Lecture 11: Time in structured populations

MATH 242
11/6/2025

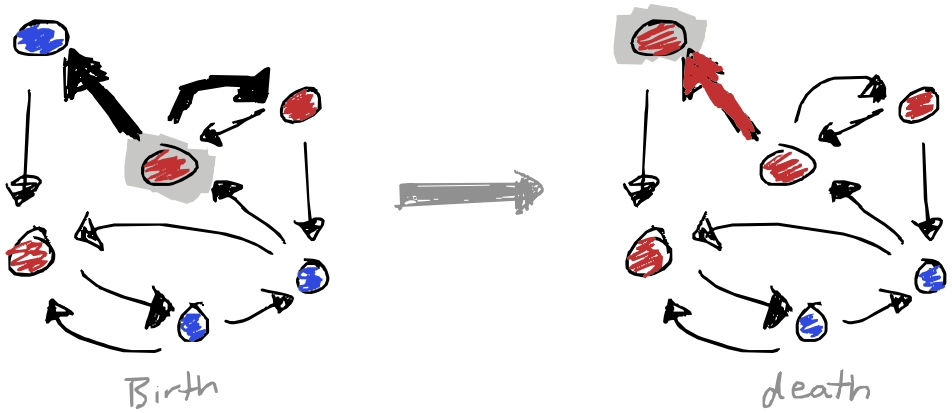
Last time, we saw in the Moran process:

(expected) absorption time is $\Theta(N \log N)$ ← N is population size

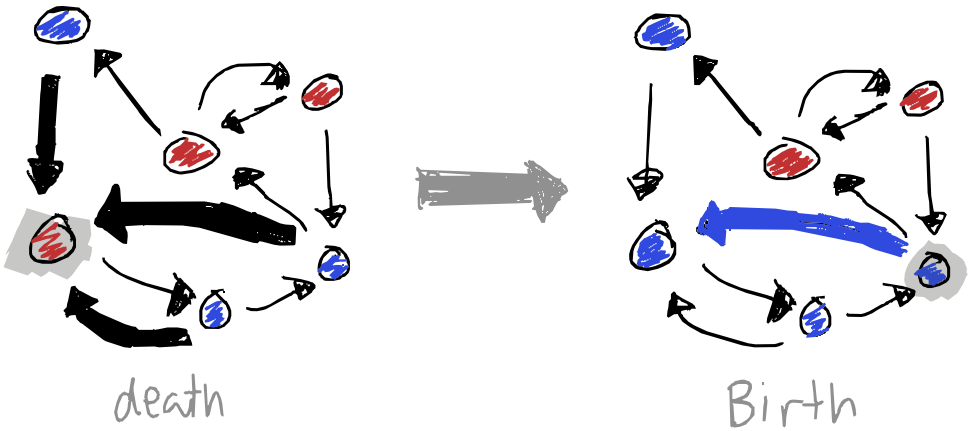
(expected) fixation time is $\Theta(N^2)$

Recall also Evolutionary Graph Theory as a way to model population structure

Bd:



dB:

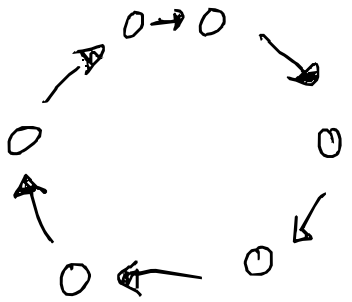


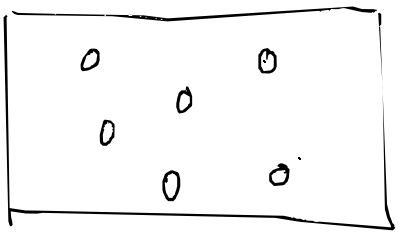
(Note: B is capitalized since selection acts on Birth in these two dynamics)

Then the Moran process can be seen as a population with a complete unweighted graph w/ self loops as its structure (ie "well-mixed").

Isothermal theorem: Under Bd dynamics, if death rate (i.e. "temperature") at each location is the same, then fixation probability is the same as Moran process $= \frac{1-1/r}{1-1/r^N}$ (or $1/N$ if $r=1$) for a single type A individual w/ relative fitness r.

Ex: Directed cycle has same fixation probability as Moran process in Bd dynamics





But what about fixation/absorption time?

For both directed cycles & well-mixed populations, the # of type A in the population is sufficient information to completely describe the state

(only true when directed cycle starts from a single type A since births/deaths keep the types segregated)

Also, absorption/fixation times only depend on $p_i^{\uparrow}$, $p_i^{\downarrow}$, and possibly ρ_i^A (fixation prob).

So even though $\gamma_i = \frac{p_i^{\downarrow}}{p_i^{\uparrow}} = \frac{1}{r}$ for both directed cycle

and well-mixed, and fixation probabilities are the same by isothermal theorem, when $r=1$ we have (Bd)

well mixed
$$p_i^{\uparrow} = \frac{i}{N} \cdot \frac{N-i}{N} = \frac{i(N-i)}{N^2}$$
$$p_i^{\downarrow} = \frac{N-i}{N} \cdot \frac{i}{N} = \frac{i(N-i)}{N^2} \qquad i=1,\dots,N-1$$
$$\Rightarrow p_i^{\circlearrowleft} = 1 - (p_i^{\uparrow} + p_i^{\downarrow}) = 1 - \frac{2i(N-i)}{N^2}$$

directed cycle
$$p_i^{\uparrow} = \frac{1}{N}, \quad p_i^{\downarrow} = \frac{1}{N}$$
$$\Rightarrow p_i^{\circlearrowleft} = 1 - \frac{2}{N}$$

Since $p_{i,\text{cycle}}^{\circlearrowleft,\text{directed}} > p_{i,\text{well-mixed}}^{\circlearrowleft}$ for $i=1,\dots,N-1$,

The fixation/absorption times of directed cycles must be longer than that of well-mixed populations (in expectation).

How much longer??

From last lecture, $x_i \triangleq \mathbb{E}AT_i$

$$y_{i+1} \triangleq x_{i+1} - x_i = -\frac{1}{p_i^{\uparrow}} + \gamma_i \, y_i = -\left(\sum_{j=2}^{i} \frac{1}{p_j^{\uparrow}} \cdot \prod_{k=j+1}^{i} \gamma_k \right) + \prod_{k=1}^{i} \gamma_k$$

($x_i - x_0 = x_i$)

for $i=1,\dots,N-1$

$$\Rightarrow x_1 = \frac{\sum_{i=0}^{N-1} r^{-i} \sum_{j=2}^{i} \frac{r^j}{p_j^{\uparrow}}}{\sum_{i=0}^{N-1} r^{-i}}$$

for both directed cycles & well-mixed

For directed cycles, $p_j^{\uparrow} = 1/N$ $(r=1)$,

So $(r=1)$

$$X_1 = N \cdot \frac{\sum_{i=0}^{N-1} 1^{-i} \sum_{j=2}^{i} 1^j}{\sum_{i=0}^{N-1} 1^{-i}}$$

When $r=1$,

$$\mathrm{EAT}_1 = X_1 = \frac{N \cdot \sum_{i=0}^{N-1} \sum_{j=2}^{i} 1}{N} = 1+2+3+\ldots+(N-2)$$

$$= \Theta(N^2).$$

For fixation times, recall

$$\mathrm{EFT}_1 = X_1' = \frac{1}{2p_1} \sum_{i=1}^{N-1} \prod_{j=2}^{i} \eta_j + \sum_{i=1}^{N-1} \sum_{j=3}^{i} \frac{1}{q_j} \prod_{k=j+1}^{i} \eta_k$$

$$= \frac{N}{2} \sum_{i=1}^{N-1} \underbrace{\left(\prod_{j=2}^{i} \frac{1-1/j}{1+1/j} \right)}_{\leq 1} + N \cdot \sum_{i=1}^{N-1} \sum_{j=3}^{i} \underbrace{\left(\prod_{k=j+1}^{i} \frac{1-1/k}{1+1/k} \right)}_{\leq 1}.$$

$p_i := p_i^{\uparrow} = \frac{1}{N}$

$q_i := p_i \cdot (1+1/i)$

$\eta_i := \frac{1-1/i}{1+1/i}$

Thus

$$\mathrm{EFT}_1 \leq N^2/2 + N \cdot N^2/2$$

$$= O(N^3)$$

and

$$\mathrm{EFT}_1 \geq N \cdot \sum_{i=1}^{N-1} \sum_{j=3}^{i} \frac{j(j+1)}{i(i+1)}$$

$$\approx N \cdot \sum_{i=1}^{N-1} \frac{1}{\Theta(i^2)} \cdot \Theta(i^3)$$

$$\approx N \sum_{i=1}^{N-1} \Theta(i)$$

$$= N \cdot \Theta(N^2)$$

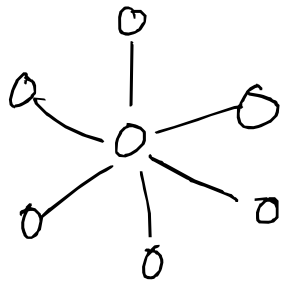
$$= \Omega(N^3).$$

So, $\mathbb{E}FT_1 = \Theta(N^3)$.

We have shown that there are simple 1D structures w/ same fixation prob. as Moran process but differing timescales.

Our analysis heavily uses the fact that the entire Markov chain state space is 1D.
But for a general population structure (eg graph), the dimensionality can become large.

Ex: Consider a star graph (undirected)



states can be reduced to
(i) # leaves containing type A
(ii) is center type A?

$\{0, 1, \ldots, N-1\} \times \{\text{Yes}, \text{No}\}$.

This is a 2D representation of the state space and is sufficient information to compute fixation probs/times

However, the recurrence equations start to became more messy.

In general, we may need N dimensions to get exact expression for fixation probabilities/times.

When $r=1$, it is known that for undirected (connected) graphs $G=(V,E)$ under Bd dynamics

$$\text{fixation prob of type A when type A occupies locations } S \subseteq V \overset{r=1}{=} \rho^{A}_{S,r=1}(G) = \frac{\sum_{u \in S} \frac{1}{\deg(u)}}{\sum_{u \in V} \frac{1}{\deg(u)}}$$

But for $r \neq 1$, there is no known exact, explicit, analytical, general formula known. :(

One approach to computationally approximate $\rho^{A}_{S,r}(G)$ for any r and S is the following:

(1) Run M independent Bd simulations starting from S

(2) Approximate $\rho^{A}_{S,r}(G)$ as $\hat{\rho}^{A}_{S,r}(G) = \frac{\#\text{ fixated simulations}}{M}$.

How good is this approximation? (Use a concentration bound)

Chernoff bound

"bad approximation" event

$$\mathbb{P}\left\{ \left| \hat{\rho}^{A}_{S,r}(G) - \rho^{A}_{S,r}(G) \right| > \varepsilon \right\} \leq \underbrace{2\exp\left(-\frac{\varepsilon^2 M}{4}\right)}_{\text{exponentially small in } M}$$

But in (1), we need the simulations to finish quickly or else our computational approach becomes intractable.

(arbitrary) (connected, unweighted)

For an undirected graph $G=(V,E)$,
it suffices to show that $\exists T: \mathbb{N} \to \mathbb{R}_{\geq 0}$ s.t.

$$\mathbb{E}\, AT_{s,r}(G) \leq T(N) \leq \text{poly}(N).$$

Then we can argue (by Markov ineq) that each one of the M simulations will absorb within time $100MT(N)$ w/ high prob:
(union bound)

$$\mathbb{P}\{\exists \text{ simulation longer than } 100M\cdot T(N)\} \leq M \cdot \mathbb{P}\{\text{simulation takes longer than } 100MT(N)\}$$

$$\leq M \frac{\mathbb{E}\, AT_{s,r}(G)}{100MT(N)}$$

$$\leq \frac{1}{100}.$$

So if we make M polynomial in N, our approximation has a very high chance of being accurate and quick to compute. ☺

Q: How to show existence of $T(N)$?

Think of a game of chess (or any 2-player game). At each point in the game, how do we evaluate who is "winning" or "how quickly the game will end"?

Assumption: The more one player is winning, the quicker the game will likely end.

More specifically, the rate at which a player

is taking the lead, the faster the game will likely end.

Q: How do we evaluate who is winning at a particular point in time?

A: We would like a function $\Phi : \Omega \to [0,1]$ (state space)

s.t. $\exists$ disjoint subsets S_1, S_2 of Ω containing all absorbing states with:

(i) $\Phi(S_1) = 0 \quad \forall S_1 \subseteq S_1$ (i.e. player 1 winning states)

(ii) $\Phi(S_2) = 1 \quad \forall S_2 \subseteq S_2$ (i.e. player 2 winning states)

(iii) $\Phi(S') \geq \Phi(S) \Rightarrow$ it is more likely to absorb in S_2 than S_1 when starting at S' rather than S.

In chess, one can look k moves into the future and compute the number of player 1 wins vs player 2 wins.

But this is very complicated even if it is accurate.

A simpler method is to assign each piece a weight and calculate the weighted sum of pieces each player has.

This method is not as accurate but may suffice in many scenarios.

Lemma: If $X_0, X_1, X_2, \ldots \subseteq \Omega$ is a Markov chain, and $\tau \triangleq \min\{t : \Phi(X_t) = 1\}$, then

(i) $\mathbb{E}\tau < \infty$

(ii) $\underset{X_{t+1}}{\mathbb{E}}\left[\Phi(X_{t+1}) - \Phi(X_t) \mid X_t\right] > \delta$

$\Rightarrow \quad \mathbb{E}\,\tau < \frac{1-\mu}{\delta}.$

(iii) $\Phi(X_0) \geq \mu$

Now consider Bd process on connected, unweighted, undirected graph $G = (V, E)$, $r > 1$, except if type A goes extinct, restart the process.

Surely the absorption time of this augmented process if at least as long as the time for the typical, no-restart Bd process.

Weight each type A individual by the inverse of the degree of the vertex it occupies, so that

$$\Phi(S) = \frac{\sum_{u \in S} \frac{1}{\deg(u)}}{\sum_{u \in V} \frac{1}{\deg(u)}} = \rho^{A}_{s, r=1}$$

Then $\mathbb{E}\left[\Phi(X_{t+1}) - \Phi(X_t) \mid X_t = S\right]$ total fitness $= F(S) \triangleq r|S| + (N - |S|)$

$$= \sum_{\substack{u \in S \\ v \notin S \\ (u,v) \in E}} \frac{r}{F(S)} \cdot \frac{1}{\deg(u)} \Phi(\{v\}) - \sum_{\substack{u \in S \\ v \notin S \\ (u,v) \in E}} \frac{1}{F(S)} \cdot \frac{1}{\deg(v)} \cdot \Phi(\{u\})$$

$$= \frac{r-1}{F(S) \cdot \sum_{u \in V} \frac{1}{\deg(u)}} \cdot \sum_{\substack{u \in S \\ v \notin S \\ (u,v) \in E}} \frac{1}{\deg(u)\deg(v)} \qquad S \neq \phi, V$$

$$\geq \frac{r-1}{N \cdot N} \cdot \frac{1}{}$$

$rN \cdot N$ $N \cdot N$

$= \left(1 - \frac{1}{r}\right) N^{-4}$.

Thus by the lemma,

$$\mathbb{E}\, AT_{x_0} \leq \underbrace{(1-\mu)}_{\leq 1} \cdot (1-\tfrac{1}{r})^{-1} N^4$$
$$\leq (1-\tfrac{1}{r})^{-1} N^4 =: T(N).$$

$\Rightarrow$ fast algorithm for approximating $\varphi^A_{s,r>1}$ for any undirected, unweighted graph.

What about directed graphs?

Unfortunately, it is known to be likely intractable in the worst case (#P-hard). ☹

But for certain graphs (eg. regular, "balanced", etc) the simulation approach still works.

Some directed graphs take ~exp(N) time to absorb.

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import random
import enum
from dataclasses import dataclass
```

In [4]:
```
class Type(enum.Enum):
  A = 0
  B = 1

@dataclass(frozen=True)
class SimulationResult:
  N: int
  population_structure: str
  fixation: bool
  time: int
```

In [5]:
```
def simulate_well_mixed(N: int) -> SimulationResult:
  i = 1 # number of type A individuals
  t = 0 # time
  while 0 < i < N:
    birther_type, = random.choices([Type.A, Type.B], weights=(i, N-i))
    dier_type, = random.choices([Type.A, Type.B], weights=(i, N-i))
    if birther_type == Type.A and dier_type == Type.B:
      i += 1
    if birther_type == Type.B and dier_type == Type.A:
      i -= 1
    t += 1
  return SimulationResult(
    N=N,
    population_structure='well-mixed',
    fixation=i==N,
    time=t,
  )

def simulate_directed_cycle(N: int) -> SimulationResult:
  i = 1 # number of type A individuals
  t = 0 # time
  while 0 < i < N:
    birther_type, = random.choices([Type.A, Type.B], weights=(i, N-i))
    dier_type, = random.choices(
      [Type.A, Type.B],
      weights=(i-1, 1) if birther_type == Type.A else (1, N-i-1))
    if birther_type == Type.A and dier_type == Type.B:
      i += 1
    if birther_type == Type.B and dier_type == Type.A:
      i -= 1
    t += 1
  return SimulationResult(
    N=N,
    population_structure='directed-cycle',
```

```
    fixation=i==N,
    time=t,
  )

def simulate(N: int, population_structure: str) -> SimulationResult:
  if population_structure == 'well-mixed':
    return simulate_well_mixed(N)
  elif population_structure == 'directed-cycle':
    return simulate_directed_cycle(N)
  else:
    raise ValueError(f'Unknown population structure: {population_structure}')
```

In [6]:
```
N = 100
TRIALS = 1000
data = [
  simulate(n, population_structure)
  for population_structure in ('well-mixed', 'directed-cycle')
  for n in range(1, N+1)
  for _ in range(TRIALS)
]
df = pd.DataFrame(data)
```

In [7]:
```
df.sample(n=10)
```

Out[7]:

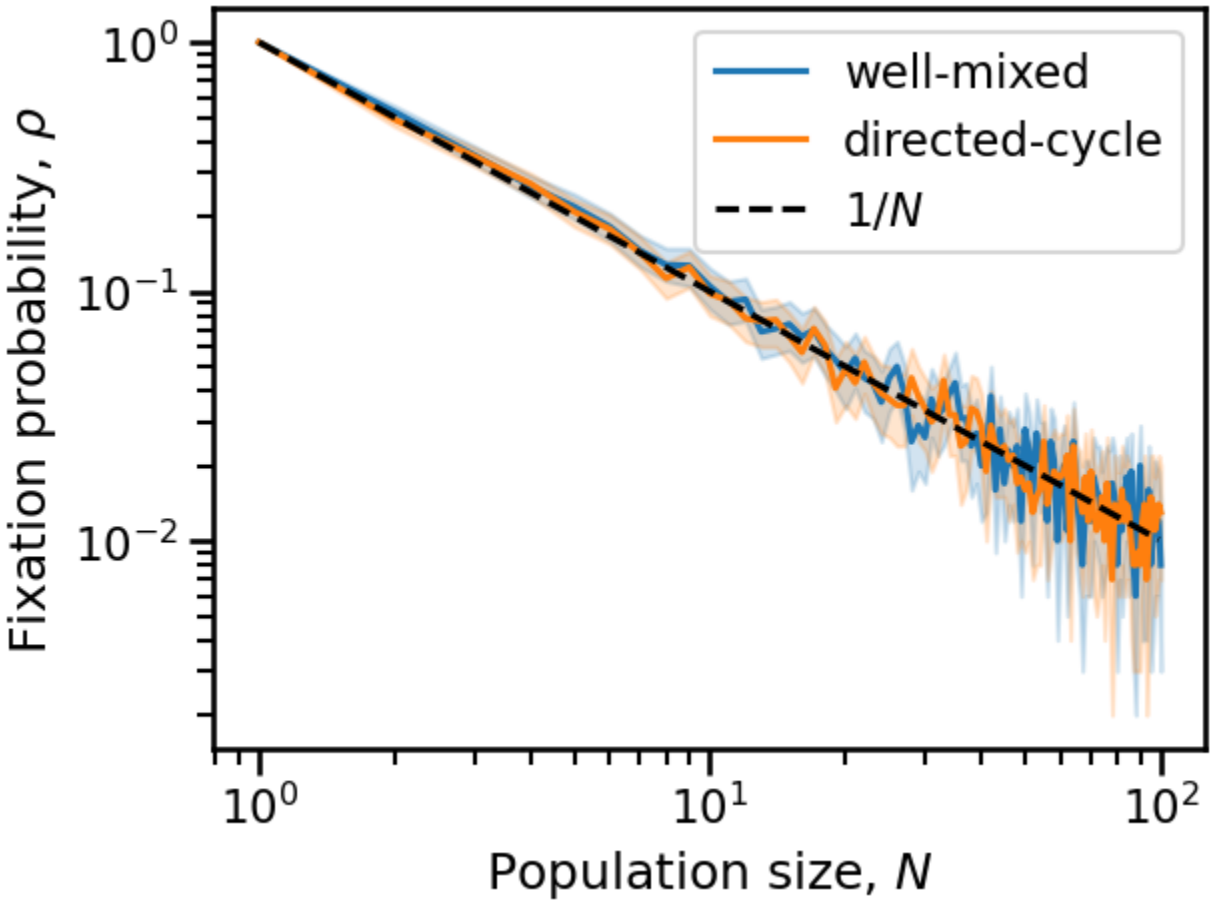
	N	population_structure	fixation	time
97178	98	well-mixed	False	67
182155	83	directed-cycle	True	49925
117962	18	directed-cycle	True	955
37218	38	well-mixed	False	2548
5722	6	well-mixed	False	5
69284	70	well-mixed	False	236
185556	86	directed-cycle	False	92
164610	65	directed-cycle	False	34
97390	98	well-mixed	False	165
86300	87	well-mixed	False	44

In [11]:
```
with sns.plotting_context(context='talk'):
  g = sns.lineplot(
    data=df,
    x='N',
    y='fixation',
    hue='population_structure',
  )

  Ns = np.arange(1, N)
```

```
  sns.lineplot(
    x=Ns,
    y=1/Ns,
    label='$1/N$',
    ax=g,
    color='black',
    linestyle='--',
  )

  _ = g.set(
    xscale='log',
    yscale='log',
    xlabel='Population size, $N$',
    ylabel='Fixation probability, $\\rho$',
  )
```

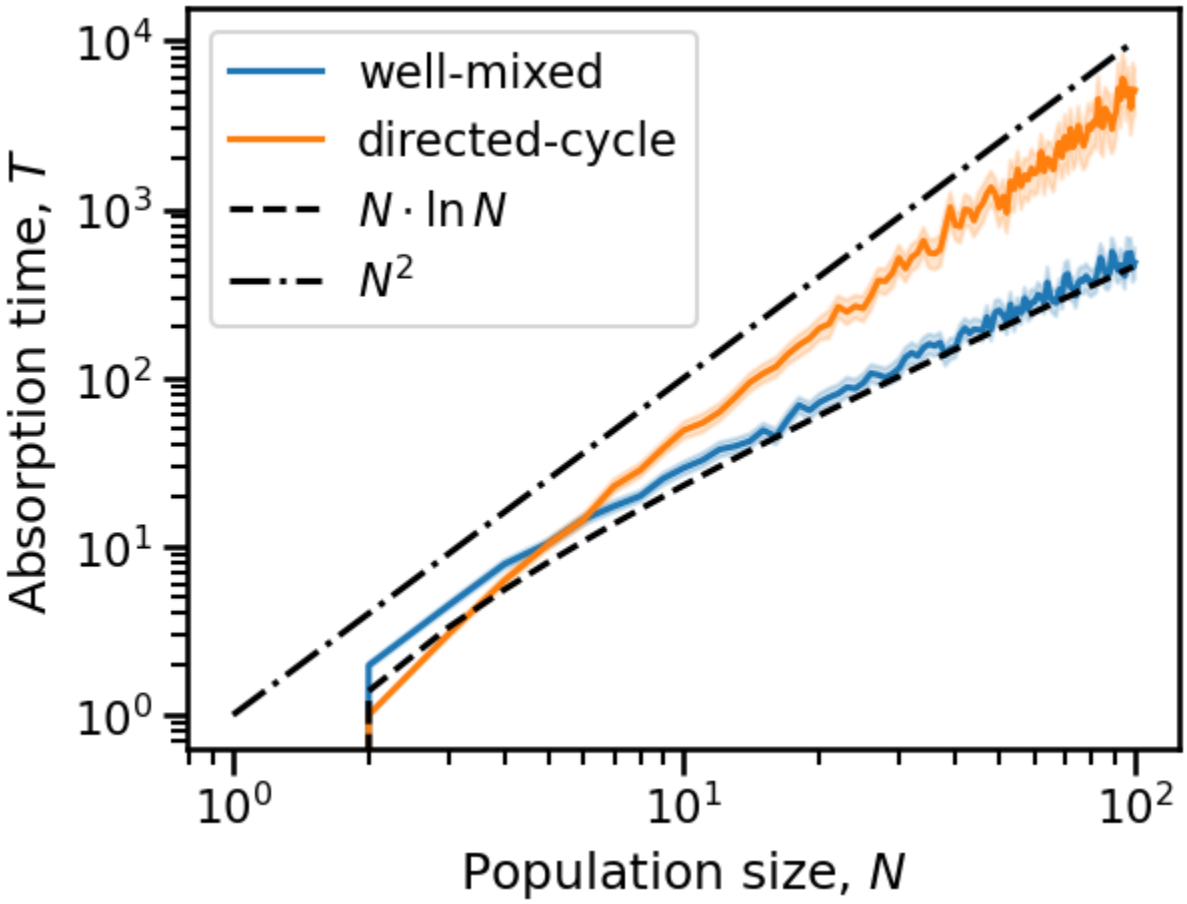


In [13]:
```
with sns.plotting_context(context='talk'):
  g = sns.lineplot(
    data=df,
    x='N',
    y='time',
    hue='population_structure',
  )

  Ns = np.arange(1, N)
```

```
  sns.lineplot(
    x=Ns,
    y=Ns*np.log(Ns),
    label='$N\\cdot \\ln N$',
    ax=g,
    color='black',
    linestyle='--',
  )
  sns.lineplot(
    x=Ns,
    y=Ns**2,
    label='$N^2$',
    ax=g,
    color='black',
    linestyle='-.',
  )

  _ = g.set(
    xscale='log',
    yscale='log',
    xlabel='Population size, $N$',
    ylabel='Absorption time, $T$',
  )
```

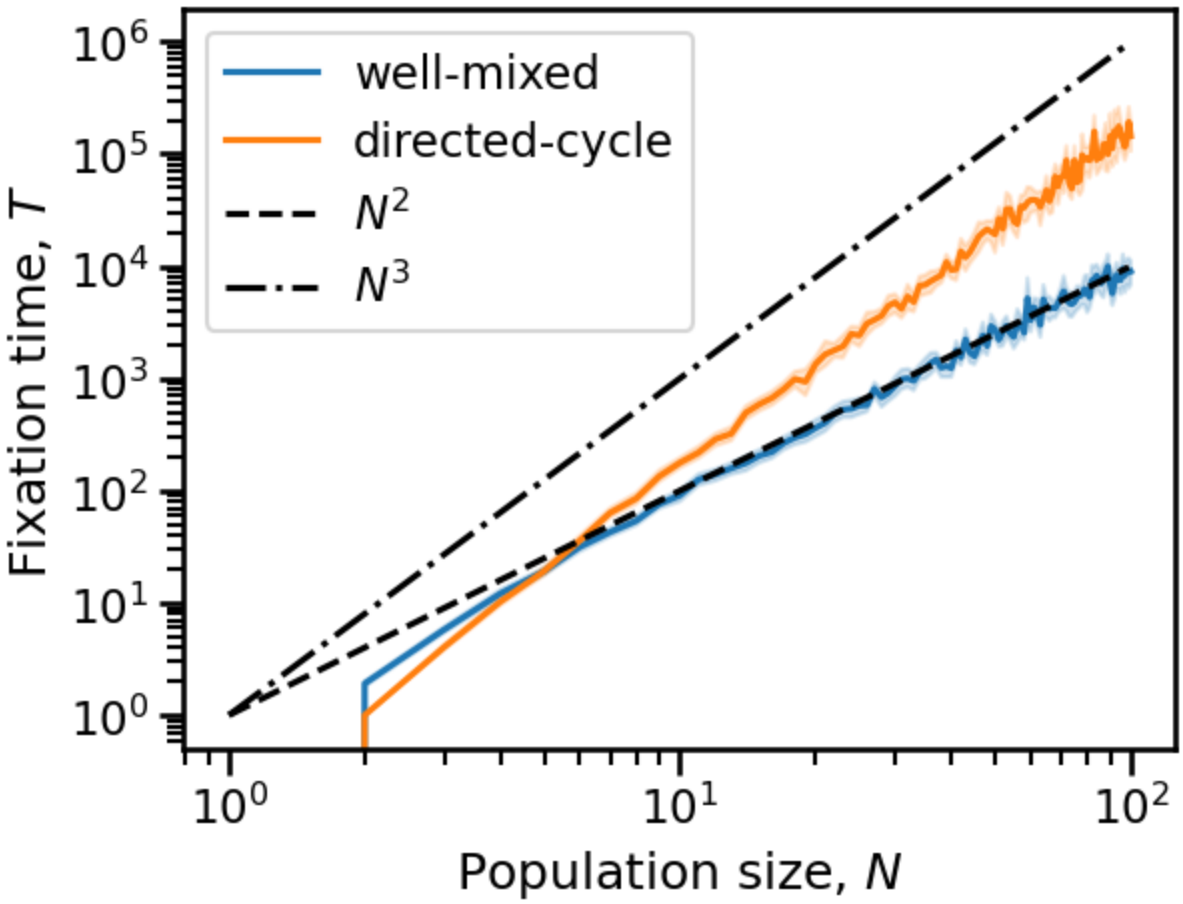


In [14]:
```
with sns.plotting_context(context='talk'):
```

```
g = sns.lineplot(
  data=df[df['fixation']==True],
  x='N',
  y='time',
  hue='population_structure',
)

Ns = np.arange(1, N)
sns.lineplot(
  x=Ns,
  y=Ns**2,
  label='$N^2$',
  ax=g,
  color='black',
  linestyle='--',
)
sns.lineplot(
  x=Ns,
  y=Ns**3,
  label='$N^3$',
  ax=g,
  color='black',
  linestyle='-.',
)

_ = g.set(
  xscale='log',
  yscale='log',
  xlabel='Population size, $N$',
  ylabel='Fixation time, $T$',
)
```


well-mixed
directed-cycle
N^2
N^3
Fixation time, T
Population size, N
10^0
10^1
10^2
10^3
10^4
10^5
10^6