Epidemiological Agent-based Model Optimization

DS-GA 1019, Spring 2023

Group 11: Xu Han, Chenxi Ning, Jason Wang, Philip Xing, Chloe Zheng

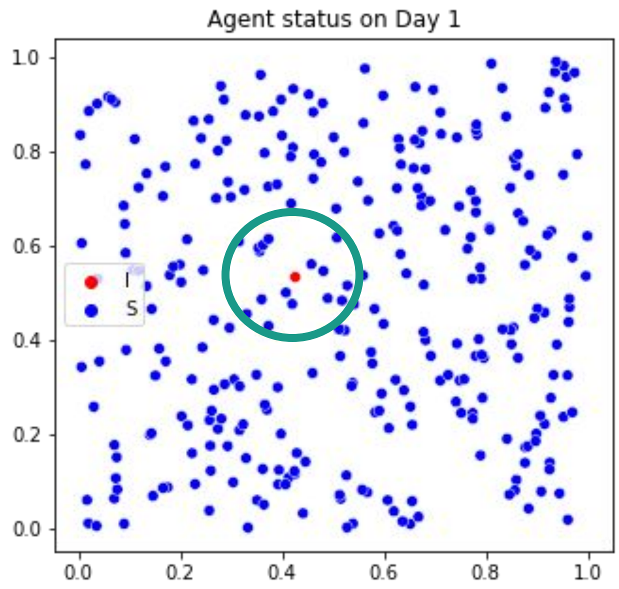
Background

Motivation: Given the COVID-19 pandemic and the public discourse on vaccinations, our group was interested in a stochastic model for visual predictions of susceptible, infected, and recovered populations over the course of a simulated epidemic.

- We first implemented an agent-based model (ABM) that can simulate how an epidemic may unfold;
- We also optimized model runtime and compared and combined various optimization strategies

Agent-based Model

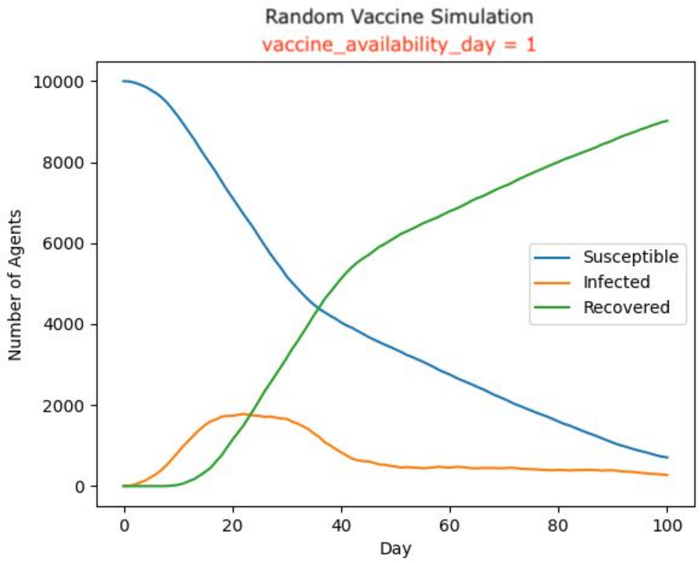
- Status (S, I, R)
  - Susceptible → Infected
    - Within a certain distance from an infected agent
    - Pass a threshold of infection probability
  - Infected → Recovered
    - Have been infected for a certain number of days
    - Pass a threshold of recovery probability
- Location
  - Assign each agent a random location in a unit square
  - Update each agent's location at each timestamp
    - Snap to the edge if the new location is out of bounds
- Days with status
- Vaccine attributes



Research Question

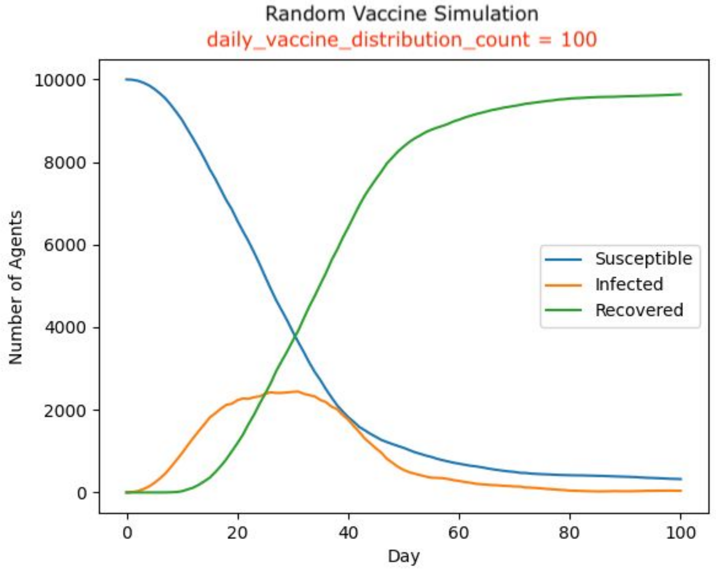
- How do different vaccination roll out methods affect a pandemic?
  - Vary vaccination start time, distribution rate, and targeting specific populations first
- Varying simulations and parameters through CLI
  - Basic Sim: Duration (*100*), Number of agents (*10000*), Infection distance (*0.03*), Infection probability (*0.3*), Minimum infection duration (*7*), Recovery probability (*0.3*)
  - Random Vaccine Sim: Vaccine availability day ***(start time)***, Daily vaccine distribution count ***(speed)***, Initial vaccine efficacy (*0.95*), Vaccinated recovery reduction (*2*)
    Targeted Vaccine Sim: Immunodeficient proportion and complete rollout day ***(target populations)***, Immunodeficient infection probability increase (*0.3*)

Varying vaccine start date



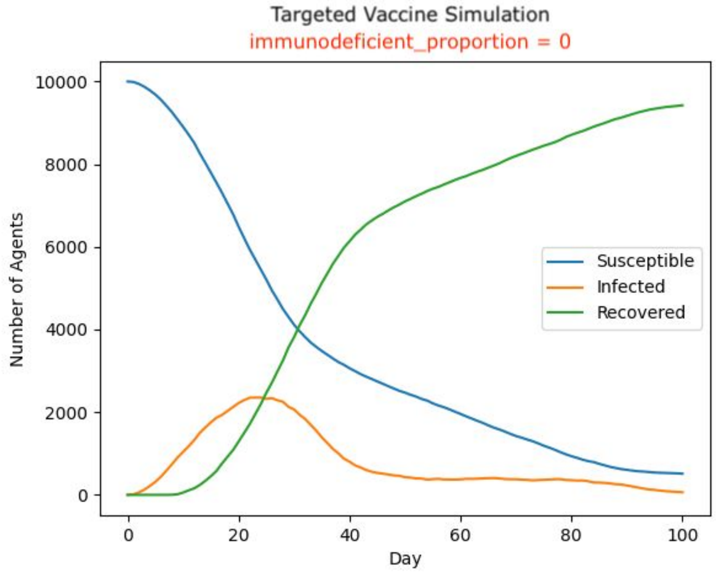
- The later the vaccine is available, the more pronounced the infected population curve

Varying daily vaccine distribution count



- The faster we distribute the vaccine, the more we flatten the infected population curve

Varying the immunodeficient population proportion



- Targeting the immunodeficient population first with the existing configuration (availability after the 10th day, 0.3 infection probability, etc.) does little to change the infection population curve overall
- Hypothesis: Given roughly 10% of the population is infected after 10 days, targeted vaccination doesn't change the overall population curves much

Optimization

Diagnosed Efficiency with Line Profiler

- Showed that **98%** of execution time was due to the function **infect()** for original code
- Infect function - nested for loops

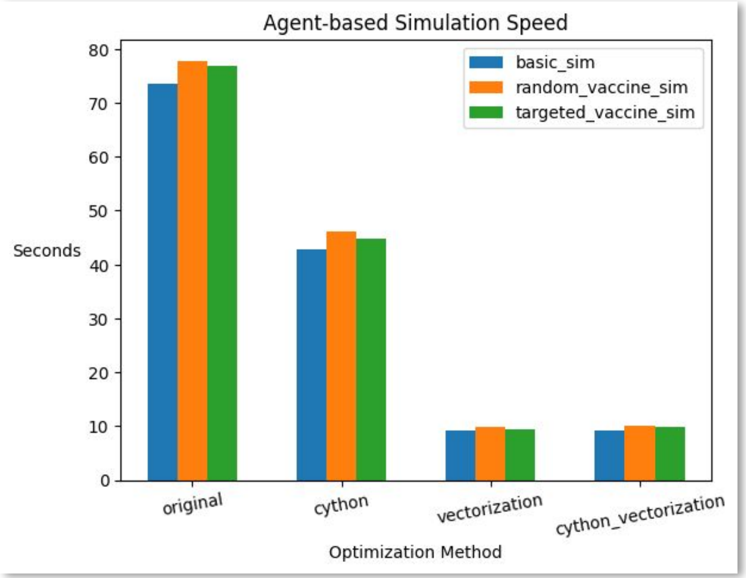
```
Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     9                                           def main(duration, num_agents, infection_dist
recovery_probability, profile=False):
    10                                               # Initialize random seed
    11         1         16.0     16.0      0.0      random.seed(42)
    12
    13                                               # Initialize the list of agents
    14         1      12315.0  12315.0      0.0      agents = [Agent("S", (random.random(), ra
    15
    16                                               # Set one agent as patient zero
    17         1          2.0      2.0      0.0      agents[0].status = "I"
    18
    19                                               # Initialize status counts
    20         1          2.0      2.0      0.0      status_counts = {"S": [], "I": [], "R": [
    21
    22
    23                                               # Run simulation for given duration
    24       101         89.0      0.9      0.0      for _ in range(duration):
    25                                                   # Update status counts for current da
    26       400        276.0      0.7      0.0          for status in ["S", "I", "R"]:
    27       300     326155.0   1087.2      0.1              count = sum(1 for agent in agents
    28       300        397.0      1.3      0.0              status_counts[status].append(coun
    29
    30                                                   # Update agent days with status and l
    31   1000100     463364.0      0.5      0.2          for agent in agents:
    32   1000000     770469.0      0.8      0.3              agent.increase_days_with_status()
    33   1000000     464459.0      0.5      0.2              max_distance = 0.01
    34   1000000    1824356.0      1.8      0.6              new_location = generate_random_lo
    35   1000000    1019553.0      1.0      0.4              new_location = snap_to_edge(new_l
    36   1000000     505493.0      0.5      0.2              agent.location = new_location
    37
    38                                                   # Infect agents
    39       100  276962804.0 2769628.0     98.0          infect(agents, infection_distance, i
    40
    41                                                   # Recover agents
    42       100     177957.0   1779.6      0.1          recover(agents, minimum_infection_dur
    43
    44                                               # Add final day status counts
    45         4          3.0      0.8      0.0      for status in ["S", "I", "R"]:
    46         3       3209.0   1069.7      0.0          count = sum(1 for agent in agents if
    47         3          1.0      0.3      0.0          status_counts[status].append(count)
    48
```

Optimization

Improved Speed with Cython and Vectorization

- C compilation allows for fast execution
- Low-level language (C) operates quickly on large data
- Vectorization with NumPy allows for parallel operations

Simulation	% Time	Line Contents
basic_sim - Original	98.0	infect(...)
basic_sim - Vectorization	55.6	infect(...)



Conclusion

- Simulation takeaways
- Optimization takeaways
  - Trade-offs
- Future Directions
  - Social distances
  - Central locations
  - Possibility of reinfections
  - Age-related vulnerabilities

Links

Our public Git repository:
https://github.com/xh852/Epidemiological-Agent-based-Model-Optimization

References:

1. The Institute for Disease Modeling: https://idmod.org/
2. 3Blue1Brown's "Simulating an epidemic": https://www.youtube.com/watch?v=gxAaO2rsdIs
3. This Week's Cartoons: April 13-17, 2020: https://www.wired.com/story/wired-cartoons-week-35/