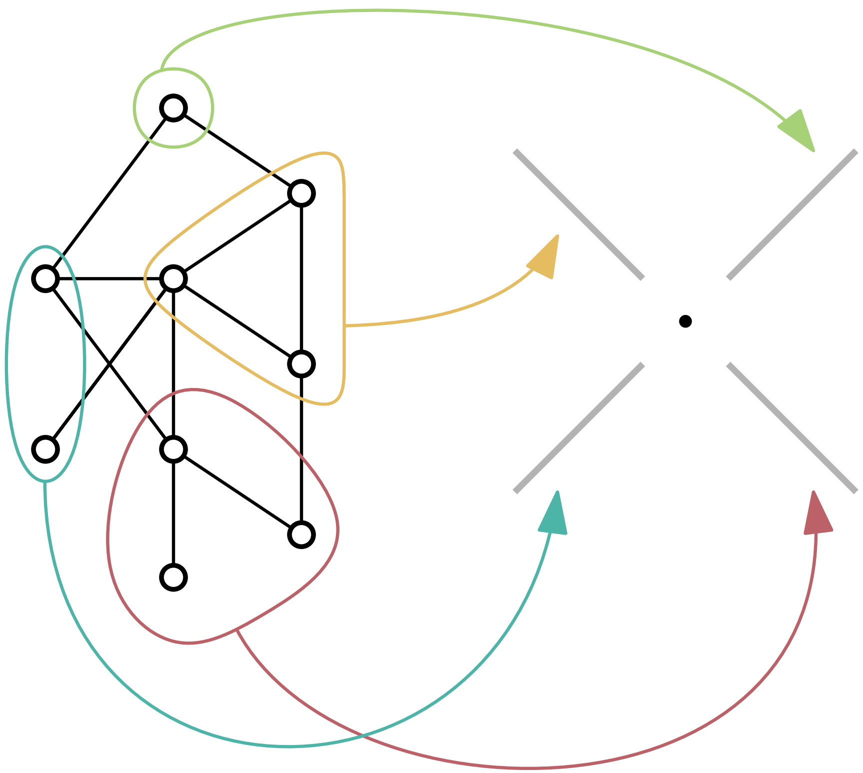
The Framework – Overview

Input

Step I

Vertex Partition

Modularity maximization **(NP-complete)** [Brandes et al., 2007]



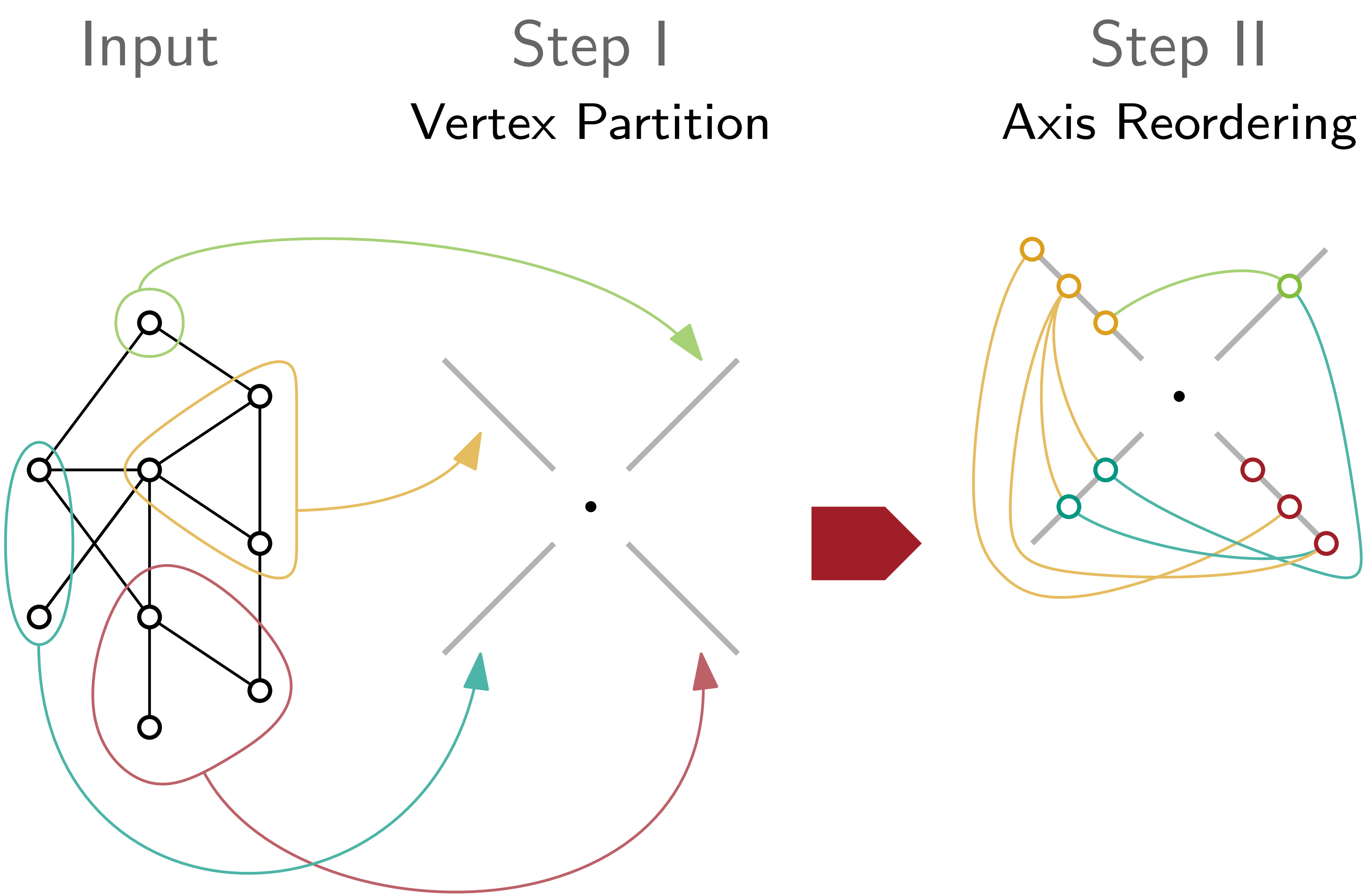
Three approaches:

1) Assume as input

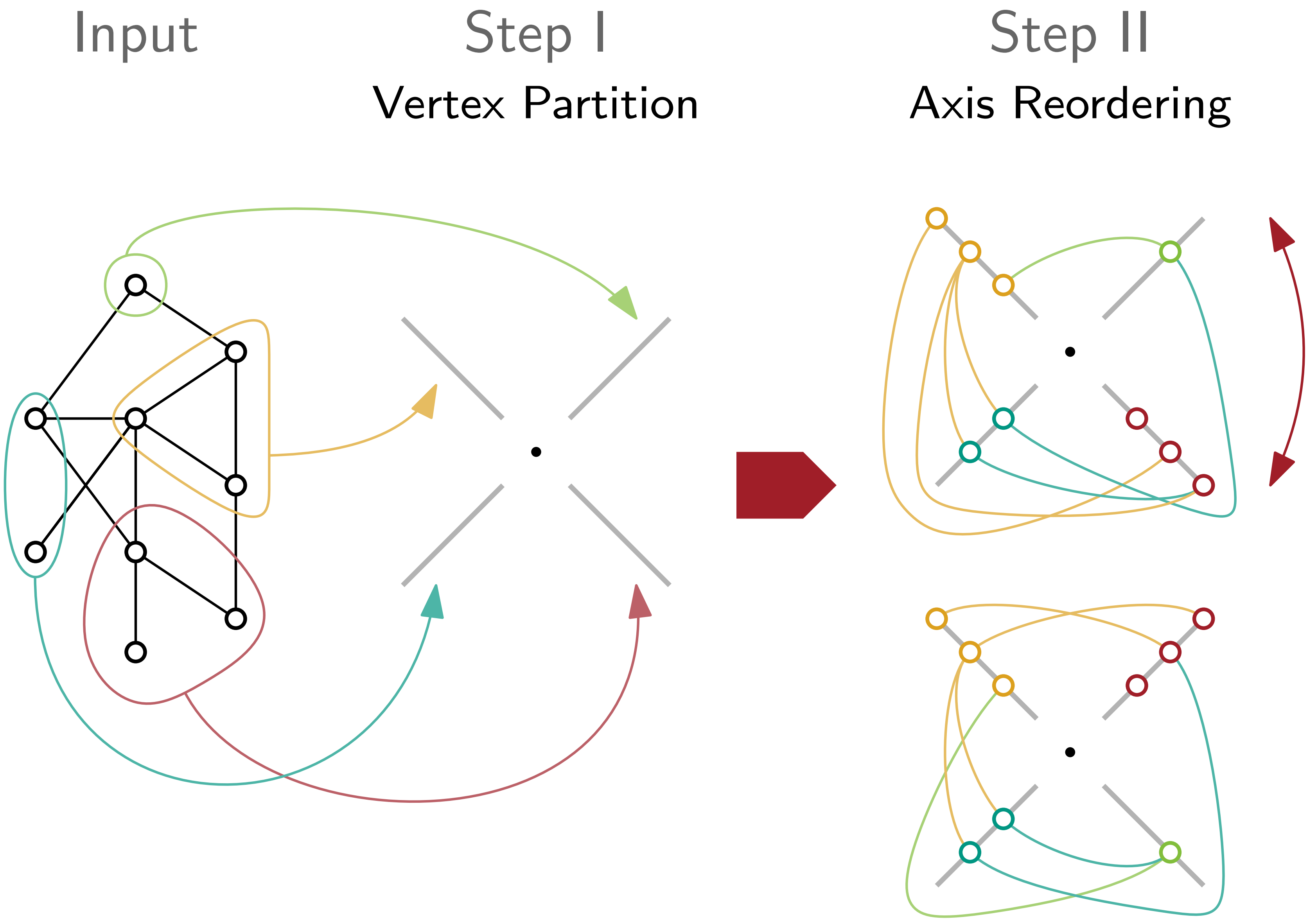
2) Clustering for fixed k [Clauset et al., 2004]

3) Let community detection algorithm decide k [Blondel et al., 2008]

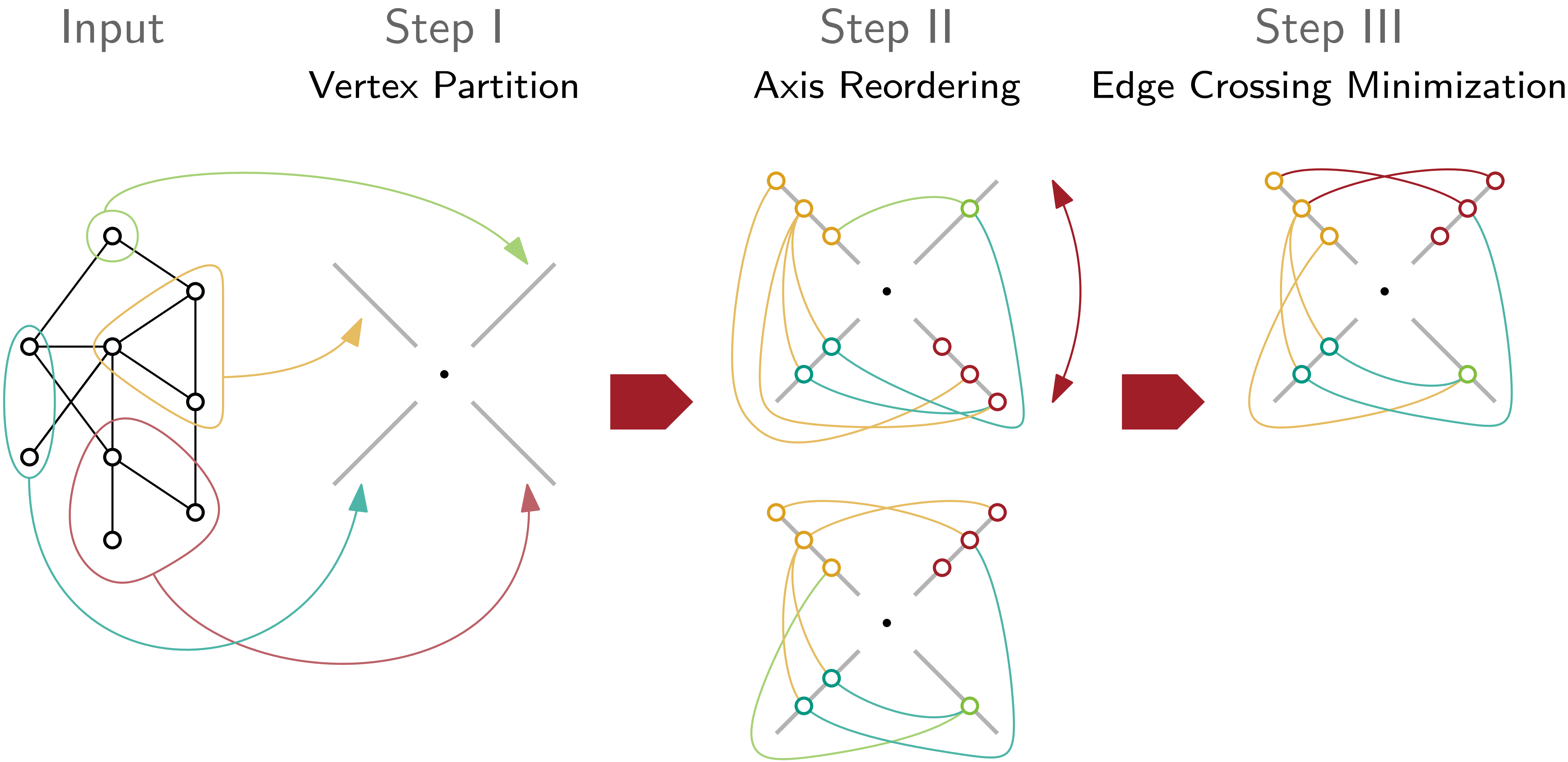
The Framework – Overview



The Framework – Overview

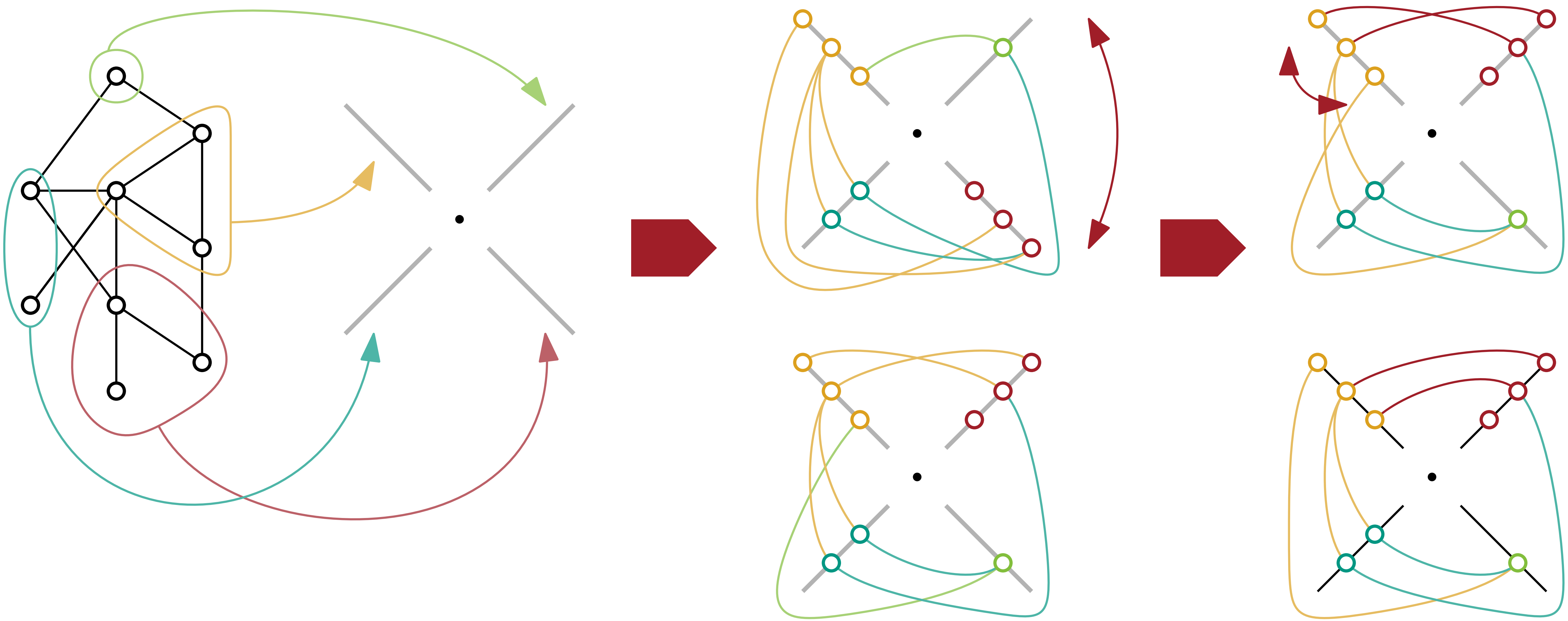


The Framework – Overview



The Framework – Overview

Input	Step I	Step II	Step III
	Vertex Partition	Axis Reordering	Edge Crossing Minimization

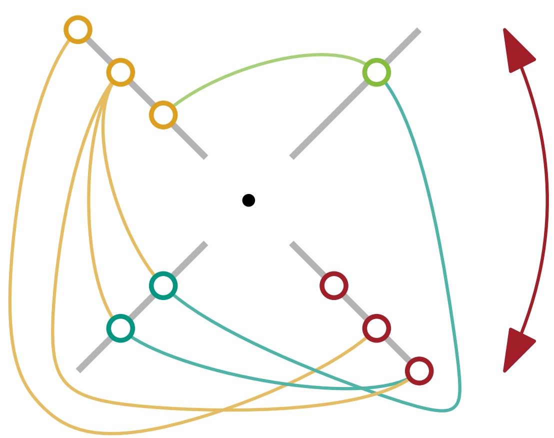


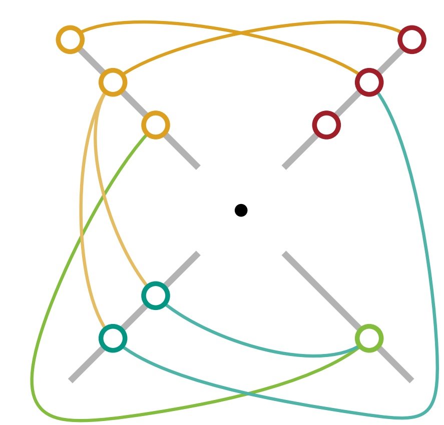
Step II – Axis Order

Order axis such that the **total span** is minimized

$$cost(\phi) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_{ij} \; span(a_i, a_j)$$

$w_{ij} \ldots$ number of edges between V_i and V_j



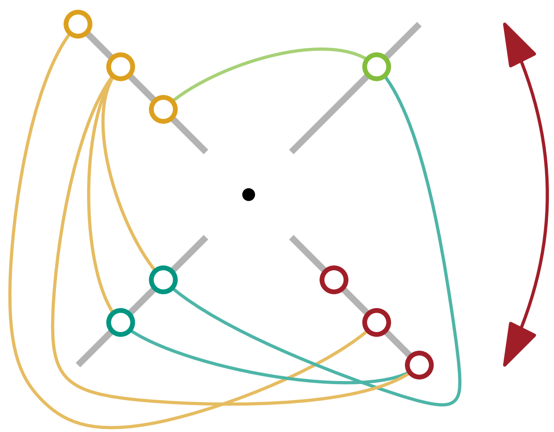


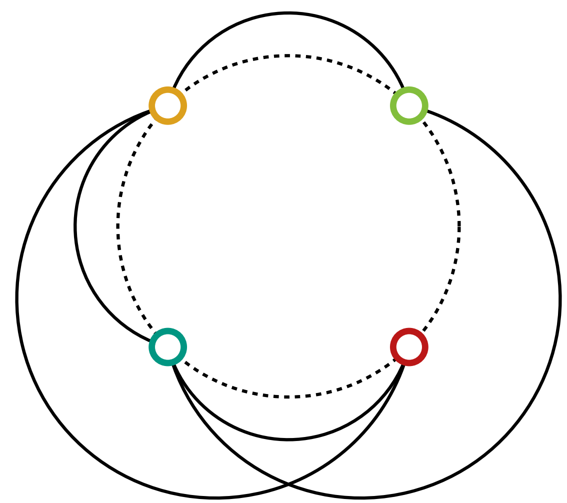
Step II – Axis Order

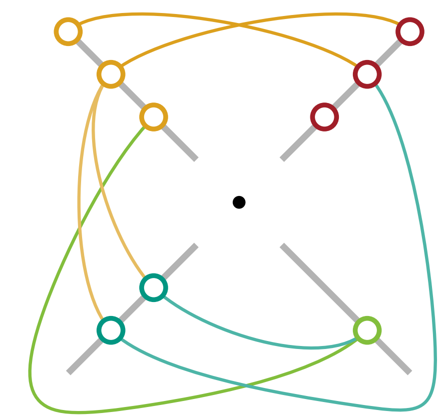
Order axis such that the **total span** is minimized

$$cost(\phi) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_{ij} \; span(a_i, a_j)$$

w_{ij} ... number of edges between V_i and V_j







Circular arrangement problem **(NP-complete)**

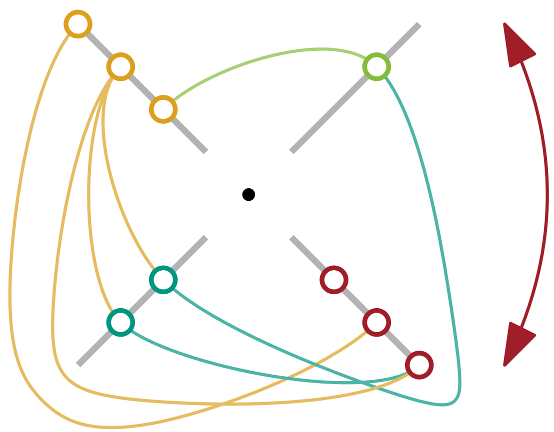
[Ghanapaty and Lhoda, 2004]

Step II – Axis Order

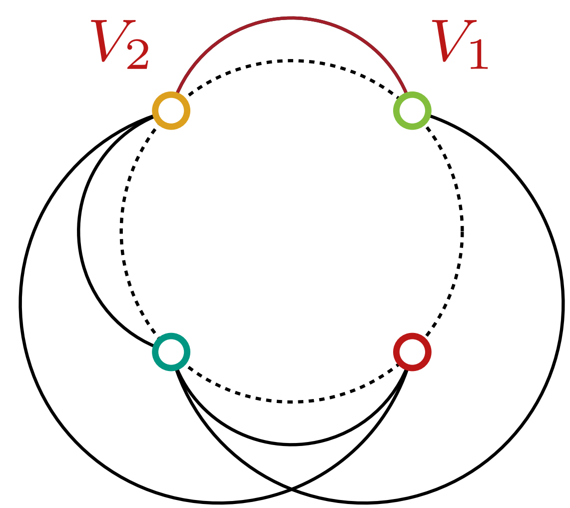
Order axis such that the **total span** is minimized

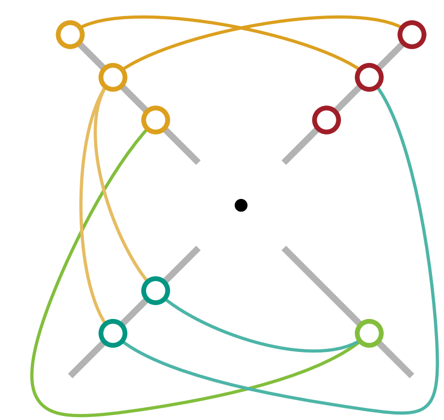
$$cost(\phi) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_{ij} \; span(a_i, a_j)$$

w_{ij} . . . number of edges between V_i and V_j



$w_{12} = 1, span(a_1, a_2) = 1$





Circular arrangement problem (NP-**complete**)

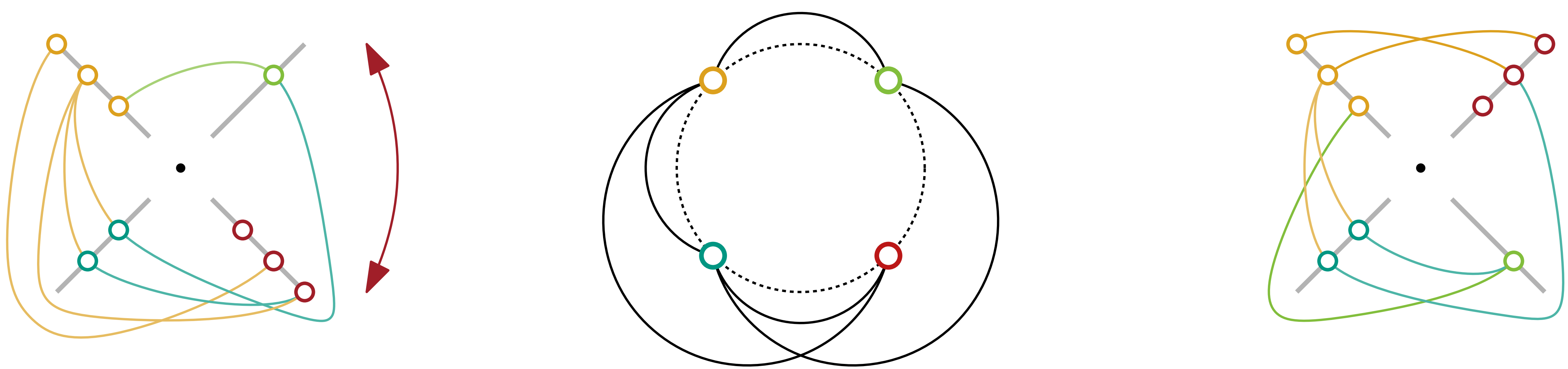
[Ghanapaty and Lhoda, 2004]

Step II – Axis Order

Order axis such that the **total span** is minimized

$$cost(\phi) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} w_{ij}\ span(a_i, a_j)$$

w_{ij} . . . number of edges between V_i and V_j



Circular arrangement problem (NP-**complete**)

[Ghanapaty and Lhoda, 2004]

- Brute-force if $k < 9$
- Simulated annealing otherwise

Step III – Crossing Minimization (Phase I)

Multilayer crossing minimization (NP-**hard**)
[Eades and Whitesides, 1994]

Two phase approach:

(I) Minimize inter-axis crossings

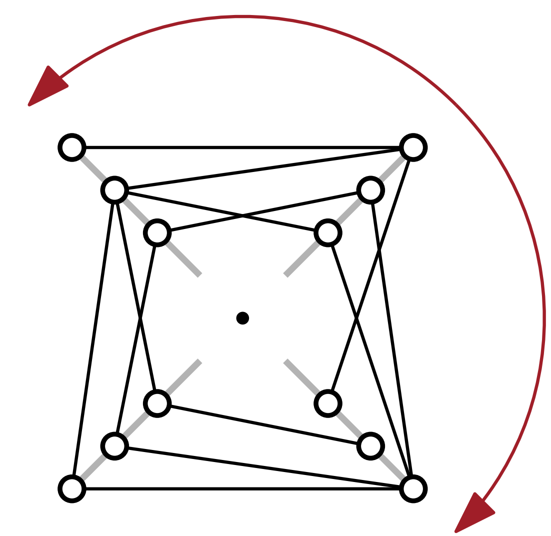
(II) Minimize intra-axis crossings

Step III – Crossing Minimization (Phase I)

Two phase approach:

- **(I) Minimize inter-axis crossings**
- (II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**
[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Step III – Crossing Minimization (Phase I)

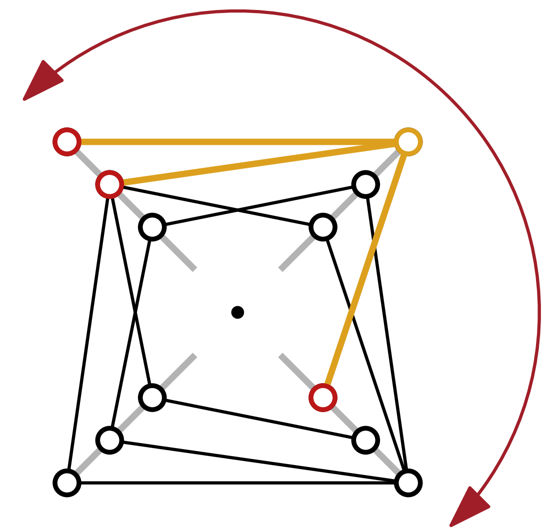
Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**

[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

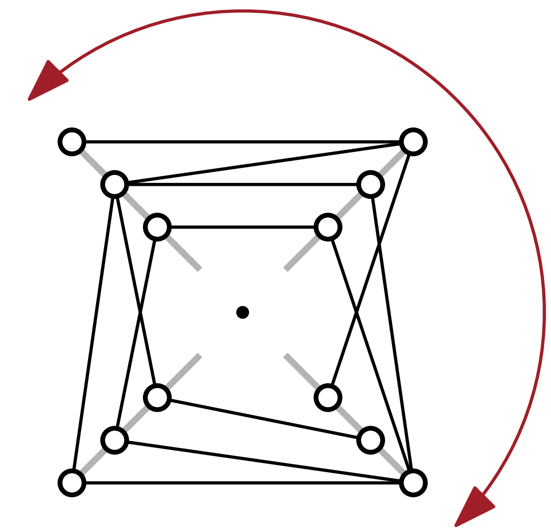
Step III – Crossing Minimization (Phase I)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization (NP-**hard**)
[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

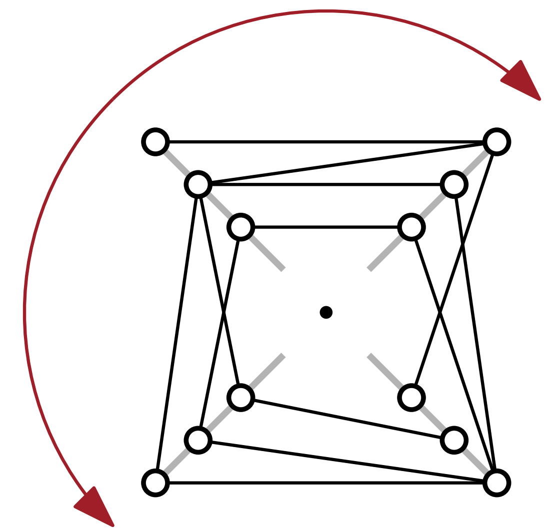
Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**

[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

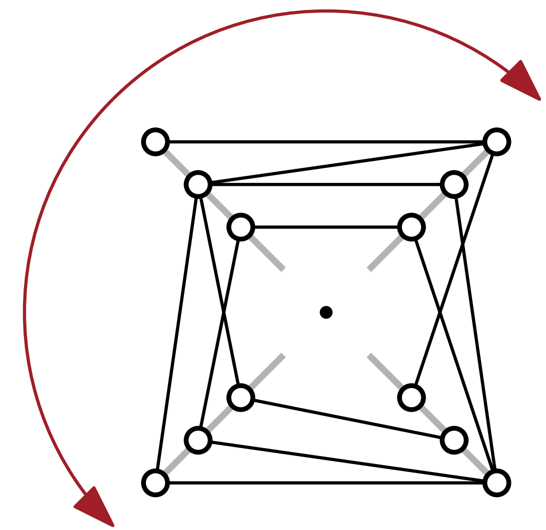
Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**

[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

Two phase approach:

(I) Minimize inter-axis crossings

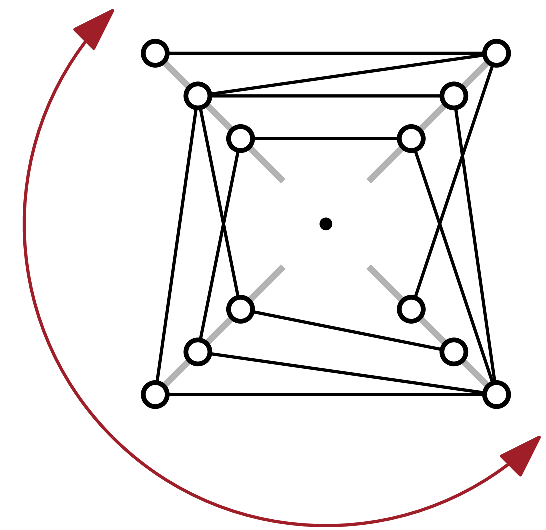
(II) Minimize intra-axis crossings

Multilayer crossing minimization (NP-**hard**)

[Eades and Whitesides, 1994]

Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$



Vertex order initialized random

Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

Two phase approach:

(I) Minimize inter-axis crossings

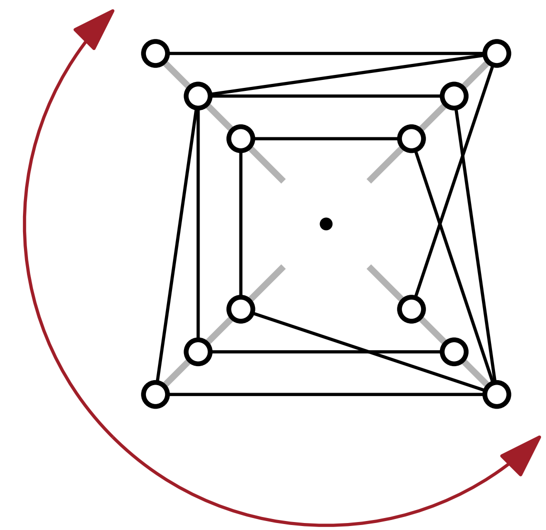
(II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**

[Eades and Whitesides, 1994]

Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$



Vertex order initialized random

Perform **layer-by-layer sweep**

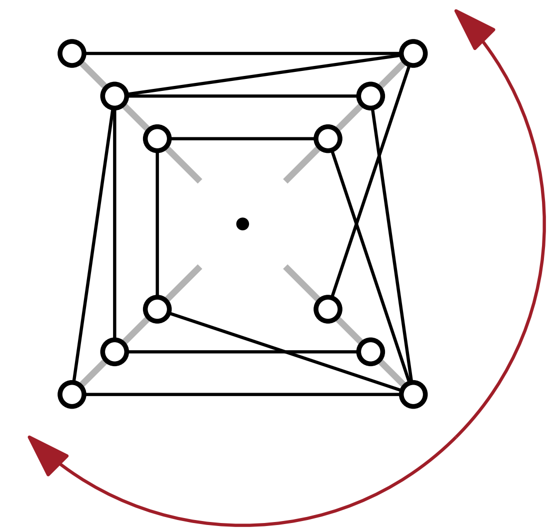
Step III – Crossing Minimization (Phase I)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization (NP-**hard**)
[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

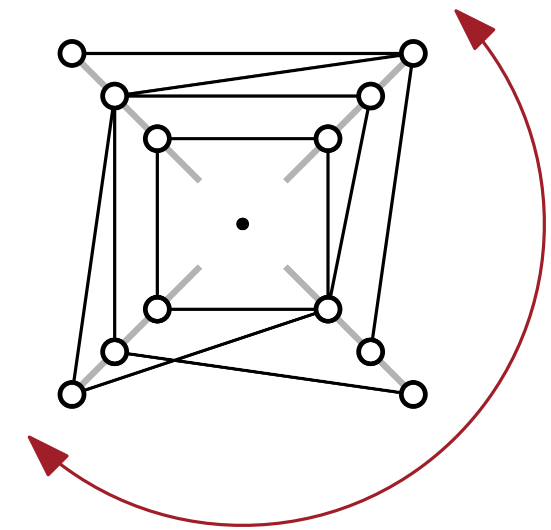
Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Multilayer crossing minimization **(NP-hard)**

[Eades and Whitesides, 1994]



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

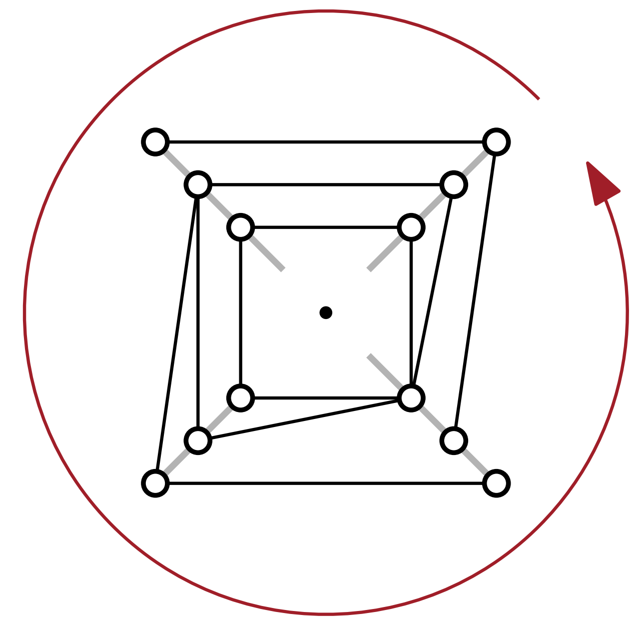
Perform **layer-by-layer sweep**

Step III – Crossing Minimization (Phase I)

Multilayer crossing minimization **(NP-hard)**
[Eades and Whitesides, 1994]

Two phase approach:

- **(I) Minimize inter-axis crossings**
- (II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Vertex order initialized random

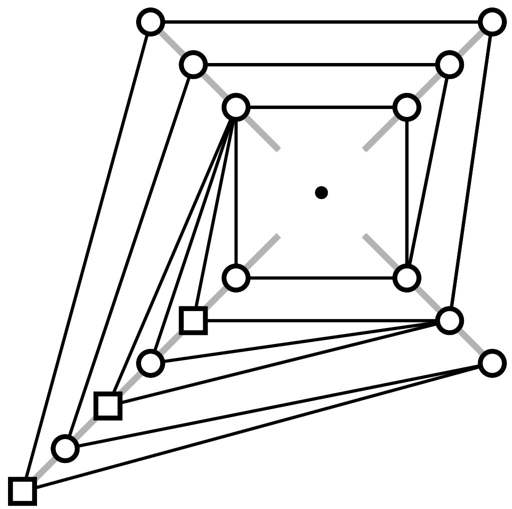
Perform **layer-by-layer sweep**

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

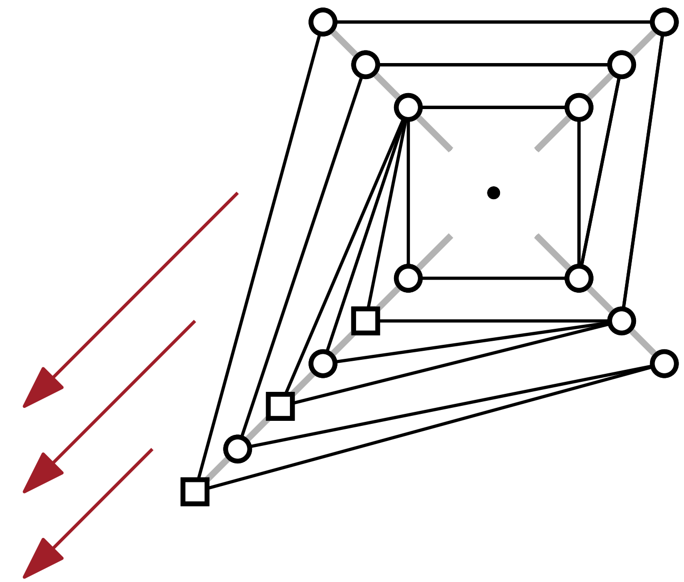
Gaps require **additional step** after computing new positions

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Gaps require **additional step** after computing new positions

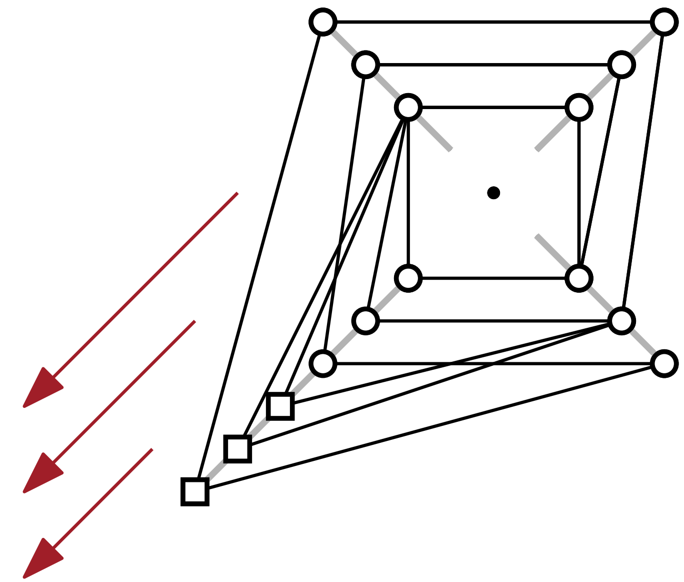
$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Gaps require **additional step** after computing new positions

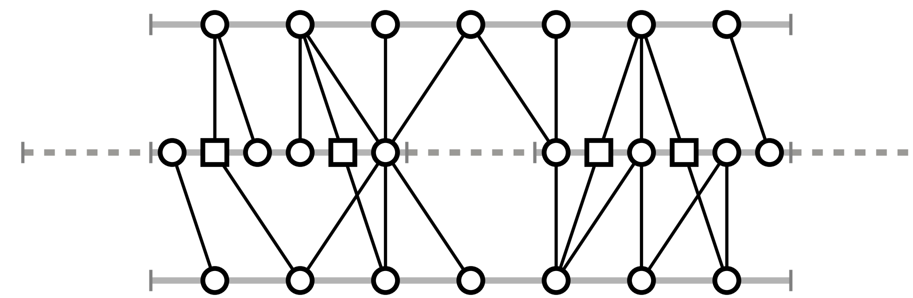
$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Gaps require **additional step** after computing new positions

$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

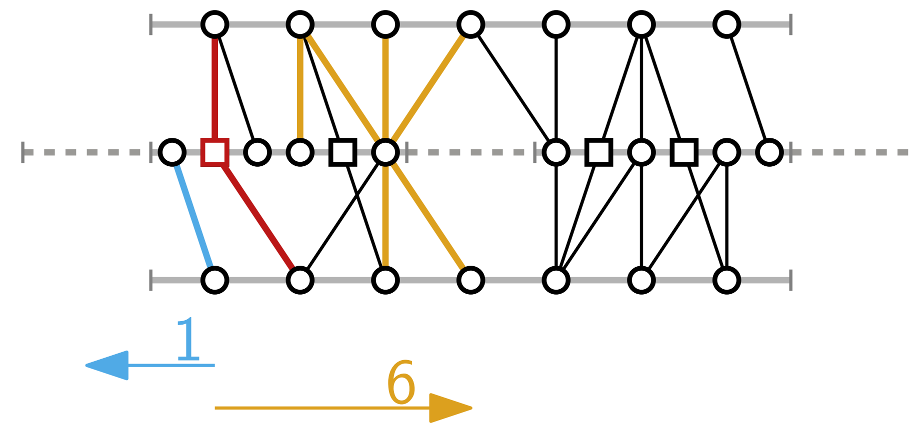
$g \geq 2$ Move dummy vertices **greedily by # crossings** to gap left or right

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Gaps require **additional step** after computing new positions

$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

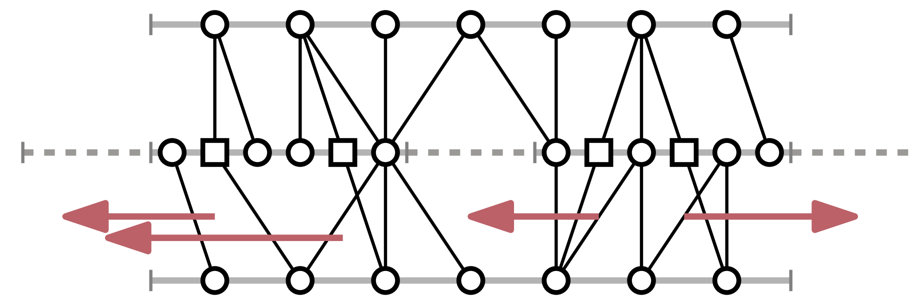
$g \geq 2$ Move dummy vertices **greedily by # crossings** to gap left or right

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

Gaps require **additional step** after computing new positions

$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

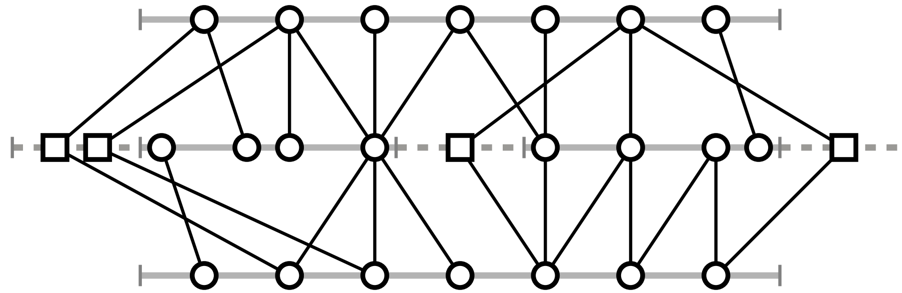
$g \geq 2$ Move dummy vertices **greedily by # crossings** to gap left or right

Crossing Minimization – Phase I (cont.)

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings



Adapted **barycenter heuristic**

$$pos(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{\pi_{\alpha(v)}(v)}{|\pi_{\alpha(v)}|}$$

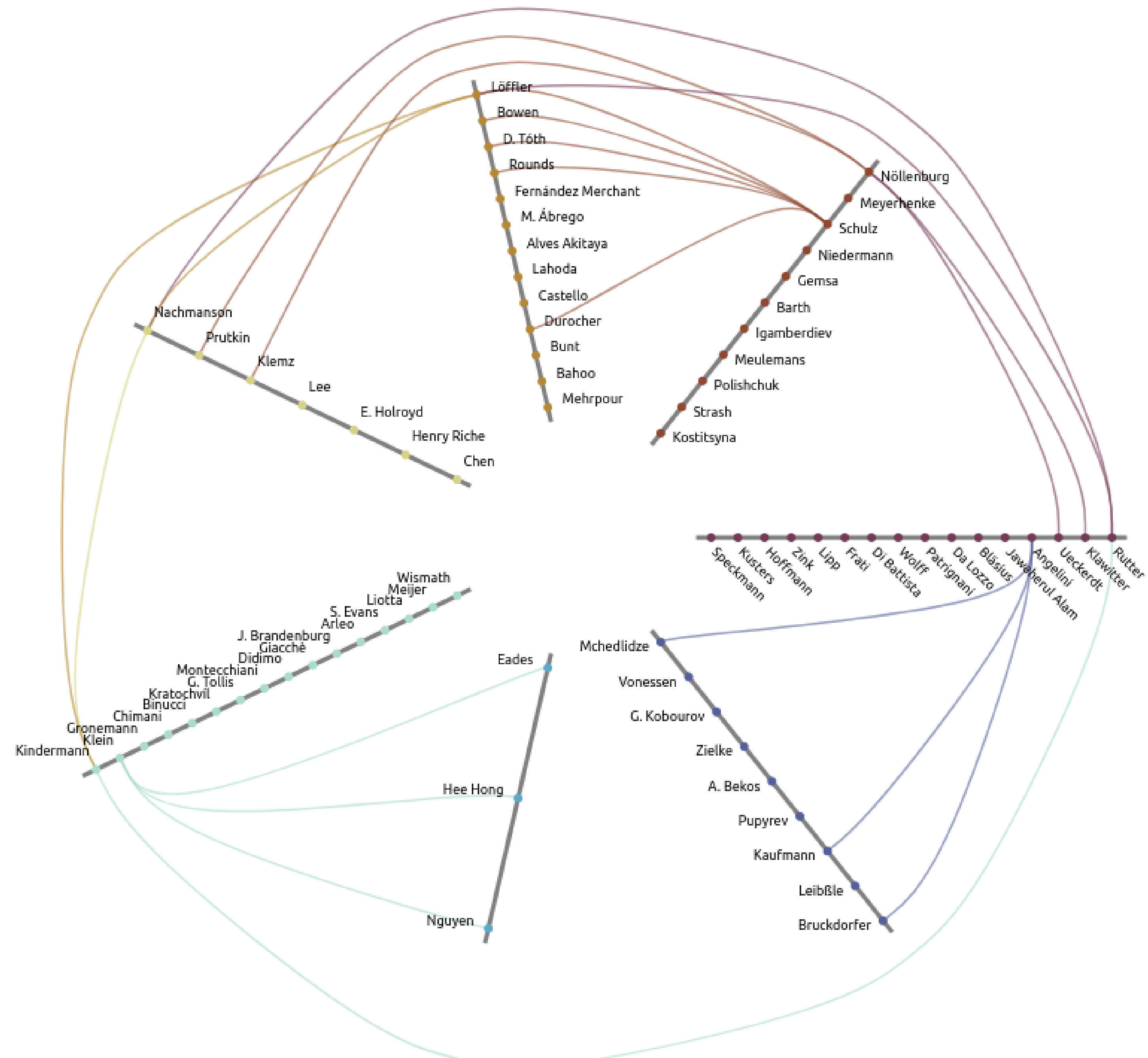
Gaps require **additional step** after computing new positions

$g = 1$ Move dummy vertices to the end of order while keeping **relative** order

$g \geq 2$ Move dummy vertices **greedily by # crossings** to gap left or right

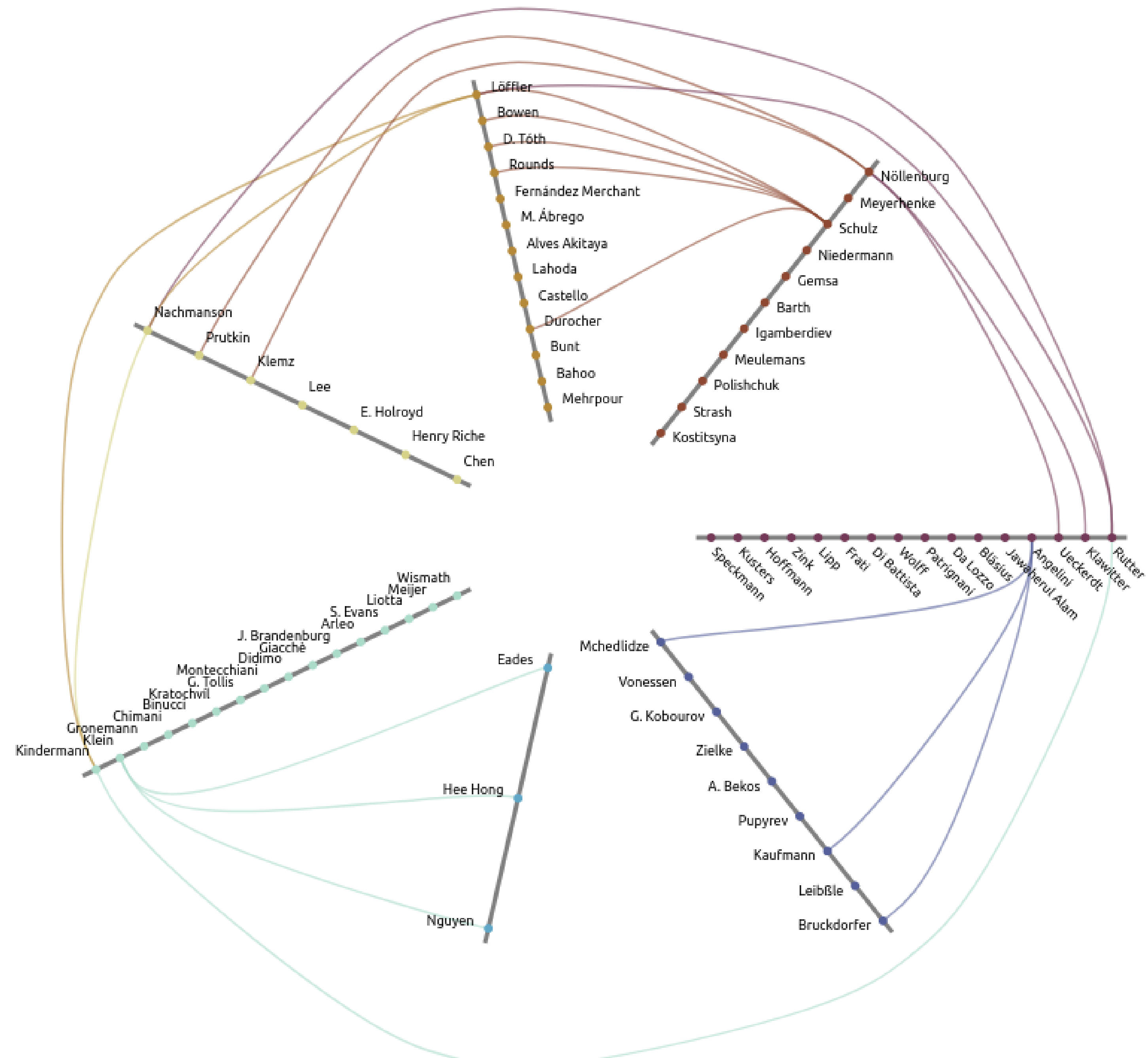
Prototype Application

- Python + D3 web application



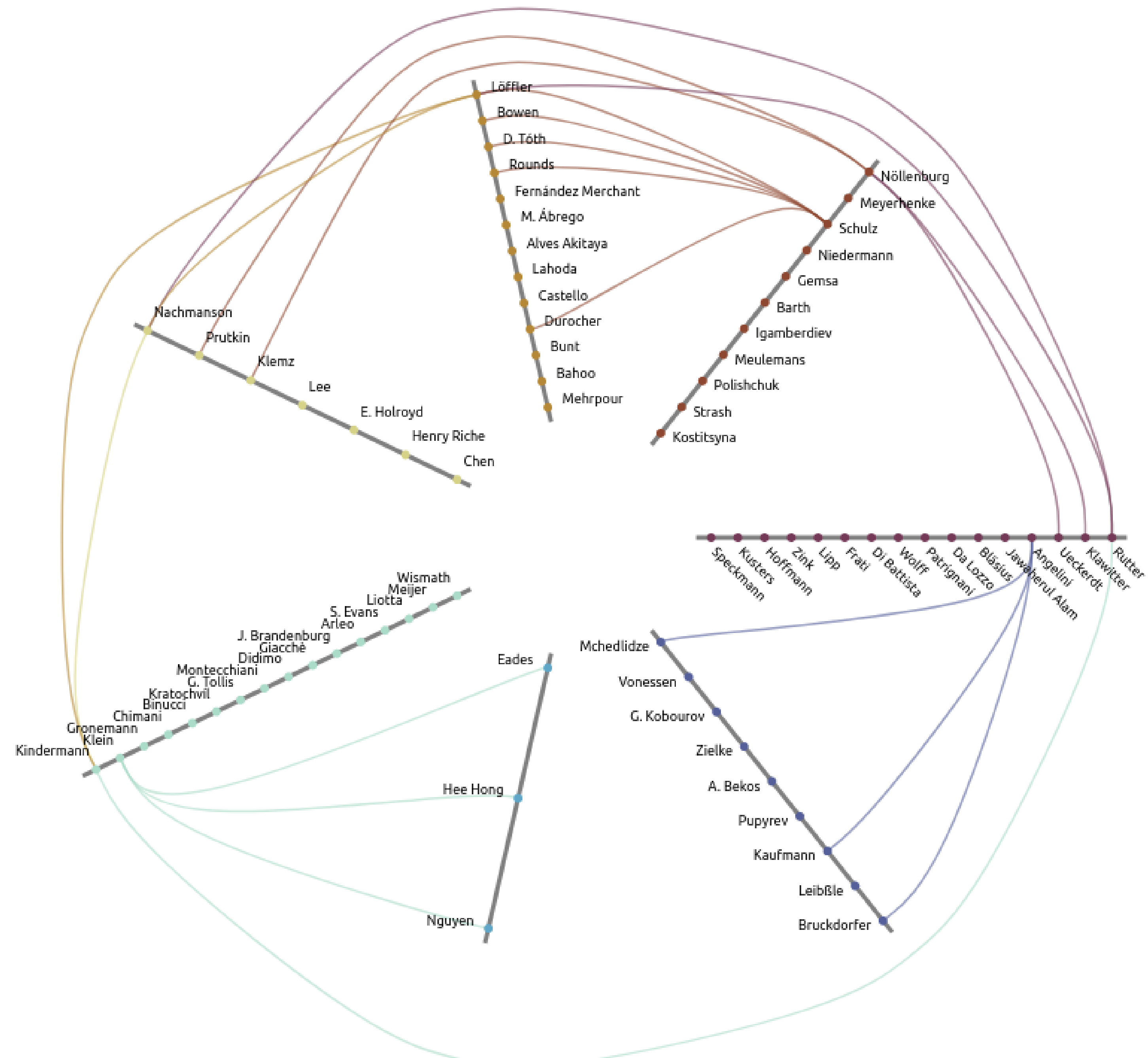
Prototype Application

- Python + D3 web application
- Initially hides intra-axis edges



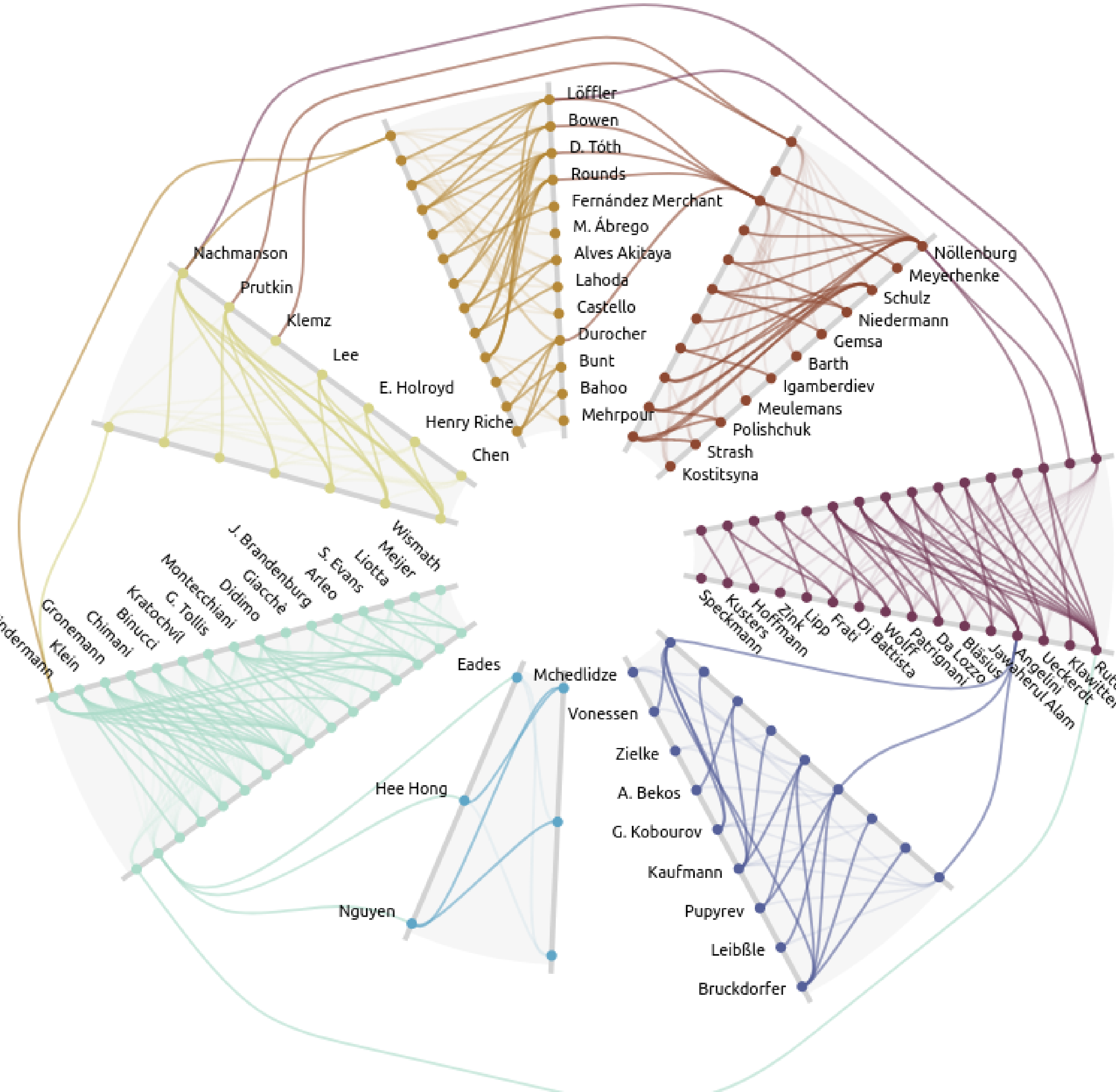
Prototype Application

- Python + D3 web application
- Initially hides intra-axis edges
- Circular color map



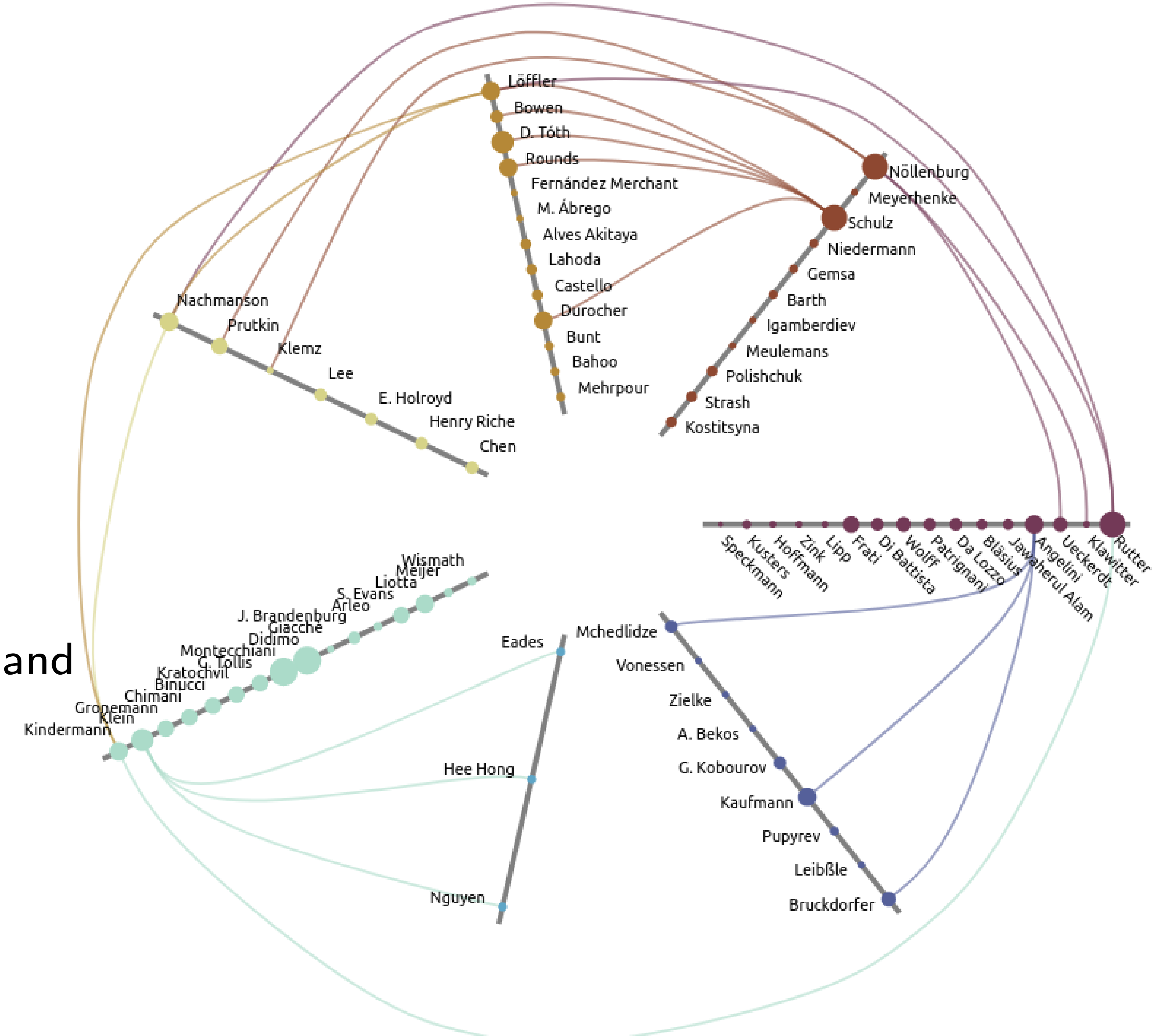
Prototype Application

- Python + D3 web application
- Initially hides intra-axis edges
- Circular color map
- Interactively expand axes on demand



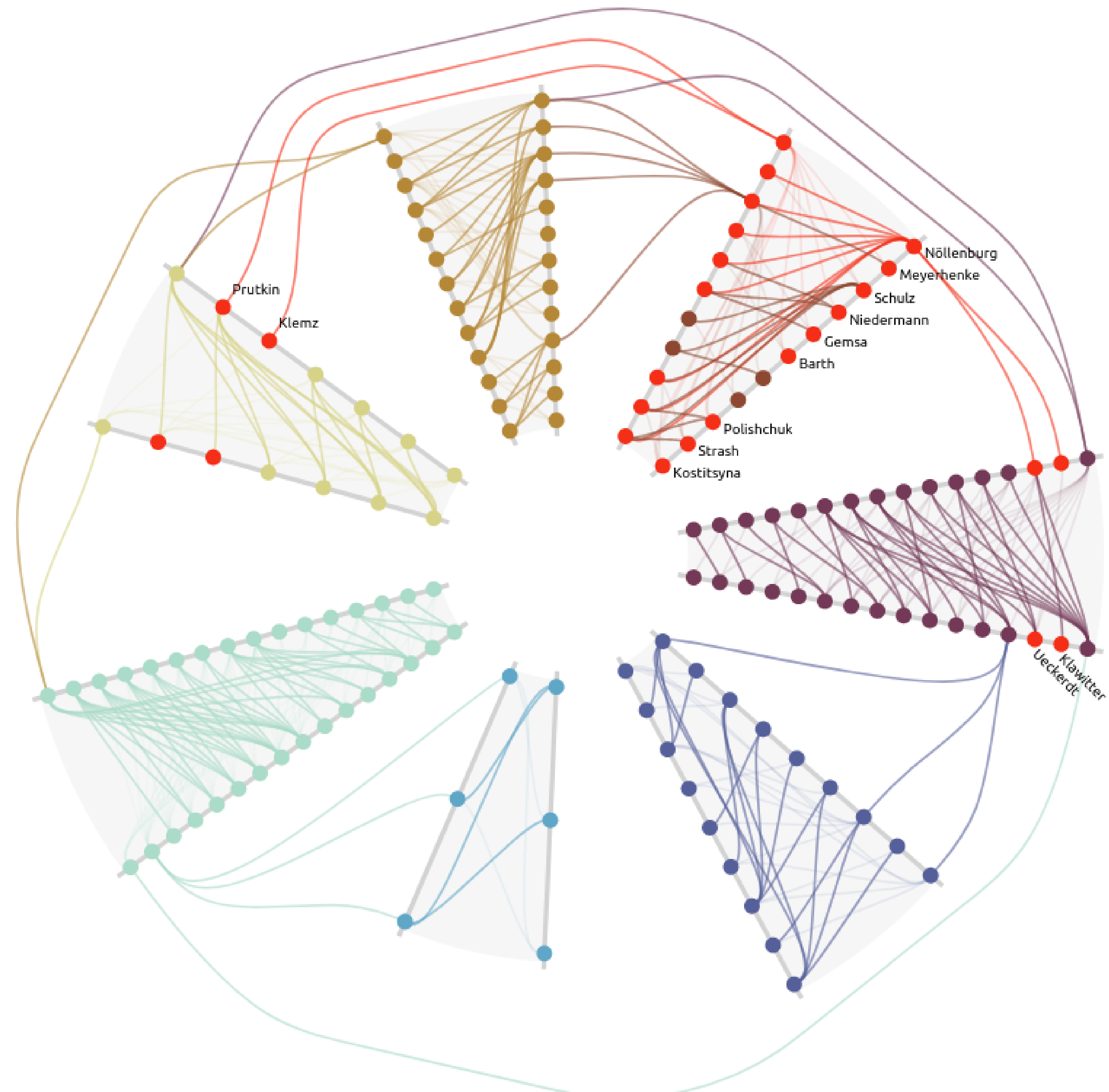
Prototype Application

- Python + D3 web application
- Initially hides intra-axis edges
- Circular color map
- Interactively expand axes on demand
- Scale vertices by intra-cluster connectivity



Prototype Application

- Python + D3 web application
- Initially hides intra-axis edges
- Circular color map
- Interactively expand axes on demand
- Scale vertices by intra-cluster connectivity
- Highlight vertex and neighborhood



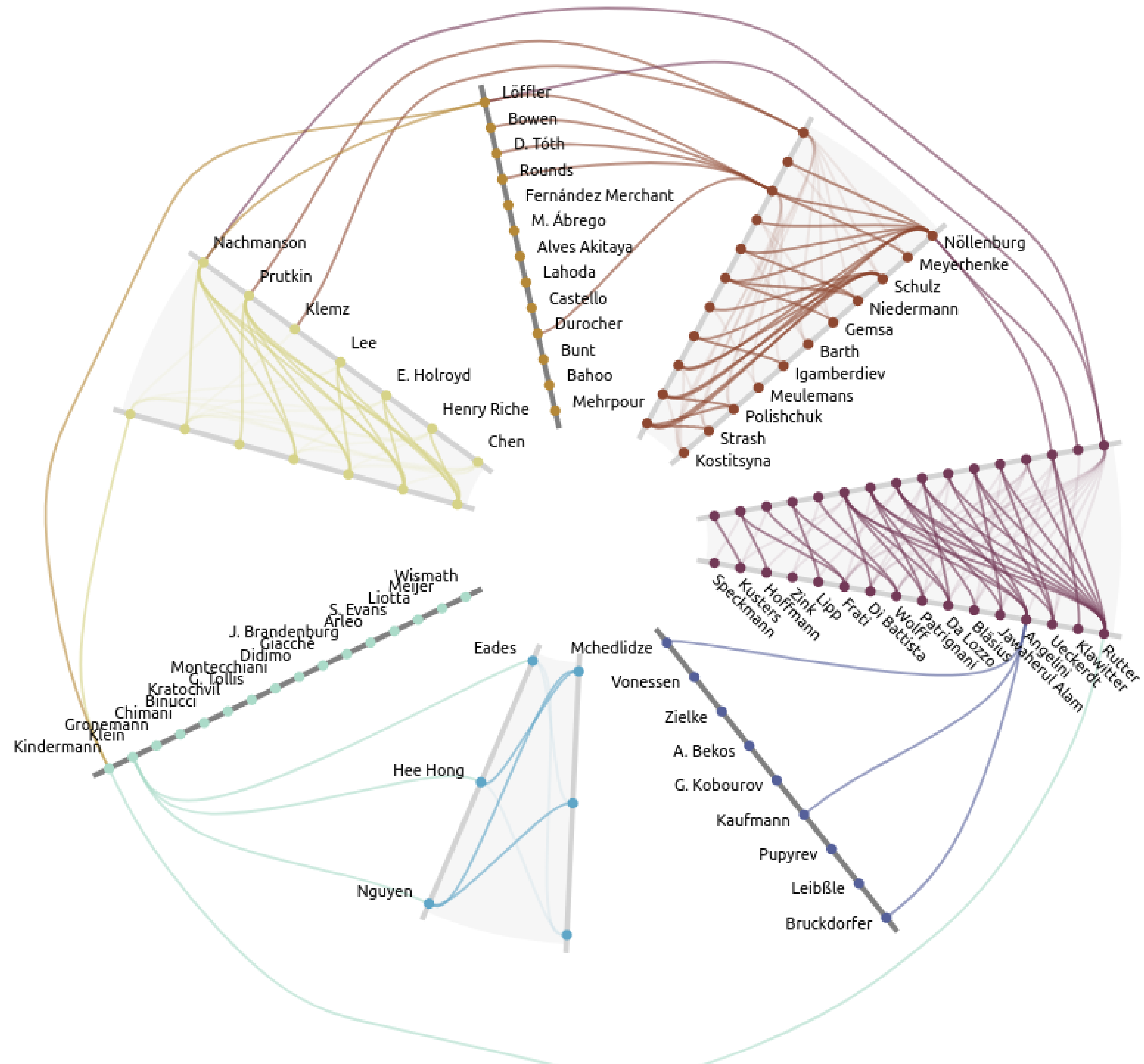
Introduction & Model

Framework

Evaluation

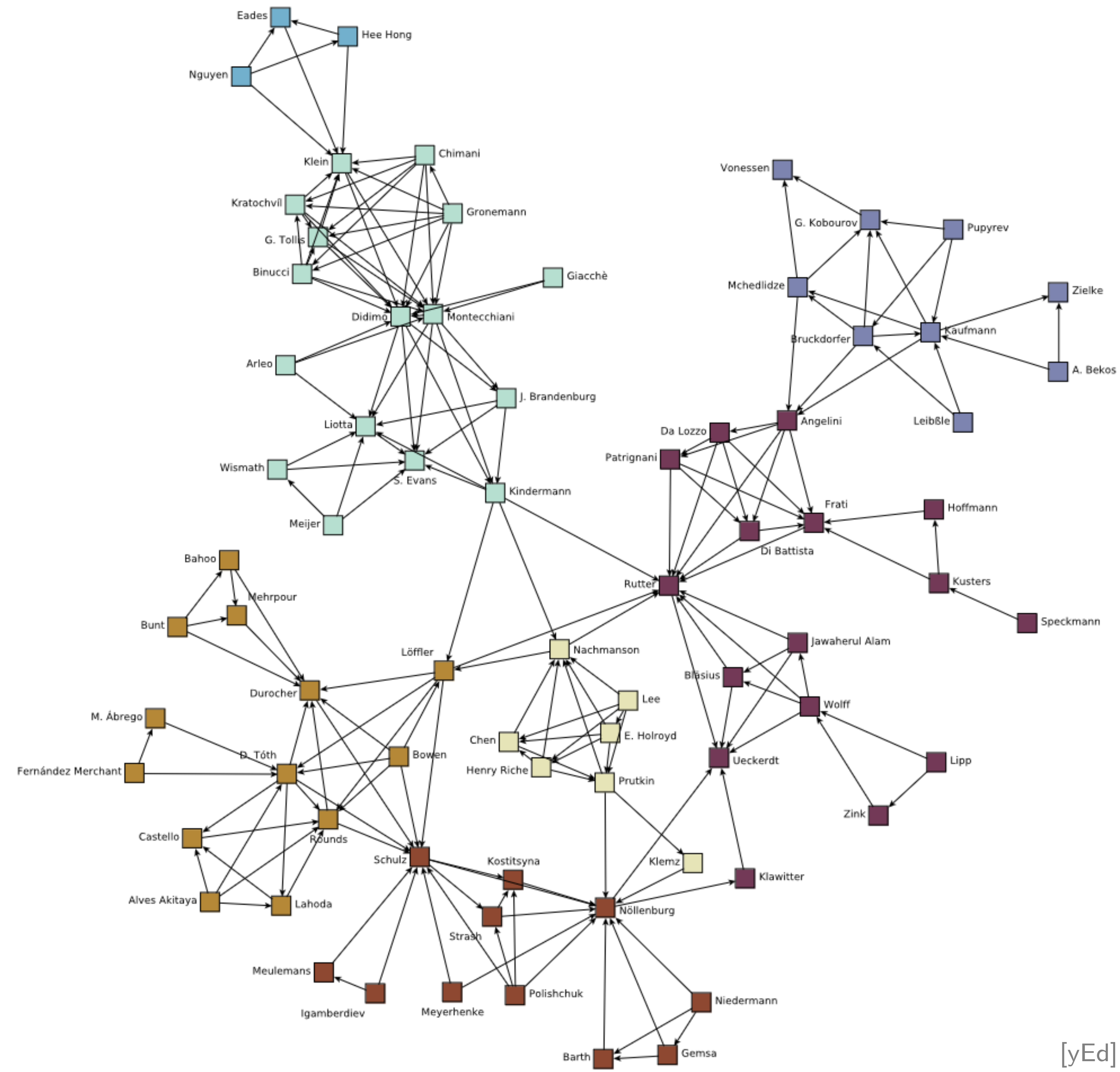
Case Study – Hive Plot

- Coauthorship network, GD 2014 ($|V| = 75, |E| = 190$)
- Cluster mainly by geographic proximity
- Clusters can be perceived without further encoding
- Predictable edge routing
- Balanced space utilization



Case Study – Force-Based

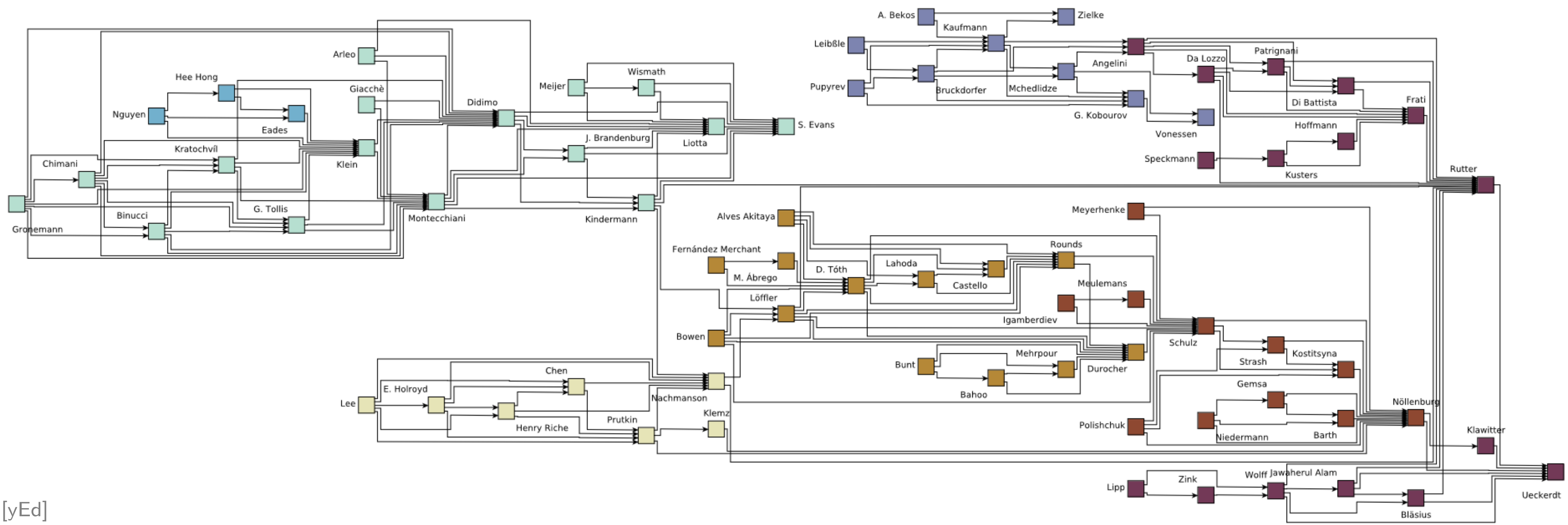
- Clusters appear naturally
- Clique structure clearer
- Macro structure less clear
- Edge-vertex overlaps
- Color necessary to communicate clusters
- Less uniform and predictable



[yEd]

Case Study – Hierarchical

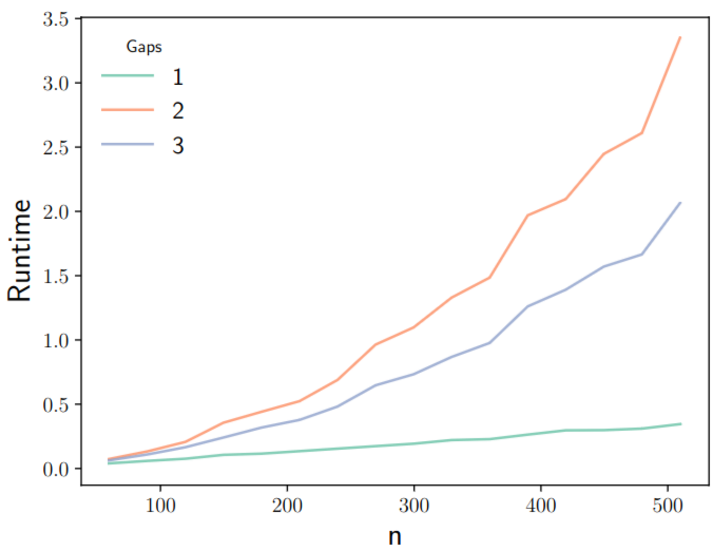
- Emphasizes imposed hierarchy
- Following edges becomes progressively more difficult
- Aspect ratio less balanced

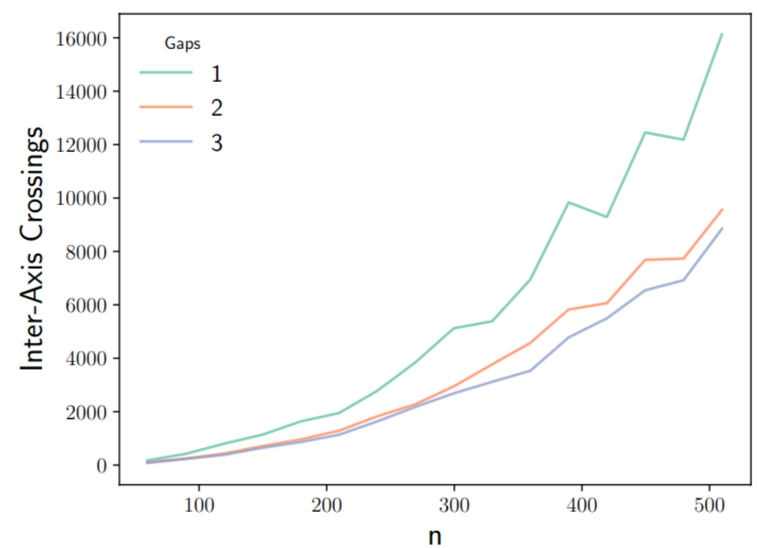


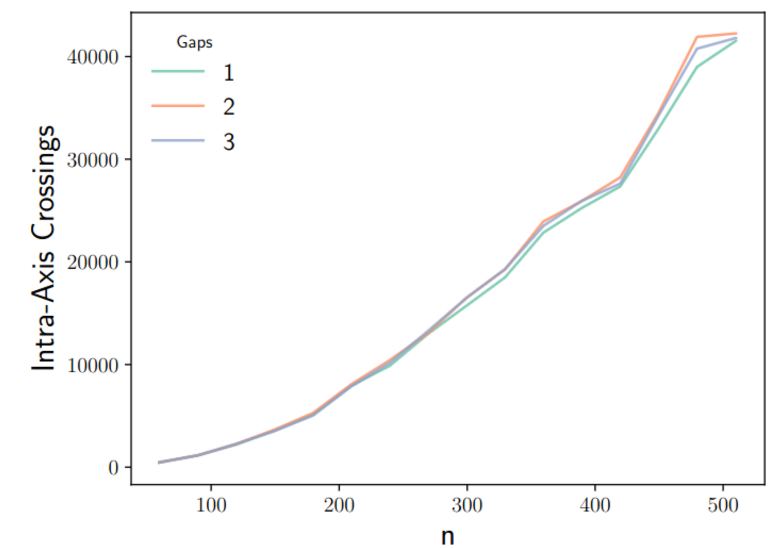
[yEd]

Computational Experiment

- Synthetic Dataset of graphs with 60 to 510 vertices
- $k = 6$ fixed
- Varied gaps $g = \{1, 2, 3\}$
- Measured runtime and counted crossings

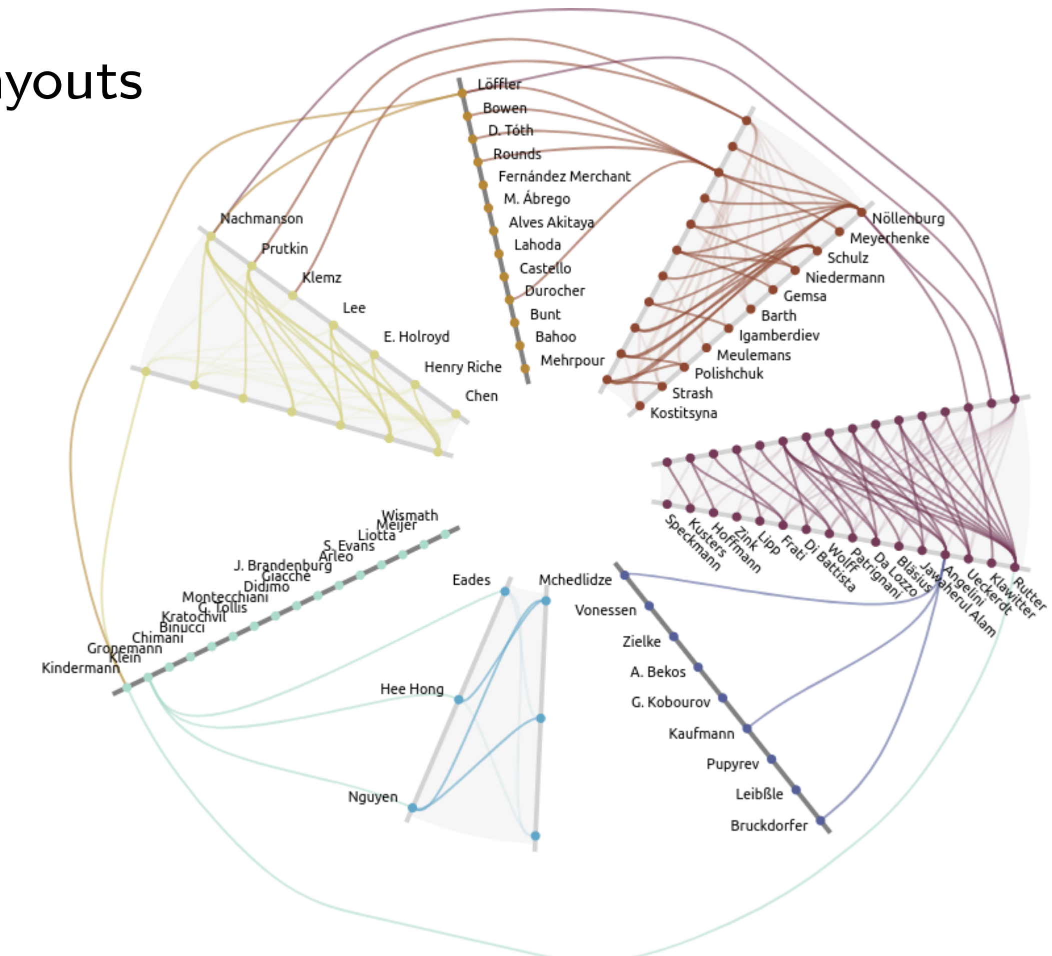






Conclusion

- Combinatorial framework to compute hive plot layouts
- Heuristic for crossing minimization
- Case study indicates possible advantages

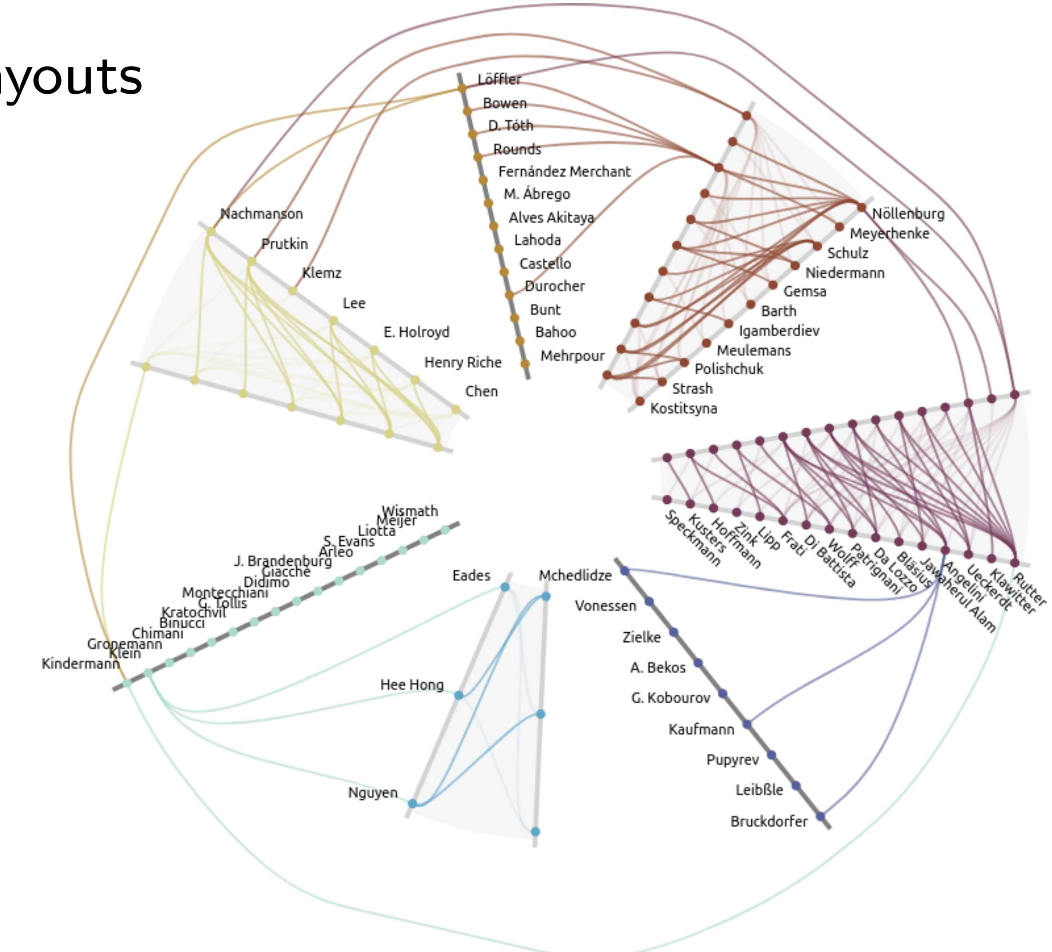


Code and prototype available:

https://osf.io/6zqx9/

Conclusion

- Combinatorial framework to compute hive plot layouts
- Heuristic for crossing minimization
- Case study indicates possible advantages



Open Problems

- Explore usability
- Optimal models & comparison
- Choice of algorithms
- Visual scalability

Code and prototype available:
https://osf.io/6zqx9/

Vertex Partition

Input: $G = (V, E)$

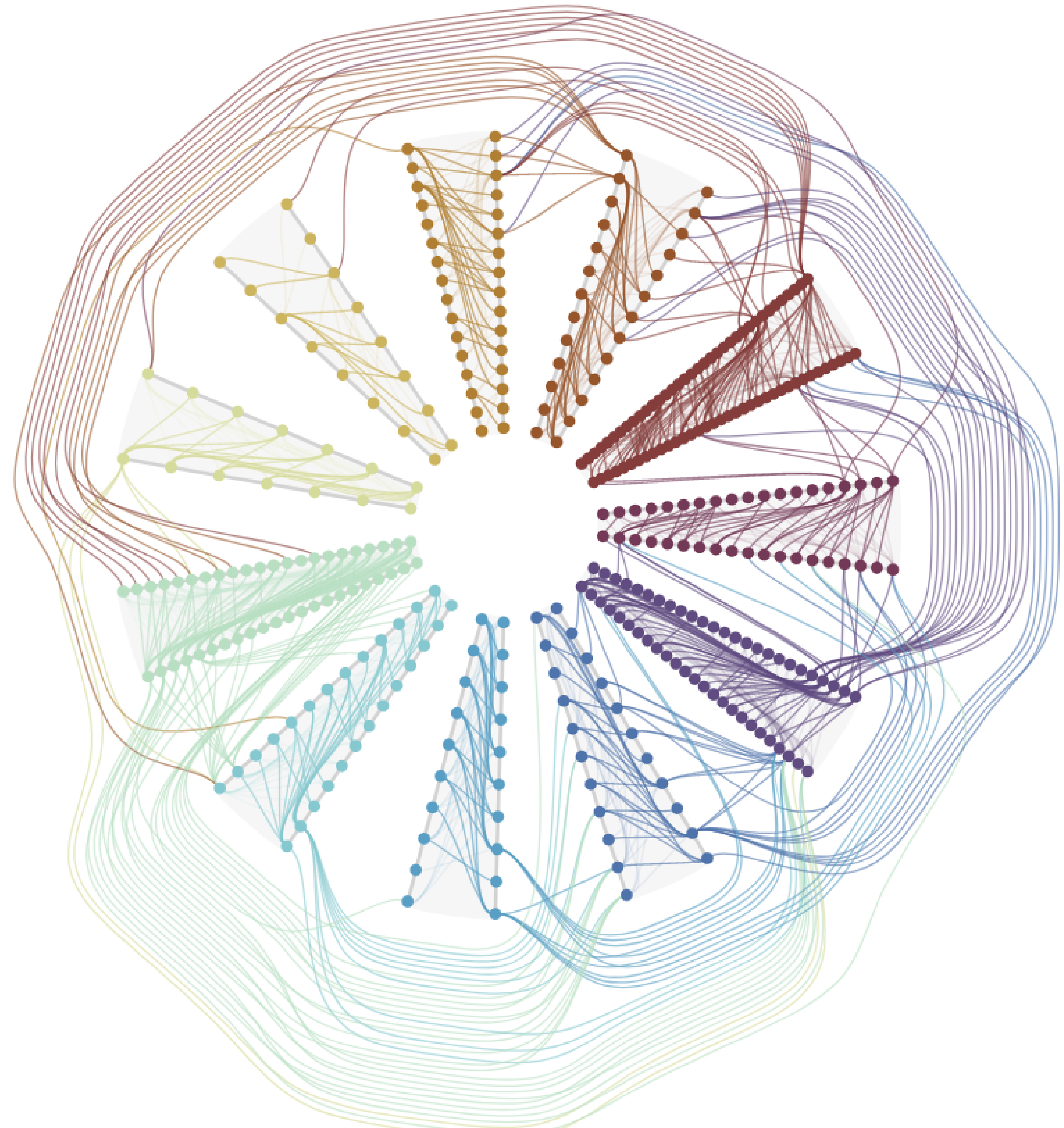
Goal: Partition that represent dense induced subgraphs

Additional input: $\{V_0, \ldots, V_{k-1}\}$	**Additional input:** k	**Additional input:** –
Algorithm: –	**Algorithm:** Greedy modularity maximization	**Algorithm:** Louvain community detection
Output: –	**Output:** $\{V_0, \ldots, V_{k-1}\}$	**Output:** k, $\{V_0, \ldots, V_{k-1}\}$

Output: k, $\{V_0, \ldots, V_{k-1}\}$

Limitation

- Use of non-deterministic algorithms
- Visual scalability
- Barycenter heuristic
- No comparison against optimal models



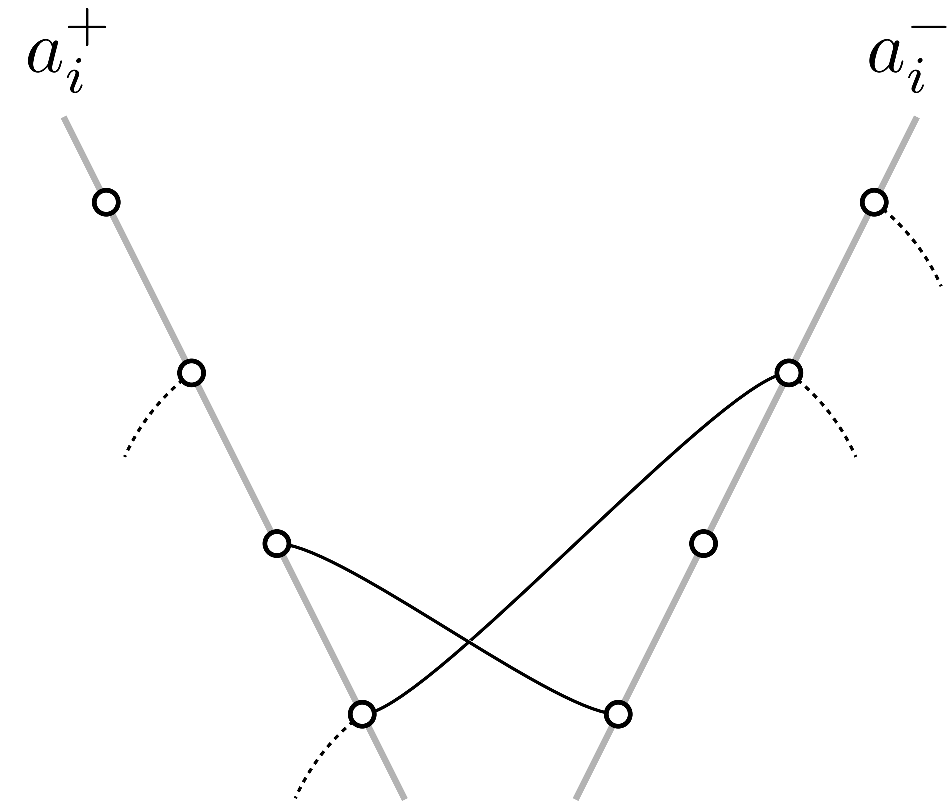
Crossing Minimization – Phase II

Two phase approach:

(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important



Crossing Minimization – Phase II

Two phase approach:

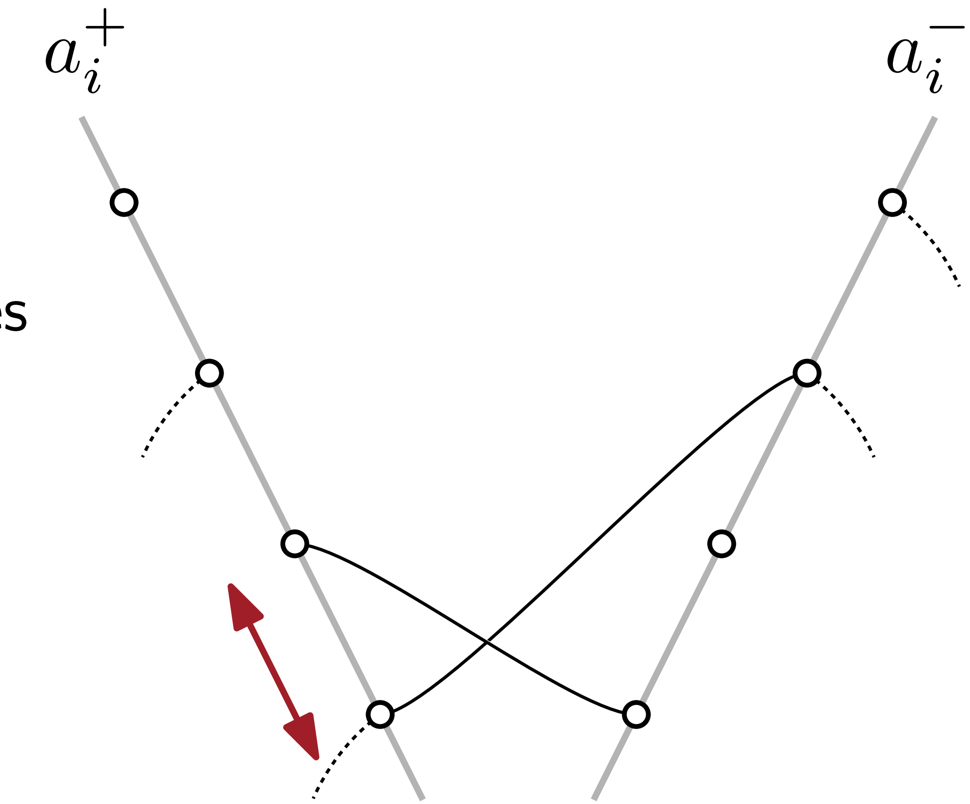
(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important

Fix relative order of all vertices with inter-axis edges

Perform one-sided crossing minimization



Crossing Minimization – Phase II

Two phase approach:

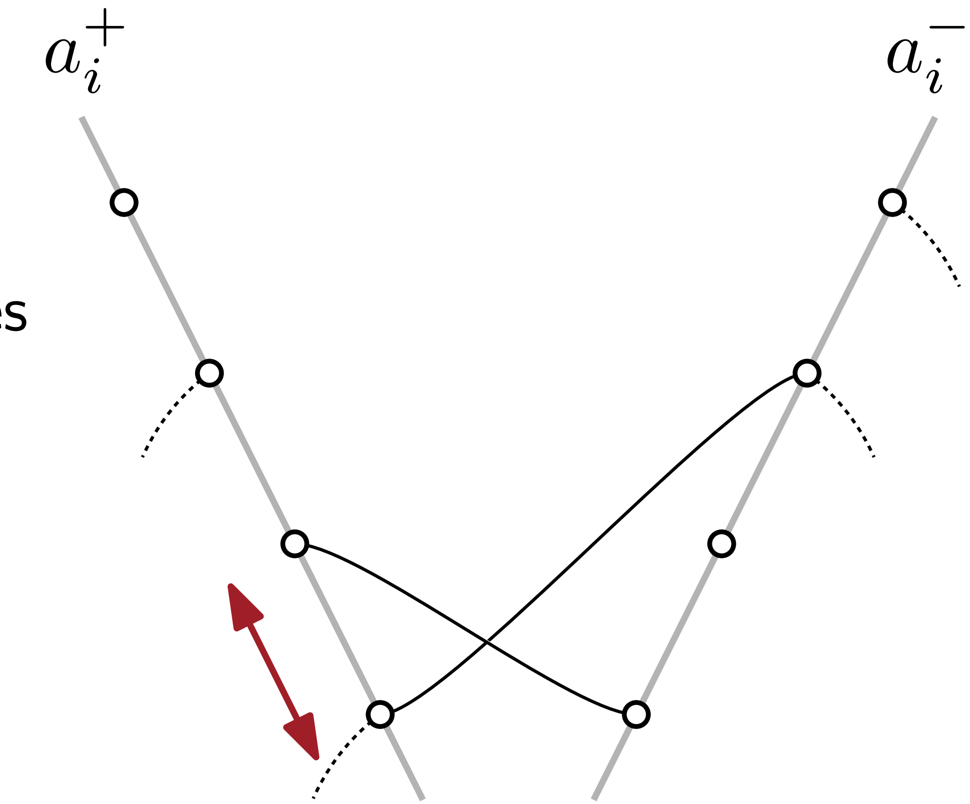
(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important

Fix relative order of all vertices with inter-axis edges

Perform one-sided crossing minimization



Crossing Minimization – Phase II

Two phase approach:

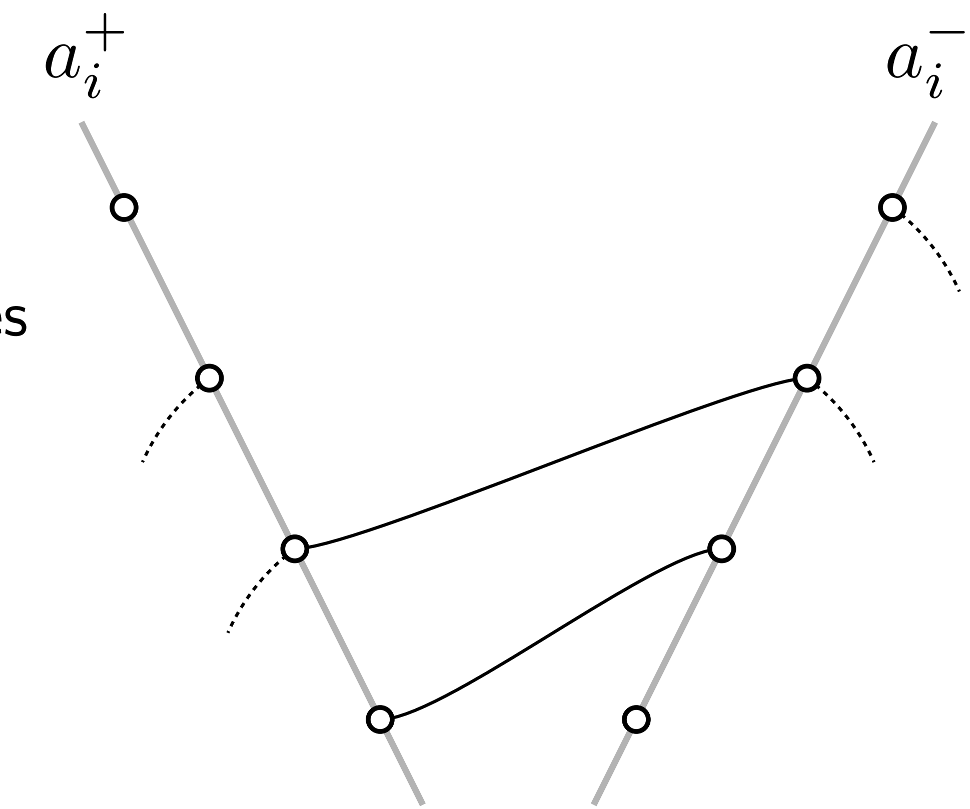
(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important

Fix relative order of all vertices with inter-axis edges

Perform one-sided crossing minimization



Crossing Minimization – Phase II

Two phase approach:

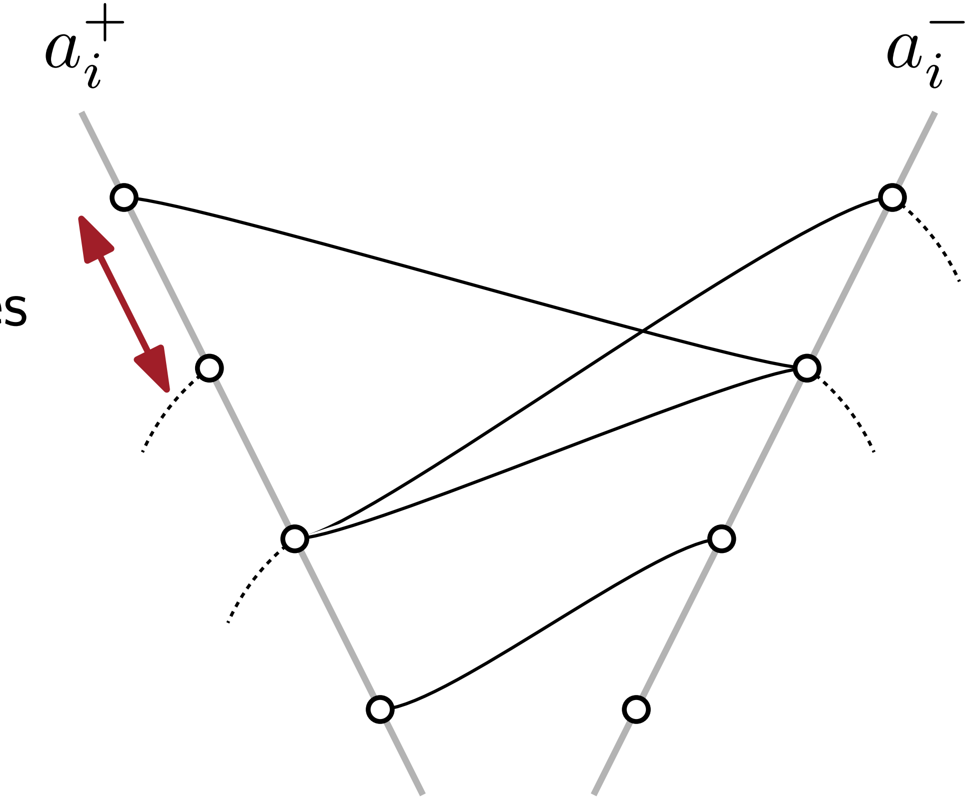
(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important

Fix relative order of all vertices with inter-axis edges

Perform one-sided crossing minimization



Crossing Minimization – Phase II

Two phase approach:

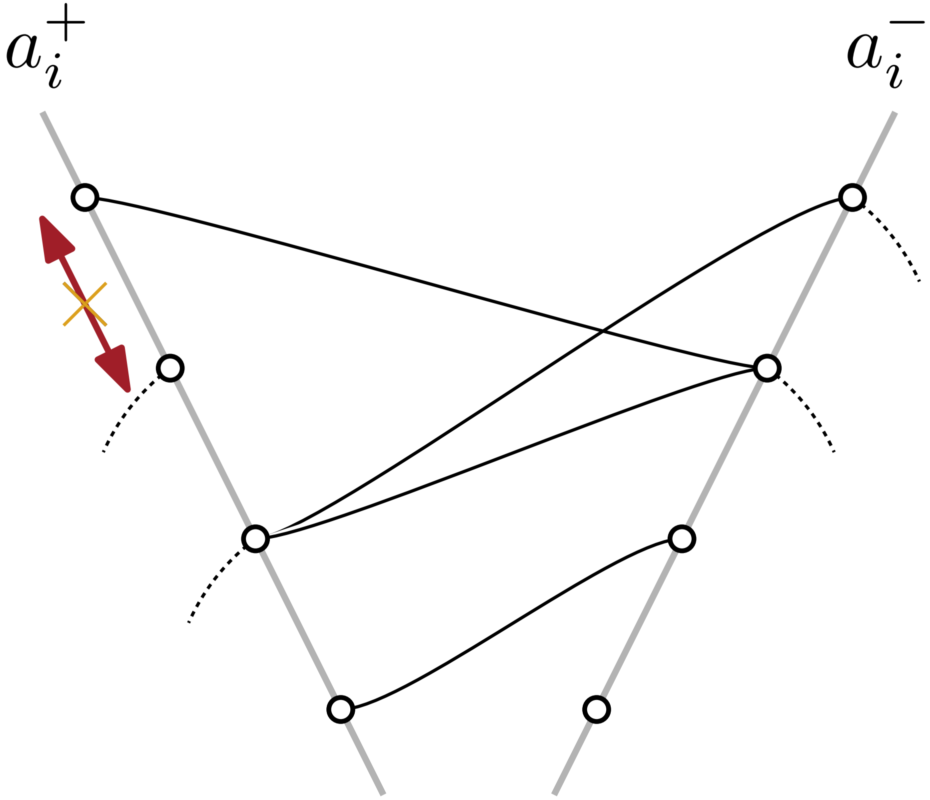
(I) Minimize inter-axis crossings

(II) Minimize intra-axis crossings

Minimizing inter-axis crossings more important

Fix relative order of all vertices with inter-axis edges

Perform one-sided crossing minimization



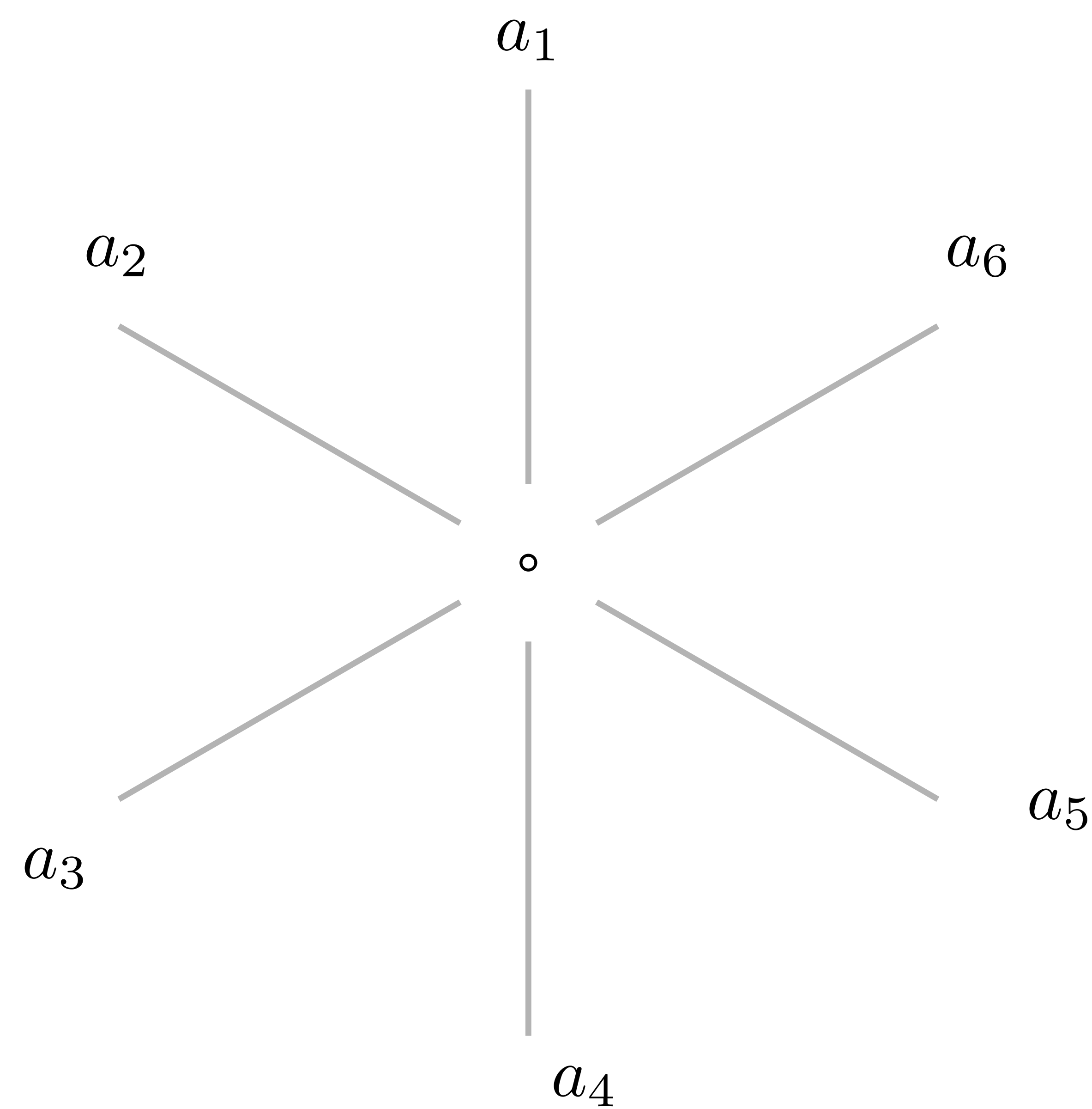
Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \ldots, a_k\}$ **Axes**



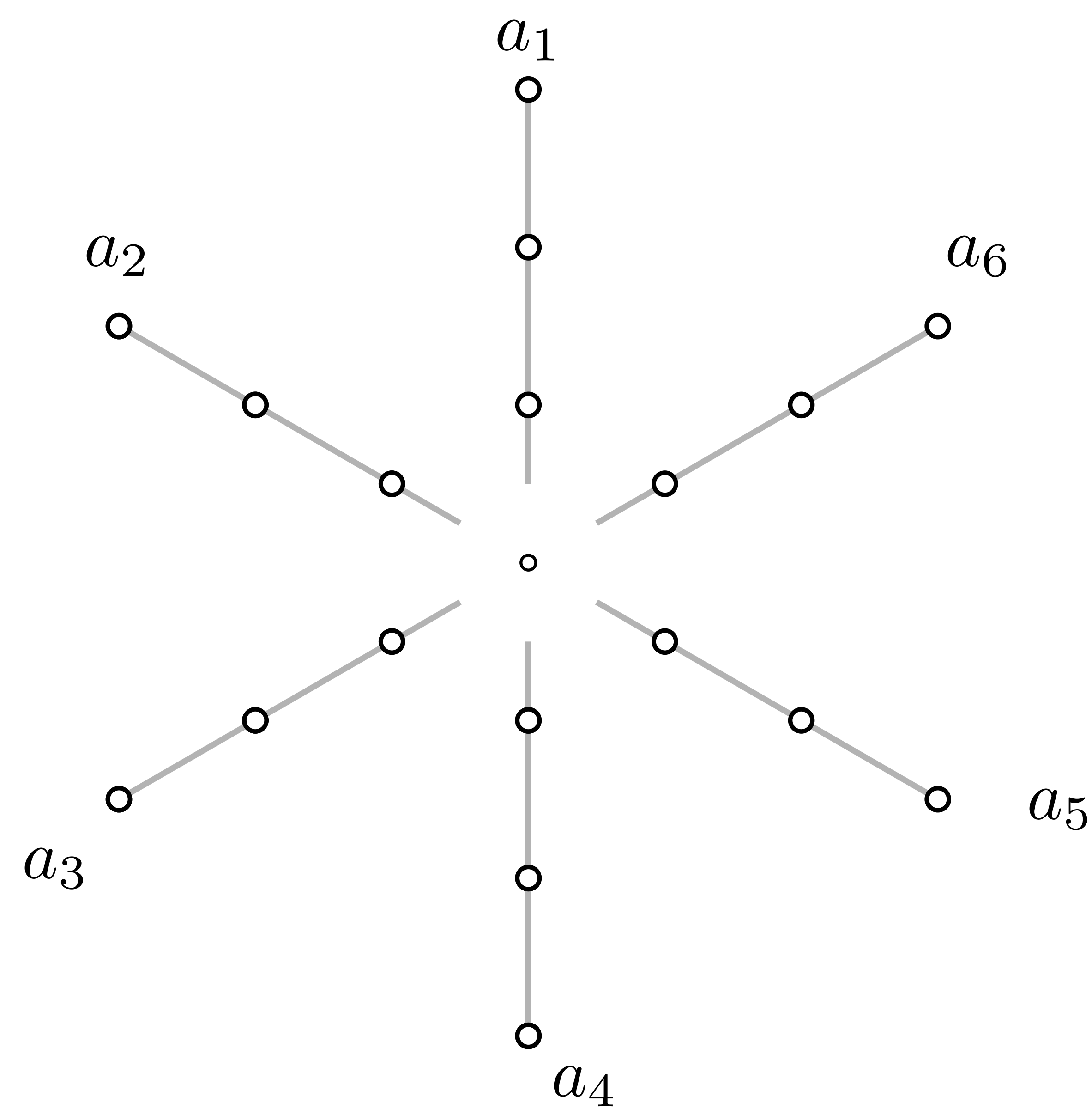
Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

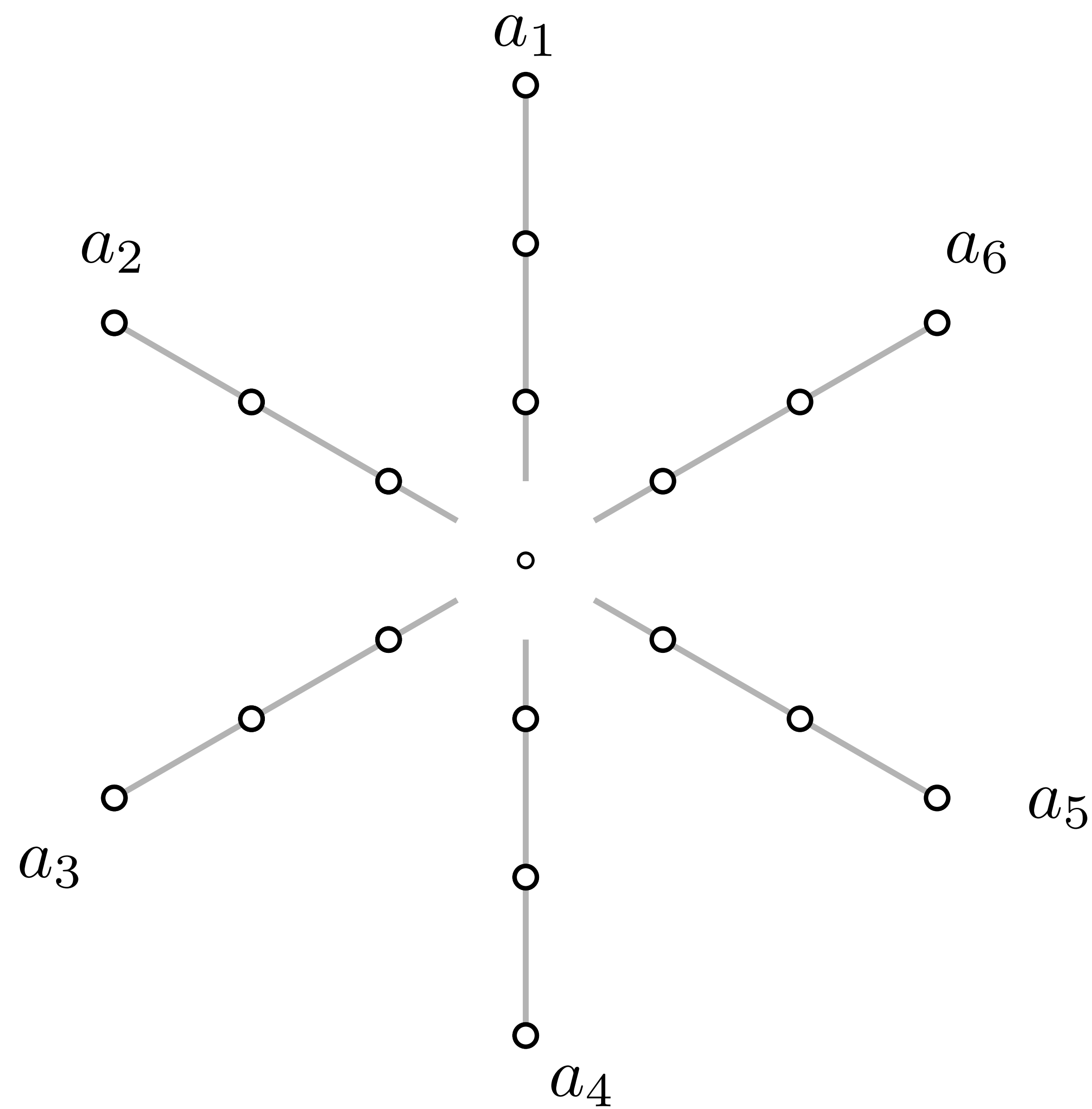
$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

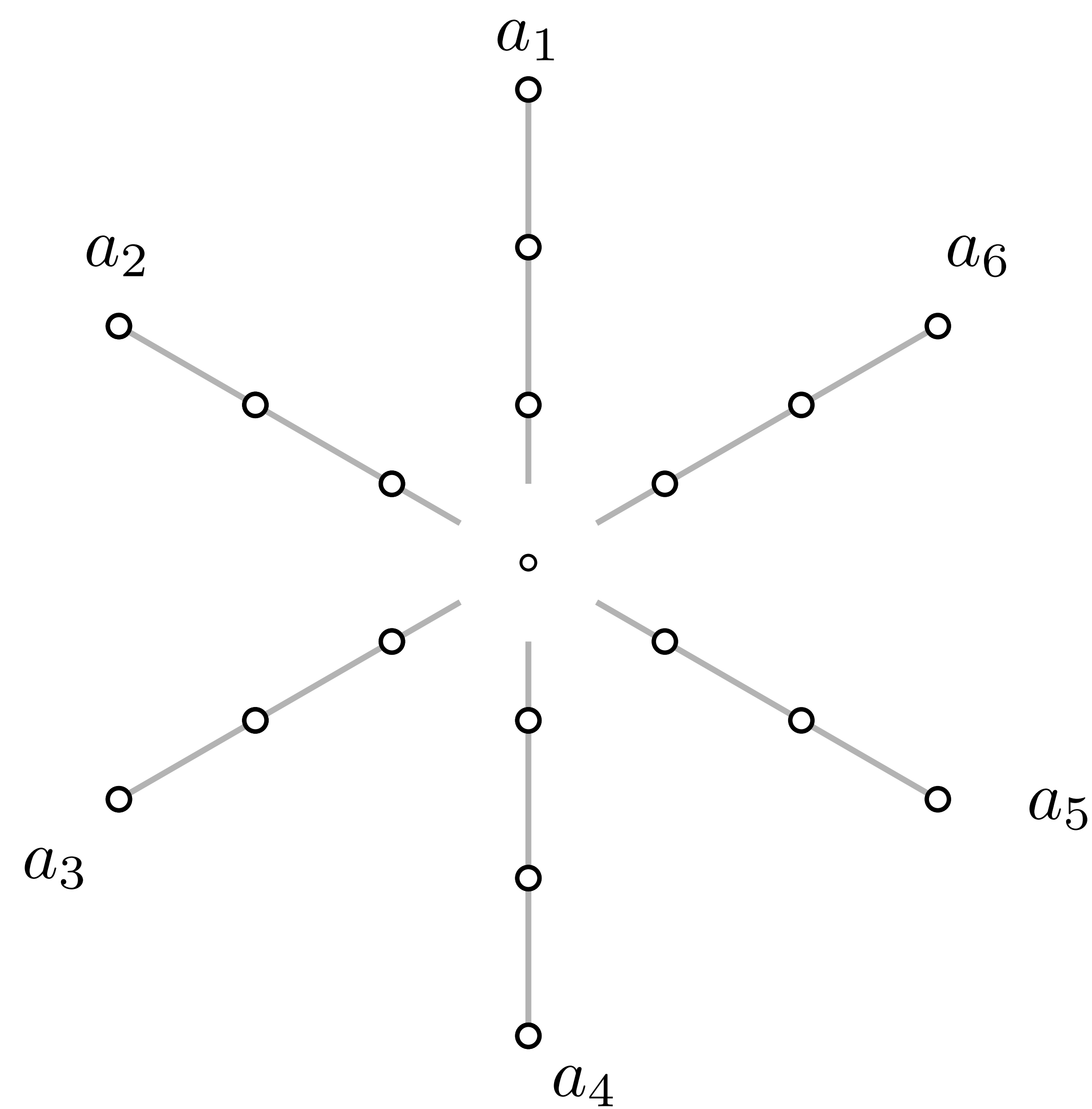
$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

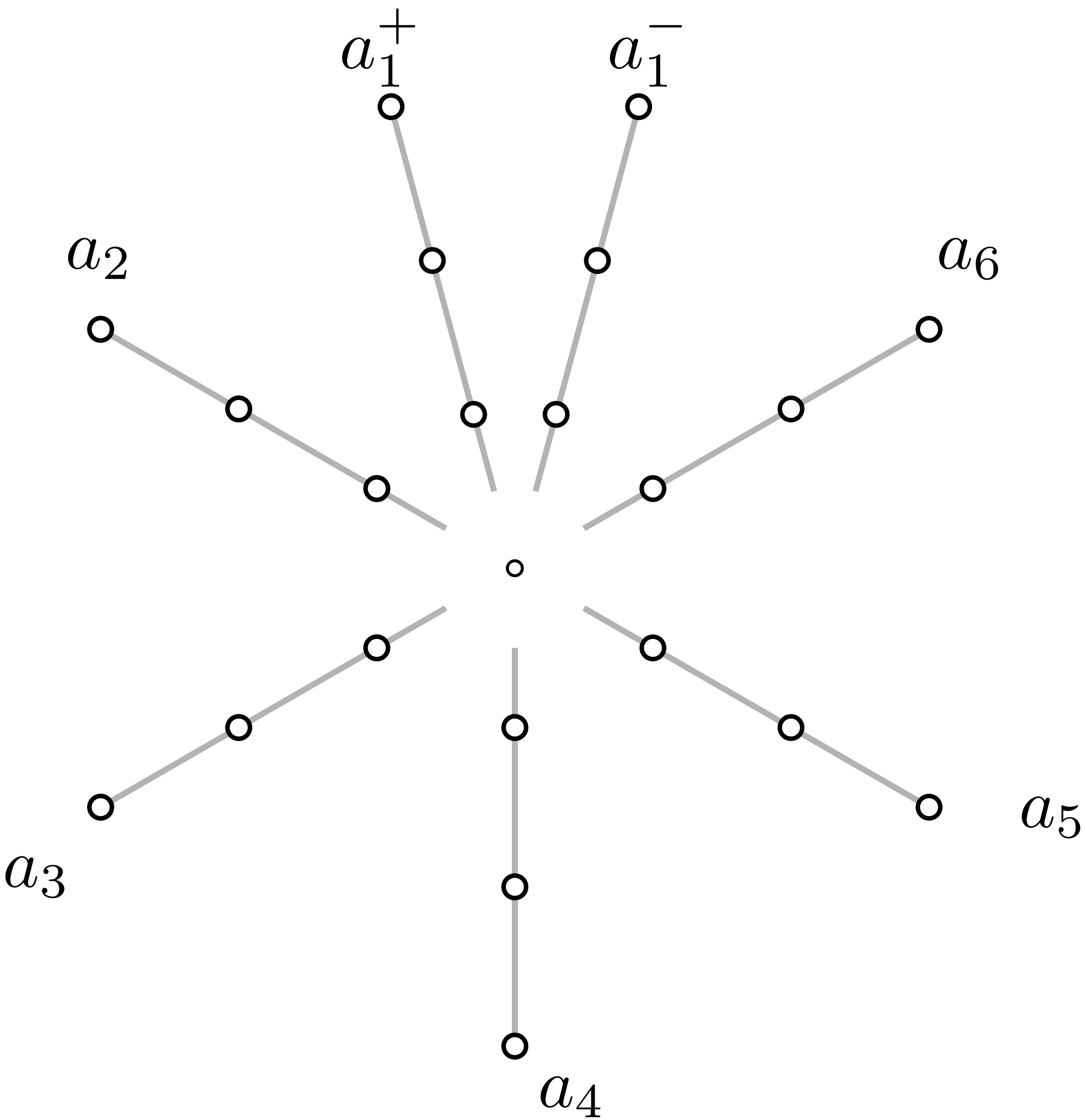
$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \rightarrow A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \rightarrow \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

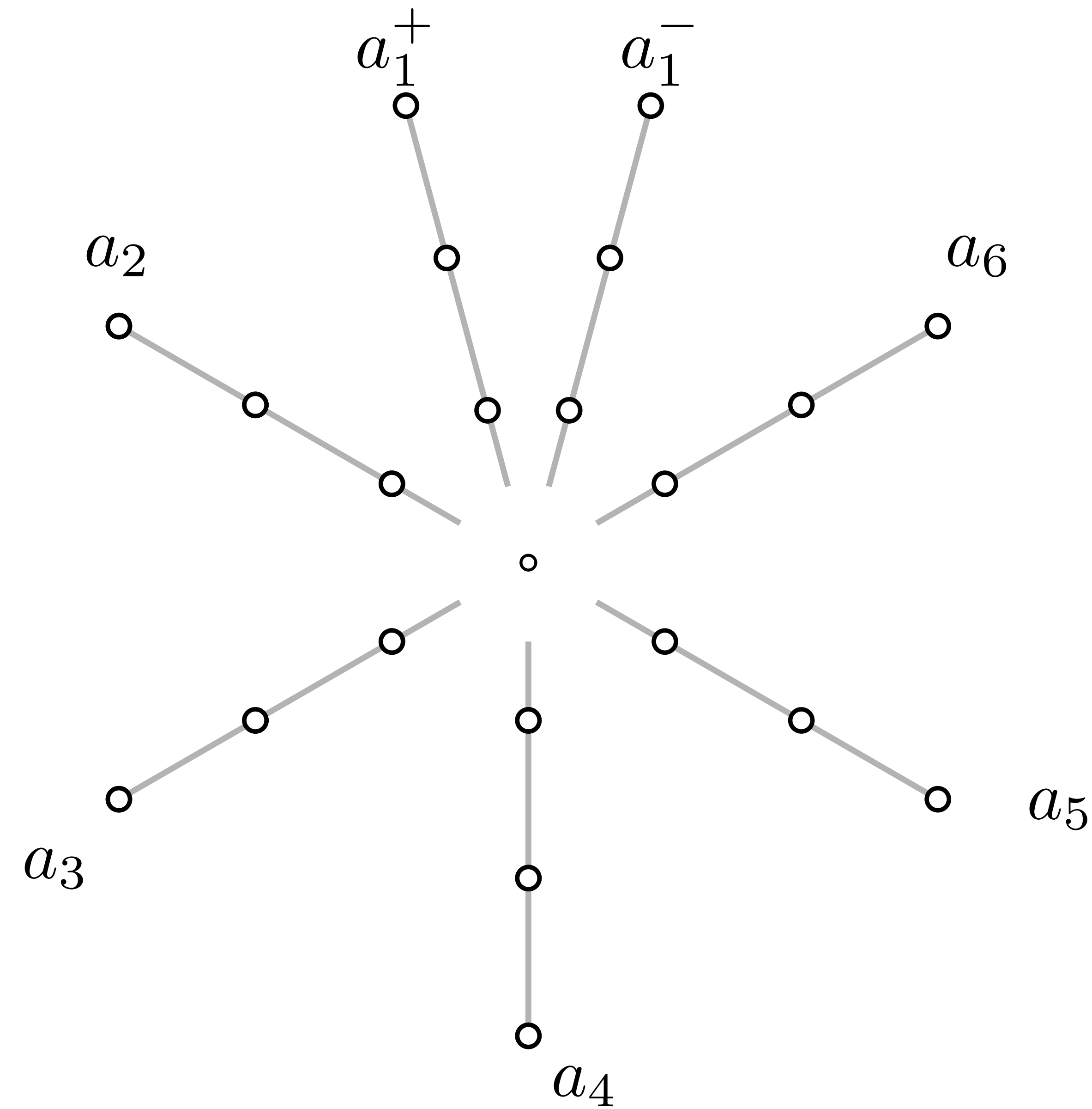
$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

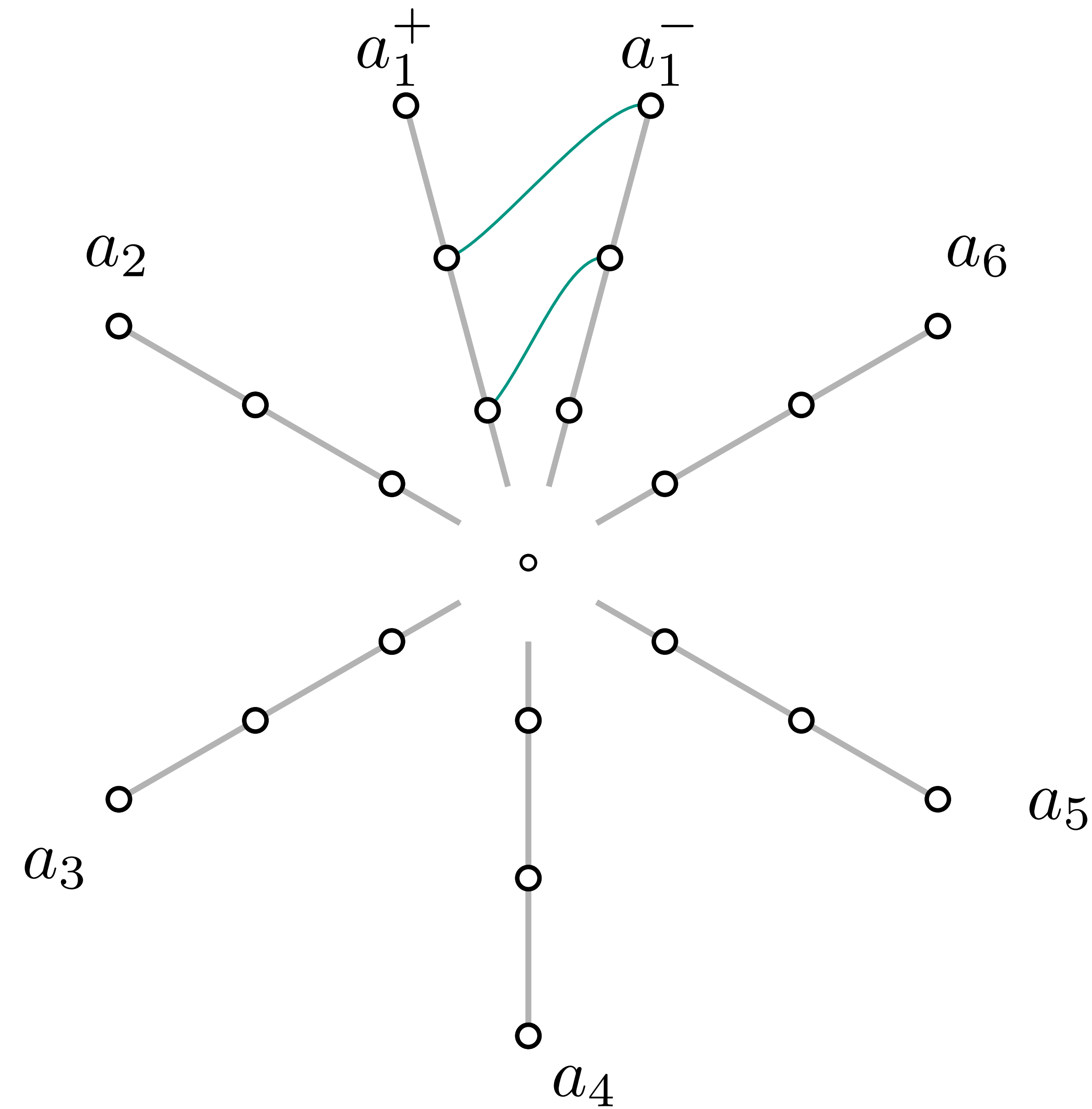
$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

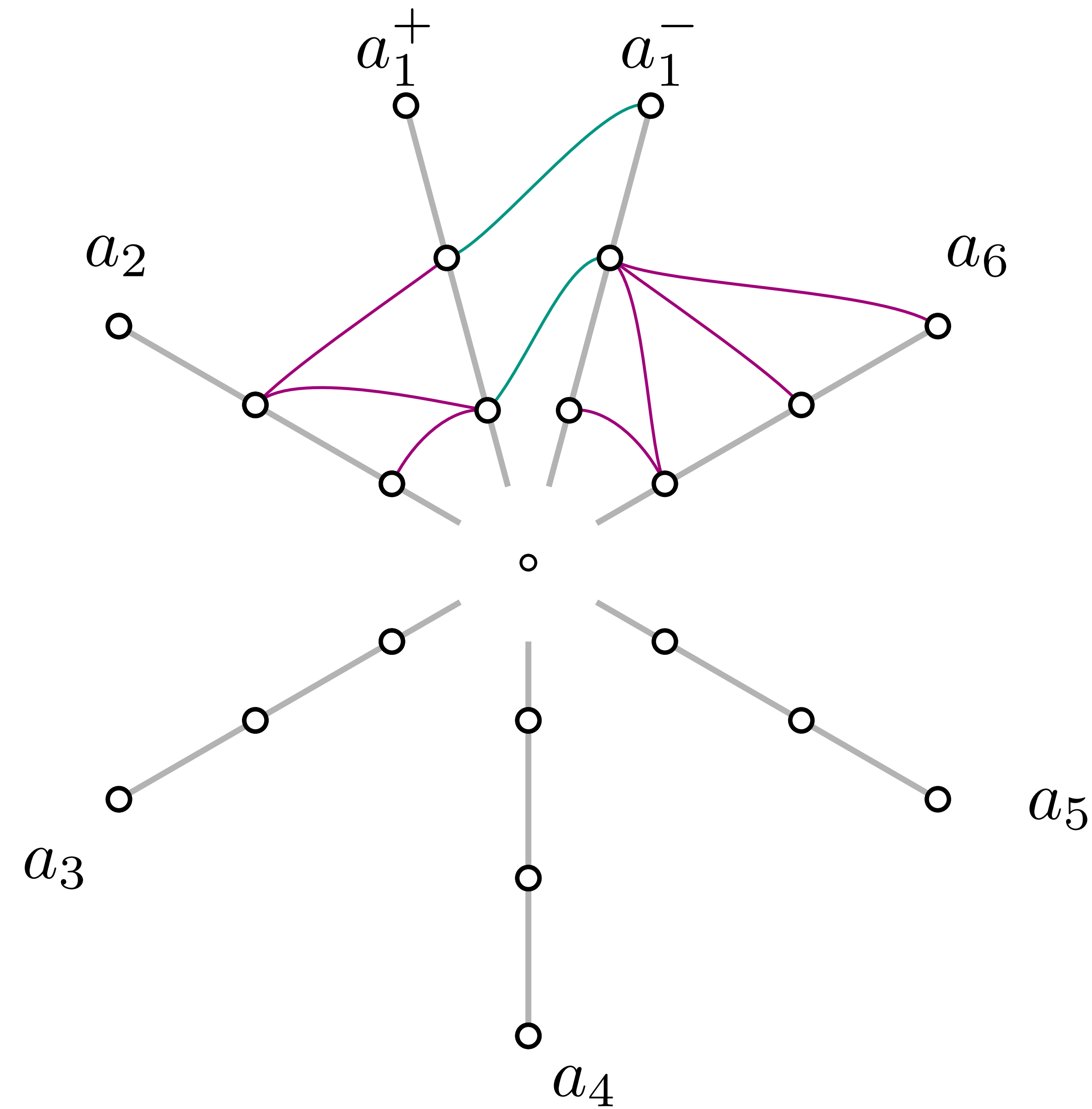
$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge

$span(u, v) = 1$ Proper edge



Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \dots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \dots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \dots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

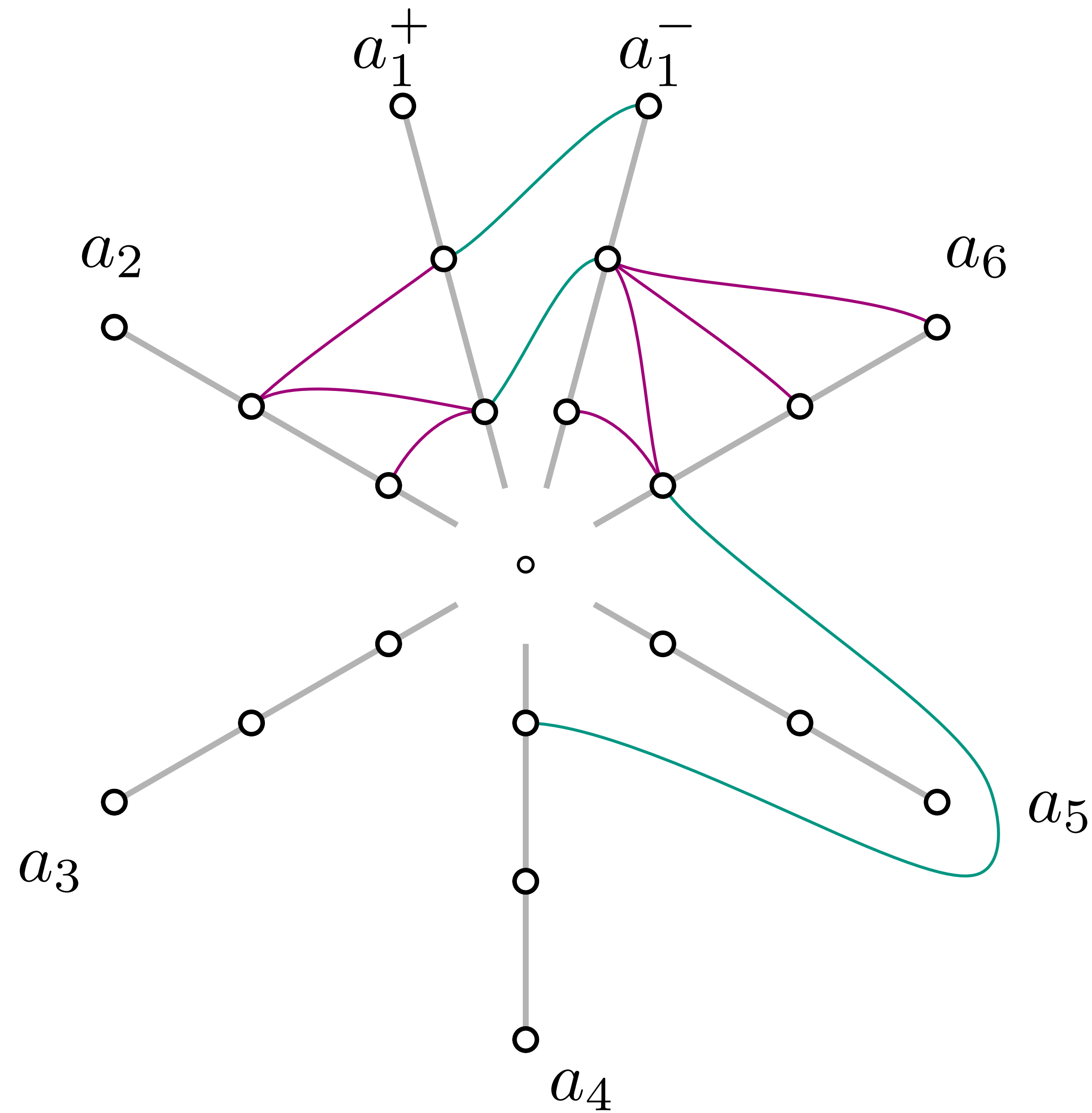
$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge
$span(u, v) = 1$ Proper edge
$span(u, v) > 1$ Long edge

(Inter-axis edge)



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

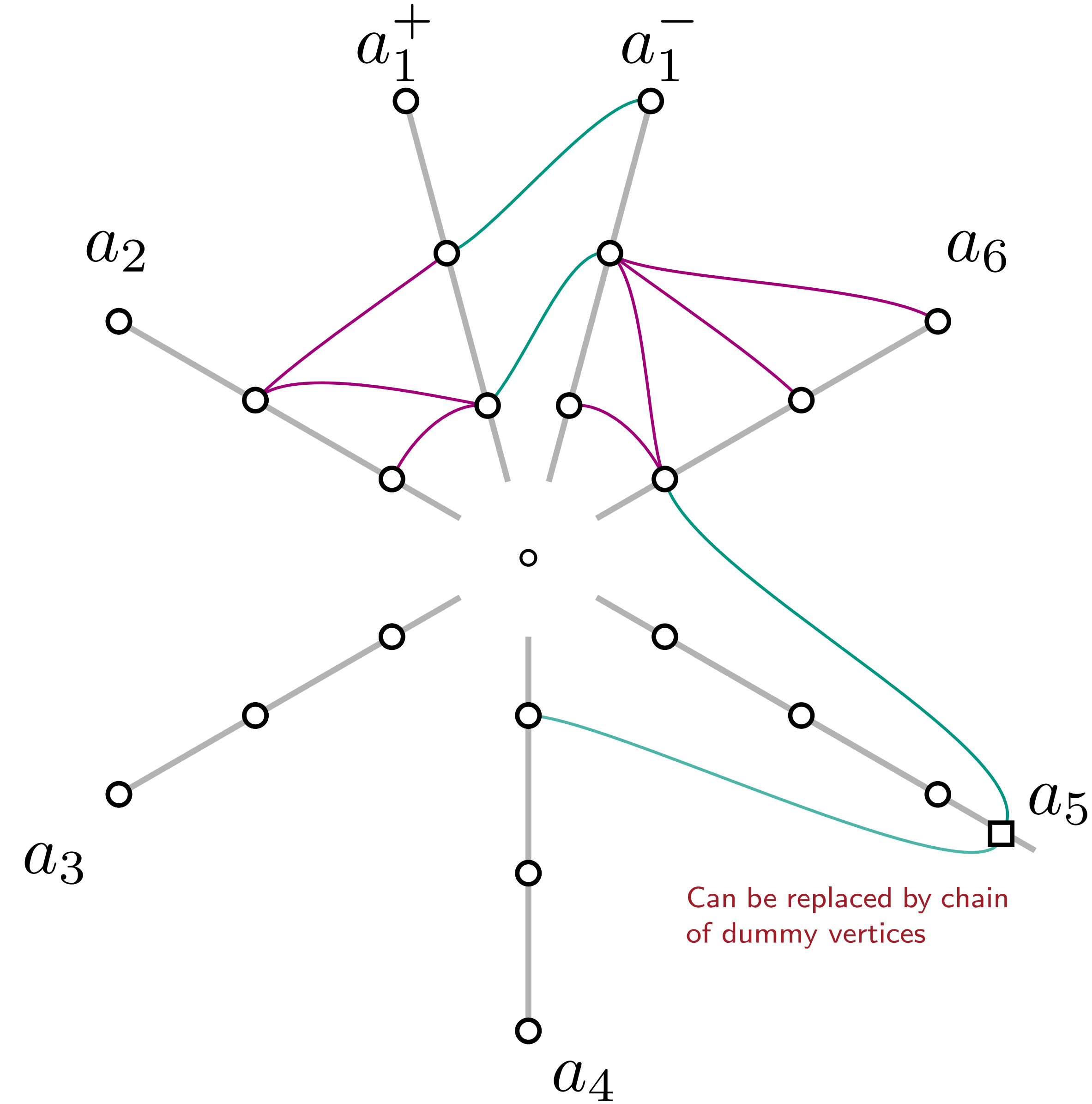
$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge

$span(u, v) = 1$ Proper edge

$span(u, v) > 1$ Long edge

(Inter-axis edge)



Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \rightarrow A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \rightarrow \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

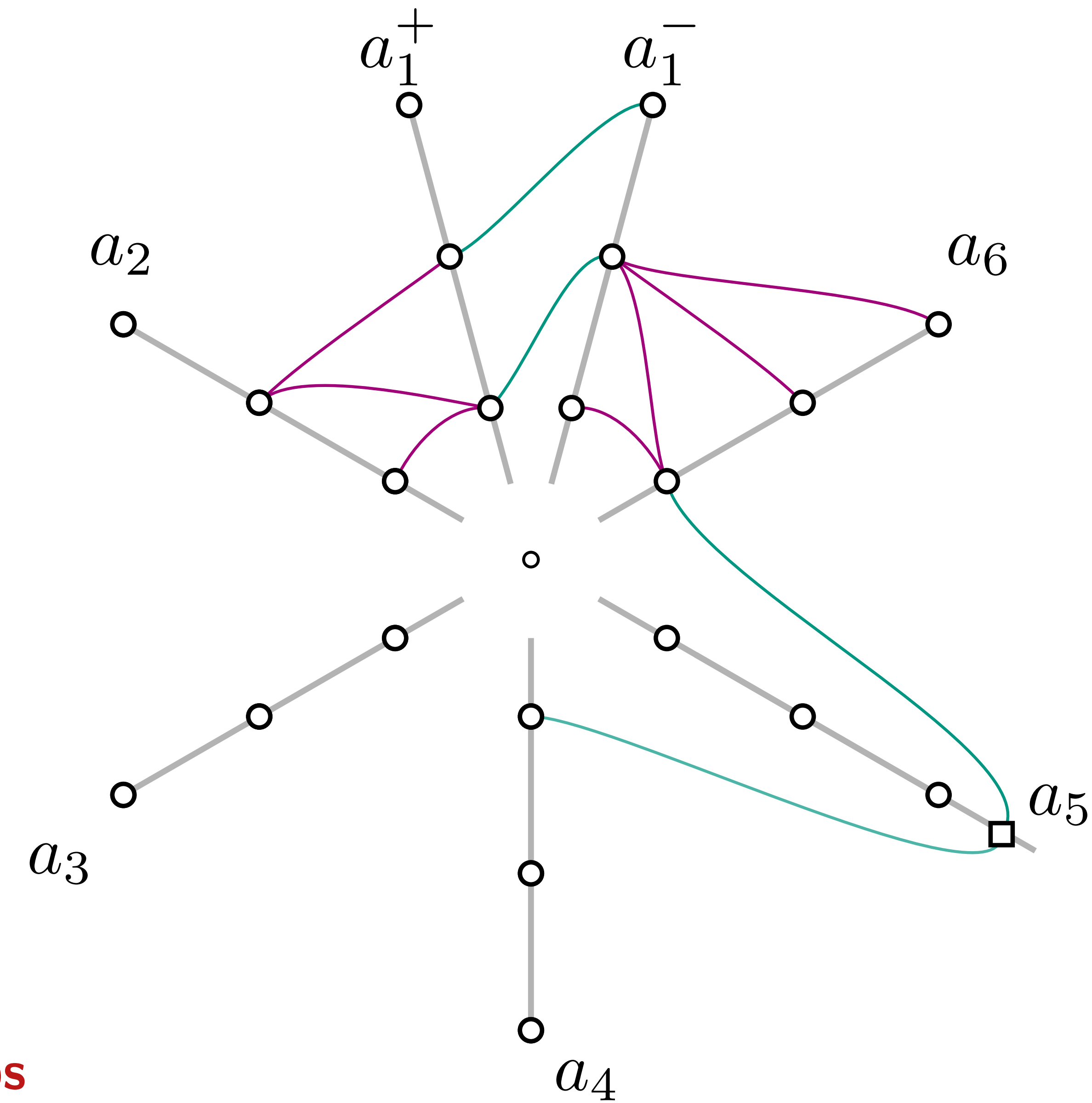
$span(u, v) = 0$ Intra-axis edge

$span(u, v) = 1$ Proper edge

$span(u, v) > 1$ Long edge

(Inter-axis edge)

Gaps



Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \rightarrow A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \rightarrow \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge

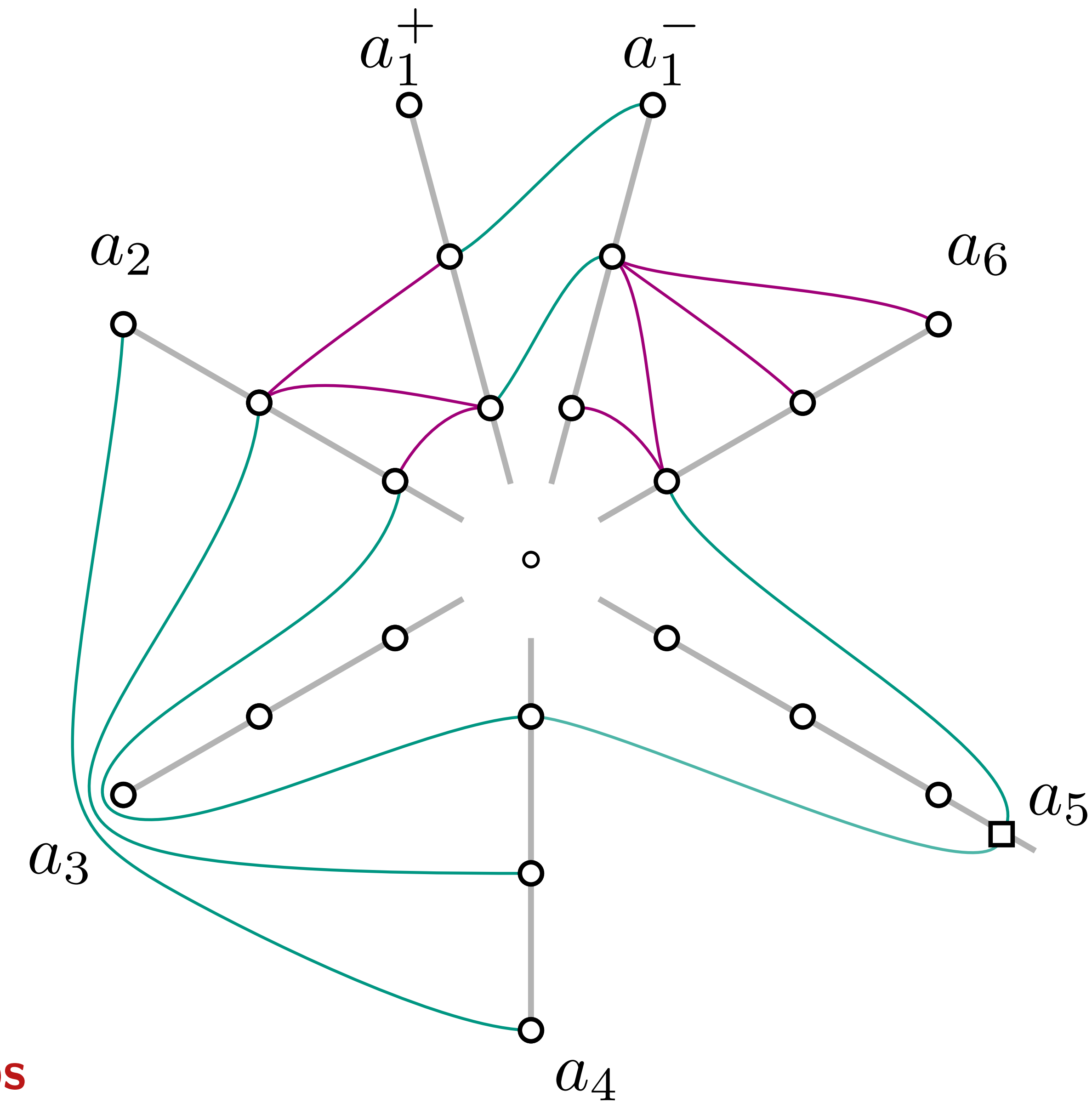
$span(u, v) = 1$ Proper edge

$span(u, v) > 1$ Long edge

(Inter-axis edge)

$g = 1$ **Gaps**

Outside



Combinatorial Model

$$H(G) = (A, \alpha, \phi, \Pi)$$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge

$span(u, v) = 1$ Proper edge

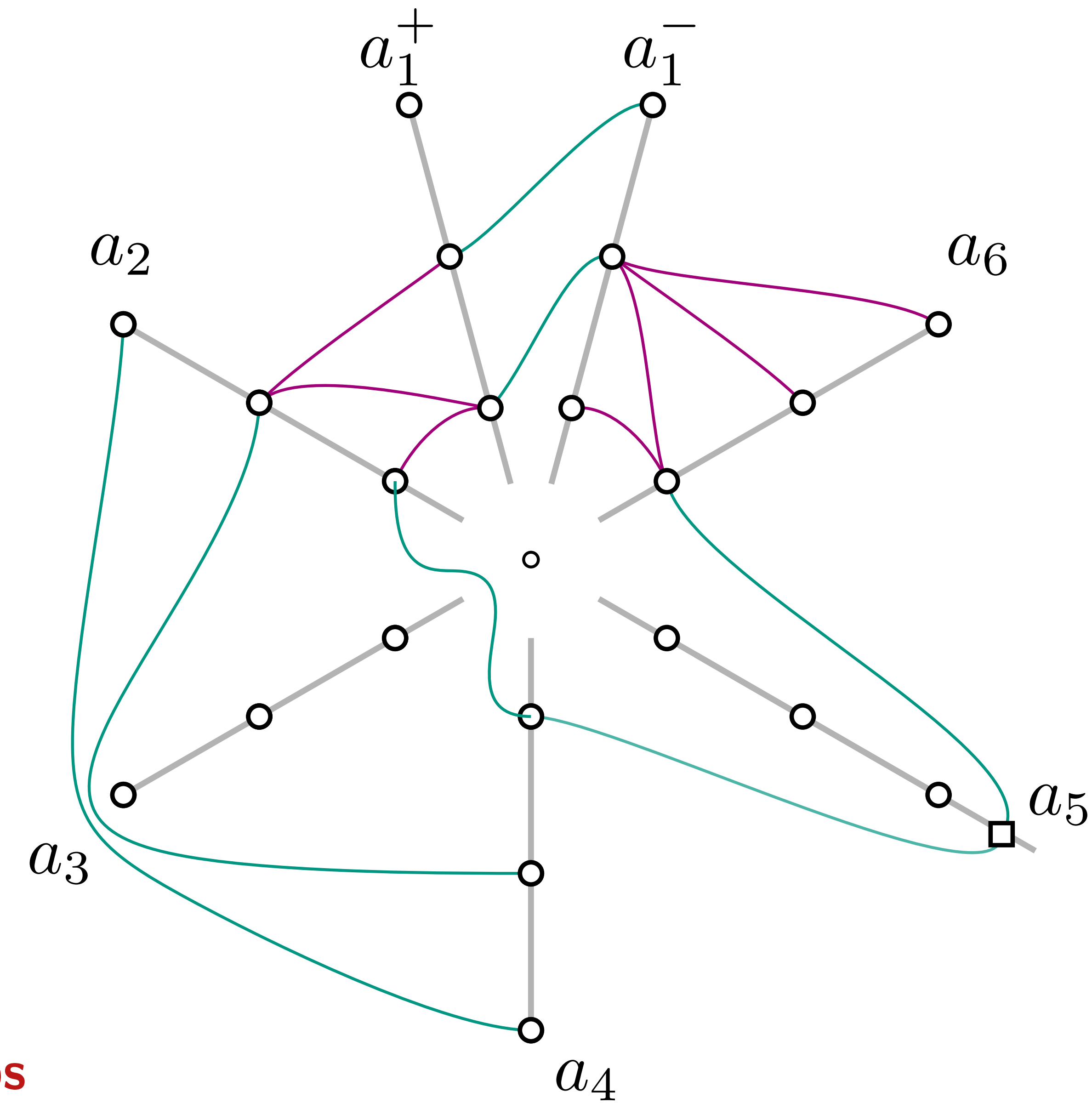
$span(u, v) > 1$ Long edge

(Inter-axis edge)

$g = 1$ Outside

$g = 2$ Outside & inside

Gaps



Combinatorial Model

$H(G) = (A, \alpha, \phi, \Pi)$

$A = \{a_1, \ldots, a_k\}$ **Axes**

$\alpha : V \to A$ **Vertex mapping**

$\alpha^{-1}(a_i) = V_i$

$\phi : A \to \{0, \ldots, |A| - 1\}$ **Axis order**

$\phi(u) = \phi(\alpha(u))$

$\Pi = \{\pi_1, \ldots, \pi_k\}$ **Vertex order**

$a_i^+, a_i^-, V_i^+, V_i^-$ **Duplicates**

$\pi_i = \pi_i^+ = \pi_i^-$

Span

$span(u, v) := min\{\phi(u) - \phi(v) \mod k, \phi(v) - \phi(v) \mod k\}$

$span(u, v) = 0$ Intra-axis edge
$span(u, v) = 1$ Proper edge
$span(u, v) > 1$ Long edge

(Inter-axis edge)

$g = 1$ Outside

$g = 2$ Outside & inside

$g > 2$ Outside & inside & gaps

Gaps