Simultaneous Inductive and Deductive Modeling of Ecological Systems via Evolutionary Computation and Information Theory

James P. Hoffmann
University of Vermont
Department of Plant Biology
109 Carrigan Drive
Burlington, VT 05405
James.Hoffmann@uvm.edu

The addition of two emerging technologies (evolutionary computation and ecoinformatics) to computational ecology can advance our ability to build better ecological models and thus deepen our understanding of the mechanistic complexity of ecological systems. This article describes one feasible approach toward this goal—the combining of inductive and deductive modeling techniques with the optimizing power of simple algorithms of Darwinian evolution that include information-theoretic model selection methods. Specifically, the author shows a way to extend classic genetic algorithms beyond typical parameter fitting of a single, previously chosen model to a more flexible technique that can work with a suite of possible models. Inclusion of the Akaike information-theoretic model selection method within an evolutionary algorithm makes it possible to accomplish simultaneous parameter fitting and parsimonious model selection. Experiments with synthetic data show the feasibility of this approach, and experiments with time-series field data of the zebra mussel invasion of Lake Champlain (United States) result in a model of the invasion dynamics that is consistent with the known hydrodynamic features of the lake and the motile life history stage of this invasive species. The author also describes a way to extend this approach with a modified genetic programming algorithm.

Keywords: Computational ecology, ecological models, evolutionary computation, information theory, model selection, genetic algorithms, genetic programming, zebra mussels

1. Introduction

Modeling ecological systems is important for many reasons. Models aid our attempts to predict future changes and can be invaluable for management purposes, and models can help us to understand better the inner workings of these systems and make our management decisions better informed and more effective. A modeler has many modeling techniques and approaches to choose from in building ecological models. Two fundamentally different ways to conduct the modeling process are with deductive and inductive reasoning. The deductive method requires expert knowledge to build a mechanistic-based model and depends on a first-principles understanding of the mechanisms acting within the ecological system. In contrast, the inductive method only uses the information content of the available empirical output data of the ecological system to construct a predictive model. Both methods have their strengths and weaknesses. The deductive approach can be more robust since its basis is the important operating mechanisms; however, this approach can be difficult because we often have an incomplete understanding of cause and effect in these systems. The inductive approach can produce models that are very precise in describing the empirical output data, but they may not generalize or scale well, and it can be difficult to extract causality from these models. In practice, scientists use a mix of both approaches to model ecological systems.

SIMULATION, Vol. 82, Issue 7, July 2006 439-450

DOI: 10.1177/0037549706069103

These different modeling approaches can produce models of diverse forms (structures), and the quality and consistency of predictions can be dependent on the particular model chosen. At times, the predictions can be contradictory among the different models. Attempts to model invasive species spread dynamics illustrate this point. Efforts to model mathematically the spatial spread of invasive species have been ongoing for nearly 50 years (for a summary, see [1]). Several invasive species model types have been developed, including both deterministic and stochastic models and various combinations of these approaches. Some of these models are parameter sparse, and others are not. Models based on partial differential equations (PDEs), integrodifference and integrodifferential equations, metapopulations, cellular automata, neural nets, and discrete event simulation techniques have been used to predict spatial spread [2-6]. Although some robust predictions have been produced (i.e., the asymptotic spread rate of invasive species appears to be linear in time), other predictions seem to be entirely dependent on the particular model chosen.

The model we choose invariably includes our biases and implicit assumptions, which sometimes leads to misspecification of the model structure (by misspecification, I mean omission of relevant explanatory mechanisms or variables, inclusion of irrelevant ones, or the adoption of wrong functional forms). Some of the problems with misspecified models are that they can be difficult to fit, and the quality of the model predictions can suffer from supersensitivity to small changes in model structure [7]. Conversely, if the model structure is correctly specified for the system of interest, then much can be learned via sensitivity tests of the model parameters, and a deeper insight can be obtained into the mechanisms that are operating to cause the observed system behavior. Furthermore, the model will likely be robust in its predictions. However, the selection of a properly specified model structure must be made prior to model fitting and sensitivity tests. Therefore, choosing a good model is more important than the subsequent fitting of model parameters. The critical question, then, is as follows: given a set of models to select from, how does one decide on the best model for a given problem? Statisticians have developed numerous ways to guide the choice of an appropriate model (see section 3), but they are all dependent on a priori specified models, whereas an evolutionary computation approach to model selection has the potential to incorporate novel models into the selection process. This is the idea that I explore in this article.

The rapidly growing field of ecoinformatics is providing us with new tools for managing and analyzing increasing amounts of spatially and temporally diverse ecological data, thus aiding our efforts for data-driven inductive modeling. At the same time, the use of model selection methods based on information theory is becoming increasingly popular among ecologists [8]. Evolutionary computation (EC), when combined with information-theoretic model selection, can serve as a bridge linking deductive knowledge-driven modeling to inductive data-driven modeling. This article shows two ways to integrate these two modeling approaches: with a model selection genetic algorithm that uses variable-length genomes and with a potentially more powerful genetic programming algorithm that incorporates domain-specific knowledge to evolve model structure. I first provide some background by describing the essentials of evolutionary computation and model selection, and then I describe the modified genetic algorithm that incorporates variable-length genomes and the information-theoretic Akaike model selection method. Following that, I will present some results of experiments that used noisy synthetic data and some promising preliminary results with real field data. Finally, I briefly describe a way to extend this approach by using a genetic programming algorithm instead of a genetic algorithm, and I end with some suggestions for future work and thoughts of the challenges that lie ahead.

2. Essentials of Evolutionary Computation

Evolutionary computation uses algorithms that emulate the basic principles of biological evolution. There are several types of evolutionary algorithms (EA); the three most common are genetic algorithms (GA), evolutionary strategies (ES), and genetic programming (GP). These algorithms are a class of nondeterministic (derivative-free), stochastic, iterative search techniques that emulate some of the principles of Darwinian evolution (precisely, selection and reproduction of the fittest individuals, with some introduced variation during the reproduction step). They resemble random search methods (i.e., a Monte Carlo approach) but with an important difference—through a selection procedure, an evolution of increasingly better solutions results in a directed search that eventually converges to an optimal solution. These algorithms are robust and are capable of solving a variety of combinatorial and numerical function global optimization problems. They are often the method of choice for difficult model fitting/optimization problems where there is a single model structure to fit, particularly when optimizing models that exhibit multimodality [9, 10]. The following description of the implementation of an evolutionary algorithm and associated data structures is for a simple classical GA; however, the essential principles and components, although often modified in their implementation, nevertheless apply to all EA.

The essential components are as follows:

- Population of individual solutions
- Fitness function to evaluate the quality of each solution (a.k.a. an objective or payoff function, or a figure of merit)
- Selection mechanism for choosing some individuals to reproduce
- Operators for rearranging and changing the information content of those individuals chosen to reproduce
- Termination criterion

Figure 1 shows an EA with these essential components in the order they are typically used.

Individual solutions, often represented as randomly generated one-dimensional arrays, encode the parameter values of a previously chosen model structure. These linear arrays equate to chromosomes, the parameter values to genes (binary, integer, character, or real values), and their information content to the genotype. Fitness evaluation consists of decoding the genotype to express the phenotype. This is done by running the chosen model with the individual's parameter values and comparing the model predictions to known data. Fitness, often calculated as some measure of total error between model prediction and observed data, is expressed as a scalar fitness score that maps directly to the quality of the predictions of that individual and is essential in the selection procedure. Selection of mating partners is either deterministic or probabilistic and favors individuals for reproduction with higher fitness values. Thus, an EA achieves a directed search and not a random walk.

Parameter values (genes) change during the search via recombination and mutation operators. Recombination, also known as crossover, mixes genes from two parent chromosomes to produce two offspring chromosomes, each of which has some genes from both parents. Mutation introduces random modifications to the genes, thus allowing the exploration of new areas in the search space, which in theory guarantees that every point in the search space is possible to reach. Mutation is the ultimate source of new genetic information for evolution since crossover only recombines existing information. Note that mutation alone, without selection or crossover, amounts to a random walk through the search space, whereas mutation with selection, but without crossover, creates a parallel, noise-tolerant, hill-climbing algorithm. In summary, many iterations of the EA fit parameter values to the data via the directed evolution of selecting good individuals to mate, and classical EA accomplishes model parameter fitting but not model selection since only one model structure is represented in the population.

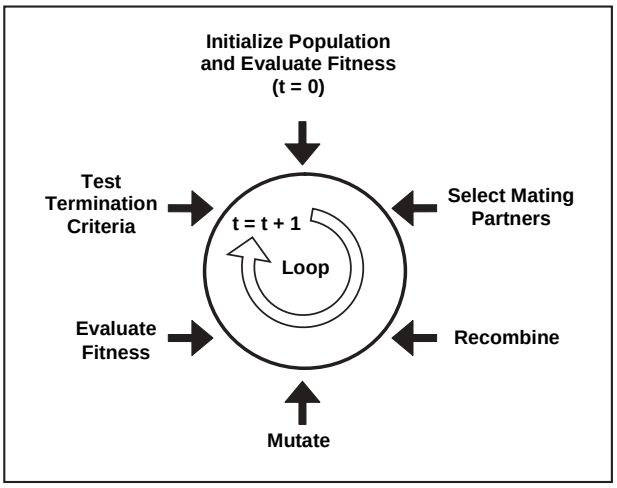

Figure 1. The basic iterative loop of an evolutionary algorithm. Modified from Schwefel and Kursawe [11].

3. Model Selection Background

Considering multiple working hypotheses helps us to minimize our human biases and tendency toward adopting a favorite hypothesis [12], which can hinder efficient progress in our understanding [13]. Our hypothetico-deductive method of science, when applied to modeling, consists of four basic steps that start with creating a plausible set of hypotheses based on our knowledge and assumptions of the system we are observing. Then, each hypothesis, which in effect is a model, is expressed mathematically, then it is confronted with data to fit the parameters, and, finally, we select either a "best" or a best set of models. These four essential steps of model selection provide the researcher with the ability to weigh the evidence for the various hypotheses and to infer the processes likely to have operated in generating the data patterns.

The formalization of the model selection step has a rich history, starting with the famous postulate of William of Occam—the simplest model that adequately describes the empirical data is usually the best one (Occam's razor). There are many techniques to select the "best" model, and Occam's emphasis on simplicity (parsimony) provides the philosophical basis of the quantitative model selection methods we use today. Some of these specific methods include the classical null hypothesis approach via likelihood ratio tests, best-subset regression, cross-validation, bootstrapping, Akaike information criterion (AIC), Bayesian information criterion (BIC), minimum descriptive length (MDL), Mallow's C_p statistic (all based on asymptotic methods), and lately nonasymptotic methods using concentration inequalities such as the Talagrand inequality (for an overview of these techniques, see [14]). Statistically, parsimony represents a trade-off between bias and variance in the parameter estimators—the former decreases and the latter increases with more parameters in the model. Therefore, too few parameters cause underfitting and fail to include effects in the model supported by the data, while too many parameters cause overfitting (i.e., fitting the noise in the data) and include effects in the model not supported by the data, resulting in poor model generalization. Both of these situations can result in misspecified models, and parsimonious model selection methods seek to minimize both underfitting and overfitting by finding an optimal balance between the bias and variance of the parameter estimators.

Hirotugu Akaike introduced a model selection method in 1973 [15] known as the AIC (see [16] for a comprehensive description of this method). The AIC method is attractive for several reasons. It is widely regarded as a

breakthrough in the theory of mathematical statistics because it formalized a robust relationship between the expected, relative Kullback-Leibler distance (a dominant paradigm in information theory) and Fisher's maximum likelihood theory. It is relatively easy to calculate and use for selecting the "best" model, and it is easy to understand qualitatively what the AIC means—a measure of the lack of model fit (negative log of maximum likelihood) corrected for bias (the number of model parameters). In addition, unlike the null hypothesis likelihood ratio tests that require nested models and cannot quantify the relative support for the various models, AIC does not require nested models and allows weighing the relative support for each model. Akaike saw his method as extending the maximum likelihood method, an extension that makes model selection and parameter fitting a joint optimization problem [17]. Note that selecting a model solely on quality of fit without regard to simplicity (i.e., via an algorithm based only on maximum likelihood or least squares) always favors the most complex models precisely because these are greedy algorithms that are biased toward models with more parameters due to their inherent advantage to better fit the data and their associated noise. Such models are overparameterized and overfit to the data. This is clearly seen in the experimental results with AIC turned off (section 4.4).

Since EC is a proven optimization technique, incorporating AIC within an EA should extend it beyond simple parameter fitting to include model selection that will enhance our abilities to choose and fit good models of ecological systems.

4. Incorporating Model Selection Criteria into Evolutionary Algorithms

Incorporating a model selection criterion into the fitness function of an EA is known as complexity-based fitness evaluation [18]. This approach to modeling is largely unexplored. Most experience with this method has been with GP (a type of EA that is based on parse trees) to control decision tree growth. Results are promising ([18] and references cited therein). However, few studies have used complexity-based fitness evaluation in GA. Konagaya and Konoto in 1993 (cited in [18]) used MDL for their fitness evaluation of a bioinformatics classification problem to minimize overlearning due to noise. Model selection and parameter estimation of linear autoregressive moving average (ARMA) models was attempted with an EA that combined GA and ES operators [19]. They tried different statistical criteria in their fitness function, and although estimation of the error series by the EA was successful, correct model identification was achieved only 20% of the time. In contrast, some success was reported with using MDL in a simplex GA for selection of regressors in linear AR models and in nonlinear polynomial models [20]. The authors accurately identified the correct operating models and demonstrated fast convergence rates compared to exhaustive search techniques. Therefore, EA with the inclusion of a model selection criterion, such as AIC, offers the potential of an automated, efficient search technique for good candidate models. In effect, the EA orchestrates a competition among a community of candidate models while simultaneously optimizing parameter fit to the observed data. Furthermore, by inserting expert knowledge into the set of candidate models, the EA can incorporate deductive modeling into the optimization process. Thus, evolutionary computation, when combined with information-theoretic model selection, can serve as a bridge linking deductive knowledge-driven modeling to inductive data-driven modeling. Figure 2 gives an overview of this combined approach. The key element that is required to implement this approach is variable-length genomes in the EA. Variable-length genomes are necessary to incorporate knowledge in the form of multiple model structures, thus creating a community of candidate models. Next, I describe one way to implement effective variable-length genomes in a GA.

4.1 Variable-Length Genomes in GA

Variable-length genomes include an evolvable on/off switch within each gene of the chromosome, and thus the modified GA can activate/inactivate each model parameter in each individual. Therefore, each individual has a complete genome and can potentially represent the most complex (global) model of a nested set (i.e., parameters for all models are contained in each individual—this is conceptually equivalent to totipotency in biological chromosomes). The switches are evolvable via mutation in the same way, as are the values of the model parameters, thus creating virtual variable-length individuals who effectively represent all the model structures of the set, depending on which model parameters (genes) are switched on. In the GA, the mutation operator randomly turns on and off these switches, and selective pressure ultimately evolves a "best" model structure. Figure 3 provides an overview of how the variable-length genome GA can conduct model selection.

The modulo-remainder function creates the evolvable switch that turns the genes on and off. Specifically, a binary switch is created by using modulus 2 on the integer-transformed gene value.

Thus,

$$\left[Int\left(a_i \times 10^k\right)\right] (\mathrm{mod} 2) = 0 \qquad (1)$$

or

$$\left[Int\left(a_i \times 10^k\right)\right] (\mathrm{mod} 2) = 1 \qquad (2)$$

for any i, where a_i refers to the real-value gene at position i on the chromosome, and k is of sufficient magnitude to shift the least significant digit to the unit position of the resulting integer. The specific value of k is dynamically determined by the magnitude of the real-value gene such that a very

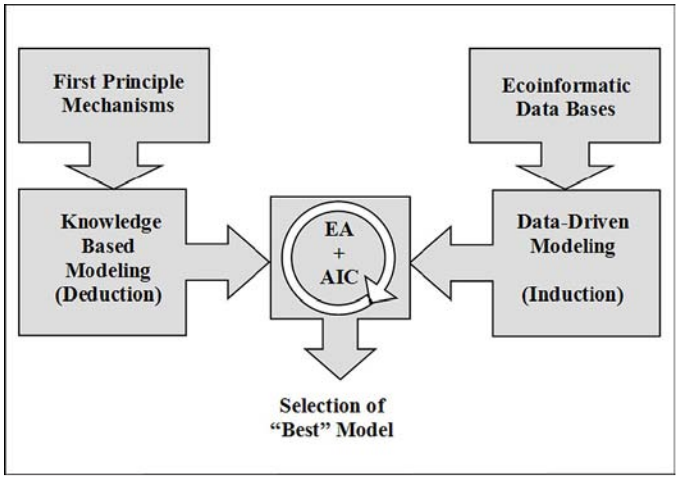

Figure 2. Conceptual view of the combining of deductive and inductive modeling via a complexity-based EA

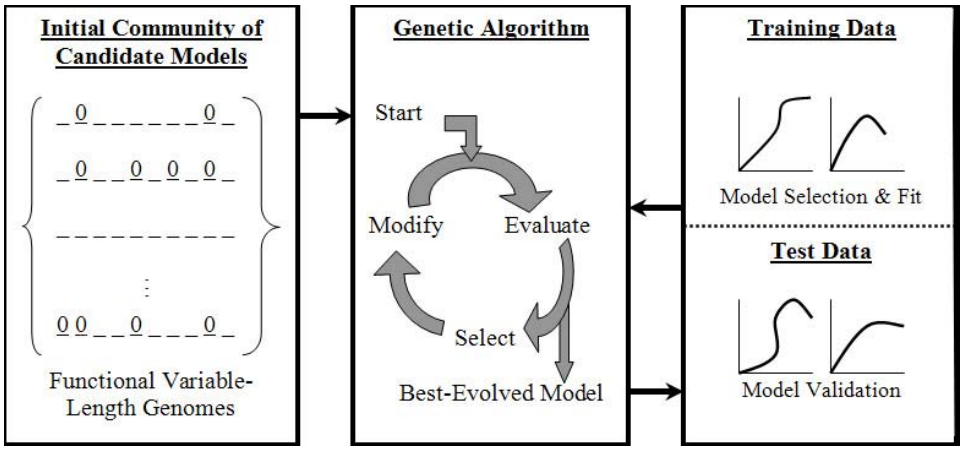

Figure 3. GA model selection and fitting procedure. The community of candidate models is randomly initialized. Some parameters are effectively turned off, depending on the state of their internal switches at initialization (0 indicates an inactive parameter). Therefore, although all models have the same fixed genome length, their structure, when evaluated by the fitness function, is functionally variable. The observations are divided into two data subsets: a training set used by the fitness function for evolving the best models and a test set for independently validating the best-evolved models.

large integer is produced on which to perform the modulo-remainder function—this increases the variability of the switch by making it very sensitive to mutation. Therefore, the least significant digit in the unit position of the integer-transformed gene determines whether that gene is active, whereas the most significant digits of the real value of the same gene determine its contribution to overall fitness (assuming the gene switch is on). Finally, the number of active genes is counted by the fitness function and is used in the calculation of the bias correction term of the AIC. This modulo approach using internal gene switches has a parallel in real organisms, where in some ribosomal RNA and tRNA genes, part of their coding sequence has a double function and serves as a regulatory switch for the gene.

In summary, these internal evolvable switches, created by overloading the gene variables, are the essence of the model selection GA method (hereafter referred to as MSGA).

4.2 Experimental Tests of the MSGA—Synthetic Data

Initial testing of this method used synthetic training data generated by a known model that was included among the set of candidate models available to the MSGA (for specific details of these tests, see [21]). Note that in each test, the GA is conducting a "blind" evolutionary search for the best model—no prior information is available to the algorithm about the correct data-generating model. Both noisy (maximum of 10% relative error as Gaussian noise added to data) and noise-free data were used in these tests. Three types of models (general polynomial equations, stock and flow system dynamic models [process models], and diffusion reaction PDE models) were tested. The GA software package used for these tests is a public-domain, parallel genetic algorithm function library written in ANSI C, known as PGAPack [22], and is available from the U.S. DOE Argonne National Laboratory (ftp://ftp.mcs.anl.gov/pub/pgapack). The fitness functions and models are coded in C, optimized, and parallelized for SMP (symmetrical multiple processors), and the numerical PDE solvers are licensed from NAG (The Numerical Algorithms Group, Inc., Downers Grove, IL) and optimized in FORTRAN for SMP. Computation times on dual-processor Xeon workstations depended on the model type and population sizes, but they generally were shortest with the fifth-order polynomial models (several minutes of CPU time) and were longest with the diffusion reaction models (approximately 6 hours of CPU time).

4.2.1 Polynomial Models

The general polynomial for these tests was

$$y(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \qquad (3)$$

where $a_0, a_1, \ldots, a_{n-1}, a_n$ are real-valued model parameters. In these experiments, both fifth and ninth-order polynomials were used (i.e., $n = 5$ or 9). Initialization by the GA created the community of competing models composed of the complete fifth- or ninth-order polynomials and their associated subset models. In the fifth-order polynomial experiments, the maximum number of competing candidate models was 64, and for the ninth order, it was 1024. Each individual genome represented the coefficients of the various model structures. The "correct" model used for generating the "true" data for these experiments was a specific fourth-order model, $y(x) = -20x^4 + 20x^2 + 14$, over the domain $\{-1.0, -0.9, -0.8, \ldots, 1.0\}$. Noise was added as "true" data + z*("true" data), where z is a random deviate from a Gaussian distribution ($\mu = 0$, $\sigma = 0.05$).

Model fitness is expressed as total error, calculated as the log residual sum of squares (RSS):

$$e = \log \sum_{j=1}^{m} \left(y_j - \hat{y}_j\right)^2, \qquad (4)$$

where m is the number of data points, y_j is the true value of the correct operating model at point j, and $\hat{y}_j$ is the predicted value of a candidate model at point j. The $\hat{y}_j$ value is calculated as

$$\hat{y}_j = \sum_{i=0}^{n} \left\{ \left[Int\left(a_i \times 10^8\right) \right] \pmod 2 \right\} a_i x_j^i, \qquad (5)$$

where n is the order of the complete polynomial model, and a_i refers to the gene (coefficient) value at position i on the chromosome. This estimated RSS is then transformed to the maximized log-likelihood [17] and the AIC bias correction term added. Therefore, the fitness of each candidate polynomial model is the true AIC.

4.2.2 Dynamic System Models of Leaf Photosynthesis

These models are useful to evaluate the ability of the MSGA to choose the correct model structure when presented with data produced from models considerably more complex than the polynomial test models. The system dynamic models for these tests simulated the physiological ecology of a leaf undergoing photosynthesis. Several submodels simulated the leaf's response to variation of different environmental factors. The leaf-photosynthesis model simulated the dynamics of the carbon, water, and heat budgets of the leaf over time. Soil water potential, herbivory, and ozone effects were also included in the model. The model comprised six ordinary differential equations that describe the state variables and fluxes. External forcing functions accounted for the influence of light intensity and duration, temperature, humidity, and wind velocity, and feedback loops linked the various model subcomponents together. The nonlinearities and interdependencies in the model produced complex behaviors in leaf temperature, heat content, and water and carbon content. Each individual in the GA contained 13 genes that represented the model parameters associated with the state variables and fluxes of the carbon, water and heat budgets, and effects of ozone and herbivory. The "true" model output data were generated from a subset model whose genes for ozone and herbivore effects were turned off. These data comprised a parallel time series of 10 metrics (photosynthetic rate, leaf carbon, etc.) observed at 15-minute intervals over a 24-hour period. For each candidate simulation model, a sum of the relative error of each metric at each time point was calculated, and a penalty for the number of active parameters in the model was added to this sum. Therefore, fitness is similar to a common analog of AIC [23].

4.2.3 Diffusion-Reaction Models

The final tests with synthetic data used diffusion-reaction (DR) models of the basic form shown in equation (6).

$$\frac{\partial n}{\partial t} = D\left(\frac{\partial^2 n}{\partial x^2} + \frac{\partial^2 n}{\partial y^2}\right) + f(n)\,n, \tag{6}$$

where D is the diffusion coefficient (a measure of how quickly the organisms move over a surface), n is the population density, and $f(n)$ is the per capita growth rate.

DR models are partial differential equations, which incorporate dispersal terms and population dynamics associated with the spread dynamics of an invasive species. Specifically, they can represent species whose densities and dynamics change due to (1) movement and (2) birth and death, and they have a continuous functional dependence on both space and time. However, they do not mechanistically describe most of the key ecological factors that influence the spread of invasions. Spatial effects such as habitat heterogeneity, mass transport via advection, linear or nonlinear density-dependent growth, and long- and short-distance dispersal can play important roles in invasive species dynamics [24-32]. It is possible to extend equation (6) to include some of these factors. Equation (7) allows for a community of candidate models that incorporate one type of dispersal (simple diffusion), two types of population growth (exponential and logistic), and advection in the X and Y directions and was used in these tests with synthetic data. The form of the complete model is

$$\frac{\partial n}{\partial t} = D\left(\frac{\partial^2 n}{\partial x^2} + \frac{\partial^2 n}{\partial y^2}\right) - w_x\frac{\partial n}{\partial x} - w_y\frac{\partial n}{\partial y} + \varepsilon\left(1 - \frac{\mu n}{\varepsilon}\right)n, \tag{7}$$

where D is the diffusion coefficient, w_x and w_y are advection parameters, and ε and μ are growth parameters relating to density dependence and per capita growth. The genome length of the DR models is 5 and allows for a community of 32 candidate models. The "true" model output data were generated from a subset model with diffusion turned on, linear density-dependent growth, and advection in the Y direction turned off. Fitness for these tests was the common analog of AIC [23]. The spatial domain was a square 21×21 grid, and the spread dynamics occurred over 10 time units.

4.3 Experimental Tests of the MSGA—Field Data

A final test of the effectiveness of MSGA used a field data set of the zebra mussel invasion of Lake Champlain (United States). Lake Champlain occupies a north-south geological fault zone and is long (193 km) and narrow (19 km at its widest point). It is located at 44.50 latitude and –73.25 longitude and is the sixth largest lake in the United States. The predominant flow is north into the Richelieu River in Quebec, Canada, and the mean hydrologic residence time is 3.3 years. Additional information on Lake Champlain is available at http://www.worldlakes.org/lakedetails.asp?lakeid=8518. The zebra mussel data consist of a 10-year time series of the densities of veliger larvae, juveniles, and adult forms of this invasive species at 23 locations in the lake. The data are publicly available (http://www.anr.state.vt.us/dec/waterq/lakes/htm/lp_lczebramon.htm) and are considered the best whole-lake zebra mussel data set in existence due to the consistency of the methods used and the fact that the initial sampling occurred at the very beginning of the invasion in 1993. Zebra mussels were first discovered in the extreme southern portion of the lake and, over the next 10 years, spread northward throughout the entire lake. A succinct summary of the life history of this invasive species is available at http://nis.gsmfc.org/nis_factsheet.php?toc_id=131.

The "true" model is unknown when using field data, so for these tests, it is necessary to use the known facts of the life history stages of zebra mussels (specifically, the passively dispersed veliger larval stage) and the known hydrodynamic features of Lake Champlain as criteria for judging whether the MSGA has evolved a "correct" model. The larvae are released by the adults in large numbers ($\sim 10^6$/adult) from late spring to early fall (a period of approximately 4 months when the water temperature of Lake Champlain is sufficiently warm to allow spawning). This stage of their life history is planktonic for approximately 1 month and occurs more or less continuously during this 4-month period. Thus, it is reasonable to expect that the large-scale hydrodynamic features of the lake (predominant northward flow) will dominate the passive dispersal of the veliger spread dynamics on an annual time scale. Therefore, a "correct" DR model of their dynamics should include anisotropic advection in the northern direction, and the magnitude of the advection should approximate the known average annual northward flow rate of the lake. Furthermore, due to the extremely high fecundity of the mussels and large-scale mixing of the planktonic larvae, no Allee effect (positive density dependence over a limited range of density) is expected, but the data do suggest some negative density dependence. To see if the MSGA would evolve a model structure consistent with these three expectations, a gridded spatial domain of the lake was constructed whose cell size was approximately 1.4 km^2.

Despite the recognized high quality of the Lake Champlain data set, the observed veliger larvae densities show considerable variability typical of field data and furthermore constitute a sparse data matrix in time and space. It was therefore necessary to process the data before conducting the MSGA experiments with the field data. First, the data were averaged over the 4-month spawning period for each station in each year of the time series—this somewhat smoothed and effectively transformed the data to the appropriate time scale for comparison to the known large-scale annual hydrodynamic features of the lake. Second, the original data were supplemented with linearly inter-

Table 1. Effect of parsimony (AIC) and noise (N) on the success of the MSGA

Treatment → Model ↓	– AIC – N	– AIC + N	+ AIC – N	+ AIC + N
Polynomial (fifth order)	0/1000	0/1000	995/1000	983/1000
Polynomial (ninth order)	0/100	0/100	91/100	94/100
Photosynthesis	0/100	0/100	96/100	93/100
Diffusion-reaction	0/50	1/50	40/50	38/50

The numerator is the number of correct models evolved, and the denominator is the total number of replicates.

polated values to fill in the empty grid cells and then locally averaged to smooth further the training data. Only the first 7 years of the 10-year time series were used for these feasibility tests because this time period best depicts the onset and subsequent spread of the invasion, whereas the more recent years show stagnation and possible decline in densities. Local investigators are currently researching the cause of the stagnation and possible decline; however, no consensus of causal factors has emerged yet.

The MSGA was then used with a genome encoding for a seven-parameter DR model whose complete form (equation (8)) can describe dispersal as both simple diffusion and mass transport advection, linear and nonlinear negative density-dependent growth, density-independent growth (exponential), and positive density-dependent growth (Allee effect).

$$\frac{\partial n}{\partial t} = D\left(\frac{\partial^2 n}{\partial x^2} + \frac{\partial^2 n}{\partial y^2}\right) - w_x\frac{\partial n}{\partial x} - w_y\frac{\partial n}{\partial y} + \frac{a}{b}n^{1-2b}\left(K^b - n^b\right)\left(n^b - q^b\right), \quad (8)$$

where D is the diffusion coefficient; $w_x$and $w_y$are advection parameters in the E-W and N-S directions, respectively; a is a scaled intrinsic growth rate; b is a dimensionless constant that describes the rate of growth and density dependence (shape parameter) of the population; n is density; K is the carrying capacity; and q is the Allee effect population density such that for $n < q$, population density declines and eventually becomes extinct, whereas for $n > q$, population density grows toward K.

4.4 Results

4.4.1 Synthetic Data

The initial testing with synthetic data showed that for all the model types, the MSGA consistently evolved "correct" model structures even when the data were degraded with noise. When parsimonious model selection via AIC was not active, all the evolved models were incorrect (with only one exception) and were overparameterized and overfit to the data (Table 1).

This is evident in Figure 4, where all the replicates without AIC (parsimony off) had a considerably better fit to the noisy data by using additional parameters to fit the noise. Note that the larger the negative fitness value, the better the fit to the data. However, this was achieved with misspecified, incorrect models, whereas greater than 90% of the runs with AIC evolved the correct model and produced accurate and precise estimates of the "true" parameters despite the noisy data. The "correct" polynomial models evolved parameter estimates that were identical (to within 0.001) of those produced with a least squares regression on the noisy data, after using the best-subset method with Mallow's C_p statistic for variable selection.

4.4.2 Field Data

Twenty-nine of 36 MSGA experiments using the Lake Champlain veliger density data evolved the "correct" model structure with appropriate parameter values (a "correct" model is defined in section 4.3 as including the three expectations of anisotropic northward advection, some negative density dependence, and no Allee effect). The "correct" models have an average parameter value for the northward advection of the larvae of 62.9 km/yr (*SD*, ±2.5), which compares favorably to the independently estimated average of 60 km/yr calculated from the known hydrologic residence time and length of Lake Champlain. The average value of parameter $\boldsymbol{b}$ in the "correct" models (–1.67, $SD \pm 0.16$) suggests a negative nonlinear density dependence. For comparison, a $\boldsymbol{b}$ value of -1.0 indicates a logistic type of negative linear density dependence, whereas a $\boldsymbol{b}$ value of 1.0 indicates no density dependence (i.e., exponential dynamics). All of the "correct" models had evolved a model structure in which the switch for the Allee effect gene was turned off, thus indicating no significant positive density dependence. Figure 5 shows the density predictions of one typical "correct" evolved model of the spread dynamics of the veliger larvae compared to the field data. Although this model overestimated the spread rate in years 2 and 3 of the invasion, the general pattern of the predicted dynamics over the 7-year time period is consistent with the observed field data.

In summary, these results with both the synthetic and field data show that the MSGA approach is feasible and that overfitting of models can be avoided with the incorporation of AIC, even with noisy data.

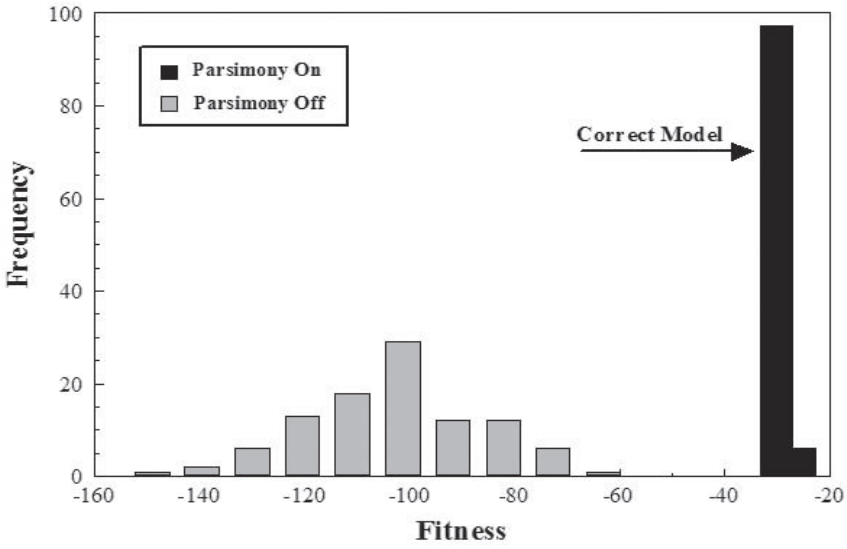

Figure 4. Success of the MSGA evolving the correct data-generating model. Frequency histogram of two experiments using the ninth-order polynomial model with Gaussian noise ($\mu = 0$, $\sigma = 0.05$). Each experiment involved 100 replicates.

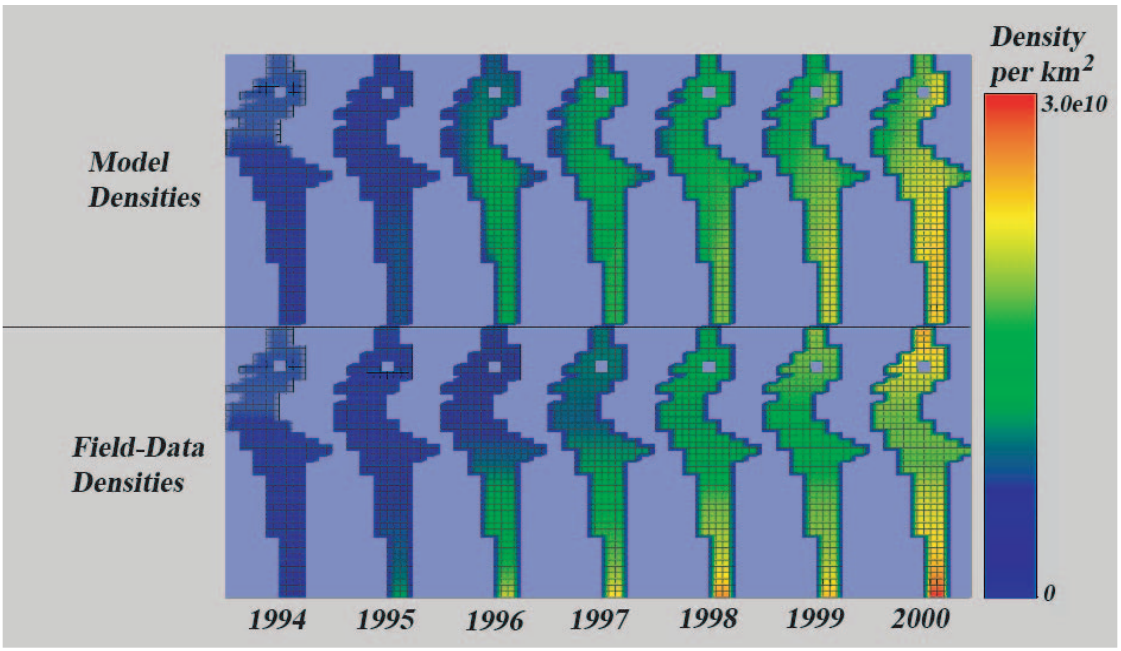

Figure 5. Time series of the zebra mussel veliger densities in Lake Champlain. The spatial domain of the lake is graphically shown here as the set of 1.4 km^2 grid cells used to model the veliger larvae spread dynamics. The upper panel depicts the DR model predictions, and the bottom panel depicts the processed field data. The noticeable light area in the northern portion of the lake in 1994 is due to the veliger larvae not reaching that region of the lake in that year. The small square region in the northern portion of the lake represents islands.

5. An Alternative Approach—Genetic Programming

GP is a variation of GA, in which both the model structure and associated parameters are encoded into the individual genomes [10]. Typically, the genome is represented as a tree that can increase or decrease in size via variation operators (mutation and recombination), and thus variable-length genomes are intrinsic to GP. Therefore, GP expands the search for a good model by allowing not only the parameter values but also the model structure to evolve during the search. However, this also greatly increases the search space and makes the evolution of a good model computationally more difficult. Nevertheless, GP has the potential to address some of the disadvantages of the MSGA while retaining its advantages.

5.1 Advantages and Disadvantages of the MSGA

In the MSGA, the modeler explicitly creates the set of competing candidate models, which incurs some advantages; the specified model structures include the empirical knowledge and mechanistic understanding of experts of the ecological system being modeled. However, there are also disadvantages; the models invariably also include the modeler's biases and implicit assumptions, which can lead to misspecification of the correct model structure. In addition, an adequately specified model must exist among the set of candidate models contained within the global model; furthermore, the set of competing models is closed, and therefore the MSGA cannot generate, via the evolutionary process, any novel model structures. Thus, our ability to discover novel models, with the ability to generate new understanding of internal mechanisms, is limited. These disadvantages are not unique to the MSGA but are recognized as general limitations of model selection methods.

Several different approaches have been used to minimize these general limitations. One method to avoid bias and unintentional assumptions is to use partially specified models to improve the fitting of complex biological systems [33]. In this approach, the model structure includes only well-understood elements, whereas less well-known parts of the biology are represented in a flexible nonparametric way. Although this approach does minimize model misspecification problems, it does not allow for the discovery of new model structures. Another approach that has been pursued in the field of artificial intelligence is known as automated modeling (AM) [34]. AM—specifically, a type called compositional modeling—has been most successful when modeling physical systems. In AM, the model is constructed automatically by using model fragment libraries of varying complexity. Some researchers have attempted to use these techniques on biological systems [35], especially ecological systems, but this effort is only in the initial stages. A related approach, known as Equation Discovery, uses a context-free grammar, parse trees, and parsimony implemented via minimum description length [36, 37]. The Equation Discovery method has been successful in constructing models of ecological systems but is limited by the exhaustive nature of its search method. A directed evolutionary search with GP may be a better way to address these limitations.

5.2 A Proposed GP Model Selection Procedure

GP can evolve a set of models from a construction set of model components (fragments) that can be assembled in various ways to represent a complex biological system. There would need to be a set of rules for their assembly (biologically impossible connections should be prevented). It would also be necessary to incorporate some measure of complexity to ensure parsimonious model evolution. It may be possible to extend AIC to include the number of components and their interconnections, and another option would be to use MDL. The library of model components in the construction set would include stocks (state variables; i.e., population density, etc.), flows (fluxes; i.e., growth or dispersal rates, etc.), inputs (environmental factors; i.e., temperature, currents, etc.), and protected combinations of the previous three components that encapsulate expert knowledge. The latter component (known as model blocks, super-blocks, or fixed submodels, depending on their complexity) would exist as predefined functions (PDF) in the construction set. Model structure would be composed of two primary components: stocks and flows. Stocks are state variables that can be modified by flows. Flows are rates of change and are determined by formulas. The operands of these formulas are combinations of external input data, previously calculated stock values (feedback), and mathematical constants. The operators are a basic set of mathematical functions (+, –, *, /, exp, etc.). Figure 6 depicts this GP model selection procedure.

This GP approach shares some similarity with AM but differs in that it relies on Darwinian evolution to direct the search for good model structures. Model fitness would be evaluated by comparing model predictions to a subset of the measured data. Another subset of the data would be used for validation of the best-evolved model. This approach is similar to one adopted for modeling industrial processes [38, 39]. Another group in New Zealand is using grammar-based GP to evolve models of water quality. Their approach differs from this method in that their grammar rules significantly bias the model search space to a limited set of structured equations—specifically, single-equation time-series models [40].

6. Summary and Conclusions

Evolution by natural selection is a superb optimizer of biological structure and function. The behavior of ecological systems derives from the optimized biological structure, function, and interactive mechanisms of its component parts. Akaike's significant contribution to our understanding of mathematical statistics brought together model

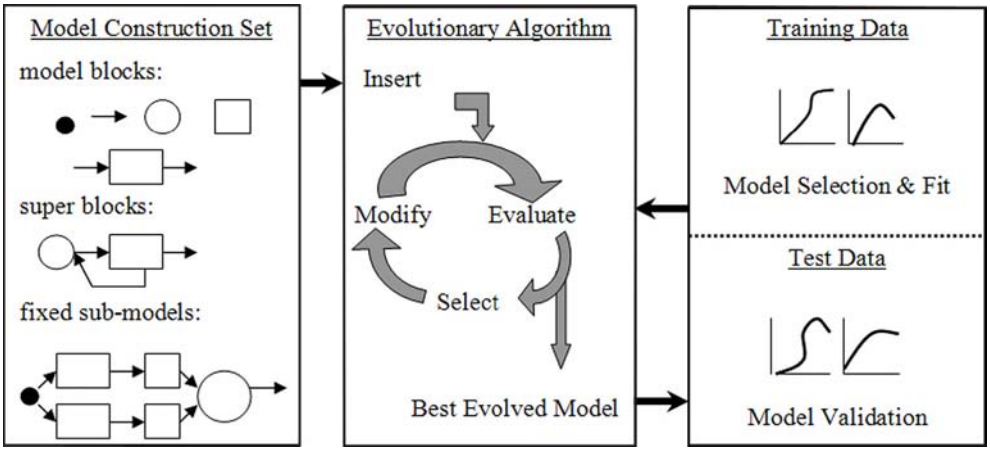

Figure 6. GP model selection and fitting procedure. The model construction set is initialized randomly but can also be "seeded" with models drawn from the set used in the MSGA approach. State variables are represented as rectangles, fluxes as arrows, and inputs as circles. The observations are divided into two data subsets: a training set used by the fitness function for evolving the best models and a test set for independently validating the best-evolved models.

selection and parameter fitting under a common theoretical framework of optimization. Therefore, since we seek to understand the important operating mechanisms of evolutionarily optimized ecological systems through modeling, it is good sense to explore the potential of combining algorithms of evolutionary optimization with model selection methods to develop better models of these systems.

The results reported here suggest that this approach has significant potential and warrants future exploration. Nevertheless, formidable challenges lie ahead, specifically in increasing the computational efficiency of these algorithms to explore adequately the very large search spaces. Faster computers, as well as clusters of computers, will help to alleviate this difficulty, but they will not eliminate this problem. Of the two methods outlined here, the MSGA is more computationally tractable than GP because it constrains the search to the a priori specified closed set of candidate models; however, it is more likely to suffer from biases and incorrect assumptions. In theory, GP does not suffer from these weaknesses and has the potential of discovering novel models, but because it attempts to search all model structure and parameter space simultaneously, it too must constrain the search by inserting expert knowledge and by the judicious choice of a finite set of mathematical operators for model building.

7. Acknowledgments

The progress reported here would not have been possible without the expert contributions of several colleagues. Dr. Daniel Bentil of the Mathematics Department at the University of Vermont and Dr. Bonsu Osei of the Mathematics and Computer Science Department of Eastern Connecticut State University are the forces behind all of the PDE-based modeling, and Mr. Chris Ellingwood of the Plant Biology Department at the University of Vermont has singly managed to implement all the diverse code necessary to bring these ideas to reality. I am most grateful for the essential contributions of these three colleagues, without which this work would not have been possible. USDA Hatch and U.S. DOE computational biology grants to the University of Vermont funded this work.

8. References

[1] Shigesada, N., and K. Kawasaki. 1997. *Biological invasions: Theory and practice*. Edited by R. May and P. Harvey. Oxford, UK: Oxford University Press.

[2] Hastings, A. 1996. Models of spatial spread: A synthesis. *Biological Conservation* 78:143-8.

[3] Higgins, S. J., and D. M. Richardson. 1996. A review of models of alien plant spread. *Ecological Modelling* 87:249-65.

[4] Kot, M., M. Lewis, and P. van den Driessche. 1996. Dispersal data and the speed of invading organisms. *Ecology* 77:2027-42.

[5] Hill, D., P. Coquillard, and J. De Vaugelas. 1997. Discrete-event simulation of Alga expansion. *SIMULATION* 68 (5): 269- 77.

[6] Wang, M. H., M. Kot, and M. G. Neubert. 2002. Integrodifference equations, Allee effects, and invasions. *Journal of Mathematical Biology* 44:150-68.

[7] Wood, S. N., and M. B. Thomas. 1999. Super-sensitivity to structure in biological models. *Proceedings of the Royal Society (B)* 266:565-70.

[8] Johnson, J. B., and K. S. Omland. 2004. Model selection in ecology and evolution. *Trends in Ecology & Evolution* 19:101-8.

[9] Bäck, T., D. B. Fogel, and T. Michalewicz. 2000. *Evolutionary computation 1: Basic algorithms and operators*. Bristol, UK: Institute of Physics Publishing.

[10] Eiben, A. E., and J. E. Smith. 2003. *Introduction to evolutionary computing*. Berlin: Springer.

[11] Schwefel, H.-P., and F. Kursawe. 1998. On natural life's tricks to survive and evolve. In *Proceedings of the Second IEEE World*

Congress on Computational Intelligence with the Fifth IEEE Conference on Evolutionary Computation, edited by D. B. Fogel, H.-P. Schwefel, T. Bäck, and X. Yao, 1-8. Piscataway, NJ: IEEE Press.
[12] Chamberlain, T. C. 1890. The method of multiple working hypotheses. *Science* 15:92-6.
[13] Platt, J. R. 1964. Strong inference. *Science* 146:347-53.
[14] Foster, M. R. 2000. Key concepts in model selection: Performance and generalizability. *Journal of Mathematical Psychology* 44 (1): 205-31.
[15] Akaike, H. 1973. Information theory and an extension of maximum likelihood principle. In *Second International Symposium on Information Theory*, edited by B. N. Petrov and F. Csaki, 267-81. Budapest, Hungary: Akademia Kiado.
[16] deLeeuw, J. 1992. Introduction to Akaike (1973) information theory and an extension of the maximum likelihood principle. In *Breakthroughs in statistics*, edited by S. Kotz and N. L. Johnson, 599-609. London: Springer-Verlag.
[17] Burnham, K. P., and D. R. Anderson. 2002. *Model selection and multimodel inference: A practical information-theoretic approach*. New York: Springer.
[18] Iba, H. 2000. Complexity-based fitness evaluation. In *Evolutionary computation 2*, edited by T. Bäck, D. B. Fogel, and T. Michalewicz, 15-24. Bristol, UK: Institute of Physics Publishing.
[19] Rolf, S., J. Sprave, and W. Urfer. 1997. Model identification and parameter estimation of ARMA models by means of evolutionary algorithms. In *Proceedings of the IEEE/IAFE Conference on Computational Intelligence for Financial Engineering*, 237-43. Piscataway, NJ: IEEE Press.
[20] Vesin, J.-M., and R. Grüter. 1999. Model selection using a simplex reproduction genetic algorithm. *Signal Processing* 78:321-7.
[21] Hoffmann, J. P., C. D. Ellingwood, O. M. Bonsu, and D. E. Bentil. 2004. Ecological model selection via evolutionary computation and information theory. *Journal of Genetic Programming and Evolvable Machines* 5 (2): 229-41.
[22] Levine, D. 1996. Users' guide to the PGAPack parallel genetic algorithm library. Technical report ANL-95/18, Argonne National Laboratory, Batavia, IL. ftp://ftp.mcs.anl.gov/pub/pgapack
[23] Hongzhi, A. 1989. Fast stepwise procedures of selection of variables by using AIC and BIC criteria. *Acta Mathematicae Applicatae Sinica* 5:60-7.
[24] Murray, J. D. 1989. *Mathematical biology*. Berlin: Springer-Verlag.
[25] Hastings, A. 1990. Spatial heterogeneity and ecological models. *Ecology* 71:426-8.
[26] Cantrell, R. S., and C. Cosner. 1991. The effects of spatial heterogeneity in population dynamics. *Journal of Mathematical Biology* 29:315-38.
[27] Kareiva, P. 1991. Population dynamics in spatially complex environments: Theory and data. *Philosophical Transactions of the Royal Society, London B*, 330:175-90.
[28] Renshaw, E. 1991. *Modelling biological populations in space and time*. Cambridge, UK: Cambridge University Press.
[29] Holmes, E. E., M. A. Lewis, J. E. Banks, and R. R. Viet. 1994. Partial differential equations in ecology: Spatial interactions and population dynamics. *Ecology* 75:17-29.
[30] Shigesada, N., K. Kawasaki, and Y. Takeda. 1995. Modeling stratified diffusion in biological invasions. *American Naturalist* 146:229-51.
[31] Cantrell, R. S., and C. Cosner. 1998. On the effects of spatial heterogeneity on persistence of interacting species. *Journal of Mathematical Biology* 37:103-45.
[32] Mendez, V., J. Fort, H. G. Rotstein, and S. Fedotov. 2003. Speed of reaction-diffusion fronts in spatially heterogeneous media. *Physical Review E* 68:041105-1–041105-11.
[33] Wood, S. N. 2001. Partially specified ecological models. *Ecological Monographs* 71:1-25.
[34] Keppens, J., and Q. Shen. 2001. On compositional modelling. *The Knowledge Engineering Review* 16:157-200.
[35] Keppens, J., and Q. Shen. 2000. Towards compositional modeling of ecological systems via dynamic flexible constraint satisfaction. In *Proceedings of the 14th International Workshop on Qualitative Reasoning*, pp. 74-82.
[36] Todorovski, L., S. Dzeroski, A. Srinivasan, J. Whiteley, and D. Gavaghan. 2000. Discovering the structure of partial differential equations from example behavior. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 991-8.
[37] Dzeroski, S., and L. Todorovski. 2002. Encoding and using domain knowledge on population dynamics for equation discovery. In *Logical and computational aspects of model-based reasoning*, edited by L. Magnani, N. J, Nersessian, and C. Pizzi, 227-47. Dordrecht, The Netherlands: Kluwer.
[38] Brucherseifer, E., P. Bechtel, S. Freyer, and P. Marenbach. 2001. An indirect block-oriented representation for genetic programming. In *Proceedings of the 4th EuroGP Conference*, pp. 268-79.
[39] Hinchliffe, M. P., and M. J. Willis. 2003. Dynamic systems modeling using genetic programming. *Computers & Chemical Engineering* 27 (12): 1841-54.
[40] Whigham, P. A. 2001. An inductive approach to ecological time series modeling by evolutionary computation. *Ecological Modeling* 146:275-87.

James P. Hoffmann *is an associate professor holding joint appointments in the Departments of Computer Science and Plant Biology and he is one of the directors of the Integrated Biological Science Program at the University of Vermont. Dr. Hoffmann has a BS degree from Cornell University and a PhD degree from the University of Wisconsin-Madison. His main research interests are ecological modeling and simulation of aquatic systems, and evolutionary computation.*