Two grumpy giants and a baby

D. J. Bernstein
University of Illinois at Chicago

Tanja Lange
Technische Universiteit Eindhoven

Discrete-logarithm problems

Fix a prime ℓ.

Input: generator g
of group of order ℓ;
element h of same group.

Output: integer $k \in \mathbf{Z}/\ell$
such that $h = g^k$, where
group is written multiplicatively.
"$k = \log_g h$".

How difficult is computation of k?

Dependence on the group

Group $\mathbf{Z}/\ell$ under addition, represented in the usual way: DLP is very easy. Divide h by g modulo ℓ; time $\exp(O(\log\log\ell))$.

Order-ℓ subgroup of $(\mathbf{Z}/p)^*$ assuming prime $p = 2\ell + 1$: DLP is not so easy. Best known attacks: "index calculus" methods; time $\exp((\log\ell)^{1/3+o(1)})$.

Order-ℓ subgroup of $(\mathbf{Z}/p)^*$
for much larger p:
DLP is much more difficult.
Best known attacks:
"generic" attacks,
the focus of this talk.
Time $\exp((1/2 + o(1)) \log \ell)$.

Order-ℓ subgroup of properly
chosen elliptic-curve group:
DLP is again difficult.
Best known attacks:
"negating" variants of
generic attacks.
(See Schwabe talk, last CWG.)

Real-world importance

Apple, “iOS Security”, 2012.05: “Some files may need to be written while the device is locked. A good example of this is a mail attachment downloading in the background. This behavior is achieved by using asymmetric elliptic curve cryptography (ECDH over Curve25519).” Also used for “iCloud Backup”.

More examples: DNSCrypt; elliptic-curve signatures in German electronic passports.

Generic algorithms

Will focus on algorithms
that work for every
group of order ℓ.

Allowed operations:
neutral element 1;
multiplication $a, b \mapsto ab$.

Will measure algorithm cost
by counting # multiplications.

Success probability:
average over groups
and over algorithm randomness.

Each group element
computed by the algorithm
is trivially expressed as
$h^x g^y$ for known $(x, y) \in (\mathbf{Z}/\ell)^2$.

$1 = h^x g^y$ for $(x, y) = (0, 0)$.
$g = h^x g^y$ for $(x, y) = (0, 1)$.
$h = h^x g^y$ for $(x, y) = (1, 0)$.

If algorithm multiplies
$h^{x_1} g^{y_1}$ by $h^{x_2} g^{y_2}$
then it obtains $h^x g^y$ where
$(x, y) = (x_1, y_1) + (x_2, y_2)$.

Slopes

If $h^{x_1} g^{y_1} = h^{x_2} g^{y_2}$
and $(x_1, y_1) \neq (x_2, y_2)$
then $\log_g h$ is the negative
of the slope $(y_2 - y_1)/(x_2 - x_1)$.

(Impossible to have $x_1 = x_2$:
if $x_1 = x_2$ then $g^{y_1} = g^{y_2}$
so $y_1 = y_2$, contradiction.)

Algorithm immediately recognizes
collisions of group elements
by putting each $(h^x g^y, x, y)$
into, e.g., a red-black tree.
(Low memory? Parallel?
Distributed? Not in this talk.)

Baby-step-giant-step

(1971 Shanks)

Choose $n \geq 1$,
typically $n \approx \sqrt{\ell}$.

Points (x, y):
$n + 1$ "baby steps"
$(0,0), (0,1), (0,2), \ldots, (0,n)$;
$n + 1$ "giant steps"
$(1,0), (1,n), (1,2n), \ldots, (1,n^2)$.

Can use more giant steps.
Stop when $\log_g h$ is found.

Performance of BSGS

Slope $jn - i$ from $(0, i)$ to $(1, jn)$.

Covers slopes
$\{-n, \ldots, -1, 0, 1, 2, 3, \ldots, n^2\}$,
using $2n - 1$ multiplications.

Finds all discrete logarithms
if $\ell \leq n^2 + n + 1$.

Worst case with $n \approx \sqrt{\ell}$:
$(2 + o(1))\sqrt{\ell}$ multiplications.
(In fact always $< 2\sqrt{\ell}$.)

Average case with $n \approx \sqrt{\ell}$:
$(1.5 + o(1))\sqrt{\ell}$ multiplications.

Interleaving (2000 Pollard)

Improve average case to
$(4/3 + o(1))\sqrt{\ell}$ multiplications:
$(0, 0), (1, 0)$,
$(0, 1), (1, n)$,
$(0, 2), (1, 2n)$,
$(0, 3), (1, 3n)$,
$\vdots$
$(0, n), (1, n^2)$.

$4/3$ arises as $\int_0^1 (2x)^2\,dx$.

Interleaving (2000 Pollard)

Improve average case to
$(4/3 + o(1))\sqrt{\ell}$ multiplications:
$(0, 0), (1, 0)$,
$(0, 1), (1, n)$,
$(0, 2), (1, 2n)$,
$(0, 3), (1, 3n)$,
$\vdots$
$(0, n), (1, n^2)$.

$4/3$ arises as $\int_0^1 (2x)^2\,dx$.

Oops: Have to start with
$(0, n)$ as step towards $(1, n)$.
But this costs only $O(\log \ell)$.

Random self-reductions

Defender slows down BSGS by choosing discrete logs found as late as possible.

Random self-reductions

Defender slows down BSGS by choosing discrete logs found as late as possible.

Attacker compensates by applying a "worst-case-to-average-case reduction": compute $\log_g h$ as $\log_g(hg^r) - r$ for uniform random $r \in \mathbf{Z}/\ell$. Negligible extra cost.

Is BSGS optimal?

After m multiplications
have $m+3$ points in $(\mathbf{Z}/\ell)^2$.
Can hope for $(m+3)(m+2)/2$
different slopes in $\mathbf{Z}/\ell$.

Is BSGS optimal?

After m multiplications have $m+3$ points in $(\mathbf{Z}/\ell)^2$. Can hope for $(m+3)(m+2)/2$ different slopes in $\mathbf{Z}/\ell$.

1994 Nechaev, 1997 Shoup: proof that generic algorithms have success probability $O(m^2/\ell)$. Proof actually gives $\le ((m+3)(m+2)/2+1)/\ell$.

Is BSGS optimal?

After m multiplications
have $m + 3$ points in $(\mathbf{Z}/\ell)^2$.
Can hope for $(m + 3)(m + 2)/2$
different slopes in $\mathbf{Z}/\ell$.

1994 Nechaev, 1997 Shoup:
proof that generic algorithms
have success probability $O(m^2/\ell)$.
Proof actually gives
$\le ((m + 3)(m + 2)/2 + 1)/\ell$.

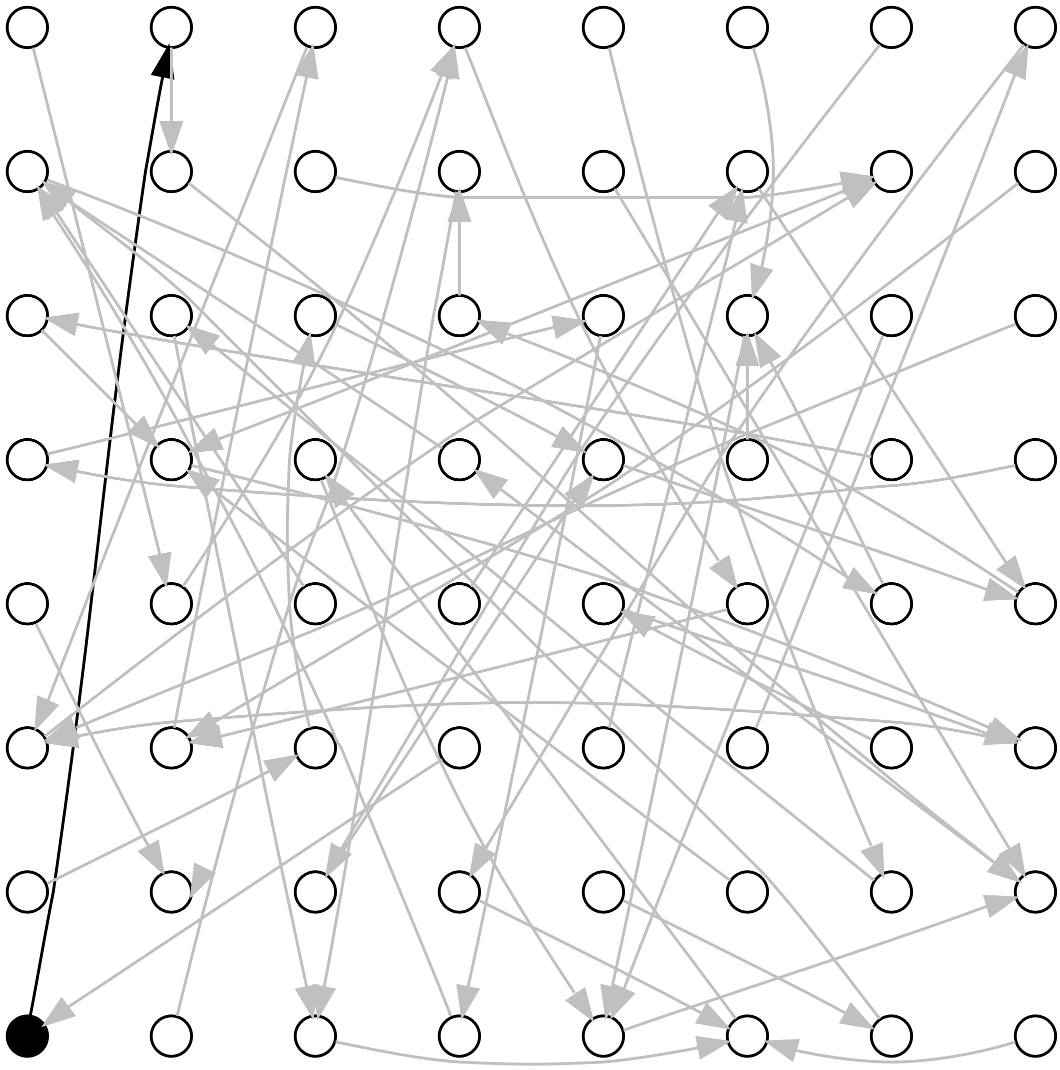
BSGS: at best $\approx m^2/4$ slopes,
taking $n \approx m/2$.
Factor of 2 away from the bound.

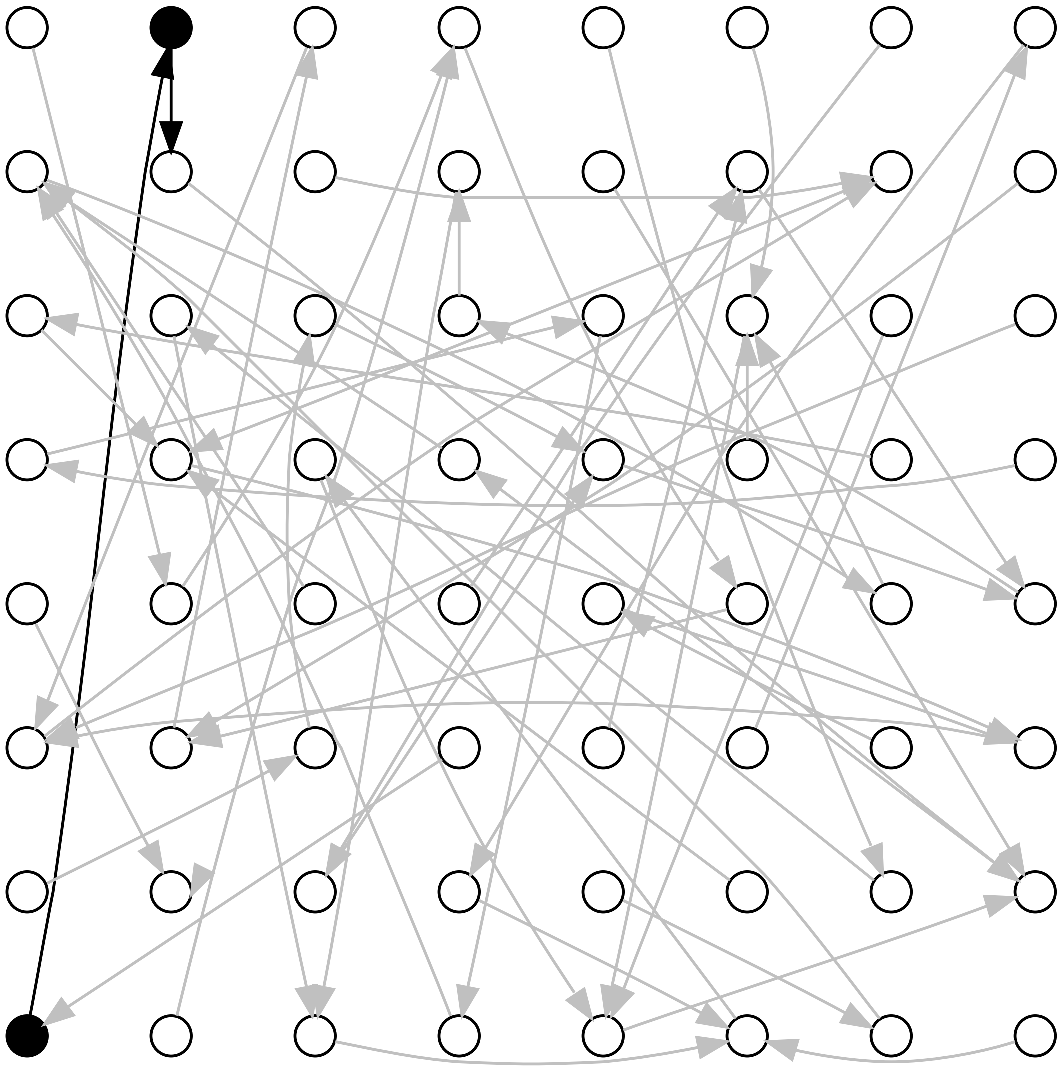
The rho method

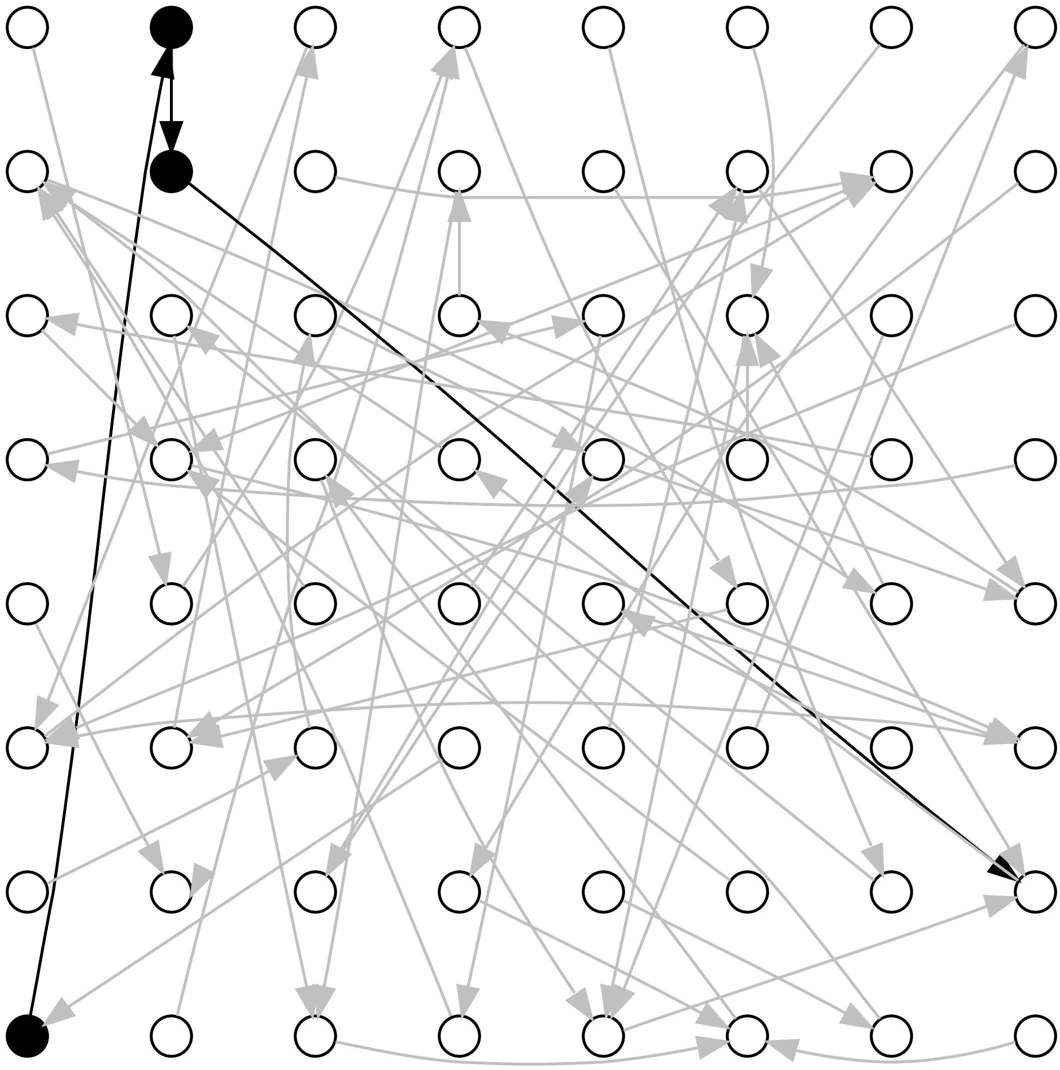
(1978 Pollard, $r = 3$ "mixed"; many subsequent variants)

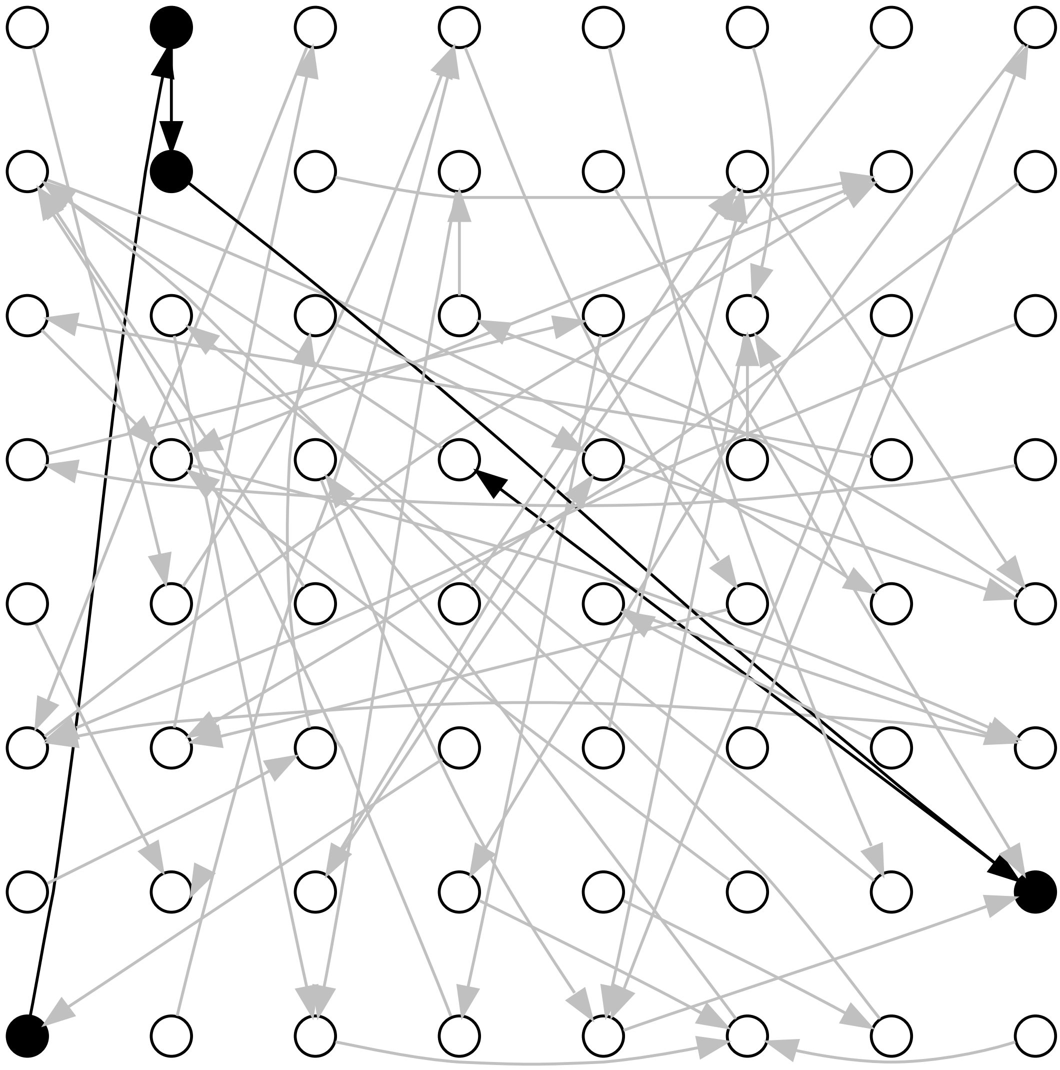
Initial computation: r uniform random "steps" $(s_1, t_1), \ldots, (s_r, t_r) \in (\mathbf{Z}/\ell)^2$. $O(r \log \ell)$ multiplications; negligible if r is small.

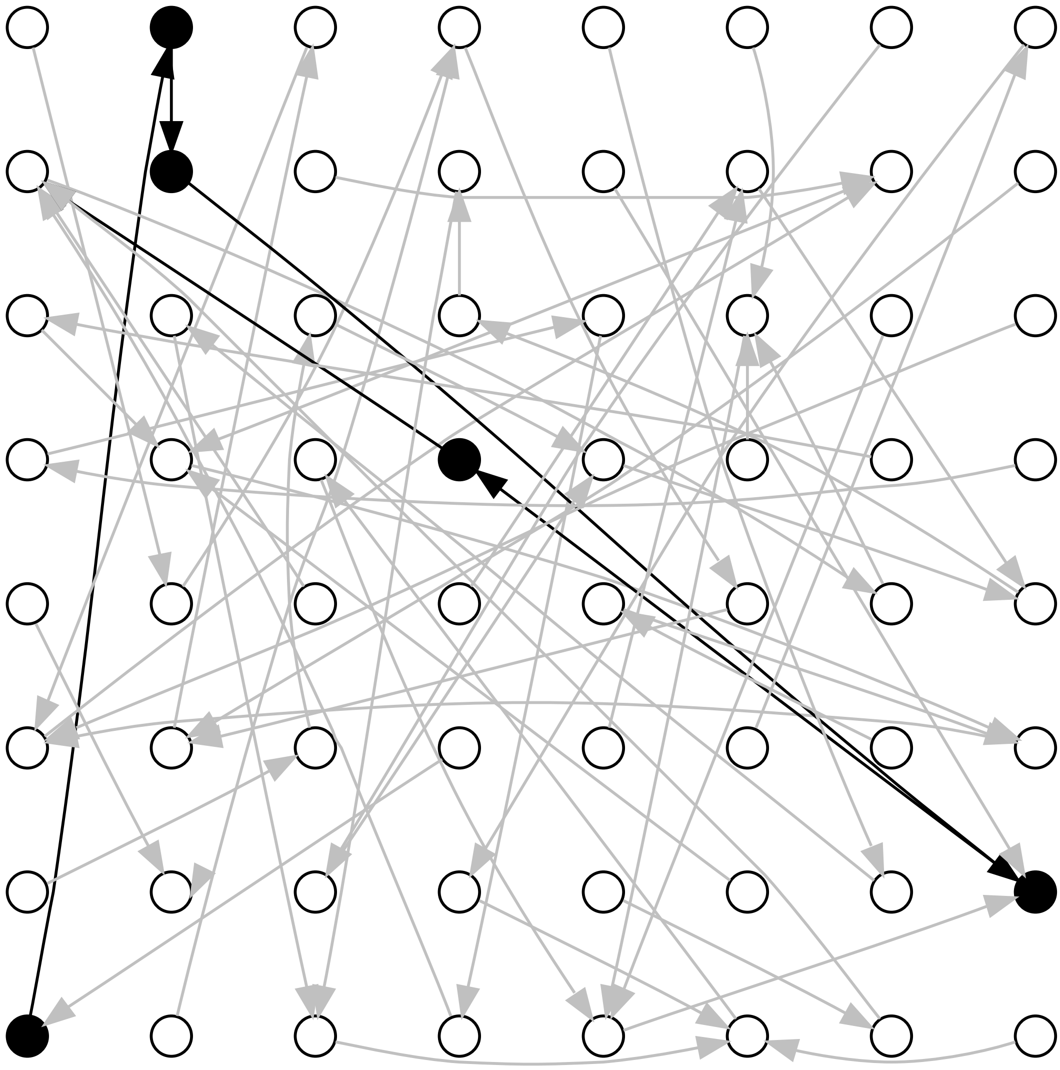
The "walk": Starting from $(x_i, y_i) \in (\mathbf{Z}/\ell)^2$ compute $(x_{i+1}, y_{i+1}) = (x_i, y_i) + (s_j, t_j)$ where $j \in \{1, \ldots, r\}$ is a hash of $h^{x_i} g^{y_i}$.

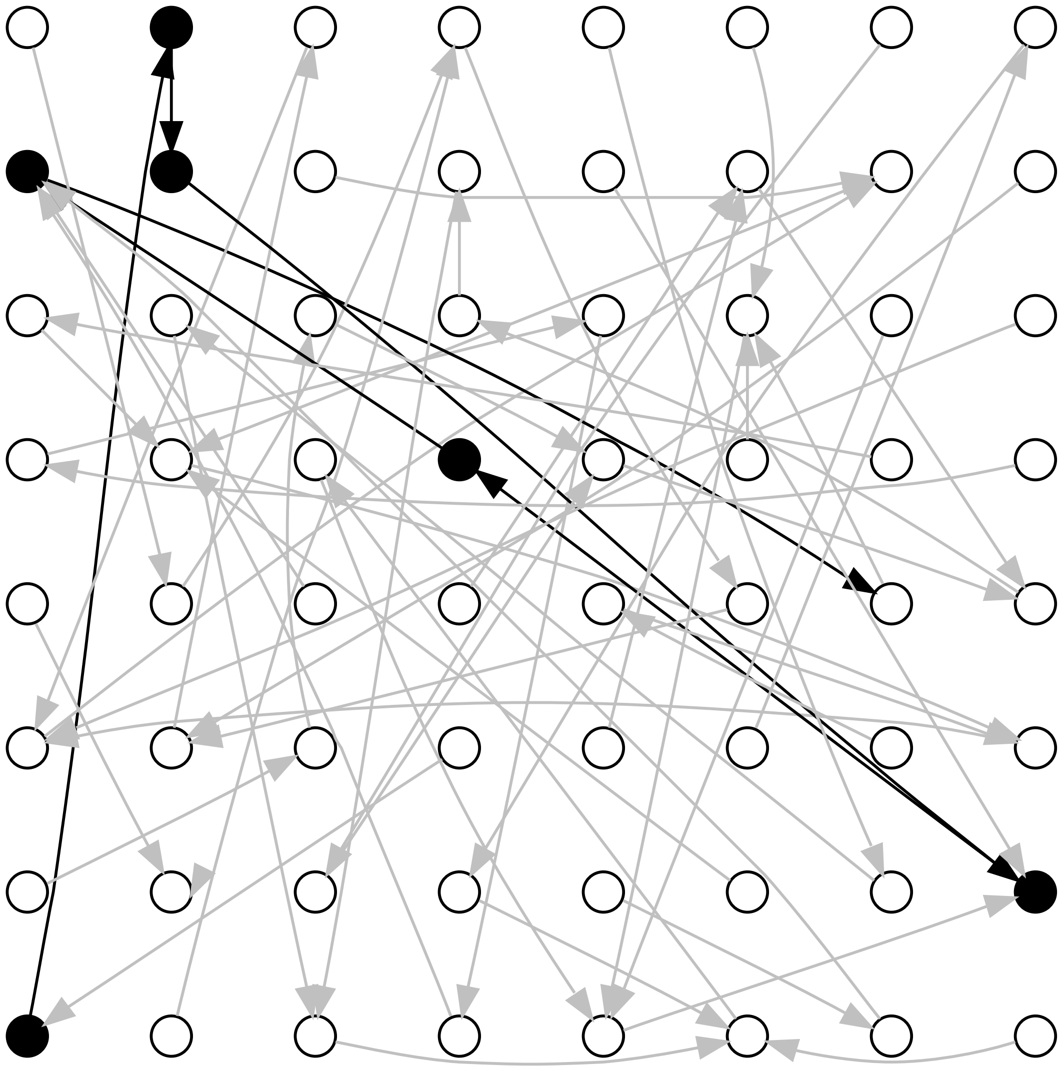


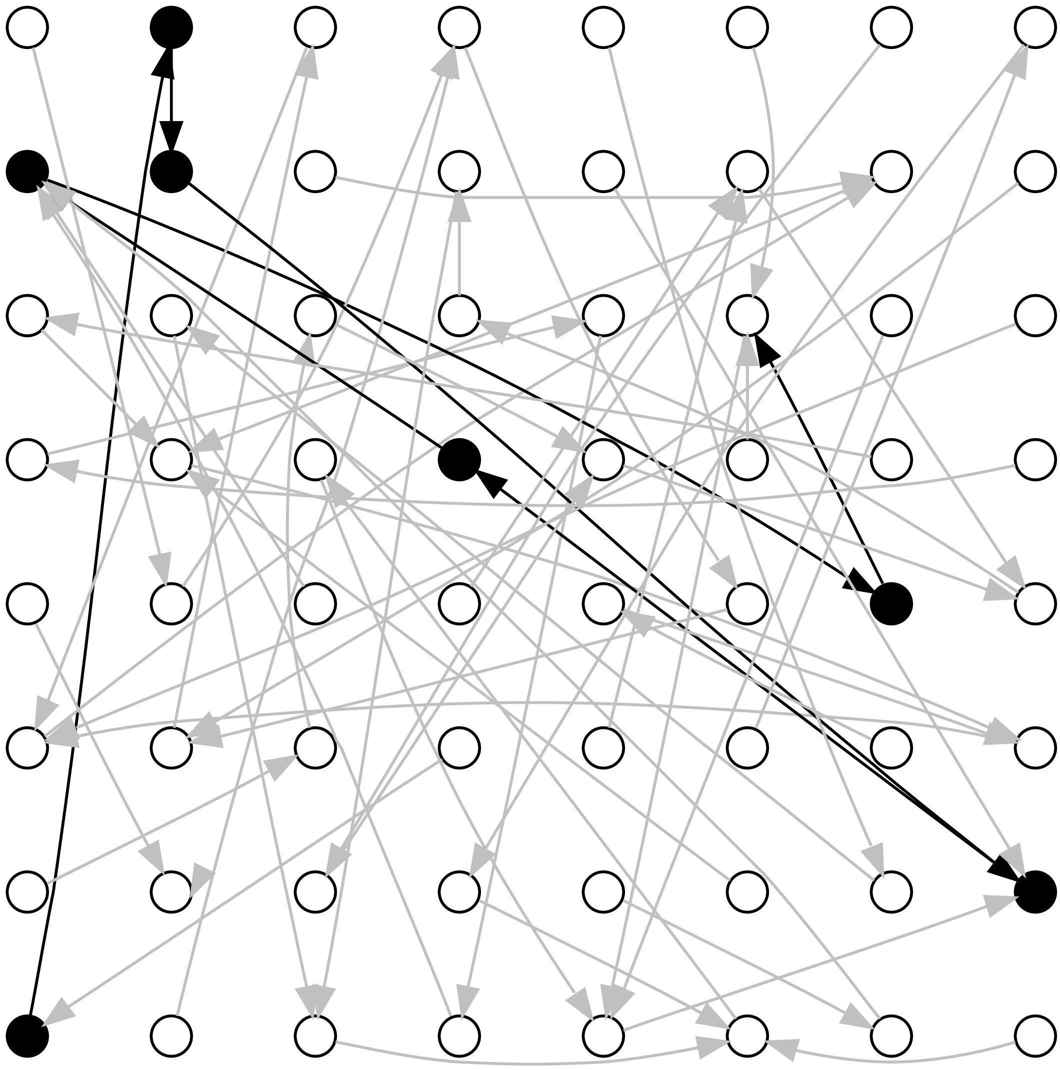


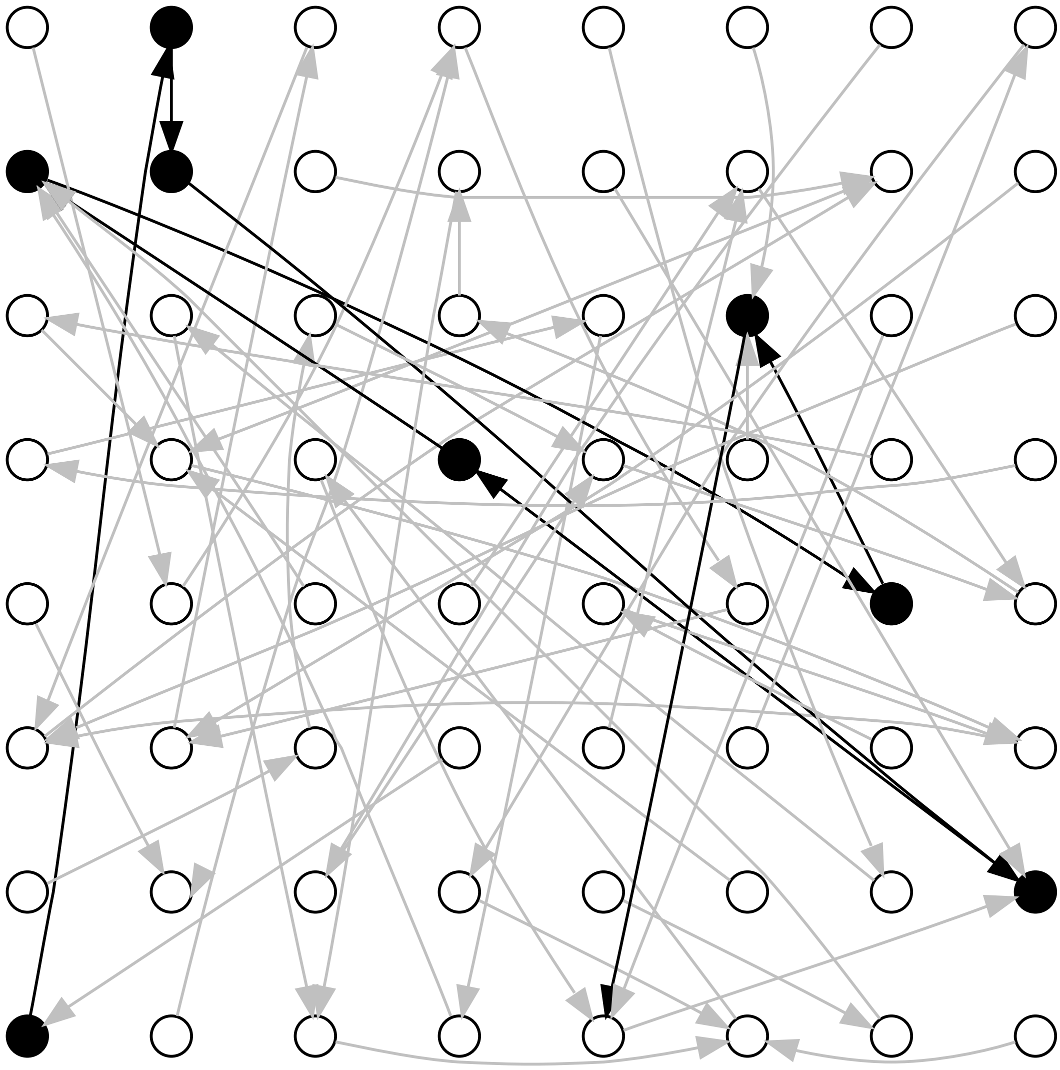


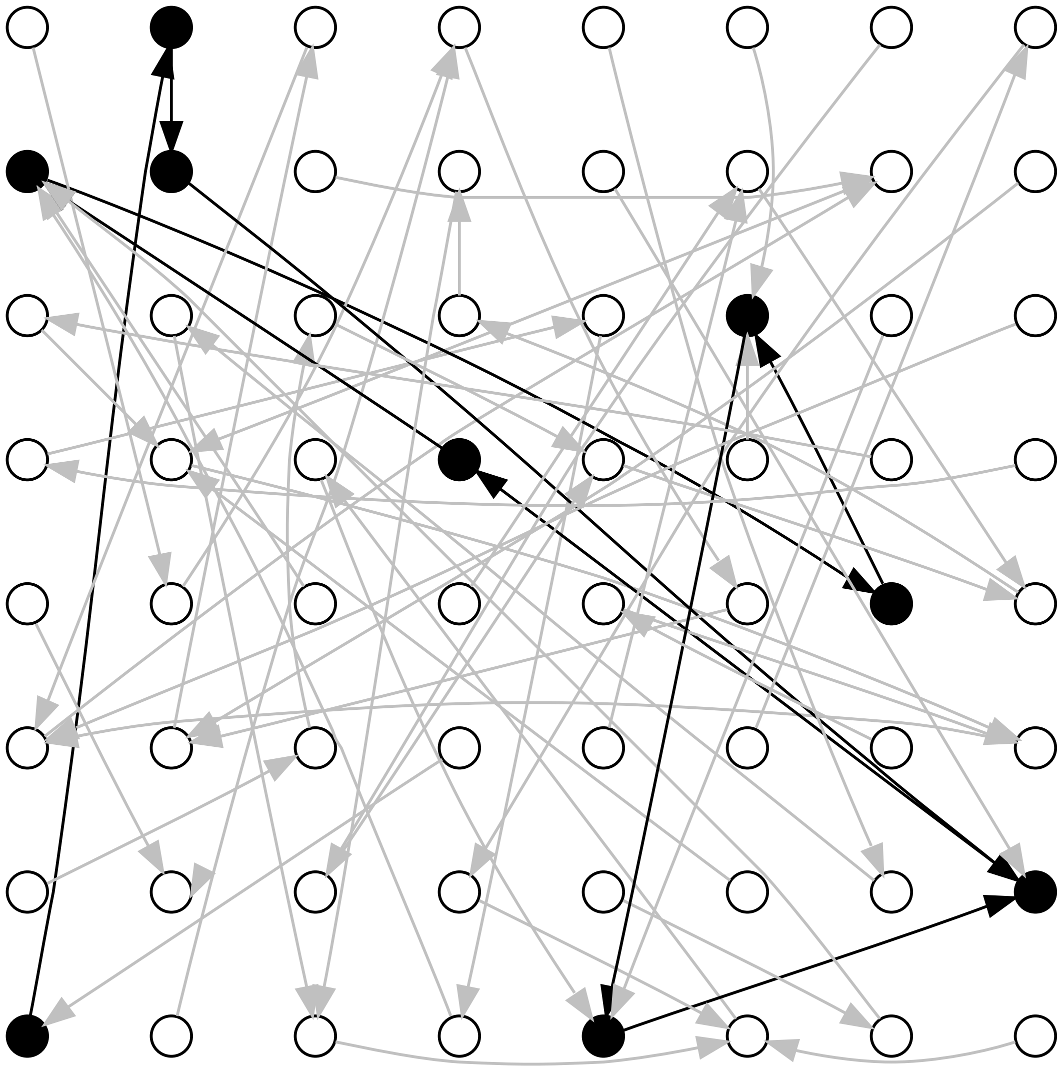


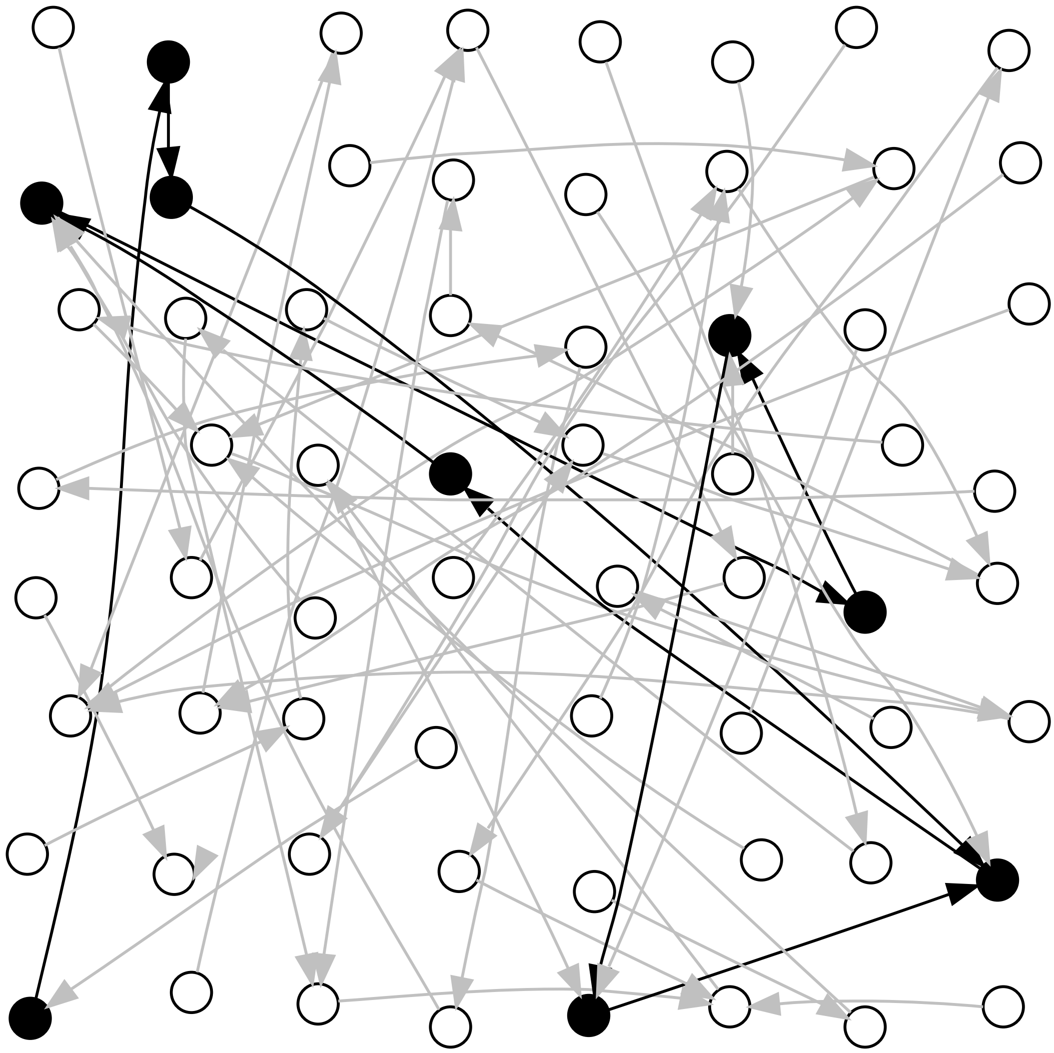


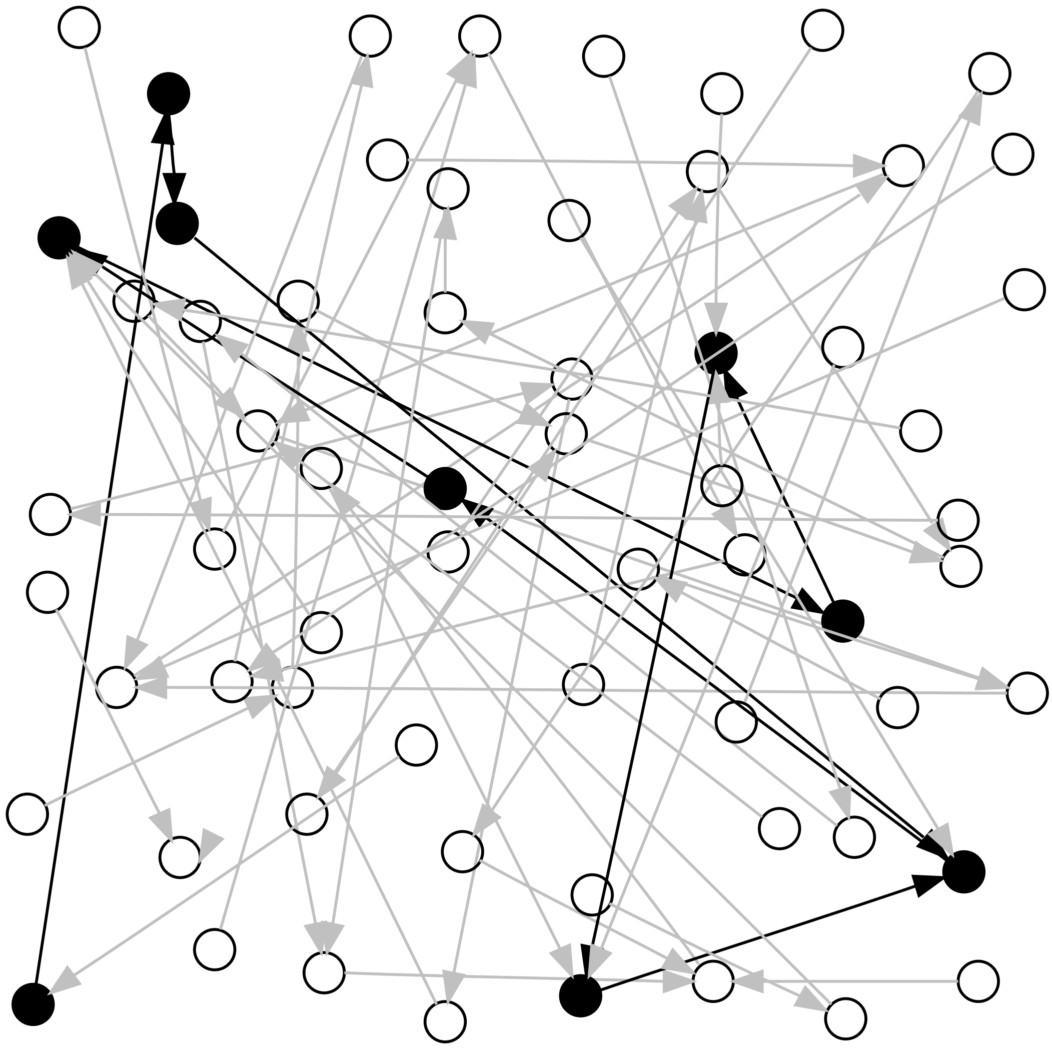


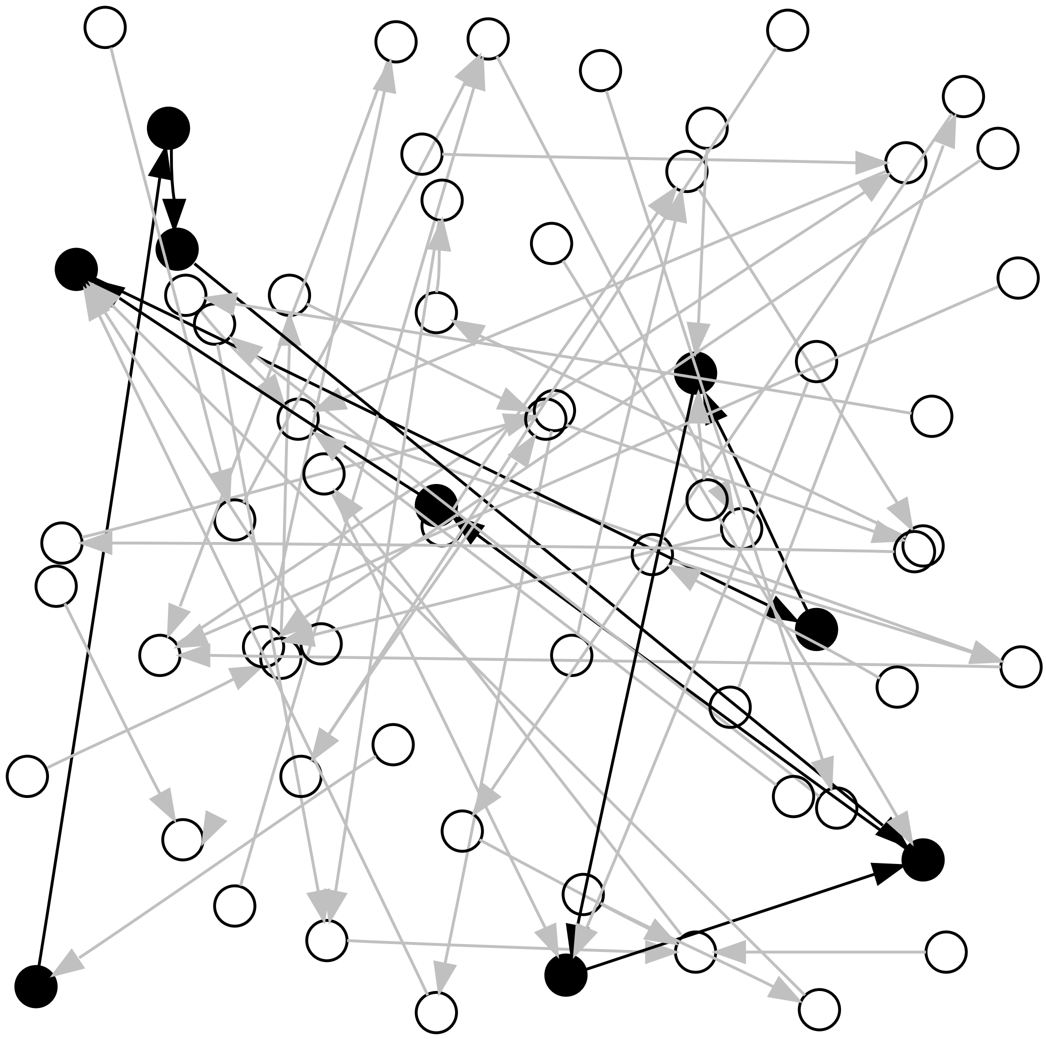


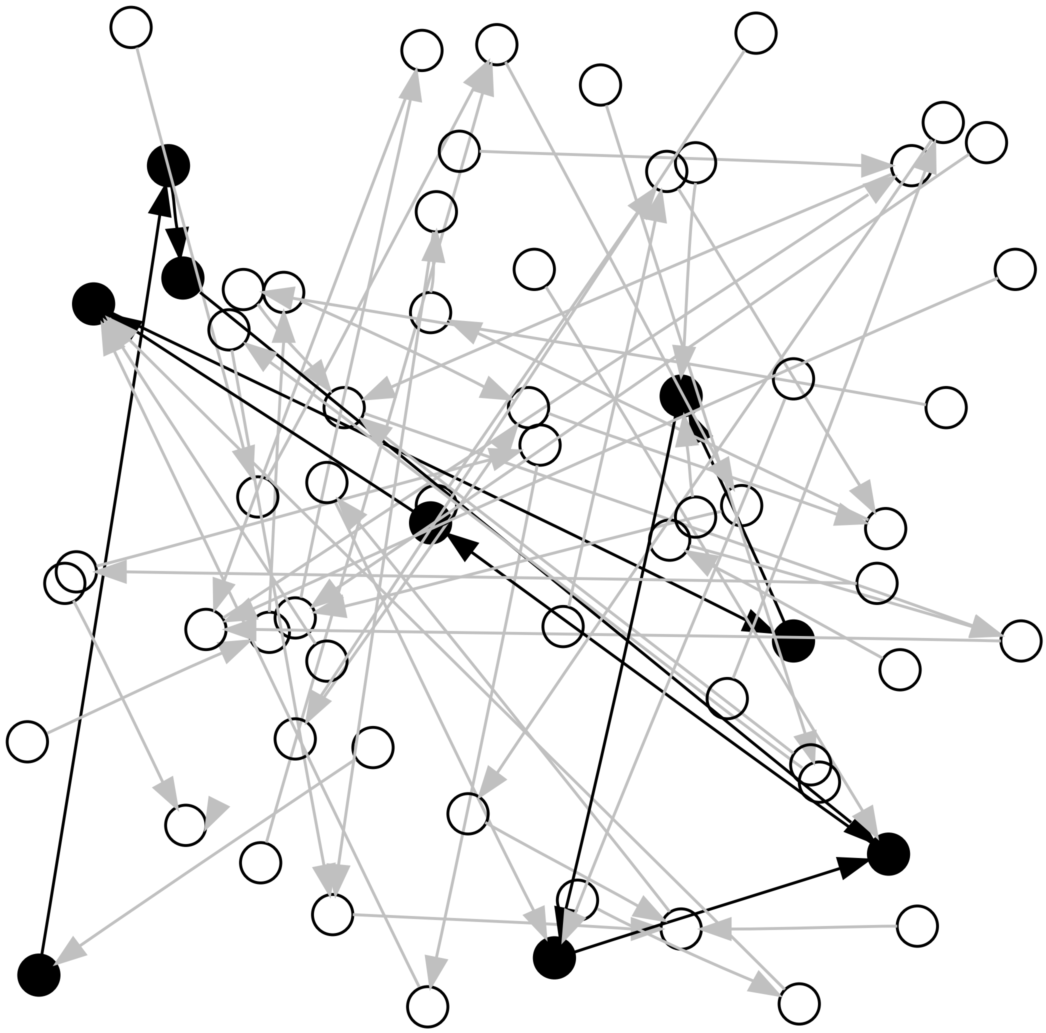


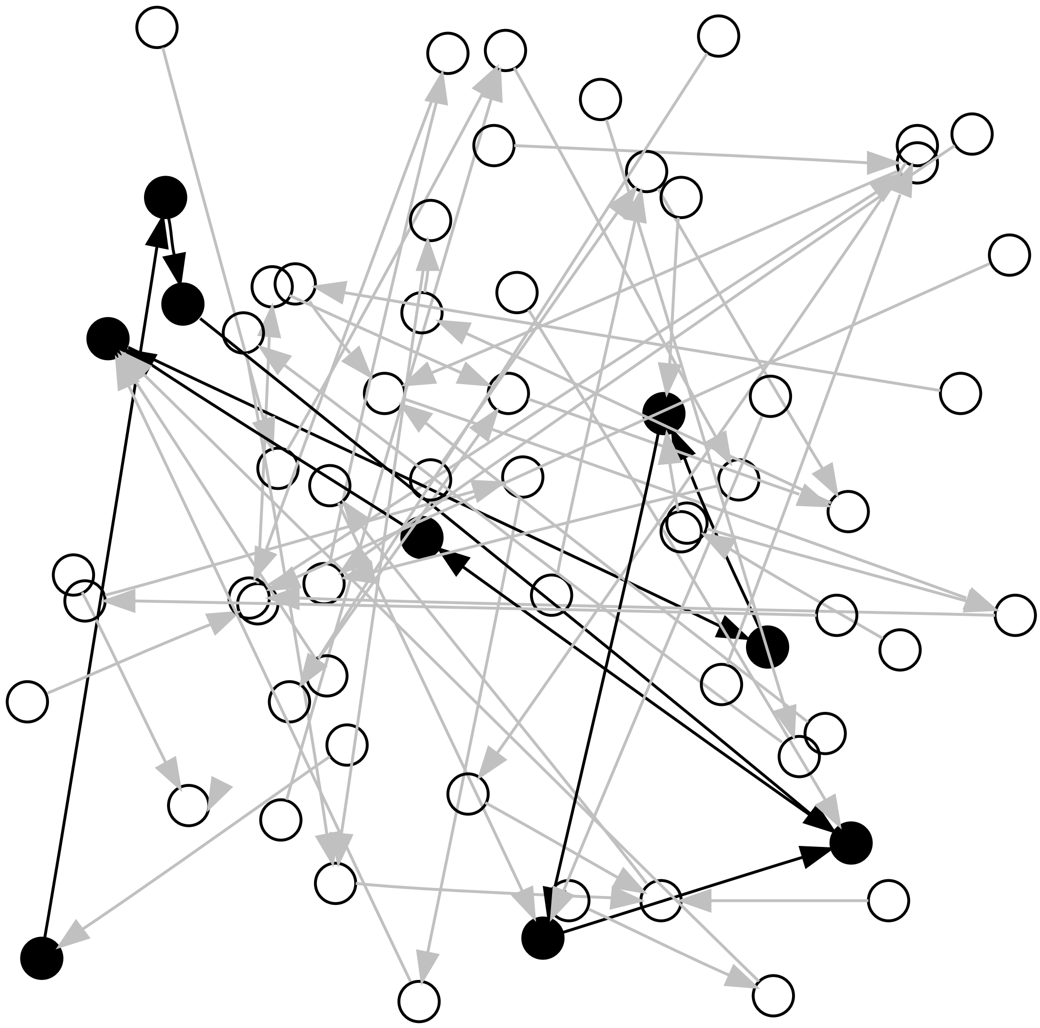


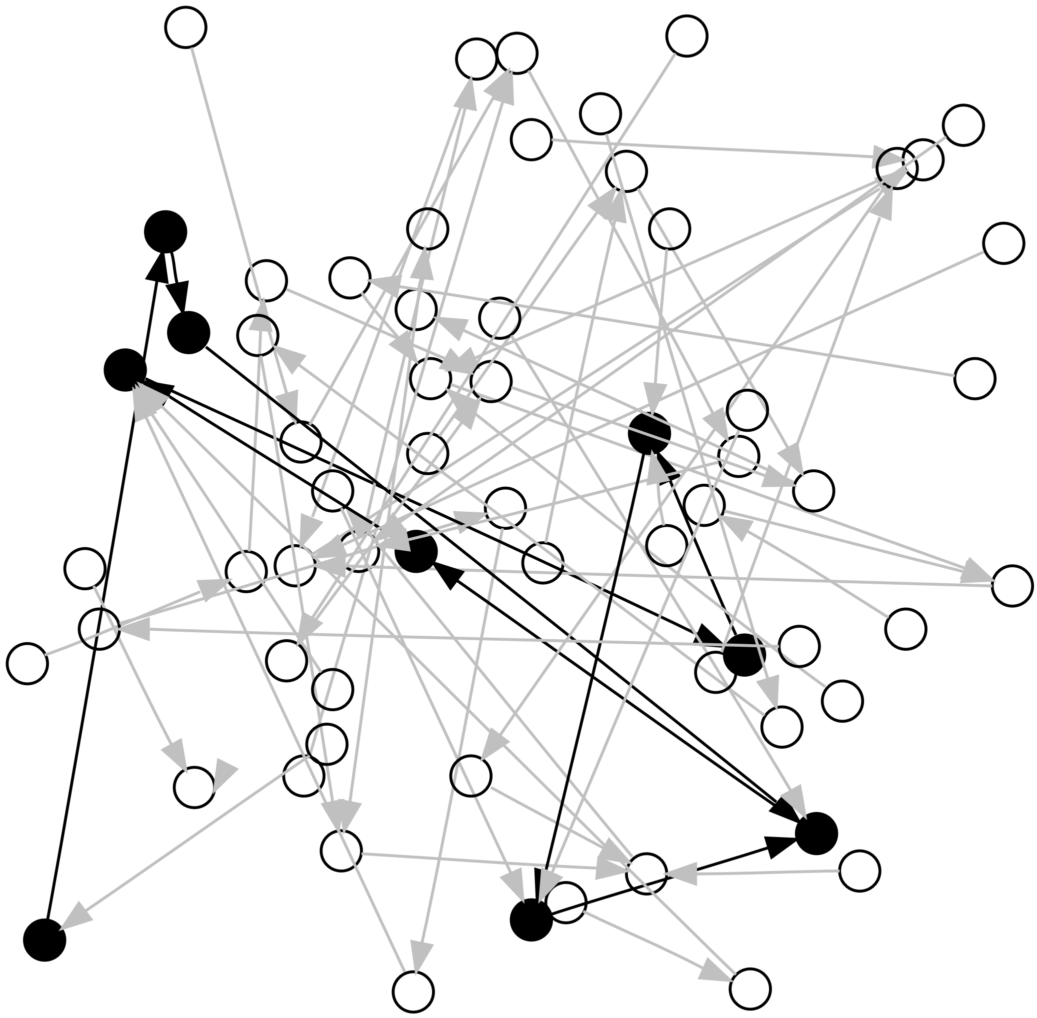


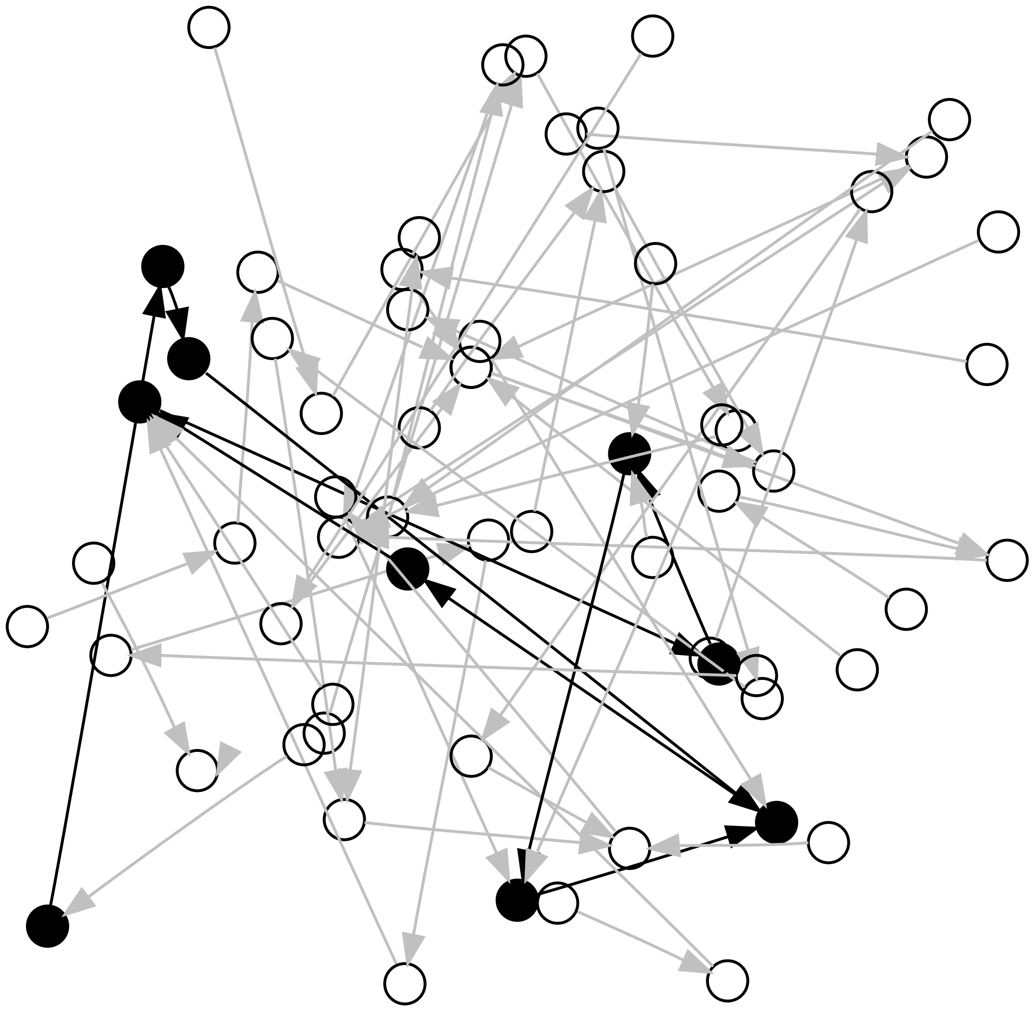


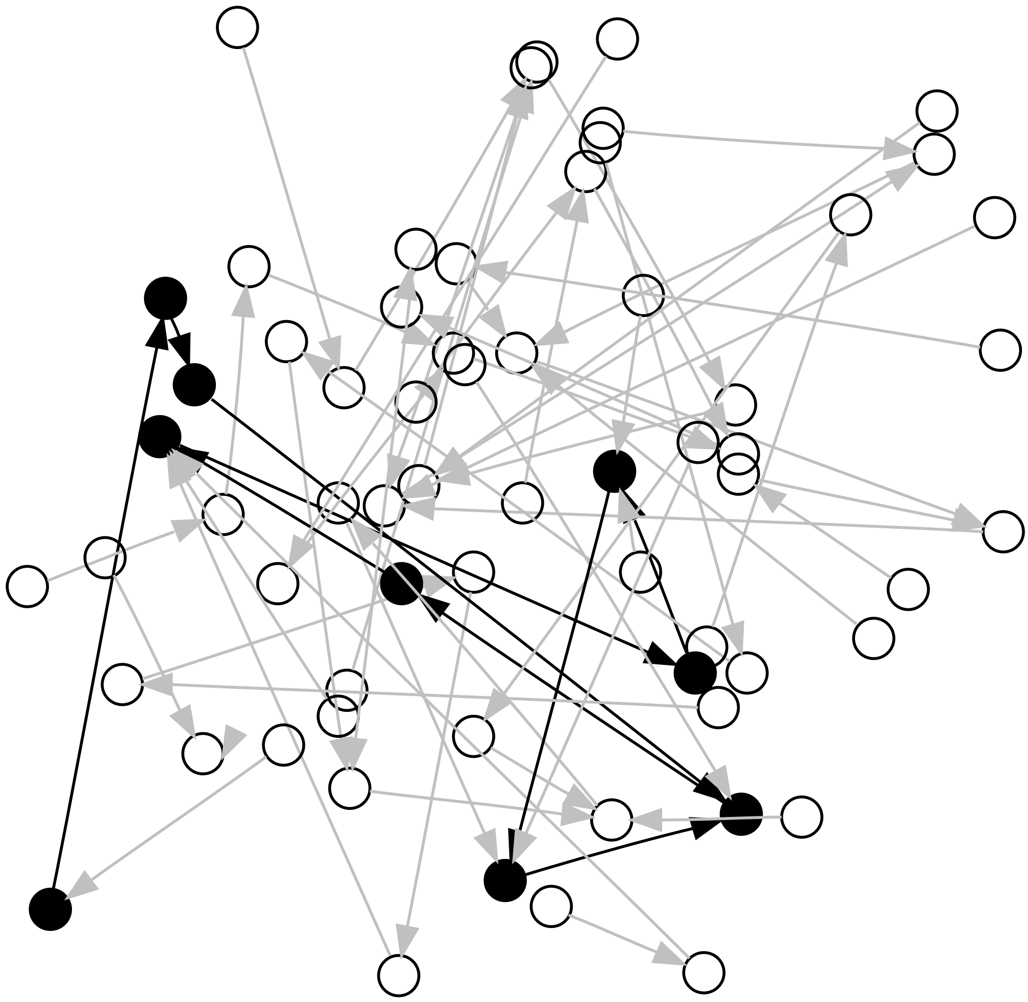


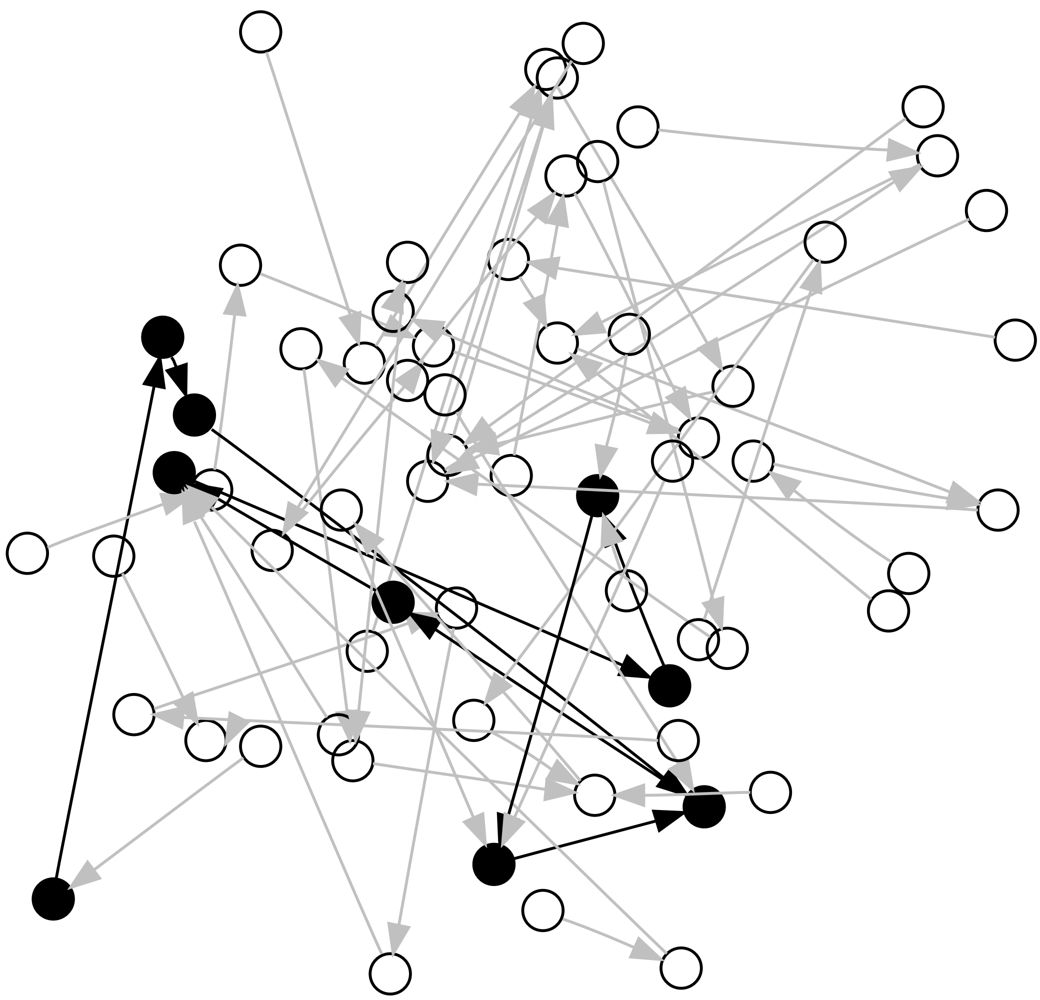


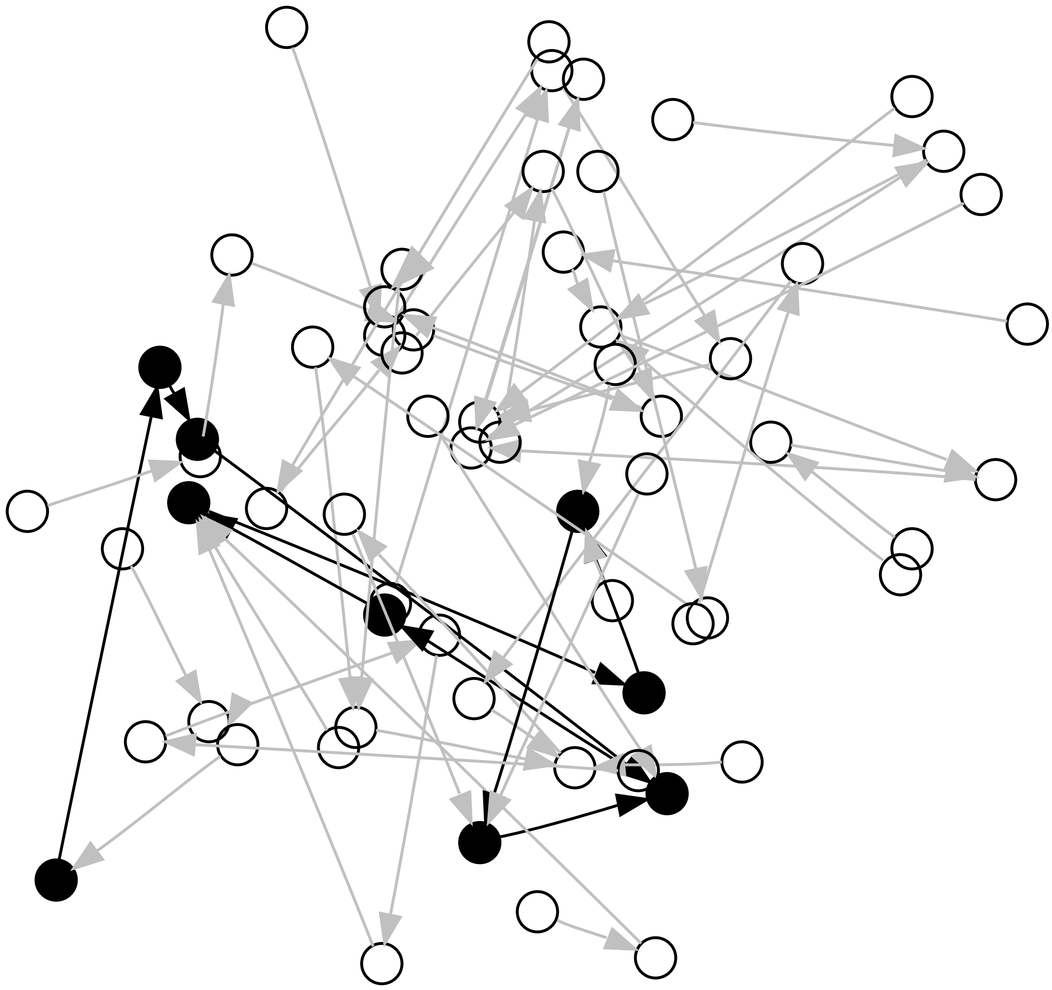


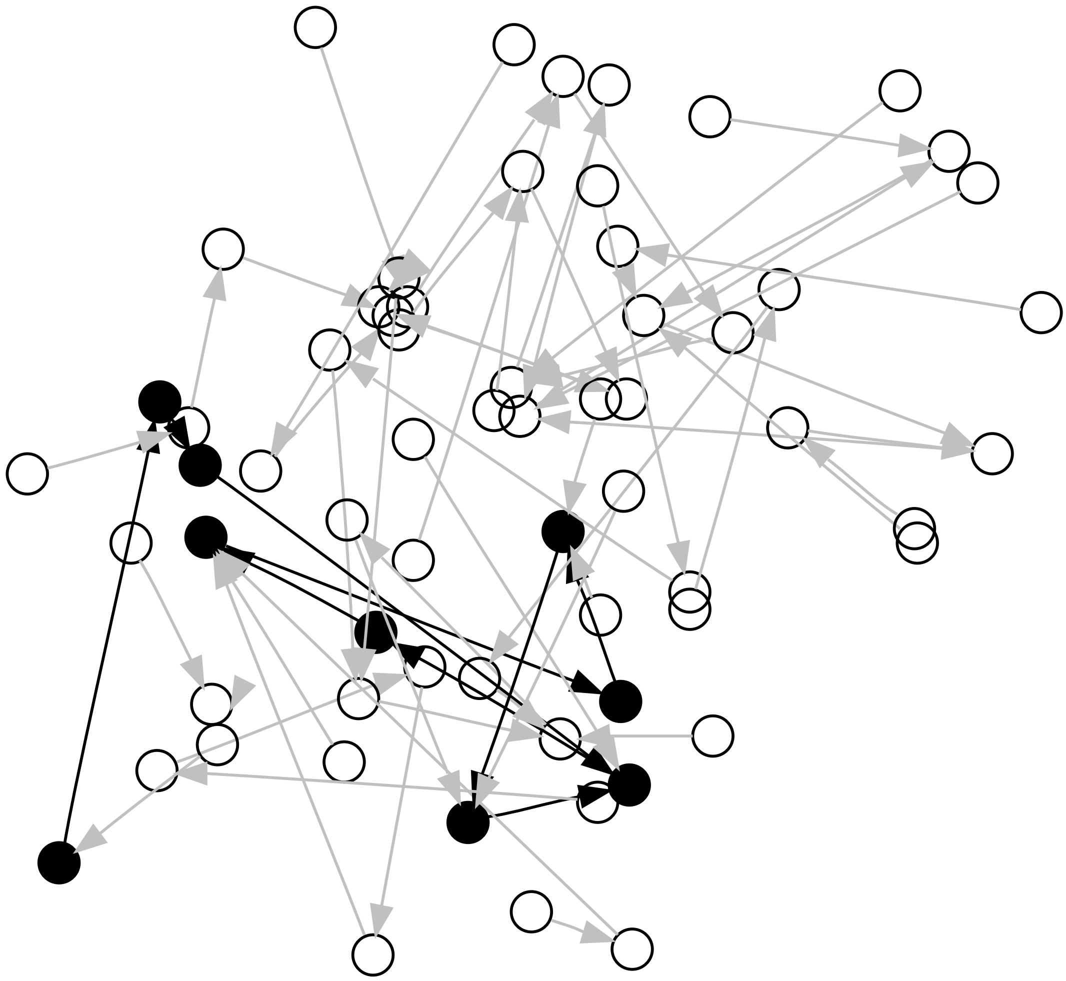


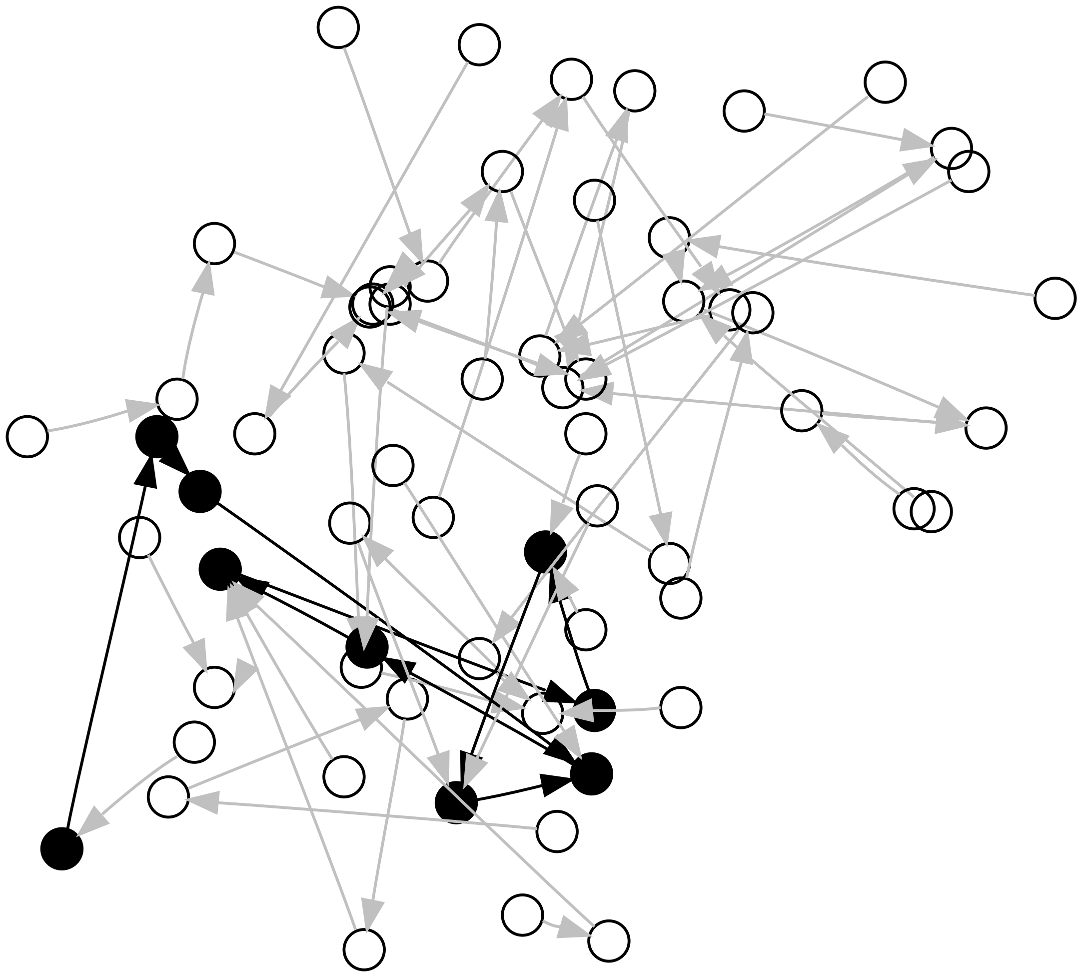


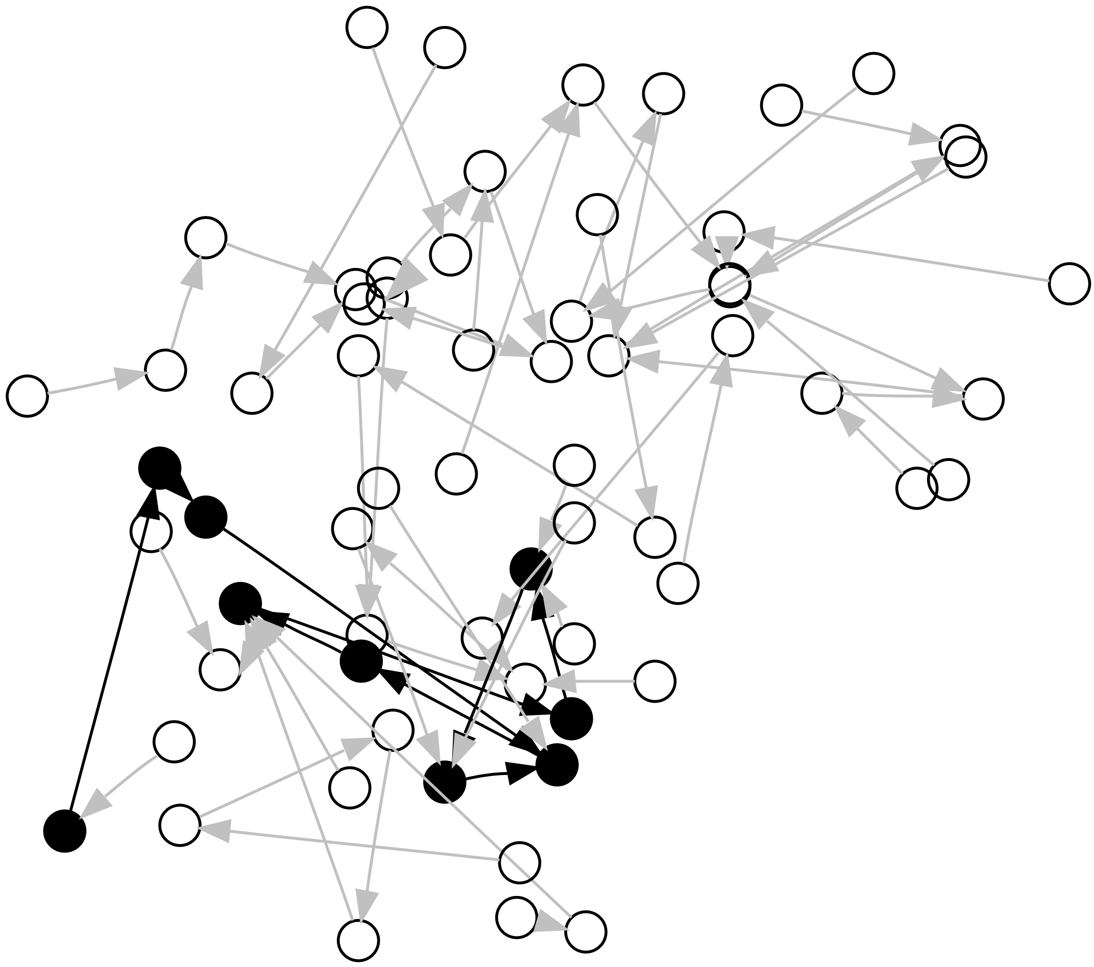


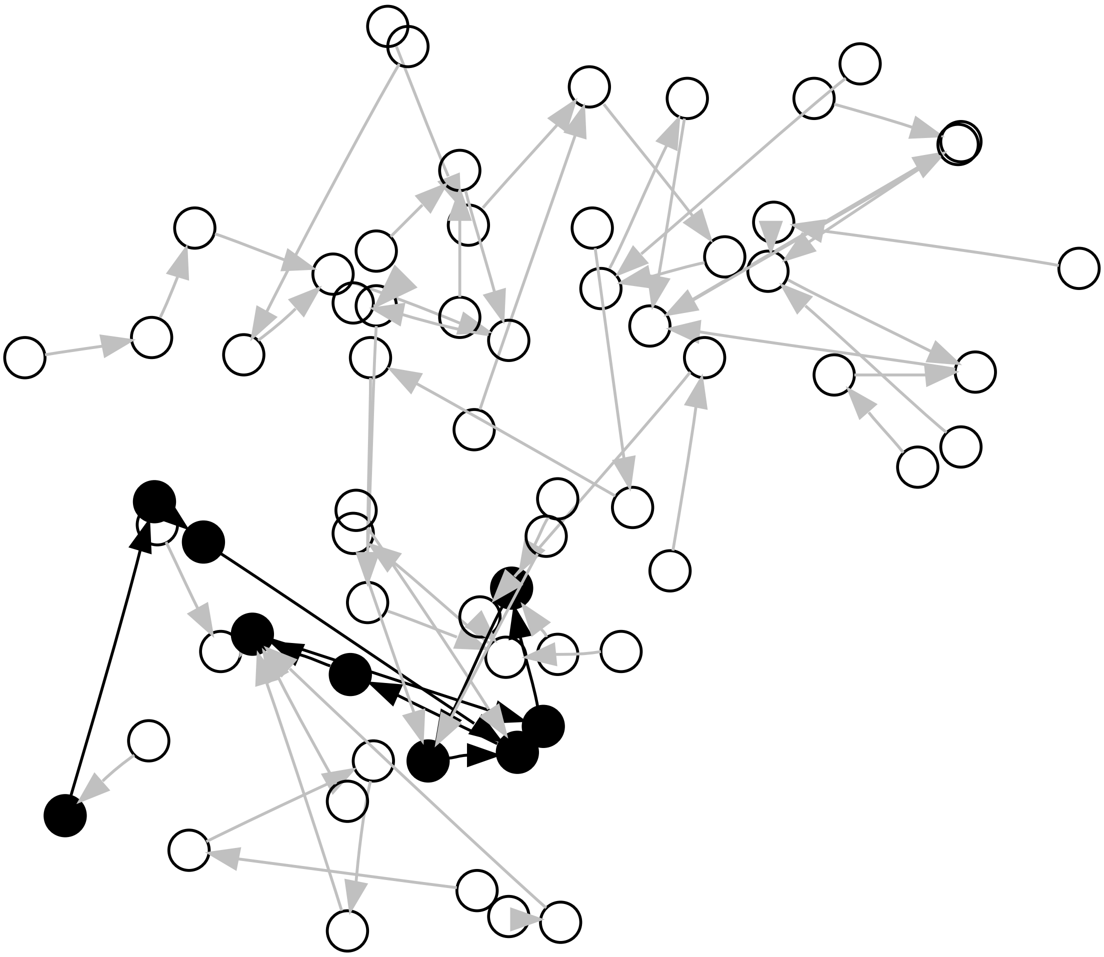


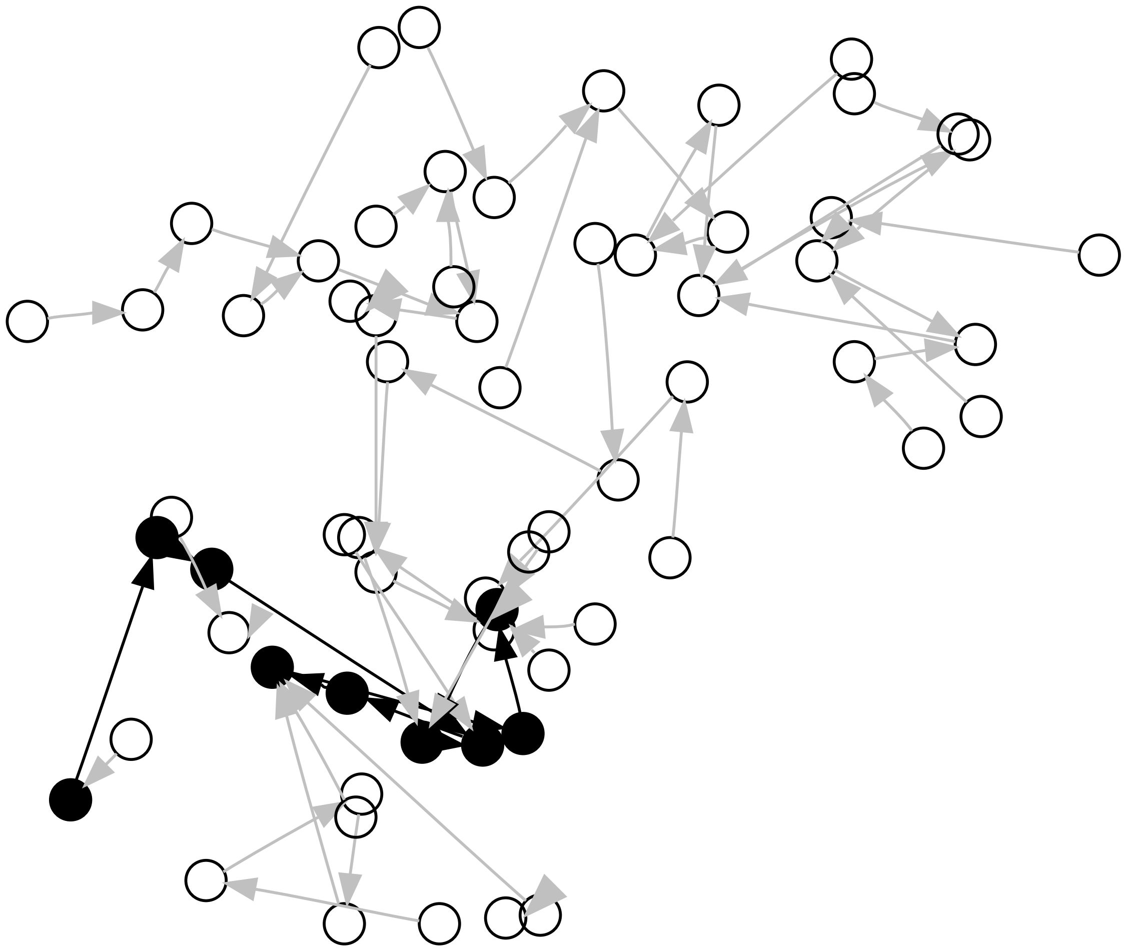


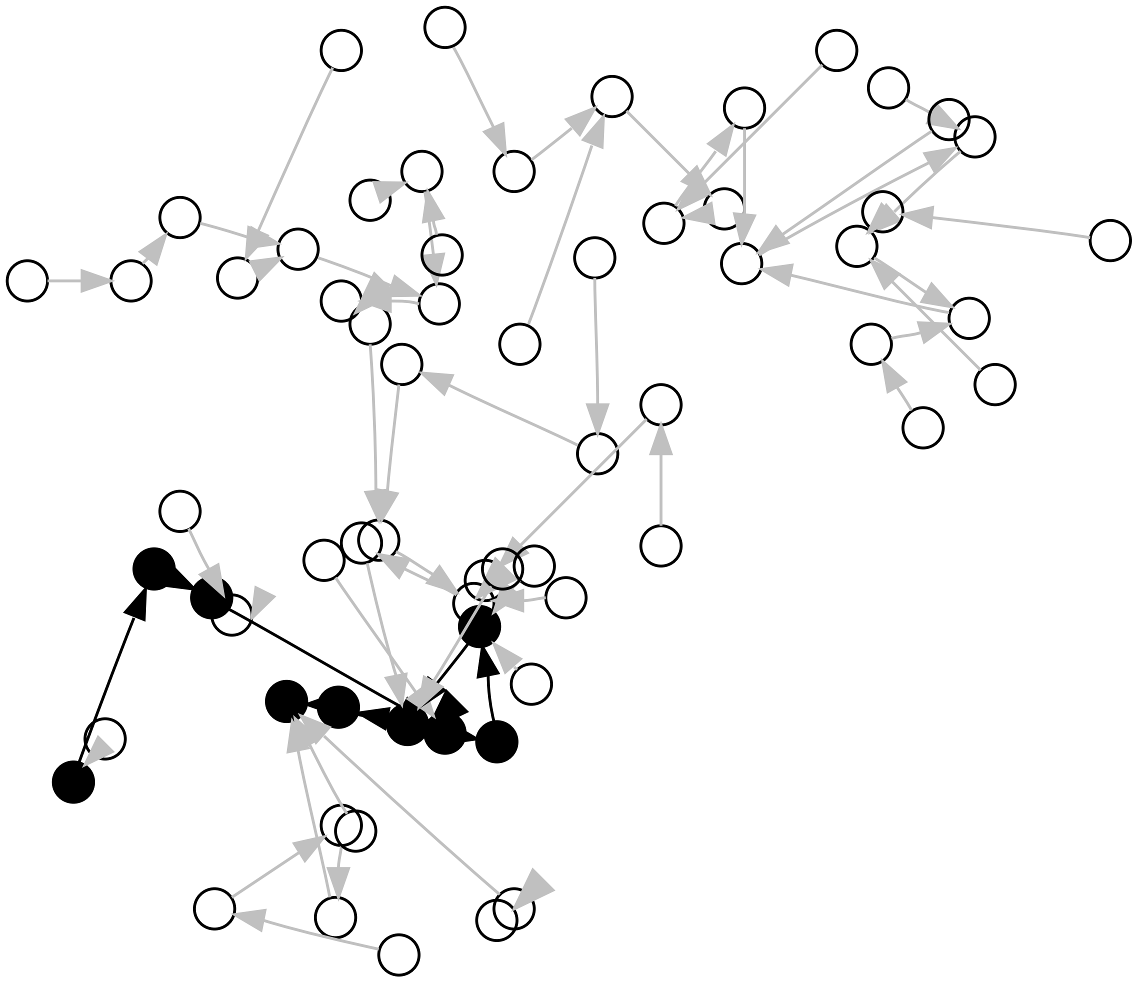


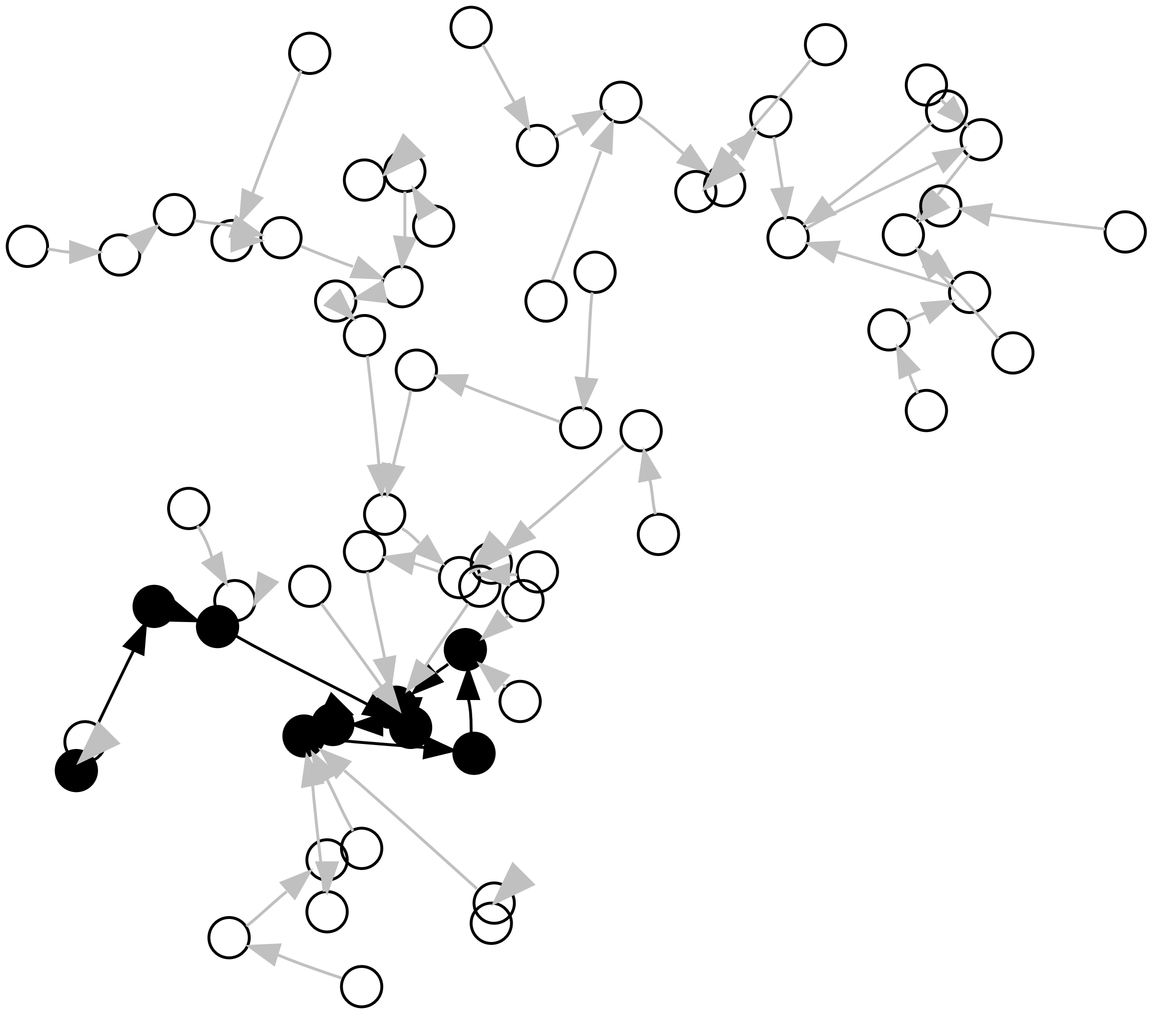


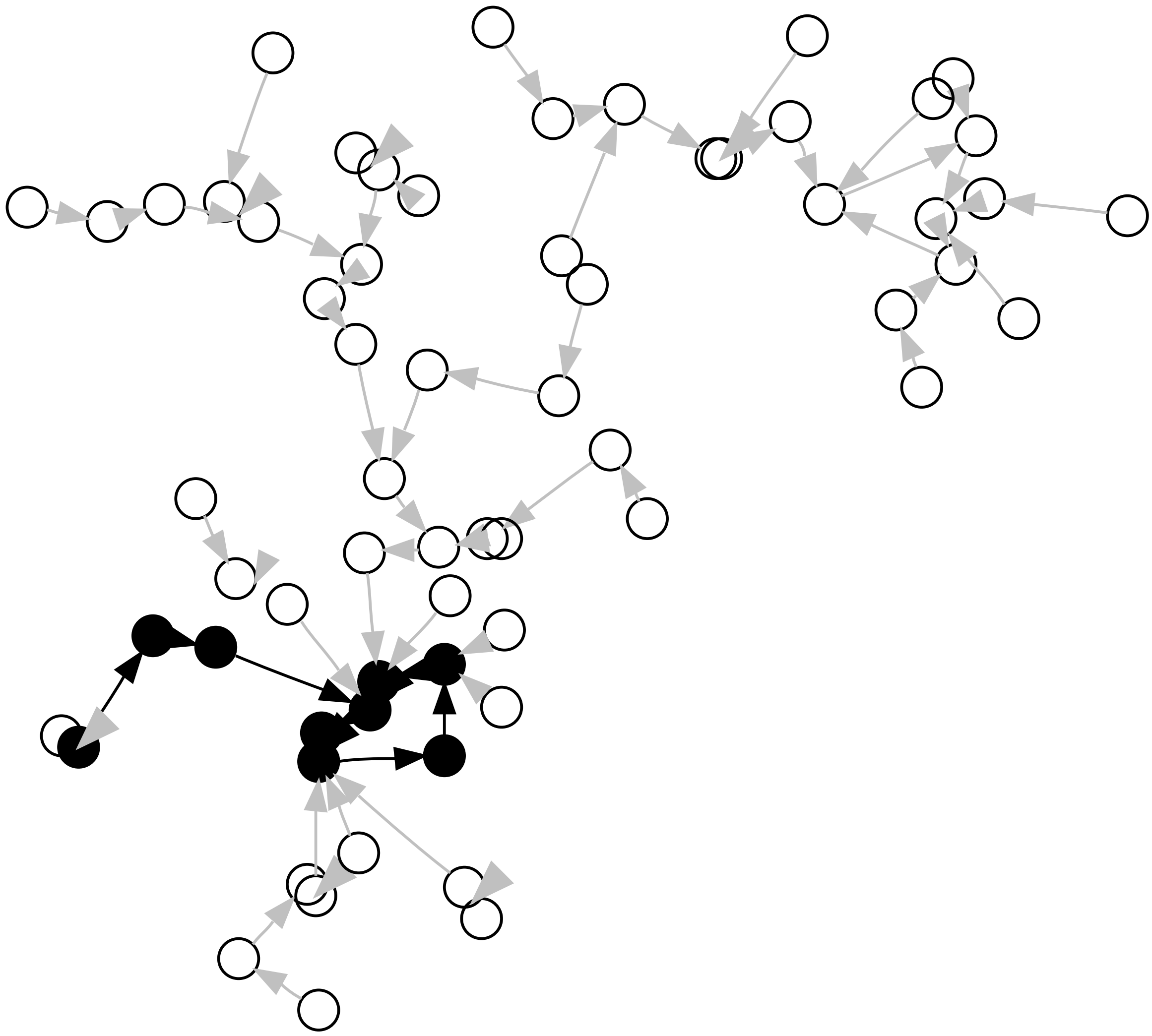


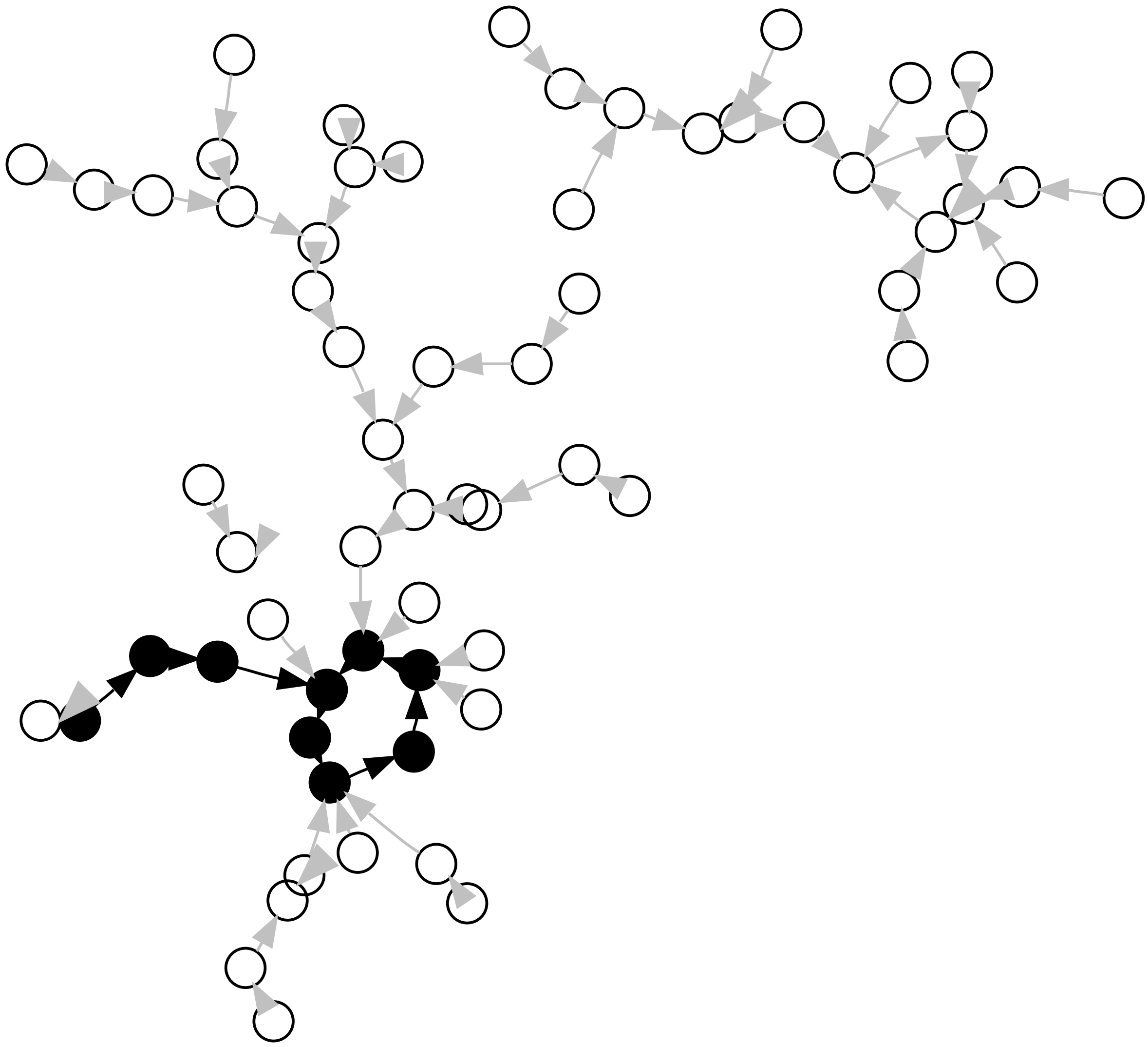


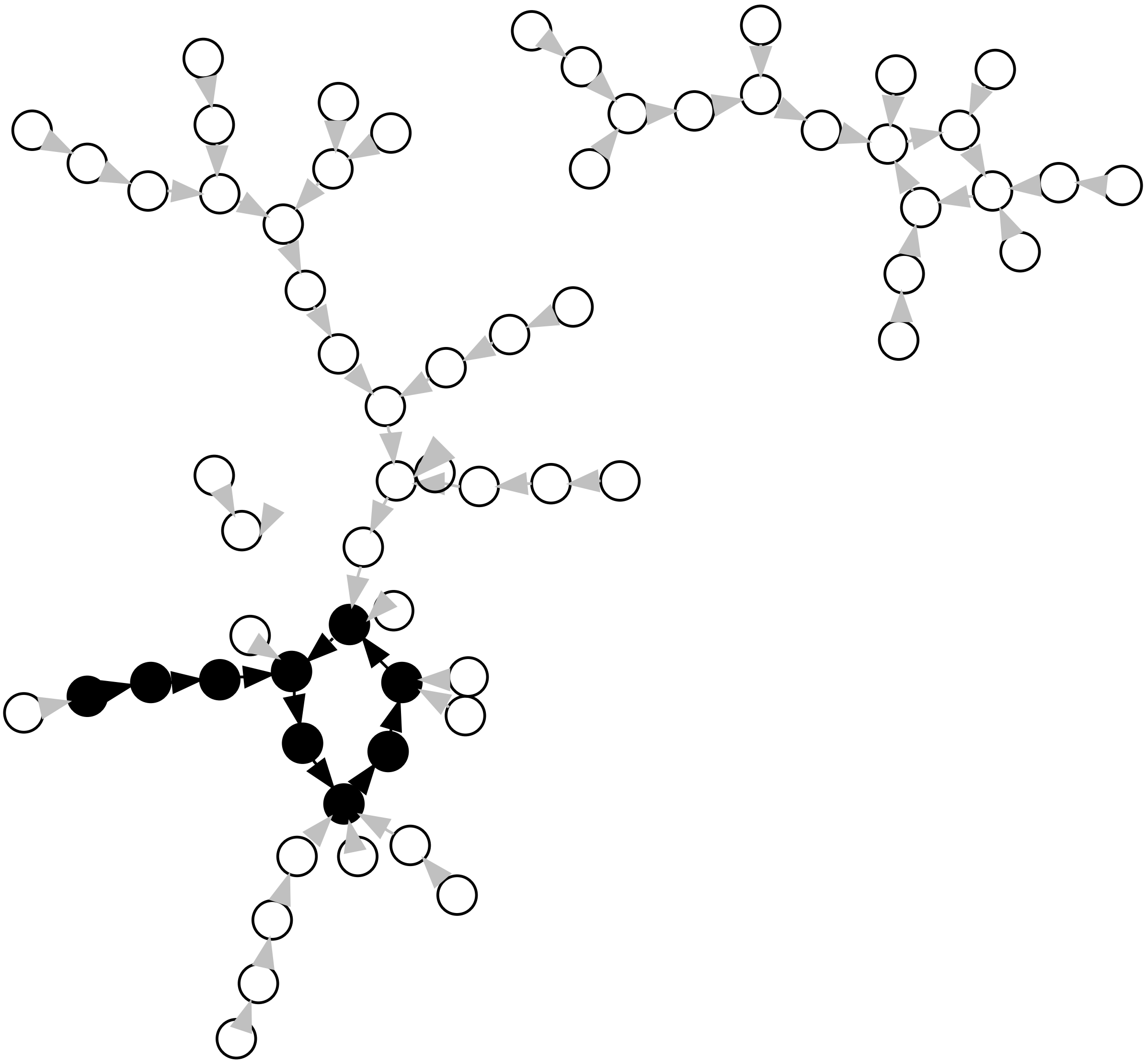












Performance of rho

Model walk as truly random.

Using m multiplications:
$\approx m$ points (x_i, y_i);
$\approx m^2/2$ pairs of points;
slope λ is missed
with chance $\approx (1 - 1/\ell)^{m^2/2}$
$\approx \exp(-m^2/(2\ell))$.

Average # multiplications
$\approx \sum_0^\infty \exp(-m^2/(2\ell))$
$\approx \int_0^\infty \exp(-m^2/(2\ell))\,dm$
$= \sqrt{\pi/4}\sqrt{2\ell} = (1.25\ldots)\sqrt{\ell}$.
Better than $(4/3 + o(1))\sqrt{\ell}$.

Performance of rho

Model walk as truly random.

Using m multiplications:
$\approx m$ points (x_i, y_i);
$\approx m^2/2$ pairs of points;
slope λ is missed
with chance $\approx (1 - 1/\ell)^{m^2/2}$
$\approx \exp(-m^2/(2\ell))$.

Average # multiplications
$\approx \sum_0^\infty \exp(-m^2/(2\ell))$
$\approx \int_0^\infty \exp(-m^2/(2\ell))\, dm$
$= \sqrt{\pi/4}\sqrt{2\ell} = (1.25\ldots)\sqrt{\ell}$.
Better than $(4/3 + o(1))\sqrt{\ell}$.

Don't ask about the worst case.

Anti-collisions

Bad news:
The walk is worse than random.

Very often have
$(x_{i+1}, y_{i+1}) = (x_i, y_i) + (s_j, t_j)$
followed later by
$(x_{k+1}, y_{k+1}) = (x_k, y_k) + (s_j, t_j)$.

Slope from
(x_{k+1}, y_{k+1}) to (x_{i+1}, y_{i+1})
is not new: same as slope from
(x_k, y_k) to (x_i, y_i).

Repeated slope: "anti-collision".

$m^2/2$ was too optimistic.
About $(1/r)m^2/2$ pairs
use same step, so only
$(1-1/r)m^2/2$ chances.

This replacement model $\Rightarrow$
$(\sqrt{\pi/2}/\sqrt{1-1/r}+o(1))\sqrt{\ell}$.

Can derive $\sqrt{1-1/r}$
from more complicated 1981
Brent–Pollard $\sqrt{V}$ heuristic.
1998 Blackburn–Murphy:
explicit $\sqrt{1-1/r}$.
2009 Bernstein–Lange:
simplified heuristic;
generalized $\sqrt{1-\sum_j p_j^2}$.

Higher-degree anti-collisions

Actually, rho is even worse!

Often have

$(x_{i+1}, y_{i+1}) = (x_i, y_i) + (s_j, t_j)$

$(x_{i+2}, y_{i+2}) = (x_{i+1}, y_{i+1}) + (s_h, t_h)$

followed later by

$(x_{k+1}, y_{k+1}) = (x_k, y_k) + (s_h, t_h)$

$(x_{k+2}, y_{k+2}) = (x_{k+1}, y_{k+1}) + (s_j, t_j)$

so slope from

(x_{k+2}, y_{k+2}) to (x_{i+2}, y_{i+2})

is not new.

"Degree-2 local anti-collisions":

$1/\sqrt{1 - 1/r - 1/r^2 + 1/r^3}$.

See paper for more.

Is rho optimal?

Allow r to grow slowly with ℓ.
(Not quickly: remember
cost of initial computation.)

$\sqrt{1-1/r} \to 1$.
$\sqrt{1-1/r-1/r^2+1/r^3} \to 1$.

Experimental evidence $\Rightarrow$
average $(\sqrt{\pi/2}+o(1))\sqrt{\ell}$.

But still have many
global anti-collisions:
slopes appearing repeatedly.

Two grumpy giants and a baby

B: $(0,0)+\{0,\ldots,n\}(0,1)$.

G1: $(1,0)+\{0,\ldots,n\}(0,n)$.

G2: $(2,0)-\{0,\ldots,n\}(0,n+1)$.

Two grumpy giants and a baby

B: $(0,0)+\{0,\ldots,n\}(0,1)$.

G1: $(1,0)+\{0,\ldots,n\}(0,n)$.

G2: $(2,0)-\{0,\ldots,n\}(0,n+1)$.

Minor initial cost: $(0,-(n+1))$.

Two grumpy giants and a baby

B: $(0,0)+\{0,\ldots,n\}(0,1)$.
G1: $(1,0)+\{0,\ldots,n\}(0,n)$.
G2: $(2,0)-\{0,\ldots,n\}(0,n+1)$.

Minor initial cost: $(0,-(n+1))$.

As before can interleave:
$(0,0),(1,0),(2,0),$
$(0,1),(1,n),(2,-(n+1)),$
$(0,2),(1,2n),(2,-2(n+1)),$
$(0,3),(1,3n),(2,-3(n+1)),$
$\vdots$
$(0,n),(1,n^2),(2,-n(n+1))$.

Grumpy performance

For $(1.5 + o(1))\sqrt{\ell}$ mults:

BSGS, with $n \approx 0.75\sqrt{\ell}$
or interleaved with $n \approx \sqrt{\ell}$,
finds $(0.5625 + o(1))\ell$ slopes.

Grumpy performance

For $(1.5 + o(1))\sqrt{\ell}$ mults:

BSGS, with $n \approx 0.75\sqrt{\ell}$
or interleaved with $n \approx \sqrt{\ell}$,
finds $(0.5625 + o(1))\ell$ slopes.

Truly random walk
finds $(0.6753\ldots + o(1))\ell$ slopes.

Grumpy performance

For $(1.5 + o(1))\sqrt{\ell}$ mults:

BSGS, with $n \approx 0.75\sqrt{\ell}$
or interleaved with $n \approx \sqrt{\ell}$,
finds $(0.5625 + o(1))\ell$ slopes.

Truly random walk
finds $(0.6753\ldots + o(1))\ell$ slopes.

Two grumpy giants and a baby,
with $n \approx 0.5\sqrt{\ell}$,
find $(0.71875 + o(1))\ell$ slopes.

Grumpy performance

For $(1.5 + o(1))\sqrt{\ell}$ mults:

BSGS, with $n \approx 0.75\sqrt{\ell}$
or interleaved with $n \approx \sqrt{\ell}$,
finds $(0.5625 + o(1))\ell$ slopes.

Truly random walk
finds $(0.6753\ldots + o(1))\ell$ slopes.

Two grumpy giants and a baby,
with $n \approx 0.5\sqrt{\ell}$,
find $(0.71875 + o(1))\ell$ slopes.

Also better average case than rho.