Random Sorting Networks
Alexander Holroyd (UBC & Microsoft)

with: Omer Angel  Dan Romik
Balint Virag  Vadim Gorin
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>×</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>×</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>×</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
To get from $1 \ldots n$ to $n \ldots 1$ requires

$$N := \binom{n}{2}$$

nearest-neighbour swaps
E.g. $n=4$:

\[
\begin{array}{cccccc}
1 & 2 & 2 & 2 & 3 & 3 & 4 \\
2 & 1 & 3 & 2 & 4 & 3 & \times \\
3 & 3 & 1 & 4 & 2 & 2 & \\
4 & 4 & 4 & 1 & 1 & 1 & 1
\end{array}
\]

A Sorting Network =

any route from $1 \ldots n$ to $n \ldots 1$ in exactly

\[
N := \binom{n}{2}
\]

nearest-neighbour swaps
Theorem (Stanley 1984). 

\[
\text{# of } n\text{-particle sorting networks} = \frac{(n)!}{1^{n-1}3^{n-2}5^{n-3}\cdots(2n-3)^1}
\]

Uniform Sorting Network (USN): choose an n-particle sorting network uniformly at random.

E.g. n=3:

\[
P\left(\begin{array}{ccc}
\times & \times & \times \\
\times & \times & \times \\
\times & \times & \times 
\end{array}\right) = P\left(\begin{array}{ccc}
\times & \times & \times \\
\times & \times & \times \\
\times & \times & \times 
\end{array}\right) = \frac{1}{2}
\]
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>×</td>
<td>1</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>×</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
swap locations
Efficient simulation algorithm for USN...
Swap locations, n=100
Swap locations, n=2000
\begin{align*}
s_1 &= 1 \\
s_2 &= 2 \\
s_3 &= 3 \\
s_4 &= 1 \\
s_5 &= 2 \\
s_6 &= 1
\end{align*}
Theorem (Angel, H, Romik, Virag, 2007) For USN:

1. Sequence of swap locations $(s_1, ..., s_N)$ is stationary \( \forall n \)

2. Scaled first swap location
\[
\frac{s_1}{n} \xrightarrow{\text{dist}} \text{semicircle random variable} \quad \text{as } n \to \infty
\]

3. Scaled swap process
\[
\xrightarrow{\text{dist}} \text{semicircle } \times \text{Lebesgue} \quad \text{as } n \to \infty
\]

(Note: not true for all sorting networks, e.g. bubble sort)
Proof of stationarity:

1 \times 2 \_ \_ \_ 2 \_ \_ \_ 2 \times 3 \_ \_ \_ 3 \times 4
2 \times 1 \_ \_ \_ 3 \_ \_ \_ 3 \times 2 \times 4 \_ \_ \_ 3
3 \_ \_ \_ 3 \_ \_ \_ 1 \times 4 \_ \_ \_ 4 \times 2 \_ \_ \_ 2
4 \_ \_ \_ 4 \_ \_ \_ 4 \times 1 \_ \_ \_ 1 \_ \_ \_ 1 \_ \_ \_ 1
Proof of stationarity:

\[ \begin{array}{ccccccc}
\times & 2 & 2 & 2 & \times & 3 & 3 & \times & 4 \\
1 & \times & 3 & 3 & \times & 2 & 4 & \times & 3 \\
3 & \times & 1 & 4 & 4 & \times & 2 & \times & 2 \\
\text{-----} & 4 & 4 & \times & 1 & 1 & 1 & 1 & 1
\end{array} \]
Proof of stationarity:

\[
\begin{array}{cccc}
1 & 1 & 1 & 3 & 3 & 4 \\
2 & 3 & 3 & 1 & 4 & 3 \\
3 & 2 & 4 & 4 & 1 & 1 \\
4 & 4 & 2 & 2 & 2 & 2 \\
\end{array}
\]
Proof of stationarity:

1—1—1 × 3—3 × 4 ..... 
2 × 3—3 × 1 × 4 × 3 ..... 
3 × 2 × 4—4 × 1—1 
4—4 × 2—2—2—2 ×
Proof of stationarity:

\[
\begin{array}{cccc}
1 & 1 & 1 & 3 & 3 & 4 & 4 \\
\times & 3 & 3 & 1 & 4 & \times & 3 & 3 \\
3 & 2 & 4 & 4 & 1 & 1 & 2 \\
\times & 4 & 4 & 2 & 2 & 2 & 2 & \times & 1 \\
\end{array}
\]
Proof of stationarity:

\[(s_1, \ldots, s_N) \mapsto (s_2, \ldots, s_N, n-s_1)\] is a bijection from \{sorting networks\} to itself.

So for USN:

\[(s_2, \ldots, s_N) \overset{d}{=} (s_1, \ldots, s_{N-1})\]
Selected trajectories, n=2000
Scaled trajectory of particle $i$: $T_i: [0,1] \rightarrow [-1,1]$
Conjecture (AHRV) trajectories $\rightarrow$ random Sine curves:

$$\max_{i,t} |T_i(t) - A_i^n \sin(\pi t + \Theta_i^n)| \xrightarrow{Prob} 0$$

\[ \text{(random)} \quad \text{as } n \to \infty \]

Theorem (AHRV) scaled trajectories have subsequential limits which are Hölder($\frac{1}{2}$) with prob 1

\[ \text{as } n \to \infty \]
Half-time permutation matrix, n=2000
Conjecture (AHRV)
scaled permutation matrix at time $N/2$ \(d\) \(\rightarrow\) Archimedes measure

projection of surface area measure on sphere $S^2 \subset \mathbb{R}^3$ onto $\mathbb{R}^2$

(unique circularly symmetric measure with uniform linear projections;)

\[
\frac{dx \ dy}{2\pi \sqrt{1 - x^2 - y^2}} \quad \text{on } x^2+y^2<1
\]

scaled permutation matrix at time $tN$ \(d\) \(\rightarrow\) \(\begin{pmatrix} 1 & 0 \\ \cos \pi t & \sin \pi t \end{pmatrix}\) \(\circ\) Arch. meas.
Theorem (AHRV) scaled permutation matrix at time $tN$ is supported within a certain octagon with prob $\to 1$ as $n \to \infty$

$$(1-\frac{1}{2}\sqrt{3-\varepsilon})n$$
Tools in proofs:

1. Bijection (Edelman-Greene 1987)
   {sorting networks} $\leftrightarrow$ {standard staircase Young tableaux}

2. New result for limiting profile of random staircase Young tableau
   (from similar result for square tableaux, Pittel-Romik)
Why do we believe the conjectures?

The permutahedron: embedding of Cayley graph \((S_n, \text{n.n. swaps})\) in \(\mathbb{R}^n\):

\[
\sigma \mapsto \sigma^{-1} = (\sigma^{-1}(1), \ldots, \sigma^{-1}(n)) \in \mathbb{R}^n
\]

embeds in \((n-2)\)-sphere

1...n and n...1 are antipodal

n=4:

n=5
Conjecture (AHRV)

USN lies close to some great circle on the permutahedron with prob $\rightarrow 1$ as $n \rightarrow \infty$

e.g. $o(n)$ in $\| \|_\infty$

In fact simulations suggest more like $O(\sqrt{n})$!

(Again, not true for every sorting network, e.g. bubble sort)
Analogous (much easier) fact:

random shortest route

1st St & 1st Ave to nth St & nth Ave

≈ straight line

as \( n \to \infty \)
Theorem (AHRV) If a (non-random) sorting network lies close to some great circle, then:

- (o(n) in \(| |_\infty |\))

1. Trajectories \(\approx\) Sine curves

2. Half-time permutation \(\approx\) Archimedes measure

3. Swap process \(\approx\) semicircle \(\times\) Lebesgue

Simulation
Proof of Theorem:

close to great circle \(\Rightarrow\)

\(\approx\) Sine trajectories (up to a time change)

\(\Leftrightarrow\) \(\approx\) rotating disc picture

projections uniform \(\Rightarrow\) \(\approx\) Archimedes

swap rate uniform \(\Rightarrow\) rotation uniform

\(\Rightarrow\) no time change

calculation \(\Rightarrow\) semicircle law
Geometric Sorting Networks
Geometric Sorting Networks
Geometric Sorting Networks

Diagram showing a network with numbers 1, 2, 3, and 4 arranged in a specific order.
Geometric Sorting Networks
Geometric Sorting Networks
Geometric Sorting Networks
Geometric Sorting Networks
Goodman, Pollack (1980):
- all 4-item sorting networks are geometric
- but not all 5-item ones:
Goodman, Pollack (1980):
- all 4-item sorting networks are geometric
- but not all 5-item ones:
Great circle conjecture says: USN is \(" \approx \) geometric" as \( n \to \infty \)

but:

**Theorem** (Angel, H, Gorin, in prep)
\[
P(\text{USN is geometric}) \to 0 \quad \text{as} \quad n \to \infty
\]

**Proof**: in fact:
\[
P(\text{USN contains fixed swap pattern}) > 1 - e^{-cn}
\]

e.g. Goodman-Pollack counterexample
Subnetworks
Subnetworks
Subnetworks
Random Subnetworks

Take an $n$-item USN. Choose $m$ out of the $n$ items uniformly at random, indep. of USN.

**Great circle conjecture**

$m$ fixed, $n \to \infty$:

random $m$-out-of-$n$ network $\xrightarrow{d}$ geom. network of $m$ indep. points from Archimedes distn.
Conjecture (Warrington, 2009)

\[ P \left( \begin{array}{c} \text{random} \\ \text{4-out-of-n} \\ \text{network} \end{array} \in \{ \text{geom. networks} \\ \text{with 1 point in} \\ \text{hull(other 3)} \} \right) = \frac{1}{4} \]

for all \( n \)!
Theorem (Angel, H 2009) Warrington's conjecture is true.

Moreover, $\forall j < m \leq n$,

$$E(\text{random \# swaps in location } j \text{ in } m\text{-out-of-}n \text{ network})$$

does not depend on $n$

and $= \frac{(j - \frac{1}{2}) \cdots \frac{753}{222} \times (m - j - \frac{1}{2}) \cdots \frac{753}{222}}{(j - 1)! \times (m - j - 1)!}$

consistent with Archimededes distribution conjecture about $n \to \infty$ limit
Ingredients of proof

\[ P(s_1=k) = P(k-1 \text{ white balls added in first } n-2 \text{ in Polya urn}) \]

1st swap location in USN

Stationarity of USN
Exchangeability of Polya urn

\[ P(wwwbb) = P(wbwbw) \]

Compute

\[ P(\text{given space-time point in USN } \Rightarrow \text{ swap at location } j \text{ in subnetwork}) \]
Uniform swap model...

Angel, H, Romik 2008
Amir, Angel, Valko

...
**N.B.** Not every sorting network lies close to a great circle! E.g. typical network through

![Diagram](image.png)

(But this permutation is very unlikely).
Staircase Young diagram:

(E.g. $n=5$)

$n-1$

$\frac{1}{2}$}

$N$ cells
Standard staircase Young tableau:

```
1 2 4 8
3 5 6
7 10
9
```

Fill with $1, \ldots, N$ so each row/col increasing
Edelman-Greene algorithm:

1. Remove largest entry
Edelman-Greene algorithm:

1. Remove largest entry
Edelman-Greene algorithm:

2. Replace with larger of neighbours \( \uparrow \leftarrow \)
Edelman-Greene algorithm:

2. Replace with larger of neighbours

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Edelman-Greene algorithm:

1 2 4 8
3 7
5 6

2. Replace with larger of neighbours ↑ ← ...repeat
Edelman-Greene algorithm:

2. Replace with larger of neighbours $\uparrow \leftarrow$
...repeat
Edelman-Greene algorithm:

3. Add 0 in top corner
Edelman-Greene algorithm:

4. Increment
Edelman-Greene algorithm:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Repeat everything...
Edelman-Greene algorithm:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

5. Repeat everything...
Edelman-Greene algorithm:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Repeat everything...
Edelman-Greene algorithm:

5. Repeat everything...
Edelman-Greene algorithm:

5. Repeat everything...
**Edelman-Greene algorithm:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>4</th>
<th>6</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Repeat everything...
Edelman-Greene algorithm:

5. Repeat everything...
Edelman-Greene algorithm:

5. Repeat everything...
Edelman-Greene algorithm:

5. Repeat everything...
Edelman-Greene algorithm:
Edelman-Greene algorithm:

etc
Edelman-Greene Theorem:

After $N$ steps,
Edelman-Greene Theorem:

After N steps, get swap process of a sorting network!
Edelman-Greene Theorem:

After $N$ steps,
get swap process
of a sorting network

And this is a bijection!

And can explicitly describe inverse!
Theorem (Pittel-Romik): For a uniform random \( n \times n \) square tableau, \( \exists \) limiting shape with contours:

\[
h_\alpha(u) = \frac{2}{\pi} \left[ u \tan^{-1}(u/R) + \tan^{-1} R \right]
\]

where \( R = \frac{\sqrt{\alpha(2 - \alpha) - u^2}}{1 - \alpha} \)

Corollary (AHRV): For uniform random staircase tableau, limiting shape is half of this. (Proof uses Greene-Nijenhuis-Wilf Hook Walk)
**Proof of LLN** (swap process $\Rightarrow$ semic. $\times$ Leb.)

Swaps in space-time window $[an, bn] \times [0, \varepsilon N]$ come from entries $(1-\varepsilon)N$ in tableau:

$\# \approx \text{area under contour} \approx \text{semicircle}$
Proof of octagon and Holder bounds

Inverse Edelman-Greene bijection
($\approx$ RSK algorithm) $\Rightarrow$

# entries $< k$ in 1st row

$\geq$ longest $\Rightarrow$ subseq. of swaps by time $k$

$\geq$ furthest any particle moves up by time $k$

So can bound this using limit shape.
Angel,H,Virag (in preparation):

Process of first $k$ swaps in positions $cn \ldots cn+k$ \(\rightarrow\) random limit as $n \rightarrow \infty$

not depending on $c \in (0,1)$