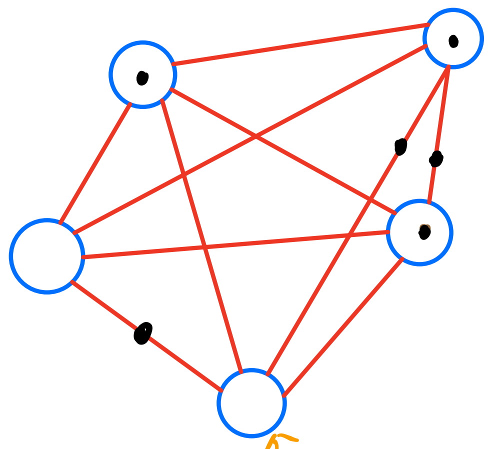
Hypercube Queuing Model Analysis

— Nino



J = 5 number of cities

N = 6 number of police car

call for service (random occurrence)

☆ Problem : where should these police vehicles be placed?

Hypercube Queuing model — Continuous time markov chain

? what is the state space ① (finite-state space)

? what is the transition matrix ②

→ Section 3-7 answered the question

why the binary arithmetic is applied?

→ N (response unit) : 1, 2, ..., N. Police car

State of a response unit: 0 (idle), 1 (busy) e.g. $B = \underbrace{[0, 1, 0, 0, \cdots, 1, 0]}_{N \text{ response unit state}} \longrightarrow$ B is state vector

State space: has 2^N binary vectors.

$\longrightarrow$ Binary arithmetic is applied in section 6.

Binary arithmetic:

a single binary $b \in \{0, 1\}$

$$B = \{b_N, b_{N-1}, \cdots, b_2, b_1\}$$

$$W(B) = \sum_{n=1}^{N} b_n \quad (\text{weight of } B)$$

count # of busy response unit.

$$V(B) = b_N 2^{N-1} + b_{N-1} 2^{N-2} + \cdots + b_2 \cdot 2 + b_1$$

$$= \sum_{n=1}^{N} b_n 2^{n-1} \quad (\text{Value of } B)$$

e.g.

$$B = [1, 0, 1]$$

$$V(B) = 2^2 + 0 + 2^0 = 5$$

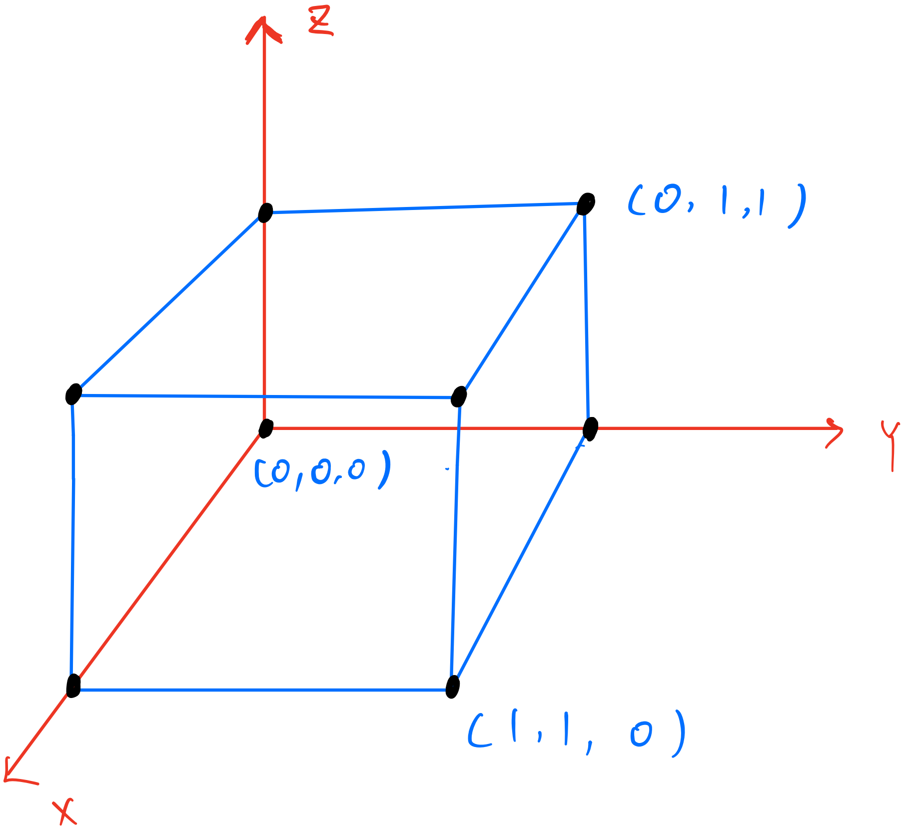
N-dimensional unit hypercube.

$$B = \{b_N, b_{N-1}, \cdots, b_1\} \in \mathbb{R}^N$$

$$b_i = 0 \text{ or } 1.$$

the set C_N of all $\mathbb{R}^N$ such vertices is the set of vertices of the N-dimensional unit hypercube in the positive orthant.

$x_i \geq 0$ for all x_i in $\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^N$



Binary set operation

$$B_1, B_2 \subset \mathbb{R}^N$$

$$(B_1 \cup B_2)_i = \{1\} \text{ iff } (B_1)_i = \{1\} \text{ or } (B_2)_i = \{1\}$$

$$(B_1 \cap B_2)_j = \{0\} \text{ iff } (B_1)_j = \{0\} \text{ or } (B_2)_j = \{0\}$$

$$(B_1)^c_i = \begin{cases} \{0\} & \text{if } (B_1)_i = 1 \\ \{0\} & \text{if } (B_i)_i = 0 \end{cases}$$

e.g. $B_1 = (1, 0, 1, 1, 0)$

$B_2 = (0, 0, 1, 0, 1)$

$B_1 \cup B_2 = (1, 0, 1, 1, 1)$

$B_1 \cap B_2 = (0, 0, 1, 0, 0)$

$(B_1)^c = (0, 1, 0, 0, 1)$

def: Hamming distance

d_{ij} : distance of B_i with B_j.

$$d_{ij} = w\left(\underbrace{[B_i \cap B_j^c]} \cup \underbrace{[B_i^c \cap B_j]} \right)$$

↳ weight.

e.g $B_i = (1, 0, 0)$

$B_j^c = (1, 0, 0)$

$B_j = (0, 1, 1)$

opposite.

$B_i \cap B_j^c = (1, 0, 0)$

It means: This calculation represents, which coordinate of B_j^c are not the same as those of B_i.

$[B_i \cap B_j^c] \cup [B_i^c \cap B_j]$ This represents a place where the coordinates of both B_i and B_j are different.

$$d_{ij} = w([B_i \cap B_j^c] \cup [B_i^c \cap B_j])$$

Here will get a number, that indicated how many coordinates of B_i are different with B_j.

$$d_{ij} = d_{ij}^{+} + d_{ij}^{-}$$

$$d_{ij}^{+} = w(B_i \cap B_j^c)$$

How many coordinate of B_i are 1 and how many cooridinate of B_j are 0.

$$d_{ij}^{-} = w(B_i^c \cap B_j)$$

How many coordinate of B_j are 1. and how many coordinate of B_i are 0.

↑ Binary Part done.

Notation

Hyperacube Queuing model:

Section 3: Model Description.

J atom: $j = \{1, 2, \dots, J\}$ Place

atom j: f_j workload at j.

$\sum_{j=1}^{J} f_j = 1$ and $0 < f_j < 1$

mean travel time:

$J \times J$ matrix τ

$\tau = (\tau_{ij})$ τ_{ij}: the average travel time from atom i to atom j.

$\tau_{ij} \neq \tau_{ji}$

i to j $\quad$ j to i

N response unit $n = \{1, 2, \dots, N\}$

location matrix : $N \times J$ matrix. $\mathcal{L}$

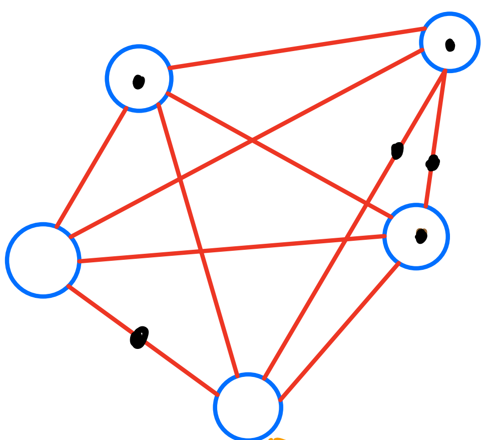
$\mathcal{L} = (\ell_{nj})$ — n : response unit n; j : atom j

ℓ_{nj} : Probability / fraction of time that response unit n is located in atom j while idle.

$\sum_{j=1}^{J} \ell_{nj} = 1$. (row sum.)

$\mathcal{L}$ is a stochastic matrix.

Queuing model. ↑ Notation



call for service

(call for service) (1) In atom j, the call for serice is generate as Poisson Process. $\sim Pois(\lambda f_j)$

(Service) (2) Exactly one response unit to a service call

(3) Service time of a call $\sim Exp(\mu)$,
(mean $= 1/\mu$)

(4) Service time independent of the response unit, the location, the customer, the history.

It means, It is a markov chain.

$M/M/N$ queuing model.

queuing { 1° zero capacity ✓
2° infinite capacity.

Section 3° ↑

Section 4

State space ?

transition matrix ?

N response unit

State space : 2^N N-dimensional binary vector.

e.g $B = \{1, 0, 1, \dots, 0, 1, 1\}$

State k if $V(B) = k$

def : state transition matrix

$\Lambda = (\lambda_{ij})$, the average rate of transitions from state i (B_i) to state j.

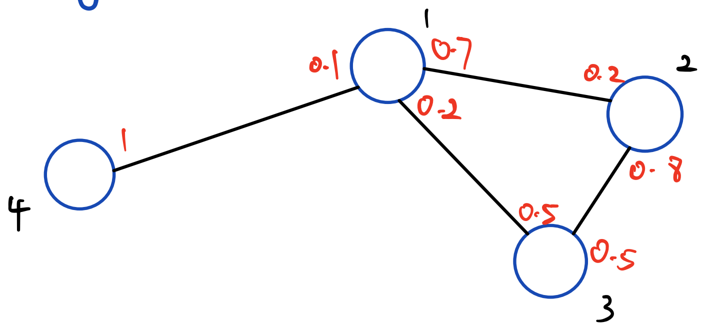
$\Lambda \in 2^N \times 2^N$

$0 \le i \le 2^N - 1$

$0 \le j \le 2^N - 1$

$$\lambda_{ij} = -\sum_{j \ne i} \lambda_{ij}$$

e.g



$$S = \begin{bmatrix} 0 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0 & 0.8 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

S_{ij} = prob from state i to state j.

def: state $i \rightleftarrows$ state j iff $d_{ij} = 1$

$\downarrow$ B_i $\downarrow$ B_j $\downarrow$ hamming distance

state i, state j, adjacent

$d_{ij} = d_{ij}^{+} + d_{ij}^{-}$

$d_{ij}^{+} = 1 \Rightarrow$ upword transition

↳ state i : has one police car idle $\Rightarrow$ busy in state j.

↳ It's hard to calculate.

$d_{ij}^{-} = 1 \Rightarrow$ downward transition

↳ state i : has one police car busy $\xrightarrow{\exp(\mu)}$ idle in state j.

$$\lambda_{ij} = \begin{cases} 0 & \text{if state } i \text{ and state } j \text{ not adjacent } (d_{ij} \ne 1) \\ > 0 & \text{otherwise } (d_{ij} = 1) \end{cases}$$

$(d_{ij} = 1)$

$$\lambda_{ij} = \begin{cases} \mu & \text{if } d_{ij}^{-} = 1 \quad (\text{assume } \mu = 1) \\ ? & \text{if } d_{ij}^{+} = 1 \end{cases}$$

$$\underset{\sim}{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{2N} \end{bmatrix}$$

state 1

state 2N

stationary distribution

(steady-state-probability)

$$\underset{\sim}{\pi} = \Lambda \underset{\sim}{\pi}$$

equation of detailed balance

transition to state j = transition out of state j

$$\sum_{i \neq j} \pi_i \lambda_{ij} = \pi_j \left(\sum_{i \neq j} \lambda_{ji} \right)$$

downward transition

$$\sum_{i \neq j,\, d_{ij}^{+}=1} \pi_i \lambda_{ij} + \sum_{i \neq j,\, d_{ij}^{-}=1} \pi_i = \pi_j \left(\lambda_j + w(B_j) \right)$$

$(\lambda_{ij} = 1)$

$$\lambda_j = \begin{cases} 0 & \text{if } j = 2^N - 1 \\ \lambda & \text{otherwise} \end{cases}$$

Section 4, 5, 6 Tour Algorithm

section 3 model Description

— continuous markov chain

State space : 2^N Binary vectors

$$B = \{0, 1, 0, \ldots, 1, 1\}$$

$\underbrace{\qquad\qquad}_{\in \{0,1\}^{2^N}}$

transition matrix ?

Section 4. Transition matrix

$$\Lambda = (\lambda_{ij}) \in \mathbb{R}^{2N \times 2N}$$

the average transition rate from State i to state j.

$$\lambda_{ij} = \begin{cases} > 0 & \text{if } d_{ij} = 1 \\ 0 & \text{if } d_{ij} \neq 1 \end{cases}$$

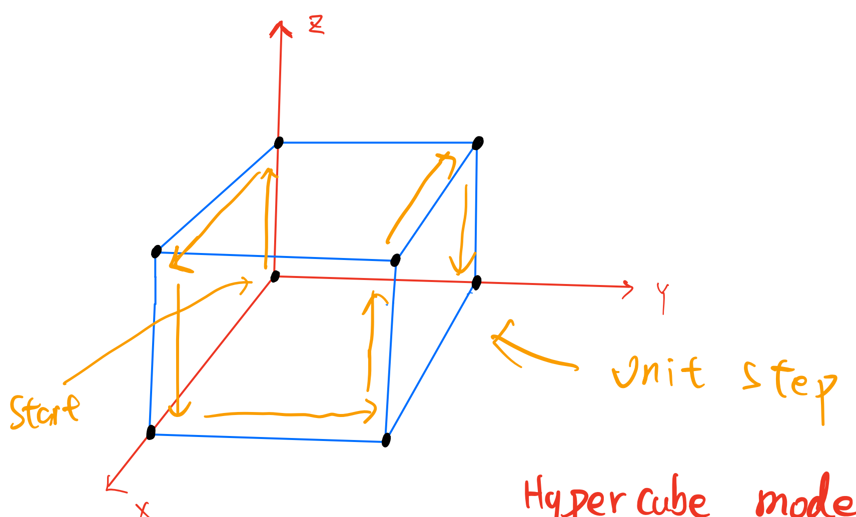
$$d_{ij} = d_{ij}^{+} + d_{ij}^{-}$$

$$\lambda_{ij} = \begin{cases} ? & \text{if } d_{ij}^{+} = 1 \text{ (upward transition)} \\ 1 & \text{if } d_{ij}^{-} = 1 \text{ (downward transition)} \end{cases}$$

detailed balance

→ Computational infeasible

Section 5° Tour algorithm



Hypercube model

Note: change State $B_i \longrightarrow B_j$, only one status changed

A Unit Step: $B_i \to B_j$, $d_{ij} = 1$

assuming that:

State B_i, location: atom j

If atom j has a call for service,

we know how to dispatch the optional unit to atom j.

State i, state k, $d_{ik}^{+} = 1$

update λ_{ik}

algorithm arrived at state B_i

assumption

$$\lambda_{ik}^{(new)} = \lambda_{ik}^{(old)} + \sum_{j=1}^{J} \lambda \frac{f_j}{\eta_{ij}}$$

we know η_{ij}

η_{ij} = State B_i, atom j, # of option units to dispatch.

Section 6:

Path on the hypercube vertices

unit - step

algorithm give state into $\underbrace{\text{ternary expansion}}$

convert

binary expansion

Section 7:

Policy: "fixed preference"

preference score: "expected modified center-of-mass" (MCM)

State B, call for service: atom j

unit i is idle/available

$$t_{ij} = \sum_k l_{ik} \cdot \tau_{kj}$$

unit i

atom j

location matrix

travel matrix

t_{ij} the expected time for unit i to travel to atom j, unconditioned on its current location.