UPPSALA
UNIVERSITET

Project in Computer Systems

Covid-19 Simulations

Uppsala University, January 2021

Contributors:

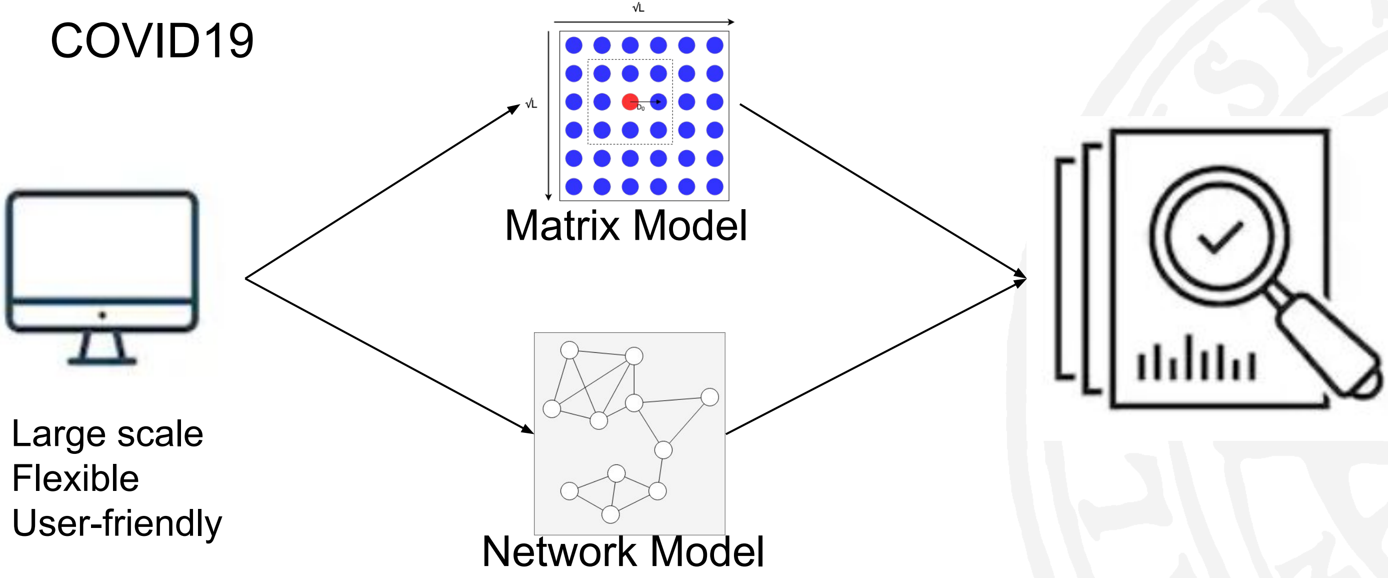
Amina Manafli, Christopher Edberg, Christopher Rydell, Hampus Adolfsson, Henrik Schulze, Isak Borg, Johan Boström, Jonatan Michalak, Nils Söderkvist, Wenhao Zhu

Supervised by: Stefanos Kaxiras

Acknowledgements: our thanks to Efthimios Kaxiras and his group for their guidance.

Introduction / Background / Motivation

- Based on: “Multiple Epidemic Wave Model of the COVID-19 Pandemic: Modeling Study” [Kaxiras, Neofotistos, JMIR’20 vol 22]
- COVID19



Matrix Model

Network Model

- Large scale
- Flexible
- User-friendly

Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

Agent based modelling

- Population of agents in some topology
- On each time step: agents interact with each other
- Based on the interaction the next time step starts with the population in a new state
- Agent interaction is governed by state transition diagram

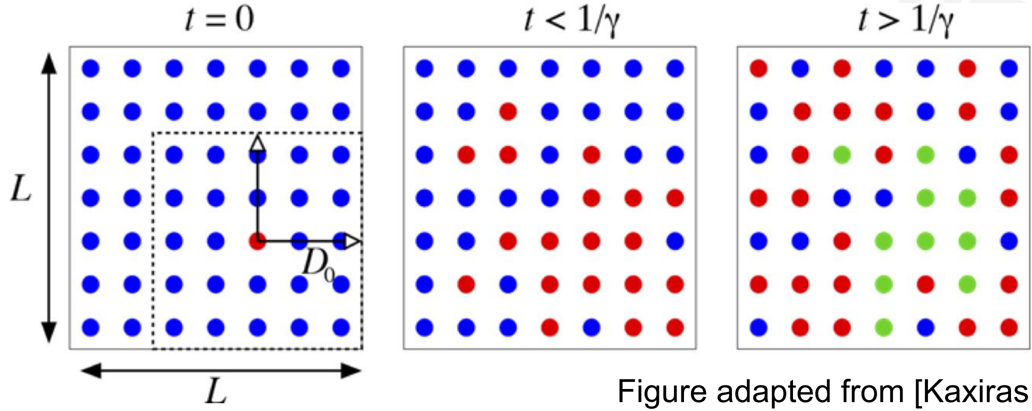


Figure adapted from [Kaxiras, Neofotistos, JMIR'20 vol 22]

The SAIVR Model

S - **Susceptible** An agent that is receptive to the disease

A - **Asymptomatic** An agent that is infected but doesn't show signs of infection

I - **Infectious** An agent that carries the disease and can infect others

V - **Vaccinated** An agent that has been vaccinated has a % chance to be immune

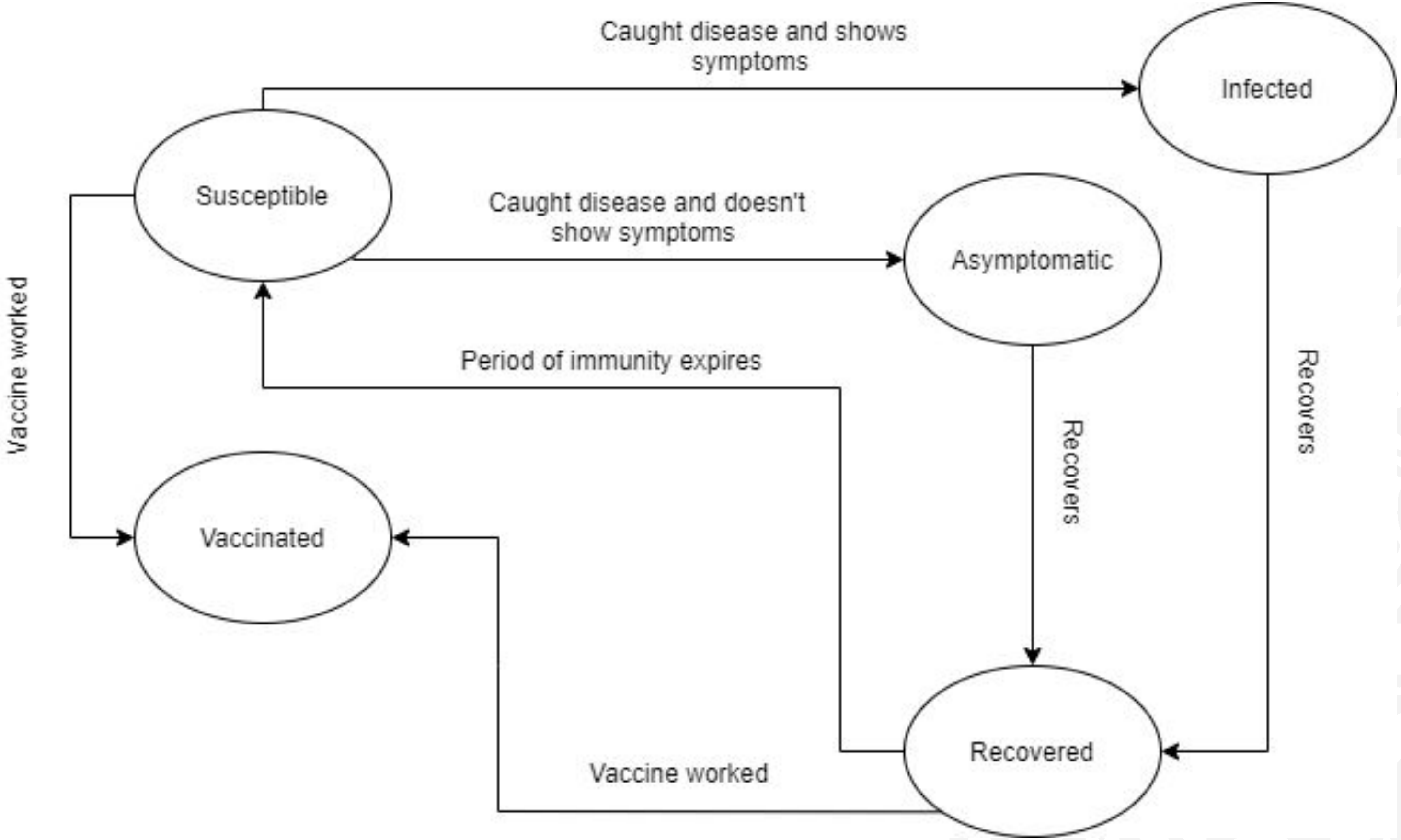
R - **Recovered** An agent that recovered can be reinfected after a period of immunity

These five states are used by both the Matrix Model and the Network Model.

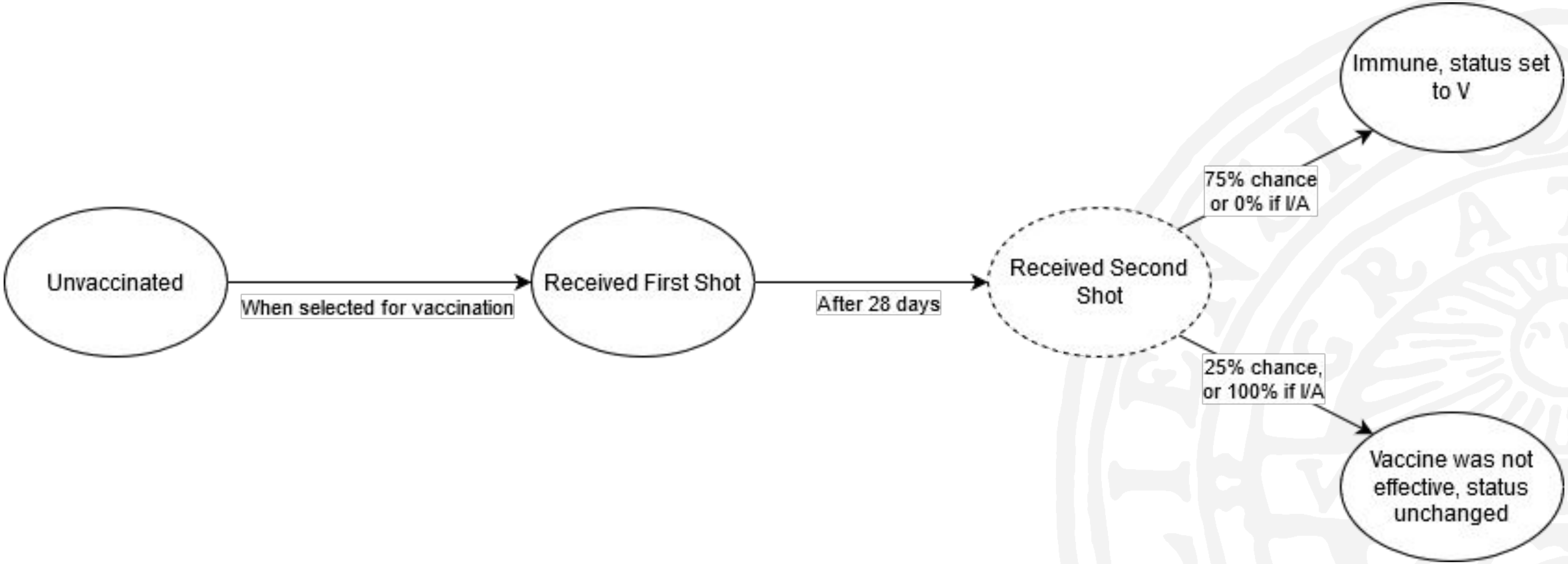
Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

Agent state diagram - Matrix Model


Caught disease and shows symptoms
Infected
Susceptible
Caught disease and doesn't show symptoms
Asymptomatic
Vaccine worked
Period of immunity expires
Recovers
Recovers
Vaccinated
Recovered
Vaccine worked

Pseudo Agent state diagram - Vaccination details



State Transition Parameters - Matrix Model

RECOVERY_RATE - Duration of infection

INFECTION_PROBABILITY - Chance to infect other

STARTER_AGENTS - Initially infected agents

SWAP_AMOUNT - Number of daily travelers

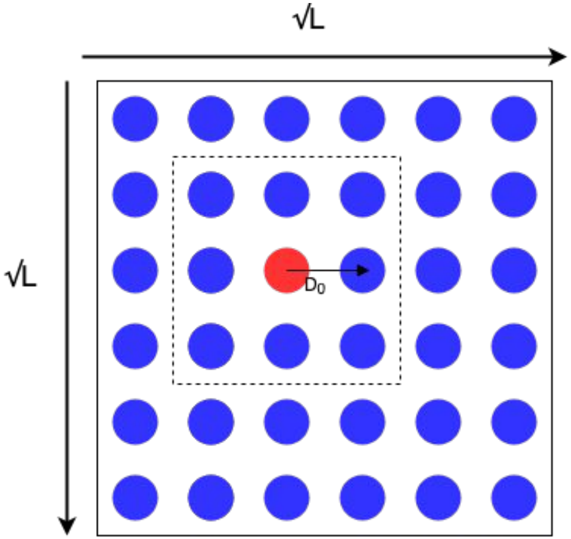
RECOVERED_MIN/MAX_THRESHOLD - Time until agent can be re-infected

VACCINATION_RATE - Duration for vaccine to take effect

VACCINATION_EFFICACY - Success rate of vaccine

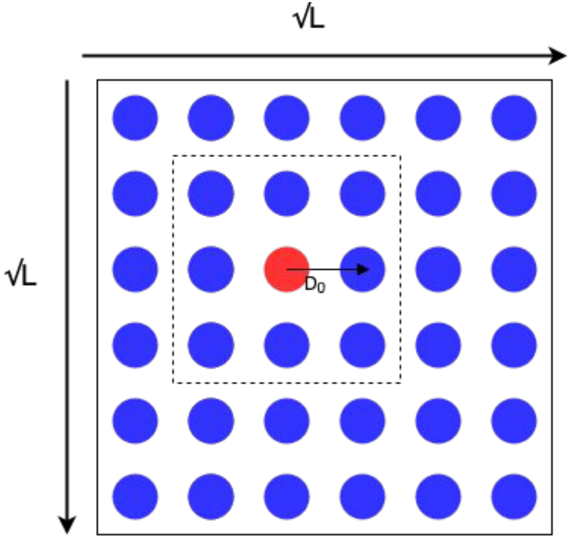
VACCINATIONS_PER_DAY - Daily distributed doses

Communities - Matrix Model

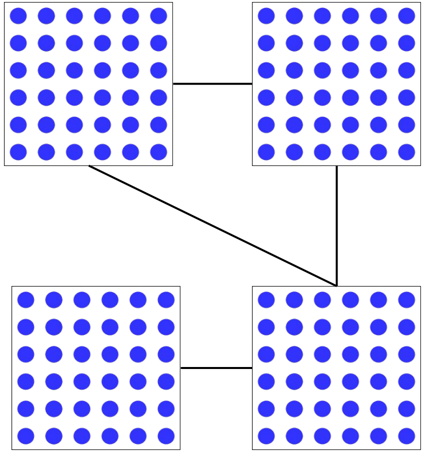


Each community is represented as a uniform square grid

Communities - Matrix Model

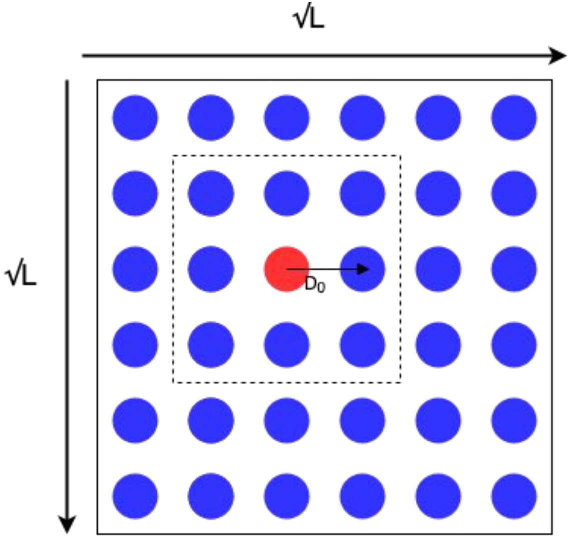


Each community is represented as a uniform square grid

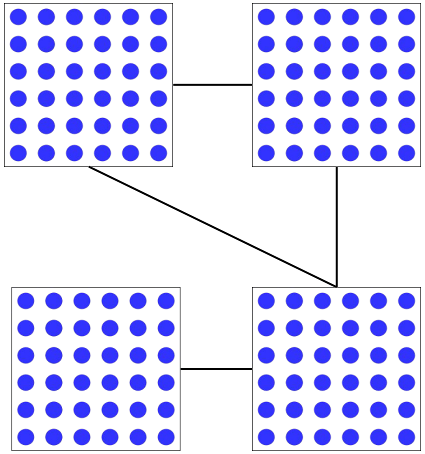


Communities are connected by using their geographical proximities

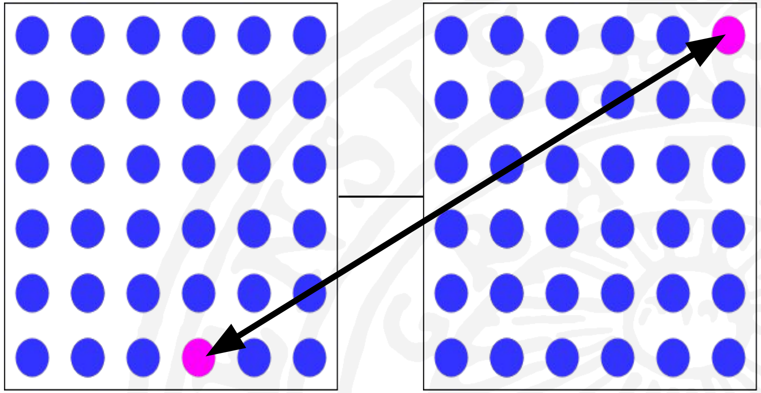
Communities - Matrix Model



Each community is represented as a uniform square grid



Communities are connected by using their geographical proximities

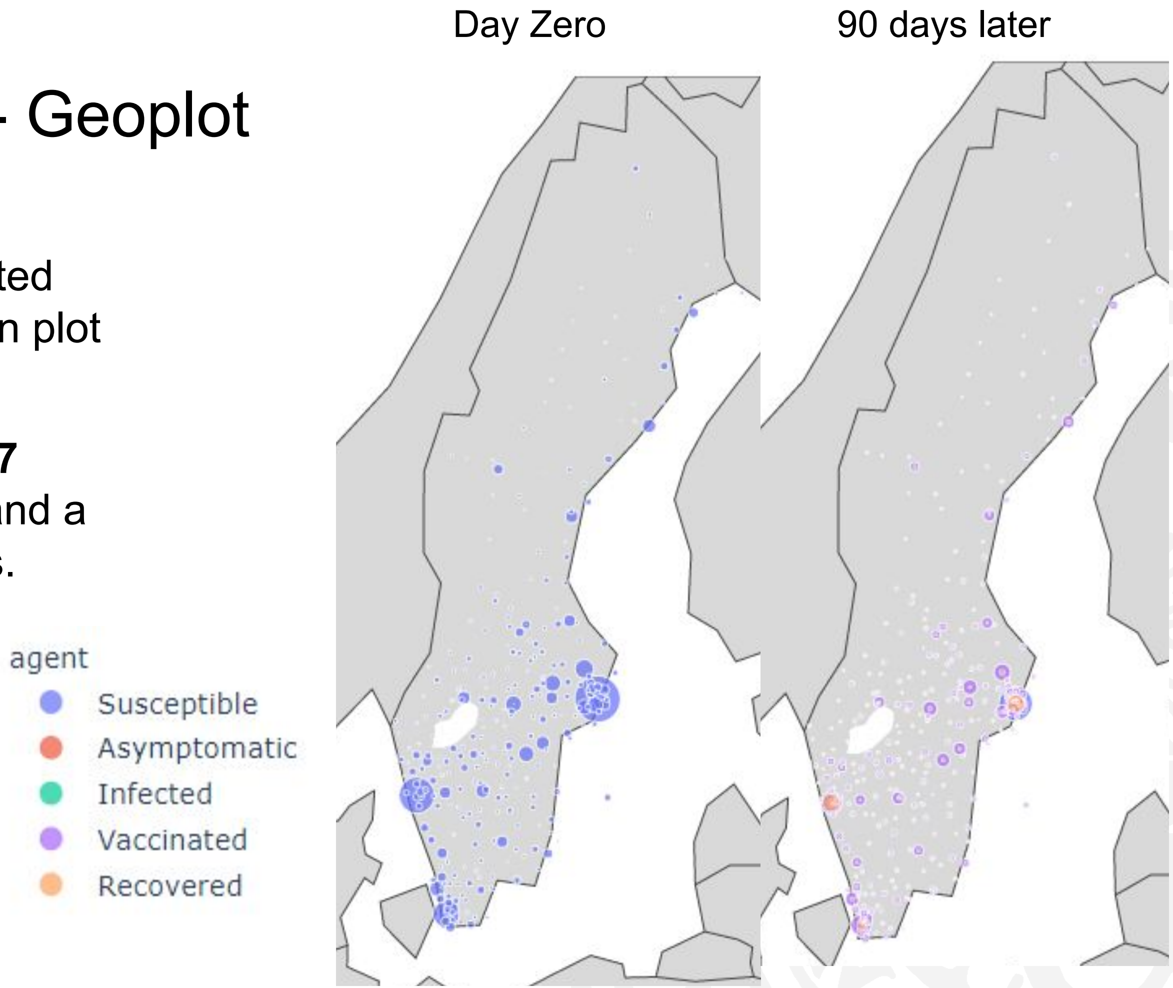


Agents can travel between connected communities by using travel weights

UPPSALA UNIVERSITET

Matrix Model - Geoplot

- By writing/reading computed data with CSV files we can plot geographical plots
- The model depicts the **297** communities of Sweden and a total of **7.6 million** agents.

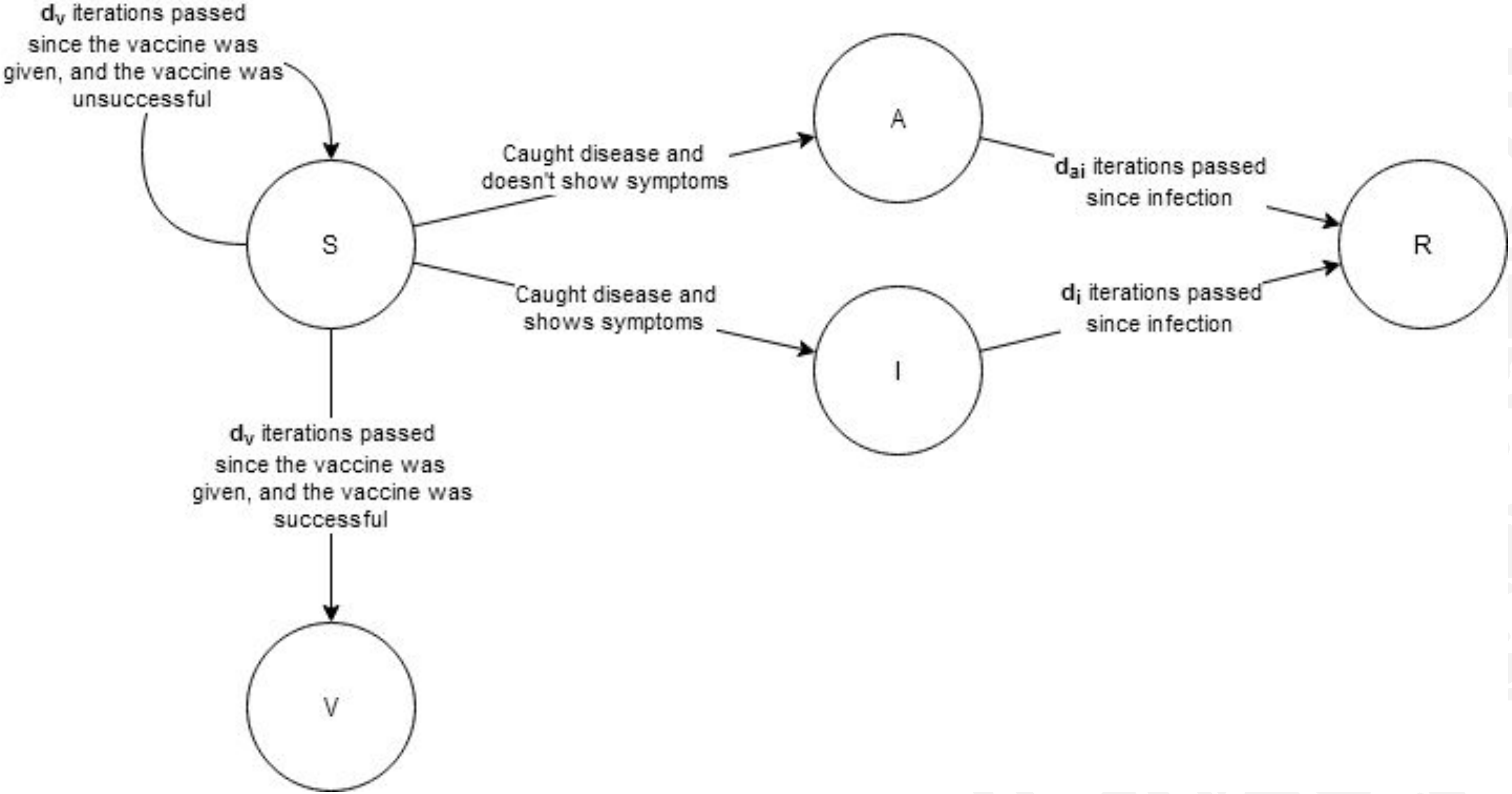

Day Zero

90 days later

Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

Agent state diagram - Network Model


d_v iterations passed since the vaccine was given, and the vaccine was unsuccessful
S
Caught disease and doesn't show symptoms
A
d_{ai} iterations passed since infection
R
Caught disease and shows symptoms
I
d_i iterations passed since infection
d_v iterations passed since the vaccine was given, and the vaccine was successful
V

State transition Parameters - Network Model

initial_infected - Initial number of randomly chosen infected
T - Number of time steps
T v - Vaccination start time
n v - Vaccines available per time step
p v - Vaccine success probability
d v - Time until vaccine success
vaccination_strategy - Strategy used for vaccinating agents

n i - Infection attempts if infected
p i - Infection probability if infected
p t - Travel probability if infected
d_i - Disease duration if infected

a_p - Probability of becoming asymptomatic

State transition Parameters - Network Model

`initial_infected` - Initial number of randomly chosen infected
`T` - Number of time steps
`T v` - Vaccination start time
`n v` - Vaccines available per time step
`p v` - Vaccine success probability
`d v` - Time until vaccine success
`vaccination_strategy` - Strategy used for vaccinating agents

`n i` - Infection attempts if infected
`p i` - Infection probability if infected
`p t` - Travel probability if infected
`d_i` - Disease duration if infected

These parameters can be set separately for asymptomatic agents

`a_p` - Probability of becoming asymptomatic

What is the Network Model?

Graphs instead of Matrices!

What are the benefits of a matrix of agents?

- Simple to construct
- Simple to dynamically change the range of infection
- Clustering behavior

What are the downsides?

- Hard to model more complex relationships

What if we use **graphs** instead?

- Node: Agent
- Edge: path of infection

Graph representations

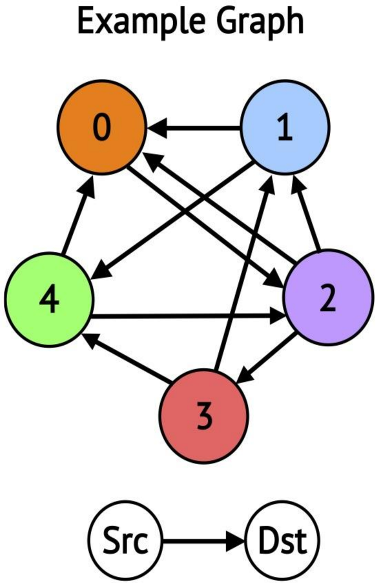
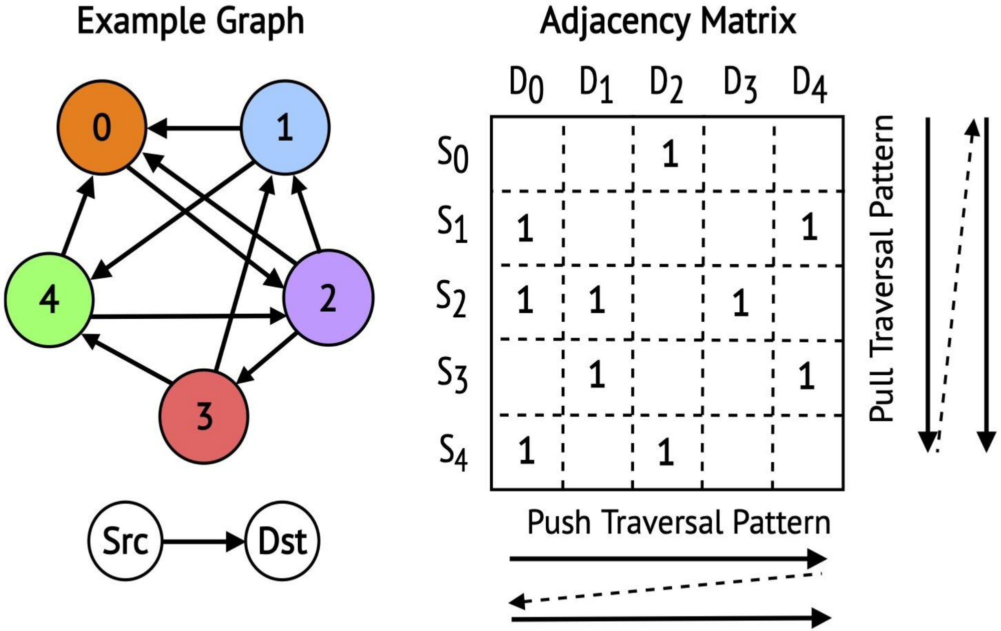


Figure adapted from [Balaji et al. HPCA'21]

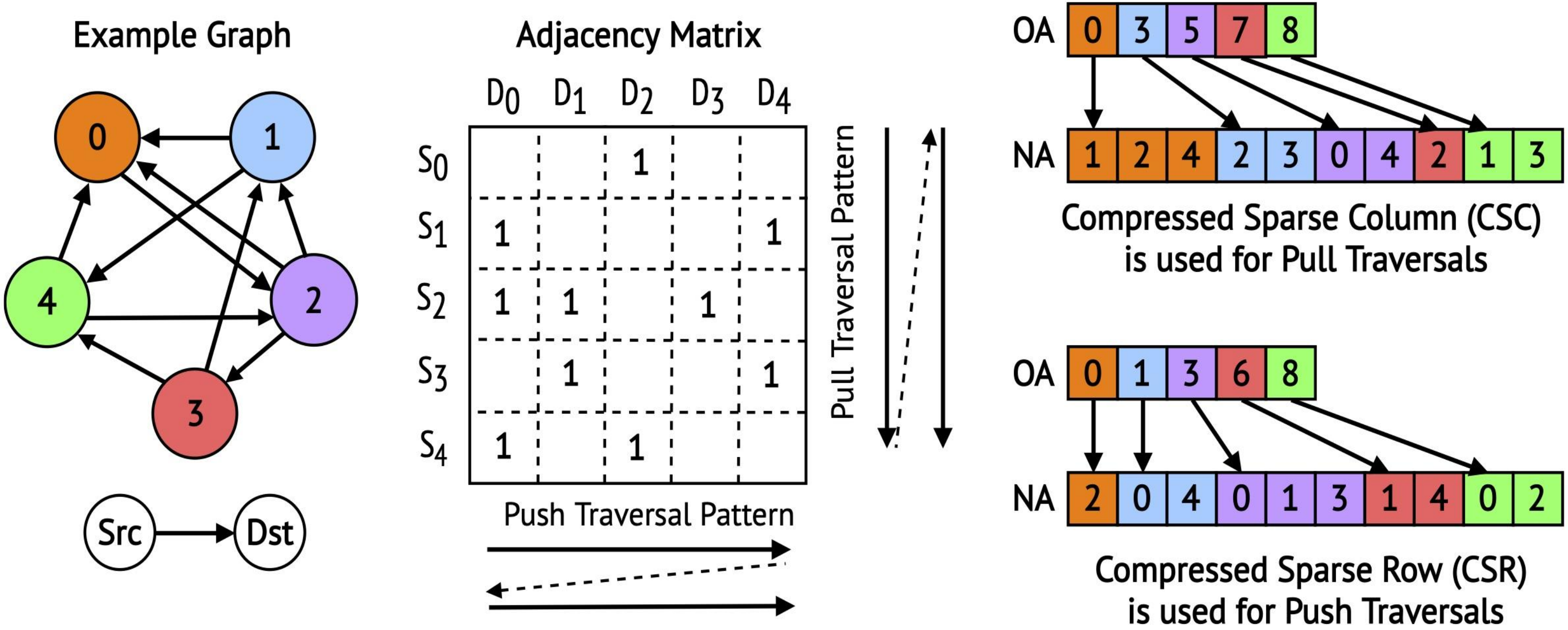
Graph representations



	D_0	D_1	D_2	D_3	D_4
S_0			1		
S_1	1				1
S_2	1	1		1	
S_3		1			1
S_4	1		1		

Figure adapted from [Balaji et al. HPCA’21]

Graph representations



Adjacency Matrix

	D_0	D_1	D_2	D_3	D_4
S_0			1		
S_1	1				1
S_2	1	1		1	
S_3		1			1
S_4	1		1		

OA	0	3	5	7	8					
NA	1	2	4	2	3	0	4	2	1	3

Compressed Sparse Column (CSC) is used for Pull Traversals

OA	0	1	3	6	8					
NA	2	0	4	0	1	3	1	4	0	2

Compressed Sparse Row (CSR) is used for Push Traversals

Figure adapted from [Balaji et al. HPCA’21]

“Random Network”, equal number of connections per agent.

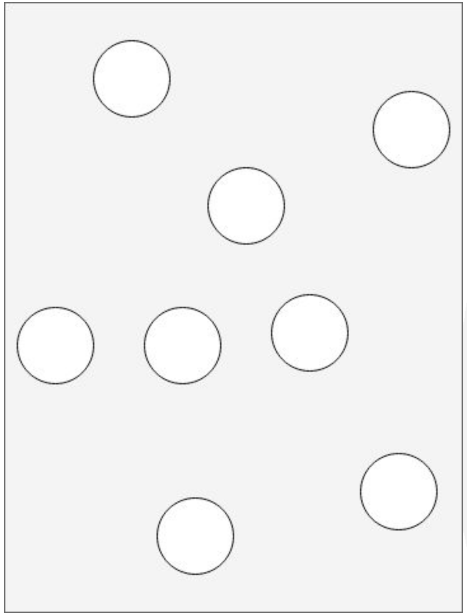
Example: population size = 8 ; each agent has 3 connections/neighbors.

All connections bidirectional
⇒ complicates the algorithm

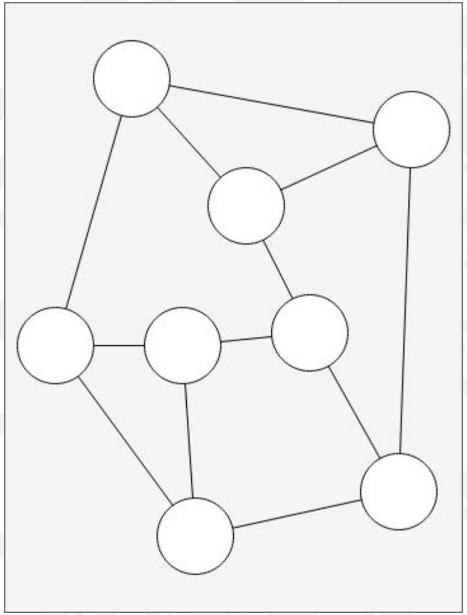
Time complexity:
$\Theta(N \times N_0)$

No clustering behavior:
– my neighbors have no connections to one another
– very unrealistic
Results in very fast transmission in simulations.

1: Create **N** = 8 agents/nodes



2: Randomly add edges to get degree $\mathbf{N_0}$ = 3

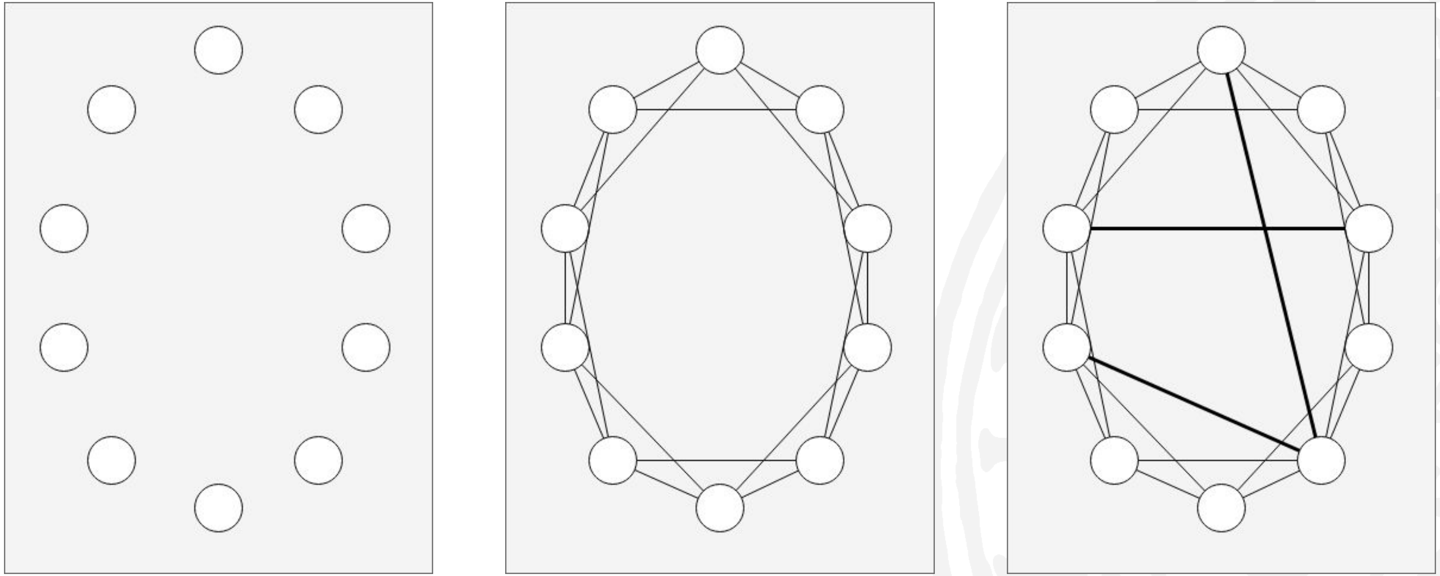


Newman-Watts small world graph

1: Create **N** = 10 nodes

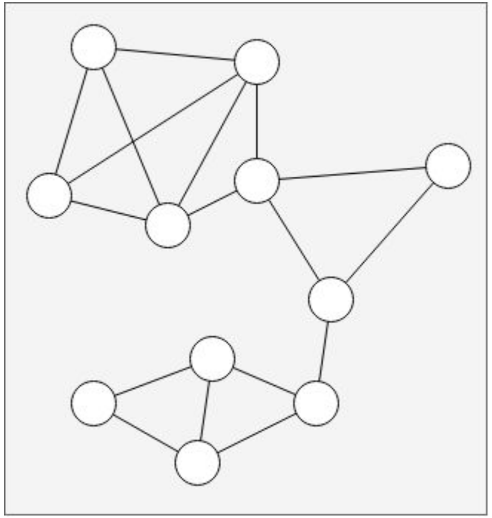
2: Connect each node to **k** = 2 neighbors on each side

3: Create random shortcuts with probability **p**



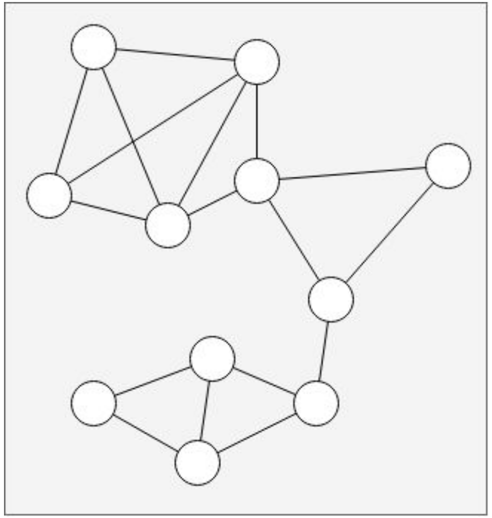
Time complexity:
$\Omega(Nk)$
$O(N^2)$

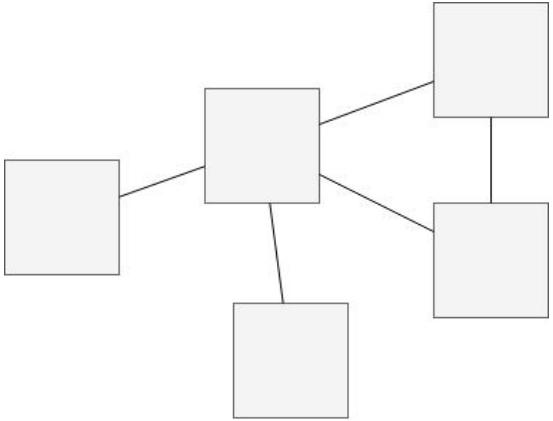
Communities - Network Model



Each community contains a graph

Communities - Network Model

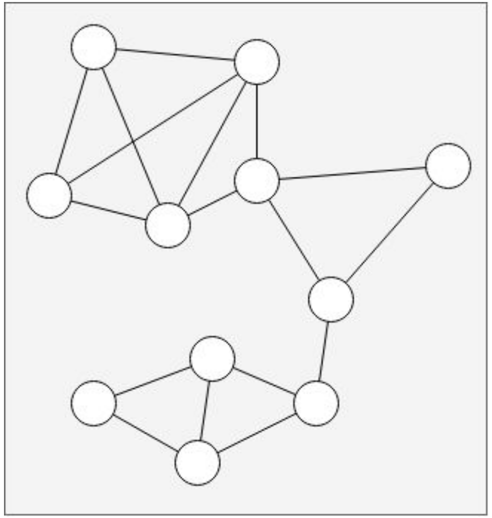




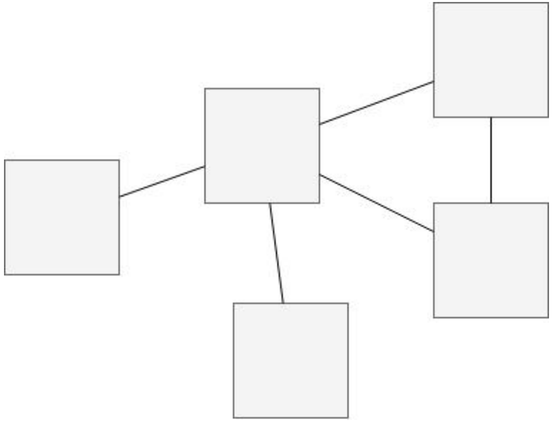
Each community contains a graph

Communities are manually connected to each other in a larger graph

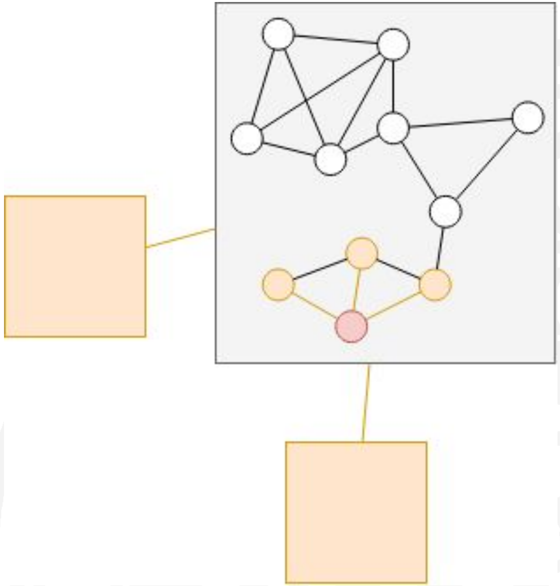
Communities - Network Model



Each community contains a graph



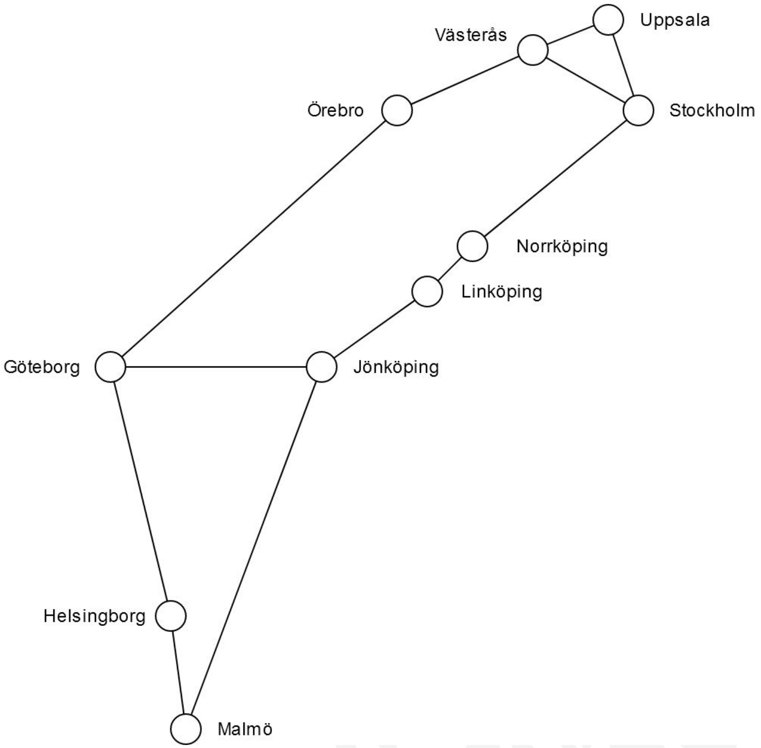
Communities are manually connected to each other in a larger graph



Based on the probability **p_t**, an infected agent either tries to infect a neighbor in their own community or a random agent in a neighboring community.

Community graph - Network Model

- 10 largest municipalities in Sweden, population of each scaled to 1/3
- Regions connected based roughly on proximity and train lines
- Total population of 1 million

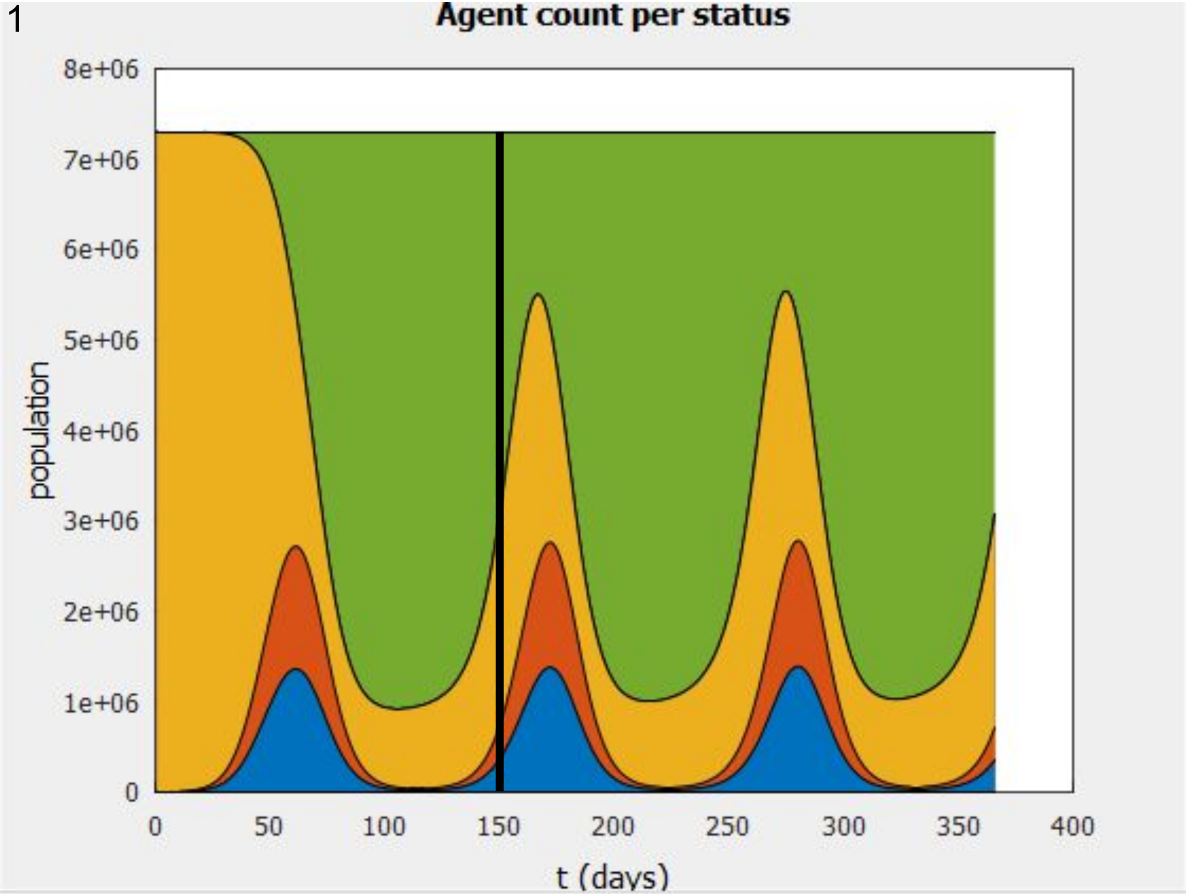


Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: asymptomatic rate, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

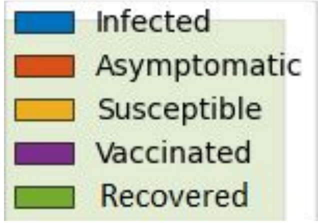
Matrix Model: Quarantine

🟦 Infected
🟧 Asymptomatic
🟨 Susceptible
🟪 Vaccinated
🟩 Recovered



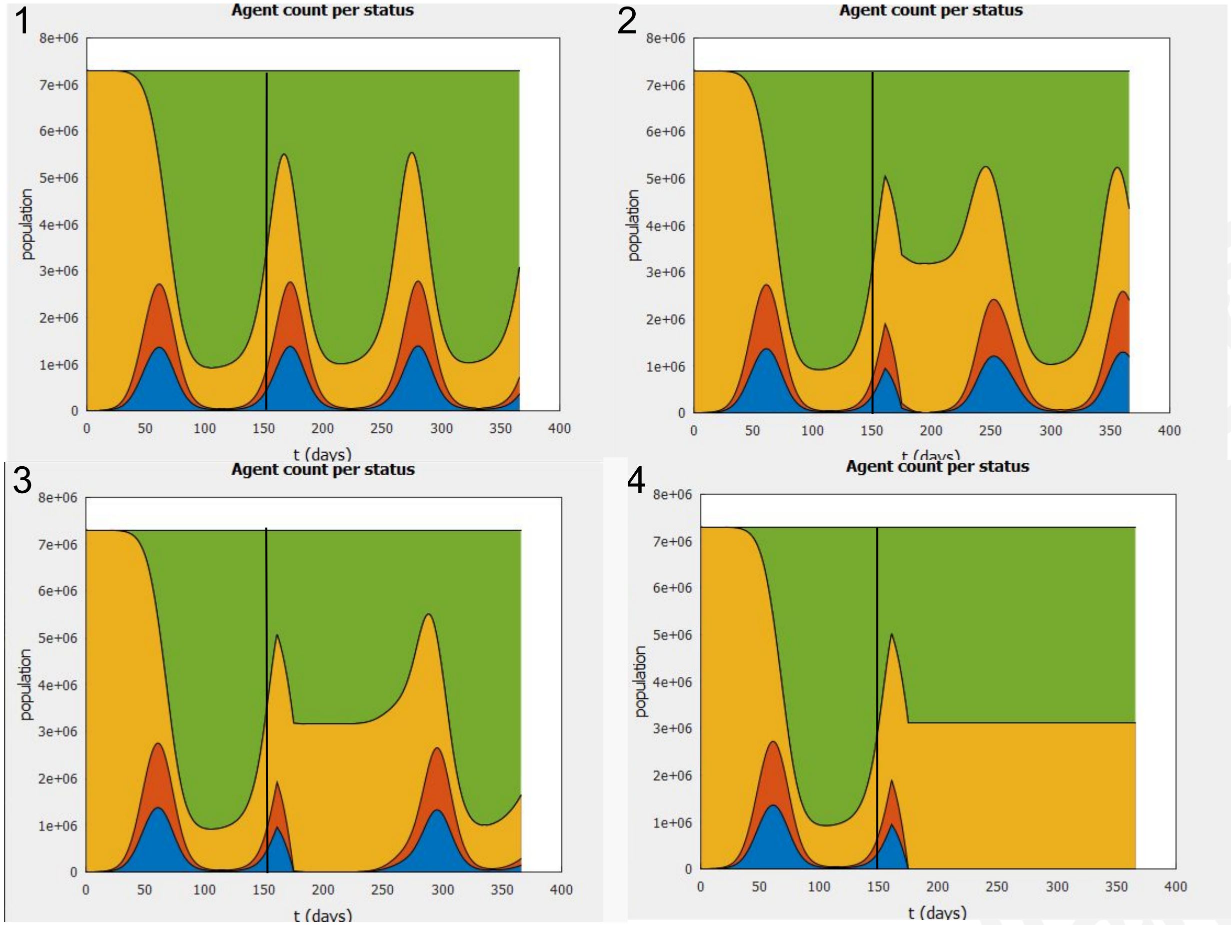
0%
efficacy

Matrix Model: Quarantine



0% efficacy

90% efficacy

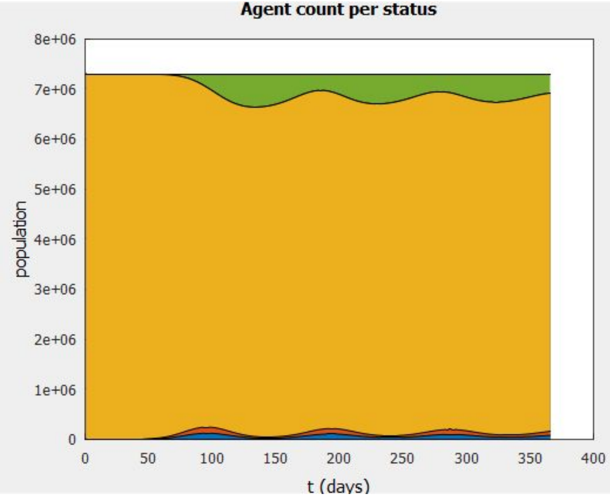


99% efficacy

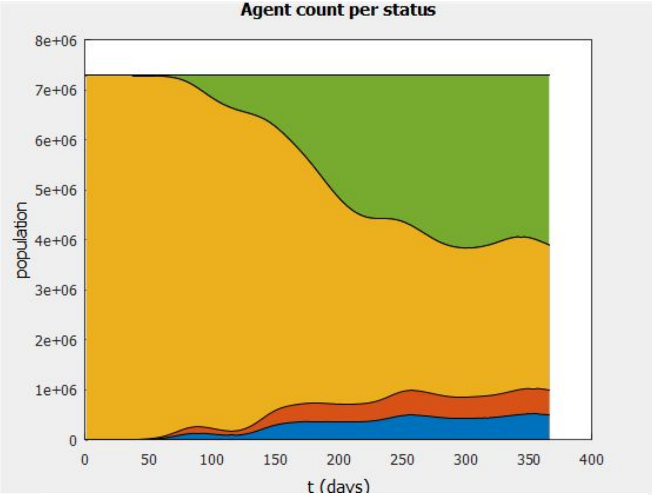
100% efficacy

Matrix Model - Effect of travel intensity

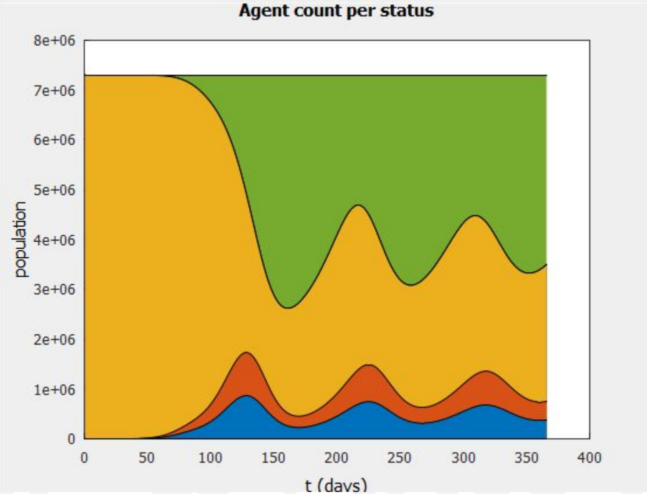
🟦 Infected
🟧 Asymptomatic
🟨 Susceptible
🟪 Vaccinated
🟩 Recovered



1: Swap:0



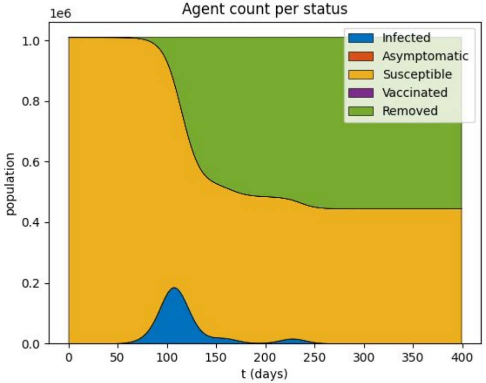
2: Swap: 100

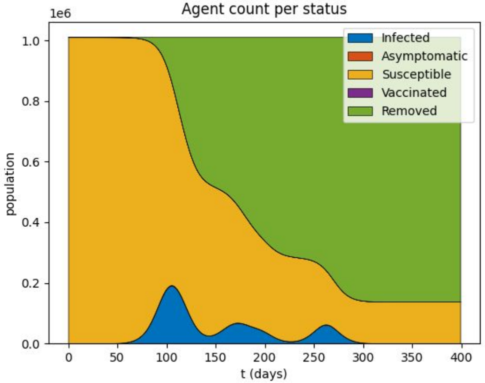


3: Swap: 1000

Network Model - Effect of asymptomatic rate

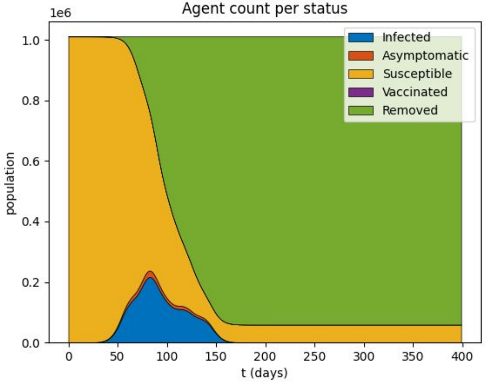
0% asymptomatic

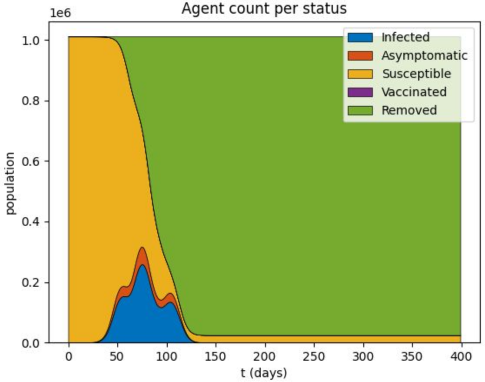




1% asymptomatic

10% asymptomatic

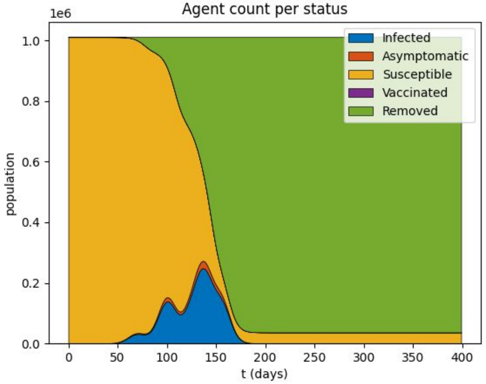


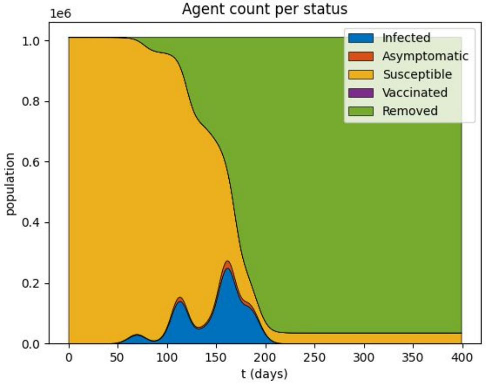


20% asymptomatic

Network Model - effect of **asymptomatic** travel intensity

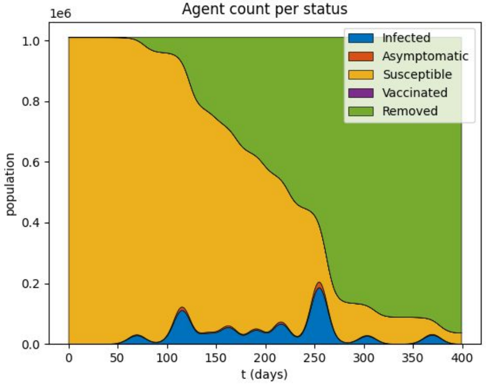
0.5% travel probability

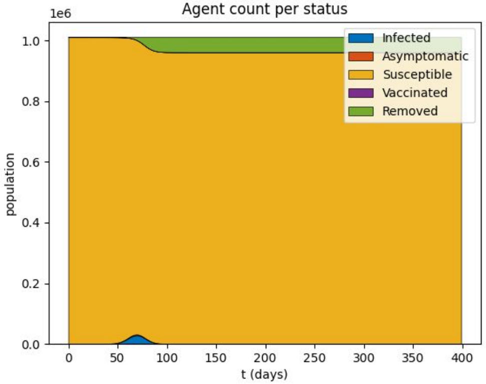




0.1% travel probability

0.01% travel probability





0.001% travel probability

Network Model variance

The Network Model more sensitive to randomization:

Depending on random seeds:

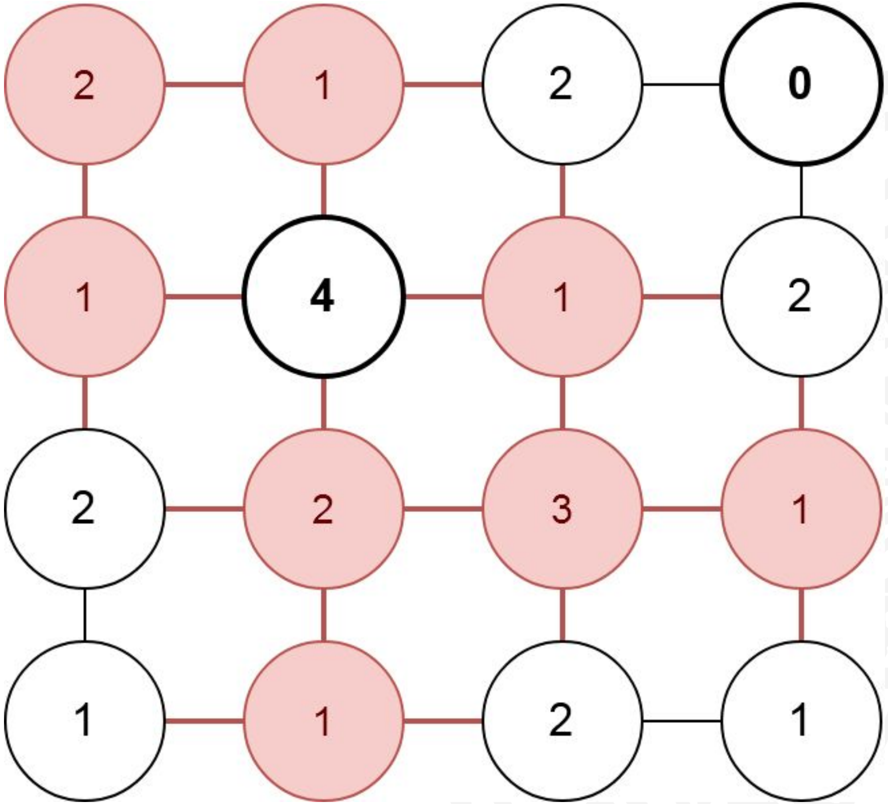
- Initial (random) placement of infected:
  - up to +/- 8%
- Travel
  - up to +/- 5%
- In large part: artifact of the small number of communities modelled (10)
- Results should be based on multiple runs and shown with confidence intervals

Road map

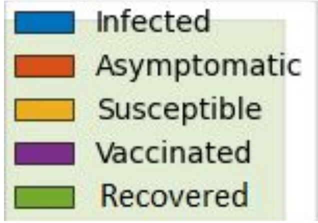
- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

Vaccinations strategies

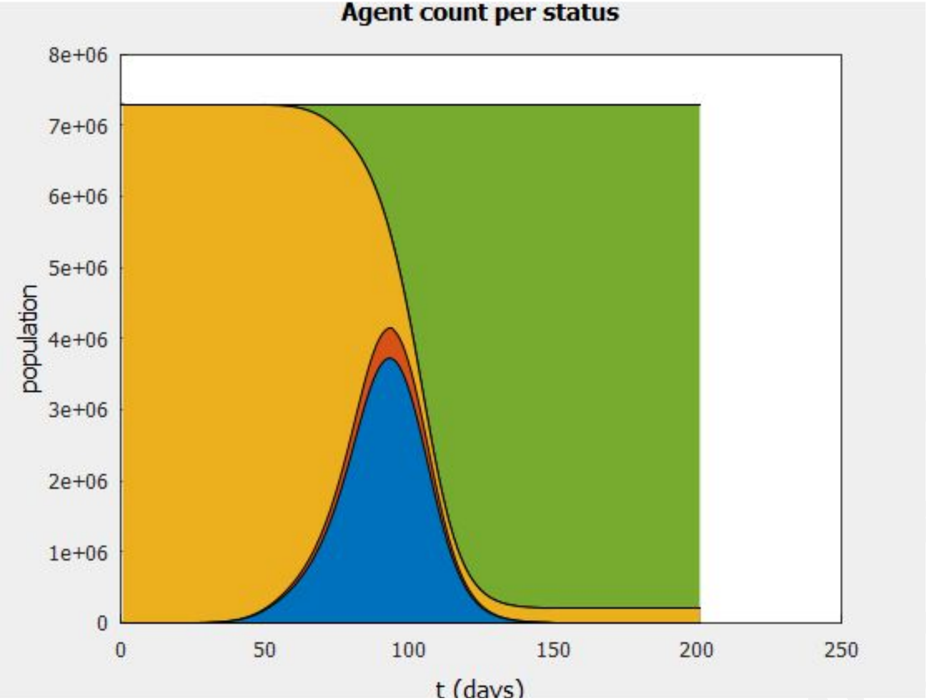
- Uniform distribution
- High Density
- Low Density



Matrix Model - Vaccination

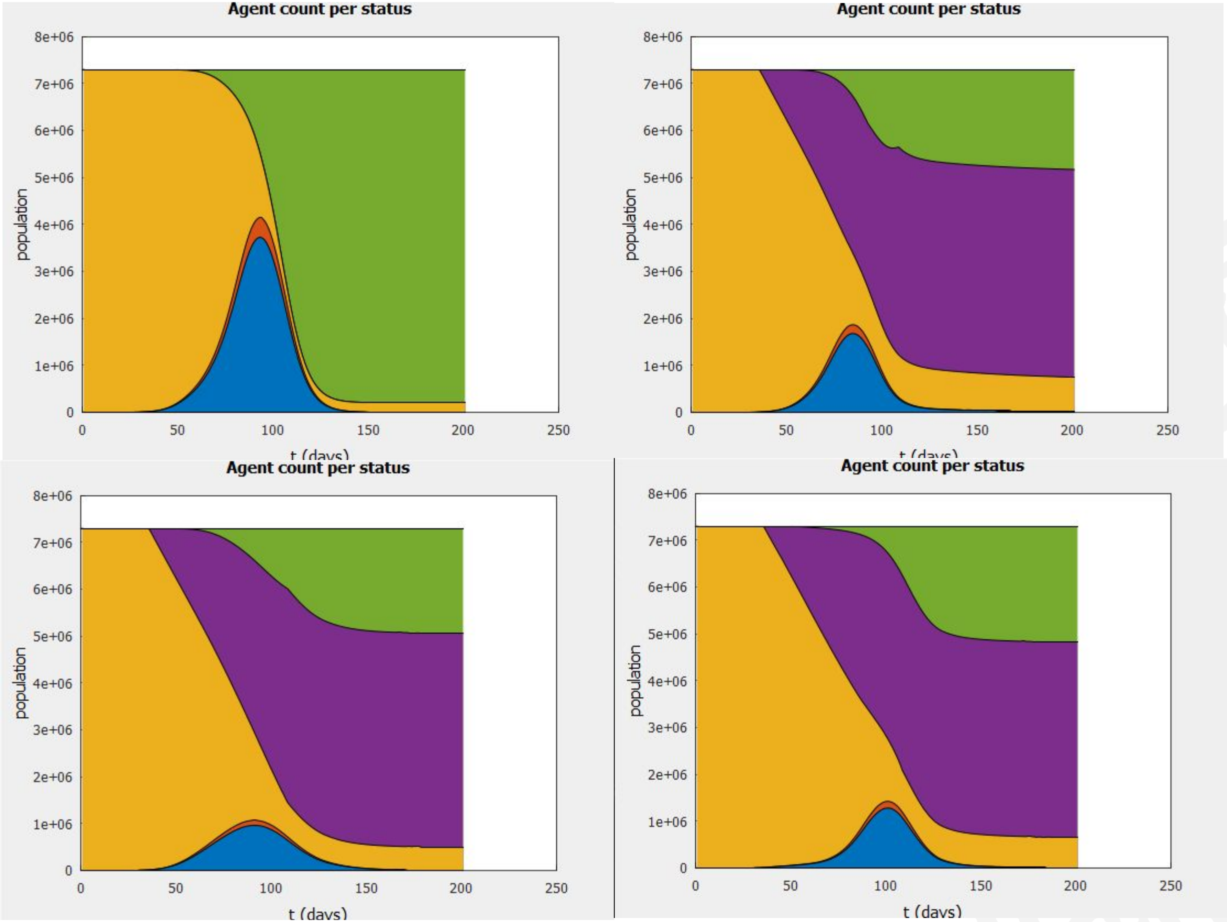


No vaccination



Matrix Model - Vaccination

🟦 Infected
🟧 Asymptomatic
🟨 Susceptible
🟪 Vaccinated
🟩 Recovered



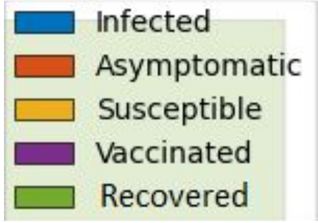
No vaccination

Low Density

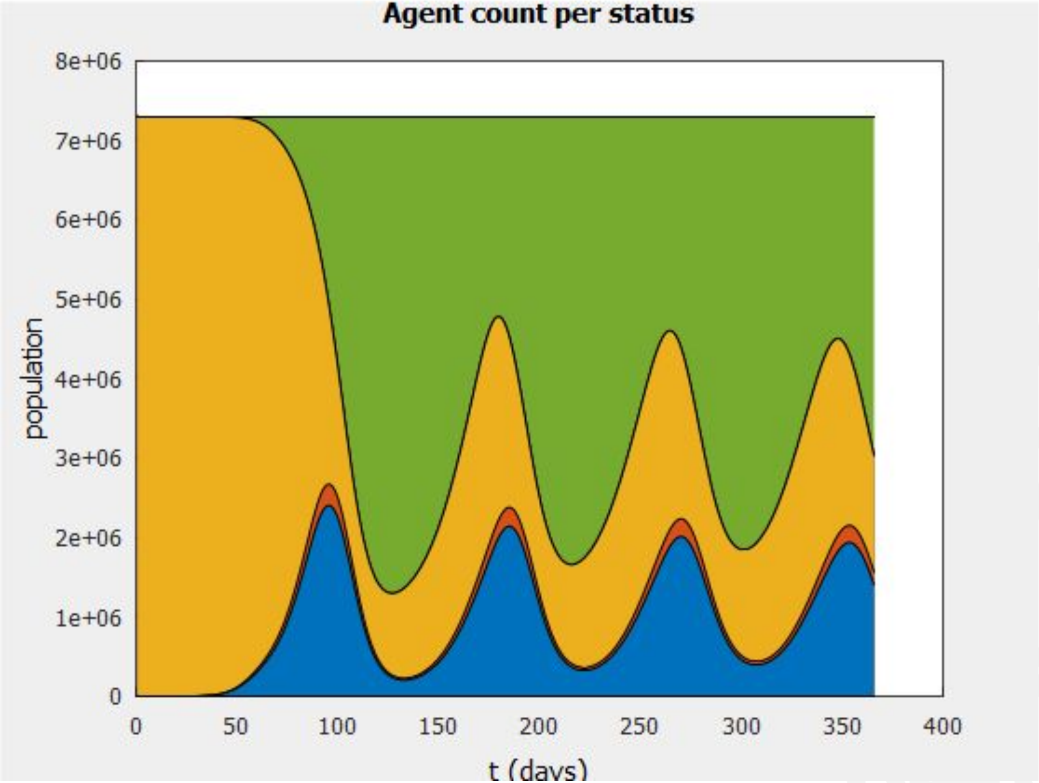
Uniform

High Density

Matrix Model - Vaccination with re-infection



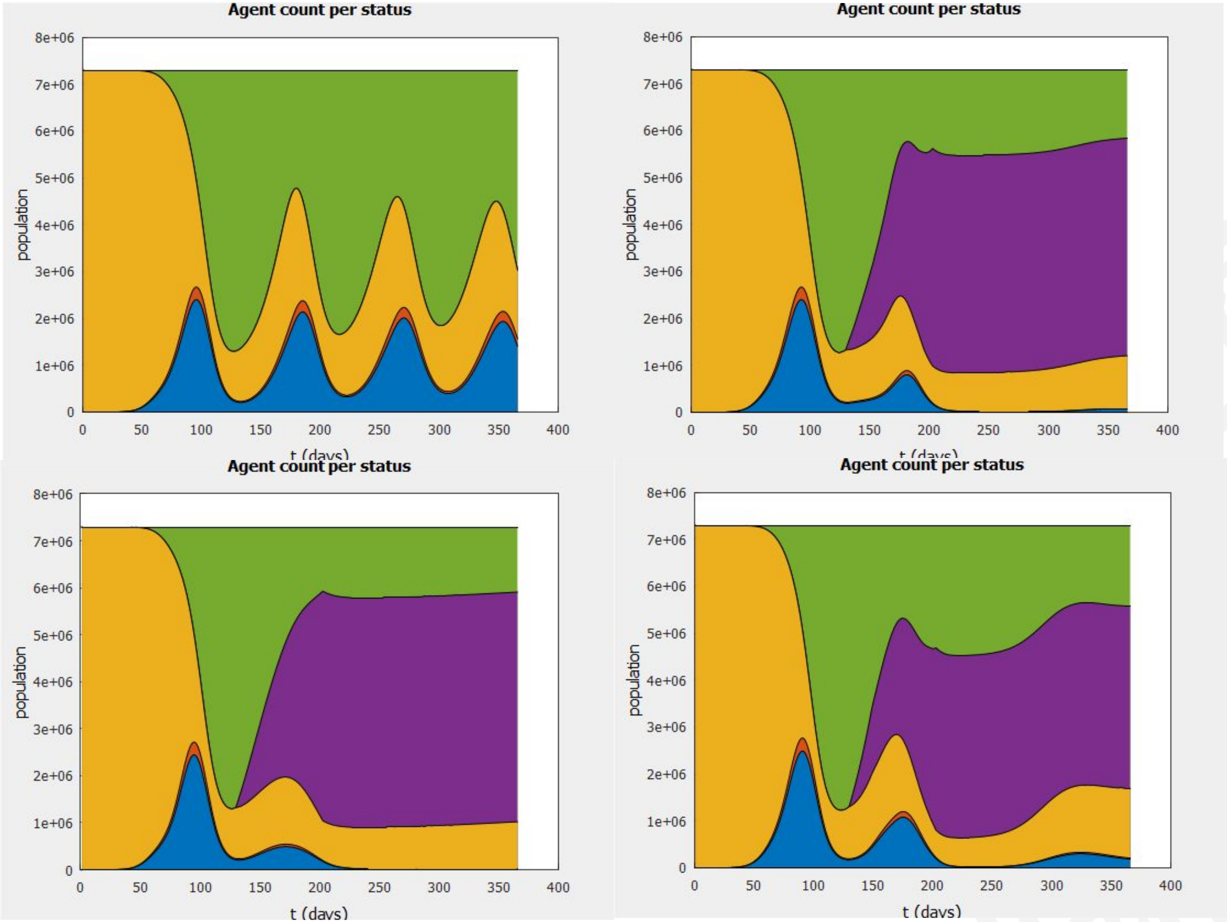
No
vaccination



Matrix Model - Vaccination with re-infection

🟦 Infected
🟧 Asymptomatic
🟨 Susceptible
🟪 Vaccinated
🟩 Recovered

No vaccination



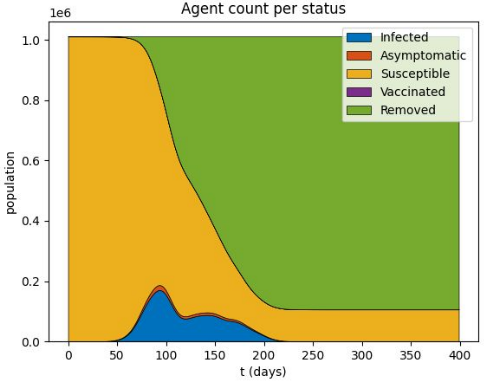
Low Density

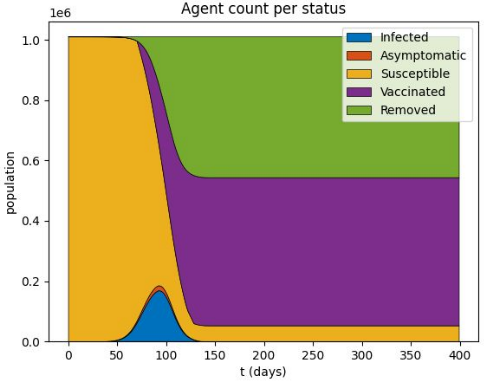
Uniform

High Density

Network Model - vaccination strategies, Newman-Watts

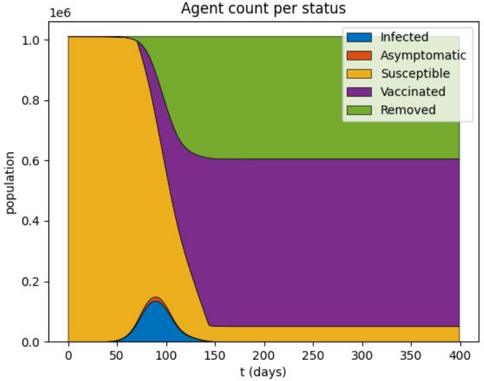
No vaccination

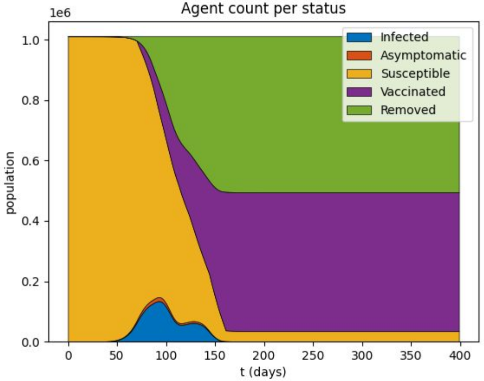




Low density

Uniform



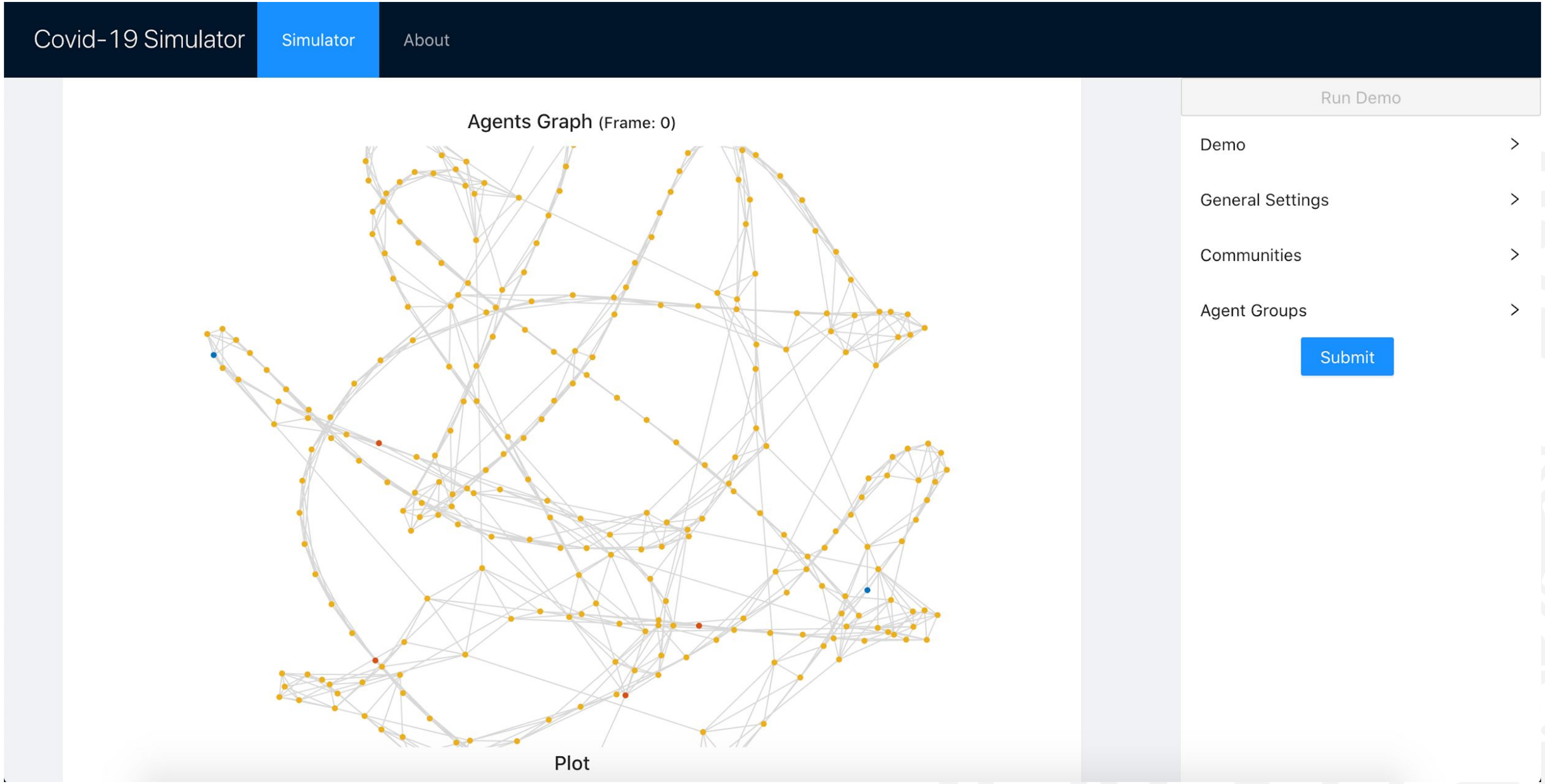


High density

Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
  - Matrix Model: quarantine, travel
  - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

Web User Interface



- ReactJS
- Graph represented by force-directed graph of D3 library, and using ChartJS for plotting

Demo

Web User Interface

Road map

- The basics: Agent based modeling and states
- The Matrix Model
- The Network Model
- Results!
    - Matrix Model: quarantine, travel
    - Network Model: initial infections, asymptomatic, travel
- Both models: vaccination strategies
- Web User Interface
- Some Final Remarks

A few Computer Science aspects

- Data Structures
  - Matrix Model is designed to be cache-friendly: spatial locality
    - Structures of Arrays vs Array of Structures
  - Network Model: Compressed Sparse Row format efficient but not cache-friendly

A few Computer Science aspects

- Data Structures
  - Matrix Model is designed to be cache-friendly: spatial locality
    - Structures of Arrays vs Array of Structures
  - Network Model: Compressed Sparse Row format efficient but not cache-friendly
- Algorithms
  - Time complexity for simulations: *O(nAgents)*
    - Creating Newman-Watts graphs is *O(nAgents^2)*
  - Tradeoff: runtime vs random accuracy

A few Computer Science aspects

- Data Structures
  - Matrix Model is designed to be cache-friendly: spatial locality
    - Structures of Arrays vs Array of Structures
  - Network Model: Compressed Sparse Row format efficient but not cache-friendly
- Algorithms
  - Time complexity for simulations: *O(nAgents)*
    - Creating Newman-Watts graphs is *O(nAgents^2)*
  - Tradeoff: runtime vs random accuracy
- Parallelization
  - Matrix Model is already parallelized with OpenMP
    - Lock-free algorithm
    - ~1.9x speedup on 6 cores
  - Network Model: considering parallelization strategies

Difference between solutions

	Speed/Efficiency	Agent connections	Vaccination strategy	Quarantine support
Matrix Model	Faster - Better spatial locality	Uniform structure	S + R can be vaccinated	Yes
Network Model	Resource intensive	More flexible structure	Only S can be vaccinated	No

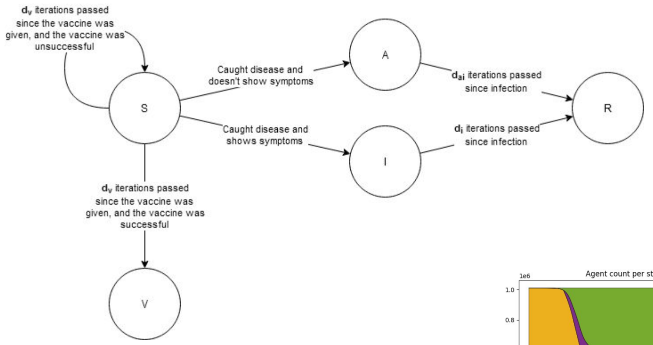
Summary

- Two different implementations of the same problem
- Highly modular/flexible, lots of parameters allowing us to create multiple different scenarios
- Test out different vaccinations strategies
- Matrix solution is optimized to be run on high performance computer, 7.6 million agents simulation can be run in approx. one and a half minute on a desktop computer
- Network solution utilizes compressed sparse row for efficiently fitting very large matrices in memory, around 0.05% size in demos shown.

Follow-up work

- Connect solutions with the Web User Interface
- Parallelize with pthreads and MPI
  (OpenMP parallelization already available for Matrix)
- Test optimization strategies on HPC system
- Additional behavior:
  - Multiple quarantines
  - Seasonal characteristics/variables/virus mutations
    - Mainly expressed as a change in the infection probability starting (and/or ending) at specific simulation times
    - Possibility to model multiple virus strains simultaneously
- More complex infection model for Network implementation

Overview


d_v iterations passed since the vaccine was given, and the vaccine was unsuccessful

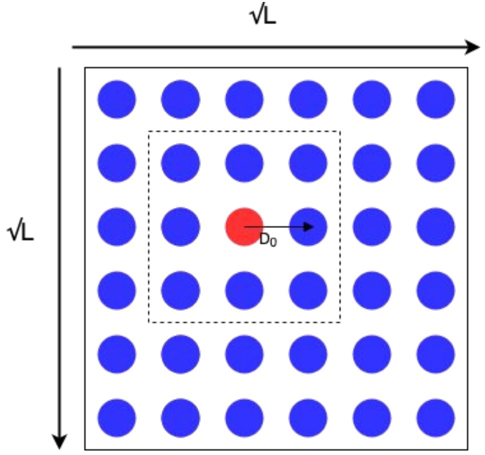
Caught disease and doesn't show symptoms

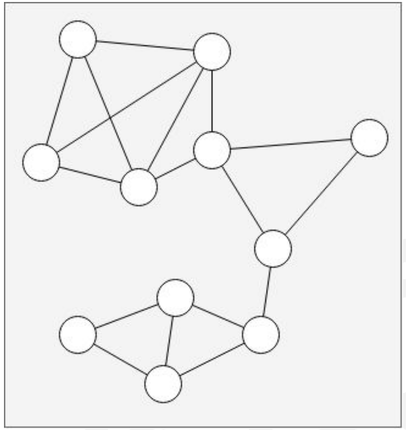
Caught disease and shows symptoms

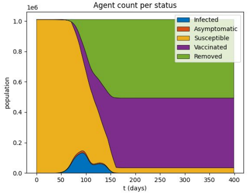
d_{ai} iterations passed since infection

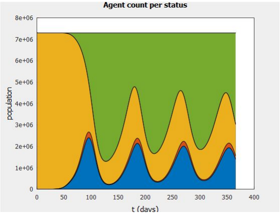
d_i iterations passed since infection

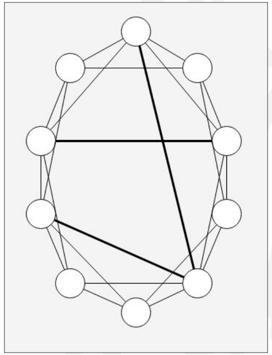
d_v iterations passed since the vaccine was given, and the vaccine was successful

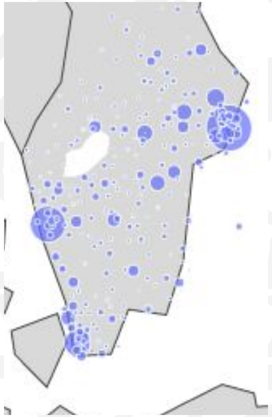


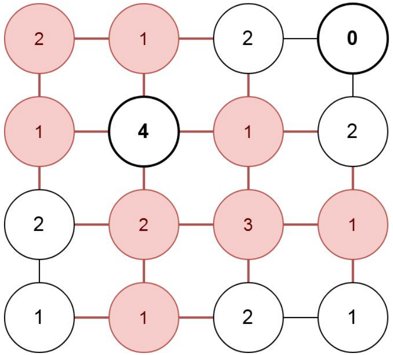


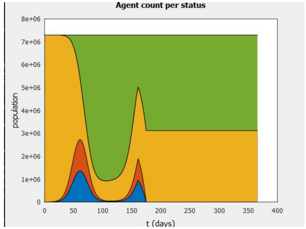

Agent count per status

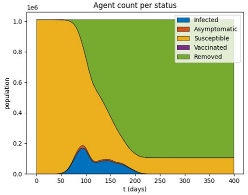

Agent count per status








Agent count per status


Agent count per status

http://www.it.uu.se/research/group/uart/projects/saivr

Follow-up work

- Connect solutions with the Web User Interface
- Parallelize with pthreads and MPI (OpenMP parallelization already available for Matrix)
- Performance comparisons
- Test optimization strategies on HPC system
- Additional behavior:
    - Additional Vaccination strategies
    - Multiple quarantines
    - Seasonal characteristics/variables/virus mutations
        - Mainly expressed as a change in the infection probability starting (and/or ending) at specific simulation times
        - Possibility to model multiple virus strains simultaneously
- More complex infection model for Network implementation

Newman-Watts small world graphs

Option 2: Newman-Watts small world

- Create a regular 1D lattice of **N** nodes with edges connecting each node to the **k** nearest neighbors on each side
- Iterate through each pair of non-connected nodes, add an edge with probability **p**

Advantages:

- Short path between any two nodes possible despite small degree
- Clustering behavior with the nodes from the regular lattice

Disadvantages:

- Difficult to dynamically change connections like in the matrix model

Wang, Xiao Fan, and Guanrong Chen. "Complex networks: small-world, scale-free and beyond." *IEEE circuits and systems magazine* 3.1 (2003): 6-20.

Newman, Mark EJ, and Duncan J. Watts. "Renormalization group analysis of the small-world network model." *Physics Letters A* 263.4-6 (1999): 341-346.

Web User Interface

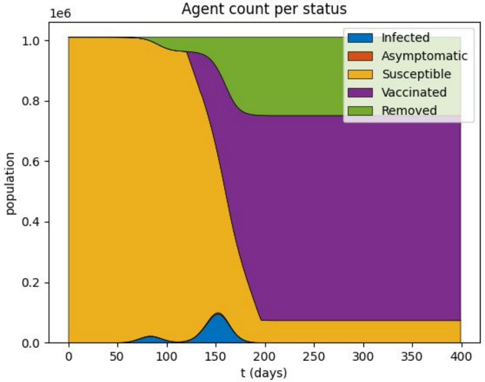
Drogon

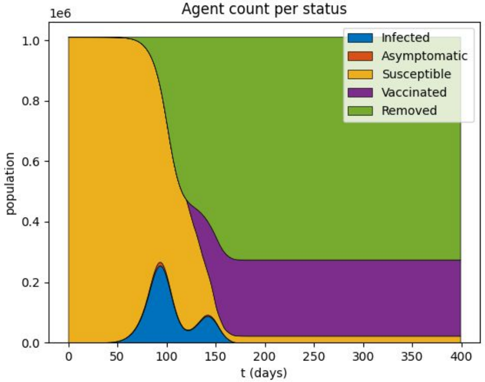
web application
framework with c++14/17

- High performance web application platforms
- cross-platform
- Support JSON format request and response

Network Model - Effect of number of initial infections

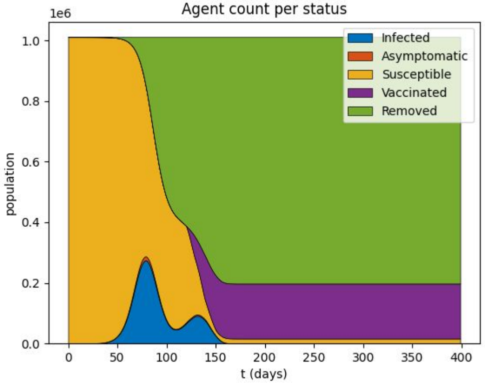
1 initial

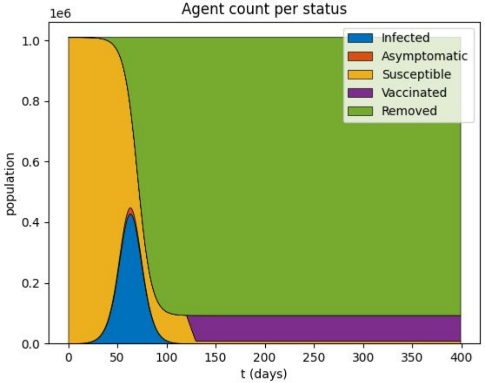




5 initial

20 initial

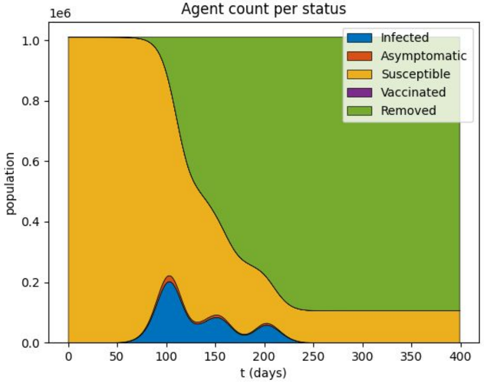


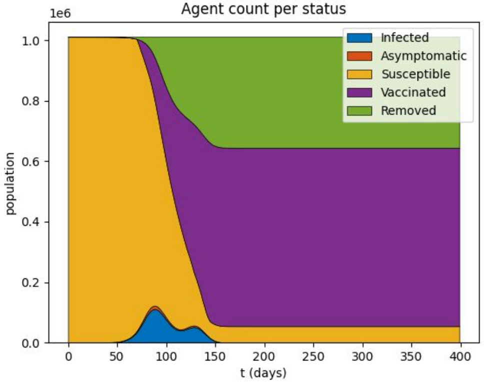


200 initial

Network Model - effect of vaccination strategy (random graphs)

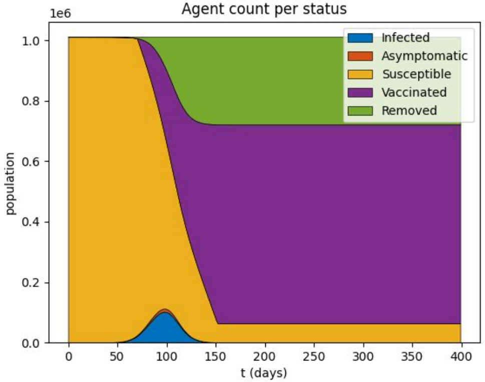
No vaccination

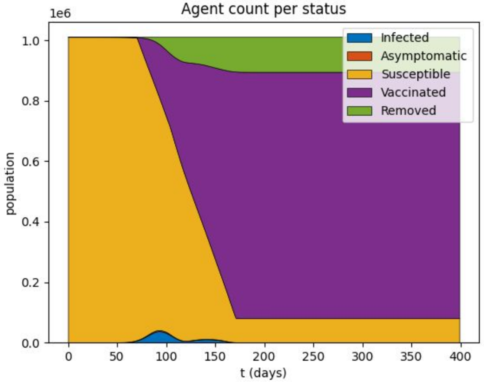




Low density

Uniform



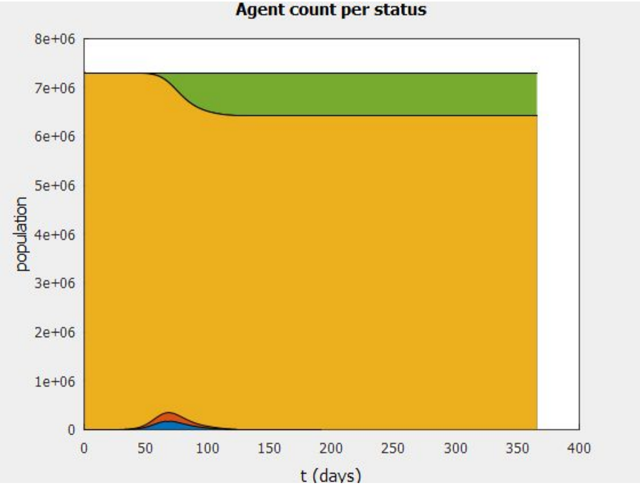


High density

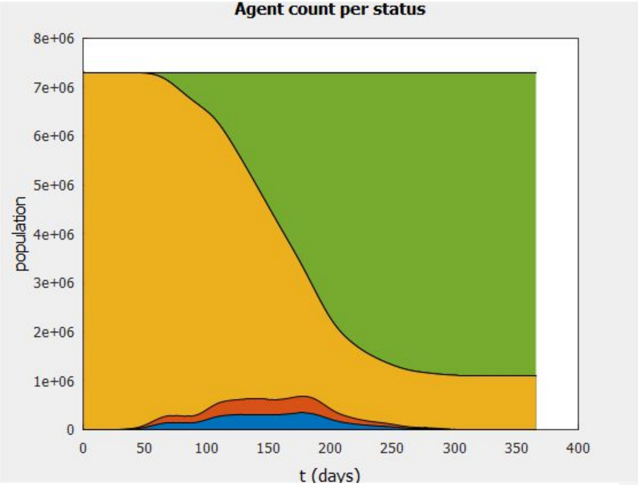
UPPSALA
UNIVERSITET

Matrix Model - Effect of travel intensity, without re-infection

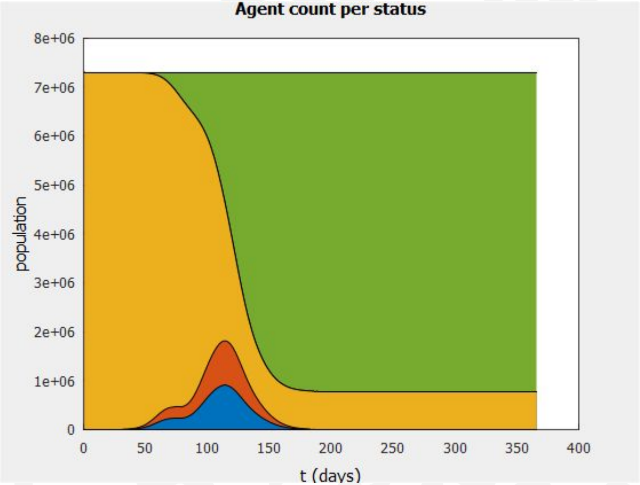
🟦 Infected
🟥 Asymptomatic
🟨 Susceptible
🟪 Vaccinated
🟩 Recovered



1: Swap: 0



2: Swap: 100



3: Swap: 1000