

High-Performance Computing Tools for Modeling Evolution in Epidemics*

William Maniatty, Boleslaw Szymanski
*Department of Computer Science
Rensselaer Polytechnic Institute
Troy, N.Y. 12180-3590
email: {maniattb, szymab}cs.rpi.edu*

Thomas Caraco
*Department of Biological Sciences
SUNY Albany
Albany, N.Y. 12192
email: caracocsc.albany.edu*

Abstract

This paper describes a series of stepwise refinements of a biological model resulting in a high-performance simulation system for individual-based models of the co-evolutionary dynamics associated with spatially explicit epidemic processes. Our model includes two competing host species, a macroparasite capable of serving as a vector, and the vector-borne microparasite. Genetic algorithms are used to simulate genetic change; we are particularly interested in the evolution of pathogen virulence. The simulation system employs cellular automata to track individual organisms distributed over a two-dimensional lattice. Our models are able to identify each individual's parentage, and to account for both biotic and abiotic spatial heterogeneity. Using the developed system we conducted a series of experiments to demonstrate how individual-based modeling and explicit representation of space, although computationally expensive, can produce qualitatively new biological results.

1 Introduction

Certain parasites evolve to impair their host's survival and reproduction only minimally. The primate lentiviruses offer an interesting example [22]. In older coevolutionary associations, such as simian immunodeficiency virus and African green monkeys, infection is essentially non-pathogenic, while in the recent association between HIV and humans, infection leads to serious disease. However, a long-standing coevolutionary association does not always imply reduced parasite virulence [9]. In fact, virulence exhibits a great deal of variation among host-parasite associations, and often varies temporally within a particular association [14, 25].

Current theory for the evolution of virulence equates virulence with extra host mortality due to parasite infection, and then assumes that selection should increase a parasite's rate of reproduction. As a parasite exploits host resources at a greater rate, it increases the rate of transmission to new hosts. But the host's mortality rate increases as a consequence, decreasing the length of the period during which the parasite can be transmitted. Depending on the functional relationship between parasite transmission rate and virulence (i.e., the trade-off between transmission rate and infectious period), selection may favor low, intermediate or increasing virulence [20, 22].

A number of recent models address variations of the trade-off just described. Important extensions include analyses of

- interactions between virulence and host recovery rate ("clearance" by the immune system [1]);
- competition between different parasite strains infecting the same host individual ("coinfection" [3, 10, 25]);
- competition between parasites when a more virulent strain excludes a less virulent strain infecting the same host ("superinfection" [14, 22, 11]); and
- effects of mutation and relatedness among parasites infecting the same and different hosts [11].

This paper presents a series of computational models, each a stepwise refinement of the previous one, with the final model including all listed above interactions. All of these models are individually-based and spatially explicit.

Ecological simulations show that spatial structure of contacts between infectives and susceptibles strongly influences a disease advance [7, 21]. Limiting infectious contacts to an interaction neighborhood around a diseased host reduces the frequency of infection compared

*This work was supported by NSF grants CCR-9527151 and ASC-9318184. The content of this paper does not necessarily reflect the position or policy of the U.S. Government—no official endorsement should be inferred or implied.

to the standard assumption of a homogeneously mixing population [2]. Increasing host spatial aggregation in a population of given density can increase the chance of parasite extinction, and reduce the endemic frequency of infection when the parasite advances [8].

Genetic algorithms define genetic data for each simulated individual and so offer an exceptionally powerful (and computationally intensive) method for analyzing evolutionary processes. Genetic algorithms have only recently been applied to questions concerning coevolution [24].

2 A Spatially Explicit Four Species Model with Genetic Information

We refer to a particular macroparasitic or microparasitic genotype as a strain [15]. The virulence of a strain reflects the strain’s capacity to exploit its host, and so controls the rate at which it converts host resources to its own reproduction.

A model that we proposed here is an extension of the spatially explicit individual based cellular automata model that we developed for vector born diseases [23]. The model enables the user to define an interaction neighborhoods for each site. This neighborhood, termed the *ecological stencil*, may extended beyond the nearest neighbors of the site. When the area of the ecological stencil approaches the environment size, the model tends to behave in a spatially homogeneous fashion, while small ecological stencil sizes support greater spatial heterogeneity.

The model represents space as a two-dimensional lattice with $J \gg 1$ sites and eight nearest neighbor connectivity with L_∞ norm as a distance metrics. Sites of a lattice can be empty or occupied by an individual host organism of one or two competing species. In addition, we allow a macroparasite and/or microparasite that infect hosts. If both macro- and micro-pathogens are present, than macroparasite acts as a vector for microparasite, so vector born diseases can be modeled. The state of a site n at time t , denoted $n(t)$, encodes the species (or lack of them) present at the site at time t . The global state space, $N(t) = \{n_1(t), n_2(t), \dots, n_J(t)\}$ is simply a collection of local states at time t . The ecological stencil of a site n is denoted as δ_n . Intuitively, the stencil of site n defines a “sphere of influence” of a host residing at site n , that is all sites from which colonization or infection can reach site n . One version of the model assumes that the time is discretized into uniform *time steps* of size Δt . For time steps small enough so there are no sites which violate the Poisson single transition rule, the model accurately approximates a continuous time model. For larger values of Δt , the model behaves in a discrete time fashion.

2.1 Modeling Mutation

We assume that there are two competing host species $h \in \{0, 1\}$ and two disease causing agents: a macroparasite species that spread via direct contact, and a vector-borne microparasite (pathogen) species. The user can selectively remove some of the species to perform controlled experiments for specific species interactions like we have done it with the TEMPEST model [16, 17, 18].

The model described here assumes haploid species. We introduce genetic data to answer two biological questions. First, we want to find out what is the probability that a host will be infected. Second, we want to measure what impact does an infection have on the host’s fecundity and mortality. The first question concerns susceptibility; the second regards recovery and its costs.

Let the macroparasite have k_{pi} bits of genetic information defining how it subverts the host’s resistance, and let the host have the same number of bits of resistance information. Likewise, let k_{vi} represent the number of bits of genetic information encoded by the microparasite as well as the number of bits of resistance information in the host. The k_{pi} and k_{vi} bits of genetic information for the host as well as for the macro- and microparasite are further partitioned based on their role in immunity (probability of successful infection and virulence).

After we describe the structure of the genotypes of hosts, as well as the macro- and microparasite, we consider many model subsets, each corresponding to a particular genetic mechanism, and the related intra-species effect, or inter-species interaction.

2.2 The Transmission Mechanisms of Genetic Data

Genetic algorithms (cf. [4, 12]), use the copying, cross-over, and mutation operators. Each operator is analogous to a biological process. The initial step in the production of an offspring requires the copying of (selected) parental genetic information. Mutation occurs when parental genetic information is copied imperfectly; a bit value of the offspring’s genotype is inverted. Cross-over takes genetic information from two parents and recombines the information to specify the offspring’s genotype. Fitness functions quantify an individual genotype’s success when it interacts with other individuals.

Immunity is an acquired form of resistance to disease which individual hosts may develop, as discussed below. The host’s *immunotype* governs the host’s specific resistance to infection, and the genotype of a strain is subdivided into the part that triggers the immunotype response, and the part which controls the strain’s viru-

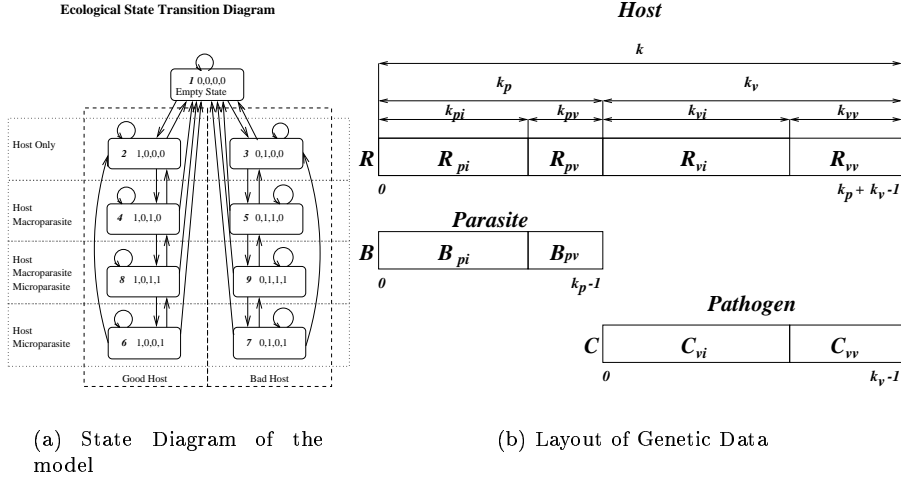


Figure 1: State information in the Host/Strain Coevolutionary Model

lence. To model virulence and infectiousness, we developed fitness measures that characterize a strain's genotype interactions with a host's genotype/immunotype. These measures are used in computing susceptibility, recovery, host fecundity and host mortality probabilities. The following bit patterns have one-to-one mappings between the hosts and the disease causing agents as shown in Figure 1.

The bit patterns indicate the combined *genotype* and *immunotype* of a particular host and the genotype of a strain of disease causing organism. The combined genotype and immunotype of the hosts and strains' genotypes govern the outcome of competition in the modeled system, by impacting fecundity and mortality. For generality, suppose that bit string $A = a_0, a_1, \dots, a_{n-1}$ represented an strain's attack strategy encoded in a bit string, and $D = d_0, d_1, \dots, d_{n-1}$ represented the corresponding host defense strategy (combined immunotype/genotype) encoded in a bit string. If bit $a_i = 1, i \in \{0, 1, \dots, n-1\}$, then a particular mode of attack is "turned on" or enabled, likewise if $d_i = 1$ the host's resistance to such an attack is enabled. Some fitness functions of the attack and/or defense bit strings which can be easily computed and used to evaluate the relative strength of attack and the probability of successful defense are described in Table 1.

2.3 Host Mortality Cost for Excessive Immunity

Consider how susceptibles' mortality is impacted by resistance to infection. The presented model assumes that resistance to infection is indicative of a more complex organism, which has reduced mortality rate, re-

Fitness Function	Counts:
$F_{RA}(A) = \sum_{i=0}^{n-1} a_i$	Raw Attacks
$F_{RD}(D) = \sum_{i=0}^{n-1} d_i$	Raw Defenses
$F_{BA}(A, D) = \sum_{i=0}^{n-1} a_i \wedge d_i$	Blocked Attacks
$F_{SD}(A, D) = \sum_{i=0}^{n-1} a_i \wedge \neg d_i$	Subverted Defenses
$F_{WD}(A, D) = \sum_{i=0}^{n-1} \neg a_i \wedge d_i$	Wasted Defenses
$F_{UA}(A, D) = \sum_{i=0}^{n-1} \neg a_i \wedge \neg d_i$	Unused Attacks
$F_{HR} = \sum_{i=0}^{n-1} a_i \Leftrightarrow d_i$	Defense efficiency
$F_{HD}(A, D) = \sum_{i=0}^{n-1} a_i \oplus d_i$	Attack efficiency

Table 1: Fitness Functions Used

duced fecundity, and reduced probability for selection at time of mating. This is in agreement with the classic ecological models of life span vs. fecundity. Let μ_s represent the probability of death during a single time step, Δt , for a host in state s . In the absence of infection, a host more resistant to infection might have a different average life span than a less resistant one. This gives a rise to either a penalty or reward function for immunity. Let an uninfected host have a base mortality rate $\mu_{hb}, h \in \{0, 1\}$. Based on the host genetic information describing its resistance to disease, the host mortality rate can be adjusted to values denoted as μ_{ha} . These values should span the interval $[0, \mu_{hb}]$ as a nonincreasing function to $F_{RD}(R_{pi})$ of the immunity and virulence resistance of the hosts (see Table 1).

2.4 Evolution of Disease Causing Organisms

Consider an evolving asexual disease causing species. The strains, also called *clones*, arise due to mutation. Selection via competition may be immediate or gradual, depending on the mechanism of competition.

When comparing two strains, a partial ordering can be imposed based on ecological fitness, with a *dominant strain* being more ecologically fit (i.e., having a greater value for the fitness function in the corresponding optimization problem) and a *suppressed strain* being less ecologically fit. Fitness functions rely on some combination of the strain's attack and host's defense capabilities as indicated by their respective genotype/immunotype information. The resources used to "fuel" the competition are derived from their host via the virulence measure, and attacker/defender string pairs are shown in Figure 1. Let A denote the bit string of the attacking strain and D denote the bit string encoding the defending host's genotype. Let $V(A, D)$ be the fitness function (virulence), of a particular strain/host genotype pair. Some candidate functions are found in Table 1.

The strain's virulence is important; increased virulence increases the probability that the pathogen will infect any available susceptibles during the next time period. This enhances the strain's fitness unless the greater virulence kills the host so quickly that a less virulent strain infects more susceptibles per infection. Hence a pathogenic clone's success can depend on its genotype, genotypes of its competitors and population density of host genotypes.

2.5 Structure of the Intra-host Competition

Both the macro- and microparasite are assumed to have asexual reproduction in this model (meaning that their genetic information is subject to the operations of copying and mutation). A strain (also called clone) on a host is *entrenched* if the host organism in question is infected by it. We begin by treating both macro- and microparasite as being asexual, and model infection with similar mechanisms. In this model, infection arises from the following events: (i) *exposure*, the contact mechanism which transmits parasites between the hosts, (ii) *inoculation*, the method by which parasites are introduced into the hosts to make the host immune to future attacks, (iii) *immediate competition*, which occurs when the number of attacking strains exceeds the capacity of a host to support them, so some strains will be removed from the host, either by superinfection (the new strain is dominant) or by competitive failure of the newly arrived strain, (iv) *gradual competition*, that happens when the strain gains access to

the hosts' resources and (if other strains are present) coinfection occurs, (v) *mutation*, the resident strains reproduce at regular (frequent) intervals (every time step) and create offspring, some of which may have altered genotypes due to mutation, (vi) *recovery*, a host survives diseases caused by a strain, with (possibly) acquired immunity.

2.6 Inoculation Mechanisms and Spread of Disease

Consider the following sequence of events assuming that the hosts acquired immunity. A host is born with no congenital immunity. Suppose that this host is later infected by a strain, having an *antigenic genotype* A_x , and then recovers. As a result, this host becomes *immune* to strain A_x . If in the future this host is exposed to A_x the host's immunological system will "recognize" the antigens associated with the strain and *antibodies* will be generated, initiating a rapid (near instantaneous) recovery from infection. Suppose that the same host is later exposed to another strain, with antigens represented via genotype A_y . The immune response to A_y depends on the similarities between A_y and A_x which decide whether the hosts acquired immunity allows it to resist infection.

Suppose the strain's antigen regulating portion of its genotype and the host's immunotype have $|D| = |A| = k_I$ bits of strategy information in their genotype. Let $D = d_0, d_1, \dots, d_{k_I}$ represent the immunotype of the host that are represented by the hosts bits R_{pi} and R_{vi} . Let $A = a_0, a_1, \dots, a_{k_I}$ represent the antigenic genotype of the strain, that is B_{pi} for macroparasite and C_{vi} for microparasite. Assuming that congenital immunity is disallowed, the initial state of D is assumed to be 0 (all defense bits disabled) upon birth of the host. After recovery from a disease, a host's defense bits (for immunity) are set to $d_i = d_i \vee a_i, 0 \leq i < k_I$ conferring immunity to inoculation to the strain. An inoculation's probability of success is a function of the attackers fitness with respect to the defender (see Table 1).

Now consider the complementary case where hosts do not acquire immunity during their lifetimes. The inoculation process becomes certain and $|A| = |D| = k_I = 0$.

An attack on the host can arise also from mutation of the resident strain which produces an offspring at every time step. Let A_x denote resident strain. If a copying error occurs in reproducing the genotype, then a new strain, $A_y \neq A_x$ will be introduced to the host. This results in an inoculation with A_y . The rate of mutation is determined by the user-specified per-bit copy error probability.

2.7 Mortality Cost of Virulence

There is a penalty to a macro(micro)-parasites that are overly aggressive in resource acquisition which represents an optimization constraint for virulence. One mechanism of introducing such penalty is to increase host mortality in proportion to the virulence of the disease causing agent, thereby killing its “meal ticket”. Let the infection of the host of type $h \in \{0, 1\}$ be indicated by state $s \in \{h + 4, h + 6, h + 8\}$.

Assume that the base host mortality probability over a time step for a given infection is denoted by μ_{sb} . An *adjusted mortality*, $\mu_{s,a}$ is based on the fitness measure of the disease and satisfies $\mu_{sb} \leq \mu_{s,a} \leq 1$.

In *simple attack* the host, is *attacked* by just the macroparasite, $s \in \{h + 4\}$, or solely by the microparasite, $s \in \{h + 6\}$. If both macro- and microparasite infect the host, then a *complex attack* occurs, i.e. $s \in \{h + 8\}$. The resulting cumulative virulence can be treated as an additive function.

2.8 Host Species Evolution

In this section we consider host reproduction and its impact on host evolution. Letting mortality (and infection) reflect the relative fitness of the host organisms, the reproduction will be (indirectly) impacted by their ability to survive until reproduction occurs, see Sections 2.3 and 2.7. Hosts may have an immunotype which records as a post recovery acquired immunity to a strain.

We want to apply genetic crossover in the reproduction of hosts. Without assigning individual hosts a gender, we can do so by having hosts “mate” and then reproduce. The host selects the mate within its neighborhood assigning the same probability of selection to all eligible organisms. If there is none in the neighborhood, the host is unable to reproduce.

After mating, a host reproduce by distributing its offspring/propagules to unoccupied sites within its stencil. The corresponding computation is broken into three stages: (i) determining the local rate of propagule creation/dispersal, (ii) computing the probability of a particular host propagating, i.e. placing its offspring at an empty site, (iii) calculating the genotype and lineage of offspring.

2.9 Local Fecundity Computations

Let ρ_{hb} represent the base (maximum) intensity with which a host of species h can produce propagules and ρ_{ha} represent the *adjusted* (or actual) intensity of a single host propagating. Note that ρ_{hb} is homogeneous across time and space, and is a user defined parameter of the model, while ρ_{ha} varies over time and space based on the occupant of a site k at time t (i.e., k and t are implicit parameters for the sake of brevity).

Conferring high levels of congenital immunity to an offspring taxes the hosts resources and can dampen host reproductive intensity. Likewise infected hosts may reproduce at a reduced rate in response to their infected condition.

2.10 Site Propagation Probability Computations

Given the individual rate of propagule/offspring creation of each site, the formulas given in [23] can be modified to take into account the reproduction (now heterogeneous). To account for the interactions between the competing species, we consider also the post dispersal competition [23].

2.11 Determining a host offspring’s genotype

The first step in computing a host offspring’s genotype is to uniquely determine the lineage to determine the parent’s genotypes. A way to do this (not necessarily computationally efficient) is to partition the interval $[0, \sum_{i \in \psi_h(k)} \rho_{ah}(i)]$ according to an ordering of the sites in the stencil, with the host at site i having a partition of size $\rho_{ah}(i)$ of the interval. Then, generating a uniformly distributed random number on this interval the identity of the parents is selected according to the ownership of a partition into which the generated random number falls.

At this point the exact genotype of the offspring should be computed using the cross-over, copy, and mutation operators. In nature, copying is assumed to be done first (via mitosis and meiosis), and at this stage mutation occurs (i.e., failure to create a perfect gamete due to copying error). Rather than modeling the copying operation directly, we merge it with the cross-over operation, invoking mutation in a probabilistic fashion. We can assume that mutation occurs probabilistically, and treat it as a Bernoulli trial for each bit copied.

3 Model’s Design and Implementation

We have implemented two stepwise refinement of the TEMPEST system [18]. First we allowed competition between two, genetically different strains in the system called STORM [19], next we refined STORM by allowing genetic drift of the hosts and parasites in the system called GALE. Below we compare the implementation of GALE with that of STORM and TEMPEST.

The GALE model was implemented using C++ and MPI. Placement of data and operations executed on a distributed-memory machine can significantly impact the total computation time. Hence, the partitioning of the simulation domain among the processors is crucial to running simulations efficiently in parallel.

The underlying model of GALE has localized interactions, so a static block data decomposition was se-

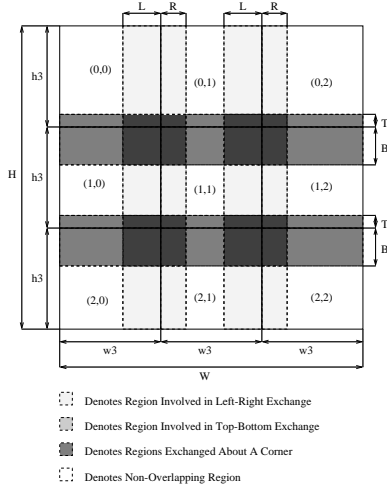


Figure 2: Block Decomposition used in STORM

lected (much like in TEMPEST and STORM [18, 19]) as shown in Figure 2. To compute state transition probabilities, GALE must traverse the set of sites within the interaction neighborhood, rather than to count the number of sites in each state as TEMPEST and STORM do [18]. To model reproduction, all genotypes that can reach a given site must be known, so GALE must maintain a dynamic set of eligible genotypes. Set union, intersection and difference operations have a cost proportional to the cardinality of the sets being merged. As a result, the state transition function computed at each simulation step is complex, compared to TEMPEST or STORM, because it involved operations on the multi-set of genotypes in the interaction neighborhood. This results in great dependence of performance on the area of the interaction neighborhood. Possible improvements include use of segmentation of the environment according to sites genotypes or classes of genotypes.

The processor which “owns” the partition in which a particular site resides is responsible for computing its next state, and therefore requires information about the current state of boundary sites on neighboring processors (since some stencils can span partitions). State information which can be efficiently computed locally and does not need to be computed across partition boundaries is maintained for each site. Data exchange across partition boundaries was done using MPI directly on the SGI, rather than allowing a direct read of data owned by neighboring processors. This avoided race conditions, and makes the software portable to networks of workstations and loosely coupled architectures. We used the shared memory chameleon (mpich) implementation.

4 Experiments with Model of Selection between Competing Strains

To verify that the spatial effects are important for evolution in epidemics, we simulated competition between two microparasite strains. The simplest case of such competition arise when there are just two competing strains transmitted via direct contact and a host is infected by at most one strain at a time [14].

For simplicity, we assume the selection on virulence. The reward function for increased virulence is defined by the primary effect of increased pathogen reproductive rate (as is manifested by infecting nearby hosts). The penalty function for increased virulence (i.e., a constraint) is defined by the increased mortality rate for hosts in presence of a more virulent pathogen.

Let s_0 , and s_1 denote two pathogen strains that differ in virulence. *Superinfection* is a disease preemption during which a host infected with s_0 becomes infected by strain s_1 via exposure from a nearby host. *Coinfection* occurs when multiple strains infect a host and order of infection matters. To denote coinfection when s_0 arrives first, we adopt the notation for the “pseudo-strain” s_{01} , but when s_1 arrives first then we introduce a pseudo-strain s_{10} (coinfective systems in which order of infection does not matter can be modeled by treating states with infection by either strain as equivalent which corresponds to merging the two defined above states into one).

Typically, coinfection and superinfection are modeled exclusively of each other, so our model provides a mechanism for disabling these events (although, as can be seen below, we permit the user to enable both as well). The state diagram of this simple model is shown in Figure 3. The parameters for the system are as follows. α_i denotes probability of exposure to strain s_i within δ whereas $\alpha_{j,i}$ is the probability of coinfection by strain s_i of the host infected with strain s_j . A_i denotes the probability of a susceptible avoiding exposure to s_i . γ stands for the competitive advantage of s_0 over s_1 . μ_i is a host mortality rate at state i whereas μ_{s_i} denotes probability of recovery from strain s_i . $\mu_{jk,i}$ stands for the probability of recovery of a coinfecting host from strain s_i assuming previous infection with s_j and s_k . p_{SI} is the probability of superinfection when a host infected with s_1 is exposed to s_0 . p_{ij} denotes the probability of coinfection with s_j after infection with s_i . ρ_i is the probability of host in state i placing a propagule at a single site within its stencil. Finally, $\sigma_i(k)$ denotes the number of hosts in state i about site k .

Let $p(x,y)$ be the probability of going from state x to state y in a single time step. An empty site will remain unpopulated with probability $p(0,0) =$

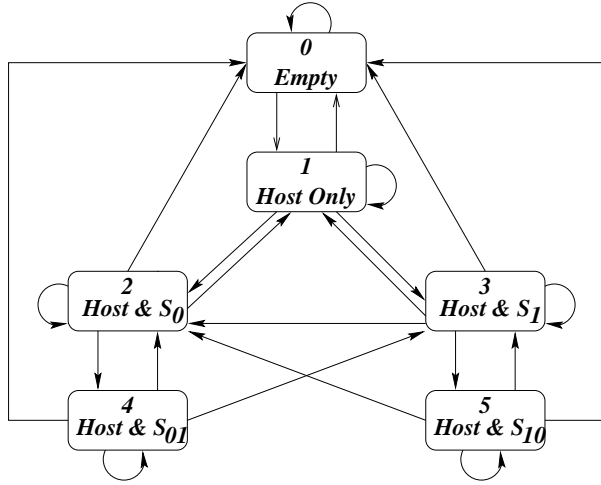


Figure 3: State Diagram of Coinfection/Competition Model

$\prod_{i=1}^5 [(1 - \rho_i)^{\sigma_i}]$, otherwise, it is colonized by an offspring from a nearby populated site. Since congenital immunity is disallowed: $p(0, 1) = 1 - p(0, 0)$.

Consider next a site occupied by a susceptible (i.e., an uninfected host). It may either die with probability $p(1, 0) = \mu_1$ or be infected by a strain from its neighbors. The probability of a host in state 1 avoiding exposure to each of the strains is:

$$\begin{aligned} A_0 &= (1 - \alpha_0)^{\sigma_2} (1 - \alpha_{10,0})^{\sigma_5} \\ A_1 &= (1 - \alpha_1)^{\sigma_3} (1 - \alpha_{01,1})^{\sigma_4} \end{aligned}$$

The Poisson single transition rule implies that if a susceptible is exposed to both strains, only one of them will be successful in infecting the host, so the probability of catching strain 0 is $p(1, 2) = (1 - \mu_1)(1 - A_0)[A_1 + \gamma(1 - A_1)]$ and the probability of catching strain 1 is then: $p(1, 3) = (1 - \mu_1)(1 - A_1)[A_0 + (1 - \gamma)(1 - A_0)]$

A host infected with strain s_0 will die (i.e., be removed) with probability $p(2, 0) = \mu_2$, and will recover (without immunity) with probability $p(2, 1) = (1 - \mu_2)\mu_{s_0}$. It will become coinfecting with strain s_1 after being infected by $s_{0,1}$ with probability $p(2, 4) = (1 - \mu_2)(1 - \mu_{s_0})(1 - A_0)$ and will remain in the same state otherwise.

Finally, consider the case of a host infected with strain s_1 . The probability that it dies is $p(3, 0) = \mu_3$. It will recover with probability $p(3, 1) = (1 - \mu_3)\mu_{s_1}$, and superinfection by s_0 will occur with probability $p(3, 2) = (1 - \mu_3)(1 - \mu_{s_1})p_{SI}(1 - p_{01})(1 - A_1)$. Coinfection by s_0 will occur with probability $p(3, 5) = (1 - \mu_3)(1 - \mu_{s_1})(1 - p_{SI})p_{01}(1 - A_0)$.

Consider a coinfecting host, where s_0 infected the host first. The host can die with probability $p(4, 0) =$

μ_4 , or it can recover from strain s_0 with probability $p(4, 3) = (1 - \mu_4)(1 - \mu_{01,1})\mu_{01,0}$. The host can recover from strain s_1 with probability $p(4, 2) = (1 - \mu_4)(1 - \mu_{01,0})\mu_{01,1}$, otherwise, the host will remain in its current state (we assume that simultaneous recovery from both strains is impossible).

Consider the complementary coinfecting host, where s_1 infected the host first. The host can die with probability $p(5, 0) = \mu_5$. It can recover from strain s_0 with probability $p(5, 3) = (1 - \mu_5)(1 - \mu_{10,1})\mu_{10,0}$. The host could recover from strain s_1 with probability $p(5, 2) = (1 - \mu_5)(1 - \mu_{10,0})\mu_{10,1}$, otherwise the host will remain in its current state (again, we assume that simultaneous recovery from both strains is impossible).

Similar reasoning can be used to extend this model to vector-borne pathogenic strains.

We implemented a tool, called STORM, to simulate the simplified model. In the simulation runs, we varied the spatial parameters governing host fecundity, $\rho_i, i \in \{1, 2, 3\}$ and rate of disease exposure from local infectives $\alpha_j, j \in \{0, 1\}$ so that the intensity of these processes would not be unduly impacted by variations in the area of the ecological stencil of each site.

The space was toroidally wrapped to avoid impact of boundaries on the simulation results. Initially 25% of the sites were populated by susceptible hosts in a spatially uniform density. Small clusters of infective hosts (0.25% of the sites) were placed in the initial environment, with maximum spatial separation to avoid premature extinctions induced by local superinfection events. The experiments used a region of 100×100 sites, however STORM has capacity for much larger simulations. STORM was run using the fixed spatial configuration for 2000 generations for each parameter combination.

The results of the experiment have shown a remarkable richness in the range of outcomes generated. First consider a case with two competing strains which are equal except for (i) a small difference in their virulence induced mortality and (ii) the possibility of superinfection by the more virulent strain, s_0 . We observed that the more virulent strain, s_0 can drive the less virulent strain, s_1 , to extinction via superinfection. The rise and decline of s_1 is governed by the rate at which the epidemic can spread through the environment, which is in turn correlated to the stencil's area, δ . With larger δ values, both s_0 and s_1 infection disperse more quickly so that the superinfection induced extinction occurs earlier.

Now consider a similar case, except that the virulence induced mortality difference is large. In this case, even the competitive advantage of superinfection is not sufficient to prevent extinction of s_0 . The rise and de-

cline of s_0 in this case reflects expansion of s_0 infection from its original small cluster and later a denial of access to susceptibles occurs due to the prevalence of s_1 infection.

Finally consider an intermediate level of virulence induced mortality. For a small stencil (e.g., the 3×3 case), the more virulent strain, s_0 , drives the less virulent strain, s_1 , extinct by gradually excluding it from susceptible hosts. For intermediate sized interaction neighborhoods (e.g., the 11×11 case), the virulent less strain, s_1 , drives the more virulent strain, s_0 extinct because of rapid death of hosts infected by s_0 . For large stencils (e.g., 33×33) s_0 and s_1 coexist, the first thanks to superinfection and the second because of increased access to susceptibles.

Such range of results can only be found in spatially explicit models and cannot be generated by spatially homogeneous models. The stepwise refinement of STORM calls for allowing the genetic mutation of strains and hosts, creating a framework for simulation of coevolution dynamics. We implemented a tool, called GALE, to simulate such extended model. The performance of GALE is the subject of the next section.

5 Performance of GALE

In this section, we compare and contrast the performance of GALE with that of STORM and TEMPEST.

The stochastic nature of the model (state transitions are selected randomly according to predefined distribution) require many runs of the same model with different parameters, initial configurations and random number generator seeds to obtain meaningful results (this process resembles sampling the state space with Monte Carlo methods in numerical analysis). Hence, the speed of computation is of utmost importance. We ran simulations for 100 time steps for an environment of 120×120 sites using stencil sizes of 3×3 , 11×11 and 33×33 on 1, 4, 9 and 12 processors. The run times were sensitive to variation of the interaction neighborhood's area as shown in Figure 4. We got a small "super-linear" speedup for small (nearest neighbor) stencils, which we attribute the high locality of reference resulting in a high cache hit ratio. The speedup curves dependent strongly on the stencil area. We believe the large slowdown observed at 12 processors for the 33×33 interaction neighborhood was caused by the large ratio of communication to computation ratio resulting from the stencil area approaching the partition size. Additionally, the systems software degrading performance by stealing cycles (because we utilized all 12 processors available in our configuration).

To measure the machine's ability to run large scale simulations, in the following experiment, we assigned a fixed number of sites to each processor. Per processor

allocation was one of the following: $100^2, 150^2, 200^2$ or 250^2 sites for 100 time steps on 1, 4, 9 or 12 processors on the SGI Origin 2000. The interaction neighborhood size was fixed at 11×11 . The speedups (see Figure 4) demonstrate some slowdown when going from single processor to multiprocessor runs (due to copying of boundary information) and another slowdown when all 12 processors were utilized. The slowdown when 12 processors were used was most pronounced for large per processor allocations. This might reflect increased contention of the cache coherent memory, and the overhead of systems software stealing resources to run during the simulations. We noted that the scaled speedup curves looked remarkably similar as problem size varied, but had some performance degradation when all 12 processors were used (achieving a speedup of about 11), which is within 92% of an optimal speedup.

Measuring the per processor throughput of the SGI Origin 2000 (denoted T_{SGI}) for the largest problem size yields:

$$T_{SGI} = \frac{100 \text{timesteps} \times 7.5 \times 10^5 \frac{\text{transitions}}{\text{timestep}}}{12 \text{processors} \times 600 \text{sec.}}$$

Hence, $T_{SGI} \approx \frac{1.04 \times 10^4 \text{transitions}}{\text{sec.}}$. Measurements for TEMPEST presented in [18] showed that the per processor throughput of a MasPar MP-1 was $T_{MP-1} \approx 44 \frac{\text{transitions}}{\text{sec.}}$, while the per processor throughput of an SP2 was $T_{SP2} = 1.2 \times 10^5 \frac{\text{transitions}}{\text{sec.}}$. The departmental network of workstations achieved $T_{NOW} = 2.7 \times 10^4 \frac{\text{transitions}}{\text{sec.}}$. STORM's performance on the SGI Origin 2000's processors [19] was $1.56 \times 10^6 \frac{\text{transitions}}{\text{sec.}}$ indicating that for that computation they did the work of over 35000 MasPar MP-1 processors, 13 SP2 processors and 57 NOW processors. GALE results indicate similar SGI performance, assuming that, based on the size of interaction neighborhood, GALE takes 150 times more work to update a site per time step than the STORM or TEMPEST do.

6 Conclusions and Future Directions

We presented a series of stepwise refinements leading to a computationally feasible and biologically interesting model of epidemics with evolving strains of pathogens. Evolution is the result of the combined effects of selection and mutation. By isolating and modeling the impact of selection on simple non-evolving systems, we provide a platform for validating and interpreting results when the complexity of evolution is added.

Efficiency and fidelity concerns motivate the consideration of whether a synchronous cellular automata or discrete event simulation engine is better for the simulation kernel. In both cases the space is partitioned into

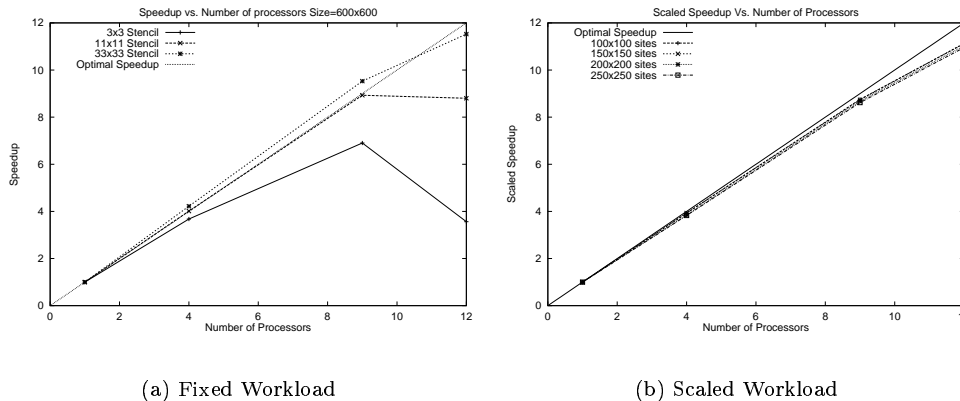


Figure 4: Speedups on an SGI Origin 2000

a two-dimensional lattice whose vertices are occupied by the simulated hosts. Cellular automata implementation inspects each point of this mesh in each discrete time step increment. The advantages of this approach are:

- High level of parallelism. Each point of the lattice can be inspected independently of others and require information about the state of the sites within its stencil only.
- Simple implementation thanks to the synchronous nature of the simulation.
- Nearly constant computational load on each processor during execution. The load is largely proportional to the the amount of space allocated to each processor, Hence, there is little need for dynamic load balance.

The disadvantage is that the time step of the simulation, i.e., the speed with the simulation progresses, has to be selected as the smallest time difference between any two events happening in the system. As a result in each simulation step many points in the lattice are inspected and left without a change. The more diversified the event structure and their inter-arrival times, the less efficient is this approach.

The second approach does not need discrete time steps to progress. Instead, each event in the system has its time selected independently of the others, based on the distribution of inter-arrival time for this kind of event as measured in nature. As a result, the points of the lattices are visited only whenever there is a change to be performed there, so the minimum number of

points are visited over the entire simulation. However, the disadvantages are:

1. Heterogeneous computational load in space. Depending on the density of hosts, macro- and microparasites, there is different speed with which events happens in different parts of space and therefore different amount of computations needed to execute those events. As a result, the dynamic load balance must be performed which is difficult because partitions must preserve continuity of the space allocation, and, if possible, regular shape of partitions.
2. Restricted parallelism [?]. Since the events should be processes in the increasing order of their time to avoid causality errors, the processes must coordinate their progress of simulation. Two approaches were developed to address this issue. One is to restrict parallelism by limiting the time by which any processor may get ahead of others in the simulated time (conservative approach). The other approach is to introduce a mechanism (often called rollback mechanism) that repairs casualty errors once they happen (optimistic approach).

Experience with Lyme disease simulation (without evolutionary effects, cf. [6, 5]), indicates that in case of complex species interactions or large diversity of time scales between species an optimistic approach is more efficient than the conservative one.

References

- [1] R. M. Anderson and R. M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, Oxford, 1991.

- [2] M. Bramson, R. Durrett, and G. Swindle. Statistical mechanics of crabgrass. *Ann. Prob.*, 17:444–481, 1989.
- [3] H. J. Bremermann and J. Pickering. A game-theoretical model of parasite virulence. *J. Theoretical Biology*, 100:400–426, 1983.
- [4] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York City, NY USA, 1991.
- [5] E. Deelman, T. Caraco, and B. Szymanski. Breadth-first rollback in spatially explicit simulations. In *Proc. PADS97, 11th Workshop on Parallel and Distributed Simulation*. IEEE Computer Society, Los Alamitos, CA, 1997.
- [6] E Deelman, T. Caraco, and B. K. Szymanski. Parallel discrete event simulation of lyme disease. In *Proceedings of the Conference on Biocomputing*. World Scientific Publishing Co, Singapore, 1996.
- [7] R. Durrett and S. A. Levin. Stochastic spatial models: a user’s guide to ecological applications. *Philosophical Transactions of the Royal Society of London, Series B*, 343:329–350, 1994.
- [8] M. Duryea, J. Gardner, T. Caraco, B. K. Szymanski, and W. A. Maniatty. Host spatial heterogeneity and extinction of an SIS epidemics. *Journal of Theoretical Biology*, 192:292–242, 1998.
- [9] P. W. Ewald. *Evolution of Infectious Disease*. Oxford University Press, New York City, NY USA, 1994.
- [10] S. A. Frank. A kin selection model for the evolution of virulence. *Proc. Royal Society London*, B250:195–197, 1992.
- [11] S. A. Frank. Models of parasite virulence. *Quarterly Review of Biology*, 71:37–78, 1996.
- [12] R. M. Fujimoto. Parallel discrete event simulation. *CACM*, 33(10):31–53, 1990.
- [13] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York City, NY USA, 1989.
- [14] C. Lefevre. Stochastic epidemic models for sir infectious diseases: a brief survey of recent general theory. *Biomath*, 86:1–12, 1990.
- [15] S. Levin and D. Pimentel. Selection of intermediate rates of virulence in parasite-host systems. *American Naturalist*, 117:308–315, 1981.
- [16] Hastings I. M. and B. Wedgwood-Oppenheim. Sex, strains and virulence. *Parasitology Today*, 13:375–383, 1997.
- [17] W. Maniatty, B.K. Szymanski, and T. Caraco. Implementation and performance of parallel ecological simulations. In *Proc. IFIP WG10.3 International Conference on Applications in Parallel and Distributed Computing*, Elsevier Science Publishers B.V., Amsterdam, Netherlands, 1993.
- [18] W. Maniatty, B.K. Szymanski, and T. Caraco. Tempest: a fast spatially explicit model of ecological dynamics on parallel machines. In *Proc. 1994 International Simulation Conference*, The Society for Computer Simulation, San Diego, CA, 1994.
- [19] W. A. Maniatty, B. K. Szymanski, and T. Caraco. Parallel computing with generalized cellular automata. *Parallel and Distributed Programming Practices*, 1(1):85–104, 1998.
- [20] W. A. Maniatty, B. K. Szymanski, and T. Caraco. High-Performance simulation of evolutionary aspects of epidemics. In *Proc. PARA98*, LNCS, Springer Verlag, Berlin, Germany, 1998.
- [21] R. M. May and Anderson R. M. Parasite-host coevolution. *Parasitology*, 100:S89–S100, 1990.
- [22] D. Mollison and S. A. Levin. Spatial dynamics of parasitism. In B. T. Grenfell and A. P. Dobson, editors, *Ecology of Infectious Diseases in Natural Populations*. Cambridge University Press, Cambridge, 1995.
- [23] M. A. Nowak and R. M. May. Superinfection and the evolution of parasite virulence. *Proc. Royal Soc. London*, B 255:81–89, 1994.
- [24] B. K. Szymanski and T. Caraco. Spatial analysis of vector-borne disease: A four species model. *Evolutionary Ecology*, 8:299–314, 1994.
- [25] M. J. Toquenaga Y., Wade. Sewell Wright meets artificial life: The origin and maintenance of evolutionary novelty. *Trends in Ecology and Evolution*, 11:478–482, 1996.
- [26] M. Van Baalen and M. W. Sabelis. The dynamics of multiple infection and the evolution of virulence. *American Naturalist*, 146:881–910, December 1995.