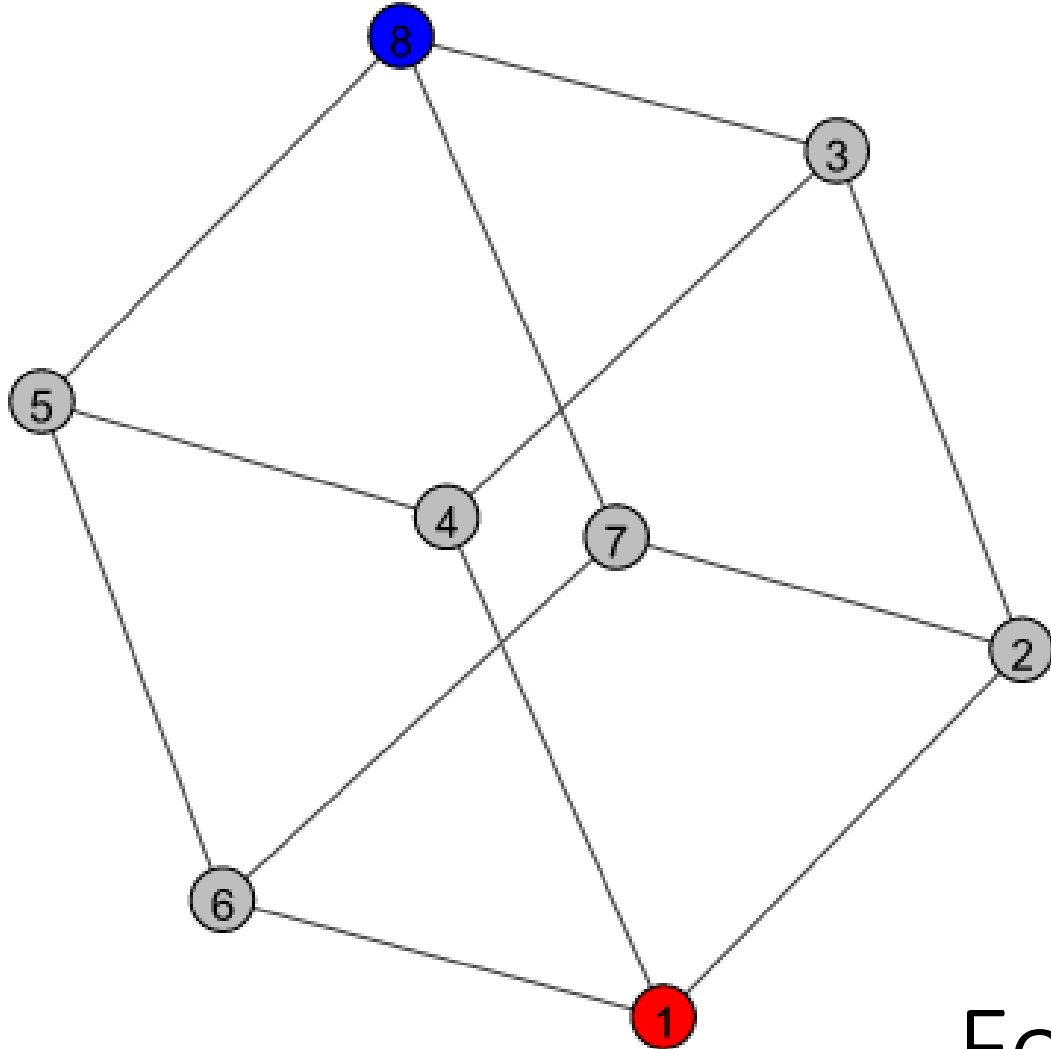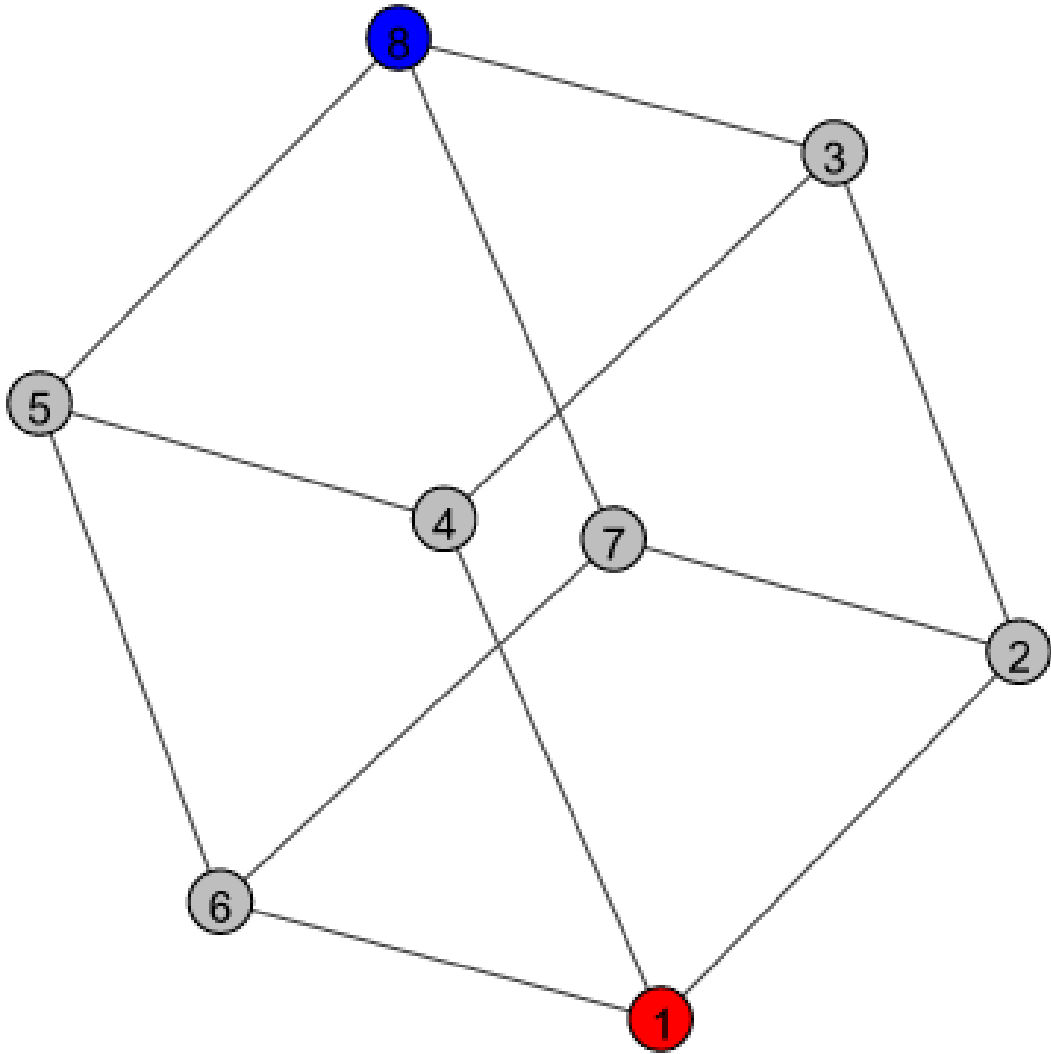# Monte-Carlo Analysis

For Recommendation Systems
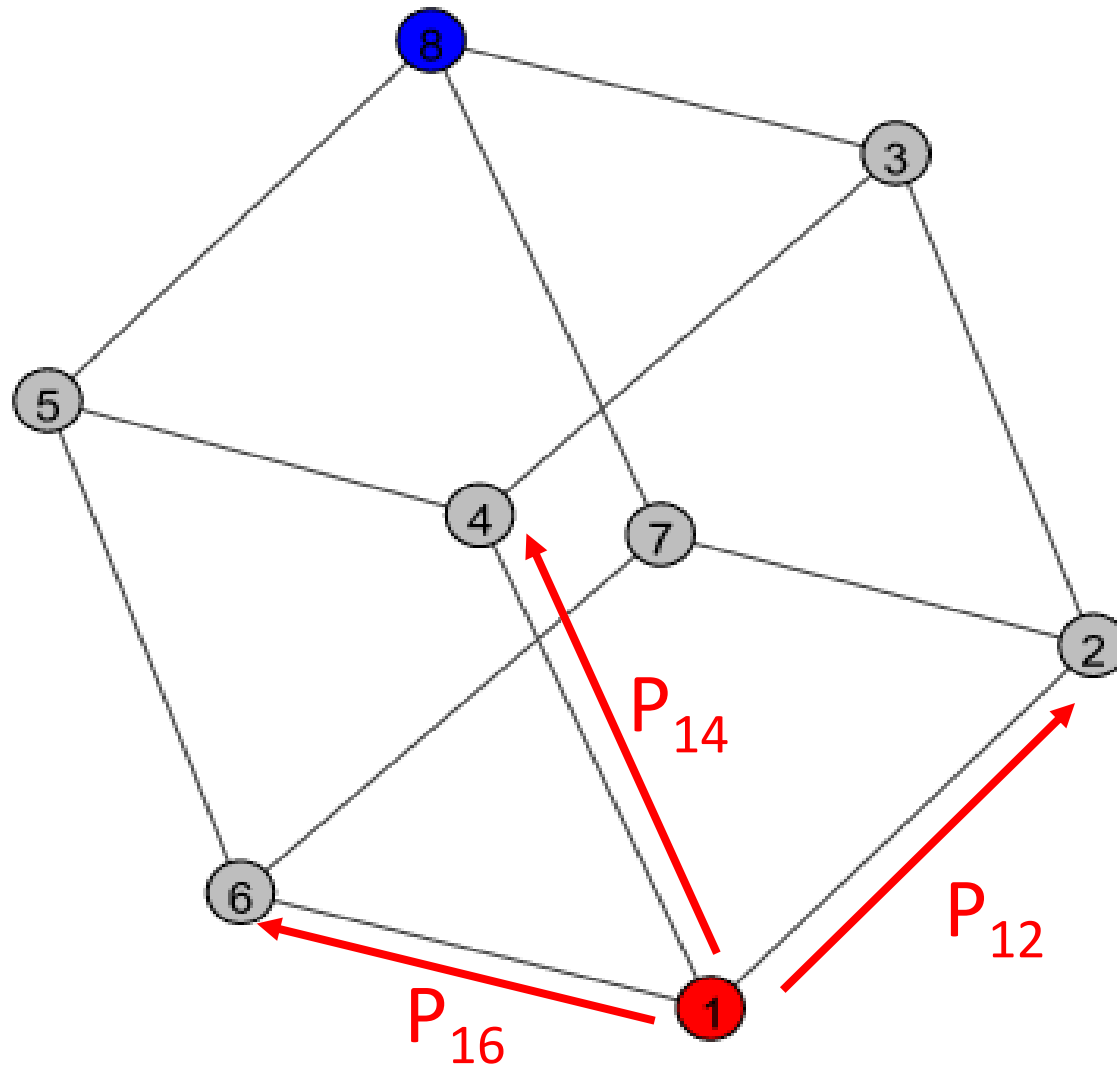
Ant on Cube
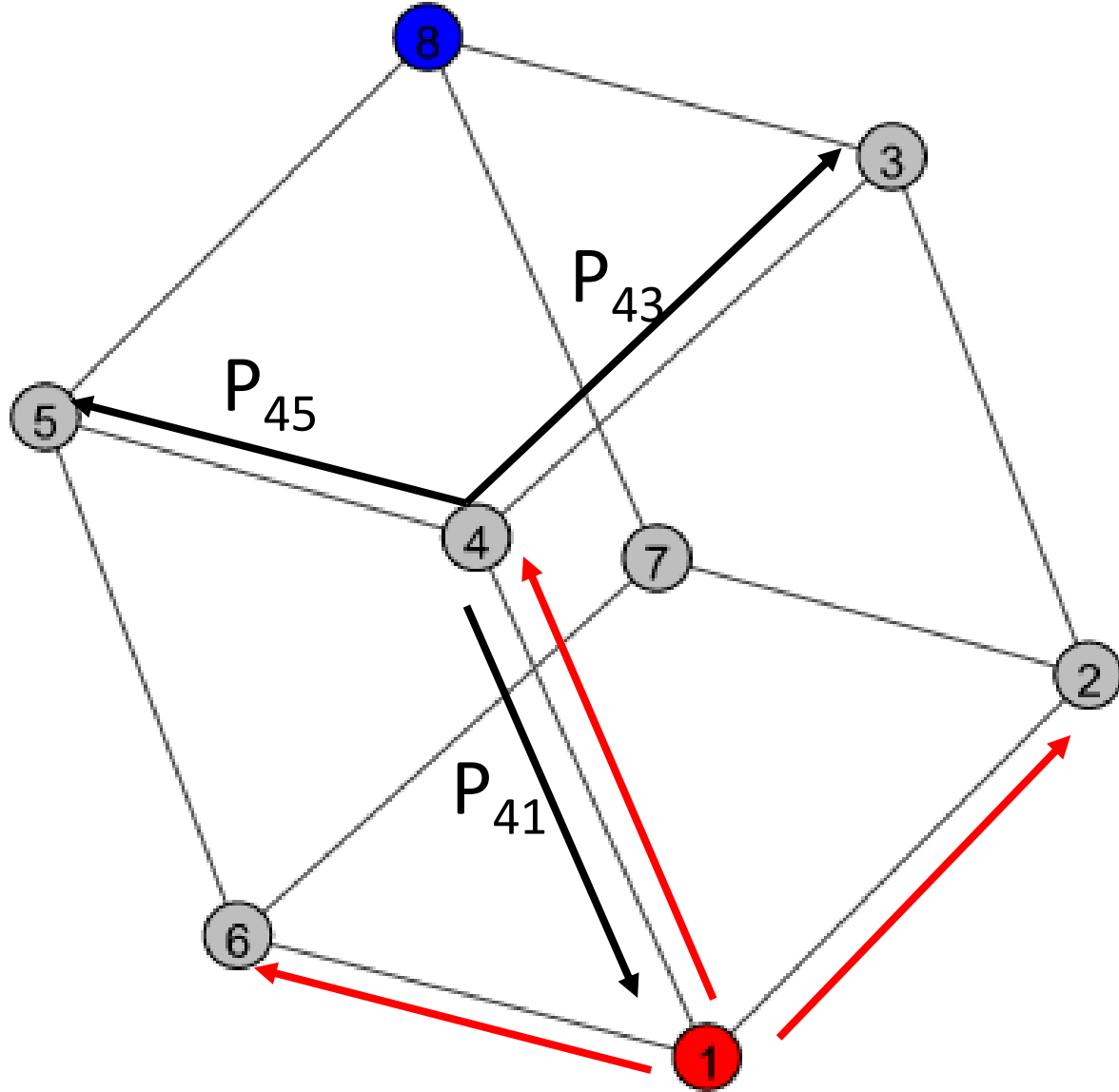

This is a simple 8-node network.

Did we really need to roll the dice to figure out how soon the ant will get from 1 to 8?

Ant on Cube

$$P_{12} = P_{14} = P_{16} = 1/3$$
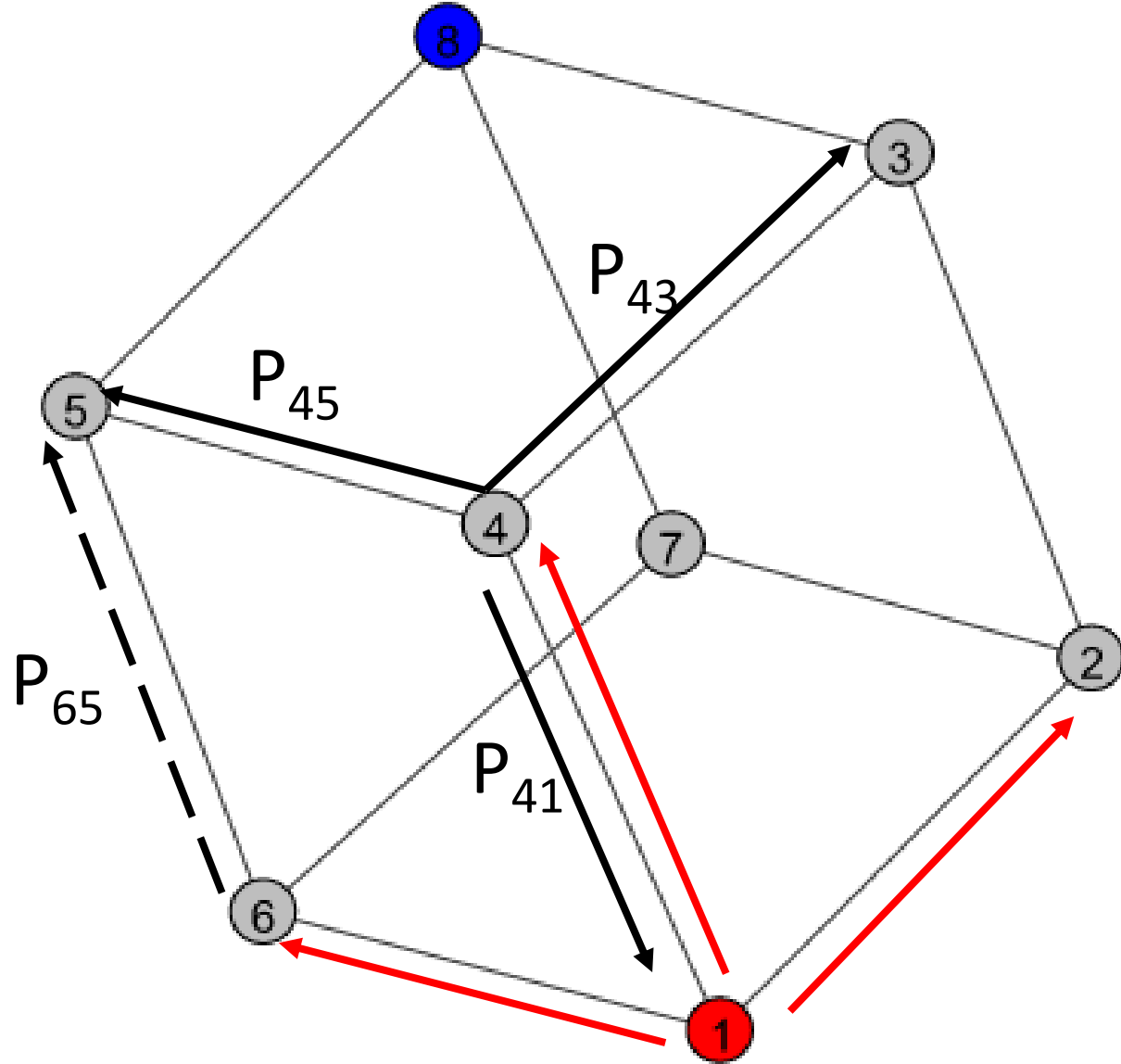
Ant on Cube

$$P_{12} = P_{14} = P_{16} = 1/3$$

$$P_{45} = P_{43} = P_{41} = 1/3$$

$$P_{15} = P_{14} * P_{45} = 1/9$$

Ant on Cube
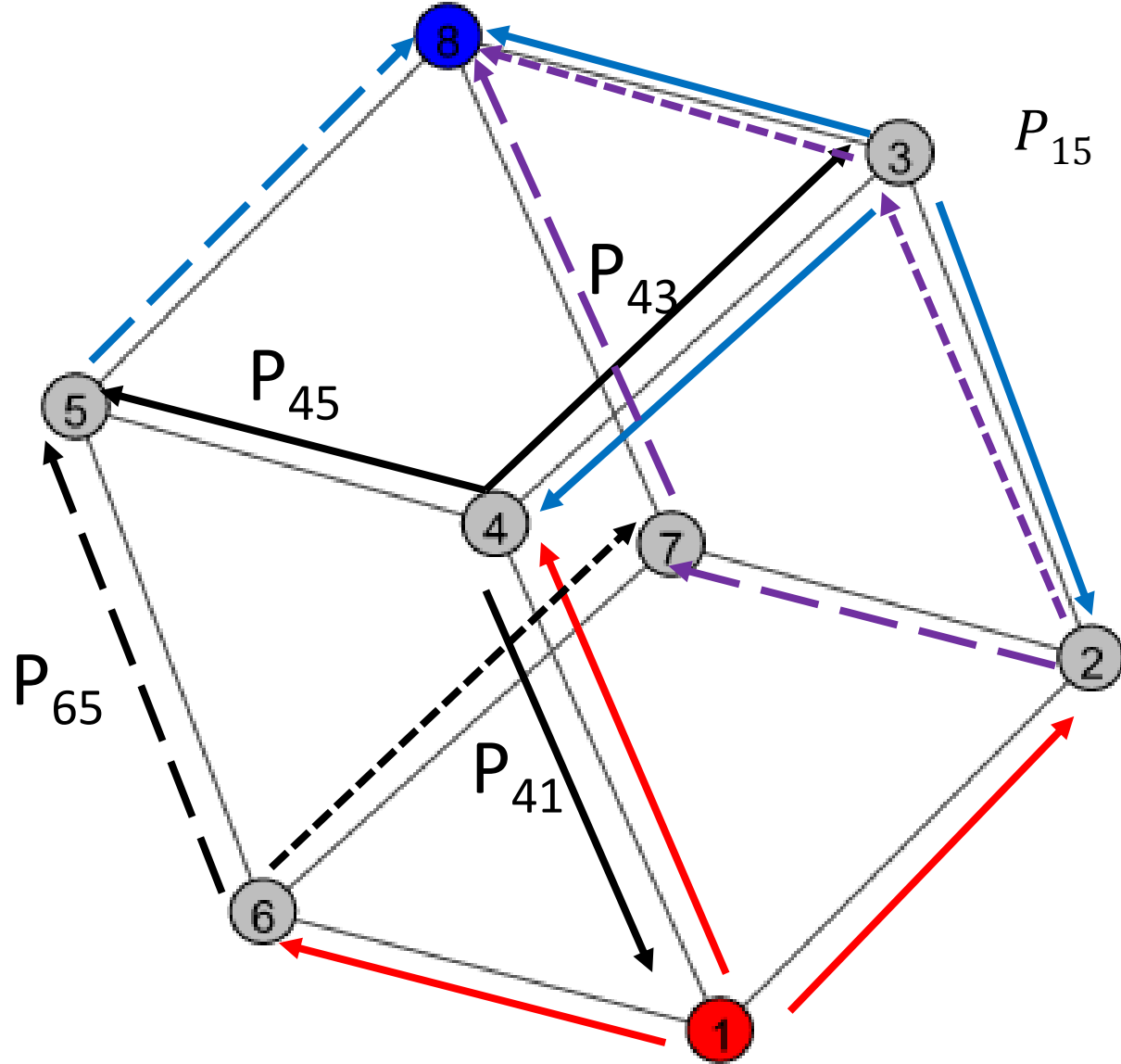

$$P_{12} = P_{14} = P_{16} = 1/3$$

$$P_{45} = P_{43} = P_{41} = 1/3$$

$$P_{15} = P_{14} * P_{45} = 1/9$$

$$P_{15} = P_{14} * P_{45} + P_{16} * P_{65} = 2/9$$

Ant is not photon
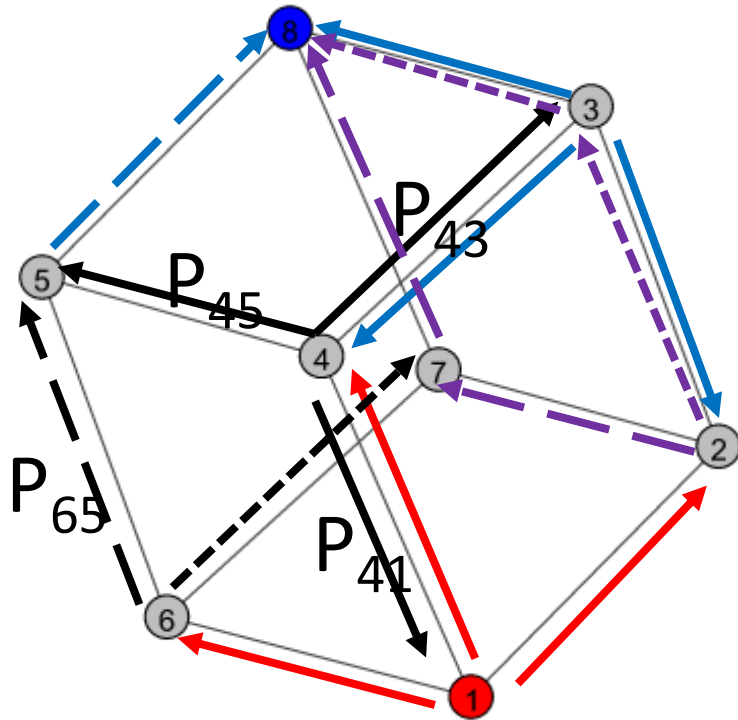
Ant on Cube

$$P_{12} = P_{14} = P_{16} = 1/3$$

$$P_{45} = P_{43} = P_{41} = 1/3$$

$$P_{15} = P_{14} * P_{45} + P_{16} * P_{65} = 2/9$$

$$P_{18} = P_{14} * P_{45} * P_{58}$$
$$+ P_{16} * P_{65} * P_{58}$$
$$+ P_{12} * P_{23} * P_{38}$$
$$+ P_{12} * P_{27} * P_{78}$$
$$+ P_{16} * P_{67} * P_{78}$$

$$P_{18} = 5/27$$

Ant on Cube

$$P_{18} = P_{14} * P_{45} * P_{58}$$
$$+ P_{16} * P_{65} * P_{58}$$
$$+ P_{12} * P_{23} * P_{38}$$
$$+ P_{12} * P_{27} * P_{78}$$
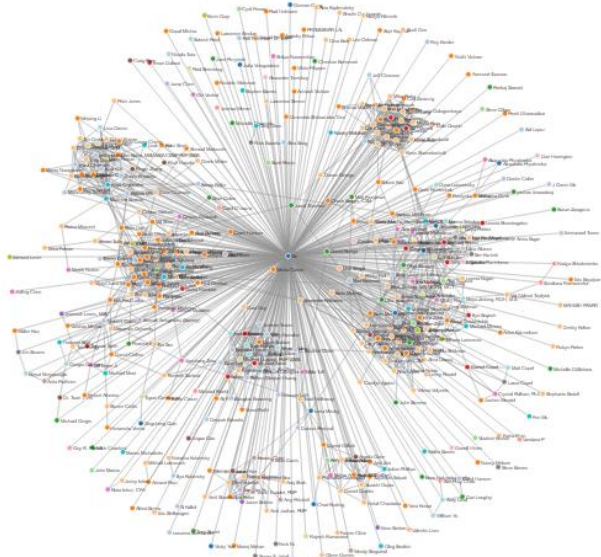$$+ P_{16} * P_{67} * P_{78}$$

$$P_{18} = 5/27$$

What if…

- Simple network:
  - small
  - highly organized
  - sparsely connected
  - unweighted
  - undirected
- No holding loops on nodes
- A single agent (Ant)
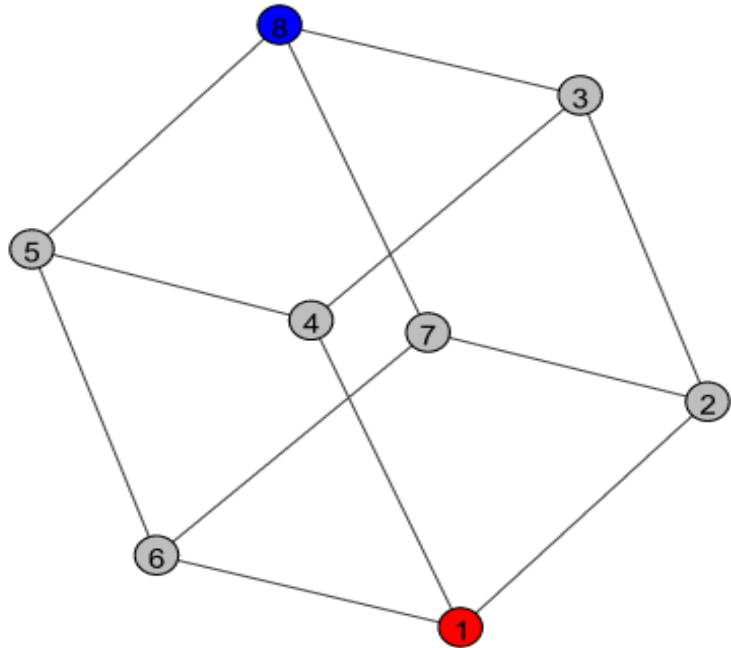
What If…
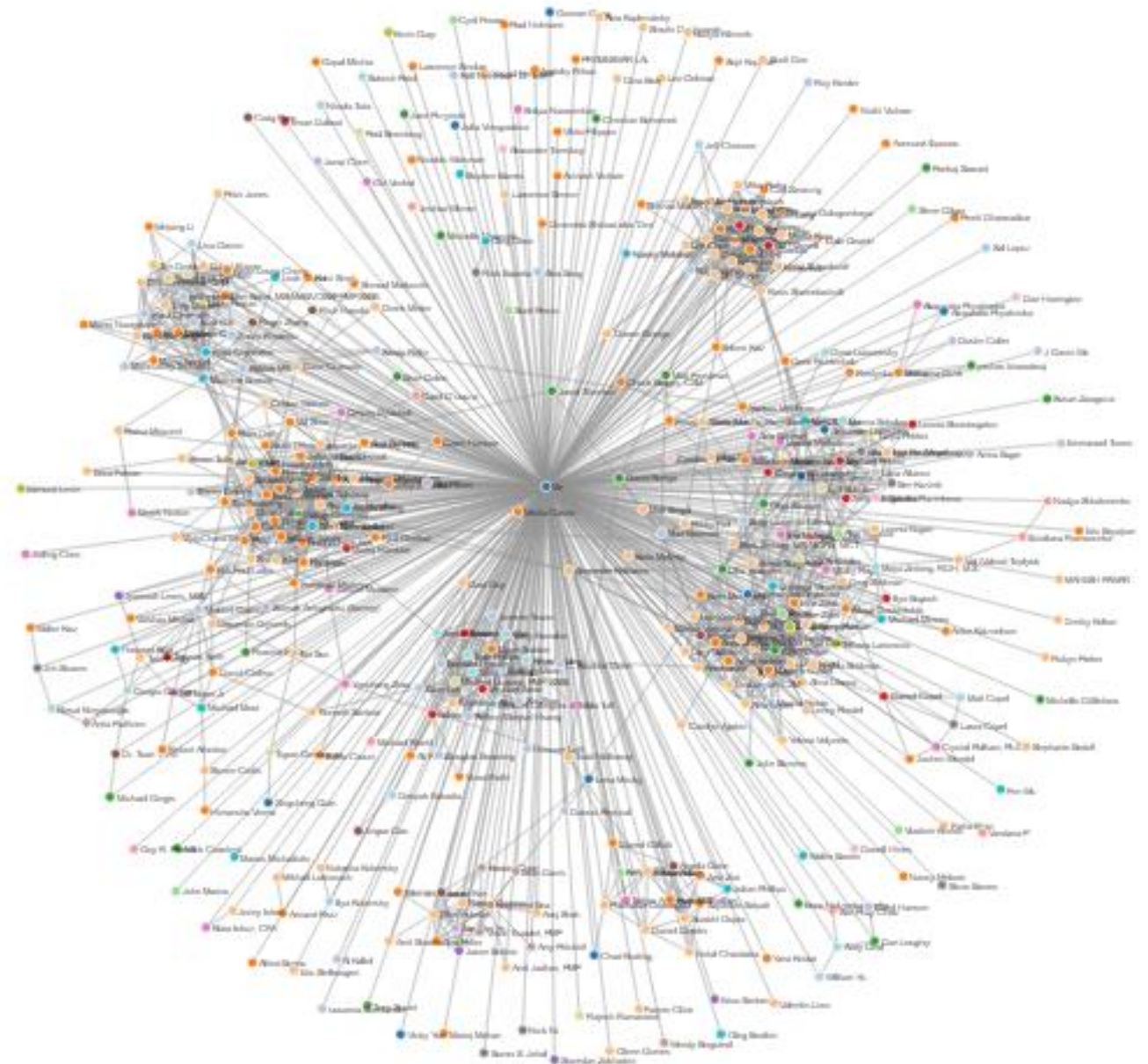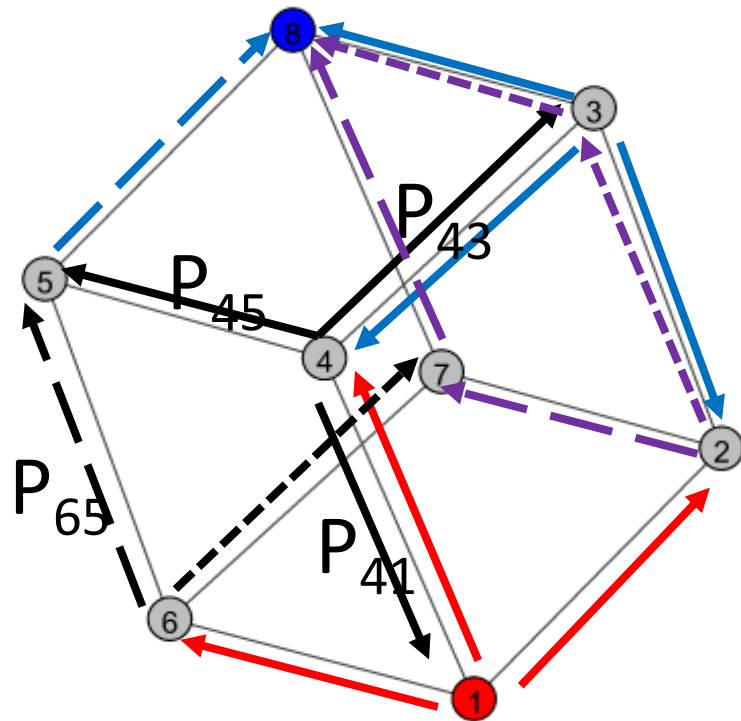
Build a Job Recommendation System:

- Job Title (s)
- Organization(s)
- Contacts

How is it related to the Ant on Cube problem?

$\sim 1e + 06$ Crawlers (Ants) = the persons looking for jobs

$\sim 1e + 03$ Environment (Cube Nodes) = (Job Descriptions, Persons Inside, Culture, Location)

$$LinkStrength[j, k]$$
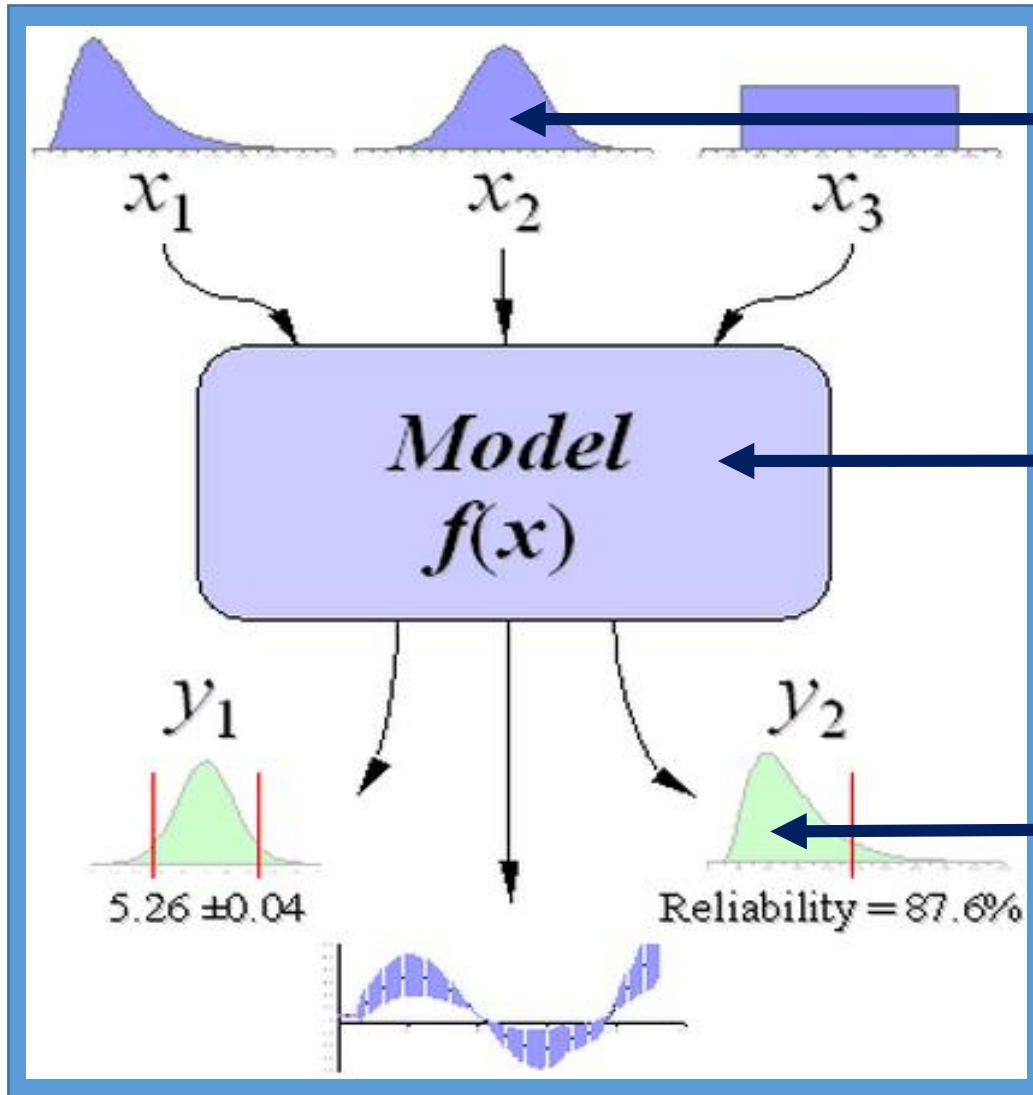$$= P\{beneficial\ [j, k]\ |\ hired\ [j, k]\}$$

$$j = 1 \dots N_{candidates}$$

$$k = 1 \dots N_{jobs}$$

$\sim 1e + 03$ Crawlers (Ants) = the job openings

$\sim 1e + 06$ Environment (Cube Nodes) = (Persons Outside, Culture, Location)

Solution – Monte-Carlo

Binary, Logistic, or Continuous

Analytical or Simulation

Resulting distributions tell the story

- http://pythonprogramming.net/monte-carlo-python/

Sampling methods for Monte-Carlo:

- Bootstrapping:
  - Resampling from the same set of samples
- Jackknifing
  - Resampling from the same distribution.

Inverse CDF Procedure:

- Get a random uniform number between 0 and 1
- Find the corresponding point on the CDF line
- Project it down to find the value of X that has been generated.

Cumulative Standard Normal [N(0, 1)] Distribution