Balanced Schnyder woods for planar triangulations: an experimental study with applications to graph drawing and graph separators

sept. 18th,
Graph Drawing 2019

Luca Castelli Aleardi
Planar graphs are ubiquitous
(from computational geometry to computer graphics, geometric processing, …)

Real-world graphs are very regular and far from random or pathological cases

**regularity measure**: we use $d_6$, the proportion of degree 6 vertices

\[
\begin{align*}
  d_6 &\approx 0.11 \\
  d_6 &\approx 0.28 \\
  d_6 &\approx 0.50 \\
  d_6 &\approx 0.82 \\
  d_6 &\approx 0.99
\end{align*}
\]
Some facts about planar graphs
("As I have known them")

Kuratowski theorem (1930) (cfr Wagner’s theorem, 1937)
• $G$ contains neither $K_5$ nor $K_{3,3}$ as minors

Schnyder woods (‘89)
• planarity criterion via dimension of partial orders: $\dim(G) \leq 3$
• linear-time grid drawing, with $O(n) \times O(n)$ resolution

Thm (Tutte barycentric method, 1963)
Every 3-connected planar graph $G$ admits a convex representation in $R^2$.

Thm (Colin de Verdière, 1990) Colin de Verdiere invariant
(multiplicity of $\lambda_2$ eigenvalue of a generalized laplacian)
• $\mu(G) \leq 3$

Thm (Koebe-Andreev-Thurston)
Every planar graph with $n$ vertices is isomorphic to the intersection graph of $n$ disks in the plane.

$L_G[i,k] = \begin{cases} \deg(v_i) & \text{if } i = k \\ -A_G[i,j] & \text{otherwise} \end{cases}$
Schnyder woods
(quick overview)

Planar triangulations [Schnyder '90]

3-connected planar graphs [Felsner '01]

toroidal triangulations [Goncalves Lévêque, '14]
genus $g$ triangulations
[Castelli Aleardi Fusy Lewiner, '08]
A Schnyder wood of a (rooted) planar triangulation is partition of all inner edges into three sets $T_0$, $T_1$ and $T_2$, s.t.

i) edge are colored and oriented in such a way that each inner nodes has exactly one outgoing edge of each color

ii) colors and orientations around each inner node must respect the local Schnyder condition
Looking for "nice" Schnyder woods

**Counting Schnyder woods:** (there are an exponential number)

[Bonichon '05]

# Schnyder woods of triangulations of size \(n\): \(\approx 16^n\)

# planar triangulations of size \(n\): \(|\mathcal{T}_n| \approx 2^{3.2451}\)

[Felsner Zickfeld '08]

(count of Schnyder woods of a fixed triangulation)

\[
2.37^n \leq \max_{T \in \mathcal{T}_n} |SW(T)| \leq 3.56^n
\]

\(\mathcal{T}_n := \) class of planar triangulations of size \(n\)

\(SW(T) := \) set of all Schnyder woods of the triangulation \(T\)

---

**Egalitarian orientations:** (only for unconstrained orientations)

[Borradaile et al. '17]

"find an orientation s. t. no vertex is unfairly hit with too many arcs directed into it"

**Goal:** find an edge orientation that minimizes the lexicographic order of indegrees (or minimize maximum indegree)
Balanced Schnyder woods

A Schnyder wood is balanced if most vertices have a small defect.

**Definition**

\[
\delta(v) := \begin{cases} 
\max_{i \in \{0, 1, 2\}} \text{indeg}_i(v) - \min_{i \in \{0, 1, 2\}} \text{indeg}_i(v) & \text{if } \deg(v) = 3k \\
\max_{i \in \{0, 1, 2\}} \text{indeg}_i(v) - \min_{i \in \{0, 1, 2\}} \text{indeg}_i(v) - 1 & \text{otherwise}
\end{cases}
\]

**Vertex defect**

\[
\text{indeg}_i(v) := \# \text{incoming edges of color } i
\]

A Schnyder wood is **balanced** if most vertices have a small defect.

- perfectly balanced
- well balanced
- strongly unbalanced
Computing balanced Schnyder woods

Proportion of balanced vertices

with our heuristic
well balanced

minimal Schnyder wood

strongly unbalanced

(d_6 := proportion of degree 6 vertices)

Incremental vertex shelling (Brehm’s diploma thesis)

priority driven vertex conquest: remove first boundary vertices with higher number of ingoing edges

balancedSchnyderWood(T, (v_0, v_1, v_2), k)
B = \{v_0, v_1, v_2\} // initialization
T = new int[n] // priority array
Q_0 = \emptyset, Q_1 = \emptyset, ..., Q_{k-1} = \emptyset // queue initialization
Q_0.addLast(v_2)
while(|B| \neq \{v_0, v_1\}) {
  let M be the largest index s.t. Q_M \neq \emptyset
  let v = Q_M.poll()
  if(v \in B and v is free) {
    T[v_l] ++, T[v_r] ++ // increase priority
    Q_{max(k-1,T[v_l])}.addLast(v_l)
    Q_0.addLast(v_{j1}), ..., Q_0.addLast(v_{jt})
  }
}

balancedSchnyderWood(T, (v_0, v_1, v_2), k)
B = \{v_0, v_1, v_2\} // initialization
T = new int[n] // priority array
Q_0 = \emptyset, Q_1 = \emptyset, ..., Q_{k-1} = \emptyset // queue initialization
Q_0.addLast(v_2)
while(|B| \neq \{v_0, v_1\}) {
  let M be the largest index s.t. Q_M \neq \emptyset
  let v = Q_M.poll()
  if(v \in B and v is free) {
    T[v_l] ++, T[v_r] ++ // increase priority
    Q_{max(k-1,T[v_l])}.addLast(v_l)
    Q_0.addLast(v_{j1}), ..., Q_0.addLast(v_{jt})
  }
}
Layout quality for Schnyder drawings

(higher values are better) high values indicates more uniform edge length

\[ \delta_{avg} := \frac{1}{n} \sum_v \delta(v) \quad \text{(average vertex defect)} \]

\[ \epsilon_l := 1 - \left( \frac{1}{|E|} \sum_{e \in E} \frac{|l(e) - l_{avg}|}{\max(l_{avg}, l_{max} - l_{avg})} \right) \]

\[ l(e) := \text{edge length of } e \]
From Schnyder woods to cycle separators

(Fox-Epstein et al. 2016, Holzer et al. 2009)

**Def (small balanced cycle separators)**

A partition \((A, B, S)\) of \(V(G)\) such that:

- \(S\) defines a simple cycle
- \(A\) and \(B\) are balanced: \(|A| \leq \frac{2}{3}n, |B| \leq \frac{2}{3}n\)
- the separator is small: \(|S| \leq \sqrt{8m}\)

\[
S = P_i(v) \cup P_{i+2}(v) \cup \{v\} \quad \text{is minimized} \\
A = \text{Int}(R_i(v) \cup R_{i+2}(v)) \\
B = \text{Int}(R_{i+1}(v)) \\
|A| \leq \frac{2}{3}n \\
|B| \leq \frac{2}{3}n
\]

choose the best index \(i\) and vertex \(v\) s.t.

\(n = \) number of vertices \(m = \) number of edges

**Boundary size**

(tests are repeated with 200 random choices of the initial seed, the root face)

**Separator balance**

\(\delta_0 = 0.42\) \(\delta_{av} = 1.18\)

\(\delta_0 = 0.485\) \(\delta_{av} = 0.931\)

\(\delta_0 = 0.485\) \(\delta_{av} = 0.921\)

\(|S| = 0.58\sqrt{m}\) \(\delta_0 = 0.543\)

\(|S| = 1.32\sqrt{m}\) \(|S| = 1.68\sqrt{m}\)

\(|S| = 0.96\sqrt{m}\) \(\delta_{av} = 1.153\)

\(n = 2012\) \(\delta_0 = 0.543\)

\(diam=202\) \(|S| = 0.15\sqrt{m}\)

\(n = 20000\) \(\delta_{av} = 1.153\)

\(diam=168\) \(|S| = 0.15\sqrt{m}\)

\(n = 8268\) \(|S| = 0.15\sqrt{m}\)

\(diam=59\) \(|S| = 2.34\sqrt{m}\)

\(n = 20000\) \(diam=168\)

\(n = 8268\) \(diam=59\)

\(n = 20000\) \(diam=168\)

\(n = 8268\) \(diam=59\)
From Schnyder woods to cycle separators

How the separator quality depends on the balance

\( \delta_{\text{avg}} := \frac{1}{n} \sum_{v} \delta(v) \) (average vertex defect)

Boundary size

lower values are better

unbalanced

well balanced (our heuristic)
Evaluation of timing costs

average timings (over 100 executions)

Our performances (pure Java, on a core i7-5600 U, 2.60GHz, 1GB Ram):
We can process $\approx 1.43M - 1.92M$ vertices/seconds

Metis can process $\approx 0.7M$ vertices/seconds (C, on a Intel core i7-5600 2.60GHz)

Previous works can process $\approx 0.54M - 0.62M$ vertices/seconds (C/C++, on a Xeon X5650 2.67GHz)

Our datasets (several tens of real-world, random and synthetic graphs)

3d meshes from aim@shape and Thingi 10k
Random triangulations
Synthetic graphs
Thanks
Improving the balance (returning oriented cycles)

Real-world meshes

- our heuristic + post-processing
- our heuristic
- minimal Schnyder wood

Synthetic and random graphs

- our heuristic + post-processing
- our heuristic
- minimal Schnyder wood

Separator quality (lower values are better)

Layout quality (higher values are better)