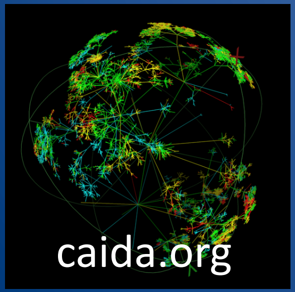
Embedding System Dynamics in Agent Based Models for Complex Adaptive Systems

Kiyan Ahmadizadeh, Maarika Teose, Carla Gomes,
Yrjo Grohn, Steve Ellner, Eoin O'Mahony, Becky Smith,
Zhao Lu, Becky Mitchell

Last time on Topics in Computational Sustainability…

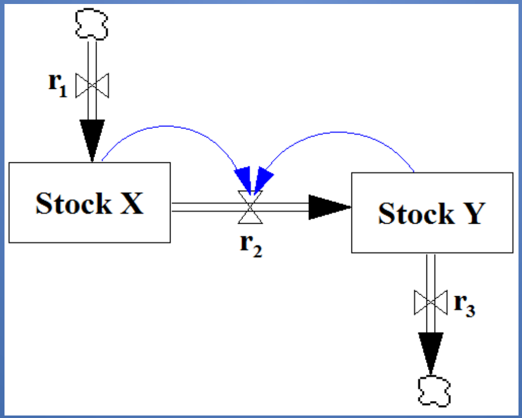
- We saw examples of different Complex Adaptive Systems
  - Disease control in Food/Animal Systems (Population Medicine)

  - Energy grids, social networks, ecosystems, ocean/ atmosphere systems, etc.



Last time on Topics in Computational Sustainability…

- We saw the System Dynamics approach to modeling CAS
- Systems of Ordinary Differential Equations (ODEs) model the flow of agents between different stocks.



- Many advantages: Easy model construction, parameterization, and validation. Efficient simulation algorithms.

System Dynamics: The Disadvantages

- The assumption that agents are essentially *homogenous*.

- Agent state space has one variable indicating stock membership.
- What about additional biological, social, economic state?
- Additional state could be dependent on stock membership.
- Can only target *interventions* based on stock-membership.
  - Can be uninformative to policy-makers.
- The assumption that agents have *well-mixed interactions*.
  - Unrealistic representation of the dynamics of many CAS.

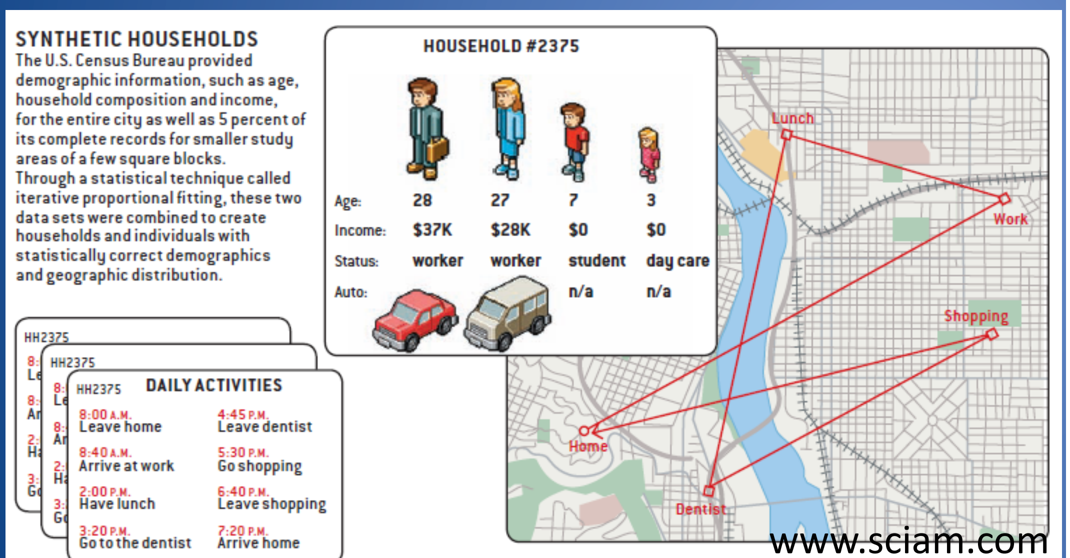
An Alternative: Agent-Based Modeling

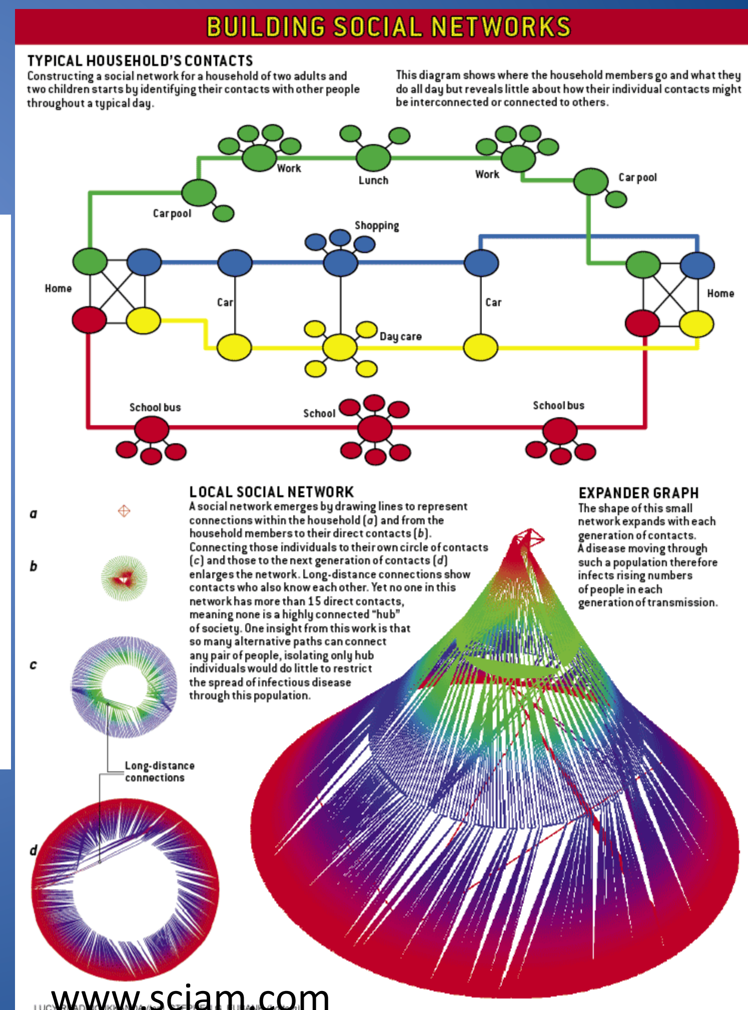
- The bottom-up approach
- Agents have heterogeneous state space updated through local interactions.
- Very general, high expressive power.
- The Cost:
  - Bottom-up construction, parameterization and validation is difficult.
    - Did we get all the feedback loops?
  - Simulation can have high time and memory requirements.
  - Models often become application specific.

Example: EpiSims

– Highly detailed

– Virtual laboratory





Striking a Balance

- *Goal*: Combine agent-based modeling and system dynamics to create models for CAS that retain the advantages of both paradigms.
- *Our Solution*:
  - Define a class of agent-based models with an embedded system dynamics model.
  - Give a simulation framework for these models.
    - Algorithm for model simulation.
    - Semantics of simulation framework specify how embedding occurs.

Our Class of Embedded Models

- We define an embedded model as a tuple *M* = (*S*, *A*, *O*, *U*, *D*, *V*)
- Let *Name* be a set of identifiers
- *S* is a set of *local state variable* names
  - Holds general agent state
- *A* is a set of *ODE state variable names*
  - A will contain one variable per embedded system dynamics model taking values that name stocks.
- Together, *S* and *A* divide the state space of agents.

Our Class of Embedded Models

- We define an embedded model as a tuple *M* = (*S*, *A*, *O*, *U*, *D*, *V*)
- *O* is a set of tuples of the form

  (*Name*, *Name*, **R**)

  Specifying the rates of transition between stocks.
- Reserve names *Gen* and *Des* for the source of generative flow and destination of destructive flow.
- *O* specifies the embedded system dynamics model.

Our Class of Embedded Models

- We define an embedded model as a tuple *M* = (*S*, *A*, *O*, *U*, *D*, *V*)
- *U* is a set of *local state update functions*
  - These functions model agent actions and interactions.
  - May read an agent's local state and ODE state variables.
  - Can only modify an agent's local state variables.
  - May suggest the generation or destruction of agents.

Our Class of Embedded Models

- We define an embedded model as a tuple *M* = (*S*, *A*, *O*, *U*, *D*, *V*)
- *D* is a set of *demographic functions*
  - These functions accept suggestions on agent generation and destruction.
  - May read an agent's local state and ODE state variables.
  - May modify the existing population of agents.

Our Class of Embedded Models

- We define an embedded model as a tuple $M = (S, A, O, U, D, V)$
- V is a set of *intervention functions*
  - May read and write to the entire state space of agents.
  - May modify the existing agent population.
  - Meant to model high-level actors with influence or control over the CAS.

Simulation framework

- Let Λ and Θ be updatable maps from agent and variable names to values.
- Let P be the current population of agents.
- Let P_{gen} and P_{des} be sets that hold suggestions on agent generation or destruction.

Simulation Framework

Algorithm 2 Execution of one time step for our simulation framework

Input: Embedded model $M = (S, A, O, U, D, V)$, set of agent names P, local state map Λ and ODE state map Θ.

$P_{gen} \leftarrow P_{des} \leftarrow \{\}$
for all local state update functions $u \in U$ **do**
  $(P_{gen}, P_{des}, \Lambda) \leftarrow u(P, P_{gen}, P_{des}, \Lambda, \Theta)$
end for
$(P_{gen}, P_{des}, \Theta) \leftarrow$ **ODESimulation**$(O, P_{gen}, P_{des}, \Theta)$
for all demography functions $d \in D$ **do**
  $P \leftarrow d(P, P_{gen}, P_{des}, \Lambda, \Theta)$
end for
for all intervention functions $i \in V$ **do**
  $(P, \Lambda, \Theta) \leftarrow i(P, \Lambda, \Theta)$
end for

Simulation Framework: Data Access

Algorithm 2 Execution of one time step for our simulation framework

Input: Embedded model $M = (S, A, O, U, D, V)$, set of agent names P, local state map Λ and ODE state map Θ.

$P_{gen} \leftarrow P_{des} \leftarrow \{\}$

for all local state update functions $u \in U$ **do**

  $(P_{gen}, P_{des}, \Lambda) \leftarrow u(P, P_{gen}, P_{des}, \Lambda, \Theta)$

end for

$(P_{gen}, P_{des}, \Theta) \leftarrow \mathbf{ODESimulation}(O, P_{gen}, P_{des}, \Theta)$

for all demography functions $d \in D$ **do**

  $P \leftarrow d(P, P_{gen}, P_{des}, \Lambda, \Theta)$

end for

for all intervention functions $i \in V$ **do**

  $(P, \Lambda, \Theta) \leftarrow i(P, \Lambda, \Theta)$

end for

Simulation Framework: Demographics

Algorithm 2 Execution of one time step for our simulation framework

Input: Embedded model $M = (S, A, O, U, D, V)$, set of agent names P, local state map Λ and ODE state map Θ.

$P_{gen} \leftarrow P_{des} \leftarrow \{\}$

for all local state update functions $u \in U$ **do**

$(P_{gen}, P_{des}, \Lambda) \leftarrow u(P, P_{gen}, P_{des}, \Lambda, \Theta)$

end for

$(P_{gen}, P_{des}, \Theta) \leftarrow$ **ODESimulation**$(O, P_{gen}, P_{des}, \Theta)$

for all demography functions $d \in D$ **do**

$P \leftarrow d(P, P_{gen}, P_{des}, \Lambda, \Theta)$

end for

for all intervention functions $i \in V$ **do**

$(P, \Lambda, \Theta) \leftarrow i(P, \Lambda, \Theta)$

end for

Simulation Framework: Interventions

Algorithm 2 Execution of one time step for our simulation framework

Input: Embedded model $M = (S, A, O, U, D, V)$, set of agent names P, local state map Λ and ODE state map Θ.

$P_{gen} \leftarrow P_{des} \leftarrow \{\}$
for all local state update functions $u \in U$ **do**
  $(P_{gen}, P_{des}, \Lambda) \leftarrow u(P, P_{gen}, P_{des}, \Lambda, \Theta)$
end for
$(P_{gen}, P_{des}, \Theta) \leftarrow \mathbf{ODESimulation}(O, P_{gen}, P_{des}, \Theta)$
for all demography functions $d \in D$ **do**
  $P \leftarrow d(P, P_{gen}, P_{des}, \Lambda, \Theta)$
end for
for all intervention functions $i \in V$ **do**
  $(P, \Lambda, \Theta) \leftarrow i(P, \Lambda, \Theta)$
end for

Embedded Model: Examples

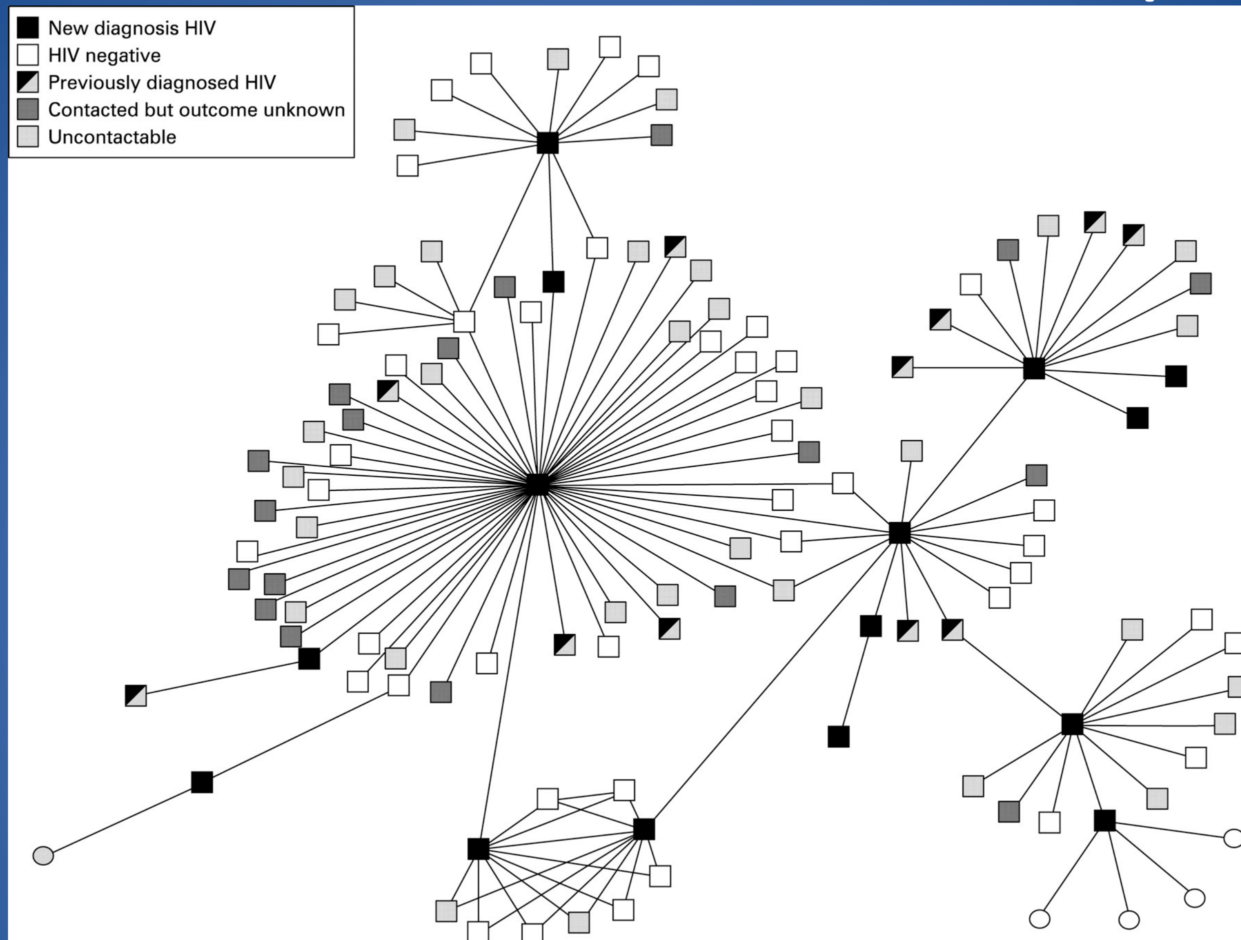
- Embedded models can be used for many CAS
  - Species distribution in ecosystems, information dispersion in a network, energy grids, etc.
- We give two examples of CAS from epidemiology that highlight the advantages of an embedded model.
  - Sexually Transmitted Infections (STI)
  - Johne's Disease (MAP)

STIs and the *well-mixed* assumption.

- WHO estimates 1 million people infected daily.
- Epidemics like HIV/AIDS impact world health and economy.

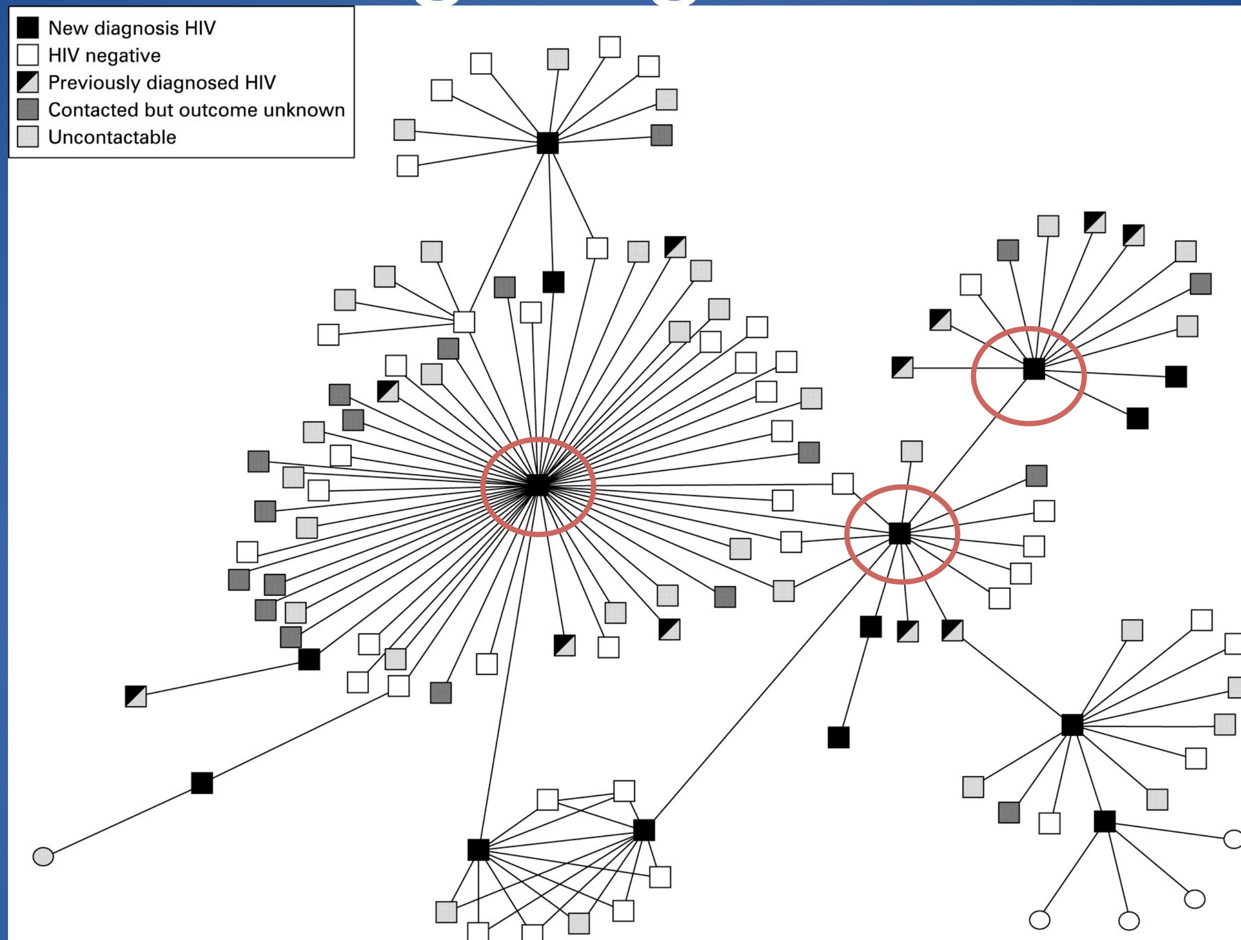
- Well-mixed assumption is fine for disease progression.
- Is it valid for transmission?

STIs and the well-mixed assumption.



- Sexual contact network in South Wales
- Network exhibits “small world” properties but is not well-mixed

STIs and targeting interventions



- Often, interventions to control STI outbreaks target agents based on their contact network (*contact tracing*)
- How can you model contact tracing with system dynamics?

STIs and targeting interventions

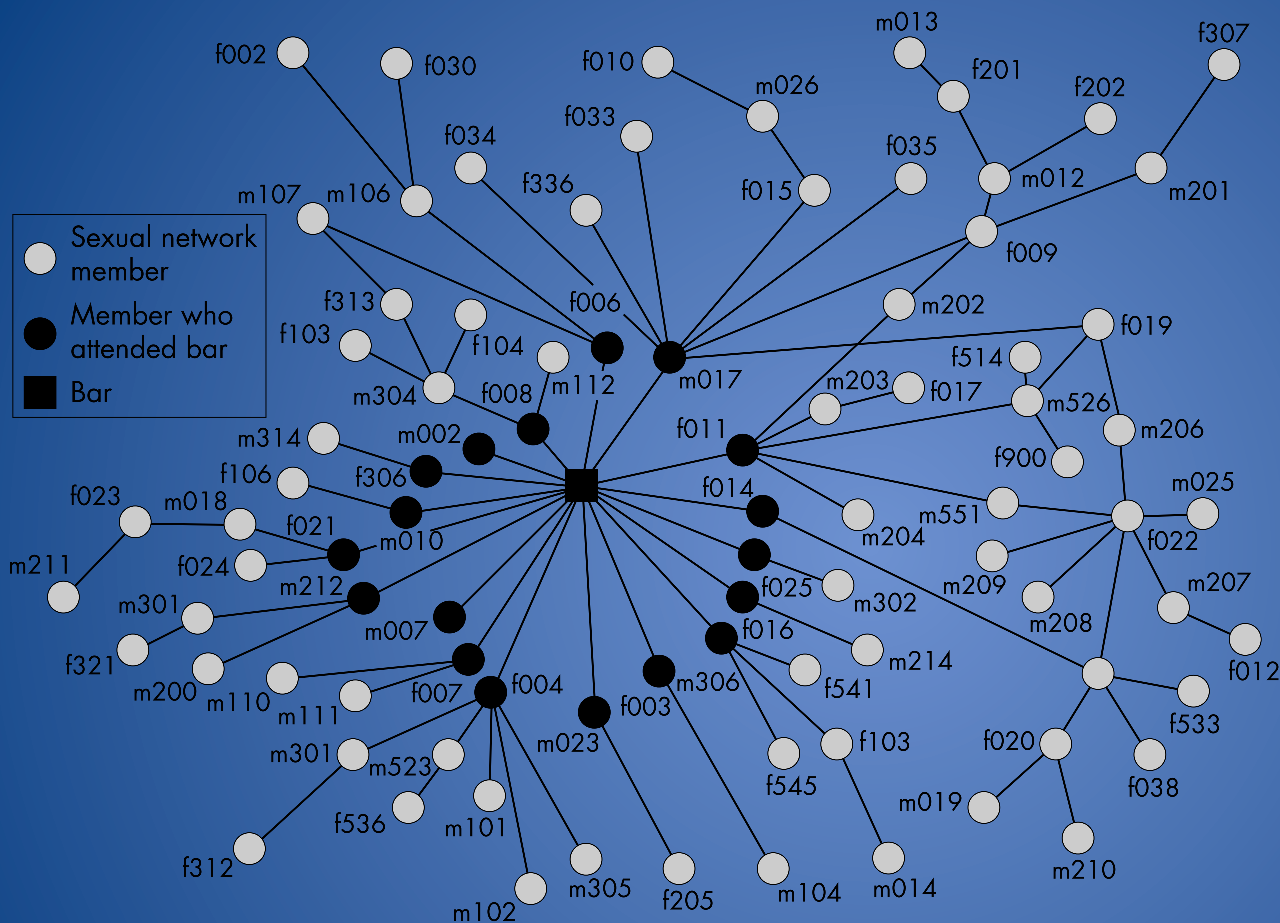
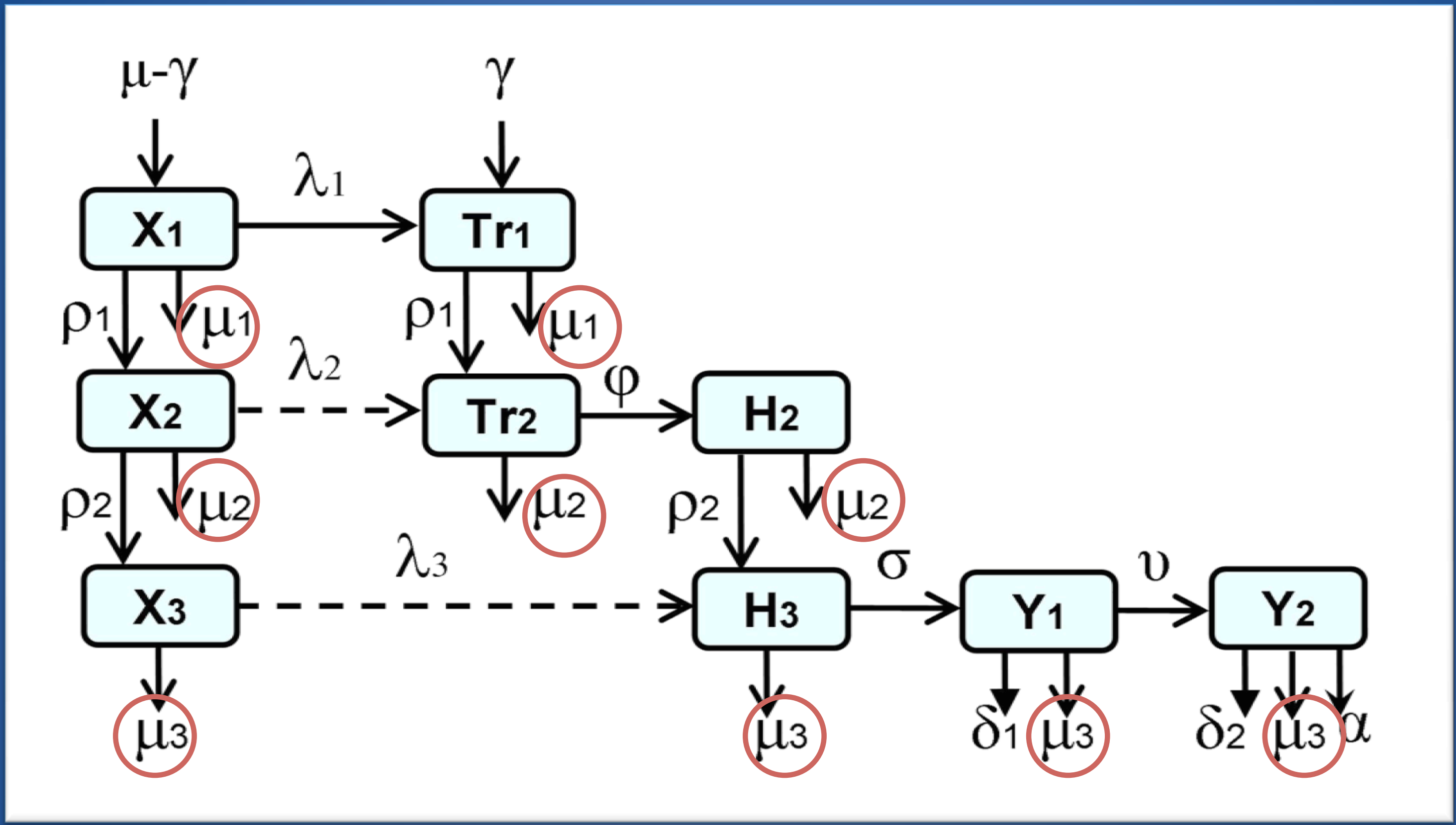


Figure 2 Network members (n = 89) viewed by their connection through a bar associated with gonorrhoea acquisition. A prefix to the unique identifier of "m" designates a male and "f" indicates a female sexual partner. Bar patrons possessed significantly higher information centrality measures compared to non-patrons (table 3).

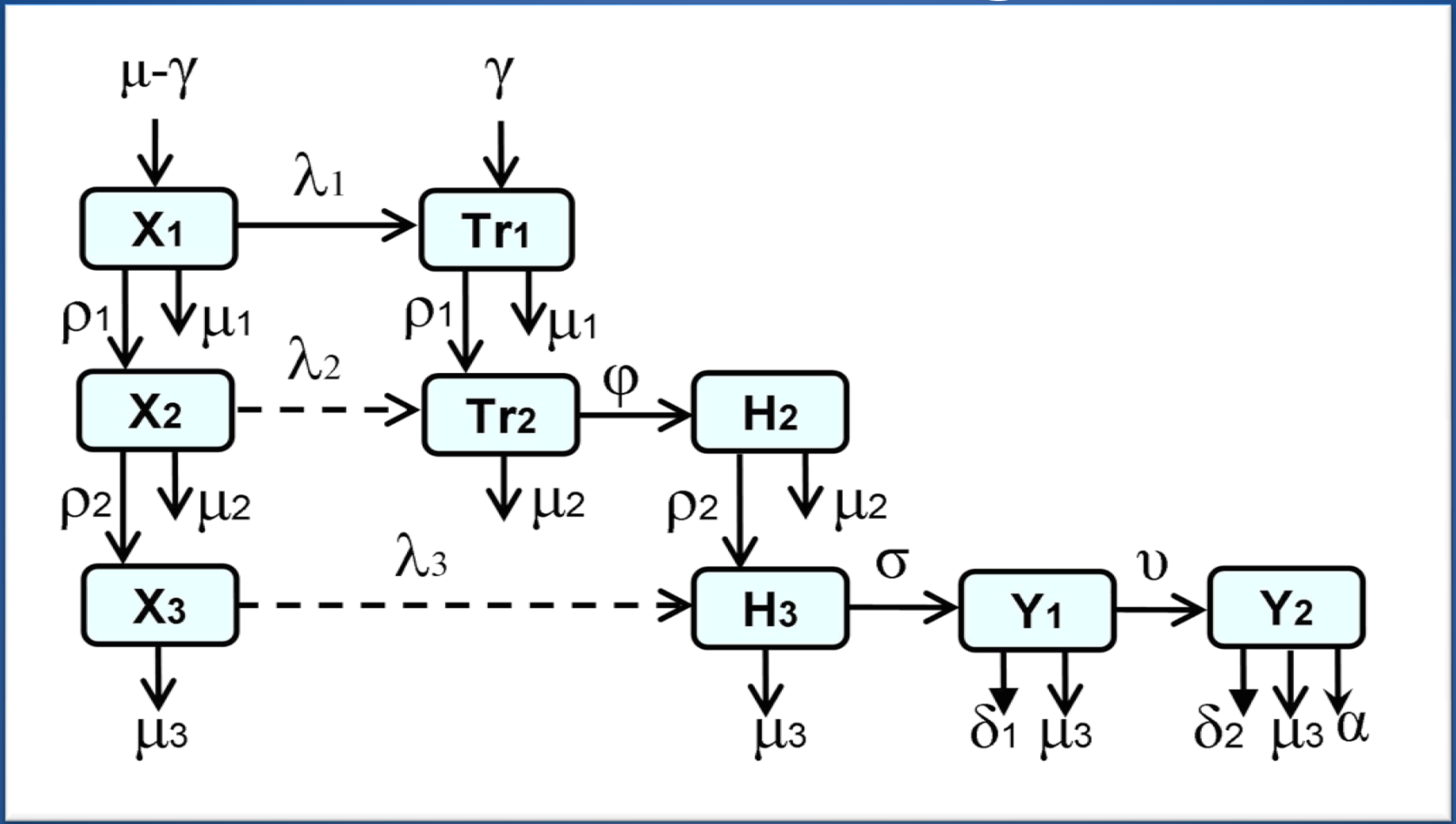
- Interventions also targeted at *locations*.
  - Additional state unattainable with ODE model.

MAP and modeling control



- Question: What are the μ s modeling?

MAP and modeling control



- Answer:
  - Animal death and *farmer management actions*
  - Farmer primarily controls farm by buying and selling animals.
  - Animal's economic value plays strong role in farmer's decisions.
  - Main objective is to make the most profit, not to control disease.
    - New management policies might be developed.

Embedded Model for MAP

- An embedded model solves many of these problems.
- Agent-based actions can model farmer's management policies.
- Extra economic state can be maintained for agents.
  - Total milk production, reproductive history, current pregnancy status, etc.
  - Milk production and reproductive functions dependent on MAP status.
  - Optimization!
- Management policies can be implemented mechanistically as in reality.

Embedded Model for MAP

- $M = (S, A, O, U, D, V)$
- *S* names local state variables that hold biological/economic data on cows.
- *A* names one variable that holds current MAP disease status.
- *O* specifies the MAP ODE model.

Embedded Model for MAP

- $M = (S, A, O, U, D, V)$
- The local update functions *U* maintain additional agent state.
  - Agent's reproductive status updated according to farm policies and biology.
  - Agent's milk production is tracked, and influenced by disease state.
  - Agent reproduction suggestions the generation of new agents.

Embedded Model for MAP

- $M = (S, A, O, U, D, V)$
- The demographic functions *D* model basic farmer management decisions.
  - Routine buying and selling of animals.
  - Value for each agent computed based on their current state (economic value).
  - Low value agents are removed from herd to make room for new agents.

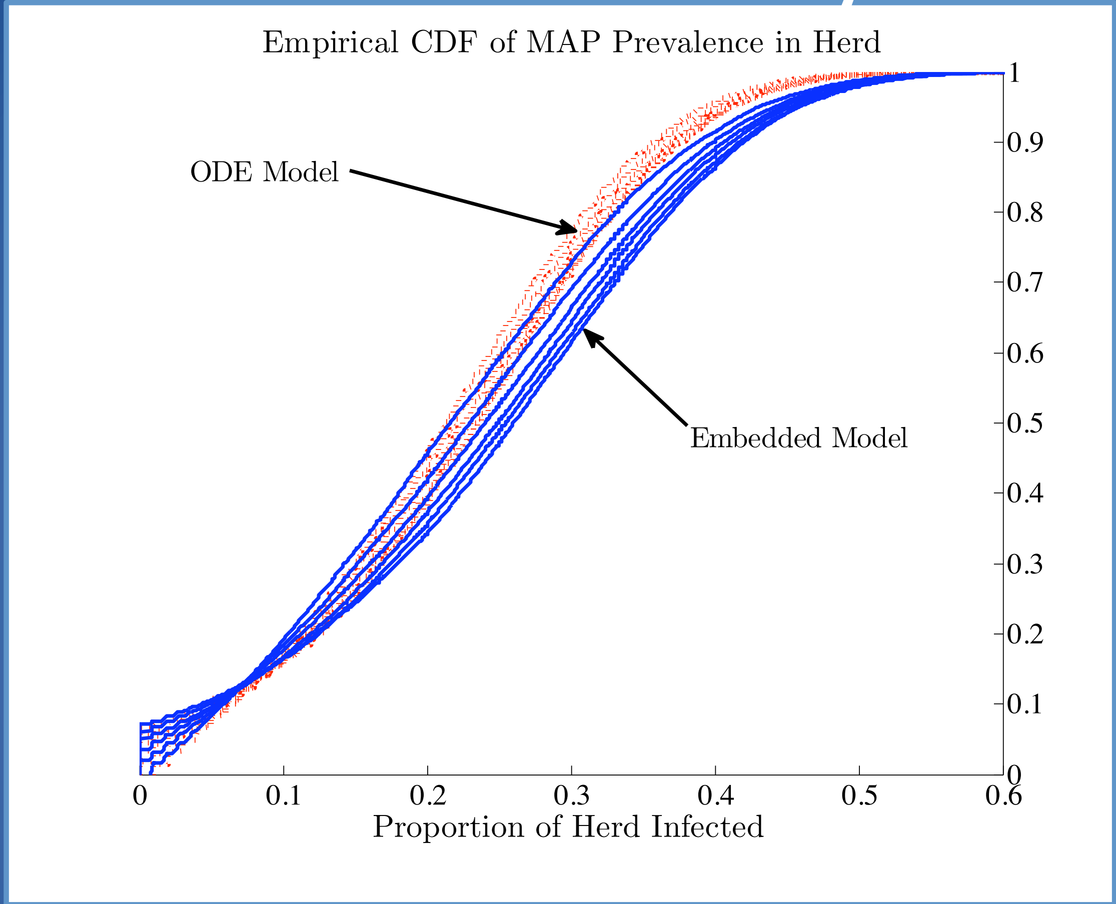
Embedded Model for MAP

- *M* = (*S*, *A*, *O*, *U*, *D*, *V*)
- The intervention functions V model controlling for MAP
  - Farmer tests for MAP every six months.
  - Two control policies:
    - ***Test-and-Cull***: Farmer removes test positive cows from herd when they're spotted.
    - ***Milk-Test-and-Cull***: Remove test positive cows from herd, but delay the removal of cows with high milk production.

Experimental Setup

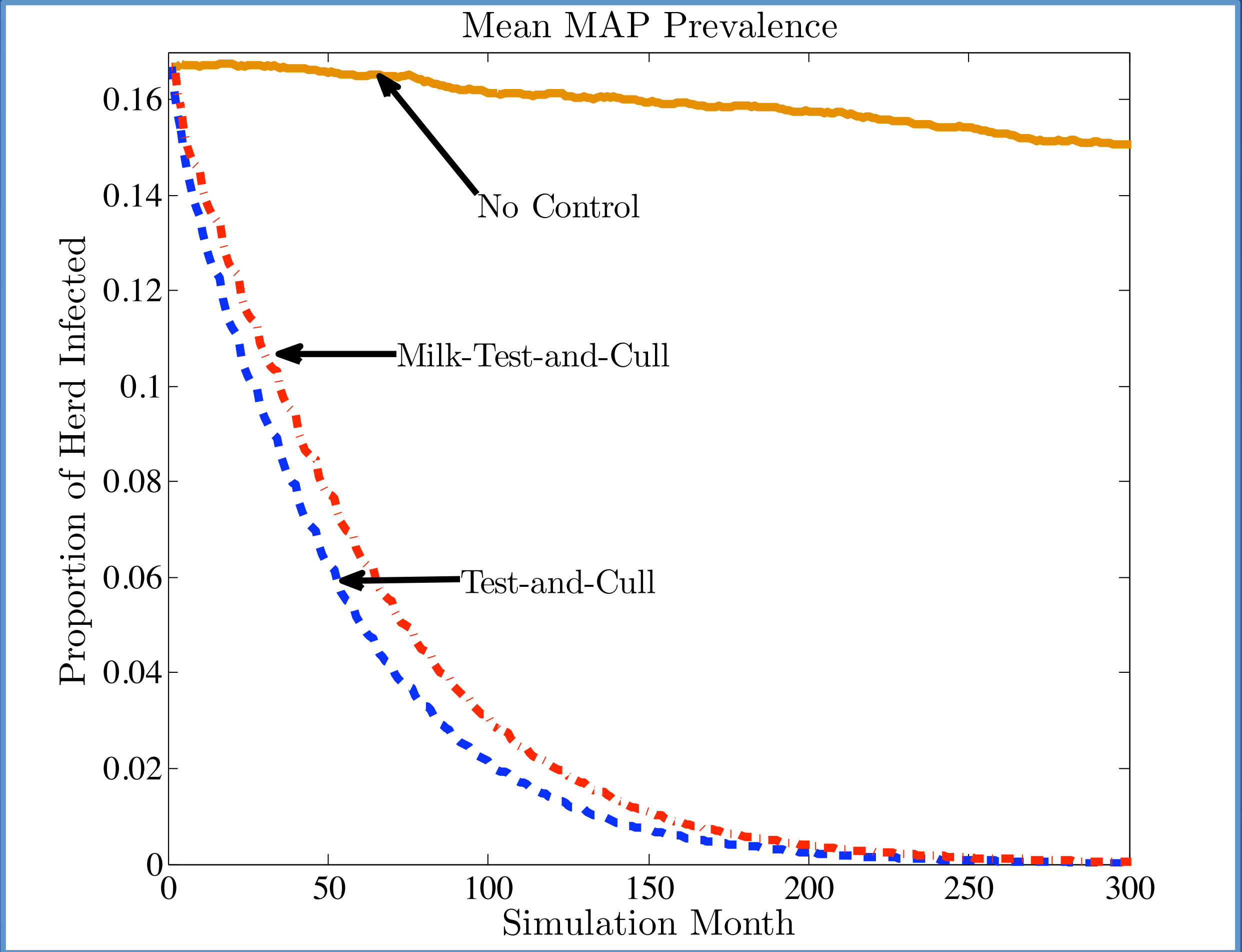
- Implemented simulation framework in Java.
- Constructed an embedded model for MAP on dairy farms.
- Ran 3 sets of 100,000 simulations:
  - 50 year run-up period to obtain *endemic equilibrium*
  - Control strategy used for next 25 years:
    - No control, Test-and-Cull, Milk-Test-and-Cull
- **Question**: Does the model produce accurate disease dynamics?
- **Question**: Which control strategy is "best”?

Results: Disease Dynamics



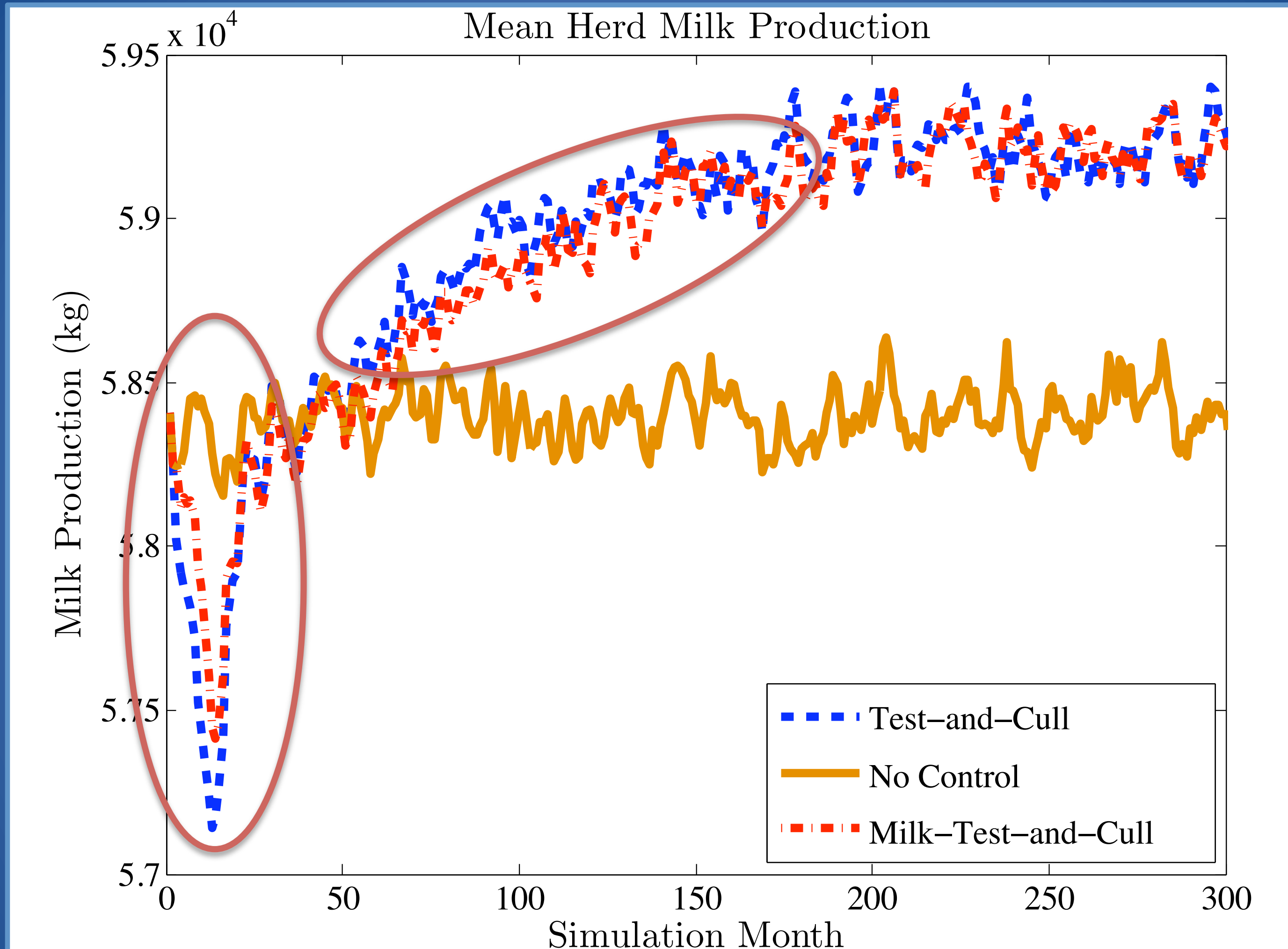
- Each line gives prevalence of MAP at five year intervals under no control.
- Distributions of results for both models are close.
- ODE results known to be valid.

Results: Disease eradication



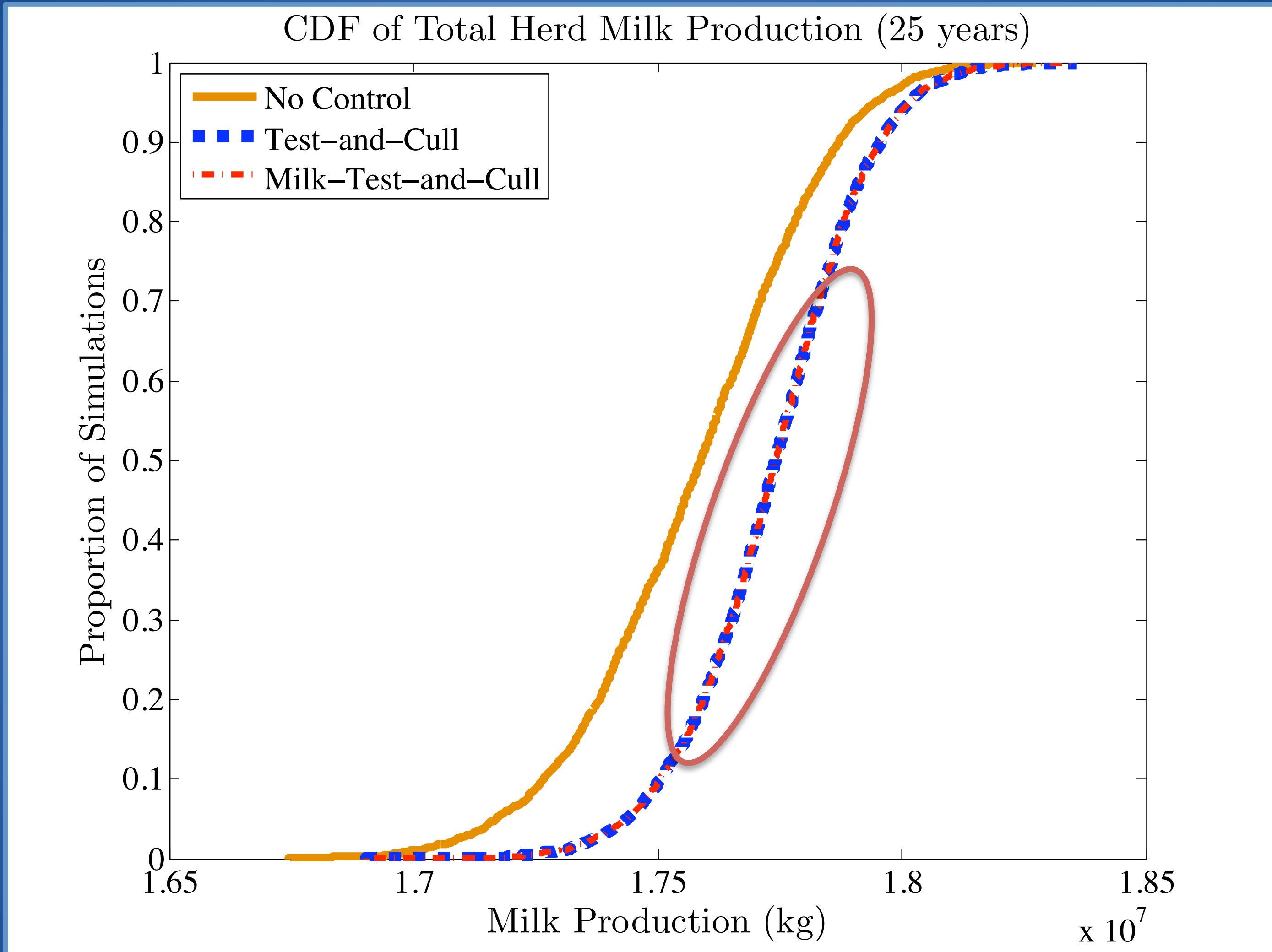
- Both Test-and-Cull and Milk-Test-and-Cull result in fadeout.
- Milk-Test-and-Cull, on average, has slower fadeout.

Results: Milk Production



- Milk-Test-and-Cull has short-term gains but may have long-term costs.

Results: Milk Production



- In the long term, both strategies have similar performance for total milk production.

Conclusions on Control Strategies:

- Both Test-and-Cull and Milk-Test-and-Cull result in MAP eradication.
- Both Test-and-Cull and Milk-Test-and-Cull have short-term costs.
  - Milk-Test-and-Cull mitigates these costs.
- Both strategies have similar long-term performance for milk production.
- ***Conclusion****: Milk-Test-and-Cull can be used to control MAP and mitigate the short-term costs of control without sacrificing long-term gains.*

Future work:

- New embedded models: Computational Sustainability seeks to influence many CAS.
  - Species distribution and social networks.
- Optimization:
  - Optimization problem is computationally difficult, and seeks to find an *optimal policy function*
  - Example for MAP:
    - $\Psi(\Lambda, \Theta)$ is a policy function that given the state space of all agents, returns a set of agents to cull.
    - $\Psi^*(\Lambda, \Theta)$ is an optimal policy function that given the state space of all agents, returns a set of agents to cull that is optimal in regards to economic output.
  - Techniques from Machine Learning and Game Playing could be very useful!