Introducing features of agency into computational models of infectious disease

Ethan Kelly
e.kelly.1@research.gla.ac.uk

27 May 2021

Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

What is a Graph?

What is a graph (or network)?

What is a Graph?

What is a graph (or network)?

- A set of objects where some pairs are related.

What is a Graph?

What is a graph (or network)?

- A set of objects where some pairs are related.
- Objects are called *vertices* or *nodes*

What is a Graph?

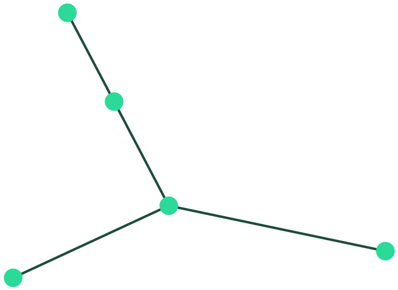
What is a graph (or network)?

- A set of objects where some pairs are related.
- Objects are called *vertices* or *nodes*
- Relations between objects are called *edges*.

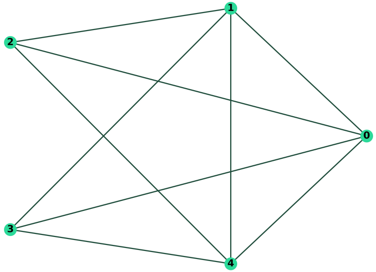
What is a Graph?

What is a graph (or network)?

- A set of objects where some pairs are related.
- Objects are called *vertices* or *nodes*
- Relations between objects are called *edges*.



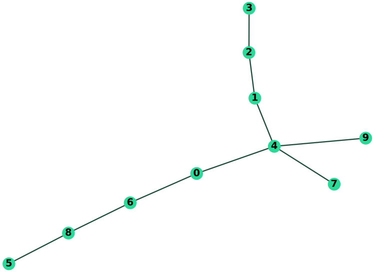
Examples of Graphs



Example: Erdős-Rényi

Random graph with a given number of vertices (in this case, 5) and a probability with which an edge exists between any two vertices.

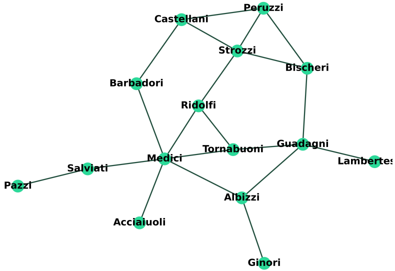
Examples of Graphs



Example: Tree

A tree is a special type of graph where *any two vertices are connected by exactly one path.*

Examples of Graphs



Example: Florentine Families Graph

Depicts the marital alliances between Renaissance Florentine families [4].

Using Graphs to Model Disease Spread

The Firefighter Problem (Firefighter) was first introduced by Hartnell in 1995 [2] and models an outbreak of fire with a firefighter strategically blocking its path.

Using Graphs to Model Disease Spread

The Firefighter Problem (Firefighter) was first introduced by Hartnell in 1995 [2] and models an outbreak of fire with a firefighter strategically blocking its path.

- At $t = 0$, a fire breaks out at some vertex in the graph.

Using Graphs to Model Disease Spread

The Firefighter Problem (Firefighter) was first introduced by Hartnell in 1995 [2] and models an outbreak of fire with a firefighter strategically blocking its path.

- At $t = 0$, a fire breaks out at some vertex in the graph.
- Firefighter then 'protects' some other vertex.

Using Graphs to Model Disease Spread

The Firefighter Problem (Firefighter) was first introduced by Hartnell in 1995 [2] and models an outbreak of fire with a firefighter strategically blocking its path.

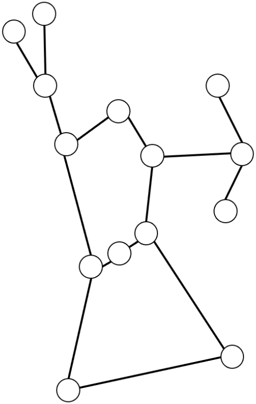
- At $t = 0$, a fire breaks out at some vertex in the graph.
- Firefighter then ‘protects’ some other vertex.
- Fire spreads to any adjacent vertices neither protected nor burnt.

Using Graphs to Model Disease Spread

The Firefighter Problem (Firefighter) was first introduced by Hartnell in 1995 [2] and models an outbreak of fire with a firefighter strategically blocking its path.

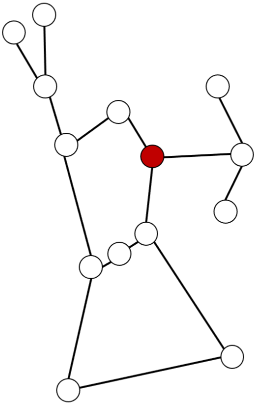
- At $t = 0$, a fire breaks out at some vertex in the graph.
- Firefighter then ‘protects’ some other vertex.
- Fire spreads to any adjacent vertices neither protected nor burnt.
- Firefighter protects another vertex, the fire spreads again and so on.

Example



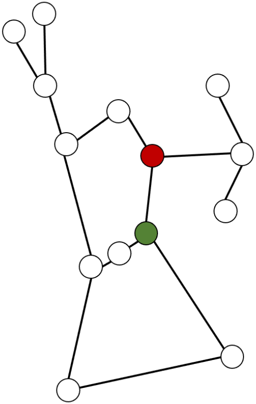
Original Graph

Example



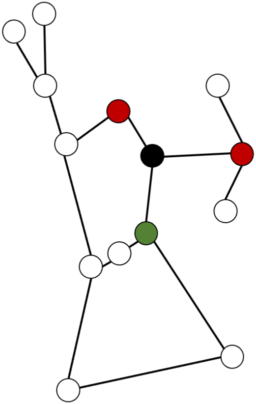
Outbreak

Example



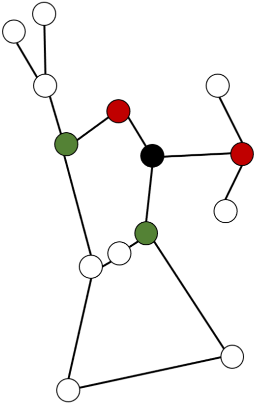
Defence

Example



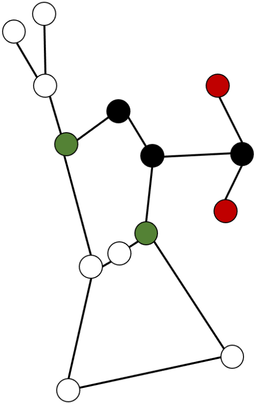
Fire Spreads

Example



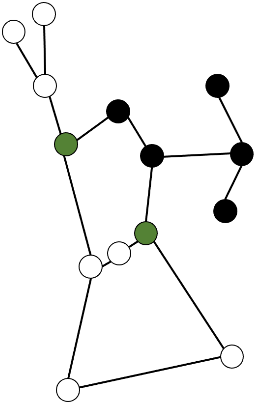
Defence

Example



Fire Spreads

Example



Fire contained

Firefighter as a Model for Disease Spread

Benefits:

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.
- We can change graphs to suit context - e.g. small world, preferential attachment, particular density.

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.
- We can change graphs to suit context - e.g. small world, preferential attachment, particular density.

Limitations:

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.
- We can change graphs to suit context - e.g. small world, preferential attachment, particular density.

Limitations:

- Fairly rudimentary model for disease spread but already NP-hard.

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.
- We can change graphs to suit context - e.g. small world, preferential attachment, particular density.

Limitations:

- Fairly rudimentary model for disease spread but already NP-hard.
- Defence and infection are discrete but epidemic propagation is a stochastic process.

Firefighter as a Model for Disease Spread

Benefits:

- Previous work means lots of results we can use.
- We can change graphs to suit context - e.g. small world, preferential attachment, particular density.

Limitations:

- Fairly rudimentary model for disease spread but already NP-hard.
- Defence and infection are discrete but epidemic propagation is a stochastic process.
- Only interventions in halting disease spread are *external*, no way for individuals to avoid contraction personally.

Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

Attributes of Agency

If each vertex in the graph represents an individual agent, we can assign several attributes. We are particularly interested in each agent's *internal inclination towards protection*. In context, this might involve:

Attributes of Agency

If each vertex in the graph represents an individual agent, we can assign several attributes. We are particularly interested in each agent's *internal inclination towards protection*. In context, this might involve:

- Wearing PPE correctly

Attributes of Agency

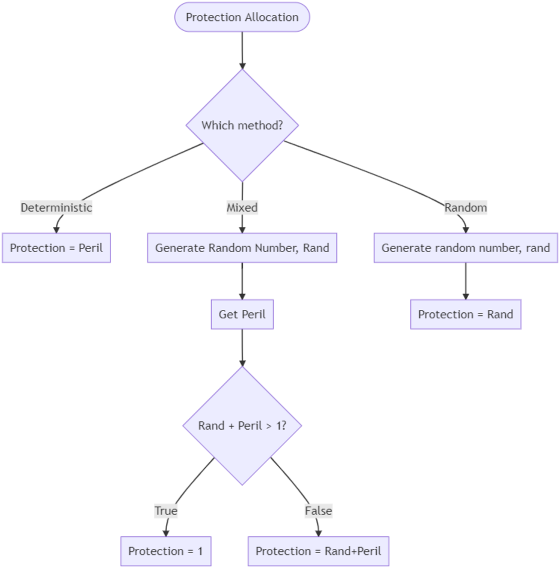
If each vertex in the graph represents an individual agent, we can assign several attributes. We are particularly interested in each agent's *internal inclination towards protection*. In context, this might involve:

- Wearing PPE correctly
- Hand hygiene

Attributes of Agency

If each vertex in the graph represents an individual agent, we can assign several attributes. We are particularly interested in each agent's *internal inclination towards protection*. In context, this might involve:

- Wearing PPE correctly
- Hand hygiene
- Strict physical distancing


Protection Allocation
Which method?
Deterministic
Mixed
Random
Protection = Peril
Generate Random Number, Rand
Generate random number, rand
Get Peril
Protection = Rand
Rand + Peril > 1?
True
False
Protection = 1
Protection = Rand+Peril

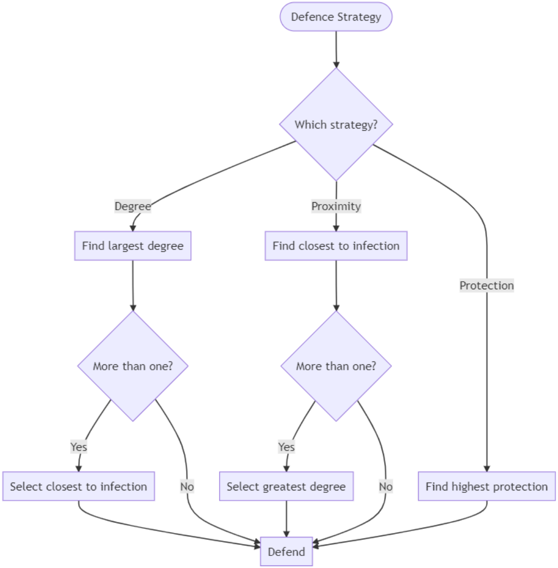
Defence Strategies

We should now ask what these amendments mean for how the game is played. In the usual formulation, general rule of thumb: *for sparse graphs, defend based on proximity to fire (breaking ties on degree); for dense graphs, defend based on degree (breaking ties on proximity)*.

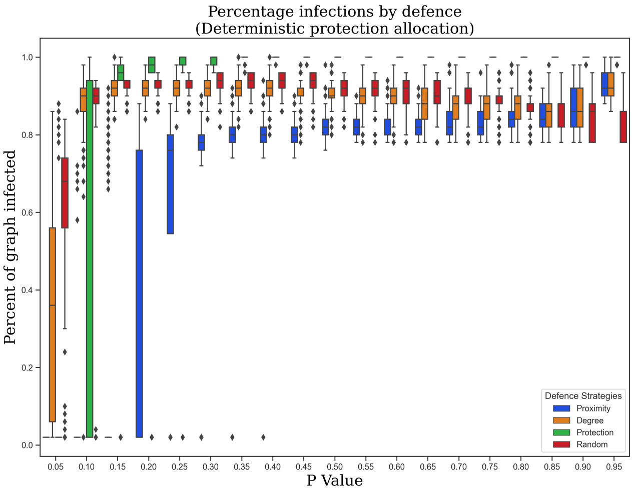
Defence Strategies

We should now ask what these amendments mean for how the game is played. In the usual formulation, general rule of thumb: *for sparse graphs, defend based on proximity to fire (breaking ties on degree); for dense graphs, defend based on degree (breaking ties on proximity)*.

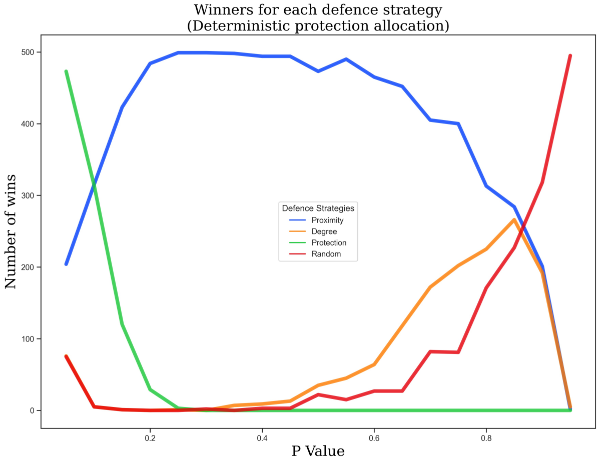
However, in our adjusted formulation we have more candidates for defence strategies. One such novel strategy is to defend based on highest agent protection rating.


Defence Strategy
Which strategy?
Degree
Proximity
Protection
Find largest degree
Find closest to infection
More than one?
More than one?
Yes
No
Yes
No
Select closest to infection
Select greatest degree
Find highest protection
Defend

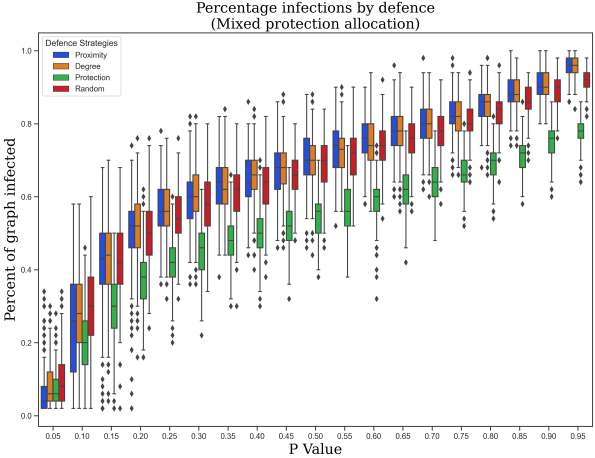
Deterministic Protection



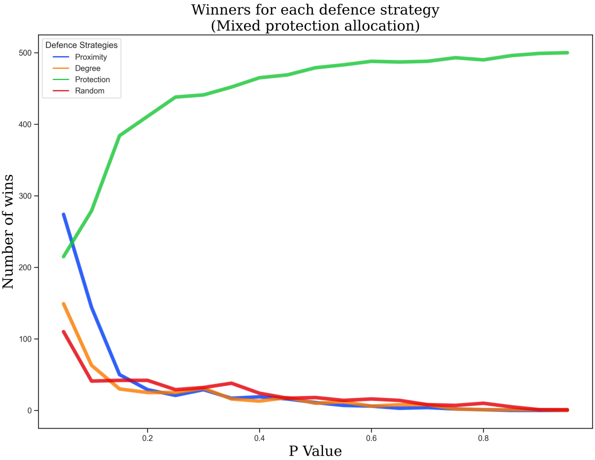
Deterministic Protection



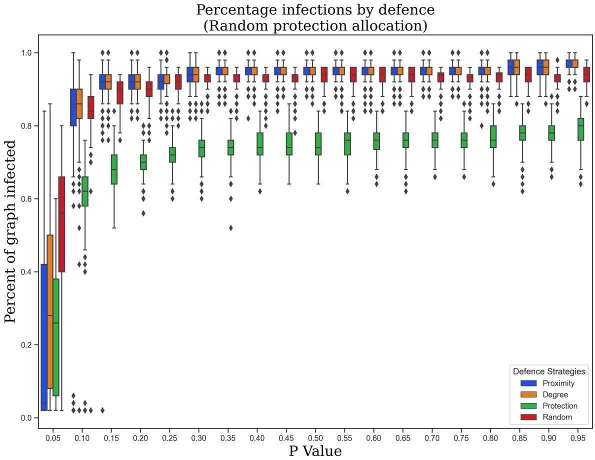
Mixed Protection



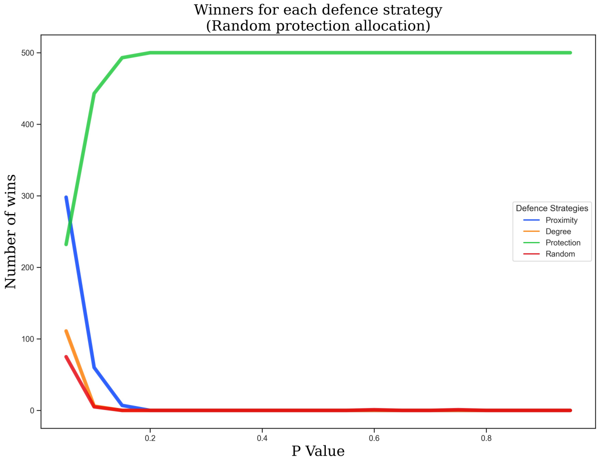
Mixed Protection



Random Protection



Random Protection



Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

Examine probability of an agent being in a given state:

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A
- $\langle A_i B_i \rangle$ - probability of i and j being in states A and B respectively.

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A
- $\langle A_i B_i \rangle$ - probability of i and j being in states A and B respectively.
- β_i - per-link infection rate for individual i.

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A
- $\langle A_i B_i \rangle$ - probability of i and j being in states A and B respectively.
- β_i - per-link infection rate for individual i.
- γ_i - recovery rate for i.

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A
- $\langle A_i B_i \rangle$ - probability of i and j being in states A and B respectively.
- β_i - per-link infection rate for individual i.
- γ_i - recovery rate for i.
- For transmission matrix T, assign $T_{ij} = \beta_i$ if there is a route of infection between i and j ($T_{ij} = 0$ otherwise).

Examine probability of an agent being in a given state:

- $\langle A_i \rangle$ - time-independent probability of i being in state A
- $\langle A_i B_i \rangle$ - probability of i and j being in states A and B respectively.
- β_i - per-link infection rate for individual i.
- γ_i - recovery rate for i.
- For transmission matrix T, assign $T_{ij} = \beta_i$ if there is a route of infection between i and j ($T_{ij} = 0$ otherwise).
- Often, we will consider unweighted and undirected graphs, but in general T_{ij} may not equal T_{ji}.

We can replace β_i with a term involving a transmission matrix of a network to begin extending the usual *SIR* model into a network realm.

$$
\begin{aligned}
\langle \dot{S}_i \rangle &= -\sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle \\
\langle \dot{I}_i \rangle &= \sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle - \gamma_i \langle I \rangle \\
\langle \dot{R}_i \rangle &= \gamma_i \langle I \rangle,
\end{aligned}
$$

which are the evolution equations given in [3].

Protected State

- ζ_i - probability that we defend individual i,
- α_i - efficacy of the protection measure for individual i (may decay over time and vary from person to person).

Protected State

- ζ_i - probability that we defend individual i,
- α_i - efficacy of the protection measure for individual i (may decay over time and vary from person to person).

Using these rates of protection and effectiveness, for fixed population size the differential equations become:

Protected State

- ζ_i - probability that we defend individual i,
- α_i - efficacy of the protection measure for individual i (may decay over time and vary from person to person).

Using these rates of protection and effectiveness, for fixed population size the differential equations become:

$$\langle \dot{S}_i \rangle = \alpha_i \langle P_i \rangle - \sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle - \zeta_i \langle S_i \rangle \tag{1}$$

$$\langle \dot{I} \rangle = \sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle - \gamma_i \langle I \rangle \tag{2}$$

$$\langle \dot{R}_i \rangle = \gamma_i \langle I \rangle \tag{3}$$

$$\langle \dot{P}_i \rangle = \zeta_i \langle S_i \rangle - \alpha_i \langle P_i \rangle . \tag{4}$$

Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

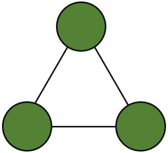
3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

Numbers of Equations

Consider the triangle graph. The equations required to precisely express the system *SIR* dynamics of this network are as below [3].



$$\text{6 singles: } \langle \dot{S_1} \rangle, \langle \dot{S_2} \rangle, \langle \dot{S_3} \rangle, \langle \dot{I_1} \rangle, \langle \dot{I_2} \rangle, \langle \dot{I_3} \rangle. \tag{5}$$

$$\text{6 doubles: } \langle \dot{S_1 I_2} \rangle, \langle \dot{I_1 S_2} \rangle, \langle \dot{S_1 I_3} \rangle, \langle \dot{I_1 S_3} \rangle, \langle \dot{S_2 I_3} \rangle, \langle \dot{I_2 S_3} \rangle. \tag{6}$$

$$\text{6 triples: } \langle \dot{S_1 I_2 I_3} \rangle, \langle \dot{S_1 I_2 S_3} \rangle, \langle \dot{S_1 S_2 I_3} \rangle, \langle \dot{I_1 S_2 S_3} \rangle, \langle \dot{I_1 I_2 S_3} \rangle, \langle \dot{I_1 S_2 I_3} \rangle. \tag{7}$$

Using the equations for the *SIRP* model, we have the following equation requirements:

$$
\begin{aligned}
\text{9 singles: (5) and } & \langle \dot{P}_1 \rangle, \langle \dot{P}_2 \rangle, \langle \dot{P}_3 \rangle. \\
\text{18 doubles: (6) and } & \langle S_1 \dot{} P_2 \rangle, \langle P_1 \dot{} S_2 \rangle, \langle I_1 \dot{} P_2 \rangle, \langle P_1 \dot{} I_2 \rangle, \langle S_1 \dot{} P_3 \rangle, \langle P_1 \dot{} S_3 \rangle, \\
& \langle I_1 \dot{} P_3 \rangle, \langle P_1 \dot{} I_3 \rangle, \langle S_2 \dot{} P_3 \rangle, \langle P_2 \dot{} S_3 \rangle, \langle I_2 \dot{} P_3 \rangle, \langle P_2 \dot{} I_3 \rangle. \\
\text{24 triples: (7) and } & \langle S_1 \dot{S}_2 P_3 \rangle, \langle S_1 \dot{P}_2 S_3 \rangle, \langle S_1 \dot{I}_2 P_3 \rangle, \langle S_1 \dot{P}_2 I_3 \rangle, \langle S_1 \dot{P}_2 P_3 \rangle, \\
& \langle I_1 \dot{S}_2 P_3 \rangle, \langle I_1 \dot{P}_2 S_3 \rangle, \langle I_1 \dot{I}_2 P_3 \rangle, \langle I_1 \dot{P}_2 I_3 \rangle, \langle I_1 \dot{P}_2 P_2 \rangle, \\
& \langle P_1 \dot{S}_2 S_3 \rangle, \langle P_1 \dot{S}_2 I_3 \rangle, \langle P_1 \dot{I}_2 S_3 \rangle, \langle P_1 \dot{I}_2 I_3 \rangle, \langle P_1 \dot{P}_2 I_3 \rangle, \\
& \langle P_1 \dot{P}_2 S_3 \rangle, \langle P_1 \dot{S}_2 P_3 \rangle, \langle P_1 \dot{I}_2 P_3 \rangle.
\end{aligned}
$$

What are closures?

What are closures?

- In the previous section, we saw equations for singles

What are closures?

- In the previous section, we saw equations for singles
- We have also seen equations needed for the triangle network (including doubles and triples)

What are closures?

- In the previous section, we saw equations for singles
- We have also seen equations needed for the triangle network (including doubles and triples)
- How many equations do we need in general?

What are closures?

- In the previous section, we saw equations for singles
- We have also seen equations needed for the triangle network (including doubles and triples)
- How many equations do we need in general?

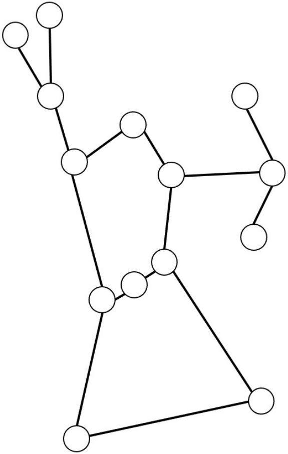
The number of other equations will usually increase when we require closures, as closure equations will sometimes require equations that we don't usually consider as they are dynamically uninteresting, e.g. $\langle S_i S_j \rangle$ where i and j are adjacent.

Consider the equations below for calculating singles and pairs:

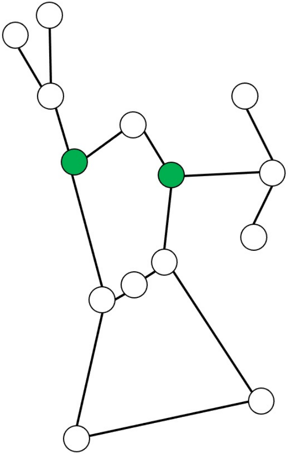
$$
\begin{aligned}
\langle \dot{S_i} \rangle &= -\sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle, \\
\langle \dot{I_i} \rangle &= \sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle - \gamma_i \langle I_i \rangle, \\
\langle \dot{S_i I_j} \rangle &= \sum_{k=1, k \neq i}^{N} T_{jk} \langle S_i S_j I_k \rangle - \sum_{k=1, k \neq j}^{N} T_i k \langle I_k S_i I_j \rangle \\
&\quad - T_{ij} \langle S_i I_j \rangle - \gamma_i \langle S_i I_j \rangle, \\
\langle \dot{S_i S_j} \rangle &= -\sum_{k=1, k \neq j}^{N} T_{ik} \langle I_k S_i S_j \rangle - \sum_{k=1, k \neq i}^{N} T_{jk} \langle S_i S_j I_k \rangle,
\end{aligned}
$$

$$
\begin{aligned}
\langle \dot{S}_i \rangle &= -\sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle, \\
\langle \dot{I}_i \rangle &= \sum_{j=1}^{N} T_{ij} \langle S_i I_j \rangle - \gamma_i \langle I_i \rangle, \\
\langle \dot{S_i I_j} \rangle &= \sum_{k=1, k \neq i}^{N} T_{jk} \langle S_i S_j I_k \rangle - \sum_{k=1, k \neq j}^{N} T_i k \langle I_k S_i I_j \rangle \\
&\quad - T_{ij} \langle S_i I_j \rangle - \gamma_i \langle S_i I_j \rangle, \\
\langle \dot{S_i S_j} \rangle &= -\sum_{k=1, k \neq j}^{N} T_{ik} \langle I_k S_i S_j \rangle - \sum_{k=1, k \neq i}^{N} T_{jk} \langle S_i S_j I_k \rangle,
\end{aligned}
$$

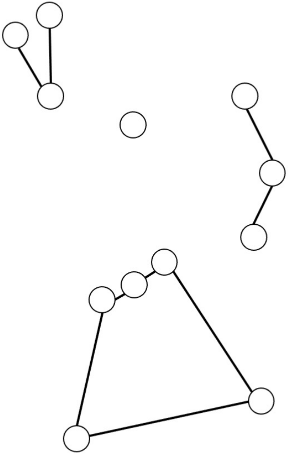
This is *not* a closed system - we require equations for triples.



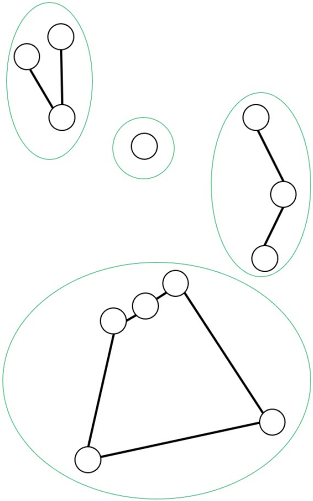
Original Graph



Identified Cut Vertices



Removed Cut Vertices



4 Connected Components

Calculating Closures from Cut-Vertex Sets

Let $G = \{V, E\}$ be a graph on N vertices.

Calculating Closures from Cut-Vertex Sets

Let $G = \{V, E\}$ be a graph on N vertices. Consider a connected subset of vertices $F = v_1, v_2, \ldots, v_k \subset V$

Calculating Closures from Cut-Vertex Sets

Let $G = \{V, E\}$ be a graph on N vertices. Consider a connected subset of vertices $F = v_1, v_2, \ldots, v_k \subset V$ and assume $\exists v_{i^*} \in F$ a cut-vertex in G, such that $F \setminus \{v_{i^*}\}$ is partitioned into at least two disjointed components with vertices $F_1 = v_1, v_2, \ldots, v_{i-1}$ and $F_2 = v_{i+1}, v_{i+2}, \ldots, v_k$ belonging to any such two, distinct and disjointed components or subnetworks.

Calculating Closures from Cut-Vertex Sets

Let $G = \{V, E\}$ be a graph on N vertices. Consider a connected subset of vertices $F = v_1, v_2, \ldots, v_k \subset V$ and assume $\exists v_{i^*} \in F$ a cut-vertex in G, such that $F \setminus \{v_{i^*}\}$ is partitioned into at least two disjointed components with vertices $F_1 = v_1, v_2, \ldots, v_{i-1}$ and $F_2 = v_{i+1}, v_{i+2}, \ldots, v_k$ belonging to any such two, distinct and disjointed components or subnetworks. Then the following equation holds:

$$\langle Z_{v_1} Z_{v_2}, \ldots Z_{v_{i-1}} S_{v_i^*} Z_{v_{i+1}} \ldots Z_{v_k} \rangle(t) = \frac{\langle Z_{v_1} Z_{v_2} \ldots Z_{v_{i-1}} S_{v_i^*} \rangle(t) \langle S_{v_i^*} Z_{v_{i+1}} \ldots Z_{v_k} \rangle}{\langle S_{v_i^*} \rangle(t)}$$

where $Z_{v_i} \in \{S, I\}$ $\forall v_i \neq v_{i^*}$. [3]

Bounding

Denote by $\mathrm{Ind}(v_{i_j})$ for $j = 1, 2, \ldots, L$ the number of subgraph v_{i_j} belongs to.

Upper bound:

$$N_{EQ}(G) = \sum_{i=1}^{P} m_i f_i - 2\sum_{j=1}^{L}(\mathrm{Ind}(v_{i_j}) - 1).$$

We take a sum across number of equations for all subgraphs and subtract unnecessary multiplications from cut-vertices being replaced into produced sub-graphs. [3]

Bounding

Upper bound:

$$N_{EQ}(G) = \sum_{i=1}^{P} m_i f_i - 2\sum_{j=1}^{L}(\mathrm{Ind}(v_{i_j}) - 1).$$

Note: m_i denotes the number of equations required for subnetwork i, e.g.

- An edge requires 7 equations (4 for singles, 3 for pairs)
- A triangle requires 22 equations
- A cycle on 4 vertices requires 45 equations

We can in fact bound these multipliers. From the requirements of a subgraph (connected, no cut vertices), for a number of vertices n our worst-case scenario is the complete graph (everything is connected to everything else) and our best-case scenario is the cycle on n vertices.

Outline

1 Introduction to Computational Disease Modelling
- Graph Theory
- Games on Graphs

2 Extending existing graph models to account for agency
- Attributes of Agency
- Protection Rating Allocation and Defence Strategies

3 Using Compartmental Frameworks
- *SIR* graph model
- Adding a new state

4 Describing Compartmental Graph Models Exactly
- Required Equations
- Closures

5 Conclusion

We discussed:

We discussed:

- Existing approaches to computational models of disease spread

We discussed:

- Existing approaches to computational models of disease spread
- How to extend existing approaches to better account for agency

We discussed:

- Existing approaches to computational models of disease spread
- How to extend existing approaches to better account for agency
- How using a compartmental framework can help introduce agency formally

We discussed:

- Existing approaches to computational models of disease spread
- How to extend existing approaches to better account for agency
- How using a compartmental framework can help introduce agency formally
- Feasibility of describing compartmental graph models exactly

References

- S. Finbow and G. MacGillivray, *The firefighter problem: A survey of results, directions and questions*, The Australasian Journal of Combinatorics, 43 (2009).
- B. L. Hartnell, *Firefighter! an application of domination*, in 25th Manitoba Conference on Combinatorial Mathematics and Computing, University of Manitoba in Winnipeg, Canada, 1995.
- I. Z. Kiss, C. G. Morris, F. Sélley, P. L. Simon, and R. R. Wilkinson, *Exact deterministic representation of markovian SIR epidemics on networks with and without loops*, Journal of Mathematical Biology, 70 (2014), pp. 437–464.
- J. F. Padgett and C. K. Ansell, *Robust action and the rise of the medici, 1400-1434*, American Journal of Sociology, 98 (1993), pp. 1259–1319.

Questions?