geographical analysis

*Geographical Analysis* (2020) **0**, 1–33

# A Concurrent Entity Component System for Geographical Wildlife Epidemiological Modeling

Austin V. Davis[1,2,3], Shaowen Wang[1,2,4,5,6]

[1]CyberInfrastructure and Geospatial Information Laboratory, University of Illinois at Urbana-Champaign, 1301 W Green St., MC 150, Urbana, IL 61801, USA, [2]Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign, 1301 W Green St., MC 150, Urbana, IL 61801, USA, [3]Environmental Laboratory, U. S. Army Engineer Research and Development Center, 3909 Halls Ferry Rd., Vicksburg, MS 39180, USA, [4]Department of Computer Science, University of Illinois at Urbana-Champaign, 201 North Goodwin Ave. MC 258, Urbana, IL 61801-2302, USA, [5]Department of Urban and Regional Planning, University of Illinois at Urbana-Champaign, 611 Taft Drive, MC 619, Champaign, IL 61820, USA, [6]National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 1205 W. Clark St., MC-257 Room 1008, Urbana, IL 61801, USA

*North American bat species have been undergoing extreme population declines due to the White-Nose Syndrome (WNS) epidemic caused by the spread of its pathogen, Pseudogymnoascus destructans. Existing models that represent the spread of the disease are limited in their scalability for use in management decisions or lacked the sophistication necessary to capture the complexity of WNS spread. Grounded in the theory of geoexpression, we exploit the latent structure of geographical process concurrency by implementing our modeling software using a concurrent Entity Component System architecture. We demonstrate our model's computational tractability for millions of individual bats. This work is significant because it lays the foundation for the use of advanced cyberinfrastructure and cyberGIS to address challenges in geographical wildlife epidemiology that can be understood using dynamic geographical Individual-Based Models.*

## Introduction

Wildlife epidemiology is a frequent subject of dynamic environmental modeling motivating geographical discovery. Understanding the interactions of epidemiological, biological, and environmental processes that affect disease spread leads to more effective treatment methods and species conservation (Maher et al. 2012; Hammerson et al. 2017; White, Forester, and Craft 2017). North American bat species have been undergoing extreme population declines due to the White-Nose Syndrome (WNS) epidemic (Foley et al. 2011; Maher et al. 2012; Thogmartin et al. 2012). The first documented outbreak of WNS occurred at a winter hibernaculum in Schoharie County, New York, during the 2005–2006 winter hibernation (Blehert et al. 2009). WNS is caused by a fungal pathogen, *Pseudogymnoascus destructans* (Pd), which is commonly spread

Correspondence: Austin V. Davis, CyberInfrastructure and Geospatial Information Laboratory, University of Illinois at Urbana-Champaign, 1301 W Green St., MC 150, Urbana, IL 61801, USA
e-mail: austin.v.davis@usace.army.mil

Submitted: December 21, 2019. Revised version accepted: August 24, 2020.

doi: 10.1111/gean.12258

Highlights
- The spread of White-Nose Syndrome in North American bat populations is challenging.
- White-Nose Syndrome spreads geographically due to geographical process concurrency.
- *Geoexpression* provides a framework to represent geographical process concurrency.
- We implement a model based on *geoexpression* and demonstrate its tractability.

between winter hibernacula by bats (Trivedi et al. 2017). Since the winter of 2005, the disease has spread throughout much of the East-Central United States and portions of Canada (Dzal et al. 2011; White-Nose Syndrome Response Team 2018). In the first six years of the White-Nose Syndrome epidemic, 6.7 million bats died of the disease (White-Nose Syndrome Response Team 2018). Victims of the disease, contracted within hibernacula, present white fungal growth over the nostrils and wings accompanied by life-threatening physical function changes. Bat biologists and natural resource managers currently conduct monitoring and surveillance efforts to assess the continued expansion of the epidemic (Bat Conservation International 2019). Many treatment approaches have been evaluated, but the effectiveness of these approaches relies upon a greater understanding of how the disease spreads geographically.

Colonial cave roosting vespertilionid species face greater risk due to increased interaction with the WNS pathogen and conspecifics (Ihlo and Baker 2013). Since the initial outbreak in 2005, due to massive population declines of multiple species, a significant amount of spatial epidemiological WNS modeling has been conducted on vespertilionid bats. Maher et al. (2012) constructed a statistical model to predict WNS spread. To address the limitations of the statistical model as a decision aid, Ihlo and Baker (2013) used spatial analysis to tie the spread of WNS to a set of geographical variables. O'Regan et al. (2015) introduced a dynamic spatial epidemiological model as a series of differential equations to address the lack of dynamic representation in the prior efforts, but represented bat species without the ability to recover. In 2018, Lilley, Anttila, and Ruokolainen developed a set of mean-field (not individual-based) models to represent WNS spread. The Lilley, Anttila, and Ruokolainen model does not represent the spread of WNS across different species of bats and treats the population as a single generic species even though different bat species are known to have different infection potentials. A model that represents the geographical spread of WNS with different parameters for different species of bats, over large numbers of roost sites, is needed to address the shortcomings of prior studies. Our study builds upon existing research efforts and aims to develop a computational Individual-Based Model (IBM) that represents the spread of WNS by exploiting geographical process concurrency.

As cyberinfrastructure has advanced, the geographical sciences have sought to exploit more of its capabilities. Investigations into the exploitation and latent parallelism of computationally intensive geographical analysis have only been possible with advancements in computer hardware, new theoretical approaches, and computational methods (Wang and Armstrong 2009). Previous research has pursued the adaptation of existing geographic problem-solving tools into cyberGIS components known as spatial middleware (Wang 2010). These investigations evolved to discern how cyberinfrastructure can serve as a basis for further inquiry toward the influence of space-time pattern representation using epidemiological modeling as a case study (Shook and Wang 2015). The next logical step, explored first in the work of Davis and Wang (2018) and now here, is the investigation of how cyberinfrastructure motivates the need to expand the theory of

geographic process representation; specifically, to understand and exploit geographic process concurrency.

High-performance parallel computing can enhance the computational performance of spatial epidemiological modeling (Shook, Wang, and Tang 2013). Developing modeling software that exploits high-performance computing power is an important step toward the seamless integration of cyberGIS and spatial modeling based on advanced cyberinfrastructure (Wang et al. 2013; Lin et al. 2015). The geographical process concurrency between biological and epidemiological processes is a key aspect of the WNS epidemic. Geographical process concurrency is the decomposability of geographical processes into a structure of partially orderable components that take place within spatial and temporal configurations to represent geospatial dynamics. Once this system of concurrent processes is structured graphically, as examined by Davis and Wang's (2018) *geoexpression*, it can be used to exploit computationally performant software to represent latent concurrency in geographical processes.

Geographic representation is segmented into the representation of pattern and the representation of the process. Research has focused heavily on the representation of the pattern and very little on the representation of the process (Goodchild, Yuan, and Cova 2007). However, as introduced in Davis and Wang (2018), cyberinfrastructure has motivated the need to explore a new process representation theory that captures and exploits the notion of geographical process concurrency. Geographical process concurrency is an underexplored frontier in geographical theories; especially as it relates to the representation of processes using computational machines. Here we demonstrate that geographical process concurrency can be exploited to identify a latent structure of concurrency to construct a geographical model. The *geoexpression* is a framework that structures geographical process concurrency using a graphical representation to express the relationships between a set of related geographical processes in a way that is mutually understandable by a human and computational machine. Davis and Wang (2018) conceptualize geographical process concurrency through *geoexpression* using a graphical language, Petri-net, to represent the concurrent execution order of a set of processes. Once the concurrent execution order is structured, it is feasible to identify regions within the model and software that are conducive to parallel execution, which cyberinfrastructure is well suited to support.

The primary objective of this research is to demonstrate and evaluate the performance of a software solution that implements an IBM of WNS spread using an Entity Component System (ECS) architectural pattern based on *geoexpression*. IBM is a modeling strategy that can be used to investigate how individual interactions produce a collective outcome. Our approach applies an IBM and *geoexpression* to capture geographical concurrency in WNS spread to support management decision making. Improvement in WNS spread modeling is important to develop better surveillance strategies, efficacious treatment techniques, and more effective disease response and species recovery plans. The implementation of our IBM WNS spread modeling software is informed by *geoexpression* to tractably structure computational and geographical concurrency within the software.

ECS is an architectural pattern drawn from the data-oriented design paradigm of software development (Danielsson and Bohlin 2015; Majerech 2015). Traditional object-oriented design constructs software by focusing on the structure of the code into classes, inheritance relationships, and abstractions. Data-oriented design views programs from the perspective of how computation transforms data. Data-oriented design is focused on reducing the hazard of cache misses to support more performant code execution (Fabian 2013). In modern computers, data in the main memory are hundreds of clock cycles away from the processing unit. Data that resides

within a processing unit's cache take significantly fewer clock cycles to access; and modern computer hardware will preload data from main memory to cache based on the principle of locality. Therefore, computer performance is heavily affected by data locality issues (Fabian 2013).

ECS structures data by composition rather than inheritance to improve the cache performance of the software by exploiting the principle of locality heuristics. Entities are represented using a single integer identifier for each entity. Components are stored in disparate arrays of memory indexed by the Entity integer value and describe the attributes of each entity, and Systems contain the logical operations of the program. ECS also has Resources that serve as accessible storage for global attributes. Our IBM is instantiated using a parallel ECS that allows multiple Systems to be ordered for concurrent execution. Additionally, a parallel ECS enables data-parallel computation to occur on Component data within a System. In our IBM, *geoexpression* is used to order the Systems for concurrent execution within the parallel ECS to model the spread of WNS.

Our modeling software uses the SPECS Parallel Entity Component System (SPECS) library authored in Rust (Klabnik & Nichols 2019). Rust was sponsored in 2009 as a systems language by the Mozilla Foundation. Rust includes many features, including guaranteed memory safety and concurrency without data races. Its guarantees are enforced at compile time and it implements a static garbage collector. The language's ownership model guides a static analysis made by the compiler to determine and enforce its guarantees. The Rust compiler reaches an intermediate representation language before the final compilation that runs on a virtual machine. It has cross-platform support, a supportive user community, an excellent build system, and growing user groups focused on extending its capabilities to advanced cyberinfrastructure (Blandy 2015; Scull 2015).

This research demonstrates the performance of IBM software that is built using a parallel ECS for modeling the geographical spread of WNS through North American bat populations. Prior modeling efforts have either been too abstract for spatial representation or lacked the sophistication necessary to capture the complexity of WNS spread (Ihlo and Baker 2013; O'Regan et al. 2015). During the initial outbreak, limited data availability, a lack of understanding of the pathogen's spread, and limited knowledge of how WNS affects the life cycle of multiple bat species constrained related methodologies to understand the hazard posed by the disease and predict its spread. We introduce a straightforward IBM of WNS spread based on existing literature, implement the model using an existing parallel ECS library, and evaluate its computational tractability for different landscape and population configurations. We conclude with a discussion of the importance of capturing geographic process concurrency, present and future benefits of using a parallel ECS for instantiating a wildlife epidemiological IBM, and remark on the use of Rust as a language for environmental modeling.

## Methods

O'Regan et al. (2015) were the first to demonstrate the utility of emergent modeling to understand the spatial spread of WNS. Their approach built on the findings of the statistical model developed by Maher et al. (2012) that categorizes the processes that spread WNS into a multiscale hierarchy: (1) bat-to-bat, individual scale, transmission processes; (2) transmission between hibernacula; and (3) transmission over large distances on the landscape. Our model builds on the work by O'Regan et al. (2015) to understand how individual movement across landscape causes the emergent spread of WNS through the interaction of biological, epidemiological, and geographical processes.

Instead of categorizing processes that spread WNS into a multi-scale hierarchy, our model categorizes processes based on the epidemiological triangle (a process is either related to the host, pathogen, or environment). Additionally, our model differs from the O'Regan et al. (2015) approach because each bat is represented as a separate individual within our IBM. However, our results are aggregated to the county level following O'Regan et al. (2015) and Maher et al. (2012), because it serves as the minimum mapping unit at which disease surveillance data are reported and management decisions are made. Our IBM is devised using the epidemiological triangle as a conceptual tool to categorize the decomposition of processes that interact to spread WNS across North America. The epidemiological triangle has three connected points describing the foundations for the spread of a disease: host, pathogen, and landscape. Interacting processes of the host, pathogen, and landscape are decomposed to identify latent process concurrency in the drivers that spread WNS. Once the processes have been enumerated, Petri networks are used to represent their order and interactions graphically. As a graphical language, Petri networks are well suited to identify regions of latent concurrency that can be represented and exploited in computation (Davis and Wang 2018).

The concurrent execution strategy of our IBM instantiated using the SPECS library is borne out of the Petri networks that represent the *geoexpression* (i.e., the structure of geographical process concurrency) framework for WNS spread. Once instantiated, the model's run-time performance is collected for a series of tests that vary the number of roost sites (e.g., the landscape size) and the bat population size to demonstrate the utility of the parallel ECS and the gains provided by explicitly representing the geographical process concurrency. Finally, the output of the IBM's WNS spread results, aggregated by county, is shown to demonstrate the utility and tractability of parallel ECS architecture for the IBMs.

**Geoexpression**

Dynamic geographical models are often composed of constituent computational processes that produce representative patterns observed in real-world geographical systems. The translation of model processes into computational processes requires new theoretical support, as the structure of how such processes interact is often fixed to represent a single order of events within the modeled world. However, fixed orders of events may not capture the inherent complexity of geographically concurrent processes. Geographically concurrent processes can become difficult to reason about and a theory that can represent this concurrency is sorely needed. Therefore, such a theory called *geoexpression* was proposed by Davis and Wang (2018) to relate model processes to computational processes based on graphical primitives and a mathematical foundation adapted from the network sciences.

*Geoexpression* represents geographical process concurrency for modeling geospatial dynamics using Petri networks. Petri networks are bipartite, directed, acyclic graphs composed of State and Transition nodes, Edges, and Tokens. Tokens represent data transformations through a network. Transition nodes apply operations to the data. States represent the input and output conditions of the Tokens as they pass into and out of Transition nodes. Edges link state nodes to transition nodes. Petri networks are able to represent the graphical structure and functioning of many operations (Peterson 1978). Intuitively rearranging the order of transition nodes changes the output. The graphical structure of *geoexpression* captures different orders to represent the same set of partially orderable components that produce a grander concurrent, geographical process.

The primary benefit that *geoexpression* provides is a representation theory for geographic processes, that when applied, captures different orderings of processes, which, in turn, leads

to different spatial patterns being produced. For example, multiple geographically concurrent processes function to spread WNS. Existing assumptions are that bats serve as the primary vector for disease spread geographically. The bats transport spores of *Pd* between roost sites, frequently during the swarm period when bats traditionally mate. The mechanisms involved in the geographical spread of *Pd* are dependent upon the life-cycle processes of the pathogen itself, the life-cycle activities of roosting bat populations, and suitable environmental conditions. Additionally, the human transmission of the spores on clothing and caving equipment is a separate vector, which spreads *Pd* spores to new roost sites. This secondary vector has motivated the need for extensive decontamination measures by cavers. However, current understanding of the seasonal transmission of spores, and whether or not transmission to a new locale is by bat or human vector, is not definitive. Modeling this aspect of WNS spread can naturally be understood as an orderability, or concurrency, problem in the way life-cycle and environmental processes exist concurrently on the landscape. By representing various orders in which different life-cycle and environmental processes interact using *geoexpression*, it becomes possible to understand how grander, geographical processes interact in a way mutually understandable by both a human and computational machine. In the Experiments section, an exploration of different orders through *geoexpression* manipulation is conducted to understand the effect that the lack of a definitive order has on our IBM's outcome and demonstrate the utility of *geoexpression* as a theory of representing geographical process concurrency.

## Model design

The epidemiological triangle denotes three categories necessary for an epidemic: the host, the pathogen, and the environment. Here, the bats are the host, the pathogen is *Pd*, and the environment is the set of roost sites. In this section, each of the three epidemiological categories is decomposed into subsequent processes, described, represented mathematically, and structured using *geoexpression* in a graphical diagram.

### *Host processes*

The life-cycle of bats can be generalized into three distinct tri-annual biological seasons: a Summer maternity season, a Swarm mating season, and a season for Winter hibernacula. During the summer months, females form maternity colonies to birth and raise their pups. At the end of the maternity season, juvenile and mature bats will leave the maternity colonies and begin to swarm. The swarm season is the primary mating period and is characterized by substantial activity occurring spatially on the landscape. Each individual bat will frequent multiple roost sites during the swarm period. During swarm, bats are likely to transport spores of *Pd* from infectious roost sites to the roost sites later used as hibernacula (which may or may not have been already exposed to *Pd*). Female bats exhibit delayed fertilization after mating with males during swarm and conceive a pup the following spring. Upon the conclusion of swarm, bats select a specific roost site to form a winter hibernacula. Many bat species exhibit fidelity toward roost sites and will return to the same winter hibernacula. Once the hibernacula is formed, the bats enter torpor and become susceptible to White-Nose Syndrome. During the winter hibernacula, if *Pd* is present at the roost site and is of sufficient bio-mass to produce infection, bats are very likely to become infected and exhibit disease-related mortality. If exposed bats survive the infection, the infection is cleared as they migrate to form summer maternity colonies. Bat populations are moderated by natural mortality. Natural mortality is more common during the winter hibernation

season, but can occur any time of the year. Natural mortality effects juveniles differently from mature bats and is caused by many factors.

A mathematical generalization of the host processes that are instantiated within our IBM is listed in Table 1. These expressions were derived through the generalization of the differential equations provided by Lilley, Anttila, and Ruokolainen's (2018) work. However, our model maintains the logistic population growth and mortality factors. Each expression is used to control a separate process related to the host's biology. These mathematical expressions are instantiated in the Systems of the parallel ECS to represent bats' biological processes. Each of the biological processes can be represented as *geoexpression* (aka. a graphical representation of the structure of interacting processes that represent geographical process concurrency) of the host processes.

Figure 1 depicts the *geoexpression* of the host's processes. It represents the interactions between host processes within the model and is used in the implementation of the model to identify areas of latent concurrency and guide the parallel execution of Systems.

***Pathogen processes***

Lilley, Anttila, and Ruokolainen's (2018) work focused on the understanding of intra-hibernacula disease spread through bat populations using a Susceptible Infected Recovered (SIR) state model. The SIR model instantiated a set of interacting differential equations from available literature to capture the parameters relevant to the interaction of bats with the WNS pathogen. The SIR model demonstrated that bats are more susceptible to WNS df4uring winter torpor because their suppressed immune systems render them vulnerable to fungal infection. Bats contract spores of *Pd* at roost sites of winter hibernacula. *Pd* is a cold-loving fungus that typically takes a year before it has grown to sufficiently infect the roosting bat population. If a bat is exposed to a roost site during Swarm with infectious levels of *Pd*, it can spread spores of *Pd* to its winter hibernacula before the bat enters torpor. Once established at a roost site, *Pd* can gain a permanent foothold that can reinfect future colonies of susceptible bats.

Figure 2 displays the structure of the pathogen's processes within our IBM. Two pathogenic processes occur: the growth of the fungal pathogen and the shedding of *Pd* spores. If bats have been exposed to spores and arrive at winter hibernacula, they shed spores into the roost site. The spores of *Pd* then grow over the course of a year to render the roost site infectious to the roosting bat population. If the mean annual temperature is too warm, the spores will not grow due to climatic constraints, and the roost site is not rendered infectious (Reynolds 2010). Due to the invasiveness of *Pd*, if spores are shed into a hibernaculum with acceptable temperature conditions, the spores will grow.

***Geographical processes***

Lilley, Anttila, and Ruokolainen (2018) developed an ecological model to investigate disease spread in spatially abstract networks using random clusters of hibernacula assigned to arbitrary distances sampled from a Poisson distribution. Their result demonstrated comparable dispersion rates as documented through disease surveillance data at varying levels of geographical proximity. Here, the intra-hibernacula infection of bats is adapted from the Lilley, Anttila, and Ruokolainen (2018) epidemiological model. While individual bat-to-bat exposure of *Pd* can be separated from bat-to-substrate exposure to *Pd*, it occurs at a spatial scale that is best approximated over for the IBM. The effect of White-Nose Syndrome is so extreme that if *Pd* has grown to viably infect local bat populations at a given roost site, the exact method of transmission (bat-to-bat or bat-to-substrate) for any given bat is of less relevant concern to species managers.

**Table 1.** A Listing of Equations that Govern the Host Processes within the Model

| Equation purpose | Equation | Term definitions |
|---|---|---|
| Instantaneous probability of age-related mortality | $M_{\text{age}} = \frac{1}{a_{max}} * a_i$ | $M_{\text{age}}$ is the mortality probability. $a_{\max}$ is the maximum species age. $a_i$ is the current age of the entity at index $i$ |
| Instantaneous probability of reproduction for a given bat given current total population | $R = \begin{cases} min\left(\frac{K-C}{K}, f_i\right), & \text{if } g_i = \text{female} \\ 0, & \wedge \text{if } g_i = \text{male} \end{cases}$ | $R$ is the probability of reproduction. $g_i$ is the gender for an entity at index $i$. $K$ is the maximum population capacity of the model. $C$ is the current population within the model. $f_i$ is the species' fecundity probability for the entity at index $i$. The fecundity probability is not modeled to differ between individuals of the same species |
| The age is incremented in 1 year increments within the model | $a_{i_t} = a_{i_{t-1}} + 1$ | $a_i$ is the current age of the entity at index $i$. $t$ is the current year |
| Instantaneous probability of mortality due to resource starvation | $M_s = \begin{cases} \frac{C_j - C_{j_{max}}}{C_j}, & C_j > C_{j_{max}} \\ 0, & C_j \le C_{j_{max}} \end{cases}$ | $M_s$ is the mortality probability. $C_j$ is the current population of bats at roost $j$. $C_{j\max}$ is the maximum population of bats supportable by roost $j$ |
| Probability of mortality due to disease | $M_d = \begin{cases} p \le I_i, & \wedge \text{if infected} \\ 0, & \text{if uninfected} \end{cases}$ | $M_d$ is the probability. $p$ is a random sample from a uniform distribution over [0,1). $I$ is the species mortality probability for the entity at index $i$ |

**Table 1.** (*Continued*)

| Equation purpose | Equation | Term definitions |
|---|---|---|
| Exposure determination by the visitation of adjacent roost sites | $f(y) = \begin{cases} \text{true}, & \text{if } y = \text{infected} \\ \text{false}, & \wedge \text{if } y \neq \text{infected} \end{cases}$ <br> $g(a,b,c) = \begin{cases} \text{true}, & \text{if } (a \leq b) \wedge f(c) \\ \text{false}, & \text{if } (a \leq b) \wedge \text{not } f(c) \\ \text{false}, & \text{if } (a > b) \wedge \text{not } f(c) \\ \text{false}, & \text{if } (a > b) \wedge \text{not } f(c) \end{cases}$ <br> $E_{\text{set}} = \{\ x \in S \mid \{ y \in \{\{\ r \in G(x) \mid f(r)\}\} \mid g\left(p, C_v * v_i, y\right)\ \} \}$ <br> $E_i = \text{any}\left(E_{\text{set}}\right)$ | $f$ is a function that determines the infection condition of its argument $y$. $x$ is a roost site in the set of all roost sites $S$. $y$ is a roost site within the set of roost sites that are adjacent to $x$ on the graph as determined by $G(x)$, the current roost site $r$, and are also infectious. $E_{set}$ is the exposure determination of all adjacent roost sites that were visited given the visitation probability of the species, $v_i$, multiplied by a roost visitation multiplier, $C_v$, a random sample, $p$, from a uniform distribution, and the roost site $y$. $E_i$ evaluates to true if any adjacent roost is infectious |
| Change winter hibernacula roost sites | $u_i = \begin{cases} \text{move}, & \text{if } p \leq d_i \\ \text{remain}, & \text{if } p > d_i \end{cases}$ | $u_i$ is true if the move conditions are met, false if not. $p$ is a random sample from a uniform distribution over [0,1). $d_i$ is the species fidelity to winter hibernacula |

Source: adapted from Lilley, Anttila, and Ruokolainen (2018).

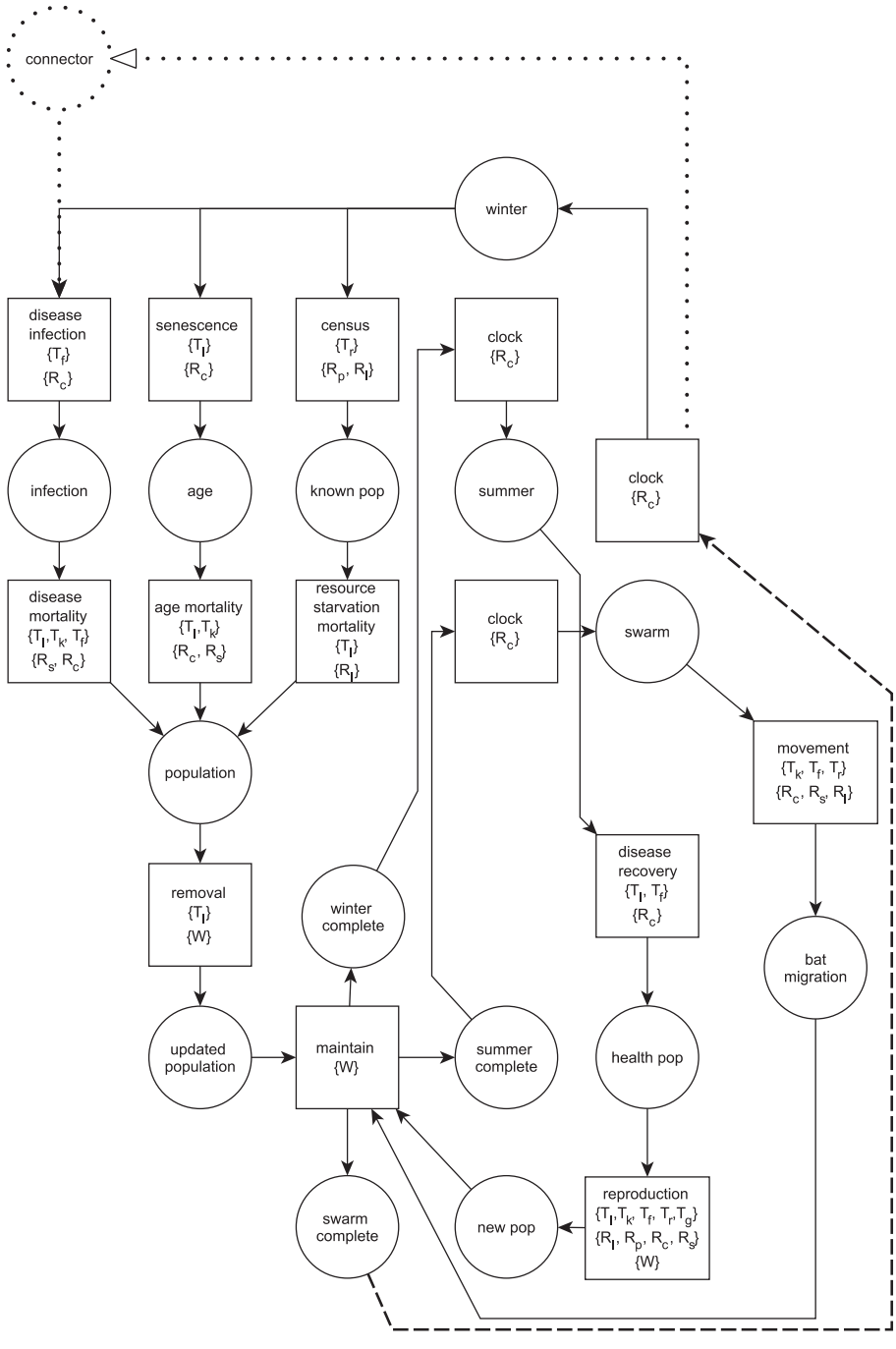


**Figure 1.** Geoexpression of Host's Processes: States are drawn as circles. Transition nodes are drawn as squares. The sets of T, R, and W represent the tokens of data used by the transitions. T is Component data; R is globally accessible attribute data; W is a world update function that allows entities to be added or removed.

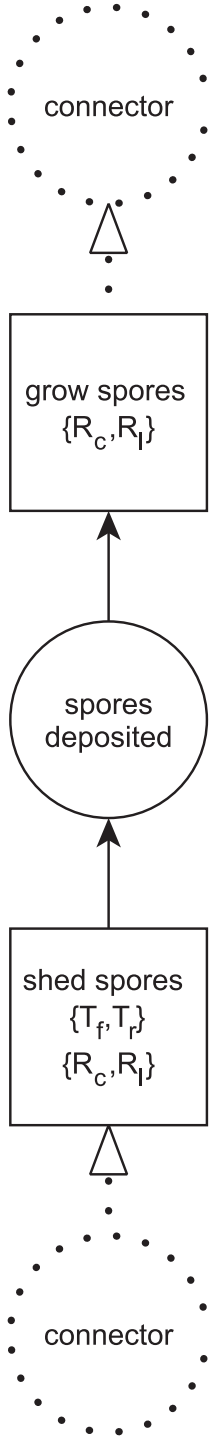


**Figure 2.** Geoexpression of Pathogen's Processes: A depiction of the geoexpression representing the simple life-cycle of the pathogen, Pd.

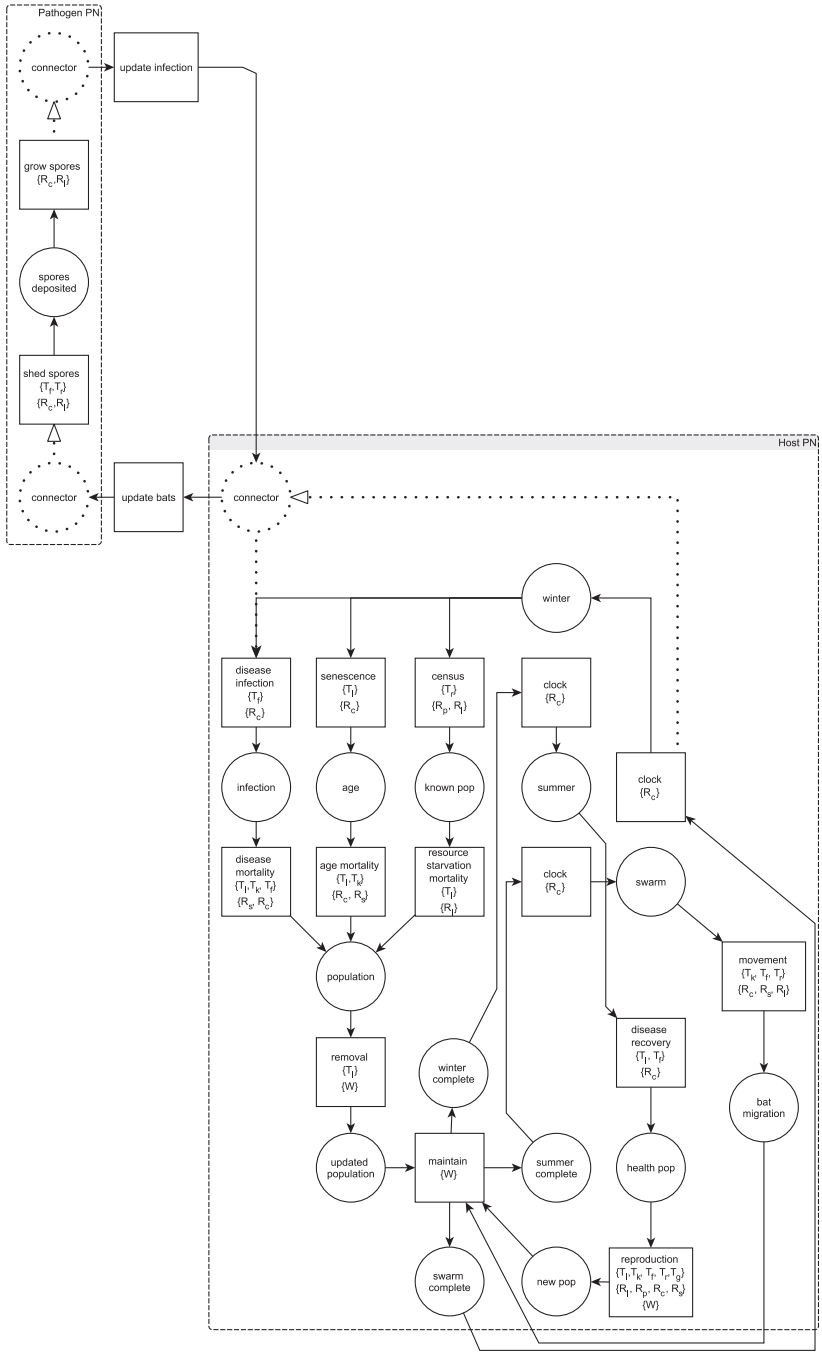


**Figure 3.** Connector Petri-Network between Host and Pathogen Processes: A depiction of how the Host's and Pathogen's geoexpressions can be linked through the use of a connector.

A function that approximates the distribution of pathogen pores.

$$F_{ij} = \beta_0 + \beta_1 \frac{d_{ij}}{\left(n_i n_j\right)} + \beta_2 \tag{1}$$

**Table 2.** Input Configuration File

| Configuration attribute | Data type | Purpose |
|---|---|---|
| Dispatch | Boolean | If true, use a concurrent execution pattern; otherwise, use a sequential execution pattern |
| Max_pop | Unsigned 32-bit Integer | The maximum population size that the model can represent |
| Years | Unsigned 32-bit Integer | The number of model years to run the model |
| Seed | Unsigned 64-bit Integer | The seed value for the ISAAC random number generator |
| Roost_csv | String | The input file containing the roost site parameters |
| Bats_csv | String | The input file containing individual parameters of the initial bat population |
| Species_csv | String | The input file containing species parameters |
| Output_directory | String | The output directory to store the model's output files |

Our IBM represents the geographical transmission of the disease using an emergent strategy. Individual bats within the model interact with the roost sites and the pathogen to spread WNS across the landscape. The inter-hibernacula distribution of spores by bats can be approximated for use here, using equation (1), as demonstrated by Maher et al. (2012). $F_{ij}$ represents the volume of *Pd* spores transmitted between two hibernacula, $n_i$ and $n_j$, over distance, $d_{ij}$. $\beta_0$ represents a constant spread rate. $\beta_1$ represents the distance between pairs of hibernacula. $\beta_2$, a negative term, represents the effect of temperature on the spread of *Pd*. Within the IBM, $F_{ij}$ is always greater than or equal to 0. Figure 3 displays a Connector Petri Network structure that unites the host and pathogen process expressions. Connector Petri networks were introduced by Pouyan and Reeves (2004) and are intended to capture the bi-directional interaction between multiple Petri networks. As WNS is spreading geographically, the processes that comprise the geographical context suitably connect the pathogens to the hosts to spread WNS. In the IBM, the total Petri Network structure, inclusive of the Connector Petri Networks, is used as a blueprint for model implementation based on the parallel ECS.

**Software**

The software accepts a single input configuration file formatted using Tom's Obvious Minimal Language (Preston-Werner 2019). The information contained in the input configuration file is given in Table 2. The input configuration file allows users to specify a set of controls for how the software should operate the IBM. Using the configuration, the user can set whether or not concurrent execution is preferred, what the maximum population of bats the IBM can support is, how many model-years to run, the seed value for the random number generator to make repeatable runs, what set of input data to use, and where to store the model's outputs. The IBM is constructed using roost, bat, and species parameter data provided as a series of input files specified in a Comma Separated Value (CSV) format. The roost CSV file contains the parameters for every roost represented within the IBM. The bat CSV file contains the parameters for each

**Table 3.** Input Configuration Attributes

| Configuration type | Attribute | Purpose |
|---|---|---|
| Bat | Species | The name of the bat species |
| Bat | Gender | The gender of the bat as male (M) or female (F) |
| Bat | Age | The initial age of the bat in years |
| Bat | Initial infection state | The initial infection state of the bat; true if infected, false if not infected |
| Bat | Roost Id | The roost id the bat currently occupies |
| Roost | Latitude | The roost site's latitude |
| Roost | Longitude | The roost site's longitude |
| Roost | FIPS | The federal information processing standard county unique identifier |
| Roost | Maximum population | The maximum population the roost site can sustain |
| Roost | Initial infection state | The initial infectious state of the roost; true if infected, false if not infected |
| Roost | Mean temperature | Annual mean temperature. If the mean temperature is too warm, then *Pd* cannot grow |
| Roost | Visitation multiplier | A value assigned to a given roost to make it more or less likely that a given roost will be visited by bats. It defaults to 1 in all of the models executed in this work |
| Species | Species name | The name of the species |
| Species | Maximum age | The maximum age these bats species can live |
| Species | Fecundity | The probability that females of this species will be successful in reproduction |
| Species | Migration distance | The maximum migration distance of bats of this species |
| Species | Infection rate | The infection probability of bats of this species when exposed to *Pd* |
| Species | Fidelity | The probability of winter hibernacula fidelity of this bat species |
| Species | Visitation rate | The probability of adjacent roost sites that will be visited during swarm |
| Species | Disease mortality rate | The probability at which bats of this species will die as a result of WNS |

individual bat that is used to represent the initial population of the IBM. The species CSV file contains species parameters for the bat population. A summary of the fields of the input CSV files is listed in Table 3.

Our IBM software constructs the model using a parallel ECS architectural pattern. Each entity represents a single individual bat. Each bat entity is associated with its unique set of Components as listed in Table 4. The Components are stored in contiguous memory arrays to reduce the cache miss rate when the parallel ECS operates on Component data. The biological and

**Table 4.** Bat Entity Components

| Component name | Purpose |
|---|---|
| SpeciesKey | An identifier for the species of the bat entity. |
| Gender | The gender of the bat entity as male or female |
| Infected | A state machine to track whether or not the bat entity has been exposed to *Pd*, infected by WNS, or recovered from WNS |
| Lifecycle | Tracks the current age of the bat entity and whether or not it is a juvenile or mature |
| RoostID | The id of the current roost site the bat entity occupies |

pathogenic processes enumerated within the IBM are instantiated as Systems using the SPECS library. The Roost and Species data are common to all entities and are stored as Resources within the parallel ECS.

O'Regan et al. (2015) use a graph representation of the roost site landscape that uses a gravity dispersal kernel to weight the probability of transmission between counties based on spatial heterogeneity. In our model, the edges between roost sites are built using the maximum migration distance specified in the input species configuration file. Our model diverges from the approach by O'Regan et al. (2015) to enable the spread of WNS to be observed in the model as emergent phenomena. Each of the roost sites are stored as a node within a graph data structure provided by the Petgraph library (Sverdrup 2019). Petgraph is a Rust library that contains graph data structures. The edges of the roost site graph are established if the haversine distance between a pair of roost sites is less than or equal to the maximum migration distance attribute supplied in the species input configuration file. The roost site graph is added as the Landscape Resource to the parallel ECS.

Our model requires a few other Resources for the parallel ECS. First, a Clock that tracks the current season (Winter, Summer, Swarm) and year the model is currently in. Second, a species lookup table, indexed by SpeciesKey that is used to store species parameters provided by the species configuration file. Third, a counter for the total population of bats currently active in the model.

The Systems of the parallel ECS reflect the processes that act to spread WNS. The processes correspond to transition nodes of *geoexpression,* while the data types that each transition uses are represented as Petri network tokens. The explicit demarcation of mutability and immutability is a central feature of Rust programming. Each System of the parallel ECS is granted mutable (Write) or immutable (Read) access to the Component or Resource data necessary for its operation. As a good practice, any given System of the parallel ECS should not require access to every Component; the benefits rendered by the parallel ECS approach are maximized by accessing subsets of Components.

The list of Systems instantiated within the model is listed in Table 5. Each of the Systems, except the Census System, corresponds to transition nodes of *geoexpression* used to represent the concurrency within the IBM. The Census System does not correspond to any transition node because it does not serve any biological or pathogenic process within the IBM. Instead, the Census System is called after the Clock updates to Winter to log how many bats are located at each roost site.

**Table 5.** Systems that Contain the Logic of the WNS Spread Model

| System name | Corresponding transition node | Purpose |
|---|---|---|
| Census system | – | Counts the number of bats in each roost site and updates the total population count resource |
| Disease infection system | Disease infection | Determines if a bat will become infected with WNS after being exposed to *Pd* |
| Senesence system | Senescence | Increments the ages of all bat by 1 year |
| Age mortality system | Age mortality | Determines if a bat will die as a function of its current age |
| Resource starvation system | Resource starvation mortality | Determines if a bat will die as a result of overpopulation at a given roost site |
| Disease mortality system | Disease mortality | Determines if a bat will die as a result of WNS infection |
| Removal system | Removal | Removes entities that have been killed by the model |
| Movement system | Movement | The movement system tracks the movement of bats to new winter hibernacula depending upon the species roost fidelity |
| Recovery system | Disease recovery | If bats survive a winter infection, the recovery system clears them of the infection |
| Reproduction system | Reproduction | Produces new entities into the model during Summer based on the fecundity of females of each species |
| Shed spore system | Shed spores | If a bat entity has been exposed to *Pd* during Swarm, it will shed spores of *Pd* into its Winter hibernacula |
| Grow pathogen system | Grow spores | If a roost site has been exposed to *Pd*, it will become infectious the following year |
| Clock system | Clock | Updates the current season and year |

**Model execution**

In this study, we evaluate the scalability and tractability of our IBM implemented using a parallel ECS architectural pattern for a range of roost counts and bat population sizes to represent the spread of WNS. Measurements are made of the wall clock execution time. The SPECS library provides easy parallelism on shared memory machines, multiple component storage structures, and documented high performance in real-world applications (*Specs*, 2019). SPECS enables the scheduling of systems to be defined sequentially or through the dispatch of an execution hierarchy. Repeatable stochasticity within the model is provided through the use of the ISAAC random number generator (Jenkins Jr. 1993) provided by the rand_isaac library (rand_isaac, 2018) authored in Rust. SPECS also supports the exploitation of data parallelism.

## Results

### Data

To evaluate the computational performance of our IBM we prepared a set of input roost site data, data for an initial bat population, and species data. To generate the roost sites on the landscape, we filtered Karst Map Project data (The National Cave and Karst Research Institute 2002) to produce a shapefile of cave bearing surficial geology, which are regions likely to support cave roosting bats. We bounded our input area to the contiguous United States. Regions without caves may still have bat roosts, therefore, the distances from counties without surficial karst features were calculated using the NNJoin plugin in QGIS (QGIS Development Team, 2018). A normalized inverse distance was computed based on the result using the QGIS Field Calculator. Cave counts aggregated by county were extracted from O'Regan et al.'s (2015) Supplementary Data are used to compute a roost count heuristic for each county within the contiguous United States. A set of random locations for each county was computed based on the value of its roost count heuristic using the Random Points tool in QGIS. The mean annual temperature was computed by averaging the mean monthly temperatures provided by Worldclim (Hijmans et al. 2005) on a 5 arc-minute grid, which is a spatial resolution sufficient for determining mean annual temperature as a surrogate for cave temperature as reported in the work of Ihlo and Baker (2013). The Sample Raster Tool in QGIS was used to extract mean annual temperatures from the mean annual temperature raster for each roost site location. The mean annual temperature is used as a proxy for cave temperature following the approach of Ihlo and Baker (2013). Roost sites outside of the mean annual temperature data area were removed from further consideration. The resulting set of roost sites was stored as a shapefile for use by subsequent R scripts to further prepare the input data.

Once the roost site locations, shown in Figure 4, were generated, the generate-roost-sites-from-shp.R R script is used to generate the input roost site data file for the IBM. The R (R Core Team, 2018) script sets the maximum population capacity of each roost site and its initial infection state. The initial infection state is set by passing the script with a set of county Federal Information Processing Standards (FIPS) codes. All of the roost sites in the counties specified will be initially infected. The maximum population capacity of each roost site is estimated by sampling over a uniform distribution of 10 to 1,000 times the cave count in the roost site's county to represent the known preference for bats roosting in cave bearing surficial geology. The script outputs the roost CSV data file to be used as an input to our IBM.

The initial bat population is constructed using the entity-spawner.R R script. The number of bats in the initial population is provided as an argument to the script. For each bat, the script determines each bat's species, age, roost site, gender, and infection state. None of the bats are set to be initially infected with WNS. The script outputs the bat CSV data file to be used as an input in our IBM.

The initial species data is created manually in a text editor and stored in CSV format to be provided as the species data input for the IBM. Species parameters were approximated from the available literature. The visitation_rate parameter is difficult to parameterize from the literature and will be used in future studies to evaluate the effect of inter-roost migration patterns on the spread of WNS. Because the purpose of this investigation focuses on computational performance evaluation, the visitation probability is arbitrarily set to 0.1 for all species; meaning that a given individual bat can visit up to 10% of its neighboring roost sites during swarm.

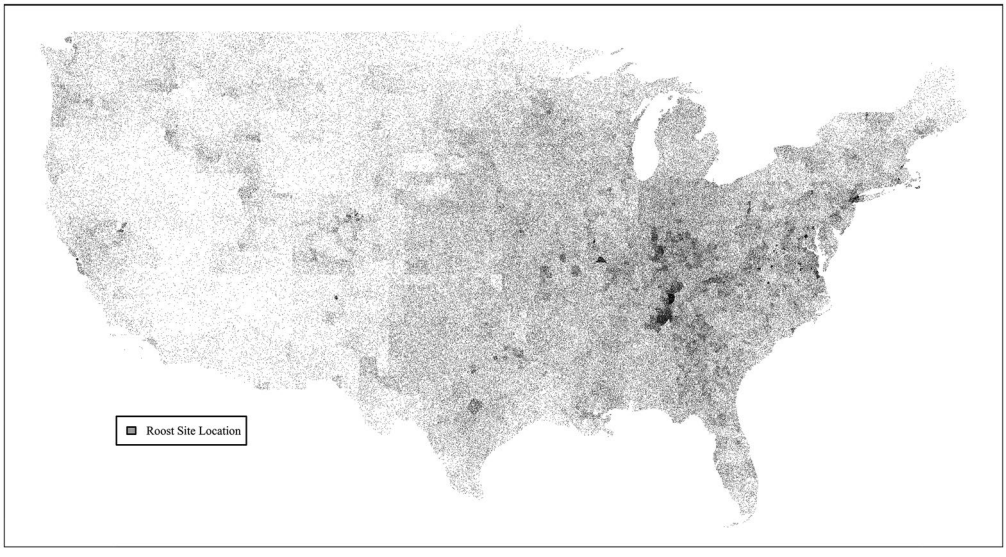


**Figure 4.** Generated roost locations: a depiction of the distribution of generated roost site locations that represent the landscape within the IBM.

**Computational performance evaluation**

A set of experiments were conducted to evaluate the performance of the software. The experiments demonstrate the speedup and efficiency of the parallel ECS IBM, implemented on a foundation of *geoexpression*. All tests were executed using an Intel i5-8350U processor with 16 gigabytes of RAM. Each test is executed with the parallel ECS in parallel dispatch mode. We evaluated two population size conditions, 10,000 and 100,000 bats, for each of three roost size conditions, 10, 100, 1,000 roosts. For each test, the IBM was provided a thread count representing the number of threads to balance the workload of the IBM. We replicated the execution of the model 30 times for each set of input conditions and thread count and used the meantime for evaluation.

Relative Scaled Speedup Formula

$$\mathrm{RSS} = \frac{100 \times \frac{t_1}{t_n}}{n} \tag{2}$$

The Relative Scaled Speedup metric measures how the execution time is affected by an increasing number of cores for a fixed problem size and is computed as shown in equation (2): where $t_1$ is the execution time on a single thread, $n$ is the number of threads, and $t_n$ is the execution time for $n$ threads (Amdahl 1967).

The results of the parallel scalability testing are shown in Figure 5. As the thread count is increased, the IBM execution time speeds up relative to the single thread speed until eight threads. Worse performance is observed beyond eight threads because the hardware used to evaluate the model has a single quad-core architecture that supports two threads per core. Efficiencies degrade when the number of threads exceeds 8, because the processor must actively park and switch between threads.

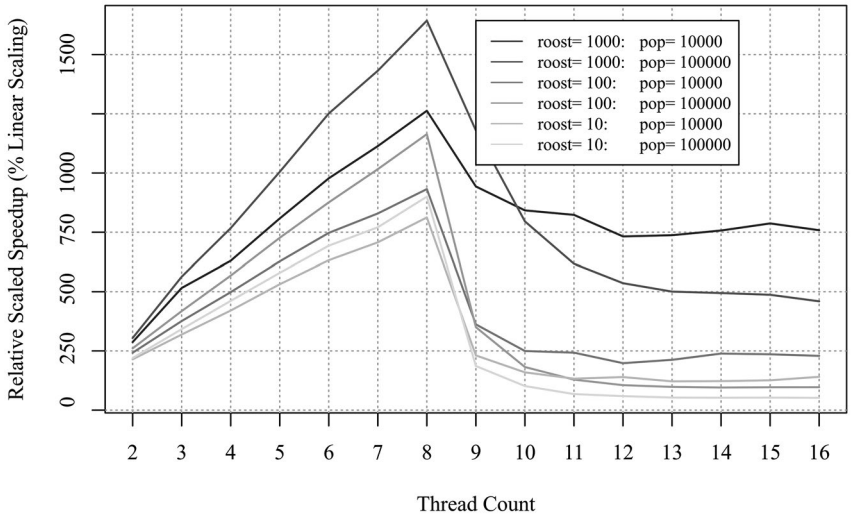


**Figure 5.** Relative Scaled Speedup: a depiction of the scalability for different problem sizes and thread counts.

Wang and Armstrong (2009) present a theoretical underpinning for relating computational intensity to spatial characteristics of data and computation. The theoretical foundation of *geo-expression* is based on Petri nets. Timing each execution of each transition node for a specific volume of tokens can establish a relationship of computational intensity. Additionally, the spatial distribution of where computation is occurring varies as the infection spreads across the set of roost sites.

To evaluate the computational intensity of the IBM, another test was established using a subset of the generated roost sites taken from the problem size experiments and arbitrarily down-selected to Susquehanna, Pennsylvania. A separate branch of the code was created called intensity-tracking that produces the runtime of each System's execution. The maximum population of these roost sites was increased to 1,000 and the migration distance of the species was reduced to 20 km and the visitation probability was increased to 0.6. The model was executed for 15 years with a single roost being initially infected with *Pd*. These changes were made to slow the propagation of WNS to demonstrate the relationship of each System's execution time to the model year, population size, and the number of infected roost sites to different dimensions of the problem within a smaller spatial region. The IBM scales near linearly as population size is increased, exponentially as roost counts are increased, and logistically as species migration distance is increased. The compute-plots.R script was used to graphically depict the relationships between each System's execution time and the problem size it operated over. Figure 6 depicts the relationship between each System's compute time and the model year. As the model progresses from year to year the computational intensity is affected by different population levels and the infectious conditions of roost sites. For example, as *Pd* is spread by bats to new roosts the number of infected roosts increases and the amount of time for the Shed Spore System to infect an uninfected roost site decreases. The Human Movement System operates independently of the population and number of infected roosts and exhibits very little change in its execution time.

Figure 7 depicts the relationship between each System's compute time and the total population size operating within the IBM over the 15 model years. The Removal System assesses

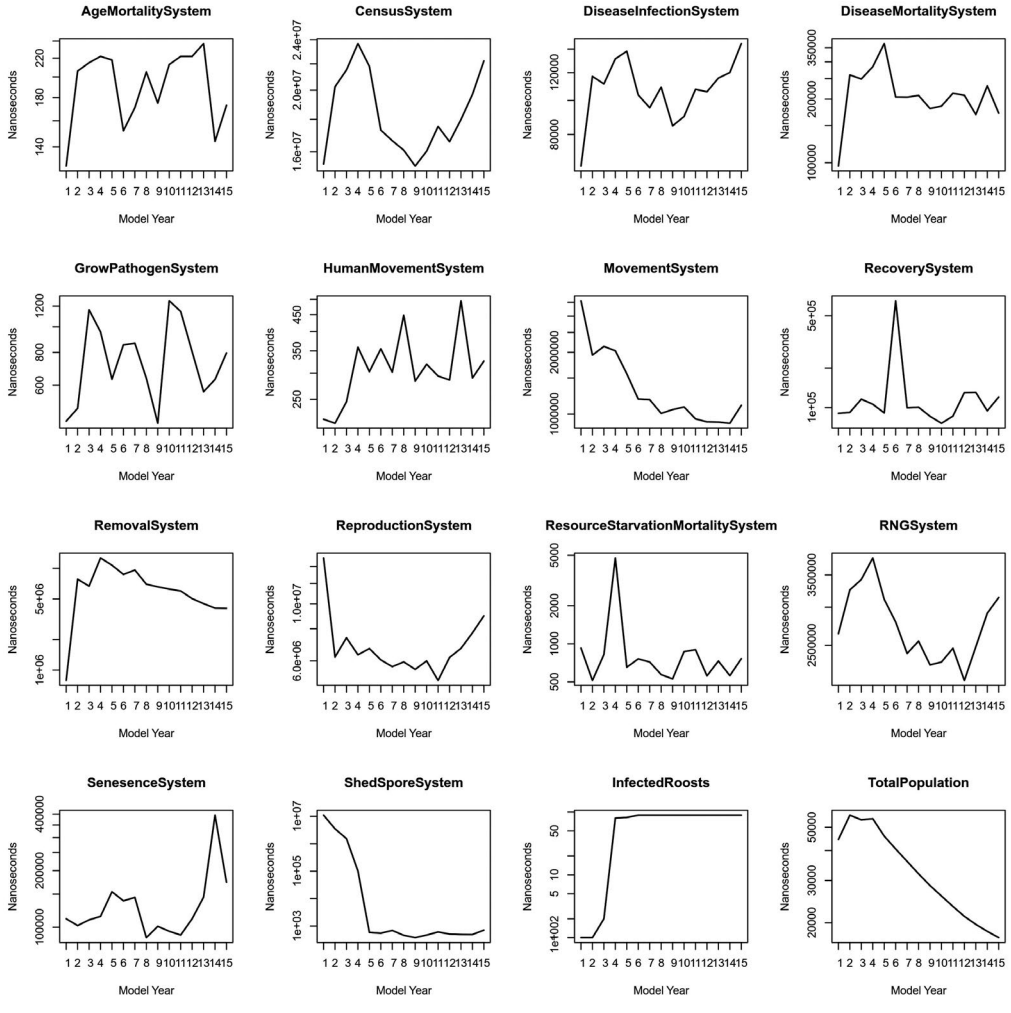


**Figure 6.** System execution time versus model year: a depiction of the variability in system execution time for a given set of model years. The performance of each system is provided to show how each differs in its scalability.

the number of bats that have died, its intensity increases as the total population increases. The Reproduction System and Disease Infection System exhibit similar behaviors.

Figure 8 depicts the relationship between each System's compute time and the total number of infected roost sites. The computational intensity of the Shed Spore System is inversely related to the number of the infected roost site. This is because the System does not operate on a roost site if it is already infected. Therefore, significant computational time is expended when the number of infected roost sites is low and much less for larger numbers of infected roost sites.

### IBM results

Further tests were conducted using generic parameters derived from the literature to demonstrate the efficacy and tractability of our IBM. Our species parameters were adopted from Reynolds

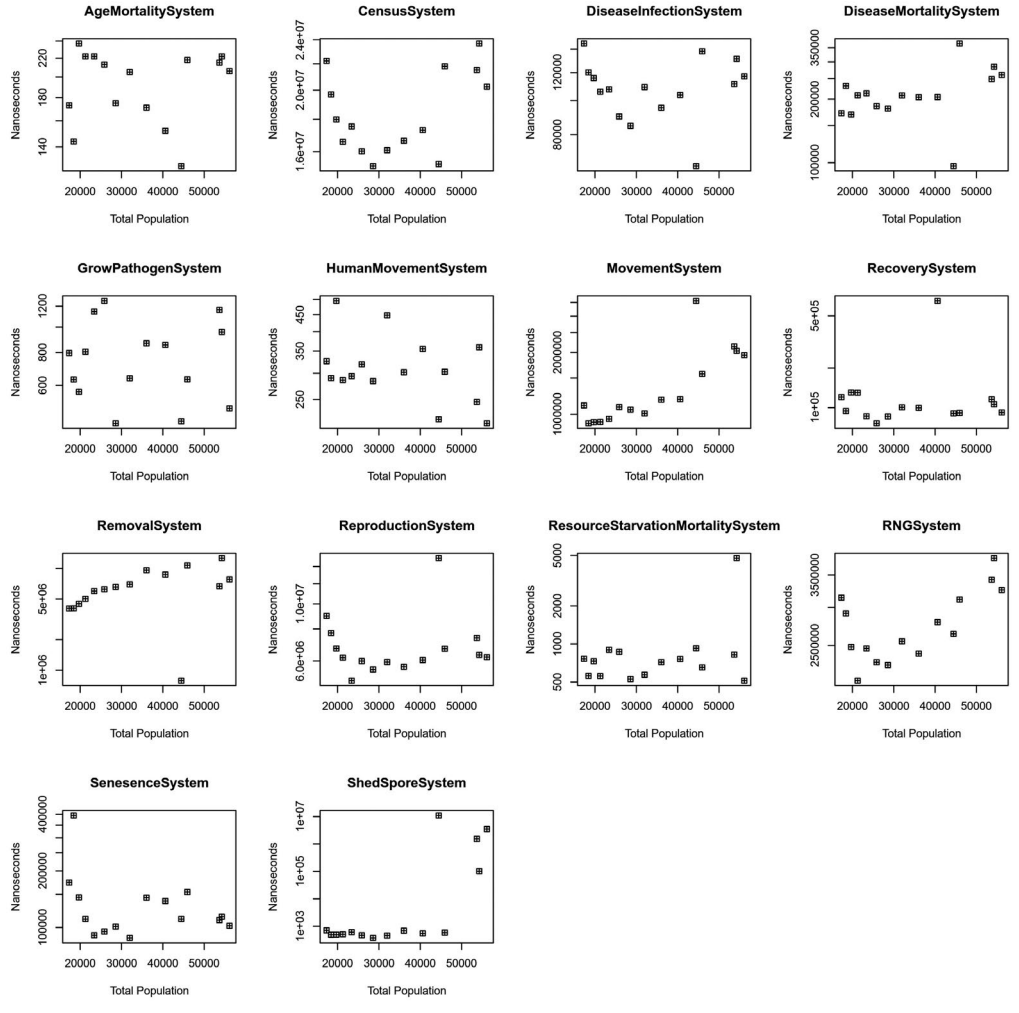


**Figure 7.** System execution time versus total population: a depiction of the variability in system execution time as total population changes. The performance of each system is provided to show how each differs in its scalability.

(2010); Fenton and Barclay (1980); Lilley, Anttila, and Ruokolainen (2018), and the visitation probability was estimated to be 0.8. A subset of 40,000 roosts was selected to serve as the roost sites of the landscape. A population of 1,000,000 bats was generated to initially populate the landscape. Once the input files were prepared the model was executed for 6 model-years.

Figure 9 shows the modeled potential spread of WNS by the initial year it occurred in the county. Figure 10 shows the increasing number of roost sites infected with *Pd* and the population decline in the initial county (Schoharie County, NY). Admittedly, these are straightforward results not designed to demonstrate the robustness of a well-parameterized model. However, the results do demonstrate that a WNS spread IBM implemented using a parallel ECS is entirely tractable. The total execution time of the single run was approximately 30 minutes on commodity laptop hardware.

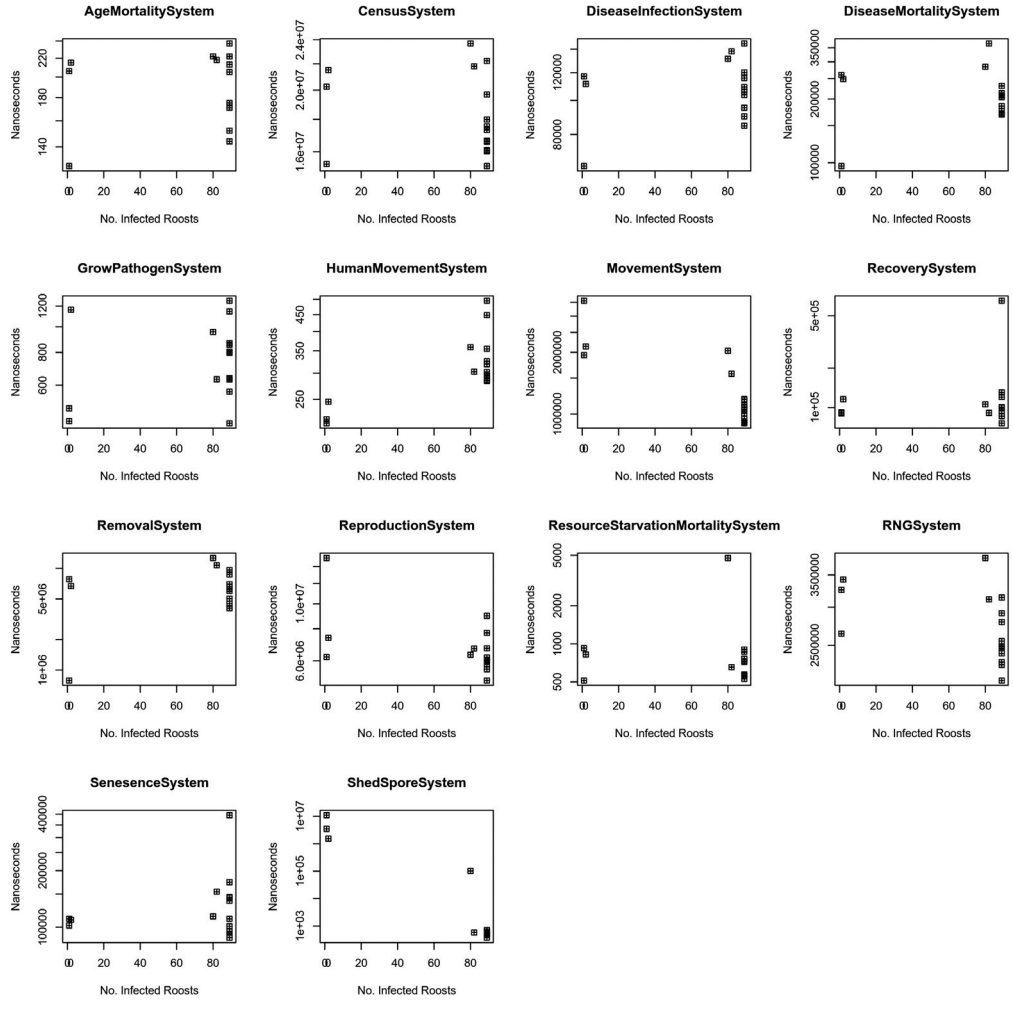


**Figure 8.** System execution time versus number of infected roosts: a depiction of the variability in system execution time as the number of infected roosts changes. The performance of each system is provided to show how each differs in its scalability.

### Extending the IBM through the manipulation of geoexpression

Here we revisit the case of human transmission of WNS and demonstrate how the IBM can be extended and adapted to other situations. The use of *geoexpression* as a framework for structuring the model allows different component orderings to be evaluated to repeatedly assess the effects of non-determinism in the real-world spread of WNS. Human transmission is less seasonally dependent compared to bats (humans do not hibernate) so we construct two *geoexpressions* that capture human transmission: one where humans transmit the disease during winter as shown in Figure 11; and one where humans transmit the disease during summer as shown in Figure 12.

The *geoexpressions* guide the modification of the model's code. Specifically, a new Resource, System, and extension to the configuration are added to represent the human population size and movement probability within the model. The Human Movement Resource stores

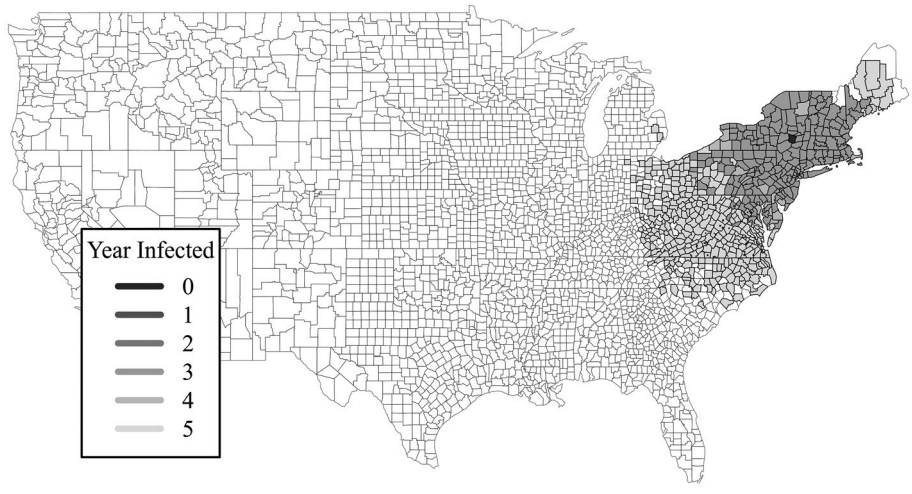


**Figure 9.** Modeled spread of WNS by year: the initial year that White-Nose Syndrome is detected summarized by county.

the size of the human population that moves between random pairs of roost sites. The additional parameter added to the configuration is the total human population size, where each human case serves as a transmission vector. When called, the Human Movement System iteratively selects a source and a destination roost for each human case. If the source roost is infected, the human case will expose the destination roost to the infection. The Human Movement System is then structured using SPECS to reflect the concurrency structure of the winter and summer *geoexpressions* being evaluated. Separate runs of the model are used to generate results for the winter and summer *geoexpressions,* respectively.

Once the IBM has been extended to support human transmission based on the structure defined in our *geoexpressions* it is executed using the parameters in the IBM Results section. The human population size parameter is set to 10,000. The results of the winter and summer human WNS transmission and transmission of WNS by bats only are shown in Figure 13.

## Discussion

Our IBM was developed through the lens of geographical process concurrency. The *geoexpression* that represents the interactions between disparate geographical processes was developed using the epidemiological triangle as a guide. Transition nodes were denoted with the sets of Component data necessary for its operation. The *geoexpression* represented the concurrency inherent in WNS spread and was used to order the execution of Systems within our parallel ECS-based IBM. The use of *geoexpression* supports the exploitation of parallelism for resolving computational intensity, and the ability to identify the processes that drive model outcomes.

### Geoexpression and parallelism

Our IBM exhibited increased run-time as the number of threads was increased until the computational hardware was fully utilized with eight threads. The use of ECS architectural patterns for IBM construction led to more efficient use of the computer's memory access. Our results demonstrate that a reasonably large population of individual entities can be represented using

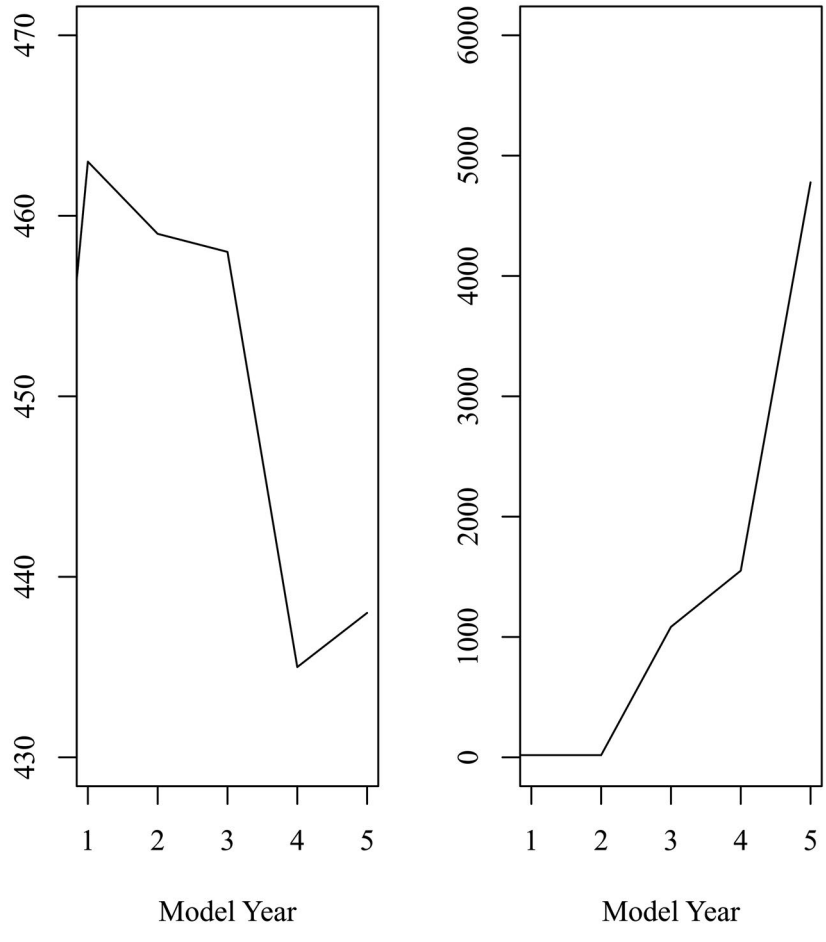


**Figure 10.** (Left) population declines in schoharie county, N.Y. (right) Increase in infectious roost sites by year: a depiction of the population declines in a county exposed to WNS and also how the number of roost sites infected increases.

commodity hardware. Admittedly the computational complexity of the algorithms instantiated using a specific programming language ultimately limit the tractability of a problem. In our IBM, the pairwise distances between each bat, an O($n$ log $n$) process, are not computed on demand. Instead, the pairwise distances between roost sites are initially computed and then stored within a graph data structure to quickly look up "landscape" distances for each individual bat's migration. This approach is reasonable because disease transmission is understood to largely occur within the roost regardless of whether or not the mode of transmission is bat-to-bat or bat-to-substrate; the effect of spreading the pathogen is the same. The ECS is known to have greater efficacy when entities are of different types (represented using different sets of Components); however, our IBM does not (yet) distinguish different classes of entities. All of the entities within the IBM are bats.

While *geoexpression* ordered the execution of Systems, data-parallel operations on Component data within Systems were also exploited. For example, the Senescence System increments the Component representing the age of every bat (entity) by time. This operation is data parallel, because, at the same time step within the IBM, the age of one bat has no bearing on the

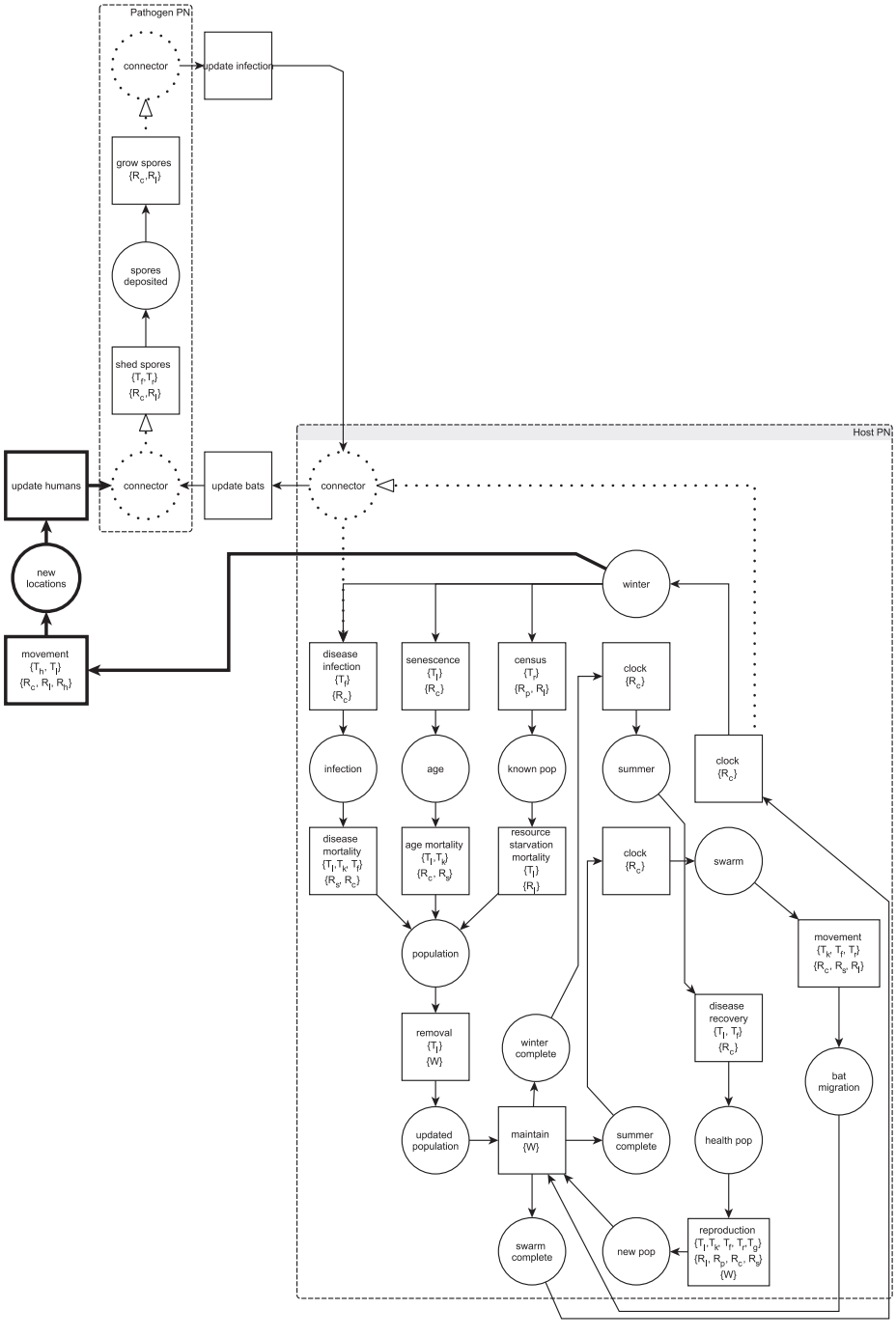


**Figure 11.** Human Winter Transmission: A geoexpression of WNS spread adding human transmission during winter.

age of another. The senescence operation can then be distributed across multiple idling threads. Therefore, our IBM exploits geographical process concurrency through *geoexpression* and data parallelism.

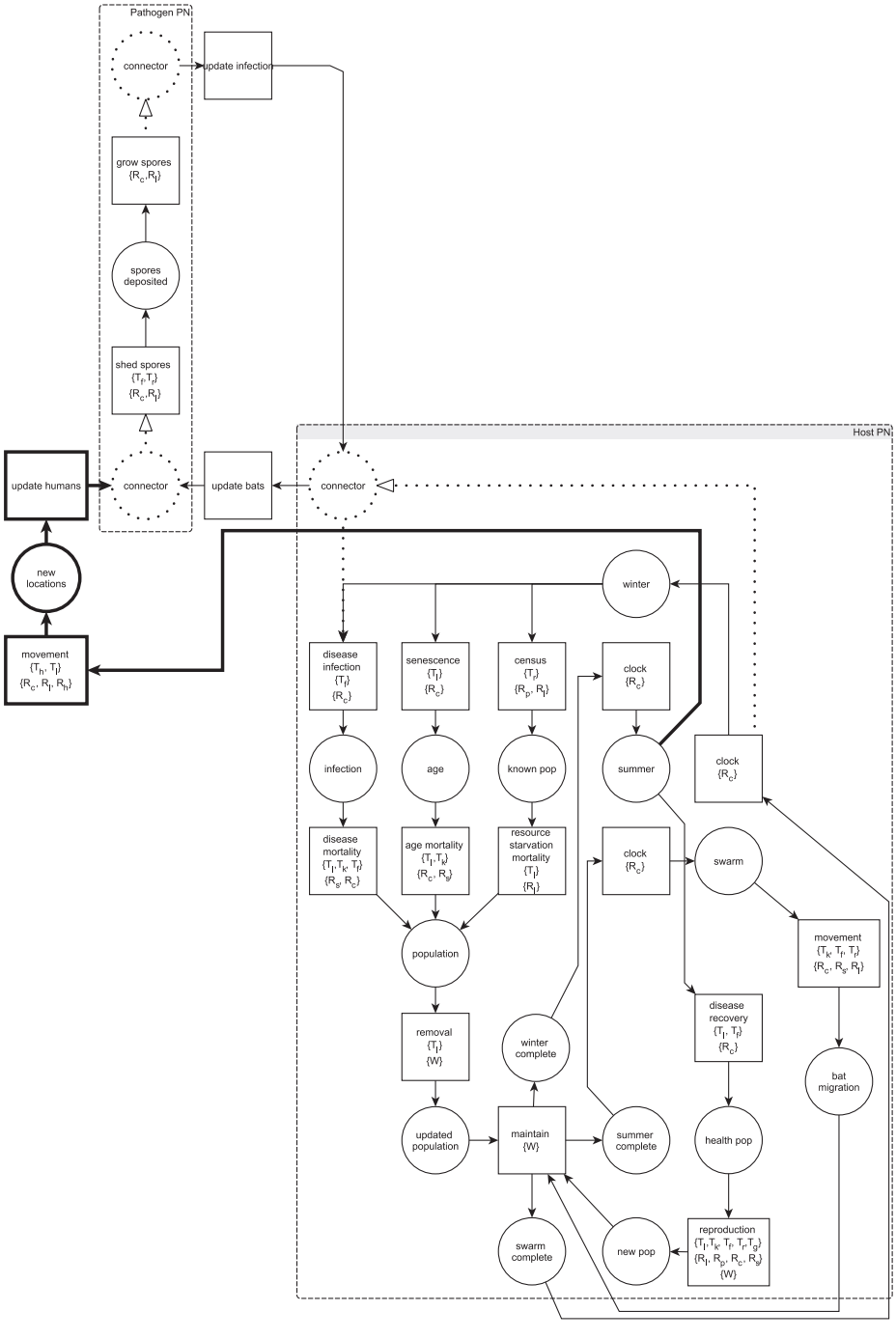


**Figure 12.** Human Summer Transmission: A geoexpression of WNS spread adding human transmission during summer.

### Geoexpression, concurrency, and computational intensity

As introduced by Wang and Armstrong (2009) the computational intensity can be unevenly distributed across a spatial domain. In our IBM, computational intensity varies across the locations

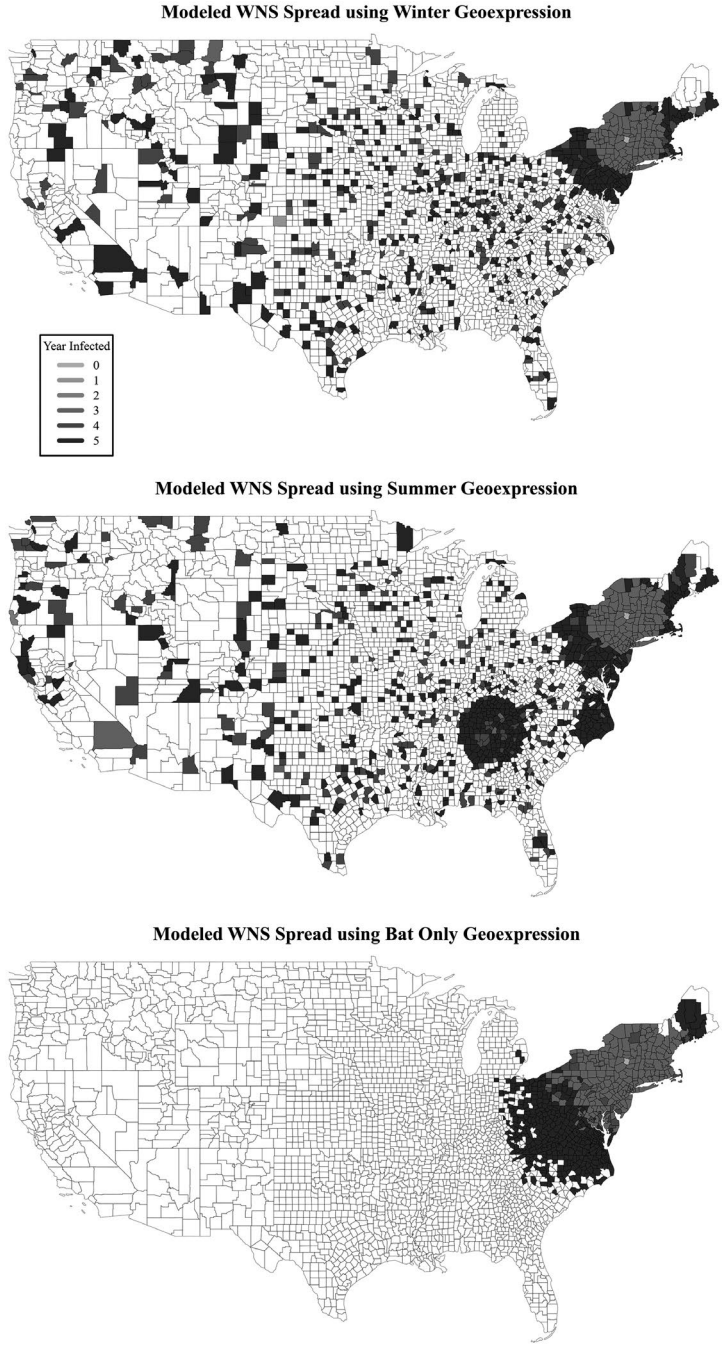


**Figure 13.** Outcomes of different geoexpressions: (top) initial infection year aggregated by county resulting from the addition of human transmission during the winter; (middle) initial infection year aggregated by county resulting from the addition of human transmission during the summer; (bottom) initial infection year aggregated by county for bat only transmission.

of roost sites depending upon the population at the roost site, the number of infected bats within the population, and the current infectious state of the roost site. The Systems of the ECS that represent the transition nodes of *geoexpression* are applied differently depending upon these conditions at each location. In Figure 9, the spread of WNS geographically coincides with the Disease Infection System, Shed Spore System, and Grow Pathogen System activating and deactivating based on local conditions. Additionally, the Movement System is dependent upon the number of spatially near roost sites and processes greater amounts of data in areas with higher roost density.

The IBM's Systems are concurrently executed based on an order guided by the *geoexpression* to produce the WNS spread outcome. The implementation of the inherent concurrency causes contention for specific roost sites and bat entities depending upon their infectious state, location, and the current model year. The *geoexpression* theory makes it possible to trace individual entities through the Petri network and identify which transition nodes operate on it each model year, based upon its spatial location, and internal and external parameters. Therefore, throughout the duration of a model's execution, the use of *geoexpression* enables tracing how computational intensity changes for specific entities and how different processes contend to concurrently apply various transformations to the entity's data.

**Geoexpression and ecological prediction**

Frequently, ecologists are asked to make predictions about the potential effects that specific environmental changes have on a specific ecosystem or landscape. Understanding the consequences of each change in the community dynamics of an ecosystem has proven to be challenging. When models implemented using ordinary differential equations, assessments of potential outcomes resulting from specific changes are somewhat manageable to determine. However, when IBMs are used, the assessment of potential outcomes resulting from specific changes has proven to be a daunting task because the methodology of IBMs can represent significantly greater details and emergent phenomena (Colon, Claessen, and Ghil 2015). This study uses *geoexpression* to provide a graphical representation of the processes and their interrelationships. Using *geoexpression* we identify how frequently specific transition nodes of *geoexpression* are activated and the computational intensity of each process. This leads to the comparable understanding of the effects each process has on the model's outcomes without needing a separate model based on ordinary differential equations.

**WNS spread IBM**

Computationally tractable and high-performance epidemiological models that can simulate large populations are important to understand how different diseases spread spatially. Wildlife epidemiology serves as an important subject of inquiry to better understand, by corollary, disease transmission in human populations and disease transmission between wildlife and humans. While our approach focuses on the WNS epidemic, human epidemics can readily benefit from *geoexpression*-based IBM methodology.

Following Lilley, Anttila, and Ruokolainen (2018), our model does not capture relative humidity due to its high variability within individual cave sites; however, it is an important climatic parameter for the growth of *Pd*. Our IBM currently abstracts over the volume of spores a bat can carry (a parameter of the O'Regan et al. (2015) model). In the future, this addition could improve our representation of how spores are spread, and the concentration level necessary to become viable at a new roost site.While prior studies use a daily time step, our IBM operates on a triannual time step based on bat function. We chose this temporal granularity because it represents

the biological phases of bat activities where disease response efforts and treatment plans could be implemented. While treatment is not a current component of our IBM, we expect to incorporate treatment in future efforts. Our IBM requires further sensitivity testing of its spatial and non-spatial parameters. With comprehensive sensitivity testing, proper parameterization of the model to represent the dynamics of WNS spread can be completed. Properly parameterized, our approach can be used to predict the spread of WNS into the future following the work of Maher et al. (2012); O'Regan et al. (2015); Ihlo and Baker (2013). Notwithstanding the lack of parameterization of the model and a more rigorous validation effort (to be pursued in future work), the results provided by our IBM are comparable to those produced in the aforecited literature.

**Model validation**

The development of a fully validated WNS model has challenged many previous efforts in the literature. Prior models with validation methods have ultimately fallen well short of their predictive capabilities. The goal of this specific study has been the implementation of a model built on *geoexpression*, based on modeling information found in published work. We readily admit that to apply our WNS IBM will require more rigorous validation efforts and extensive parameterization, especially as our model relates to the roost-site landscape network. To tackle the validation challenge, we have focused on a critical metric our model provides, the average distance of WNS spread per year, against the average historical distance of WNS spread captured by the WNS disease surveillance data disseminated by the WNS Response Team (White-Nose Syndrome Response Team 2018). The specific measure is the set of distances, as measured between Schoharie County, NY (where WNS began in North America), and the counties that became infected each year, correlated with the same distance measures computed from the WNS Response Team surveillance data. Our bat-only *geoexpression* captures 60% of the observed WNS average distance spread per year. The *geoexpressions* that include human summer and winter transmission capture 70% and 84% of the observed WNS average distance spread per year, respectively. While these results may seem low at first glance, they are not indicative of artificial geographical constraints that have been imposed by others. Existing studies impose geographical restrictions to only those counties with known caves (Lilley, Anttila, and Ruokolainen 2018), while our study endeavors to capture a broader landscape. Furthermore, our approach shows how different patterns of geographical concurrency can lead to significantly different modeling results.

**Rust in environmental modeling**

Many have remarked on the steepness of Rust's learning curve (Blandy 2015). Much of the difficulty revolves around developing a thorough understanding of Rust's ownership model. Leaders within the Rust community of practice have worked to make many pain points of the language much less painful and have future improvements planned to reduce the steepness of Rust's learning curve (Blandy 2015). Overall, we found Rust and the SPECS library to be approachable; albeit more difficult to learn than the less stringently typed languages Java, R, and Python. The execution time of interpreted languages like R and Python is a noted shortcoming for their use in simulation. Rust has a comparable syntax to higher level languages, but compiles into a binary executable as opposed to being interpreted. Rust's compiler enforces its memory safety guarantees, which means that if the code compiles, in all likelihood, the executable will not be plagued by many classes of errors that are common in C or C++ (Klabnik & Nichols 2019).

**Software accessibility**

The WNS IBM software is maintained in a Bitbucket repository that interested researchers may access and adapt for their own purposes. A set of scripts has been created to support the automatic generation of the input configuration files. The model may be extended to support greater complexity of WNS spread by adding new Systems, Components, and Resources based on a new *geoexpression* and then implement it using the SPECS dispatch ordering rules.

While the software can be used to represent other spreading diseases, it would require significant changes to make it disease-specific. However, the use of *geoexpression* to structure the order of process components in a model and the use of SPECS to implement the execution of the model has flexible applicability to other environmental modeling domains.

As demonstrated in our results, a System can be added to extend the functionality of the WNS IBM. The System can be scheduled for dispatch in a specific order using SPECS. Each System can be named by a researcher. An array of 0 or more System names can be passed to the SPECS dispatcher and associated with a specific System. A System will not be executed until all of the names in the aforementioned array are completed. This approach allows SPECS to support the concurrency graphically represented using *geoexpression*.

## Conclusions

WNS is a serious epidemic. Much prior work has sought to model the spread of the epidemic through statistical, dynamic, and mean-field models. We build on prior studies to develop an IBM from the perspective of geographical process concurrency. Using the epidemiological triangle to guide the process decomposition, we construct Petri network representations of the process interactions. These processes map cleanly onto the ECS architectural pattern through *geoexpression* to order the concurrent execution of Systems. Next, we demonstrate the computational performance of our ECS approach using the SPECS library, authored in Rust, to instantiate the IBM.

Our contributions demonstrate that the use of *geoexpression* enables geographical process models to be developed for enhancing geographical discovery, specifically related to geographical process concurrency. We achieve our objective by implementing an IBM of WNS spread leveraging previous WNS modeling work using an ECS architectural pattern. The processes of our IBM are represented graphically using Petri networks in a way mutually comprehensible by a human and computational machine. Next, we evaluate the performance of our software solution to demonstrate the tractability of our approach and its utility in showing how modifications to a *geoexpression* alter the geographical patterns being produced.

WNS modeling research benefits from *geoexpression* that sheds light on how the life-cycle of *Pd* and bats interact concurrently under a set of environmental conditions to produce the epidemic. Different structures of *geoexpression* used to model the spread of WNS lead to different spatial outcomes, with varying levels of correlation to observed spread, despite the fact that the modeled processes themselves are unchanged. As disease intervention technologies become more mature, species management decisions about where to field these technologies will become more important. Improved spatial modeling, that captures the interplay between concurrent processes can help isolate the mechanisms of spread biologically and geographically to achieve desirable impacts.

While this work relies on commodity computing hardware, our next step is to increase the population size and roost count by leveraging advanced cyberinfrastructure. On advanced

cyberinfrastructure, the same code should become readily tractable for tens of millions of bats over a roost network of a few hundred thousand roosts. The ability to simulate a North American bat population of 50 million bats with a roost network of 250,000+ nodes with well-defined WNS spread parameters to demonstrate the emergent spread of WNS across the landscape would be extremely valuable to tackle related public health challenges. Such a simulation is made possible through the exploitation of geographical process concurrency based on a *geoexpression* framework. While much work remains to be done to develop an IBM that accurately models WNS spread, our *geoexpression* approach implemented on ECS represents a novel contribution to advancing an important branch of geographical analysis—one not limited to spatial epidemiology, but applicable to many situations where geographical process concurrency is present.

## Acknowledgements

This research is supported in part by the U.S. National Science Foundation under grant number: 1743184 and 1824961. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This research is also supported by the United States Army Engineer Research and Development Center Long-Term Training Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Army Engineer Research and Development Center or the United States Government.

## References

Amdahl, G. M. (1967). "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities." In Proceedings of the April 18–20, 1967, Spring Joint Computer Conference on - AFIPS '67, 483. Atlantic City, NJ: ACM Press. https://doi.org/10.1145/1465482.1465560.

Bat Conservation International. (2019). White-Nose Syndrome [WWW Document]. URL http://www.batcon.org/white-nose-syndrome (accessed 18 February 2019).

Blandy, J. (2015). Why Rust? Newton, MAO'Reilly Media Inc.

Blehert, D. S., A. C. Hicks, M. Behr, C. U. Meteyer, B. M. Berlowski-Zier, E. L. Buckles, J. T. H. Coleman, S. R. Darling, A. Gargas, R. Niver, J. C. Okoniewski, R. J. Rudd, and W. B. Stone. (2009). "Bat White-Nose Syndrome: An Emerging Fungal Pathogen?" *Science* 323, 227–7. https://doi.org/10.1126/science.1163874.

Colon, C., D. Claessen, and M. Ghil. (2015). "Bifurcation Analysis of an Agent-Based Model for Predator-Prey Interactions." *Journal of Ecological Modeling* 317, 93–106. https://doi.org/10.1016/j.ecolmodel.2015.09.004.

Danielsson, M., and G. P. Bohlin. (2015). A High Performance Data-Driven, Entity-Component Framework For Game Engines With Focus on Data-Oriented Design. Karlskrona, Swedent: Blekinge Institute of Technology.

Davis, A., and S. Wang. (2018). "Geoexpression: A Petri Network Framework for Representing Geographic Process Concurrency." *Transactions in GIS* 22, 1390–405. https://doi.org/10.1111/tgis.12477.

Dzal, Y., L. P. McGuire, N. Veselka, and M. B. Fenton. (2011). "Going, Going, Gone: The Impact of White-Nose Syndrome on the Summer Activity of the Little Brown Bat (*Myotis lucifugus*)." *Biology Letters* 7, 392–4. https://doi.org/10.1098/rsbl.2010.0859.

Fabian, R. (2013). Data-Oriented Design. Richard Fabian.

Fenton M. B., Barclay R. M. R. (1980). "Myotis lucifugus." *Mammalian Species*, (142), 1–8. https://doi.org/10.2307/3503792

Foley, J., D. Clifford, K. Castle, P. Cryan, and R. S. Ostfeld. (2011). "Investigating and Managing the Rapid Emergence of White-Nose Syndrome, a Novel, Fatal, Infectious Disease of Hibernating Bats." *Conservation Biology* 25, 223–31. https://doi.org/10.1111/j.1523-1739.2010.01638.x.

Goodchild, M. F., M. Yuan, and T. J. Cova. (2007). "Towards a General Theory of Geographic Representation in GIS." *International Journal of Geographical Information Science* 21(3), 239–60.

Hammerson, G. A., M. Kling, M. Harkness, M. Ormes, and B. E. Young. (2017). "Strong Geographic and Temporal Patterns in Conservation Status of North American Bats." *Biological Conservation* 212, 144–52. https://doi.org/10.1016/j.biocon.2017.05.025.

Hijmans, R. J., S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis. (2005). "Very High Resolution Interpolated Climate Surfaces for Global Land Areas." *International Journal of Climatology* 25, 1965–78. https://doi.org/10.1002/joc.1276.

Ihlo, C. M., and D. P. Baker. (2013). Predicting the Spread of White-nose Syndrome in Bats: A strategy for Prioritizing Resources. Durham, NCDuke University.

Jenkins Jr., R. J. (1993). ISAAC and RC4 [WWW Document]. URL http://burtleburtle.net/bob/rand/isaac.html (accessed 13 January 2019).

Klabnik, S., and Nichols, C. (2019). The Rust Programming Language. San Francisco, CA: No Starch Press.

Lilley, T. M., J. Anttila, and L. Ruokolainen. (2018). "Landscape Structure and Ecology Influence the Spread of a Bat Fungal Disease." *Functional Ecology* 32(11), 2483–96. https://doi.org/10.1111/1365-2435.13183.

Lin, T., S. Wang, L. F. Rodríguez, H. Hu, and Y. Liu. (2015). "CyberGIS-enabled Decision Support Platform for Biomass Supply Chain Optimization." *Environmental Modelling and Software* 70, 138–48. https://doi.org/10.1016/j.envsoft.2015.03.018.

Maher, S. P., A. M. Kramer, J. T. Pulliam, M. A. Zokan, S. E. Bowden, H. D. Barton, K. Magori, and J. M. Drake. (2012). "Spread of White-Nose Syndrome on a Network Regulated by Geography and Climate." *Nature Communications* 3, 2301. https://doi.org/10.1038/ncomms2301.

Majerech, F. (2015). A Concurrent Component-based Entity Architecture for Game Development. Prague, CZ: Ustav Informatiky, P. J Safarik University.

O'Regan, S. M., K. Magori, J. T. Pulliam, M. A. Zokan, R. B. Kaul, H. D. Barton, and J. M. Drake. (2015). "Multi-Scale Model of Epidemic Fade-Out: Will Local Extirpation Events Inhibit the Spread of White-Nose Syndrome?" *Ecological Applications* 25, 621–33. https://doi.org/10.1890/14-0417.1.

Peterson, J. L. (1978). "Introduction to Petri Nets." *Proceedings of the National Electronics Conference* 32, 144–8.

Pouyan, A. A., and S. Reeves. (2004). "Behavioral Modeling for Mobile Agent Systems Using Petri Nets." In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583). Presented at the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), 5, 4935–40. https://doi.org/10.1109/ICSMC.2004.1401313.

Preston-Werner, T. (2019). Tom's Obvious, Minimal Language. toml-lang. https://tom.preston-werner.com/

QGIS Development Team. (2018). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.org

R Core Team. (2018). R: A Language and Environment for Statistical Computing. Vienna, AustriaR Foundation for Statistical Computing.

rand_isaac. (2018). The Rust Project Developers. https://crates.io/crates/rand_isaac

Reynolds, A. M. (2010). "On the Origin of Bursts and Heavy Tails in Animal Dynamics." *Physica A: Statistical Mechanics and its Applications* 390, 245–9. https://doi.org/10.1016/j.physa.2010.09.020.

Scull, A. M. (2015). Hephaestus: A Rust Runtime for a Distributed Operating System. Cambridge, UK: University of Cambridge Computer Laboratory.

Shook, E., and S. Wang. (2015). "Investigating the influence of Spatial and Temporal Granularities on Agent-Based Modeling." *Geographical Analysis* 47(4), 321–48. https://doi.org/10.1111/gean.12080.

Shook, E., S. Wang, and W. Tang. (2013). "A Communication-Aware Framework for Parallel Spatially Explicit Agent-Based Models." *International Journal of Geographical Information Science* 27, 2160–81. https://doi.org/10.1080/13658816.2013.771740.

Specs: Parallel ECS. (2019). The Specs Project. https://github.com/slide-rs/specs

Sverdrup, U. (2019). Graph Data Structure Library for Rust. https://github.com/petgraph/petgraph

The National Cave and Karst Research Institute. (2002). The National Karst Map. Carlsbad, NM: The National Cave and Karst research Institute.

Thogmartin, W. E., R. A. King, P. C. McKann, J. A. Szymanski, and L. Pruitt. (2012). "Population-Level Impact of White-Nose Syndrome on the Endangered Indiana Bat." *Journal of Mammalogy* 93, 1086–98. https://doi.org/10.1644/11-MAMM-A-355.1.

Trivedi, J., J. Lachapelle, K. J. Vanderwolf, V. Misra, C. K. R. Willis, J. M. Ratcliffe, R. W. Ness, J. B. Anderson, and L. M. Kohn. (2017). "Fungus Causing White-Nose Syndrome in Bats Accumulates Genetic Variability in North America with No Sign of Recombination." *mSphere* 2, e00271-17. https://doi.org/10.1128/mSphereDirect.00271-17.

Wang, S. (2010). "A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis." *Annals of the Association of American Geographers* 100(3), 535–57. https://doi.org/10.1080/00045601003791243.

Wang, S., L. Anselin, B. Bhaduri, C. Crosby, M. F. Goodchild, Y. Liu, and T. L. Nyerges. (2013). "CyberGIS Software: A Synthetic Review and Integration Roadmap." *International Journal of Geographical Information Science* 27, 2122–45. https://doi.org/10.1080/13658816.2013.776049.

Wang, S., and M. P. Armstrong. (2009). "A Theoretical Approach to the Use of Cyberinfrastructure in Geographical Analysis." *International Journal of Geographical Information Science* 23, 169–93. https://doi.org/10.1080/13658810801918509.

White, L. A., J. D. Forester, and M. E. Craft. (2017). "Using Contact Networks to Explore Mechanisms of Parasite Transmission in Wildlife: Contact Networks: Wildlife Parasite Transmission." *Biological Reviews* 92, 389–409. https://doi.org/10.1111/brv.12236.

White-Nose Syndrome Response Team. (2018). White-Nose Syndrome [WWW Document]. URL https://www.whitenosesyndrome.org/ (accessed 10 June 2018).