The Supermarket Model with Memory

Rapid Mixing by Coupling

Derek Wan

London School of Economics

10 March 2010

The supermarket model

- A system of n FIFO queues. Represent states by vectors in $\mathbb{Z}^n_{\geq}$.
  - E.g., $n = 7$: $(1, 4, 7, 0, 4, 4, 6)$ has queue 3 with 7 customers, etc...
- Arrival times form a Poisson process of rate λn, where $\lambda \in (0, 1)$. Each arriving customer chooses $d \leq n$ queues uniformly at random (with replacement), then joins the shortest queue.
  - E.g., if choices $= (5, 7)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Resolve ties by joining the leftmost queue.
  - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Service times are i.i.d. $\mathrm{Exp}(1)$ random variables. So model all departures by a Poisson process of rate n, and for each departure time, pick a queue uniformly at random and remove a customer.
  - E.g., if choice $= 7$, then $(1, 4, 7, 0, 4, 4,$ [illegible]$) \to (1, 4, 7, 0, 4, 4,$ [illegible]$)$.
- Ignore departures from empty queues.
  - E.g., if choice $= 4$, then $(1, 4, 7,$ [illegible]$, 4, 4, 6) \to (1, 4, 7,$ [illegible]$, 4, 4, 6)$.

The supermarket model

- A system of n FIFO queues. Represent states by vectors in $\mathbb{Z}^n_{\geq}$.
    - E.g., $n = 7$: $(1, 4, 7, 0, 4, 4, 6)$ has queue 3 with 7 customers, etc...
- Arrival times form a Poisson process of rate λn, where $\lambda \in (0, 1)$. Each arriving customer chooses $d \leq n$ queues uniformly at random (with replacement), then joins the shortest queue.
    - E.g., if choices $= (5, 7)$, then $(1, 4, 7, 0, 4, 4, 6) \rightarrow (1, 4, 7, 0, 5, 4, 6)$.
- Resolve ties by joining the leftmost queue.
    - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \rightarrow (1, 4, 7, 0, 5, 4, 6)$.
- Service times are i.i.d. $\text{Exp}(1)$ random variables. So model all departures by a Poisson process of rate n, and for each departure time, pick a queue uniformly at random and remove a customer.
    - E.g., if choice $= 7$, then $(1, 4, 7, 0, 4, 4, 6) \rightarrow (1, 4, 7, 0, 4, 4, 5)$.
- Ignore departures from empty queues.
    - E.g., if choice $= 4$, then $(1, 4, 7, 0, 4, 4, 6) \rightarrow (1, 4, 7, 0, 4, 4, 6)$.

The supermarket model

- A system of n FIFO queues. Represent states by vectors in $\mathbb{Z}^n_{\geq}$.
  - E.g., $n = 7$: $(1, 4, 7, 0, 4, 4, 6)$ has queue 3 with 7 customers, etc...
- Arrival times form a Poisson process of rate λn, where $\lambda \in (0, 1)$. Each arriving customer chooses $d \leq n$ queues uniformly at random (with replacement), then joins the shortest queue.
  - E.g., if choices $= (5, 7)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Resolve ties by joining the leftmost queue.
  - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Service times are i.i.d. $\mathrm{Exp}(1)$ random variables. So model all departures by a Poisson process of rate n, and for each departure time, pick a queue uniformly at random and remove a customer.
  - E.g., if choice $= 7$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 5)$.
- Ignore departures from empty queues.
  - E.g., if choice $= 4$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 6)$.

The supermarket model

- A system of n FIFO queues. Represent states by vectors in $\mathbb{Z}_{\geq}^{n}$.
  - E.g., $n = 7$: $(1, 4, 7, 0, 4, 4, 6)$ has queue 3 with 7 customers, etc...
- Arrival times form a Poisson process of rate λn, where $\lambda \in (0, 1)$. Each arriving customer chooses $d \leq n$ queues uniformly at random (with replacement), then joins the shortest queue.
  - E.g., if choices $= (5, 7)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Resolve ties by joining the leftmost queue.
  - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Service times are i.i.d. $\mathrm{Exp}(1)$ random variables. So model all departures by a Poisson process of rate n, and for each departure time, pick a queue uniformly at random and remove a customer.
  - E.g., if choice $= 7$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 5)$.
- Ignore departures from empty queues.
  - E.g., if choice $= 4$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 6)$.

The supermarket model

- A system of n FIFO queues. Represent states by vectors in $\mathbb{Z}_{\geq}^{n}$.
  - E.g., $n = 7$: $(1, 4, 7, 0, 4, 4, 6)$ has queue 3 with 7 customers, etc...
- Arrival times form a Poisson process of rate λn, where $\lambda \in (0, 1)$. Each arriving customer chooses $d \leq n$ queues uniformly at random (with replacement), then joins the shortest queue.
  - E.g., if choices $= (5, 7)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Resolve ties by joining the leftmost queue.
  - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Service times are i.i.d. $\text{Exp}(1)$ random variables. So model all departures by a Poisson process of rate n, and for each departure time, pick a queue uniformly at random and remove a customer.
  - E.g., if choice $= 7$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 5)$.
- Ignore departures from empty queues.
  - E.g., if choice $= 4$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 4, 4, 6)$.

The supermarket model with memory

- A memory tracks the index of an additional queue. Represent states by pairs in $\hat{\mathcal{Q}} := \mathbb{Z}^n_{\geq} \times \{1, ..., n\}$.
  - E.g., $((1, 4, 7, 0, 4, 4, 6)\,, 1)$ has queue 1 as its memory. $(\underline{1}, 4, 7, 0, 4, 4, 6)$ is more readable.
- Each arriving customer will add the memory queue to his/her d choices, before joining the shortest queue. The memory then saves the index of the shortest queue out of those under consideration. Saving the memory: resolve ties by saving the leftmost queue out of those considered.
  - E.g., if choices $= (5, 6)$, then $(\underline{1}, 4, 7, 0, 4, 4, 6) \to (\underline{2}, 4, 7, 0, 4, 4, 6)$.
- Joining a queue: resolve ties by joining the leftmost *non-memory* queue.
  - E.g., if choices $= (5, 6)$, then $(1, \underline{4}, 7, 0, 4, 4, 6) \to (1, \underline{4}, 7, 0, 5, 4, 6)$.
- Departures still the same.

The supermarket model with memory

- A memory tracks the index of an additional queue. Represent states by pairs in $\hat{\mathcal{Q}} := \mathbb{Z}^n_{\geq} \times \{1, \ldots, n\}$.
    - E.g., $((1, 4, 7, 0, 4, 4, 6), 1)$ has queue 1 as its memory. $(\underline{1}, 4, 7, 0, 4, 4, 6)$ is more readable.
- Each arriving customer will add the memory queue to his/her d choices, before joining the shortest queue. The memory then saves the index of the shortest queue out of those under consideration. Saving the memory: resolve ties by saving the leftmost queue out of those considered.
    - E.g., if choices $= (5, 6)$, then $(\underline{1}, 4, 7, 0, 4, 4, 6) \to (\underline{2}, 4, 7, 0, 4, 4, 6)$.
- Joining a queue: resolve ties by joining the leftmost *non-memory* queue.
    - E.g., if choices $= (5, 6)$, then $(1, 4, 7, 0, 4, 4, 6) \to (1, 4, 7, 0, 5, 4, 6)$.
- Departures still the same.

The supermarket model with memory

- A memory tracks the index of an additional queue. Represent states by pairs in $\hat{\mathcal{Q}} := \mathbb{Z}^n_{\geq} \times \{1, \ldots, n\}$.
  - E.g., $((1,4,7,0,4,4,6), 1)$ has queue 1 as its memory. $(\underline{1},4,7,0,4,4,6)$ is more readable.
- Each arriving customer will add the memory queue to his/her d choices, before joining the shortest queue. The memory then saves the index of the shortest queue out of those under consideration. Saving the memory: resolve ties by saving the leftmost queue out of those considered.
  - E.g., if choices $= (5,6)$, then $(\underline{1},4,7,0,4,4,6) \to (\underline{2},4,7,0,4,4,6)$.
- Joining a queue: resolve ties by joining the leftmost *non-memory* queue.
  - E.g., if choices $= (5,6)$, then $(1,\underline{4},7,0,4,4,6) \to (1,\underline{4},7,0,5,4,6)$.
- Departures still the same.

The supermarket model with memory

- A memory tracks the index of an additional queue. Represent states by pairs in $\hat{\mathcal{Q}} := \mathbb{Z}^n_{\geq} \times \{1, ..., n\}$.
  - E.g., $((1, 4, 7, 0, 4, 4, 6), 1)$ has queue 1 as its memory. $(\underline{1}, 4, 7, 0, 4, 4, 6)$ is more readable.
- Each arriving customer will add the memory queue to his/her d choices, before joining the shortest queue. The memory then saves the index of the shortest queue out of those under consideration. Saving the memory: resolve ties by saving the leftmost queue out of those considered.
  - E.g., if choices $= (5, 6)$, then $(\underline{1}, 4, 7, 0, 4, 4, 6) \to (\underline{2}, 4, 7, 0, 4, 4, 6)$.
- Joining a queue: resolve ties by joining the leftmost *non-memory* queue.
  - E.g., if choices $= (5, 6)$, then $(1, \underline{4}, 7, 0, 4, 4, 6) \to (1, \underline{4}, 7, 0, 5, 4, 6)$.
- Departures still the same.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)
With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\Longrightarrow$ faster search times.
  - In network systems, better distribution $\Longrightarrow$ faster processing times.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)
With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\implies$ faster search times.
  - In network systems, better distribution $\implies$ faster processing times.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)

With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\Longrightarrow$ faster search times.
  - In network systems, better distribution $\Longrightarrow$ faster processing times.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)

With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\Longrightarrow$ faster search times.
  - In network systems, better distribution $\Longrightarrow$ faster processing times.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)

With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\implies$ faster search times.
  - In network systems, better distribution $\implies$ faster processing times.

Motivation and applications

- Studied by numerous authors, e.g. Graham, Luczak and McDiarmid, Luczak and Norris, Mitzenmacher.
- Origins in the classical *balls and bins* model:
  Throw n balls into n bins, with each ball going into the least loaded of $d \leq n$ bins chosen uniformly at random.

Theorem (Gonnet, 1981; Azar, Broer, Karlin and Upfal 1999.)

With high probability, the maximum queue length is

$$\frac{\log n}{\log \log n} \text{ if } d = 1, \quad \frac{\log \log n}{\log d} \text{ if } d \geq 2.$$

- Theme: the **power of two choices** in load distribution.
  Applications in computer science, e.g.:
  - In a table to be searched, lower maximum load $\implies$ faster search times.
  - In network systems, better distribution $\implies$ faster processing times.

Queue ranks

- For $q \in \hat{\mathcal{Q}}$, rank the queues from 1 to n so that each arriving customer joins the lowest ranked queue amongst his/her choices. Let $R(q,j)$ be the rank of queue j in state q, and let

$$R(q) := (R(q,1), ..., R(q,n)).$$

Example

If $q = (4, \underline{1}, 8, 4)$ then $R(q) = (2, 1, 4, 3)$, since

1. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, 1, \square, \square)$.
   $q = (4, \underline{1}, 8, 4) \implies R(q) = (2, \square, \square, 3)$.
   $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, \square, 4, \square)$.

Queue ranks

- For $q \in \hat{\mathcal{Q}}$, rank the queues from 1 to n so that each arriving customer joins the lowest ranked queue amongst his/her choices. Let $R(q,j)$ be the rank of queue j in state q, and let

$$R(q) := (R(q,1), ..., R(q,n)).$$

Example

If $q = (4, \underline{1}, 8, 4)$ then $R(q) = (2, 1, 4, 3)$, since

1. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, 1, \square, \square)$.
2. $q = (4, \underline{1}, 8, 4) \implies R(q) = (2, \square, \square, 3)$.
3. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, \square, 4, \square)$.

Queue ranks

- For $q \in \hat{\mathcal{Q}}$, rank the queues from 1 to n so that each arriving customer joins the lowest ranked queue amongst his/her choices. Let $R(q, j)$ be the rank of queue j in state q, and let

$$R(q) := (R(q,1), ..., R(q,n)).$$

Example

If $q = (4, \underline{1}, 8, 4)$ then $R(q) = (2, 1, 4, 3)$, since

1. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, 1, \square, \square)$.
2. $q = (4, \underline{1}, 8, 4) \implies R(q) = (2, \square, \square, 3)$.
3. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, \square, 4, \square)$.

Queue ranks

- For $q \in \hat{\mathcal{Q}}$, rank the queues from 1 to n so that each arriving customer joins the lowest ranked queue amongst his/her choices. Let $R(q,j)$ be the rank of queue j in state q, and let

$$R(q) := (R(q,1), \ldots, R(q,n)).$$

Example

If $q = (4, \underline{1}, 8, 4)$ then $R(q) = (2, 1, 4, 3)$, since

1. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, 1, \square, \square)$.
2. $q = (4, \underline{1}, 8, 4) \implies R(q) = (2, \square, \square, 3)$.
3. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, \square, 4, \square)$.

Queue ranks

- For $q \in \hat{\mathcal{Q}}$, rank the queues from 1 to n so that each arriving customer joins the lowest ranked queue amongst his/her choices. Let $R(q,j)$ be the rank of queue j in state q, and let

$$R(q) := (R(q,1), ..., R(q,n)).$$

Example

If $q = (4, \underline{1}, 8, 4)$ then $R(q) = (2, 1, 4, 3)$, since

1. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, 1, \square, \square)$.
2. $q = (4, \underline{1}, 8, 4) \implies R(q) = (2, \square, \square, 3)$.
3. $q = (4, \underline{1}, 8, 4) \implies R(q) = (\square, \square, 4, \square)$.

Lengths processes

- Let $\mathbf{Q} = (Q_t)_{t \geq 0}$ be a copy of the supermarket model with memory. This is a stochastic process on $\hat{\mathcal{Q}}$; call this a *lengths process*. It is easy to see that lengths processes are Markov.

Example

If $\mathbf{Q}$ has initial state $q = (5, 17, 20, 14, 6, \underline{6}, 14, 11)$, then writing T_r for the time of the rth event ($r \geq 1$), a sample path might look like:

```
Q_0     = ( 5 17 20 14  6  6 14 11)
Q_{T_1} = ( 5 17 20 13  6  6 14 11)
Q_{T_2} = ( 5 17 20 13  6  6 14 10)
Q_{T_3} = ( 5 17 20 13  6  7 14 10)
Q_{T_4} = ( 5 17 20 13  6  8 14 10)
Q_{T_5} = ( 5 16 20 13  6  8 14 10)
Q_{T_6} = ( 5 16 19 13  6  8 14 10)
                                ...
```

Lengths processes

- Let $\mathbf{Q} = (Q_t)_{t \geq 0}$ be a copy of the supermarket model with memory. This is a stochastic process on $\hat{\mathcal{Q}}$; call this a *lengths process*. It is easy to see that lengths processes are Markov.

Example

If $\mathbf{Q}$ has initial state $q = (5, 17, 20, 14, 6, \underline{6}, 14, 11)$, then writing T_r for the time of the rth event $(r \geq 1)$, a sample path might look like:

```
Q_0     = ( 5 17 20 14  6  6 14 11)
Q_{T_1} = ( 5 17 20 13  6  6 14 11)
Q_{T_2} = ( 5 17 20 13  6  6 14 10)
Q_{T_3} = ( 5 17 20 13  6  7 14 10)
Q_{T_4} = ( 5 17 20 13  6  8 14 10)
Q_{T_5} = ( 5 16 20 13  6  8 14 10)
Q_{T_6} = ( 5 16 19 13  6  8 14 10)
                               ...
```

What do we study about $\mathbf{Q}$?

- $\mathbf{Q}$ converges to its unique stationary distribution π. We are interested in how fast this convergence is.
    - Justification of existence coming up...
    - Precise definitions coming up...

The model

Coupling, Markov chain mixing and path coupling

The profile coupling

Further work

Coupling of Markov chains

- A *Markovian coupling* of Markov chains on Ω with transition matrix M is a Markov process $((X_t, X'_t))_{t\geq 0}$ on $\Omega \times \Omega$ such that
  1. $\mathbf{X} = (X_t)_{t\geq 0}$ and $\mathbf{X}' = (X'_t)_{t\geq 0}$ are both Markov chains on Ω with transition matrix M, and
  2. if $X_s = X'_s$ for some $s \geq 0$, then $X_t = X'_t$ for all $t \geq s$.

- Note that the initial distributions of $\mathbf{X}$ and $\mathbf{X}'$ maybe arbitrary.

Coupling of Markov chains

- A *Markovian coupling* of Markov chains on Ω with transition matrix M is a Markov process $((X_t, X'_t))_{t \geq 0}$ on $\Omega \times \Omega$ such that
  1. $\mathbf{X} = (X_t)_{t \geq 0}$ and $\mathbf{X}' = (X'_t)_{t \geq 0}$ are both Markov chains on Ω with transition matrix M, and
  2. if $X_s = X'_s$ for some $s \geq 0$, then $X_t = X'_t$ for all $t \geq s$.

  - Note that the initial distributions of $\mathbf{X}$ and $\mathbf{X}'$ maybe arbitrary.

Example of a coupling

Example (Simple random walk on $\{0, 1, \ldots, n\}$)

If ♠ and ♣ are independent, a sample path might look like:

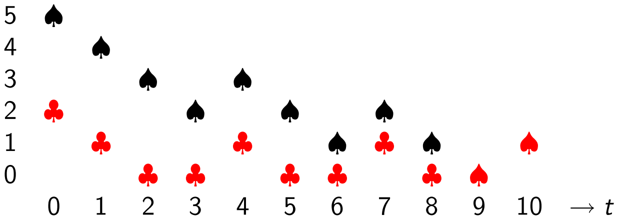
5	♠										
4		♠									♣
3			♠		♠					♣	
2	♣			♠		♠		♠	♣		
1		♣		♣		♣	♠	♣	♠		♠
0			♣		♣		♣			♠	
	0	1	2	3	4	5	6	7	8	9	10 $\to t$

(Move down with probability $\frac{1}{2}$, doing nothing at state 0. Move up with probability $\frac{1}{2}$, doing nothing at state n.)

Example of a coupling

Example (Simple random walk on $\{0, 1, ..., n\}$)

Consider letting ♠ walk randomly as usual, but make ♣ walk in the same direction as ♠. Then they will eventually meet (here at $t = 9$):



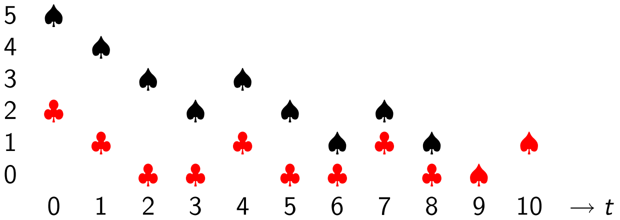
- We make deductions like: if $x \leq y$, then

$$M^t(y, 0) = \mathbb{P}(♠_t = 0) \leq \mathbb{P}(♣_t = 0) = M^t(x, 0).$$

Example of a coupling

Example (Simple random walk on $\{0, 1, \dots, n\}$)

Consider letting ♠ walk randomly as usual, but make ♣ walk in the same direction as ♠. Then they will eventually meet (here at $t = 9$):



- We make deductions like: if $x \leq y$, then

$$M^t(y, 0) = \mathbb{P}(♠_t = 0) \leq \mathbb{P}(♣_t = 0) = M^t(x, 0).$$

How we couple the supermarket model(s)

- Couple $\mathbf{Q}$ and $\mathbf{Q}'$ so they share event times (a common source of randomness).
  - So a single Poisson process of rate λn (resp. n) gives all arrival (resp. potential departure) times.
- Make random choices for $\mathbf{Q}$ as usual:

$$\left(C_T^1, \ldots, C_T^d\right) \text{ for arrivals,} \quad C_T \text{ for departures.}$$

- Make choices for $\mathbf{Q}'$ based on those made for $\mathbf{Q}$. In particular, for each event time T, construct a permutation α_T on $\{1, \ldots, n\}$. Then

$$\left(\alpha_T\left(C_T^1\right), \ldots, \alpha_T\left(C_T^d\right)\right) \text{ for arrivals,} \quad \alpha_T\left(C_T\right) \text{ for departures.}$$

As required, these are still uniformly random choices.

How we couple the supermarket model(s)

- Couple $\mathbf{Q}$ and $\mathbf{Q}'$ so they share event times (a common source of randomness).
  - So a single Poisson process of rate λn (resp. n) gives all arrival (resp. potential departure) times.
- Make random choices for $\mathbf{Q}$ as usual:

$$\left(C_T^1, \ldots, C_T^d\right) \text{ for arrivals, } \quad C_T \text{ for departures.}$$

- Make choices for $\mathbf{Q}'$ based on those made for $\mathbf{Q}$. In particular, for each event time T, construct a permutation α_T on $\{1, \ldots, n\}$. Then

$$\left(\alpha_T\left(C_T^1\right), \ldots, \alpha_T\left(C_T^d\right)\right) \text{ for arrivals, } \quad \alpha_T\left(C_T\right) \text{ for departures.}$$

  As required, these are still uniformly random choices.

How we couple the supermarket model(s)

- Couple $\mathbf{Q}$ and $\mathbf{Q}'$ so they share event times (a common source of randomness).
  - So a single Poisson process of rate λn (resp. n) gives all arrival (resp. potential departure) times.
- Make random choices for $\mathbf{Q}$ as usual:

$$\left(C_T^1, \ldots, C_T^d\right) \text{ for arrivals, } \quad C_T \text{ for departures.}$$

- Make choices for $\mathbf{Q}'$ based on those made for $\mathbf{Q}$. In particular, for each event time T, construct a permutation α_T on $\{1, \ldots, n\}$. Then

$$\left(\alpha_T\left(C_T^1\right), \ldots, \alpha_T\left(C_T^d\right)\right) \text{ for arrivals, } \quad \alpha_T\left(C_T\right) \text{ for departures.}$$

As required, these are still uniformly random choices.

Why the stationary distribution exists

- Let $\mathbf{Q}$ be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.

  Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, \ldots, n\}$.
- Then $\mathbf{Q}$ is *at least* as well-behaved as $\mathbf{Q}'$.

  E.g., if the customer in $\mathbf{Q}'$ joins the shortest queue (rank 1), then so does his/her counterpart in $\mathbf{Q}$.
- So $\mathbf{Q}$ has a unique stationary distribution π, to which it converges.

  But what does this mean?

Why the stationary distribution exists

- Let $\mathbf{Q}$ be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.
  - Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, \ldots, n\}$.
- Then $\mathbf{Q}$ is *at least* as well-behaved as $\mathbf{Q}'$.
- So $\mathbf{Q}$ has a unique stationary distribution π, to which it converges.
  - But what does this mean?

Why the stationary distribution exists

- Let $\mathbf{Q}$ be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.
  - Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, ..., n\}$.
- Then $\mathbf{Q}$ is *at least* as well-behaved as $\mathbf{Q}'$.
  - E.g., if the customer in $\mathbf{Q}'$ joins the shortest queue (rank 1), then so does his/her counterpart in $\mathbf{Q}$.
- So $\mathbf{Q}$ has a unique stationary distribution π, to which it converges.

  But what does this mean?

Why the stationary distribution exists

- Let $\mathbf{Q}$ be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.
  - Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, ..., n\}$.
- Then $\mathbf{Q}$ is *at least* as well-behaved as $\mathbf{Q}'$.
  - E.g., if the customer in $\mathbf{Q}'$ joins the shortest queue (rank 1), then so does his/her counterpart in $\mathbf{Q}$.
- So $\mathbf{Q}$ has a unique stationary distribution π, to which it converges.
  - But what does this mean?

Why the stationary distribution exists

- Let **Q** be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.
  - Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, ..., n\}$.
- Then **Q** is *at least* as well-behaved as $\mathbf{Q}'$.
  - E.g., if the customer in $\mathbf{Q}'$ joins the shortest queue (rank 1), then so does his/her counterpart in **Q**.
- So **Q** has a unique stationary distribution π, to which it converges.
  - But what does this mean?

Why the stationary distribution exists

- Let $\mathbf{Q}$ be a lengths process for the supermarket model with memory as usual. Let $\mathbf{Q}'$ be one for the supermarket model (*without memory*) with $d = 1$ arrival choice.
  - So $\mathbf{Q}'$ is a system of n independent M/M/1 queues. They are stable since birth rate $= \lambda n < n =$ death rate.
  - So $\mathbf{Q}'$ has a well-known, unique stationary distribution.
- For each event T, let ρ_T be the permutation bijecting between queues of equal rank.
  - Formally, let ρ_T satisfy $R(Q_T, j) = R(Q'_T, \rho_T(j))$ for all $j \in \{1, ..., n\}$.
- Then $\mathbf{Q}$ is *at least* as well-behaved as $\mathbf{Q}'$.
  - E.g., if the customer in $\mathbf{Q}'$ joins the shortest queue (rank 1), then so does his/her counterpart in $\mathbf{Q}$.
- So $\mathbf{Q}$ has a unique stationary distribution π, to which it converges.
  - But what does this mean?

Total variation distance

- *Total variation distance* is a metric on the space of distributions on Ω. For distributions μ, ν on Ω,

$$d_{\mathsf{TV}}(\mu, \nu) = \max\{|\mu(A) - \nu(A)| : A \subseteq \Omega\}.$$

- Then '$\mathbf{Q}^{(n)}$ converges to $\pi^{(n)}$' means

$$d_{\mathsf{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \to 0 \text{ as } t \to \infty, \quad \forall q \in \hat{\mathcal{Q}},$$

where $\mathcal{L}\left(Q_t^{(n)}, q\right)$ is the law of $Q_t^{(n)}$ given initial state q.

  - Recall: the law $\mathcal{L}(X)$ of an V-valued random variable X is the distribution $v \mapsto \mathbb{P}\left(X^{-1}(v)\right)$ on V.

- We are interested in the speed of this convergence as a function of n. But what does 'speed' mean?

Total variation distance

- *Total variation distance* is a metric on the space of distributions on Ω. For distributions μ, ν on Ω,

$$d_{\mathrm{TV}}(\mu, \nu) = \max\{|\mu(A) - \nu(A)| : A \subseteq \Omega\}.$$

- Then '$\mathbf{Q}^{(n)}$ converges to $\pi^{(n)}$' means

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \to 0 \text{ as } t \to \infty, \quad \forall q \in \hat{\mathcal{Q}},$$

where $\mathcal{L}\left(Q_t^{(n)}, q\right)$ is the law of $Q_t^{(n)}$ given initial state q.

  - Recall: the law $\mathcal{L}(X)$ of an V-valued random variable X is the distribution $v \mapsto \mathbb{P}\left(X^{-1}(v)\right)$ on V.

- We are interested in the speed of this convergence as a function of n. But what does 'speed' mean?

Total variation distance

- *Total variation distance* is a metric on the space of distributions on Ω. For distributions μ, ν on Ω,

$$d_{\mathsf{TV}}(\mu, \nu) = \max\{|\mu(A) - \nu(A)| : A \subseteq \Omega\}.$$

- Then '$\mathbf{Q}^{(n)}$ converges to $\pi^{(n)}$' means

$$d_{\mathsf{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \to 0 \text{ as } t \to \infty, \quad \forall q \in \hat{\mathcal{Q}},$$

  where $\mathcal{L}\left(Q_t^{(n)}, q\right)$ is the law of $Q_t^{(n)}$ given initial state q.
    - Recall: the law $\mathcal{L}(X)$ of an V-valued random variable X is the distribution $v \mapsto \mathbb{P}\left(X^{-1}(v)\right)$ on V.
- We are interested in the speed of this convergence as a function of n. But what does 'speed' mean?

Mixing times

- The *mixing time* of $\mathbf{Q}^{(n)}$ is

$$\tau^{(n)}(\varepsilon, q) := \inf\left\{t \geq 0 : d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \leq \varepsilon\right\},$$

  defined for $0 < \varepsilon \leq 1, q \in \hat{\mathcal{Q}}$.

- The mixing is *rapid* if

$$\tau^{(n)}\left(\tfrac{1}{4}, q\right) = O(\log n),$$

  for all sufficiently 'nice' initial states $q \in \hat{\mathcal{Q}}$ (made precise later).
  - $\varepsilon = \frac{1}{4}$ is canonical as it gives neater algebra.
  - Cannot require $\log n$ time for all initial states, consider (a) $q_1 = (50, 50, \underline{50})$ with *many customers* or (b) $q_2 = (100, 0, \underline{0})$ with *high maximum queue-length*.

- 'Speed' is then described by upper bounds on $\tau^{(n)}$. But how do we establish such bounds? Coupling is one way.

Mixing times

- The *mixing time* of $\mathbf{Q}^{(n)}$ is

$$\tau^{(n)}(\varepsilon, q) := \inf\left\{t \geq 0 : d_{TV}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \leq \varepsilon\right\},$$

  defined for $0 < \varepsilon \leq 1, q \in \hat{\mathcal{Q}}$.
- The mixing is *rapid* if

$$\tau^{(n)}\left(\tfrac{1}{4}, q\right) = O(\log n),$$

  for all sufficiently 'nice' initial states $q \in \hat{\mathcal{Q}}$ (made precise later).
    - $\varepsilon = \frac{1}{4}$ is canonical as it gives neater algebra.
    - Cannot require $\log n$ time for all initial states, consider (a) $q_1 = (50, 50, \underline{50})$ with *many customers* or (b) $q_2 = (100, 0, \underline{0})$ with *high maximum queue-length*.
- 'Speed' is then described by upper bounds on $\tau^{(n)}$. But how do we establish such bounds? Coupling is one way.

Mixing times

- The *mixing time* of $\mathbf{Q}^{(n)}$ is

$$\tau^{(n)}(\varepsilon, q) := \inf\left\{t \geq 0 : d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \leq \varepsilon\right\},$$

  defined for $0 < \varepsilon \leq 1, q \in \hat{\mathcal{Q}}$.

- The mixing is *rapid* if

$$\tau^{(n)}\left(\tfrac{1}{4}, q\right) = O(\log n),$$

  for all sufficiently 'nice' initial states $q \in \hat{\mathcal{Q}}$ (made precise later).
    - $\varepsilon = \frac{1}{4}$ is canonical as it gives neater algebra.
    - Cannot require $\log n$ time for all initial states, consider (a) $q_1 = (50, 50, \underline{50})$ with *many customers* or (b) $q_2 = (100, 0, \underline{0})$ with *high maximum queue-length*.

- 'Speed' is then described by upper bounds on $\tau^{(n)}$. But how do we establish such bounds? Coupling is one way.

Mixing times

- The *mixing time* of $\mathbf{Q}^{(n)}$ is

$$\tau^{(n)}(\varepsilon, q) := \inf\left\{t \geq 0 : d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t^{(n)}, q\right), \pi^{(n)}\right) \leq \varepsilon\right\},$$

defined for $0 < \varepsilon \leq 1, q \in \hat{\mathcal{Q}}$.

- The mixing is *rapid* if

$$\tau^{(n)}\left(\tfrac{1}{4}, q\right) = O(\log n),$$

for all sufficiently 'nice' initial states $q \in \hat{\mathcal{Q}}$ (made precise later).

  - $\varepsilon = \frac{1}{4}$ is canonical as it gives neater algebra.
  - Cannot require $\log n$ time for all initial states, consider (a) $q_1 = (50, 50, \underline{50})$ with *many customers* or (b) $q_2 = (100, 0, \underline{0})$ with *high maximum queue-length*.

- 'Speed' is then described by upper bounds on $\tau^{(n)}$. But how do we establish such bounds? Coupling is one way.

The coupling inequality

Theorem (Coupling inequality – Aldous, 1983)

For a coupling $(\mathbf{X}, \mathbf{X}')$*, all* $t \geq 0$ *and* $x, x' \in \Omega$*,*

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(X_t, x\right), \mathcal{L}\left(X_t', x'\right)\right) \leq \mathbb{P}\left(X_t \neq X_t' \mid X_0 = x, X_0' = x'\right).$$

- Note that different couplings give different bounds.
- So if $(\mathbf{Q}, \mathbf{Q}')$ is a coupling of the *lengths process* with $\mathbf{Q}'$ in equilibrium, then

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t, q\right), \pi\right) \leq \mathbb{P}\left(Q_t \neq Q_t' \mid Q_0 = q\right),$$

  for all $t \geq 0, q \in \hat{\mathcal{Q}}$.
- In general, constructing a coupling (let alone a good one) is quite difficult. Path coupling makes this easier.

The coupling inequality

Theorem (Coupling inequality – Aldous, 1983)

For a coupling $(\mathbf{X}, \mathbf{X}')$, *all* $t \geq 0$ *and* $x, x' \in \Omega$,

$$d_{TV}\left(\mathcal{L}\left(X_t, x\right), \mathcal{L}\left(X_t', x'\right)\right) \leq \mathbb{P}\left(X_t \neq X_t' \mid X_0 = x, X_0' = x'\right).$$

- Note that different couplings give different bounds.
- So if $(\mathbf{Q}, \mathbf{Q}')$ is a coupling of the *lengths process* with $\mathbf{Q}'$ in equilibrium, then

$$d_{TV}\left(\mathcal{L}\left(Q_t, q\right), \pi\right) \leq \mathbb{P}\left(Q_t \neq Q_t' \mid Q_0 = q\right),$$

  for all $t \geq 0, q \in \hat{\mathcal{Q}}$.
- In general, constructing a coupling (let alone a good one) is quite difficult. Path coupling makes this easier.

The coupling inequality

Theorem (Coupling inequality – Aldous, 1983)

For a coupling $(\mathbf{X}, \mathbf{X}')$, *all* $t \geq 0$ *and* $x, x' \in \Omega$,

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(X_t, x\right), \mathcal{L}\left(X_t', x'\right)\right) \leq \mathbb{P}\left(X_t \neq X_t' \mid X_0 = x, X_0' = x'\right).$$

- Note that different couplings give different bounds.
- So if $(\mathbf{Q}, \mathbf{Q}')$ is a coupling of the *lengths process* with $\mathbf{Q}'$ in equilibrium, then

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t, q\right), \pi\right) \leq \mathbb{P}\left(Q_t \neq Q_t' \mid Q_0 = q\right),$$

  for all $t \geq 0, q \in \hat{\mathcal{Q}}$.
- In general, constructing a coupling (let alone a good one) is quite difficult. Path coupling makes this easier.

The coupling inequality

Theorem (Coupling inequality – Aldous, 1983)

For a coupling $(\mathbf{X}, \mathbf{X}')$, *all* $t \geq 0$ *and* $x, x' \in \Omega$,

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(X_t, x\right), \mathcal{L}\left(X_t', x'\right)\right) \leq \mathbb{P}\left(X_t \neq X_t' \mid X_0 = x, X_0' = x'\right).$$

- Note that different couplings give different bounds.
- So if $(\mathbf{Q}, \mathbf{Q}')$ is a coupling of the *lengths process* with $\mathbf{Q}'$ in equilibrium, then

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(Q_t, q\right), \pi\right) \leq \mathbb{P}\left(Q_t \neq Q_t' \mid Q_0 = q\right),$$

  for all $t \geq 0, q \in \hat{\mathcal{Q}}$.
- In general, constructing a coupling (let alone a good one) is quite difficult. Path coupling makes this easier.

Path coupling

Theorem (Path coupling – Bubley and Dyer, 1997)

Let Ω be the vertex set of some connected graph with the graph metric ρ. If there exists $\alpha \leq 1$ and a coupling $(\mathbf{X}, \mathbf{X}')$ such that

$$\mathbb{E}\left[\rho\left(X_{t+1}, X'_{t+1}\right) \mid X_t, X'_t\right] \leq \alpha\rho\left(X_t, X'_t\right)$$

whenever $\rho\left(X_t, X'_t\right) = 1$, for all $t \geq 0$, then the coupling can extended to $\Omega \times \Omega$.

- Recall: the graph metric $\rho\left(x, x'\right)$ gives the length of the shortest path from x to x'.
- Key: we only need to consider pairs (x, x') which are edges, and not arbitrary pairs.

Path coupling

Theorem (Path coupling – Bubley and Dyer, 1997)

Let Ω be the vertex set of some connected graph with the graph metric ρ. If there exists $\alpha \leq 1$ and a coupling $(\mathbf{X}, \mathbf{X}')$ such that

$$\mathbb{E}\left[\rho\left(X_{t+1}, X'_{t+1}\right) \mid X_t, X'_t\right] \leq \alpha\rho\left(X_t, X'_t\right)$$

whenever $\rho\left(X_t, X'_t\right) = 1$, for all $t \geq 0$, then the coupling can extended to $\Omega \times \Omega$.

- Recall: the graph metric $\rho\left(x, x'\right)$ gives the length of the shorest path from x to x'.
- Key: we only need to consider pairs (x, x') which are edges, and not arbitrary pairs.

Two stage coupling

- We couple $\mathbf{Q}$ and $\mathbf{Q}'$ in two stages:
  1. first so that their *profiles* agree (defined next slide)
  2. then so that they themselves agree.
- Multi-stage coupling is a commonly used technique.
- We only present the *profile coupling*.

Two stage coupling

- We couple $\mathbf{Q}$ and $\mathbf{Q}'$ in two stages:
  1. first so that their *profiles* agree (defined next slide)
  2. then so that they themselves agree.
- Multi-stage coupling is a commonly used technique.
- We only present the *profile coupling*.

Two stage coupling

- We couple $\mathbf{Q}$ and $\mathbf{Q}'$ in two stages:
  1. first so that their *profiles* agree (defined next slide)
  2. then so that they themselves agree.
- Multi-stage coupling is a commonly used technique.
- We only present the *profile coupling*.

Two stage coupling

- We couple $\mathbf{Q}$ and $\mathbf{Q}'$ in two stages:
  1. first so that their *profiles* agree (defined next slide)
  2. then so that they themselves agree.
- Multi-stage coupling is a commonly used technique.
- We only present the *profile coupling*.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q, r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h, i)$ is the pair

$$\Pr(q) := ((L(q,1), L(q,2), \ldots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$, since

$q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \ldots), \square)$.

$q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \ldots), \square)$.

$q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \ldots), \square)$.

$q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q, r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h, i)$ is the pair

$$\Pr(q) := \left(\left(L(q, 1), L(q, 2), \ldots\right), h(i)\right).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \ldots), \square)$.
   $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \ldots), \square)$.
   $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \ldots), \square)$.
   $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q,r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h,i)$ is the pair

$$\Pr(q) := ((L(q,1), L(q,2), \dots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4,3,2,2,0,0,0,\dots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \dots), \square)$.
2. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \dots), \square)$.
3. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \dots), \square)$.
4. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \dots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q, r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h, i)$ is the pair

$$\Pr(q) := ((L(q,1), L(q,2), \ldots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \ldots), \square)$.
2. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \ldots), \square)$.
3. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \ldots), \square)$.
4. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q,r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h,i)$ is the pair

$$\Pr(q) := ((L(q,1), L(q,2), \dots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4,3,2,2,0,0,0,\dots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \dots), \square)$.
2. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \dots), \square)$.
3. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \dots), \square)$.
4. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \dots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q, r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h, i)$ is the pair

$$\Pr(q) := ((L(q, 1), L(q, 2), \ldots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \ldots), \square)$.
2. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \ldots), \square)$.
3. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \ldots), \square)$.
4. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$.

Profile vectors

- Informally: profiles capture queue-lengths information but not queue-order information.
- For $q \in \hat{\mathcal{Q}}$, let $L(q, r)$ be the number of queues of length $\geq r$ in q. Then the *profile* of $q = (h, i)$ is the pair

$$\Pr(q) := ((L(q,1), L(q,2), \ldots), h(i)).$$

  - Note that $h(i)$ is the length of the memory queue in q.

Example

If $q = (4, \underline{1}, 3, 4)$ then $\Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$, since

1. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, \square, \square, \square, \ldots), \square)$.
2. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, \square, \square, \ldots), \square)$.
3. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, \ldots), \square)$.
4. $q = (4, \underline{1}, 2, 4) \implies \Pr(q) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$.

Profile processes

- From last slide, the profile vector

$$\Pr(q) = (l, m) = ((4, 3, 2, 2, 0, 0, 0, \ldots), 1)$$

is an element of the *profiles space*

$$\hat{\mathcal{P}} := \left\{ (l, m) \in \mathbb{Z}_{\geq}^{\mathbb{N}} \times \mathbb{Z}_{\geq} : \begin{array}{l} n \geq l(1) \geq l(2) \geq \ldots \geq 0, \\ l(r) > 0 \text{ for finitely many } r, \\ l(m) - l(m+1) \geq 1 \\ \quad \text{with the convention } l(0) = n \end{array} \right\}.$$

- The *profile* of $\mathbf{Q} = (Q_t)_{t \geq 0}$ is the process $\mathbf{P} = (\Pr(Q_t))_{t \geq 0}$. This is a stochastic process on $\hat{\mathcal{P}}$, and can be shown to also be Markov.

Profile processes

- From last slide, the profile vector

$$\Pr(q) = (l, m) = ((4,3,2,2,0,0,0,\dots), 1)$$

is an element of the *profiles space*

$$\hat{\mathcal{P}} := \left\{ (l, m) \in \mathbb{Z}_{\geq}^{\mathbb{N}} \times \mathbb{Z}_{\geq} : \begin{array}{l} n \geq l(1) \geq l(2) \geq \dots \geq 0, \\ l(r) > 0 \text{ for finitely many } r, \\ l(m) - l(m+1) \geq 1 \\ \quad \text{with the convention } l(0) = n \end{array} \right\}.$$

- The *profile* of $\mathbf{Q} = (Q_t)_{t \geq 0}$ is the process $\mathbf{P} = (\Pr(Q_t))_{t \geq 0}$. This is a stochastic process on $\hat{\mathcal{P}}$, and can be shown to also be Markov.

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

$Q_T = (1, \ , 2, 7), Q'_T = (2, 5, \ , 4) \implies \phi_T(\ ) =$

$Q_T = (\ , \ , 2, 7), Q'_T = (\ , 5, \ , 4) \implies \phi_T(\ ) =$

$Q_T = (\ , \ , \ , 7), Q'_T = (\ , 5, \ , \ ) \implies \phi_T(\ ) =$

$Q_T = (\ , \ , \ , \ ), Q'_T = (\ , \ , \ , \ ) \implies \phi_T(\ ) =$

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

$$Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3.$$
$$Q_T = (1, \;, 2, 7)\,, Q'_T = (2, 5, \;, 4) \implies \phi_T(1) = 1.$$
$$Q_T = (\;, \;, 2, 7)\,, Q'_T = (\;, 5, \;, 4) \implies \phi_T(3) = 4.$$
$$Q_T = (\;, \;, \;, 7)\,, Q'_T = (\;, 5, \;, \;) \implies \phi_T(4) = 2.$$

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)$, $Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)$, $Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3$.
   $Q_T = (1, \ , 2, 7)$, $Q'_T = (2, 5, \ , 4) \implies \phi_T(1) = 1$.
   $Q_T = (\ , \ , 2, 7)$, $Q'_T = (\ , 5, \ , 4) \implies \phi_T(3) = 4$.
   $Q_T = (\ , \ , \ , 7)$, $Q'_T = (\ , 5, \ , \ ) \implies \phi_T(4) = 2$.

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3$.
2. $Q_T = (1, \ \ , 2, 7)\,, Q'_T = (2, 5, \ \ , 4) \implies \phi_T(1) = 1$.
3. $Q_T = (\ \ , \ \ , 2, 7)\,, Q'_T = (\ \ , 5, \ \ , 4) \implies \phi_T(3) = 4$.
4. $Q_T = (\ \ , \ \ , \ \ , 7)\,, Q'_T = (\ \ , 5, \ \ , \ \ ) \implies \phi_T(4) = 2$.

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T\,(2) = 3.$
2. $Q_T = (1, \ \ , 2, 7)\,, Q'_T = (2, 5, \ \ , 4) \implies \phi_T\,(1) = 1.$
3. $Q_T = (\ \ , \ \ , 2, 7)\,, Q'_T = (\ \ , 5, \ \ , 4) \implies \phi_T\,(3) = 4.$
4. $Q_T = (\ \ , \ \ , \ \ , 7)\,, Q'_T = (\ \ , 5, \ \ , \ \ ) \implies \phi_T\,(4) = 2.$

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3.$
2. $Q_T = (1, \ \ , 2, 7)\,, Q'_T = (2, 5, \ \ , 4) \implies \phi_T(1) = 1.$
3. $Q_T = (\ , \ \ , 2, 7)\,, Q'_T = (\ , 5, \ \ , 4) \implies \phi_T(3) = 4.$
4. $Q_T = (\ , \ \ , \ \ , 7)\,, Q'_T = (\ , 5, \ \ , \ ) \implies \phi_T(4) = 2.$

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)$, $Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)$, $Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3$.
2. $Q_T = (1, \ , 2, 7)$, $Q'_T = (2, 5, \ , 4) \implies \phi_T(1) = 1$.
3. $Q_T = (\ , \ , 2, 7)$, $Q'_T = (\ , 5, \ , 4) \implies \phi_T(3) = 4$.
4. $Q_T = (\ , \ , \ , 7)$, $Q'_T = (\ , 5, \ , \ ) \implies \phi_T(4) = 2$.

The profile coupling

- As before: couple $\mathbf{P}$ and $\mathbf{P}'$ so they share event times. Make random choices for $\mathbf{P}$, then dependent choices for $\mathbf{P}'$.
- Transitions will be described in $\hat{\mathcal{Q}}$. The choice of $q \in \hat{\mathcal{Q}}$ to represent a profile will not matter.
- For each event T, let ϕ_T be the permutation defined as follows. Let ϕ_T biject between the memory queues, then pair off the remaining queues by rank (lowest up).

Example

If $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4)$ then $\phi_T = (2, 3, 4)$, since

1. $Q_T = (1, \underline{4}, 2, 7)\,, Q'_T = (2, 5, \underline{3}, 4) \implies \phi_T(2) = 3$.
2. $Q_T = (1, \ , 2, 7)\,, Q'_T = (2, 5, \ , 4) \implies \phi_T(1) = 1$.
3. $Q_T = (\ , \ , 2, 7)\,, Q'_T = (\ , 5, \ , 4) \implies \phi_T(3) = 4$.
4. $Q_T = (\ , \ , \ , 7)\,, Q'_T = (\ , 5, \ , \ ) \implies \phi_T(4) = 2$.

The neighbourhood structure

- Say $p = (l, m)$ and $p' = (l', m')$ are *adjacent*, i.e., set $\rho(p, p') = 1$, if

$$\exists! k > 0 \text{ s.t. } \begin{cases} l(r) = l'(r) - \delta_{r,k}, \ \forall r \in \mathbb{N}, & \text{and} \\ m = m' \text{ or } m = m' - 1 = k - 1. \end{cases}$$

- Must verify $\hat{\mathcal{P}}$ is connected, so that ρ is finite-valued.

Lemma (Monotonicity of distance)

Under the profile coupling, we have $\rho(P_t, P'_t) \leq \rho(P_s, P'_s)$ *for all* $0 \leq s \leq t$.

Proof (outline).

The neighbourhood structure

- Say $p = (l, m)$ and $p' = (l', m')$ are *adjacent*, i.e., set $\rho(p, p') = 1$, if

$$\exists! k > 0 \text{ s.t. } \begin{cases} l(r) = l'(r) - \delta_{r,k}, \ \forall r \in \mathbb{N}, & \text{and} \\ m = m' \text{ or } m = m' - 1 = k - 1. \end{cases}$$

  - Must verify $\hat{\mathcal{P}}$ is connected, so that ρ is finite-valued.

Lemma (Monotonicity of distance)

Under the profile coupling, we have $\rho(P_t, P'_t) \leq \rho(P_s, P'_s)$ for all $0 \leq s \leq t$.

Proof (outline).

- Check the result for $\rho(P_s, P'_s) = 1$, i.e. check that adjacent states remain adjacent or coalesce.
- For $\delta := \rho(P_s, P'_s) > 1$, let $P_s = P_s^0, \ldots, P_s^\delta = P'_s$ be a path, then

$$\rho(P_t, P'_t) \leq \sum_{j=1}^{\delta} \rho\left(P_t^{j-1}, P_t^j\right) \leq \sum_{j=1}^{\delta} \rho\left(P_s^{j-1}, P_s^j\right) = \rho(P_s, P'_s).$$

The neighbourhood structure

- Say $p = (l, m)$ and $p' = (l', m')$ are *adjacent*, i.e., set $\rho(p, p') = 1$, if

$$\exists! k > 0 \text{ s.t. } \begin{cases} l(r) = l'(r) - \delta_{r,k},\ \forall r \in \mathbb{N}, & \text{and} \\ m = m' \text{ or } m = m' - 1 = k - 1. \end{cases}$$

  - Must verify $\hat{\mathcal{P}}$ is connected, so that ρ is finite-valued.

Lemma (Monotonicity of distance)

Under the profile coupling, we have $\rho(P_t, P'_t) \leq \rho(P_s, P'_s)$ for all $0 \leq s \leq t$.

Proof (outline).

- Check the result for $\rho(P_s, P'_s) = 1$, i.e. check that adjacent states remain adjacent or coalesce.
- For $\delta := \rho(P_s, P'_s) > 1$, let $P_s = P_s^0, \ldots, P_s^\delta = P'_s$ be a path, then

$$\rho(P_t, P'_t) \leq \sum_{j=1}^{\delta} \rho\left(P_t^{j-1}, P_t^j\right) \leq \sum_{j=1}^{\delta} \rho\left(P_s^{j-1}, P_s^j\right) = \rho(P_s, P'_s).$$

The neighbourhood structure

- Say $p = (l, m)$ and $p' = (l', m')$ are *adjacent*, i.e., set $\rho(p, p') = 1$, if

$$\exists! k > 0 \text{ s.t. } \begin{cases} l(r) = l'(r) - \delta_{r,k}, \; \forall r \in \mathbb{N}, & \text{and} \\ m = m' \text{ or } m = m' - 1 = k - 1. \end{cases}$$

  - Must verify $\hat{\mathcal{P}}$ is connected, so that ρ is finite-valued.

Lemma (Monotonicity of distance)

Under the profile coupling, we have $\rho(P_t, P'_t) \leq \rho(P_s, P'_s)$ *for all* $0 \leq s \leq t$.

Proof (outline).

- Check the result for $\rho(P_s, P'_s) = 1$, i.e. check that adjacent states remain adjacent or coalesce.
- For $\delta := \rho(P_s, P'_s) > 1$, let $P_s = P_s^0, \ldots, P_s^\delta = P'_s$ be a path, then

$$\rho(P_t, P'_t) \leq \sum_{j=1}^{\delta} \rho\left(P_t^{j-1}, P_t^j\right) \leq \sum_{j=1}^{\delta} \rho\left(P_s^{j-1}, P_s^j\right) = \rho(P_s, P'_s).$$

The neighbourhood structure

- Say $p = (l, m)$ and $p' = (l', m')$ are *adjacent*, i.e., set $\rho(p, p') = 1$, if

$$\exists! k > 0 \text{ s.t. } \begin{cases} l(r) = l'(r) - \delta_{r,k}, \ \forall r \in \mathbb{N}, & \text{and} \\ m = m' \text{ or } m = m' - 1 = k - 1. \end{cases}$$

  - Must verify $\hat{\mathcal{P}}$ is connected, so that ρ is finite-valued.

Lemma (Monotonicity of distance)

Under the profile coupling, we have $\rho(P_t, P'_t) \leq \rho(P_s, P'_s)$ for all $0 \leq s \leq t$.

Proof (outline).

- Check the result for $\rho(P_s, P'_s) = 1$, i.e. check that adjacent states remain adjacent or coalesce.
- For $\delta := \rho(P_s, P'_s) > 1$, let $P_s = P_s^0, \dots, P_s^\delta = P'_s$ be a path, then

$$\rho(P_t, P'_t) \leq \sum_{j=1}^{\delta} \rho\left(P_t^{j-1}, P_t^j\right) \leq \sum_{j=1}^{\delta} \rho\left(P_s^{j-1}, P_s^j\right) = \rho(P_s, P'_s).$$

Rapid mixing

Lemma

Let $c = c(\lambda) > \frac{\lambda}{1-\lambda}$, let $\mathbf{P}$ and $\mathbf{P}'$ have adjacent inital states p and p'. Then there exist $\alpha = \alpha(c), \beta = \beta(c) > 0$ such that

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\mathbf{1}_{A_n}\mathbf{1}_{B_{n,t}}\right) \leq e^{-\beta t} + 2e^{-\beta n},$$

for all $n \in \mathbb{N}, t \geq 0$. Here $A_n = \{\#\text{customers in } p \leq cn\}$ and $B_{n,t} = \{p \text{ has a queue length} \leq t/\alpha\}$.

Rapid mixing

Proof (outline).

- By *monotonicity of distance*, there is a process $\mathbf{K} = (K_t)_{t \geq 0}$ on $\mathbb{Z}_{\geq}$ such that if $K_t = 0$ then $P_t = P'_t$, and if K_t is large then K_t has drift < 0.
- Upper bound

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\right) = \mathbb{P}\left(\rho\left(P_t, P'_t\right) = 1\right) = \mathbb{P}\left(P_t \neq P'_t\right)$$

by the probability of 3 events... By time t,

1. P has not had many customers and K has not reached 0; unlikely by a technical lemma.
2. P has had many customers. Then a coupled process in equilibrium will have many customers too; unlikely.
3. K has not had many jumps. Then a Poisson random variable is far from its mean; unlikely.

□

Rapid mixing

Proof (outline).

- By *monotonicity of distance*, there is a process $\mathbf{K} = (K_t)_{t \geq 0}$ on $\mathbb{Z}_{\geq}$ such that if $K_t = 0$ then $P_t = P'_t$, and if K_t is large then K_t has drift < 0.
- Upper bound

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\right) = \mathbb{P}\left(\rho\left(P_t, P'_t\right) = 1\right) = \mathbb{P}\left(P_t \neq P'_t\right)$$

by the probability of 3 events... By time t,

1. **P** has not had many customers and **K** has not reached 0; unlikely by a technical lemma.
2. **P** has had many customers. Then a coupled process in equilibrium will have many customers too; unlikely.
3. **K** has not had many jumps. Then a Poisson random variable is far from its mean; unlikely.

□

Rapid mixing

Proof (outline).

- By *monotonicity of distance*, there is a process $\mathbf{K} = (K_t)_{t \geq 0}$ on $\mathbb{Z}_{\geq}$ such that if $K_t = 0$ then $P_t = P'_t$, and if K_t is large then K_t has drift < 0.
- Upper bound

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\right) = \mathbb{P}\left(\rho\left(P_t, P'_t\right) = 1\right) = \mathbb{P}\left(P_t \neq P'_t\right)$$

by the probability of 3 events... By time t,

1. **P** has not had many customers and **K** has not reached 0; unlikely by a technical lemma.
2. **P** has had many customers. Then a coupled process in equilibrium will have many customers too; unlikely.
3. **K** has not had many jumps. Then a Poisson random variable is far from its mean; unlikely.

□

Rapid mixing

Proof (outline).

- By *monotonicity of distance*, there is a process $\mathbf{K} = (K_t)_{t \geq 0}$ on $\mathbb{Z}_{\geq}$ such that if $K_t = 0$ then $P_t = P'_t$, and if K_t is large then K_t has drift < 0.
- Upper bound

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\right) = \mathbb{P}\left(\rho\left(P_t, P'_t\right) = 1\right) = \mathbb{P}\left(P_t \neq P'_t\right)$$

by the probability of 3 events... By time t,

1. $\mathbf{P}$ has not had many customers and $\mathbf{K}$ has not reached 0; unlikely by a technical lemma.
2. $\mathbf{P}$ has had many customers. Then a coupled process in equilibrium will have many customers too; unlikely.
3. $\mathbf{K}$ has not had many jumps. Then a Poisson random variable is far from its mean; unlikely.

□

Rapid mixing

Proof (outline).

- By *monotonicity of distance*, there is a process $\mathbf{K} = (K_t)_{t \geq 0}$ on $\mathbb{Z}_{\geq}$ such that if $K_t = 0$ then $P_t = P'_t$, and if K_t is large then K_t has drift < 0.
- Upper bound

$$\mathbb{E}\left(\rho\left(P_t, P'_t\right)\right) = \mathbb{P}\left(\rho\left(P_t, P'_t\right) = 1\right) = \mathbb{P}\left(P_t \neq P'_t\right)$$

by the probability of 3 events... By time t,

1. **P** has not had many customers and **K** has not reached 0; unlikely by a technical lemma.
2. **P** has had many customers. Then a coupled process in equilibrium will have many customers too; unlikely.
3. **K** has not had many jumps. Then a Poisson random variable is far from its mean; unlikely.

□

Rapid mixing

Theorem

Let $c = c(\lambda) > \frac{\lambda}{1-\lambda}$. *Then there exists* $\eta = \eta(c) > 0$ *such that*

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(P_t, p\right), \pi\right) \leq ne^{-\eta t} + e^{-\eta n} + \mathbb{P}\left(\overline{A_n}\right) + \mathbb{P}\left(\overline{B_{n,t}}\right),$$

for all $n \in \mathbb{N}, t \geq 0$. *Here* $A_n = \{\#\text{customers in } p \leq cn\}$ *and* $B_{n,t} = \{p \text{ has a queue length} \leq t/\alpha\}$, *and* A'_n, $B'_{n,t}$ *defined similarly.*

Proof (outline).

- Build path to apply previous lemma. Handle 'bad cases' separately.

□

Rapid mixing

Theorem

Let $c = c(\lambda) > \frac{\lambda}{1-\lambda}$. Then there exists $\eta = \eta(c) > 0$ such that

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(P_t, p\right), \pi\right) \leq ne^{-\eta t} + e^{-\eta n} + \mathbb{P}\left(\overline{A_n}\right) + \mathbb{P}\left(\overline{B_{n,t}}\right),$$

for all $n \in \mathbb{N}, t \geq 0$. Here $A_n = \{\#\text{customers in } p \leq cn\}$ and $B_{n,t} = \{p \text{ has a queue length} \leq t/\alpha\}$, and $A'_n, B'_{n,t}$ defined similarly.

Proof (outline).

- Build path to apply previous lemma. Handle ‘bad cases’ separately.

□

The model

Coupling, Markov chain mixing and path coupling

The profile coupling

Further work

Further work

- We have shown rapid mixing of $\mathbf{P}$ and $\mathbf{P}'$.
- Assuming identical profiles, we have successfully coupled $\mathbf{Q}$ and $\mathbf{Q}'$.
    - Trying to show this is rapid mixing.
- Thank you!

Further work

- We have shown rapid mixing of $\mathbf{P}$ and $\mathbf{P}'$.
- Assuming identical profiles, we have successfully coupled $\mathbf{Q}$ and $\mathbf{Q}'$.
  - Trying to show this is rapid mixing.
- Thank you!

Further work

- We have shown rapid mixing of $\mathbf{P}$ and $\mathbf{P}'$.
- Assuming identical profiles, we have successfully coupled $\mathbf{Q}$ and $\mathbf{Q}'$.
    - Trying to show this is rapid mixing.
- Thank you!