Tipsy Cop and Tipsy Robber on Vertex and Edge Symmetric Graphs

Joint work with Pamela Harris,
Alicia Pietro-Langarica, Charles Smith
and Grabiel Sosa Castillo

Presented by Giancarlo Arcese

Research Description, Objective

- This research studies a variation of the well-known "cops and robbers" game in graph theory.
- The main difference lies in that the cop and robber occasionally move at random (hence the "tipsy"). This random movement is governed by the parameters p and q.
- The goal is to calculate the **expected number of movements** until the game ends as a function of p and q.
- Today I will present our solutions for three graphs/families of graphs: the Petersen graph, cycle graphs, and hypercube graphs.

Game Rules

The rules of the game are as follows:

Game Rules

The rules of the game are as follows:

1. A graph is chosen to play on and the agents (survivor and zombie) start on separate vertices.

Game Rules

The rules of the game are as follows:

1. A graph is chosen to play on and the agents (survivor and zombie) start on separate vertices.
2. The agents alternate in moving. Either agent may move first.

Game Rules

The rules of the game are as follows:

1. A graph is chosen to play on and the agents (survivor and zombie) start on separate vertices.
2. The agents alternate in moving. Either agent may move first.
3. **Survivor Move Rule:** With probability p, the survivor moves to an adjacent vertex that maximizes its shortest-path distance from the zombie. If multiple vertices satisfy this condition, one is chosen uniformly at random. Otherwise, (with probability $1 - p$), the survivor moves to an adjacent vertex uniformly at random.

Game Rules

The rules of the game are as follows:

1. A graph is chosen to play on and the agents (survivor and zombie) start on separate vertices.
2. The agents alternate in moving. Either agent may move first.
3. **Survivor Move Rule:** With probability p, the survivor moves to an adjacent vertex that maximizes its shortest-path distance from the zombie. If multiple vertices maximize the distance, one is chosen uniformly at random. Otherwise, (with probability $1 - p$), the survivor moves to an adjacent vertex uniformly at random.
4. **Zombie Move Rule:** With probability q, the zombie moves to an adjacent vertex that minimizes its shortest-path distance from the survivor. If multiple vertices minimize the distance, one is chosen uniformly at random. Otherwise, (with probability $1 - q$), the zombie moves to an adjacent vertex uniformly at random.

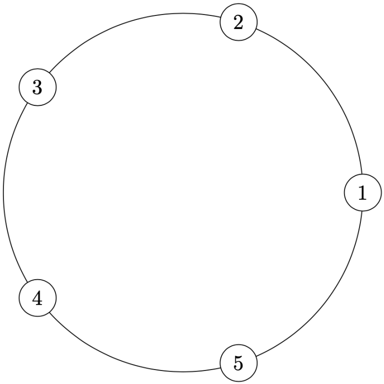
Game Rules

The rules of the game are as follows:

1. A graph is chosen to play on and the agents (survivor and zombie) start on separate vertices.
2. The agents alternate in moving. Either agent may move first.
3. **Survivor Move Rule:** With probability p, the survivor moves to an adjacent vertex that maximizes its shortest-path distance from the zombie. If multiple vertices maximize the distance, one is chosen uniformly at random. Otherwise, (with probability $1 - p$), the survivor moves to an adjacent vertex uniformly at random.
4. **Zombie Move Rule:** With probability q, the zombie moves to an adjacent vertex that minimizes its shortest-path distance from the survivor. If multiple vertices minimize the distance, one is chosen uniformly at random. Otherwise, (with probability $1 - q$), the zombie moves to an adjacent vertex uniformly at random.
5. The game ends when one agent moves onto the same vertex as the other agent. If that occurs on the n^{th} move, the game is said to have lasted n moves.

Game Example

Let's see an example of the game being played on the 5-cycle graph:

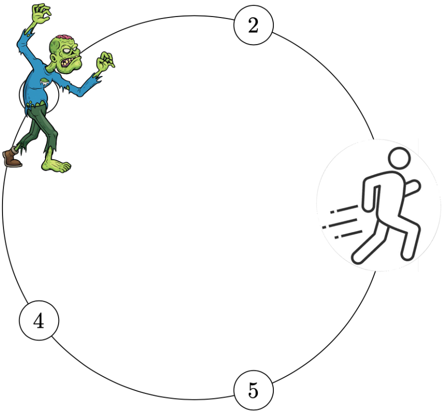


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3

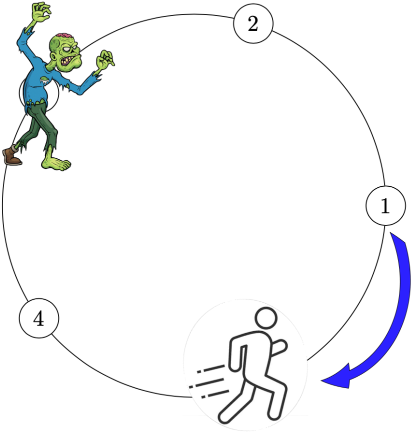


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3

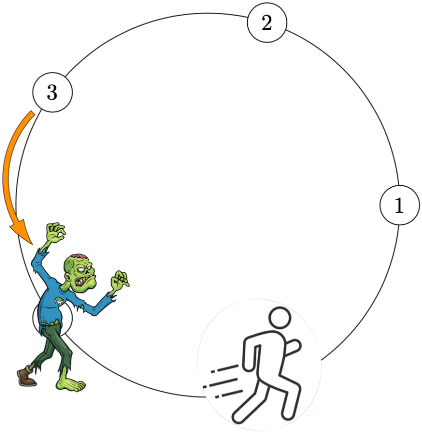


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3
2	-	0.32	5	4

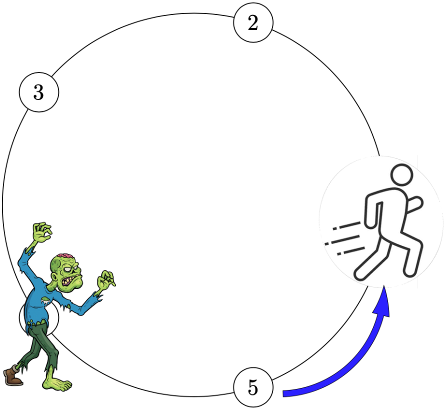


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3
2	-	0.32	5	4
3	0.57	-	1	4

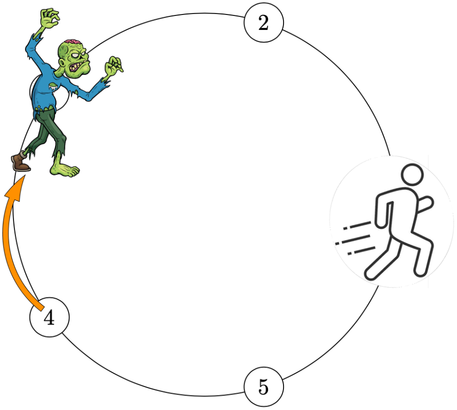


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3
2	-	0.32	5	4
3	0.57	-	1	4
4	-	0.37	1	3

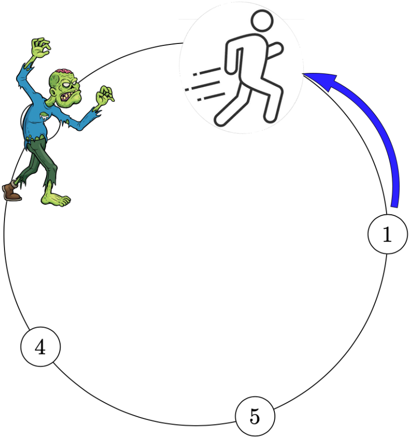


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3
2	-	0.32	5	4
3	0.57	-	1	4
4	-	0.37	1	3
5	0.11	-	2	3

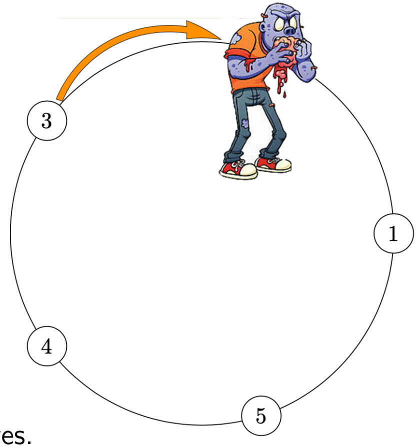


Game Example

Let's see an example of the game being played on the 5-cycle graph:

Let $p = q = 0.5$

Move	Sroll	Zroll	Spos	Zpos
0	-	-	1	3
1	0.92	-	5	3
2	-	0.32	5	4
3	0.57	-	1	4
4	-	0.37	1	3
5	0.11	-	2	3
6	-	0.79	2	2



In this example, the game lasted 6 moves.

Solving Strategy

- The general strategy used to solve for expected length involves encoding the game as a **Markov chain** where each state of the transition matrix is a possible distance between the agents.
- The state denoting a distance of 0 corresponds to the game ending. This is the chain's sole **absorbing state**, and is denoted "End."
- Expected length is calculated by way of the row sums of the associated **fundamental matrix**.
- We therefore have the relatively straightforward plan of:

Construct transition matrix $\rightarrow$ **Calculate fundamental matrix**

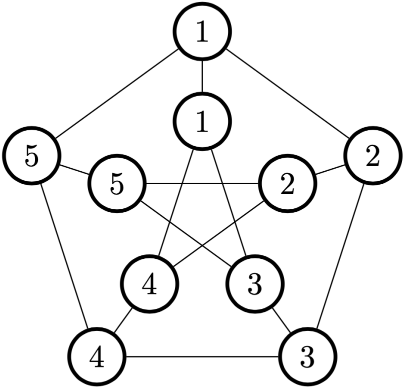
for the three graphs/families of graphs of interest.

Graph Choice Rationale

- Not all graphs can be modeled with a distance-stated Markov chain; only those that are **vertex and edge symmetric** are able to.
- Other graphs can still be modeled as a Markov chain by using the **positioning of the agents**, but this creates large, irregularly-patterned transition matrices.
- This makes solving for the fundamental matrix difficult, since that requires inverting a large, irregularly-patterned matrix.
- For that reason, this research chooses to focus on graphs like the Petersen graph, cycle-graphs, and the hypercube graphs, since they are all symmetric.

Section Preface

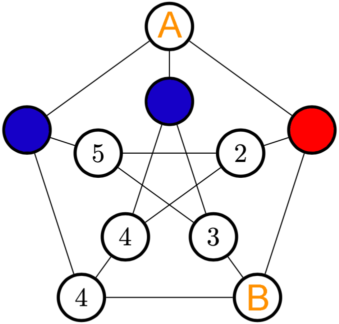
The Petersen Graph



Deriving Transition Matrix

Two useful observations about the Petersen:

1. The max distance between vertices is 2.
2. Relative to a vertex B, a vertex A borders **2** vertices that are the max distance from B, and **1** that is closer to B.



From this we formulate $P_s(C)$, $P_z(C)$, the probability the agents have moved closer after the survivor or zombie's move, respectively:

$$P_s(C) = \frac{1-p}{3} \qquad P_z(C) = \frac{2q+1}{3}$$

Their converses will be denoted as $P_s(F)$ and $P_z(F)$.

Transition Matrix

Using the last slide's results, we obtain the following transition matrix T:

$$T = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccc} \text{End} & S_1 & Z_1 & S_2 & Z_2 & \\ \end{array}$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ P_s(C) & 0 & 0 & 0 & P_s(F) \\ P_z(C) & 0 & 0 & P_z(F) & 0 \\ 0 & 0 & P_s(C) & 0 & P_s(F) \\ 0 & P_z(C) & 0 & P_z(F) & 0 \end{pmatrix} \begin{array}{l} \text{End} \\ S_1 \\ Z_1 \\ S_2 \\ Z_2 \end{array}$$

Where the entry (i, j) entry denotes the probability of transitioning to state j having started in state i.

From here, the fundamental matrix N is easily calculated as $(I - Q)^{-1}$, where Q is the 4×4 submatrix formed by the transient states of T.

Inverting (I-Q)

Let $a = P_z(C)$, $b = P_s(C)$, $c = P_z(F)$, $d = P_s(F)$. Then $(I - Q)$ and its determinant D are the following:

$$(I-Q) = \begin{array}{c c} \begin{array}{cccc} S_1 & Z_1 & S_2 & Z_2 \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 & -d \\ 0 & 1 & -c & 0 \\ 0 & -b & 1 & -d \\ -a & 0 & -c & 1 \end{pmatrix} & \begin{array}{c} S_1 \\ Z_1 \\ S_2 \\ Z_2 \end{array} \end{array} \qquad D = (1-ad)(1-bc) - ab$$

Multiplying $\frac{1}{D}$ by the adjugate matrix of $(I - Q)$ yields the fundamental matrix N:

$$N = \frac{1}{D} \times \begin{array}{c c} \begin{array}{cccc} S_1 & Z_1 & S_2 & Z_2 \end{array} & \\ \begin{pmatrix} 1-ab-cb & abd & ad & d(1-cb) \\ abc & 1-ab-ad & c(1-ad) & bc \\ ab & b(1-ad) & 1-ad & b \\ a(1-bc) & ab & a & 1-bc \end{pmatrix} & \begin{array}{c} S_1 \\ Z_1 \\ S_2 \\ Z_2 \end{array} \end{array}$$

Solution

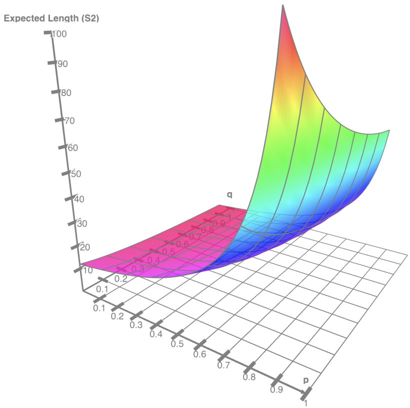
The row sums of N give the expected length of the game for the different possible starting conditions. For example, the row sum of the S_2 row yields the expected length when the agents start at a distance of two and the survivor moves first. In terms of p and q, we get:

$$\mathbf{E}(S_2) = \frac{3(2p^2q + p^2 + 2pq + p - 4q + 52)}{(p-1)(2q+1)(2pq - 2p + 4q - 13)}$$

Evaluated for $p = q = 0.5$ yields:

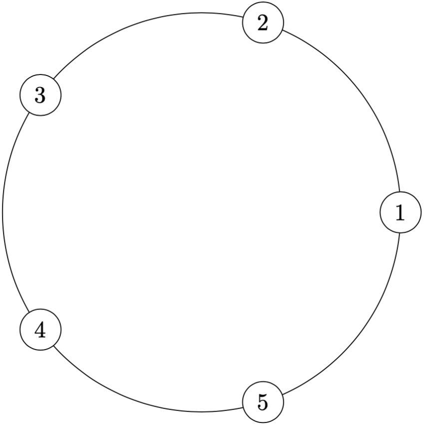
$$\mathbf{E}(S_2) = 13.43478$$

Graph


Expected Length (S2)
100
90
80
70
60
50
40
30
20
10
q
p
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

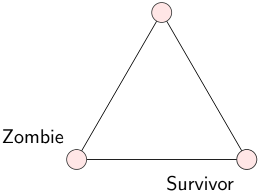
Section Preface

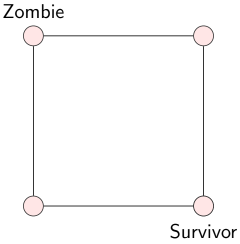
Cycle Graphs



Deriving Transition Matrix

- The max distance m is $\frac{n}{2}$ for n even and $\frac{n-1}{2}$ for n odd.
- Given a pair of vertices A and B at a distance d:
    - **If $\mathbf{d \neq m}$:** A borders **1** vertex that is closer to B and **1** that is further
    - **If $\mathbf{d = m}$ and n is even:** A borders **2** vertices that are closer to B.
    - **If $\mathbf{d = m}$ and n is odd:** A borders **1** vertex closer to B and one at the same distance d.





- This means our transition probabilities $(P_s(C), P_z(C), P_s(F), P_z(F))$ are equivalent across all distances except the max distance m.

Transition Matrix Precursor

- Unlike the Petersen graph, the transition matrices for n-cycles will be on a two-move scale instead of a one-move scale. This helps reduce the number of states and simplify the matrix structure.
- Furthermore, the structure of the transition matrices depends on n. There are three different cases:
  1. $n = 4k,\ k \in \mathbb{N}_1$
  2. $n = 4k + 2,\ k \in \mathbb{N}_1$
  3. n odd
- Thus, we will need to investigate all three cases in order to solve for all n-cycles.
- Fortunately, all cases can be solved in an analogous fashion. This presentation will only cover case 1.

Case 1: $n = 4k$

Transition matrix and $(I - Q)$ for the 12-cycle:

$$T = \begin{array}{cc} \begin{array}{cccc} \text{End} & S_2 & S_4 & S_6 \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ -B & 1+A+B & -A & 0 \\ 0 & -B & 1+A+B & -A \\ 0 & 0 & -C & 1+C \end{pmatrix} & \begin{array}{c} \text{End} \\ S_2 \\ S_4 \\ S_6 \end{array} \end{array}$$

$$(I - Q) = \begin{array}{cc} \begin{array}{ccc} S_2 & S_4 & S_6 \end{array} & \\ \begin{pmatrix} -(A+B) & A & 0 \\ B & -(A+B) & A \\ 0 & C & -C \end{pmatrix} & \begin{array}{c} S_2 \\ S_4 \\ S_6 \end{array} \end{array}$$

Where:

$$A = -P_s(F)P_z(F) \qquad B = -P_s(C)P_z(C) \qquad C = -P_z(C)$$

Computing the Inverse - Determinant Recursion (1)

Define $\mathcal{M}_k$ the family of $(I - Q)$ matrices when $n = 4k$, $k \in \mathbb{N}_1$.

$$\mathcal{M}_k = \begin{pmatrix} -(A+B) & A & 0 & \dots & 0 & 0 & 0 \\ B & -(A+B) & A & \dots & 0 & 0 & 0 \\ 0 & B & -(A+B) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(A+B) & A & 0 \\ 0 & 0 & 0 & \dots & B & -(A+B) & A \\ 0 & 0 & 0 & \dots & 0 & C & -C \end{pmatrix}$$

The determinant of $\mathcal{M}_k$ is given recursively as:

$$\det(\mathcal{M}_k) = -(A+B)\det(\mathcal{M}_{k-1}) - AB\det(\mathcal{M}_{k-2}) \quad k \geq 2$$

We define $\det(\mathcal{M}_0) = \dfrac{C}{B}$ so that $\det(\mathcal{M}_2) = BC$ agrees with the formula.

Computing the Inverse - Determinant Recursion (1)

Define $\mathcal{M}_k$ the family of $(I - Q)$ matrices when $n = 4k$, $k \in \mathbb{N}_1$.

$$\mathcal{M}_k = \begin{pmatrix} -(A+B) & A & 0 & \dots & 0 & 0 & 0 \\ B & -(A+B) & A & \dots & 0 & 0 & 0 \\ 0 & B & -(A+B) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(A+B) & A & 0 \\ 0 & 0 & 0 & \dots & B & -(A+B) & A \\ 0 & 0 & 0 & \dots & 0 & C & -C \end{pmatrix}$$

The determinant of $\mathcal{M}_k$ is given recursively as:

$$\det(\mathcal{M}_k) = \underbrace{-(A+B)\det(\mathcal{M}_{k-1})}_{\text{Pivotting on (1,1)}} - (AB)\det(\mathcal{M}_{k-2}) \quad k \geq 2$$

We define $\det(\mathcal{M}_0) = \dfrac{C}{B}$ so that $\det(\mathcal{M}_2) = BC$ agrees with the formula.

Computing the Inverse - Determinant Recursion (1)

Define $\mathcal{M}_k$ the family of $(I - Q)$ matrices when $n = 4k$, $k \in \mathbb{N}_1$.

$$\mathcal{M}_k = \begin{pmatrix} -(A+B) & A & 0 & \dots & 0 & 0 & 0 \\ B & -(A+B) & A & \dots & 0 & 0 & 0 \\ 0 & B & -(A+B) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -(A+B) & A & 0 \\ 0 & 0 & 0 & \dots & B & -(A+B) & A \\ 0 & 0 & 0 & \dots & 0 & C & -C \end{pmatrix}$$

The determinant of $\mathcal{M}_k$ is given recursively as:

$$\det(\mathcal{M}_k) = \underbrace{-(A+B)\det(\mathcal{M}_{k-1})}_{\text{Pivotting on (1,1)}} \underbrace{-(AB)\det(\mathcal{M}_{k-2})}_{\substack{\text{Pivotting on (1,2),} \\ \text{then the new (1,1)}}} \quad k \geq 2$$

We define $\det(\mathcal{M}_0) = \dfrac{C}{B}$ so that $\det(\mathcal{M}_2) = BC$ agrees with the formula.

Computing the Inverse - Determinant Recursion (2)

We obtain the following characteristic equation from this recursion:

$$X^2 + (A+B)X + AB = 0$$
$$\rightarrow (X+A)(X+B) = 0$$

Therefore, using Binet's formula, we have that:

$$\det(\mathcal{M}_k) = \alpha(-A)^k + \beta(-B)^k$$

Plugging in $k = 0{,}1$ to solve for α, β:

$$k=0: \ \frac{C}{B} = \alpha + \beta \qquad\qquad k=1: \ -C = -\alpha A - \beta B$$

$$\rightarrow \beta = \frac{C}{B}, \ \alpha = 0$$

Plugging in α, β gives a closed form expression for $\det(\mathcal{M}_k)$

$$\det(\mathcal{M}_k) = \frac{C}{B}(-B)^k$$

Computing the Inverse - Matrix Minors

Next we need to compute the minors of $\mathcal{M}_k$. We'll investigate the minors of $\mathcal{M}_7$ and use its example to help generalize to $\mathcal{M}_k$.

$$\mathcal{M}_7 = \begin{pmatrix} -(A+B) & A & 0 & 0 & 0 & 0 & 0 \\ B & -(A+B) & A & 0 & 0 & 0 & 0 \\ 0 & B & -(A+B) & A & 0 & 0 & 0 \\ 0 & 0 & B & -(A+B) & A & 0 & 0 \\ 0 & 0 & 0 & B & -(A+B) & A & 0 \\ 0 & 0 & 0 & 0 & B & -(A+B) & A \\ 0 & 0 & 0 & 0 & 0 & C & -C \end{pmatrix}$$

Computing the Inverse - Matrix Minors

Next we need to compute the minors of $\mathcal{M}_k$. We'll investigate the minors of $\mathcal{M}_7$ and use its example to help generalize to $\mathcal{M}_k$.

$$\mathcal{M}_7 = \begin{pmatrix} -(A+B) & A & 0 & 0 & 0 & 0 & 0 \\ B & -(A+B) & A & 0 & 0 & 0 & 0 \\ 0 & B & -(A+B) & A & 0 & 0 & 0 \\ 0 & 0 & B & (A+B) & A & 0 & 0 \\ 0 & 0 & 0 & B & -(A+B) & A & 0 \\ 0 & 0 & 0 & 0 & B & -(A+B) & A \\ 0 & 0 & 0 & 0 & 0 & C & -C \end{pmatrix}$$

The minor of a diagonal entry yields a block diagonal matrix. The bottom right block is simply a matrix of $\mathcal{M}$. The top left block is a unaberrated version of $\mathcal{M}$. We will define this unaberrated family of matrices as $\mathcal{U}$.

Like $\mathcal{M}_k$, $\mathcal{U}_k$, the determinant of $\mathcal{U}_k$ is easily derived by way of recursion.

Computing the Inverse - Matrix Minors

Next we need to compute the minors of $\mathcal{M}_k$. We'll investigate the minors of $\mathcal{M}_7$ and use its example to help generalize to $\mathcal{M}_k$.

$$\mathcal{M}_7 = \begin{pmatrix} -(A+B) & A & 0 & 0 & 0 & 0 & 0 \\ B & -(A+B) & A & 0 & 0 & 0 & 0 \\ 0 & B & -(A+B) & A & 0 & 0 & 0 \\ 0 & 0 & B & -(A+B) & A & 0 & 0 \\ 0 & 0 & 0 & B & -(A+B) & A & 0 \\ 0 & 0 & 0 & 0 & B & -(A+B) & A \\ 0 & 0 & 0 & 0 & 0 & C & -C \end{pmatrix}$$

The minor of a non-diagonal entry yields a block triangular matrix with three blocks: one of $\mathcal{M}$, $\mathcal{U}$, and a middle block which is itself triangular.

Whether the 'middle block' contains B or A along the diagonal depends on whether the minor being computed is above or below the main diagonal.

Computing the Inverse - Matrix Minors

Next we need to compute the minors of $\mathcal{M}_k$. We'll investigate the minors of $\mathcal{M}_7$ and use its example to help generalize to $\mathcal{M}_k$.

$$\mathcal{M}_7 = \begin{pmatrix} -(A+B) & A & 0 & 0 & 0 & 0 & 0 \\ B & -(A+B) & A & 0 & 0 & 0 & 0 \\ 0 & B & -(A+B) & A & 0 & 0 & 0 \\ 0 & 0 & B & -(A+B) & A & 0 & 0 \\ 0 & 0 & 0 & B & -(A+B) & A & 0 \\ 0 & 0 & 0 & 0 & B & -(A+B) & A \\ 0 & 0 & 0 & 0 & 0 & C & -C \end{pmatrix}$$

The minor of a non-diagonal entry yields a block triangular matrix with three blocks: one of $\mathcal{M}$, $\mathcal{U}$, and a middle block which is itself triangular.

Whether the 'middle block' contains B or A along the diagonal depends on whether the minor being computed is above or below the main diagonal.

$\det(\mathcal{U})$ Recursion, Adjugate Formulas

Then $\det(\mathcal{U}_k)$ is given recursively as:

$$\det(\mathcal{U}_k) = -(A+B)\det(\mathcal{M}_{k-1}) - AB\det(\mathcal{U}_{k-2}) \quad k \geq 2$$

Solving in an analogous fashion to $\det(\mathcal{M}_k)$, we get:

$$\det(\mathcal{U}_k) = (-1)^k \frac{A^{k+1} - B^{k+1}}{A - B}.$$

With $\det(\mathcal{U}_k)$ established, we can formulate a closed form expression for adjugate$(M_k)_{ij}$. For the sake of notational succinctness, the boldface $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{U}}$ will denote the determinants of $\mathcal{M}$, $\mathcal{U}$.

$$\text{adjugate}(\mathcal{M}_k)_{ij} = \begin{cases} \boldsymbol{\mathcal{U}}_{i-1}\boldsymbol{\mathcal{M}}_{k-j} & \text{for } i = j \\ (-1)^{i+j}A^{j-i}\boldsymbol{\mathcal{U}}_{i-1}\boldsymbol{\mathcal{M}}_{k-j} & \text{for } i < j \\ (-1)^{i+j}B^{i-j}\boldsymbol{\mathcal{U}}_{j-1}\boldsymbol{\mathcal{M}}_{k-i} & \text{for } i > j \end{cases}$$

Adjugate Matrix

The fundamental matrix for the 20-cycle, N_{20}, is given as the following:

$$N_{20} = \frac{1}{\mathcal{M}_5} \times \begin{pmatrix} \mathcal{M}_4 & -A\mathcal{M}_3 & A^2\mathcal{M}_2 & -A^3\mathcal{M}_1 & A^4 \\ -B\mathcal{M}_3 & \mathcal{U}_1\mathcal{M}_3 & -A\mathcal{U}_1\mathcal{M}_2 & A^2\mathcal{U}_1\mathcal{M}_1 & -A^3\mathcal{U}_1 \\ B^2\mathcal{M}_2 & -B\mathcal{U}_1\mathcal{M}_2 & \mathcal{U}_2\mathcal{M}_2 & -A\mathcal{U}_2\mathcal{M}_1 & A^2\mathcal{U}_2 \\ -B^3\mathcal{M}_1 & B^2\mathcal{U}_1\mathcal{M}_1 & -B\mathcal{U}_2\mathcal{M}_1 & \mathcal{U}_3\mathcal{M}_1 & -A\mathcal{U}_3 \\ B^4 & -B^3\mathcal{U}_1 & B^2\mathcal{U}_2 & -B\mathcal{U}_3 & \mathcal{U}_4 \end{pmatrix}$$

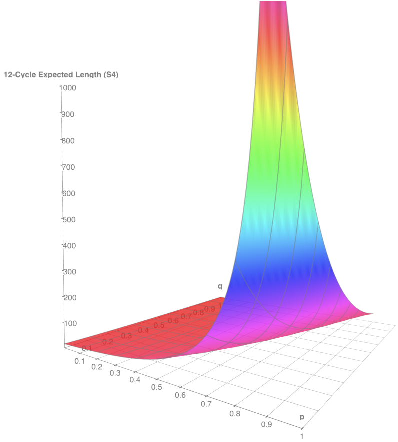
The third row sum of N_{20} is given by:

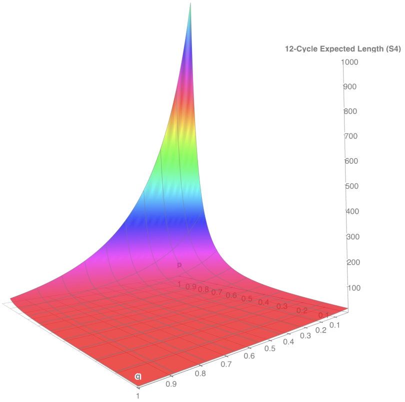
$$(B^2\mathcal{M}_2 - B\mathcal{U}_1\mathcal{M}_2 + \mathcal{U}_2\mathcal{M}_2 - A\mathcal{U}_2\mathcal{M}_1 + A^2\mathcal{U}_2)$$

Generalizing to the i^{th} row sum of N_{4k}:

$$\left[\left(\sum_{x=1}^{i}(-1)^{i+x}B^{i-x}\mathcal{U}_{x-1}\mathcal{M}_{k-i} + \sum_{y=0}^{k-i}A^y\mathcal{U}_{i-1}\mathcal{M}_{k-i-y}\right) - \mathcal{U}_{i-1}\mathcal{M}_{k-i}\right]$$

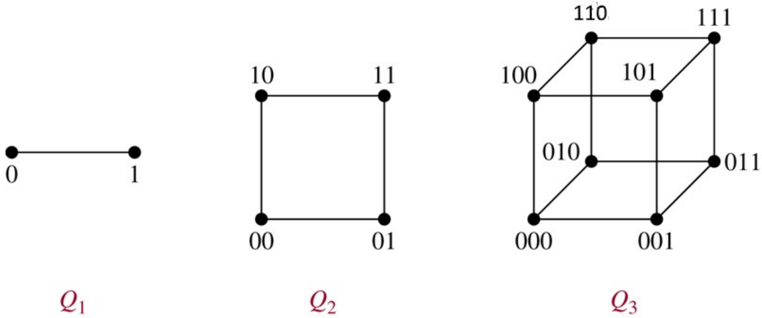
Graph - 12-cycle





Section Preface

Hypercube Graphs



Deriving Transition Matrix

As we saw on the section preface, we can represent the vertices of an n-dimension hypercube as length n binary words, allowing for initial 0's.

With this, the following properties of an n-dimension hypercube follow:

- Each vertex is adjacent to n vertices.
- The distance between two vertices equals the number of distinct positions in their corresponding binary words.
- The maximum distance between two vertices is n.

$$\underbrace{\{1111\} \qquad \{0000\}}_{\text{Max distance}} \qquad\qquad \underbrace{\{1010\} \qquad \{0011\}}_{\text{Distance} = 2}$$

Therefore we see that our transition probabilities will always depend on the current distance between the agents. A superscript will denote the current distance (e.g. $P_s^3(C)$).

(I-Q)

The transition matrices for the hypercube graphs depend on two cases:

1. n even (analogous to n-cycle case 1)
2. n odd (analogous to n-cycle case 2)

We will focus on the first of these cases. Define $\mathcal{Q}_k$ the family of $(I - Q)$ matrices when $n = 2k, k \in \mathbb{N}_1$. The general $\mathcal{Q}_k$ is given below:

$$\mathcal{Q}_k = \begin{array}{c c} \begin{array}{ccccc} S_2 & S_4 & \dots & S_{2k-2} & S_{2k} \end{array} & \\ \begin{pmatrix} -(B_2+A_2) & A_2 & \dots & 0 & 0 \\ B_4 & -(B_4+A_4) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -(B_{2k-2}+A_{2k-2}) & A_{2k-2} \\ 0 & 0 & \dots & C_{2k} & -C_{2k} \end{pmatrix} & \begin{array}{c} S_2 \\ S_4 \\ \vdots \\ S_{2k-2} \\ S_{2k} \end{array} \end{array}$$

Where: $A_i = -P_s^i(F)P_z^{i+1}(F)$ $B_i = -P_s^i(C)P_z^{i-1}(C)$ $C_{2k} = -P_z^{2k-1}(C)$

Determinant Recursion

Below is $\mathcal{Q}_k$. While is is possible to construct submatrices in the same fashion as the *n*-cycle, the submatrices do not yield previous iterations of $\mathcal{Q}_k$. **The red submatrix is not $\mathcal{Q}_{k-1}$, since the entries depend on k.**

$$
\begin{pmatrix}
-(A_2+B_2) & A_2 & 0 & \dots & 0 & 0 & 0 \\
B_4 & -(A_4+B_4) & A_4 & \dots & 0 & 0 & 0 \\
0 & B_6 & -(A_6+B_6) & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \dots & -(A_{2k-4}+B_{2k-4}) & A_{2k-4} & 0 \\
0 & 0 & 0 & \dots & B_{2k-2} & -(A_{2k-2}+B_{2k-2}) & A_{2k-2} \\
0 & 0 & 0 & \dots & 0 & C_{2k} & -C_{2k}
\end{pmatrix}
$$

This inhibits us from using Binet's formula to calculate $\det(\mathcal{Q}_k)$. Fortunately, we are able to find a closed form solution using a different technique.

New Determinant Formula (1)

Let's calculate the first few determinants of $\mathcal{Q}_k$:

$$\begin{aligned}
\det(\mathcal{Q}_1) &= -C_2 \\
\det(\mathcal{Q}_2) &= (C_4)(A_2 + B_2) - (A_2)(C_4) = C_4 B_2 \\
\det(\mathcal{Q}_3) &= -(A_6 + B_6)(B_4 C_6) - A_6(B_4 C_6) = -C_6 B_2 B_4
\end{aligned}$$

One can prove this cancellation between the (1,1) and (1,2) pivot always occurs, yielding the following formula for $\det(\mathcal{Q}_k)$:

$$\det(\mathcal{Q}_k) = (-1)^k C_{2k} B_2 B_4 \dots B_{2k-2}$$

Or:

$$\det(\mathcal{Q}_k) = (-1)^k C_{2k} \prod_{x=1}^{k-1} B_{2x}$$

New Determinant Formula (2)

Furthermore, one can show that:

$$B_i = \frac{[(k-1)q + i]\,(1-p)(i-1)}{k^2}$$
$$C_{2k} = \frac{-(1-p)(2k-1)}{2k}$$

Putting this together, we get that:

$$\det(\mathcal{Q}_k) = \frac{(-1)^{k+1}(2k-1)(1-p)^k \prod_{x=1}^{k-1}[(k-2x)q + 2x]\,(2x-1)}{2k^{2k-1}}$$

Matrix of Minors

With a formula for $\det(\mathcal{Q}_k)$, let's now try to calculate our matrix minors.

$$
\begin{pmatrix}
-(A_2+B_2) & A_2 & 0 & 0 & 0 & 0 & 0 \\
B_4 & -(A_4+B_4) & A_4 & 0 & 0 & 0 & 0 \\
0 & B_6 & -(A_6+B_6) & A_6 & 0 & 0 & 0 \\
0 & 0 & B_8 & -(A_8+B_8) & A_8 & 0 & 0 \\
0 & 0 & 0 & B_{10} & -(A_{10}+B_{10}) & A_{10} & 0 \\
0 & 0 & 0 & 0 & B_{12} & -(A_{12}+B_{12}) & A_{12} \\
0 & 0 & 0 & 0 & 0 & C_{14} & -C_{14}
\end{pmatrix}
$$

The 3-block pattern we saw for the n-cycle is present here in the hypercube as well. The middle block is still triangular. At this time, a formula for the upper, unaberrated block $\mathcal{U}$ is still unknown.

Adjugate Formulas

We can now construct our formula for the adjugate when $i > j$. $i < j$ follows analogously. Let's first remind ourselves of our $\det(\mathcal{Q}_k)$ formula:

$$\det(\mathcal{Q}_k) = \frac{(-1)^{k+1}(2k-1)(1-p)^k \prod_{x=1}^{k-1}[(k-2x)q + 2x]\,(2x-1)}{2k^{2k-1}}$$

The n-cycle formula may be helpful as well:

$$\text{adjugate}(M_k)_{ij} = (-1)^{i+j} B^{i-j} \boldsymbol{\mathcal{U}}_{j-1} \boldsymbol{\mathcal{M}}_{k-i} \qquad \text{for } i > j$$

Let $\text{U} = \det(\mathcal{U}_{j-1})$. We get that $\text{adjugate}(Q_k)_{ij}$ is equivalent to the following expression:

$$\frac{1}{U} \cdot \frac{(-1)^{k-i+1}(2k-1)(1-p)^{k-i} \prod_{x=1}^{k-i-1}[(k-2i-2x)q + 2i + 2x]\,(2i+2x-1)}{2k^{2k-2i-1}}$$

Final Thoughts

- We provided solutions for the expected length on the Petersen Graph, n-cycles, and hypercube graphs.
- We are still working on making our hypercube solution completely closed-form.
- We are still working on interpreting the graphs we produce.
- Note that not all symmetric graphs can be solved in such a way. A Toroidal graph, for example, is symmetric, but produces a much different transition matrix.

Thank you for watching! Questions?