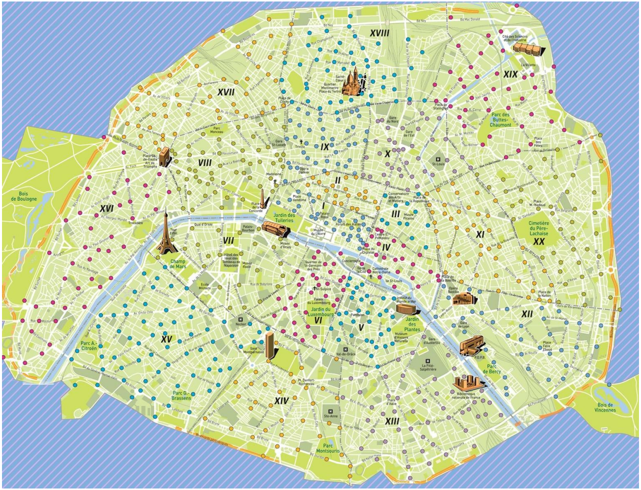
The Power of Two Choices on Graphs: the Pair-Approximation is Accurate

Nicolas Gast

Inria

ACM MAMA Workshop, June 15, 2015, Portland, Oregon

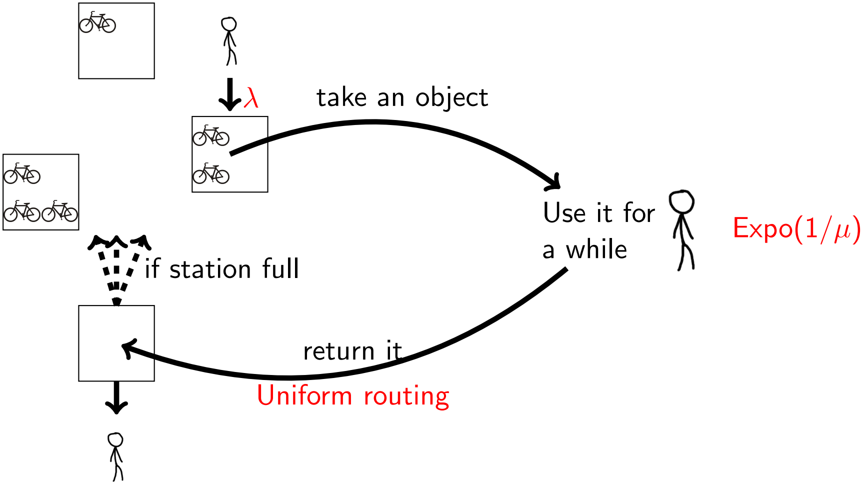
Motivating scenario is to study incentives in bike-sharing systems



Map of Velib' stations (Paris)

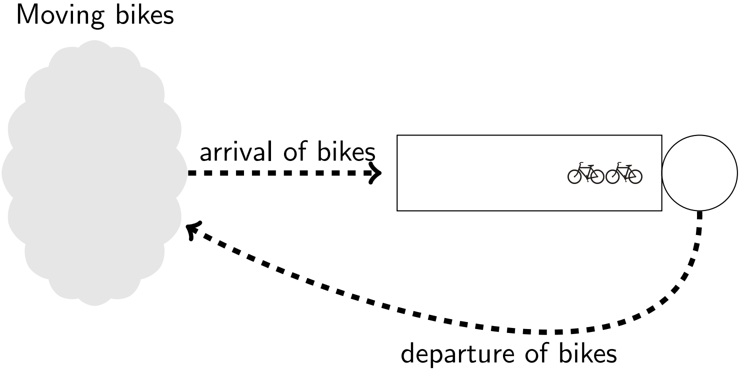
- 1200 stations
- 20k bikes

These system can be viewed as closed queuing-networks



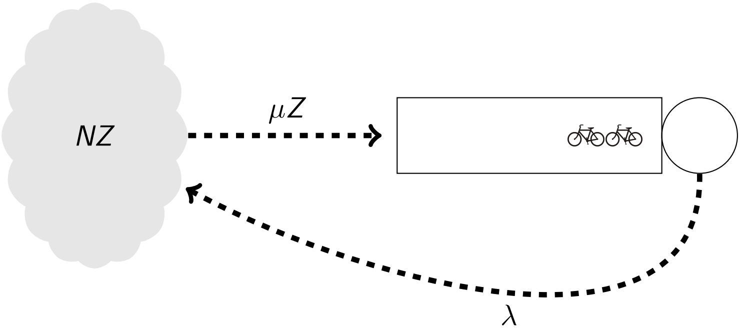
N stations, capacity C bikes per station.

When the number of stations $N \to \infty$, we can show that the model boils down to a single (open) queue.



When the number of stations $N \to \infty$, we can show that the model boils down to a single (open) queue.

Moving bikes



$$i \mapsto i+1 \text{ at rate } \mu Z \qquad (i < K)$$

$$i \mapsto i-1 \text{ at rate } \lambda \qquad (i > 0)$$

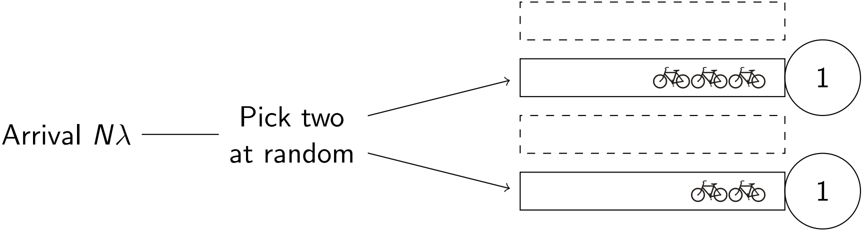
Can we improve performance?

- Even in a uniform scenario, the proportion of problematic stations (*i.e.* empty or full) is at least $1/(C+1)$.

What if a user chooses to go to a less crowded station?

In this talk, I study a generalization of the two-choice models

- N identical servers
- Exponential service time



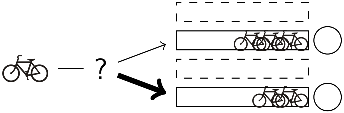
What happens when we restrict the choice to two neighbors?

Outline

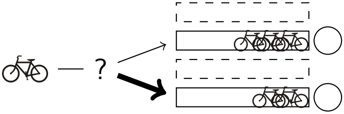
1. The classical two-choice model
2. Construction of the pair approximation equations
3. Numerical validation: the pair approximation is accurate!
4. Remarks and open questions

Outline

Two-choice rule: each incoming job/bike is routed to the least loaded of two servers picked at random.



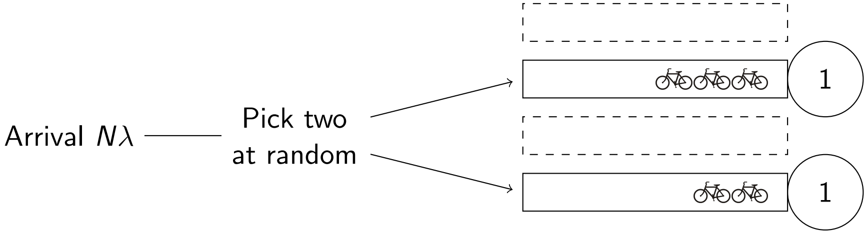
Two-choice rule: each incoming job/bike is routed to the least loaded of two servers picked at random.



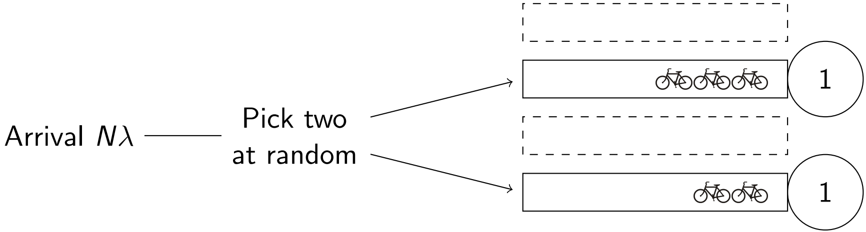
Paradigm known as "*the power of two choices*":

- Comes from balls and bills [Azar et al. 94]:
  - Throw n balls into n bins: what is the maximal number of balls in a bin?
    - $\log(n)$ if no choice
    - $\log(\log(n))$ is two choices.
- Drastic improvement of service time in server farm [Vvedenskaya 96, Mitzenmacher 96]
  - $P(\#jobs \geq i)\rho^i$ (no choice)
  - $P(\#jobs \geq i) = 2^{\lambda^{i+1}-1}$ (two choices)
- Interesting advances for non-exponential service times (Bramson 2000, Ramanan 2014)

We use mean-field to solve the two-choice equations



We use mean-field to solve the two-choice equations



Let x_j be the proportion of stations with j bikes.

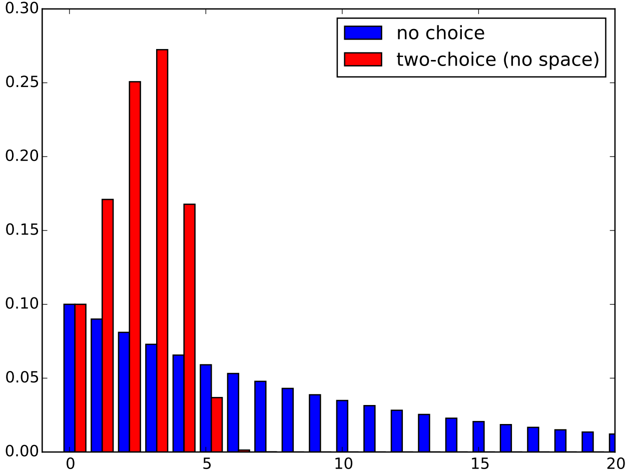
$$(i \mapsto i-1) \text{ at rate } 1$$

$$(i \mapsto i+1) \text{ at rate } \lambda(x_i + 2\sum_{j=i+1}^{\infty} x_j)$$

Note: the rate of change of x_i has to be multiplied by x_i.

With no geometry, we can solve the equation in close-form

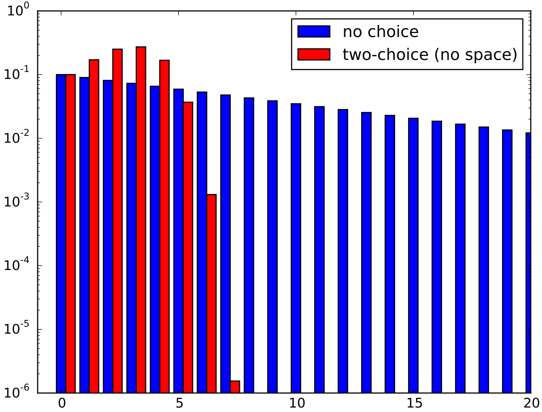
$$x_i = \lambda^{2^i} - \lambda^{2^{i+1}}$$



For bike-sharing, choosing two stations at random, decreases the number of problematic stations from $1/C$ to $\sqrt{C}2^{-C/2}$

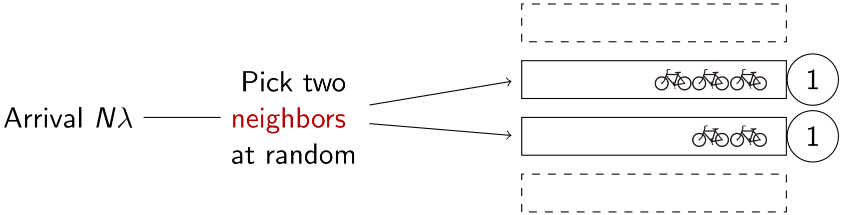
With no geometry, we can solve the equation in close-form

$$x_i = \lambda^{2^i} - \lambda^{2^{i+1}}$$



For bike-sharing, choosing two stations at random, decreases the number of problematic stations from $1/C$ to $\sqrt{C}2^{-C/2}$

What if we add geometry?

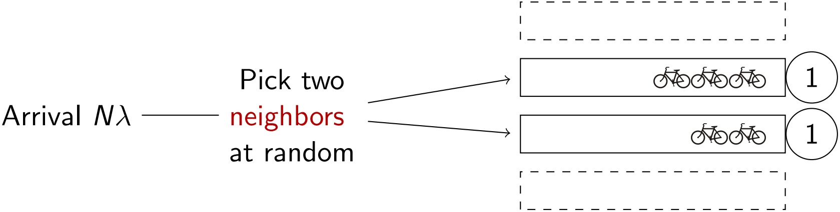


Mean field do not apply (geometry) :(.

- For balls and bins, the power of two-choice does not work (see [Kenthapadi et al. 06])
- Only numerical results?

Outline

I consider that stations are placed on a ring



Let y_{ij} be the proportion of (ordered) pairs having (i, j) jobs.

We track the proportion of (ordered) pairs (i, j)

We focus on the transitions that modify i (equations are similar for j).

$(i, j) \mapsto (i-1, j)$ at rate 1 *departure*

We track the proportion of (ordered) pairs (i, j)

We focus on the transitions that modify i (equations are similar for j).

$$(i,j) \mapsto (i-1,j) \quad \text{at rate } 1 \qquad \textit{departure}$$

$$(i,j) \mapsto (i+1,j) \quad \text{at rate } \begin{cases} \lambda & \text{if } i < j \\ \lambda/2 & \text{if } i = j \\ 0 & \text{if } i > j \end{cases} \qquad \textit{arrival on } (i,j)$$

We track the proportion of (ordered) pairs (i, j)

We focus on the transitions that modify i (equations are similar for j).

$$(i, j) \mapsto (i-1, j) \quad \text{at rate } 1 \qquad \textit{departure}$$

$$(i, j) \mapsto (i+1, j) \quad \text{at rate } \begin{cases} \lambda & \text{if } i < j \\ \lambda/2 & \text{if } i = j \\ 0 & \text{if } i > j \end{cases} \qquad \textit{arrival on } (i, j)$$

$$(i, j) \mapsto (i+1, j) \quad \text{at rate } \lambda \underbrace{\left(\frac{1}{2} z_{i,i,j} + \sum_{\ell=i+1}^{\infty} z_{\ell,i,j} \right)}_{=:p_i} / y_{ij} \qquad \textit{arrival on } (\ell, i),$$

where $z_{\ell,i,j}$ is the proportion of triplets.

We track the proportion of (ordered) pairs (i, j)

We focus on the transitions that modify i (equations are similar for j).

$$(i,j) \mapsto (i-1,j) \quad \text{at rate } 1 \qquad \textit{departure}$$

$$(i,j) \mapsto (i+1,j) \quad \text{at rate } \begin{cases} \lambda & \text{if } i<j \\ \lambda/2 & \text{if } i=j \\ 0 & \text{if } i>j \end{cases} \qquad \textit{arrival on } (i,j)$$

$$(i,j) \mapsto (i+1,j) \quad \text{at rate } \lambda \underbrace{\left(\frac{1}{2} z_{i,i,j} + \sum_{\ell=i+1}^{\infty} z_{\ell,i,j} \right)}_{=: p_i} / y_{ij} \qquad \textit{arrival on } (\ell, i),$$

where $z_{\ell,i,j}$ is the proportion of triplets.

The pair approximation is $z_{\ell,i,j} \approx y_{\ell,i} y_{i,j} / x_i$ or:

$$p_i \approx \frac{Y_{ii}/2 + \sum_{k>i} Y_{ki}}{\sum_k Y_{ki}}.$$

The pair approximation ODE is composed of four terms

Y_{ij} decreases at rate:

$$\mu Y_{ij} \qquad (departure)$$

$$\lambda Y_{i,j} \qquad (arrival\ on\ (i,j)\ when\ (i<j))$$

$$\lambda Y_{i,j}/2 \qquad (arrival\ on\ (i,i)\ when\ i=j)$$

$$\lambda p_i Y_{i,j} \qquad (arrival\ on\ neighbor)$$

The pair approximation ODE is composed of four terms

Y_{ij} decreases at rate:

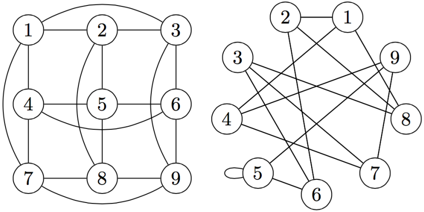
$$\mu Y_{ij} \qquad (departure)$$

$$\lambda Y_{i,j}\frac{2}{k} \qquad (arrival\ on\ (i,j)\ when\ (i<j))$$

$$\lambda Y_{i,j}/k \qquad (arrival\ on\ (i,i)\ when\ i=j)$$

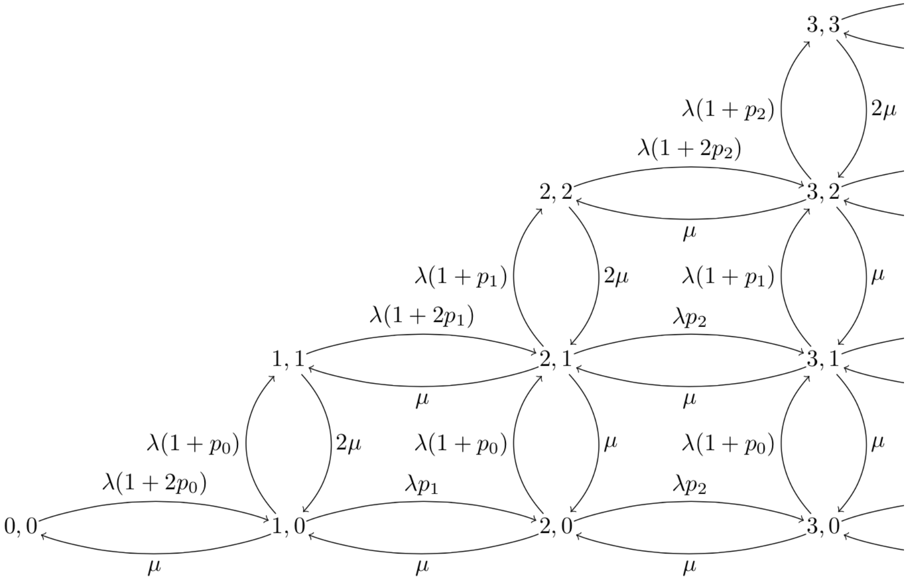
$$\lambda p_i Y_{i,j} 2\frac{k-1}{k} \qquad (arrival\ on\ neighbor)$$

The equations can be generalized to graph with fixed degree $k \geq 2$:



(a) 2D torus (b) Fixed degree $k = 3$

There is no (known) close-form for the fixed point...



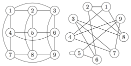
...but we can simulate the ODE!

```
for i in range(0,N):
    xi = sum(y[i]);
    if (xi>0):
        p[i] = (sum (y[i][i+1:N]) + y[i][i]/2) / xi;
for i in range(0,N):
    for j in range(0,N):
        if (i>0):
            derivative[i][j]   +=  lam*p[i-1]*y[i-1][j] - mu*y[i][j];
            derivative[i-1][j] += -lam*p[i-1]*y[i-1][j] + mu*y[i][j];
            if (i<=j):
                derivative[i][j]   +=  lam*y[i-1][j];
                derivative[i-1][j] += -lam*y[i-1][j];
            elif(i-1==j):
                derivative[i][j]   +=  lam*y[i-1][j]/2;
                derivative[i-1][j] += -lam*y[i-1][j]/2;
        if (j>0):
            derivative[i][j]   +=  lam*p[j-1]*y[i][j-1] - mu*y[i][j];
            derivative[i][j-1] += -lam*p[j-1]*y[i][j-1] + mu*y[i][j];
            if (j<=i):
                derivative[i][j]   +=  lam*y[i][j-1];
                derivative[i][j-1] += -lam*y[i][j-1];
            elif (i==j-1):
                derivative[i][j]   +=  lam*y[i][j-1]/2;
                derivative[i][j-1] += -lam*y[i][j-1]/2;
```

Outline

I compare numerically four values

Simu Simulation



(a) 2D torus (b) Fixed degree $k = 3$

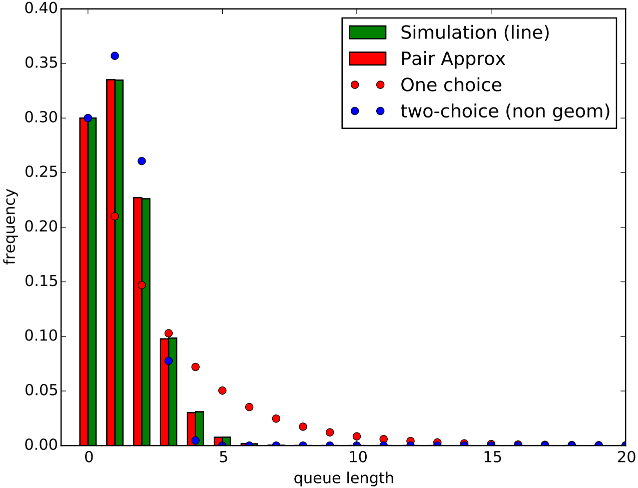
Pair-approx Fixed point of the pair-approximation ODE
ODE of size 100×100.

```
for i in range(0,N):
    xi = sum(y[i]);
    if (xi>0):
        p[i] = (sum (y[i][i+1:N]) + y[i][i]/2) / xi;
for i in range(0,N):
    for j in range(0,N):
        if (i>0):
            derivative[i][j]   +=  lam*p[i-1]*y[i-1][j] - mu*y[i][j];
            derivative[i-1][j] += -lam*p[i-1]*y[i-1][j] + mu*y[i][j];
            if (i<=j):
                derivative[i][j]   +=  lam*y[i-1][j];
                derivative[i-1][j] += -lam*y[i-1][j];
            elif(i-1==j):
                derivative[i][j]   +=  lam*y[i-1][j]/2;
                derivative[i-1][j] += -lam*y[i-1][j]/2;
        if (j>0):
            derivative[i][j]   +=  lam*p[j-1]*y[i][j-1] - mu*y[i][j];
            derivative[i][j-1] += -lam*p[j-1]*y[i][j-1] + mu*y[i][j];
            if (j<=i):
                derivative[i][j]   +=  lam*y[i][j-1];
                derivative[i][j-1] += -lam*y[i][j-1];
            elif (i==j-1):
                derivative[i][j]   +=  lam*y[i][j-1]/2;
                derivative[i][j-1] += -lam*y[i][j-1]/2;
```

No choice Theory for the M/M/1 queue $x_i = (1-\lambda)\lambda^i$

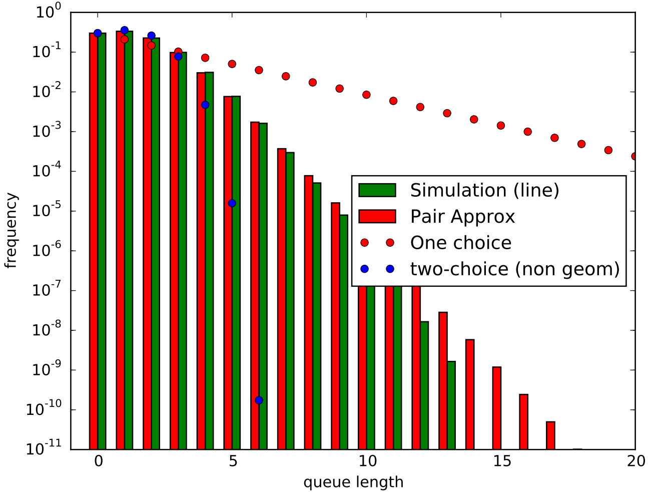
Two-choice Theory (without geometry) $x_i = \lambda^{2^i} - \lambda^{2^{i+1}}$

The fixed point of the pair-approximation is close to the system's steady-state (checked for $\lambda = .5$ to $\lambda = .99$)



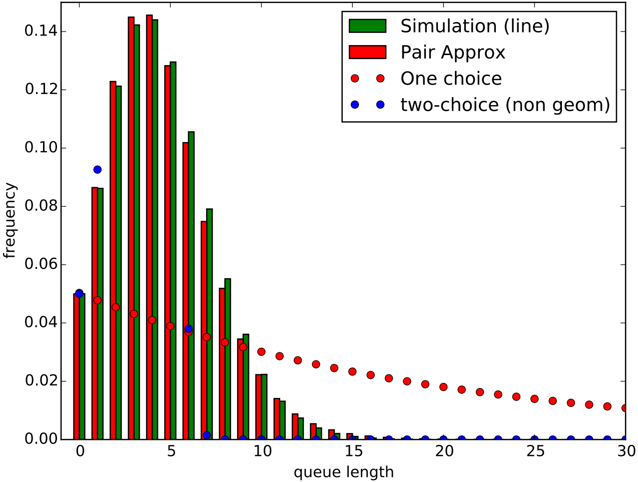
$\lambda = 0.7$

The fixed point of the pair-approximation is close to the system's steady-state (checked for $\lambda = .5$ to $\lambda = .99$)



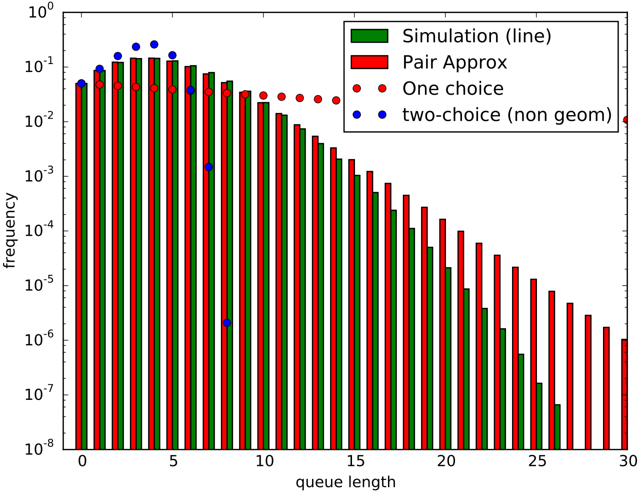
$\lambda = 0.7$

The fixed point of the pair-approximation is close to the system's steady-state (checked for $\lambda = .5$ to $\lambda = .99$)



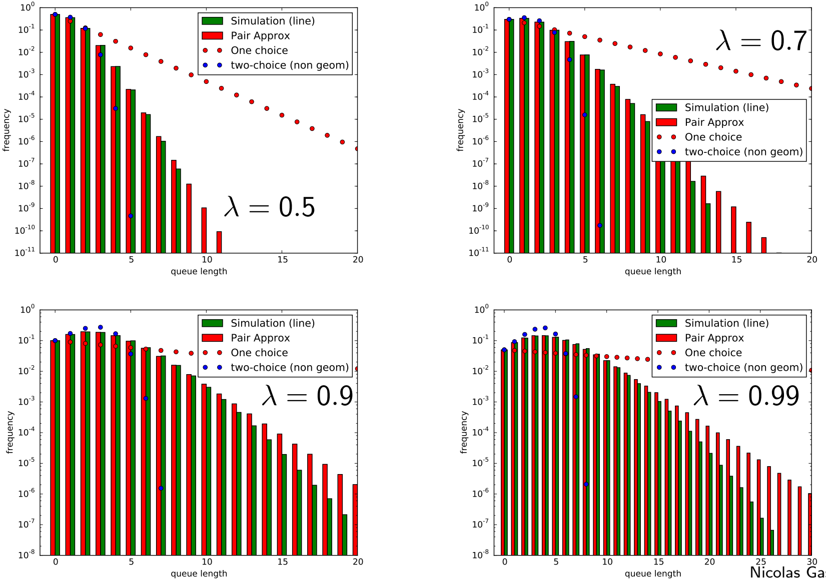
$\lambda = 0.95$

The fixed point of the pair-approximation is close to the system's steady-state (checked for $\lambda = .5$ to $\lambda = .99$)

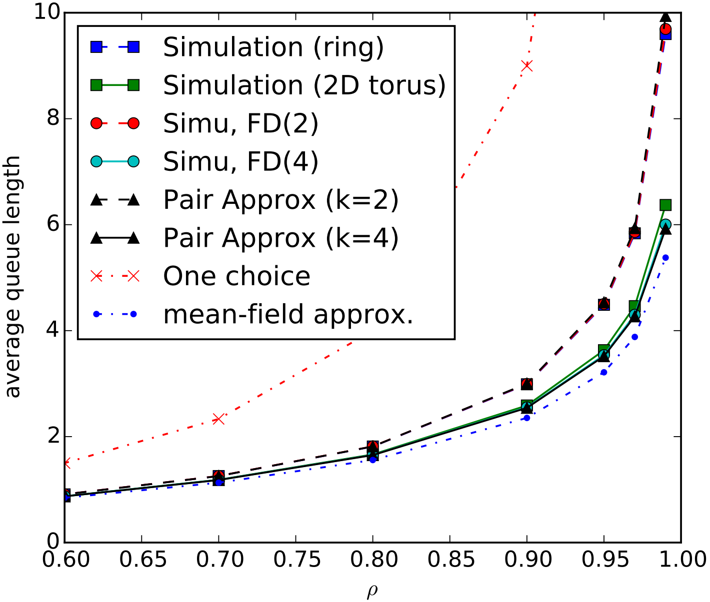


$\lambda = 0.95$

The fixed point of the pair-approximation is close to the system's steady-state (checked for $\lambda = .5$ to $\lambda = .99$)



The (steady-state) average queue length is very well approximated by pair-approximation



Outline

Recap

I study a spatial version of the two-choice model.

- Motivation comes from bike-sharing systems.
- Without geometry, the problem can be solved by using a mean-field approximation (one-choice: $\sum_{j\geq i} x_j = \lambda^i$, two-choice, $\sum_{j\geq i} x_j = \lambda^{2^i-1}$).
- Pair-approximation:
  - How to construct the equations
  - Numerically, they are very accurate

Open questions / Future work

Why does it work so well? (in some other cases, *e.g.*, SIR, it does not)	?
Is the pair approximation exact?	No
For a torus, is the decrease doubly-exponential? (recall: two-choice without geometry: $\sum_{j\geq i} x_j = \lambda^{2^i-1}$)	No?
Can we solve analytically the PA equations (or bound?)	?
Can we add heterogeneity?	seems OK
Non-exponential service time?	(maybe later)