Applying the weighted barycentre method to interactive graph visualization

Peter Eades
University of Sydney

Thanks for some software:
• Hooman Reisi Dekhordi
• Patrick Eades
Graphs and Graph Drawings
What is a graph?
A *graph* consists of

- Nodes, sometimes called “vertices”, and
- Binary relationships, called “edges” between the nodes.

Example: a “Linked-In” style social network

<table>
<thead>
<tr>
<th>Nodes:</th>
<th>Edges:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice, Andrea,</td>
<td>- Bob is connected to Alice</td>
</tr>
<tr>
<td>Annie, Amelia,</td>
<td>- Bob is connected to Andrea</td>
</tr>
<tr>
<td>Bob, Brian,</td>
<td>- Bob is connected to Amelia</td>
</tr>
<tr>
<td>Bernard, Boyle</td>
<td>- Brian is connected to Alice</td>
</tr>
<tr>
<td></td>
<td>- Brian is connected to Andrea</td>
</tr>
<tr>
<td></td>
<td>- Brian is connected to Amelia</td>
</tr>
<tr>
<td></td>
<td>- Boyle is connected to Alice</td>
</tr>
<tr>
<td></td>
<td>- Boyle is connected to Andrea</td>
</tr>
<tr>
<td></td>
<td>- Boyle is connected to Annie</td>
</tr>
<tr>
<td></td>
<td>- Bernard is connected to Alice</td>
</tr>
<tr>
<td></td>
<td>- Bernard is connected to Andrea</td>
</tr>
<tr>
<td></td>
<td>- Bernard is connected to Amelia</td>
</tr>
<tr>
<td></td>
<td>- Bernard is connected to Annie</td>
</tr>
</tbody>
</table>
Notation:
The **degree** $\deg(u)$ of a node $u$ is the number of edges attached to $u$.

Example:
$$\deg(Brian) = 3; \quad \deg(Alice) = 4.$$  

Nodes:
- Alice, Andrea, Annie, Amelia, Bob, Brian, Bernard, Boyle

Edges
- Bob is connected to Alice
- Bob is connected to Andrea
- Bob is connected to Amelia
- Brian is connected to Alice
- Brian is connected to Andrea
- Brian is connected to Amelia
- Boyle is connected to Alice
- Boyle is connected to Andrea
- Boyle is connected to Annie
- Bernard is connected to Alice
- Bernard is connected to Andrea
- Bernard is connected to Annie
Notation:
The neighbour set $N(u)$ of a node $u$ is the set of nodes that are linked to $u$ by an edge.

Example:

$N(Brian) = \{Alice, Andrea, Amelia\}$;
$N(Alice) = \{Bob, Brian, Boyle, Bernard\}$.

Nodes:

• Alice, Andrea, Annie, Amelia, Bob, Brian, Bernard, Boyle

Edges

• Bob is connected to Alice
• Bob is connected to Andrea
• Bob is connected to Amelia
• Brian is connected to Alice
• Brian is connected to Andrea
• Brian is connected to Amelia
• Boyle is connected to Alice
• Boyle is connected to Andrea
• Boyle is connected to Annie
• Bernard is connected to Alice
• Bernard is connected to Andrea
• Bernard is connected to Annie
Note:

- Edges can be **weighted**.

Example: an *edge-weighted* “Linked-In” style social network

**Nodes:**

- Alice, Andrea, Annie, Amelia, Bob, Brian, Bernard, Boyle

**Edges**

- $w(\text{Bob} -- \text{Alice}) = 1$
- $w(\text{Bob} -- \text{Andrea}) = 5$
- $w(\text{Bob} -- \text{Amelia}) = 1$
- $w(\text{Brian} -- \text{Alice}) = 2$
- $w(\text{Brian} -- \text{Andrea}) = 100$
- $w(\text{Brian} -- \text{Amelia}) = 3$
- $w(\text{Boyle} -- \text{Alice}) = 1$
- $w(\text{Boyle} -- \text{Andrea}) = 1$
- $w(\text{Boyle} -- \text{Annie}) = 5$
- $w(\text{Bernard} -- \text{Alice}) = 1$
- $w(\text{Bernard} -- \text{Andrea}) = 9$
- $w(\text{Bernard} -- \text{Annie}) = 1$
What is a graph drawing?
A graph consists of
• Nodes, and
• Edges

A *graph drawing* is a picture of a graph. That is, a graph drawing is a mapping that assigns
• a location for each node, and
• a curve to each edge.
Nodes:
- Bob, Brian, Bernard, Boyle, Alice, Andrea, Annie, Amelia

Edges
- Bob is connected to Alice
- Bob is connected to Andrea
- Bob is connected to Amelia
- Brian is connected to Alice
- Brian is connected to Andrea
- Brian is connected to Amelia
- Boyle is connected to Alice
- Boyle is connected to Andrea
- Boyle is connected to Annie
- Bernard is connected to Alice
- Bernard is connected to Andrea
- Bernard is connected to Annie
A drawing of the graph

Nodes
0, 1, 2, 3, 4, 5, 6, 7

Edges
0 – 1
0 – 4
1 – 2
1 – 4
1 – 7
2 – 3
2 – 4
2 – 5
3 – 4
4 – 5
4 – 7
5 – 6
5 – 7
6 – 7

A graph

A drawing of the graph
A graph drawing is a *straight-line drawing* if every edge is a straight line segment.
What is *connectivity*?
Connectivity notions are fundamental in any study of graphs or networks

- A graph is connected if for every pair $u, v$ of vertices, there is a path between $u$ and $v$.

- A graph is $k$-connected if there is no set of $(k-1)$ vertices whose deletion disconnects the graph.
  
  - $k = 1$: “1-connected” ≡ “connected”
  - $k = 2$: “2-connected” ≡ “biconnected”
  - $k = 3$: “3-connected” ≡ “triconnected”
This graph is connected

This graph is not connected

Connected components

This graph is connected
Connectivity notions are fundamental in any study of networks

- A graph is \textit{connected} if for every pair \(u, v\) of vertices, there is a path between \(u\) and \(v\).

- A graph is \textit{k-connected} if there is no set of \((k-1)\) vertices whose deletion disconnects the graph.

  - \(k = 1\): “1-connected” \(\equiv\) “connected”
  - \(k = 2\): “2-connected” \(\equiv\) “biconnected”
  - \(k = 3\): “3-connected” \(\equiv\) “triconnected”
“2-connected” ≡ “biconnected”
- A cutvertex is a vertex whose removal would disconnect the graph.
- A graph without cutvertices is biconnected.
“3-connected” $\equiv$ “triconnected”

- A **separation pair** is a pair of vertices whose removal would disconnect the graph.
- A graph without separation pairs is **triconnected**.

This graph is triconnected

This graph is *not* triconnected
What is a *planar* graph?
A graph is *planar* if it can be drawn without edge crossings.
A graph is \textit{planar} if it can be drawn without edge crossings.

Nodes:
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Edges
- 0 – 4
- 0 – 9
- 1 – 2
- 1 – 6
- 1 – 7
- 2 – 3
- 2 – 8
- 3 – 4
- 4 – 5
- 4 – 8
- 5 – 6
- 5 – 7
- 7 – 8
A graph is **non-planar** if every drawing has at least one edge crossing.
What is a topological embedding?
A planar drawing divides the plane into faces.

The boundary-sharing relationships of the faces defines a topological embedding of the graph drawing.

- $F_0$ shares a boundary with $F_1$
- $F_0$ shares a boundary with $F_2$
- $F_0$ shares a boundary with $F_3$
- $F_0$ shares a boundary with $F_4$
- $F_1$ shares a boundary with $F_2$
- $F_1$ shares a boundary with $F_4$
- $F_2$ shares a boundary with $F_1$
- $F_2$ shares a boundary with $F_3$
- $F_2$ shares a boundary with $F_4$
- $F_3$ shares a boundary with $F_4$
What is a *good* graph drawing?
The input is a graph with no geometry

The output should be a good graph drawing:
- easy to understand,
- easy to remember,
- beautiful.
~1979 Intuition (Sugiyama et al. 1979; Batini et al 1982; etc.):

- Planar straight-line drawings make good pictures
Purchase et al., 1997:
Significant correlation between edge crossings and human understanding

➢ More edge crossings means more human errors in understanding
Purchase et al., 1997: Significant correlation between *straightness of edges* and human understanding

- More bends mean more human errors in understanding

Fig. 3. Results for the dense graph
What makes a *good* drawing of a graph?

- lack of edge crossings
- straightness of edges
- (plus some other things)

→ Planar straight-line drawings of graphs are GOOD!

(This talk is about planar straight-line drawings of graphs!)
Tuttle’s Barycentre Algorithm
W. T. Tutte 1917 - 2002
- Code breaker at Bletchley park
- Pioneer of graph theory

Tutte’s barycentre algorithm is the original graph drawing algorithm
Run VinciTest1
**Tutte’s barycentre algorithm**

Input: A planar graph $G = (V, E)$
Output: A straight-line drawing $p$ of $G$

Step 1. Choose a face $f$ of $G$.
Step 2. Draw $f$ as a convex polygon.
Step 3. For all $u$ not in $f$, choose $p(u)$ by

$$p(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} p(v)$$

where the sum is over all neighbors $v$ of $u$.

Vertex $u$ is placed at the “barycenter” of its neighbors.
Example output
Step 1. Choose a face $f$ of $G$

Step 2. Draw $f$ as a convex polygon

Step 3. For all $u$ not on $f$, choose $p(u)$ by

$$p(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} p(v)$$

where the sum is over all neighbors $v$ of $u$

$$x(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} x(v)$$

and

$$y(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} y(v)$$

$$2(|V| - |f|) = O(|V|)$$

linear equations

“barycentre equations”
Step 1. Choose a face \( f \) of \( G \)
Step 2. Draw \( f \) as a convex polygon
Step 3. For all \( u \) not on \( f \), choose 
\[
p(u) = (x(u), y(u))
\]
by
\[
x(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} x(v)
\]
and
\[
y(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} y(v)
\]
where the sum is over all neighbors \( v \) of \( u \)

The critical part is step 3 ....
Step 1. \( f = \{4, 5, 6, 7, 8\} \)

Step 2. For all \( i = 4, 5, 6, 7, 8 \), choose the location \((x_i, y_i)\) of \( i \) so that they form a convex polygon.

Step 3. Find \( x_1, x_2, x_3 \) such that:
\[
\begin{align*}
x_1 &= \frac{1}{4} (x_2 + x_3 + x'_4 + x'_8) \\
x_2 &= \frac{1}{4} (x_1 + x_3 + x'_5 + x'_6) \\
x_3 &= \frac{1}{3} (x_1 + x_2 + x'_7)
\end{align*}
\]

and find \( y_1, y_2, y_3 \) such that:
\[
\begin{align*}
y_1 &= \frac{1}{4} (y_2 + y_3 + y'_4 + y'_8) \\
y_2 &= \frac{1}{4} (y_1 + y_3 + y'_5 + y'_6) \\
y_3 &= \frac{1}{3} (y_1 + y_2 + y'_7)
\end{align*}
\]

where \( x'_i \) and \( y'_i \) are the values chosen at step 2.
Re-write the equations for step 3:

\[
4x_1 - x_2 - x_3 = x'_4 + x'_8 \\
-x_1 + 4x_2 - x_3 = x'_5 + x'_6 \\
-x_1 - x_2 + 3x_3 = x'_5
\]

and

\[
4y_1 - y_2 - y_3 = y'_4 + y'_8 \\
-y_1 + 4y_2 - y_3 = y'_5 + y'_6 \\
-y_1 - y_2 + 3y_3 = y'_5
\]

That is:

\[
4x_1 - x_2 - x_3 = c_1 \\
-x_1 + 4x_2 - x_3 = c_2 \\
-x_1 - x_2 + 3x_3 = c_3
\]

and

\[
4y_1 - y_2 - y_3 = d_1 \\
-y_1 + 4y_2 - y_3 = d_2 \\
-y_1 - y_2 + 3y_3 = d_3
\]

where \( c_1, c_2, c_3, d_1, d_2, d_3 \) are constants
Re-writing the equations using a matrix:

We want vectors $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ such that $Mx = c$ and $My = d$ where

$M = \begin{bmatrix} 4 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 3 \end{bmatrix}$

The essence of Tutte’s barycentre algorithm is inverting a matrix.
**Tutte’s barycentre algorithm**

Input: A triconnected planar graph \( G = (V, E) \)

Output: A planar straight-line drawing \( p \) of \( G \)

Step 1. Choose a face \( f \) of \( G \)

Step 2. Draw \( f \) as a convex polygon

Step 3.

(a) Let \( M \) be the square symmetric matrix indexed by the vertices of \( G - f \) defined by

\[
M_{uv} = \begin{cases} 
\deg(u) & \text{if } u = v \\
-1 & \text{if } (u, v) \in E \\
0 & \text{otherwise}
\end{cases}
\]

The matrix of the barycentre equations is a Laplacian submatrix.

(b) Let \( c \) and \( d \) be vectors indexed by the vertices of \( G - f \) defined by

\[
c_u = \sum_{w \in f} x_w \quad \text{and} \quad d_u = \sum_{w \in f} y_w
\]

(c) Let \( x \) and \( y \) be vectors indexed by the vertices of \( G - f \) defined by

\[
x = M^{-1}c \quad \text{and} \quad y = M^{-1}d
\]

(d) Let \( p(u) = (x_u, y_u) \) for all \( u \) not on \( f \).
A more general barycentre algorithm

Step 1. Choose a subset $A$ of $V$
Step 2. Choose a location $p(a) = (x(a), y(a))$ for each vertex $a \in A$
Step 3. Solve the barycentre equations to give a position $p(u) = (x(u), y(u))$
for each vertex $u \in V - A$

The subset $A$ of $V$ is called the set of fixed nodes.
A few lemmas and theorems about the barycentre equations
Definition: If $A \subseteq V$ then the $A$-contraction $G_A$ of $G$ is the graph obtained from $G$ by identifying all vertices in $A$ (ie, contracting $A$ to a single vertex).

**Determinant Lemma:** If $G_A$ is connected then the matrix $M$ of the barycentre equations has a positive determinant.

Proof: Follows from Kirchhoff's spanning tree theorem.

(actually, even more is true: The barycentre matrix is positive definite)
Uniqueness Lemma: If $G_A$ is connected then the barycentre equations have a unique solution.
Tutte’s amazing theorem

**Theorem** (Tutte 1960/63)
If the input graph is planar and triconnected, then
  • the drawing output by the barycentre algorithm is planar, and
  • every face is convex.
Implementation

The main part of the barycentre algorithm inverting the $n \times n$ barycentre matrix $M$.

- The usual Gaussian elimination: $O(n^3)$

- Williams algorithm: $O(n^{2.373})$

- If the graph is planar, then the matrix $M$ is sparse, then we can use Lipton-Tarjan nested dissection: $O(n^{1.5})$

- In practice we can use numerical methods (such as Gauss-Seidel / Jacobi iterations); these converge rapidly since $M$ is diagonally dominant.
Example (planar graph, 100 vertices)
Example (planar graph, 100 vertices)
Example (planar graph, 200 vertices)
Example (planar graph, 1000 vertices)
The Weighted Barycentre Algorithm
**Weighted barycentre algorithm**

Input: A positive-edge-weighted triconnected planar graph $G = (V, E, w)$
Output: A straight-line drawing $p$ of $G$

Step 1. Choose a face $f$ of $G$
Step 2. Draw $f$ as a convex polygon
Step 3. For all $u$ not on $f$, choose $p(u)$ by

$$p(u) = \frac{1}{w\text{Deg}(u)} \sum_{v \in N(u)} w_{uv}p(v)$$

where

$$w\text{Deg}(u) = \sum_{v \in N(u)} w_{uv}$$

Vertex $u$ is placed at the "weighted barycenter" of its neighbors.
Weighted barycentre algorithm
Input: A weighted triconnected planar graph \( G = (V, E, w) \)
Output: A planar straight-line drawing \( p \) of \( G \)

Step 1. Choose a face \( f \) of \( G \)
Step 2. Draw \( f \) as a convex polygon
Step 3.
(a) Let \( M \) be the square symmetric matrix indexed by the vertices of \( G - f \) defined by
\[
M_{uv} = \begin{cases} 
\sum_{v \in N(u)} w_{uv} & \text{if } u = v \\
- \sum_{v \in N(u)} w_{uv} p(v) & \text{if } (u, v) \in E \\
0 & \text{otherwise}
\end{cases}
\]
(b) Let \( x \) and \( y \) be vectors indexed by the vertices of \( G - f \) defined by
\[
x = M^{-1} c \quad \text{and} \quad y = M^{-1} d
\]
(c) Let \( p(u) = (x_u, y_u) \) for all \( u \) not on \( f \).

The matrix is diagonally dominant and has positive entries.
A few lemmas and theorems about the *weighted* barycentre equations
**Determinant Lemma**: If $G_A$ is connected then the matrix $M$ of the barycentre equations has a positive determinant.

**Uniqueness Lemma**: If $G_A$ is connected then the barycentre equations have a unique solution.
Corollary to Tutte’s **amazing** theorem (Floater 1997, maybe earlier)

If the input graph is planar and triconnected, then
- the drawing output by the *weighted* barycentre algorithm is planar, and
- every face is convex.
The Energy View
Tutte’s barycentre algorithm is “force-directed”
Tutte’s barycenter algorithm:

**The energy view**

1. Choose a set $A$ of vertices.
2. Choose a location $p(a)$ for each $a \in A$.
3. Place all the other vertices to minimize energy.

What is the energy of a drawing $p$?

- The Euclidean distance between $u$ and $v$ in the drawing $p$ is:
  \[
  d(u, v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}
  \]

- The energy in the edge $(u, v)$ is $d(u, v)^2 = (x_u - x_v)^2 + (y_u - y_v)^2$

- The energy in the drawing $p$ is the sum of the energy in its edges, ie,
  \[
  \text{energy}(p) = \sum_{(u,v) \in E} d(u, v)^2 = \sum_{(u,v) \in E} (x_u - x_v)^2 + (y_u - y_v)^2
  \]
Tutte’s barycenter algorithm: *The energy view*

We represent each vertex by a steel ring, and represent each edge by a spring of natural length zero connecting the rings at its endpoints.

1. Choose a set $A$ of vertices.
2. For each $a \in A$, nail the ring representing $a$ to the floor at some position.
3. The vertices in $V - A$ will move around a bit; when the movement stops, take a photo of the layout; this is the drawing.

Note: for Tutte’s barycentre algorithm:
- The springs have zero natural length
- Similar to elastic bands
How to minimize energy:

➢ Choose a location \( p(u) = (x_u, y_u) \) for each \( u \in V - A \) to minimize

\[
\text{energy}(p) = \sum_{(u,v) \in E} d(u,v)^2 = \sum_{(u,v) \in E} (x_u - x_v)^2 + (y_u - y_v)^2
\]

Note that the minimum is unique, and occurs when

\[
\frac{\partial (\text{energy}(p))}{\partial x_u} = 0 \quad \text{and} \quad \frac{\partial (\text{energy}(p))}{\partial y_u} = 0
\]

for each \( u \in V - A \).
For $x_u$, the minimum occurs when:

$$\frac{\partial \ (\text{energy}(p))}{\partial x_u} = 0$$

$$\frac{\partial}{\partial x_u} \left( \sum_{(u,v)\in E} (x_u - x_v)^2 + (y_u - y_v)^2 \right) = 0$$

$$\sum_{v \in V \ s.t. (u,v) \in E} 2(x_u - x_v) = 0$$

$$x_u = \frac{1}{\text{deg}(u)} \sum_{v \in V \ s.t. (u,v) \in E} x_v$$
The weighted barycentre algorithm is a *force directed algorithm*

- Edges are Hooke’s law springs with zero natural length.
- The weights define strengths for the springs.
- The fixed set are nodes nailed down (i.e., they don’t move).
Tutte’s barycenter algorithm: \textit{The energy view}

Step 1. Choose a face $f$ of $G$
Step 2. Draw $f$ as a convex polygon
Step 3: \textit{Place all the other vertices to minimize energy.}

The \textit{biggest differences} between the weighted barycentre algorithm and other force-directed methods are:

- For the barycentre algorithm, the springs have zero natural length.
- For the barycentre algorithm, the equations are easy to solve.
Animation
The animation problem:

- Given two drawings of a graph $G$, find a “nice” animation that takes one to the other.

Problem: how to compute the frames in between.
Run VinciTest2
The animation problem:

- Given two planar drawings \( \vec{p}_0 \) and \( \vec{p}_1 \) of a graph \( G \):
  - Find a “nice” animation that takes \( \vec{p}_0 \) to \( \vec{p}_1 \).
- Here we represent the drawing \( \vec{p}_0 \) as a \( 2 \times n \) vector with a row \((x_0(u), y_0(u))\) for each vertex \( u \); similarly for \( \vec{p}_1 \).

The animation consists of a drawing \( \vec{p}_t \) for each \( t \) with \( 0 \leq t \leq 1 \).

We want two properties of the animation:

1. **Planarity**: The drawing \( \vec{p}_t \) is planar for \( 0 \leq t \leq 1 \),
2. **Smoothness**: The animation is smooth, that is, the animation function \( a: t \to \vec{p}_t \) is continuous and differentiable.
The animation problem:

Given two planar drawings \( \tilde{p}_0 \) and \( \tilde{p}_1 \) of a graph \( G \):
Find an animation \( a: t \rightarrow \tilde{p}_t \) with two properties:

- Planarity
- Smoothness

Solution (Floater and Gotsman, 1999, maybe others):

If \( \tilde{p}_0 \) and \( \tilde{p}_1 \) are weighted barycentre drawings, then the animation problem is easy: just *interpolate the weights.*
Say drawings $\tilde{p}_0$ and $\tilde{p}_1$ are barycentre drawings with barycentre matrices $M_0$ and $M_1$, that is,
\[ M_0 \tilde{p}_0 = \tilde{c} \text{ and } M_1 \tilde{p}_1 = \tilde{c} \]
for a constant $2 \times n$ vector $\tilde{c}$.

Let $M_t = (1 - t)M_0 + tM_1$.

**Theorem:**
Suppose that $\tilde{p}_t$ is a solution to the barycentre equation with matrix $M_t$, that is, $M_t \tilde{p}_t = \tilde{c}$.
Then the animation function $a: t \to \tilde{p}_t$ is planar and smooth.

**Proof**
- It is planar by Tutte’s amazing Theorem.
- It is smooth by the Determinant Lemma.
The Bi-Weighted Barycentre Algorithm
Run VinciTest3
The bi-weighted barycentre algorithm with weak parameter $\delta$

Suppose that

- $G = (V, E)$ is a triconnected planar graph and $f$ is a face of $G$.
- Suppose that $G' = (V', E')$ is a subgraph of $G$.
- The $f$-contractions $G_f$ and $G'_f$ are connected.
- $\delta > 0$ is small.

We define weights $w_{uv} = \begin{cases} 1 & \text{if } (u, v) \in E' \\ \delta & \text{otherwise} \end{cases}$.

The **strong graph** $G' = (V', E')$ is the subgraph with all edge weights 1, plus the vertices and edges of $f$.

The **weak graph** $G'' = (V'', E'')$ is the subgraph with all edge weights $\delta$, plus the vertices and edges of $f$.
The *strong graph theorem*

and

The *weak graph theorem*
**Strong Graph Theorem**

Suppose that the $G_f$ is connected, and the strong graph is $G'$. Suppose that $G'_f$ is connected, $\tilde{p}_G$ is the bi-weighted barycentre drawing of $G$ restricted to the vertices of $G'$, with weak parameter $\delta$, and $\tilde{p}'$ is the barycentre drawing of $G'$. Then for every $\epsilon > 0$ there is a $\delta > 0$ such that $||\tilde{p}_G - \tilde{p}'|| < \epsilon$.

**Informally:**

If the strong edges are connected to the fixed set $f$, then the barycentre drawing of the whole graph is arbitrarily close to the barycentre drawing of the strong graph.
**Strong Graph Theorem**

Suppose that the $G_f$ is connected, and the strong graph is $G'$. Suppose that $G'_f$ is connected, $\hat{p}_{G'}$ is the bi-weighted barycentre drawing of $G$ restricted to the vertices of $G'$, with weak parameter $\delta$, and $\hat{p}'$ is the barycentre drawing of $G'$. Then for every $\epsilon > 0$ there is a $\delta > 0$ such that $\|\hat{p}_{G'} - \hat{p}'\| < \epsilon$. 

The drawing of the whole graph is similar to the drawing of the strong graph.
**Strong Graph Theorem**

Suppose that the $G_f$ is connected, and the strong graph is $G'$.

Suppose that $G'_f$ is connected, $\hat{\mathbf{p}}_{G'}$ is the bi-weighted barycentre drawing of $G$ restricted to the vertices of $G'$, with weak parameter $\delta$, and $\hat{\mathbf{p}}'$ is the barycentre drawing of $G'$.

Then for every $\epsilon > 0$ there is a $\delta > 0$ such that $\|\hat{\mathbf{p}}_{G'} - \hat{\mathbf{p}}'\| < \epsilon$.

Proof. Use two results:
1. The “Hoffman principle”: if $\|Mx - b\|$ is small then $x$ is close to a solution of $Mx = b$
2. The uniqueness lemma.
**Weak Graph Theorem**

Suppose that the $G_f$ is connected, and the weak graph is $G''$. Suppose that the strong graph $G'$ is connected but has no vertex in $f$. Let $\tilde{p}_{G'}$ be the barycentre drawing of the graph formed from $G$ by collapsing all of $G'$ to a single node. Suppose that $G''_f$ is connected, $\tilde{p}_{G''}$ is the bi-weighted barycentre drawing of $G$ restricted to the vertices of $G'$, with weak parameter $\delta$. Then for every $\epsilon > 0$ there is a $\delta > 0$ such that $\|\tilde{p}_{G'} - \tilde{p}_{G''}\| < \epsilon$.

**Informally:**
If the strong graph is connected but not connected to the fixed set $f$, then the barycentre drawing of the whole graph is arbitrarily close to the barycentre drawing where the strong graph is collapsed to a single node.
The drawing of the whole graph is similar to the drawing of the weak graph.
Clustered Planar Graphs
Clustered graphs
Run VinciTest3 (again)
Semantic Zoom
Focus + context display:

*Focus*: we want to zoom in to the neighbourhood of \( u \), to see it in detail.

*Context*: we can reduce detail for parts of G that are further from \( u \).

This can be achieved via geometric or semantic zooming.

*Geometric zoom*: “neighbourhood” and “distance” are defined geometrically

*Semantic zoom*: “neighbourhood” and “distance” are defined graph-theoretically

Semantic zoom is better!
VinciTest4
Weighted barycentre for semantic zoom for focus vertex $r$:

- For each vertex $u$, let $d(u) =$ graph theoretic distance between $r$ and $u$.

- For each edge $(u, v)$, $w_{uv} = k^d(u) + k^d(v)$

That is, edges become weaker as they get close to the focus node.
Finishing Up
The weighted barycentre algorithm can do many things

- Animation
- Clustered graph drawing
- Semantic zoom

But there is more …
there’s more ….

• Emphasize a specified spanning tree

• Emphasize a specified path (run VinciTest10)

• Adjust layout after adding/deleting and edge/vertex
The weighted barycentre algorithm can do many things

- Animation
- Clustered graph drawing
- Semantic zoom
- Emphasize a specified spanning tree
- Emphasize a specified path
- Adjust layout after adding/deleting and edge/vertex

But some things are still a problem …
Consider this graph $G_k$ with $k + 2$ nodes

In a barycentre drawing of graph $G_k$, nodes $u_{k-1}$ and $u_k$ are exponentially close together (distance less than $\frac{1}{2^k}$)
Run VinciTest7
**Important note:**

For practical graph drawing, vertex resolution is very important, because nontrivial labels need to be attached to nodes.
But by weighting the edges, we can get a better drawing:

\[ w(u_i, u_{i+1}) = 2^{k-i} \]
Run VinciTest8, then 9
**Open problem**

Given a triconnected planar graph $G = (V, E)$

Find weights on the edges so that the vertex resolution is good.

That is:

Given a graph $G = (V, E)$, find a diagonally dominant matrix $M$ such that

- $M_{uv} > 0$ if $u \neq v$ and $(u, v) \in E$
- $M_{uv} = 0$ if $u \neq v$ and $(u, v) \notin E$

and

$$\min_{u, v \in V} \|p(u) - p(v)\|$$

is maximal.