

## Article

# Evolutionary computation in zoology and ecology

**Randall B. Boone**

Natural Resource Ecology Laboratory and the Department of Ecosystem Science and Sustainability, Colorado State University, Fort Collins, CO 80523-1499, USA

Address correspondence to Randall B. Boone. E-mail: [Randall.Boone@ColoState.edu](mailto:Randall.Boone@ColoState.edu).

Received on 15 February 2017; accepted on 4 October 2017

**Abstract**

Evolutionary computational methods have adopted attributes of natural selection and evolution to solve problems in computer science, engineering, and other fields. The method is growing in use in zoology and ecology. Evolutionary principles may be merged with an agent-based modeling perspective to have individual animals or other agents compete. Four main categories are discussed: genetic algorithms, evolutionary programming, genetic programming, and evolutionary strategies. In evolutionary computation, a population is represented in a way that allows for an objective function to be assessed that is relevant to the problem of interest. The poorest performing members are removed from the population, and remaining members reproduce and may be mutated. The fitness of the members is again assessed, and the cycle continues until a stopping condition is met. Case studies include optimizing: egg shape given different clutch sizes, mate selection, migration of wildebeest, birds, and elk, vulture foraging behavior, algal bloom prediction, and species richness given energy constraints. Other case studies simulate the evolution of species and a means to project shifts in species ranges in response to a changing climate that includes competition and phenotypic plasticity. This introduction concludes by citing other uses of evolutionary computation and a review of the flexibility of the methods. For example, representing species' niche spaces subject to selective pressure allows studies on cladistics, the taxon cycle, neutral versus niche paradigms, fundamental versus realized niches, community structure and order of colonization, invasiveness, and responses to a changing climate.

**Key words:** agent-based modeling, case studies, evolutionary programming, evolutionary strategies, genetic algorithms, genetic programming.

**Introduction**

Darwin described the origin of species based on extraordinary data collection, perseverance, and reasoning (Darwin 1859). He recognized that the process of natural selection had evolved species with adaptations that allowed them to survive the challenges of the habitats they occupied. His experiments with artificial selection in pigeons added to his observations of the natural world, as breeding of the birds and observations of the outcomes informed his thinking. Darwin would likely envy our ability today to simulate this selection and speed understanding, while never minimizing the value of observations, which, among other things, provide the patterns for which hypotheses

may be formed (e.g., Grimm and Railsback 2006). Through coding of so-called pure processes, analysts have full control over experimental settings (Peck 2004), and avoid the ambiguities inherent in real-world experimentation. Moreover, in simulation we can adopt a pathway to understanding called abduction (Griffin 2006), where rules of interaction are described that are hypothesized to explain a suite of observations. Through bottom-up approaches such as agent-based modeling, the interactions can be implemented to grow the response of interest (Boone and Galvin 2014). For example, hundreds or thousands of simulated individuals may be bred in moments in a controlled setting, and the nature of their offspring described.

Engineers too have long recognized that evolution in natural systems has solved many complex problems. That realization led to nature-inspired engineering and design in a field called biomimetics. An example is adoption of countless small hairs on tape to increase adhesion (Geim et al. 2003), which was inspired by the feet of geckos *Gekko gecko* that have many thousands of setae that allow the geckos to climb polished glass through van der Waals forces (Autumn et al. 2002). Engineers have also adopted the pathway nature uses in problem solving more directly, through evolutionary computation. By using computational pathways that emulate genetic mechanisms and natural selection, novel solutions have been evolved to complex problems. Rather than attempting to solve problems directly, efforts are put to designing systems that allow robust solutions to evolve. These approaches comprise methods within evolutionary computation (Fogel and Fogel 1996; Bäck et al. 1997; Eiben and Smith 2003). Applications are diverse (Kicinger et al. 2005), with examples (and example citations) being electrical circuits (Koza et al. 1997), mechanical components (Deb and Goel 2001), software design (Salustowicz and Schmidhuber 1997), hardware (Lohn and Hornby 2006), economics (Holland and Miller 1991), and even combat maneuvers (Smith et al. 1999), music (Tokui and Iba 2000), and art (Bentley 1999).

Despite frequent problem solving in computer science and engineering using computational methods inspired by natural processes that have their roots in ecology, the methods are less often used in ecology and zoology (e.g., Alander 1994). Many problems in ecology are certainly more complex than in engineering, and there is an appreciation for the ability of the evolutionary process to craft extraordinary solutions to challenges in survival and reproduction. More frequent adoption of evolutionary computation may help us speed testing theories in zoology and devising means to promote sustainability in ecosystems. Toward that end, concepts of evolutionary computation and 4 main methods within that group are reviewed. The scope of case studies is defined, and while doing so other methods of artificial intelligence are introduced to readers and put in context. Several case studies that introduce readers to the utility and flexibility of evolutionary computation in ecology and zoology are provided, followed by concluding remarks. The introduction and case studies cited may inspire the creative application of these methods to problems of interest to readers.

## Evolutionary Computation

Evolutionary computation may be defined narrowly or broadly. Broad definitions include many nature-inspired searching and learning algorithms, such as swarm optimization, bacteria foraging algorithms, neural networks, and many others (e.g., Eberhart and Kennedy 1995; Haykin 2009). Examples used here focus more narrowly on optimization techniques that adopt aspects of biological evolution, with individual reproducing and mutating solutions competing to solve a given problem. This includes 4 well-developed fields within evolutionary computation: genetic algorithms, evolutionary strategies, genetic programming, and evolutionary programming.

Evolutionary computation analyses begin by defining a function that reflects the feature to be optimized given the problem at hand. That objective function may seek to maximize some quantity, improve fit to a pattern, minimize resource use, maximize access to resources, or maximize production of offspring. Multiple constraints may apply in an objective function, seeking a solution that balances demands. For example, the objective for an electric circuit design may be to maximize performance while minimizing component and

construction costs. In zoology, example objective functions may be to locate optimal habitat, minimize predation risk, increase resource intake, improve biological fitness, or a combination of these—the objective function may include biological fitness of the type zoologists are accustomed to, or may be very different. The function includes parameters called control variables that comprise the components that evolve in an application. The values these variables adopt may be bounded in analyses. Lastly, the optimal solution spoken of in evolutionary computation is often not an optimum in a mathematical sense. An analyst defines some local optimum from the objective function that is sufficient to be considered a solution. This termination criterion may be adequate performance of an engineered item, or in zoology, the persistence of a population, sufficient agreement with observations, convergence of attributes among population members, a maximum number of generations, no change over many generations, or a combination of these or others. Defining objective functions and stopping conditions such that the local optimum approximates the global optimum is a main challenge in evolutionary computation.

The 4 general methods of evolutionary computation addressed were developed by different teams that worked independently in their formative years, and have had many modifications and improvements applied in years hence. With that, they share some similarities and difference. The methods are described in the following sections, and briefly compared in Table 1. The general steps are described for the most commonly used method, genetic algorithms, and visualized for evolutionary programming.

## Genetic algorithms

Genetic algorithms (Holland 1975; D'Angelo et al. 1995) are the most commonly applied evolutionary computation approach. The method adopts many aspects of natural genetic processes to rapidly search a parameter space. A character or bit string analogous to a chromosome is defined that is composed of genes, or bit patterns, that code for alleles (values) of control variables that in turn describe features of a solution. Sets of strings form a population, with members of that population being selected for based on their performance as judged by the objective function (often termed the fitness function in genetic algorithms). Strings are often initialized using random draws from within the reasonable bounds of the control variables. Definition of a chromosome and the linkages between genotypes and phenotypes are the most challenging aspects of genetic algorithms. Wagner and Altenberg (1996) cite this “representational problem” and provide discussion.

Genetic algorithm applications use mutation to add variability and drive selection, but also use so-called horizontal events, such as recombination through mating and hybridization, to create new allele combinations and improve the search of the parameter space (Holzinger et al. 2014). Fitness scores for strings are calculated based on the objective function, and only the best solutions survive, providing selective pressure. The best performing genotypes are most likely to mate and rebuild the population, yielding improved solutions. Echoing genetics in natural systems, when 2 parents breed, there is a chance that a crossover function combines complementary portions of the parent strings to yield new offspring that include unique genotypes. A mutation function will alter a randomly selected locus within a chromosome, under a rare probability. The best performing individuals may be protected from mutation and ensured to enter the next generation, termed elitism in genetic algorithms.

**Table 1.** The 4 main approaches used in evolutionary computation, selected attributes, and citations of interest to those wishing to adopt the approaches

Approach	Early proponents and citation	General focus	Control variables	Source of variability	Example challenge	Relative user base	To learn more
Genetic algorithms	Holland (1962, 1975)	Genotypic	Coded as alleles	Mutation, crossover, mating	Linking genotype to phenotype	+++++	Mitchell (1998); Hamblin (2013)
Evolutionary programming	Fogel et al. (1966)	Phenotypic	Diverse options	Mutation, mating	Phenotypes may be application specific	++	Fogel (1996)
Genetic programming	Koza (1992)	Tree-based	Parameters and functions	Mutation, crossover, mating	Tree bloat, overfitting, and trimming	+++	Langdon and Poli (2013)
Evolutionary strategies	Rechenberg (1965, 1971); Schwefel 1965, 1975)	Phenotypic	Vectors of parameters	Mutation, recombination	Phenotypes may be application specific	+	Beyer and Schwefel (2002)

Notes: The relative user base markers were assigned based on quoted searches of the 4 techniques in Web of Knowledge®. For more on the history of evolutionary computation, see Fogel (1998). Numerous commercial and non-commercial tools support evolutionary computation; in the R software (Vienna, Austria), for example, packages include DEoptim (Mullen et al. 2011), GA (Scrucca 2013), Rfreak (Nunkesser 2008), rgp (Flasch 2014), rgenoud (Mebane and Sekhon 2011), and others.

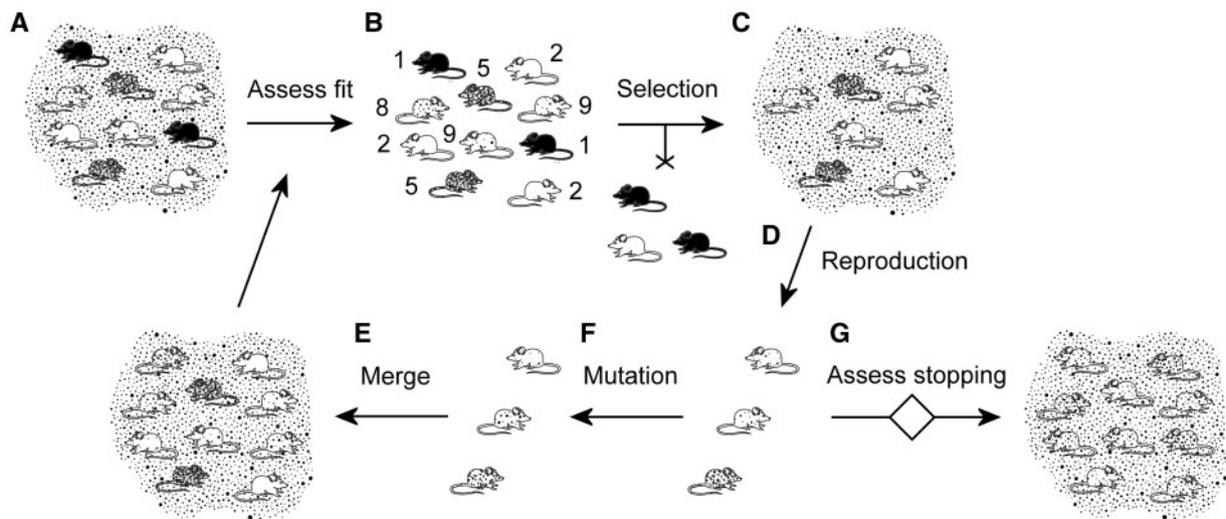
These functions form the building blocks for an iterative process in applications. A 1) population is initialized, and then 2) the fitness of each member of the population is assessed. The 3) best performing genotypes are retained in the population, and the remainder are removed. From the surviving members, 4) an individual is selected randomly and crossover of portions of strings may occur. The same or another individual may be selected 5) and randomly selected bits mutated. This process repeats through generations until the defined termination criterion is met, and the best-fitting solution is retained. Hamblin (2013) provides a primer on using genetic algorithms in ecological research, and includes citations for further reading on the topic.

## Evolutionary programming

Evolutionary programming focuses on phenotypic differences between individuals rather than genotypes, as in genetic algorithms. Initially developed by L. Fogel and advanced by him, family members, and colleagues (Fogel and Fogel 1996; Fogel 2006), evolutionary programming is now a common approach to design, especially in engineering. Evolutionary programming dispenses with chromosomal representations, crossover functions, and most other genetic mechanisms, relying upon mutation for variability in candidate solutions. Objective functions and the phenotypic descriptions that accompany them are flexible in this method; their definition remains a challenge and critical aspect of evolutionary programming, but they tend to be application specific rather than fixed structures.

The steps in evolutionary programming are streamlined relative to genetic algorithms (Figure 1). A population is represented in the figure as mice inhabiting a textured background. Some individuals will be poorly camouflaged and apt to be preyed upon, providing the selective pressure driving the evolutionary program. Here the phenotype includes control parameters that influence the pattern of fur color in the mice. The objective function plays the role of the perception by predators, quantifying the presumed visibility of mice within their patterned habitat. A simulation may begin with an initial population of mice with random coat patterns (Figure 1A). The objective function is then assessed (Figure 1B), assigning a fitness score to each individual, here the visibility of each mouse. Selection (Figure 1C) removes a portion of the population, simulating predation of the most visible mice. The remaining individuals then reproduce (Figure 1D), either through sexual mating of randomly selected individuals, asexual fissioning of individuals, mating that favors the most fit individuals, or other means. Some offspring may be mutated (Figure 1F), represented here by subtle changes to the pattern on coats of mice. Those individuals are then merged (Figure 1E) back into the larger group, restoring the size of the population. This completes a generation of the application, and the cycle then continues with the fitness of individuals again assessed (Figure 1B). Following reproduction in each generation, the application assesses whether or not the solution derived meets or exceeds a termination condition (Figure 1G). Here, that is represented by the fitness of all individuals reaching some maximum value, and through natural selection the camouflage of the mice in the population has improved.

A powerful approach, used in several of the examples cited below, is to leverage the representation of populations in evolutionary programming as many potential solutions, and employ an agent-based perspective, with agents as individual or groups of animals, plants, or people. With that, the selection represented in evolutionary programming represents natural selection in a truer sense, with organisms evolving to optimize access to resources, survival, maximize their range, etc.



**Figure 1.** A schematic representation of steps in evolutionary programming applications. Here predation on potentially camouflaged mice provides selective pressure. Steps in the algorithm are reviewed in the description of evolutionary programming. Steps in the algorithm are reviewed in the description of evolutionary programming, but in general, (A) represents an initial population, (B) is assessment of an objective function, (C) is following selection, (D) is reproduction, step (F) mutates a subset of organisms, and (E) is merging the offspring back into the population. After each generation, a stopping condition is assessed, (G).

### Genetic programming

Genetic programming, introduced by *Koza (1992)*, is a unique use of natural principles to evolve computer programs. Computer programs may be conceptualized as binary trees composed of parameters in leaves affected by operators in nodes. For example, a program to calculate area of a rectangle may include a length and width parameter in the leaves of the binary tree, and a multiplication operator at the node. In genetic programming, a population of program trees is generated that include random parameters and operators selected randomly from a defined set. With this initialized population, a generation is simulated using methods that are similar to those in genetic algorithms. The fitness of each tree is assessed using training data (or cross-validation), judging how close the result from each program is matching the data. The best performing program trees are preferentially selected for breeding to rebuild the program population. Crossover is represented by exchanging subtrees of trees selected for mating, and mutation may replace subtrees with newly generated random subtrees. The process then repeats until a termination criterion is met, and the best performing program tree is retained.

Of course, this brief introduction excludes many aspects of genetic programming, such as encapsulation (e.g., *Roberts et al. 2001*), where well-performing subtrees of programs (e.g., those appearing frequently in well-performing program trees) are prevented from being modified by crossover or mutation. Genetic programming is used in machine learning, image processing, and elsewhere, but has many applications in ecology and zoology as well.

### Evolutionary strategies

Evolutionary strategies (*Beyer and Schwefel 2002*) shares similarities of genetic algorithms and evolutionary programming, but was developed independently from those fields until the early 1990s (*Bäck et al. 1993*). Like genetic algorithms, recombination, mutation, and selection are used, but like evolutionary programming, a focus is on phenotypic rather than genotypic representations. Vectors of real values represent parameters in an objective function (newer forms of evolutionary strategies may use other types of values as well), and mutations are selected from normal distributions. The ability of individuals

in the population (early applications involved just an individual and 1 offspring) to solve the problem at hand helps determine mating, and offspring replace parents if they are more fit. In evolutionary strategies using standard methods, only the best-fitting solutions are allowed to produce related offspring, whereas in evolutionary programming, the best or randomly selected individuals may breed.

### Scope of Case Studies

The scope of case studies used is defined using an expanded discussion that introduces other aspects of nature-inspired computational methods. In brief, examples focus upon applications that include evolutionary principles applied to the behaviors of real-world organisms and their populations in zoology and ecology (as in *Figure 1*), although not always specific. An early effort by *Reynolds (1987)* sought to describe the complex and coordinated movements of birds, herbivores, or fish using simple rules. Three rules defining separation, alignment, and cohesion allowed coordinated movements to emerge. Conway applied simple rules in a cellular automata (*Gardner 1970*) that exhibited complex responses. Reynold's Boids, Conway's Game of Life, and other such biomimetic efforts have inspired myriad scientists and helped frame complexity science, but are not the types of applied studies that adopt evolutionary principles of interest here. Swarm intelligence is biologically inspired, based on sharing of information from decentralized and often simple agents that can lead to emergence of intelligence not held by any one individual (*Garnier et al. 2007; Parpinelli and Lopes 2011*), but typically does not include evolutionary components. Game theory has explored numerous topics of interest in zoology, such as altruism, cooperation, and competition. When applied in an agent-based setting (e.g., *Axelrod 1997*), evolutionary selective pressure has been incorporated, where strategies compete to yield optimum solutions. These settings, such as Prisoner's Dilemma, Hawk-Dove, and Rock-Paper-Scissors are often highly stylized (although they may apply to real-world settings, e.g., *Kerr et al. 2002*), and so are not a focus here. Evolutionary computational methods have been used with taxonomic databases in data mining exercises, for example, and learning classifier systems have been used in classification and

matching efforts. The Genetic Algorithm for Rule-set Production (GARP) Modeling System (Stockwell and Peters 1999) is a popular software approach and package that uses a genetic algorithm to extrapolate the distribution of species given a suite of environmental data and known occurrences. The method uses evolutionary techniques to define an ecological niche model for a species and a probability surface that shows where that species may persist. A related technique developed by Whigham (1995) uses genetic programming to define distributions. The method has been used to extrapolate the occurrence of marsupials, for example (Whigham 2000). These approaches to classification, clustering, mapping, and machine learning are excellent uses of evolutionary methods, but outside the scope of this introduction.

## Case Studies

### Physiology and Animal Behavior

The first example is a clever application of evolutionary programming to what appears a 2-dimensional variant of a popular problem in mathematics, random close packing (OMPC 1972). Barta and Szekely (1997) cite that egg shape is often explained by avian physiology or mechanical strength. Instead, the authors considered that the brood patch (a vascular and featherless area that develops on the abdomen of brooding bird that helps warm eggs) may be represented as a circle of limited area. Egg shape may be expected to vary for different clutch sizes. For example, for a single egg in a clutch, a purely spherical egg seems most appropriate. Barta and Szekely (1997) used a genetic algorithm to evolve optimal egg shapes for clutch sizes from 1 to 10. Four control parameters were used to describe egg shape, one controlling how round the egg was, and another the degree to which the egg was pointed, plus 2 that define the shapes of the ends of the eggs. After their application was run 30 times for each clutch size, the authors defined average egg shapes for different clutch sizes. They confirmed that an ideal egg for a clutch of one was spherical, and in other examples, a clutch of 2 yielded an almost symmetrical, double-pointed egg, a clutch of 5 yielded an egg shaped much like a chicken egg, and eggs from larger clutches were generally spherical. Their findings generally agreed with observed clutch size and egg shape combinations (Barta and Szekely 1997). Their work has been critiqued (e.g., Hutchinson 2000), but introduces the potential of evolutionary computation.

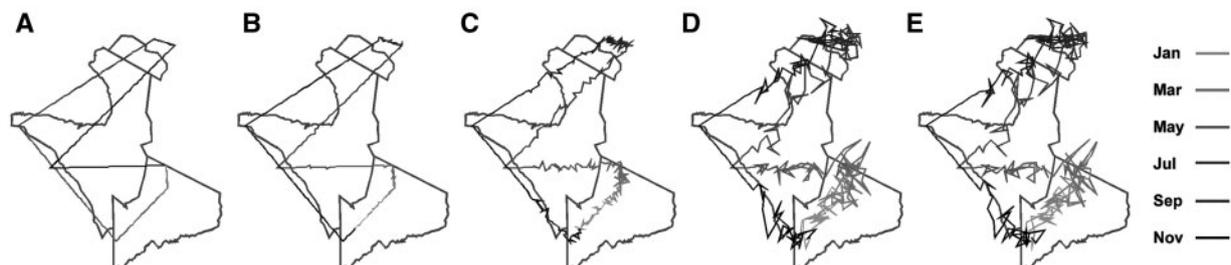
The flexibility in defining objective functions in evolutionary programming is evident in Boone et al. (2006). My colleagues and I considered the possibility that migratory pathways may be evolved; animals would either be better than their competitors at accessing resources through the year, or they would die. To test this, we used a well-known migration, that of white-bearded wildebeest *Connochaetes taurinus* in the Serengeti-Mara Ecosystem. About

1.3 million wildebeest join zebra *Equus burchelli* and Thomson's gazelle *Gazella thomsonii* in an annual migration. Migratory patterns are variable, but in general, animals are in and around the Maasai Mara National Reserve in southwest Kenya and in the western corridor in Serengeti National Park, Tanzania in August to October. In December–March, wildebeest are at calving grounds in the southern part of Serengeti National Park and the plains of Ngorongoro Conservation Area. Our goal was to simulate the evolution of this annual pattern of migration.

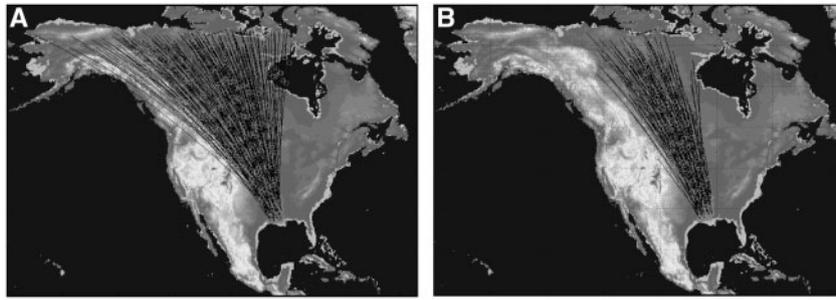
The objective of simulated wildebeest in Boone et al. (2006) was to maximize access to forage in and across years. We used 2 types of surfaces to represent forage availability, leveraging the highest spatial and temporal resolution datasets available. We used precipitation estimated from satellite images and ground-based observations, summarized every 10 days for a 5-year period, at 8 km × 8 km spatial resolution (Xie and Arkin 1997). We also acquired, for the same 10 day periods, surfaces of normalized difference vegetation derived from satellite images (VITO 2002), which reflected standing biomass and plant vigor, at 1 km × 1 km resolution. We standardized the pattern of rainfall within the period at hand to between 1 and 255, so that movements at each time of the year were equally important in the objective function. For NDVI, we calculated the difference between images from a given period and the previous period, to highlight areas of new vegetation growth (Boone et al. 2006).

Wildebeest phenotypes were represented by positional vectors of X, Y pairs that showed the daily locations of animals. The phenotypes were initialized entirely randomly by connecting 8 randomly selected locations in the ecosystem. Simulations proceeded much as in Figure 1, with 250 wildebeest competing to maximize access to new vegetation growth and rainfall. Mutation was represented by single pixel (i.e., ≤2 km) shifts in a single, randomly selected daily location. The simulation continued until the best performing migratory pathway had not changed in 5,000 generations.

Reasonable wildebeest annual migratory patterns evolved remarkably quickly. Typically in fewer than 10 generations, a migratory pathway that had animals using the southern and northern parts of their range at appropriate seasons was identified, and fitness improved over succeeding generations. Changing migratory patterns for the best performing animal in a single simulation are in Figure 2, and Boone et al. (2006) compares the best routes from 5 simulations to VHF- and GPS-collar data for real animals, plus the average monthly distributions of simulated animals to observed distributions gathered in 1969–1972. This approach may be used to evolve novel movement patterns to altered landscapes. A migratory pattern may be simulated as was done here, then a proposed land cover change, fence, road, or similar change in access may be incorporated into the spatial layers of a model, and migratory patterns



**Figure 2.** The best performing wildebeest migratory pathway across generations for a single simulation. Generations shown are (A) 1, (B) 100, (C) 500, (D) 50,000, and the final pathway, (E) 180,960.



**Figure 3.** Migratory pathways of pectoral sandpipers simulated using remotely sensed images and climate data for 10,000 birds without (A) environmental learning and with (B) environmental learning. Reproduced, with permission, from Smith and Deppe (2008), which includes a color version.

evolved again. Comparing the before and after patterns would quantify potential responses by species to the changes proposed.

A similar problem formulation was addressed by Smith and Deppe (2008), when they assessed the effects of wetland availability and variability on the fitness of migratory birds. Some bird species migrate to reach suitable breeding areas, and on migration stopover points are required for the birds to rest and build fat reserves. These wetland stopover sites in the central United States are declining or becoming more variable in their availability due to a warming and more variable climate, draining for agriculture, and other land-use changes. Smith and Deppe (2008) used an individual-based model applied to much of North America of female pectoral sandpipers *Calidris melanotos* that made use of remotely sensed land surfaces and climate data merged with biological data to forecast potential outcomes of future changes in wetland availability.

The authors used maximizing body fat as the objective function for successful bird migration (Smith and Deppe 2008). A bird's activity during migration is heavily influenced by the need to build fat reserves for reproduction and survival. Birds that had high-quality wetland habitats on their migration routes were assumed to build more fat reserves than other birds. Movement of birds was determined by their energy status, physiology, wind speed and direction, climate, and the quality of the habitat they occupied during stopovers. Stopover strategies, initial flight paths, and starting points were randomly assigned from candidate values. Birds were simulated flying through a landscape with maximum numbers of wetlands, and again with wetlands evident in remotely sensed images from the mid-1980s, when a drought was ongoing.

As the authors expected, sandpipers had greater fitness when flying over non-limiting stopover points versus those during drought, and they were more dispersed across the landscape (Figure 3). When the evolutionary programming approach was engaged, migratory routes shifted and birds avoided flying over water bodies or high elevations, where suitable wetlands were uncommon. Overall, pectoral sandpipers spent 12.75 days in stopover locations under variable wetland conditions, which was more than in the baseline result (Smith and Deppe 2008).

In an application using genetic algorithms, Dermody et al. (2011) explored the evolutionary history of feeding in vultures (*Gyps* sp.). Vultures are the only vertebrate obligate scavengers, having lost the ability to kill prey. The birds observe others in flight, and as an individual drops to a carcass others follow, and soon dozens of birds may tend a carcass. That behavior contrasts with information transfer that may occur at roosts, although the means by which that occurs is debated. The authors contend that simply being concentrated at roosts at the beginning of each day aids formation of foraging groups and that aids in locating carcasses (Dermody et al. 2011). They created a simulation that incorporated the 3 rules

Reynolds (1987) applied to Boids to represent vulture flocking behaviors: repulsion, orientation, and attraction. Birds may be searching, descending, or feeding. Searching proceeds from the beginning of the day, with a travel rate defined and turning rates constrained to be realistic. When a vulture encounters a carcass, it begins to descend, and depending upon model settings, others may follow. When a carcass is reached, the bird is then feeding, and the visibility of the carcass to other vultures increases greatly due to the presence of the bird.

A genetic algorithm was used to optimize the 5 controls on bird flight; turning rate; turning angle; and the distances of repulsion, orientation, and attraction, with each representing a gene within the chromosome. Elitist selection favored reproduction of individual birds that had spent the most time feeding, which was the fitness function being optimized (Dermody et al. 2011). Genes mutated during simulations of 100 days, with each step representing 10 s.

In an example of hypothesis testing using simulation (Peck 2004; Railsback and Grimm 2011), Dermody et al. (2011) assessed 4 strategies and compared their results, where: 1) vultures started each day at a roost, 2) vultures started the day randomly distributed in their spatial simulation, and in each case, vultures either: 3) ignored other vultures unless the focal animal was descending or feeding, or 4) reacted to other vultures within their field of view.

Roosting yielded the highest average fitness for all but the highest density of carcasses, where the difference between group and individual roosting approached zero. Moreover, group responses to sighting carcasses outperformed individual responses. Vulture density was then varied, but the group-roost strategy remained most fit (i.e., ordered as group-roost, individual-roost, group-dispersed, and individual-dispersed). The authors cite that roosting affects fitness outcomes more than group foraging behavior, and that information transfer is sufficient to explain roosting (Dermody et al. 2011). Many birds scan for carcasses and share information through observing their neighbors in flight, and young birds are introduced to good foraging areas by older individuals.

Mate selection, which by its nature deals with interactions between individuals, is amendable to an agent-based approach using evolutionary computation. A difficulty in evolutionary computation, whether in computer science, engineering, or in zoology and ecology, is the diversity of mate selection algorithms that may be used. For example, individuals may pair randomly, by spatial proximity, or may adopt assortative mating, where individuals select mates that are more similar to them phenotypically than expected by chance. Jaffe (1999 and earlier papers cited therein) used this approach to explore effects of different forms of mate selection. A genetic algorithm with 14 genes was used to represent a generic population of organisms susceptible to antibiotics and pesticides. Females selected

male mates of the same species in ways that varied genetically, either randomly, through a sex appeal gene, resistance to biocides, young or old males, assortative or disassortative mating (i.e., selecting males dissimilar to themselves). Jaffe (1999) demonstrated the benefit of female mate choice on overall fitness of the organisms, as expected. Moreover, sexually selected genes were sometimes fixed in the population very quickly, what Jaffe called run-away sexual selection. Assortative mating yielded a fit and evolutionarily stable gene pool, whereas disassortative mating was unstable.

Elk *Cervus elaphus* populations in Yellowstone's northern range have been the focus of agent-based modeling, reviewed by Bennett and Tang (2006), who focused on aspects of learning and memory. All of our examples include a memory or instinctual component, in so far as information is stored reflecting attributes or behaviors in chromosomes or phenotypes that can be passed on to successive generations. But here, Bennett and Tang (2006) take a more direct approach at representing memory. They adopted a method that incorporated a type of cognitive map that captured the repeated interactions thought to play a role in herbivore memory.

The authors used a 1 km<sup>2</sup> grid of landscape layers that describe snow, vegetation biomass, and topography in a graph. Coarse-resolution decisions by elk were made referencing this graph. A finer scale grid is used in statistical modeling of snow cover. Movement choices are made based on the environment, attributes of the individual elk, and short- and long-term memory (Bennett and Tang 2006). Elk may move slowly while foraging, or more quickly when traveling, and will stop if their daily forage intake is reached or their maximum travel distance for the day is met. A graph represented connectedness between patch centroids and the vertices and weights within that graph provided a means to represent memory for individual elk. Migration was represented as risk balancing potentially increased energy acquisition in a distant patch and the energy required to reach that patch.

Bennett and Tang (2006) used a genetic algorithm with an objective function that maximized animal fitness to allow elk to learn when to migrate, given snow depths and forage availability. Chromosomes represented edge weights that connect patches in the landscape. Habitat indices reflecting snow depth at time  $t-1$  and  $t$  formed an array used by animals when evolving the timing of migration. Less successful elk learned migratory behavior from more successful elk through mimicry. Paths were reinforced for animals through Hebbian learning, which, in brief, strengthens decision-making pathways that are used the most (Bennett and Tang 2006). The authors describe their work as proof-of-concept and conducted some preliminary evaluations of their approach (e.g., its stability and response to varying snow depth), but in general found the results promising. For example, Bennett and Tang (2006) found that elk migratory pathways were relatively stable.

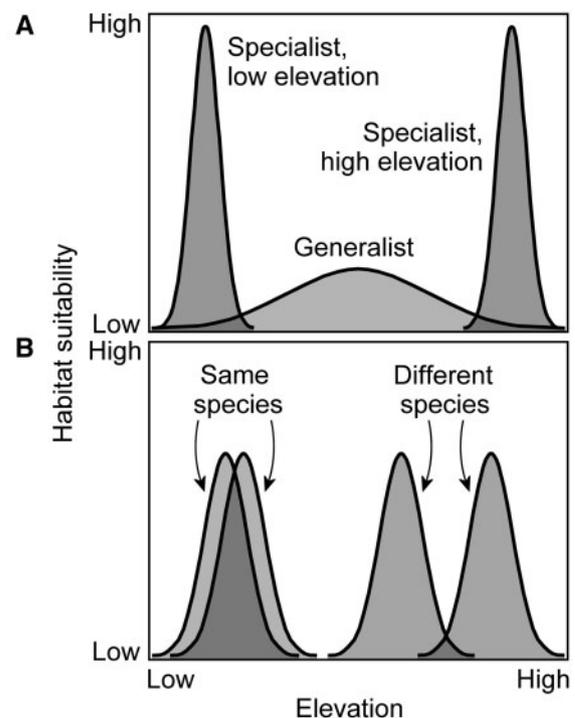
### Species Niches and Distributions

Energy is used by individuals of species for maintenance, growth, and reproduction and limited available energy can limit the number of individuals supported (Brown et al. 2004). Extinction probability is related to population size, and so if richness is higher, on average, fewer individuals of each species may be supported, increasing extinction risk for species with fewer individuals (Hurlbert and Stegen 2014). Over sufficient time, a relationship between energy and richness may be expected. A 1-dimensional model of an environmental gradient was used by Hurlbert and Stegen (2014) to simulate effects of energy on species richness, represented by temperature gradients. A zero sum approach was used in some simulations, where increases in the numbers of 1 species implied fewer resources for

another, and in some simulations that constraint was removed, allowing them to quantify the relevance of the zero sum hypothesis. They compared the model predictions to the distribution of a set of rockfish species (*Sebastes* sp.) in the northeastern Pacific. Through an evolutionary approach and using species with niches that mutated, they were able to simulate latitudinal species richness gradients. Their approach also points to another benefit of an individual-based approach merged with evolutionary programming, the relatedness of individuals is fully known, supporting clade analyses (Boone 2010; Hurlbert and Stegen 2014). In general, among their findings is that subclades may take advantage of resources (e.g., energy) through rapid diversification, helping to explain why environmental gradients for specific taxa may not match typical higher level gradients.

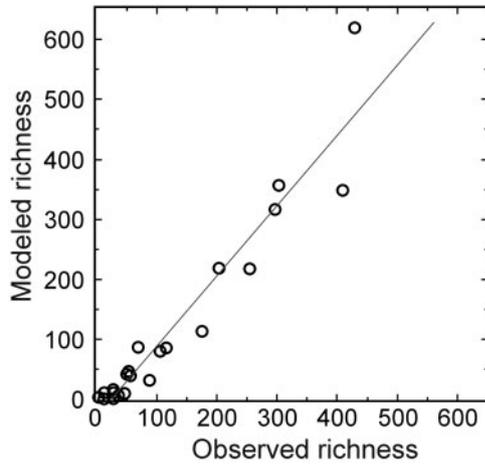
Boone (2010) simulated speciation in a spatially explicit way by linking evolutionary programming with an agent-based representation. MacArthur and Wilson (1967) used island area as a correlate of niche diversity in their famous theory on biodiversity. I sought a somewhat more direct measure. I represented niche hypervolumes that would mutate, and if niches of 2 individuals varied sufficiently, they were considered no longer able to breed and 2 species. To assess the technique, I created an application seeking to simulate speciation of plants on the Galápagos Archipelago. Twenty-two islands comprise the main archipelago. Parts of some islands are lava fields, which were not used in modeling, defining a binary portion of species' niche dimensions. Normalized representations of elevation and slope were the 2 main niche dimensions, derived from a relatively high-resolution (90 m) digital elevation model (SRTM 2004).

Niche dimensions were represented by unit normal curves (Figure 4), allowing the dimensions to be represented by 2



**Figure 4.** A schematic demonstrating plant species niche spaces, showing (A) specialists and generalists relative to elevation. Specialists have small standard deviations in niche dimension and generalists have large standard deviations. Plants with niches that overlap (B) sufficiently in niche dimension are considered the same species and are able to breed in Boone (2010). Plants with niches that do not overlap sufficiently are separate species (from Boone 2010).

parameters, a mean and a standard deviation. When 2 species competed to germinate in a given grid cell, random draws from a uniform distribution were compared with normal curves in each dimension, and if appropriate in both dimensions, the seed



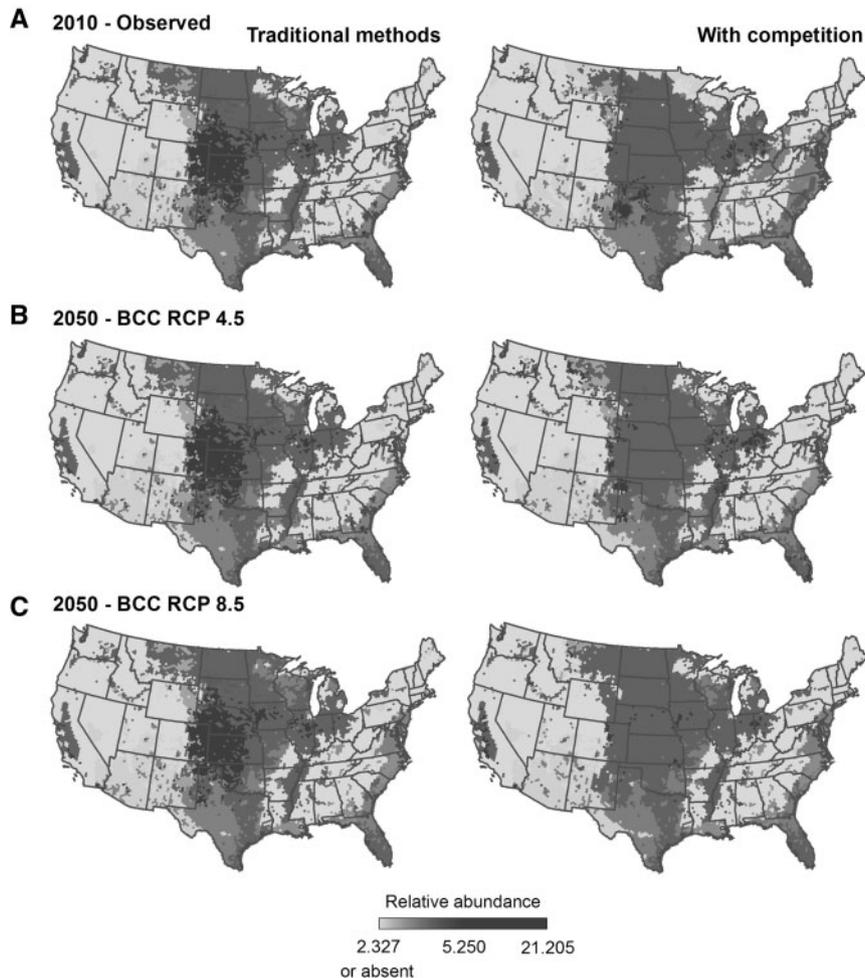
**Figure 5.** Simulated plant species richness plotted against observed richness for 22 islands of the Galápagos Archipelago. A regression line provides reference ( $r=0.957$ , 60 simulations) (from Boone 2010).

germinated. This led species with higher normal curves at a given location to be most successful in competitions. Because *unit* normal curves were used, no species could be well adapted to a wide variety of habitats. Instead, species could be generalists, with short but broad curves, or specialists, with tall but narrow curves (Figure 4A).

All but 2 cells were unpopulated when a simulation was initialized, and cells became unoccupied as plants reached a maximum age and died. Plants of the same species that bred produced seed that may have germinated on a neighboring open cell or one onto which a seed may have fallen during rare dispersal events, if the seed had niche dimensions appropriate for the cell. Plants bred if their niches overlapped sufficiently (Figure 4B). Plants that bred produced seeds that had niche dimensions intermediate of the parents (i.e., averaged mean and standard deviations), with some mutation.

At initialization, 2 plants of the same species occupied 2 randomly selected neighboring cells. At the conclusion of the baseline simulation, that species had evolved to hundreds of species that correlated well with observed species richness on the 22 islands ( $r^2=0.92$ ,  $P<0.001$ , mean of 60 simulations, with 550 native observed species, and 753 simulated species; Figure 5).

A classic example in the use of genetic programming in ecology is provided by Muttill and Lee (2005), who derived an equation predicting coastal algal blooms. Algal blooms can be harmful to coastal ecosystems and the people who inhabit them. For example, red tides



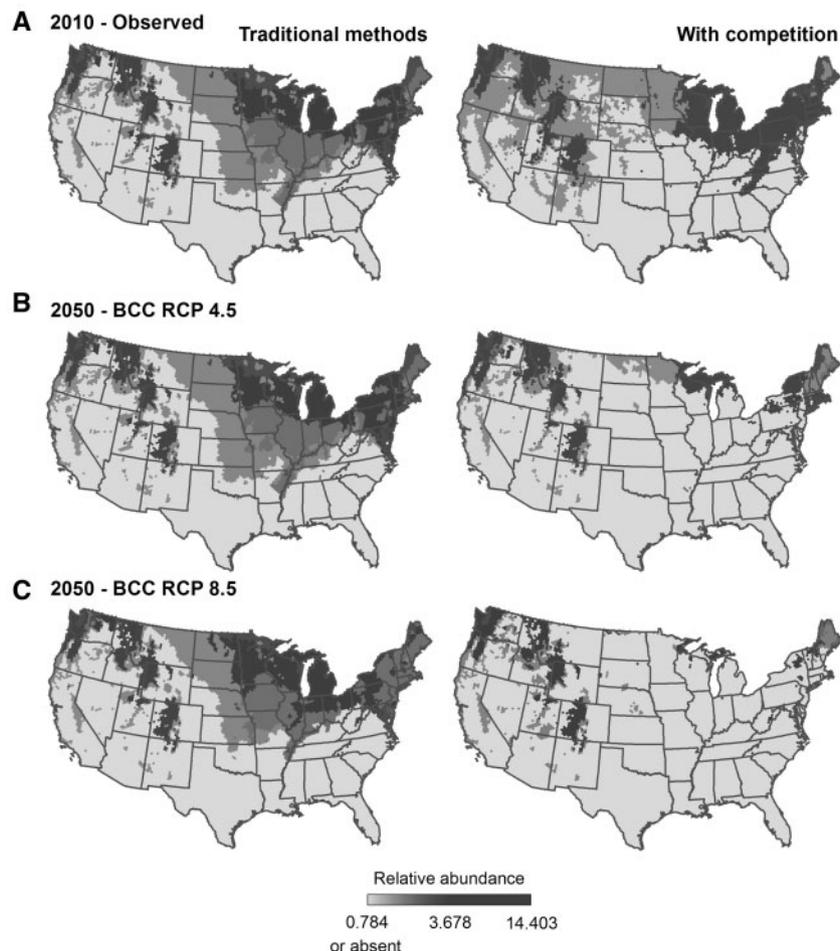
**Figure 6.** The relative abundance of Mourning Dove as represented by a binary regression tree (A, traditional methods), and following simulation to incorporate competition, phenotypic plasticity, and limited dispersal (A, with competition). The relative abundance under BCC 4.5 (B) and BCC 8.5 (C) in 2050 were mapped using traditional methods, and with our methods that incorporate competition, plasticity, and limited dispersal.

can devastate aquaculture. [Muttill and Lee \(2005\)](#) used 3 years of chlorophyll fluorescence, water quality, and other physical data collected every 2 h in a bay near Hong Kong to train a genetic programming algorithm. Variables were defined to be included as candidates and basic operators formed functions in the algorithm. Trees were evolved that attempted to best describe chlorophyll fluorescence. Over many generations, the so-called parse trees competed to predict fluorescence, and steadily improved through selection of the best performing trees to breed related trees. Correlation coefficients between 0.58 and 0.86 were calculated for the equation that was generated by the genetic programming, which was on par with results from artificial neural networks, for example, but more efficient.

Typical niche envelope modeling predicts the distribution of a species based on a set of observed occurrences and spatial surfaces. Tools such as MaxEnt use presence or presence/absence data and their statistical relationships with spatial surfaces to extrapolate occurrences (reviewed in [Elith and Leathwick 2009](#)). When conducting climate change research, for example, analysts extrapolate ranges based on niche envelopes to yield current distributions. If the resulting statistical model includes layers associated with a changing climate, they replace those surfaces with others representing future conditions and reapply the statistical model. That yields a prediction of a species' range under future climate.

Assessing species responses to climate change using niche envelope modeling as generally applied has been criticized in 3 general ways, 1) interspecific interactions are ignored or taken to be constant, 2) species are considered static in genotypes and phenotypes, and 3) individuals are able to disperse unlimited distances ([Davis et al. 1998](#); [Martinez-Meyer 2005](#); [Wiens et al. 2009](#)). I devised a method of forecasting shifts in species ranges that uses evolutionary programming and agent-based modeling to incorporate interspecific interactions, allow phenotypic evolution of niche dimensions, and limit dispersal. Niche dimensions are defined using an occurrence database and biologically relevant spatial data, and many species distributed across a region based on those niches. In an evolutionary process, generations of individuals are simulated to increase niche packing and improve resource partitioning, with the species best adapted to a given site most likely to win in competition to become the occupant. Mirroring the 3 concerns listed, in those simulations, species will compete to occupy landscape patches, and 1) those with the best niche fit will most often succeed in occupying the site, reflecting interspecific competition. Mutations 2) of niche dimensions provide the phenotypic variability that is leveraged by the selective pressure to occupy given patches, and 3) individuals can only disperse into neighboring landscape patches.

My colleagues and I used North American Breeding Bird Survey data from a recent 5 year period as observations. A suite of spatial



**Figure 7.** The relative abundance of American Robin as represented by a binary regression tree (A, traditional methods), and following simulation to incorporate competition, phenotypic plasticity, and limited dispersal (A, with competition). The relative abundance under BCC 4.5 (B) and BCC 8.5 (C) in 2050 were mapped using traditional methods, and with our methods that incorporate competition, plasticity, and limited dispersal.

surfaces representing potentially relevant biophysical variables, such as the BIOCLIM collection (Booth et al. 2014) and land cover were generalized to 635 km<sup>2</sup> EMAP hexagons (Diaz-Ramos et al. 1996), of which there were more than 13,000 for the coterminous United States. Binary regression trees were created that described the relative abundances of 145 species. The surfaces and the trees were read into an agent-based model. For every hexagon for a given species, a local copy of the tree could be traversed to identify the predicted relative abundance. Species competed to occupy local and neighboring hexagons, with the outcome of the competition between 2 species potentially influenced by their relative abundances, depth within a tree (relationships reflected in splits deep in trees presumably represent more specialized adaptation and should outcompete generalists), and occupancy; species already occupying a hexagon may be favored to continue occupation. Note that with these methods, the distributions of breeding birds that do not have surfaces sensitive to climate in their spatial models may still have ranges that shift in the future because of changing competition pressures from birds that are sensitive to a changing climate, as in reality.

In simulations, the structures of the trees were static, but the values used at splits within the tree for a given species within a given hexagon slowly changed using an evolutionary programming approach. Mutated agents competed to occupy a hexagon and neighboring hexagons, representing limited dispersal. Species adapted their phenology to local conditions as they competed to occupy hexagons.

Incorporating competition, phenotypic plasticity, and limited dispersal caused sometimes large differences in species relative abundance distributions under future climate. Two species provide examples, using the RCP 4.5 and 8.5 pathways and Beijing Climate Center Climate System Model results (Wu 2012). Mourning dove *Zenaidura macroura* are distributed through the coterminous United States, with their relative abundance greatest in the central part of the country and along the east coast (Figure 6A, traditional). With competition and plastic phenotypes included through evolutionary programming, plus limited dispersal, the distribution of doves changes, with less area of the highest relative density but more medium density areas along the east coast (Figure 6A, with competition). Under a changing climate, the range of doves is not projected to change if traditional methods are used, but with competition included, their range shrinks and relative abundance may decrease (RCP 4.5, Figure 6B). Areas of highest relative abundance shrank when RCP 8.5 was used as input to the simulation. American robin *Turdus migratorius* are summer breeders in all but the southernmost portions of the coterminous United States, and are most common in intermountain areas in the west and northcentral and northeastern states (Figure 7A, traditional). Incorporating competition led to expansion in the highest relative density areas and shifts in the medium relative densities (Figure 7A, competition). Whereas ranges of American Robin changed in small ways only using traditional envelope modeling methods (Figure 7B,C, traditional), with competition included, the ranges of robins shrank markedly (Figure 7B,C, competition).

## Concluding Remarks

Several case studies were introduced. Many other applications of evolutionary computation in zoology and biology are available [e.g., Houser et al. (1999) regarding dolphin hearing; Hirasawa et al. (2001) using ant behavior in methodological queries; speciation (Ashlock and von Konigsow 2008) and species ranges (Ashlock

et al. 2006)]; tools for education, such as Dawkins' biomorphs (Dawkins 1996), Wilinski's sunflower biomorphs (Nichols and Wilensky 2006) packaged with NetLogo (Wilensky 1999), and Sims' revolutionary evolving creatures (Sims 1994) and more recent treatments (Taylor and Massey 2001); insect physiological attributes (Downing 1997, Maron 2004); queries regarding plankton (Whigham and Recknagel 2001, Recknagel et al. 2013), and phylogenetic reconstruction (Cancino and Delbem 2010)].

Evolutionary programming is an attractive approach because of its simplicity, and the flexibility of objective functions in evolutionary programming. Mutation and selection may apply to phenotypes directly, without the need to incorporate crossover and other approaches that are faithful to biological responses. This is likely to reduce efficiency in locating optima (Bäck 1996), but the simplicity and flexibility is attractive. It is also intuitive to merge evolutionary programming with agent-based modeling. At its core, most evolutionary computational analyses may seemingly be considered agent-based, in that individual solutions or individual agents are competing with others to improve fitness (Railsback and Grimm 2011). Such an approach can be extremely flexible. For example, the niche packing and speciation analyses described in Boone (2010) may be used to explore island biogeography; in that paper individual islands were removed sequentially and richness re-evolved to deduce the influence of each island on richness on the other islands. Cladistics and the taxon cycle may be explored, in so far as the lineage of every individual in the simulation is fully known (Hubbell 2001). Niche dimensions are defined explicitly for organisms, but their simulated distributions may be over less area because of competition, allowing fundamental and realized niches to be compared. Moreover, the relevance of neutral versus niche paradigms may be studied, in that species occupying a given space can be a zero-sum game and that abundances can be fully tallied (Hubbell 2001). The importance of order of colonization on outcomes may be investigated through changes to initial conditions. Perhaps most importantly, the community structure is flexible. If a simulation is run for a few generations, resource partitioning is poorly developed and may represent a disturbed area. If the simulation is run for a long period, niches are tightly packed and may represent a long-established and stable community. For example, the resistance of different communities to invasive species or variation in the attributes of invasive species may be quantified (Boone 2010).

## Acknowledgments

Funding and support was provided by the National Science Foundation (Macrobiology Grant 1241583). My thanks to the Guest Editor, G. Wang, for his assistance and thanks to 2 anonymous reviewers, whose comments helped improve the manuscript.

## References

- Alander JJ, 1994. *An Indexed Bibliography of Genetic Algorithms in Ecology*. Vaasa, Finland: University of Vaasa.
- Ashlock D, Cottenie K, Carson L, Bryden KM, Corns S, 2006. An evolutionary algorithm for the selection of geographically informative species. In: *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*. New York: IEEE. 1–7.
- Ashlock D, von Konigsow T, 2008. Evolution of artificial ring species. In: *IEEE Congress on Evolutionary Computation and IEEE World Congress on Computational Intelligence*. New York: IEEE. 653–659.

- Autumn K, Sitti M, Liang YA, Peattie AM, Hansen WR et al. 2002. Evidence for van der Waals adhesion in gecko setae. *Proc Natl Acad Sci USA* **99**: 12252–12256.
- Axelrod R, 1997. *The Complexity of Cooperation: Agent-based Models of Competition and Collaboration*. Princeton (NJ): Princeton University Press.
- Bäck T, 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford: Oxford University Press.
- Bäck T, Hammel U, Schwefel H-P, 1997. Evolutionary computation: comments on the history and current state. *IEEE Trans Evol Comput* **1**:3–17.
- Bäck T, Rudolph G, Schwefel H-P, 1993. Evolutionary programming and evolutionary strategies: similarities and differences. In: Fogel DB, Atmar W, editors. *Proceedings of the Second Conference on Evolutionary Programming*. San Diego (CA): Evolutionary Programming Society. 11–22.
- Barta Z, Szekely T, 1997. The optimal shape of avian eggs. *Funct Ecol* **11**: 656–662.
- Bennett DA, Tang W, 2006. Modeling adaptive, spatially aware, and mobile agents: elk migration in Yellowstone. *Int J Geogr Inf Sci* **20**:1039–1066.
- Bentley P, 1999. *Evolutionary Design by Computers*. Burlington (MA): Morgan Kaufmann.
- Beyer H-G, Schwefel H-P, 2002. Evolution strategies: a comprehensive introduction. *Nat Comput* **1**:3–52.
- Boone RB, 2010. Simulating species richness using agents with evolving niches, with an example of Galápagos plants. *Int J Ecol* **2010**: Article ID 150606. doi:10.1155/2010/150606.
- Boone RB, Galvin KA, 2014. Simulation as an approach to social-ecological integration, with an emphasis on agent-based modeling. In: Manfredo MJ, Vaske JJ, Rechkemmer A, Duke EA, editors. *Understanding Society and Natural Resources*. Dordrecht (The Netherlands): Springer Netherlands. 179–202.
- Boone RB, Thirgood SJ, Hopcraft JGC, 2006. Serengeti wildebeest migratory patterns modeled from rainfall and new vegetation growth. *Ecology* **87**: 1987–1994.
- Booth TH, Nix HA, Busby JR, Hutchinson MF, 2014. BIOCLIM: the first species prediction modelling package, its early applications and relevance to most current MaxEnt studies. *Divers Distribut* **20**:1–9.
- Brown JH, Gillooly JF, Allen AP, Savage VM, West GB, 2004. Toward a metabolic theory of ecology. *Ecology* **85**:1771–1789.
- Cancino W, Delbem ACB, 2010. A multi-criterion evolutionary approach applied to phylogenetic reconstruction. In: Korosec P, editor. *New Achievements in Evolutionary Computation*. Croatia: InTech. 135–156.
- D'Angelo D, Howard LM, Meyer JL, Gregory SV, Ashkenas LR, 1995. Ecological uses for genetic algorithms: predicting fish distributions in complex physical habitats. *Can J Fish Aquat Sci* **52**:1893–1908.
- Darwin C, 1859. *On the Origin of Species by Means of Natural Selection*. 1st edn. London: John Murray.
- Davis AJ, Jenkinson LS, Lawton JH, Shorrocks B, Wood S, 1998. Making mistakes when predicting shifts in species range in response to global warming. *Nature* **391**:783–786.
- Dawkins R, 1996. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*. New York (NY): W.W. Norton and Company.
- Deb K, Goel T, 2001. A hybrid multi-objective evolutionary approach to engineering shape design. In: Zitzler E, Thiele L, Deb K, Coello CA, Corne D, editors. *Evolutionary Multi-Criterion Optimization. EMO 2001*. Lecture Notes in Computer Science, Vol. **1993**. Berlin (Germany): Springer. 385–399.
- Dermody BJ, Tanner CJ, Jackson AL, 2011. The evolutionary pathway to obligate scavenging in *Gyps* vultures. *PLoS One* **6**:e24635.
- Diaz-Ramos S, Stevens DL Jr, Olsen AR, 1996. *EMAP Statistical Methods Manual*. Corvallis (OR): Environmental Monitoring and Assessment Program EPA/620/R-96/002.
- Downing K, 1997. The emergence of insect protandry: a “natural” evolutionary computation application. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*. New York (NY): IEEE. 325–330.
- Eberhart R, Kennedy J, 1995. A new optimizer using particle swarm theory. In: *Sixth International Symposium on Micro Machine and Human Science*. New York: IEEE. 39–43.
- Eiben AE, Smith JE, 2003. *Introduction to Evolutionary Computing*. Berlin (Germany): Springer-Verlag.
- Elith J, Leathwick JR, 2009. Species distribution models: ecological explanation and prediction across space and time. *Annu Rev Ecol Evol Syst* **40**: 677–697.
- Flasch O, 2014. *A Friendly Introduction to RGP*. Gummersbach (Germany): Cologne University of Applied Sciences.
- Fogel DB, 1998. *Evolutionary Computation: The Fossil Record*. Hoboken (NJ): Wiley-IEEE Press.
- Fogel DB, 1996. Evolutionary programming: an introduction and some current directions. *Statist Comput* **4**:113–129.
- Fogel DB, 2006. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Hoboken (NJ): John Wiley & Sons.
- Fogel DB, Fogel LJ, 1996. An introduction to evolutionary programming. In: Alliot, J-M, Lutton E, Ronald, EMA, Schoenauer, M, Snyers D editors. *AE '95 Selected Papers from the European conference on Artificial Evolution*. London (UK): Springer-Verlag. 21–33.
- Fogel L, Owens AJ, Walsh MJ, 1966. *Artificial Intelligence through Simulated Evolution*. New York (NY): Wiley.
- Gardner M, 1970. Mathematical games: the fantastic combinations of John Conway's new solitaire game “life”. *Sci Am* **223**:120–123.
- Garnier S, Gautrais J, Theraulaz G, 2007. The biological principles of swarm intelligence. *Swarm Intel* **1**:3–31.
- Geim AK, Dubonos SV, Grigorieva IV, Novoselov KS, Zhukov AA, 2003. Microfabricated adhesive mimicking gecko foot-hair. *Nat Mater* **2**:461–463.
- Griffin WA, 2006. Agent-based modeling for the theoretical biologist. *Biol Theory* **1**:404–409.
- Grimm V, Railsback SF, 2006. Agent-based models in ecology: patterns and alternative theories of adaptive behaviour. In: Billari FC, Fent T, Prskawetz A, Scheffran J, editors. *Agent-Based Computational Modelling*. Heidelberg (Germany): Physica-Verlag HD. 139–152.
- Hamblin S, 2013. On the practical usage of genetic algorithms in ecology and evolution. *Methods Ecol Evol* **4**:184–194.
- Haykin SS, 2009. *Neural Networks and Learning Machines*. New York (NY): Prentice Hall/Pearson.
- Hirasawa K, Okubo M, Katagiri H, Hu J, Murata J, 2001. Comparison between genetic network programming (GNP) and genetic programming (GP). In: *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Cat. No.01TH8546. New York: IEEE. 1276–1282.
- Holland JH, 1962. Outline for a logical theory of adaptive systems. *J ACM* **9**: 297–314.
- Holland JH, 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge (MA): MIT Press.
- Holland JH, Miller JH, 1991. Artificial adaptive agents in economic theory. *Am Econ Rev* **81**:365–370.
- Holzinger K, Palade V, Rabadan R, Holzinger A, 2014. LNCS 8401: Darwin or Lamarck? Future challenges in evolutionary algorithms for knowledge discovery and data mining. *LNCS 8401*:35–56.
- Houser DS, Helweg DA, Chellapilla K, Moore PWB, 1999. Creation of a biomimetic model of dolphin hearing through the use of evolutionary computation. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Cat. No. 99TH8406. New York: IEEE. 496–502.
- Hubbell SP, 2001. *The Unified Neutral Theory of Biodiversity and Biogeography*. Princeton (NJ): Princeton University Press.
- Hurlbert AH, Stegen JC, 2014. When should species richness be energy limited, and how would we know? *Ecol Lett* **17**:401–413.
- Hutchinson JMC, 2000. Three into two doesn't go: two-dimensional models of bird eggs, snail shells and plant roots. *Biol J Linn Soc* **70**:161–187.
- Jaffe K, 1999. On the adaptive value of some mate selection strategies. *Acta Biotheor* **47**:29–40.
- Kerr B, Riley MA, Feldman MW, Bohannan BJM, 2002. Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors. *Nature* **418**: 171–174.

- Kicinger R, Arciszewski T, De Jong K, 2005. Evolutionary computation and structural design: a survey of the state-of-the-art. *Comput Struct* 83: 1943–1978.
- Koza JR, Bennett FH, Andre D, Keane MA, Dunlap F, 1997. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Trans Evol Comput* 1:109–128.
- Koza JR, 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press.
- Langdon WB, Poli R, 2013. *Foundations of Genetic Programming*. New York (NY): Springer Science & Business Media.
- Lohn JD, Hornby GS, 2006. Evolvable hardware using evolutionary computation to design and optimize hardware systems. *IEEE Comput Intell Mag* 1: 19–27.
- MacArthur RH, Wilson EO, 1967. *The Theory of Island Biogeography*. Princeton (NJ): Princeton University Press.
- Maron M, 2004. *Evolution of Industrial Melanism: A Spatial, Predator–Prey Genetic Algorithm Project for Artificial Life*. UK: Sussex.
- Martinez-Meyer E, 2005. Climate change and biodiversity: some considerations in forecasting shifts in species' potential distributions. *Biodivers Inform* 2:42–55.
- Mebane WR Jr, Sekhon JS, 2011. Genetic optimization using derivatives: the rgenoud package for R. *J Stat Softw* 42:1–26.
- Mitchell M, 1998. *An Introduction to Genetic Algorithms*. Cambridge (MA): MIT Press.
- Mullen K, Ardia D, Gil D, Windover D, Cline J, 2011. 'DEoptim': an R package for global optimization by differential evolution. *J Stat Softw* 40:1–26.
- Muttill N, Lee JHW, 2005. Genetic programming for analysis and real-time prediction of coastal algal blooms. *Ecol Model* 189:363–376.
- Nichols N, Wilensky U, 2006. *NetLogo Sunflower Biomorphs Model*. Center for Connected Learning and Computer-based Modeling. Evanston, Illinois: Northwestern University [cited 2017 October 13] Available from: <http://ccl.northwestern.edu/netlogo/models/SunflowerBiomorphs>.
- Nunkesser R, 2008. *RFreak: an R Package for Evolutionary Computation*. Technical Report, SFB 475. Dortmund (Germany): Technische Universität Dortmund.
- OMPC (Our Molecular Physics Correspondent). 1972. What is random packing? *Nature* 239:488–489.
- Parpinelli RS, Lopes HS, 2011. New inspirations in swarm intelligence: a survey. *Int J Bio-Inspir Comput* 3:1.
- Peck SL, 2004. Simulation as experiment: a philosophical reassessment for biological modeling. *Trends Ecol Evol* 19:530–534.
- Railsback SF, Grimm V, 2011. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton (NJ): Princeton University Press.
- Recknagel F, Ostrovsky I, Cao H, Zohary T, Zhang X, 2013. Ecological relationships, thresholds and time-lags determining phytoplankton community dynamics of Lake Kinneret, Israel elucidated by evolutionary computation and wavelets. *Ecol Model* 255:70–86.
- Reynolds CW, 1987. Flocks, herds, and schools: a distributed behavioral model. *Comput Graph* 21:25–34.
- Roberts SC, Howard D, Koza JR, 2001. Evolving modules in genetic programming by subtree encapsulation. In: Miller J, Tomassini M, Lanzi PL, Ryan C, Tettamanzi AGB et al. editors. *Genetic Programming, EuroGP 2001*. Lecture Notes in Computer Science, Vol. 2038. Berlin (Germany): Springer. 1–175.
- Salustowicz R, Schmidhuber J, 1997. Probabilistic incremental program evolution. *Evol Comput* 5:123–141.
- Schwefel H-P, 1965. *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik*. Master's thesis, Technical University of Berlin.
- Schwefel H-P, 1975. *Evolutionsstrategie und numerische Optimierung*. Dissertation, Technical University of Berlin, Germany.
- Scrucca L, 2013. GA: a package for genetic algorithms in R. *J Stat Softw* 53: 1–37.
- Sims K, 1994. Evolving virtual creatures. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH '94*. New York (NY): ACM Press. 15–22.
- Smith RE, Dike BA, Mehra RK, Ravichandran B, El-Fallah A, 1999. Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft. *Comput Methods Appl Mech Eng* 186:421–437.
- Smith JA, Deppe JL, 2008. Simulating the effects of wetland loss and inter-annual variability on the fitness of migratory bird species. *IEEE Int Geosci Remote Sens Symp* 4:838–841.
- SRTM. 2004. *Shuttle Radar Topography Mission: Mapping the World in 3 Dimensions*. [cited 2007 April 3] Available from: <http://www2.jpl.nasa.gov/srtm/> (<http://srtm.usgs.gov> when published).
- Stockwell D, Peters D, 1999. The GARP modelling system: problems and solutions to automated spatial prediction. *Int J Geogr Inf Sci* 13:143–158.
- Taylor T, Massey C, 2001. Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artif Life* 7:77–87.
- Tokui N, Iba H, 2000. Music composition with interactive evolutionary computation. In: *Proceedings of the Third International Conference on Generative Art*, Milan. 17:215–226.
- VITO. 2002. *VEGETATION Home Page and Catalogue*. Belgium: Centre de Traitement des Images Vegetation [cited 2017 October 13] Available from: <http://www.vgt.vito.be/indexstart.htm>.
- Wagner GP, Altenberg L, 1996. Perspective: complex adaptations and the evolution of evolvability. *Evolution* 50:967.
- Whigham PA, 1995. Inductive bias and genetic programming. In: *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. New York (NY): IEEE. 461–466.
- Whigham PA, 2000. Induction of a marsupial density model using genetic programming and spatial relationships. *Ecol Model* 131:299–317.
- Whigham PA, Recknagel F, 2001. An inductive approach to ecological time series modelling by evolutionary computation. *Ecol Model* 146:275–287.
- Wiens JA, Stralberg D, Jongsomjit D, Howell CA, Snyder MA, 2009. Niches, models, and climate change: assessing the assumptions and uncertainties. *Proc Natl Acad Sci USA* 106(Suppl):19729–19736.
- Wilensky U, 1999. *NetLogo. Center for Connected Learning and Computer-Based Modeling*. Evanston: Northwestern University.
- Wu T, 2012. A mass-flux cumulus parameterization scheme for large-scale models: description and test with observations. *Clim Dyn* 38:725–744.
- Xie P, Arkin PA, 1997. Global precipitation: a 17-year monthly analysis based on gauge observations, satellite estimates, and numerical model outputs. *Bull Am Meteorol Soc* 78:2539–2558.