

Emergent Computation and Modelling: Complex organization and Bifurcation within Environmental Bounds (COBWEB)

N. Suh¹, B. Bass^{2*}, E. Chan¹ and N. Toller¹

¹Division of the Environment, University of Toronto, 33 Willcocks Street, Toronto, Ontario M5S 3E8, Canada

²Adaptation & Impacts Research Group, Environment Canada at the University of Toronto, Institute for Environmental Studies, 33 Willcocks Street, Toronto, Ontario M5S 3E8, Canada

ABSTRACT. Currently, emergent computation (EC) is a relatively new approach for understanding ecosystem dynamics. Central to this approach is the idea that high-level ecosystem dynamics emerging from low-level interactions of individual agents. In an effort to further understand this dynamic new field, this paper will examine the conceptual background which researchers use to EC models and review the main classes of EC models: cellular automata, genetic algorithms, classifier systems, and neural networks. Finally, the paper will introduce and discuss the results of trials investigating the boundaries of Complex Organization and Bifurcation within Environmental Bounds (COBWEB) model. COBWEB obeys emergent computation characteristics and is a general simulation platform developed to support a large number of independent agents, each encoded by a GA, in a 2-D environment with resources encoded by a cellular automaton. COBWEB displays non-linear behaviour and is characterized by five main attractors: predator-prey cycling between resources and a dominant agent, agent elimination, unlimited growth of agents, predator-prey cycling with no dominant agent and unpredictability.

Keywords: Agent-based modeling, anticipatory agents, emergent computation, genetic algorithms

1. Introduction

Human influence on earth has resulted in changes that are outpacing the rate of natural change at the local, regional and global scales (Harris, 2002). The prominence of global issues such as greenhouse gases, global warming, loss of biodiversity and alterations in global nutrient cycles has altered the way in which research supports the development of policy. Whereas policy-oriented research focused on problem solving and forecasting with known uncertainties, decisions now have to be made on adapting to the impacts of different scenarios that are generated with some unknown uncertainty for problems that have yet to occur (Moss et al., 2001). A focus on scenarios, impacts and adaptation has forced many disciplines to evaluate their approaches to scientific inquiry and develop new methodologies to assess a new set of demands from the policy community.

Most of the impacts of global change will be first observed in ecosystems as they are integrators of water, energy, information and the interactions of many different species. The goal of this research is to assess how ecosystems might respond to environmental variation and change, how predictable is the response and how stable is the response. This goal is met by using an agent-based simulation based on emergent computation (EC). EC models offer a new approach in simulating the environment-species relationship.

Early on, the quantification of the environment-species relationship was best described through predictive modelling. The development of computer simulations led to the simulations of multiple scenarios “more rigorously scientific, more informative and more useful ecology” (Deadman, 1999; Guisan & Zimmermann, 2000). Not only were simulations cost-effective but they were immensely time-effective since many environmental management plans could be tested numerous times prior to lengthy field experiments. Today, the field of environmental and ecological simulation modelling has expanded considerably. Currently, a wide variety of approaches are available to study different situations defined at different scales of time, space and complexity (Villa & Constanza, 2000).

The initial forays into ecological modelling had to avoid what Olson and Sequiera (1995b) called structure dominance. In ecosystems, it is the interaction between various components that determines the complex dynamics of the system. Not surprisingly, preliminary proposals to study ecosystem dynamics through computer modelling were inadequate for understanding real systems (Taylor & Jefferson, 1995). Simulation models were limited by the following three shortcomings (Taylor & Jefferson, 1995). First, models derived from equations assumed a large population size, potentially ignoring the effects of small population effects such as genetic drift. Second, the number of computations that are necessary to model an organism’s behavior could be too complex to be completely captured by current mathematical tools or they could potentially reduce

* Corresponding author: brad.bass@ec.gc.ca

the speed of the simulation. Finally, these models tended to perform poorly in simulating the non-linear system dynamics, so often observed in nature.

The discovery of chaos and semi-stability in simple population models (Byers et al., 1996; Bass, 1998) lent further evidence that conventional ecological models might not be simulating ecosystem dynamics and adaptation to environmental change. In particular, the discovery of semi-stable attractors suggested the potential for new emergent properties or sudden changes of state as ecosystems adapted to change. These types of changes could not be predicted and added a new degree of uncertainty to conventional models.

The EC approach offers a different approach to conventional ecological simulation. In this methodology, the overall system dynamics are the result of local interactions of independent agents whose behavioral response to its environment is completely encoded in a set of rules represented by an artificial intelligence (AI) engine. Thus each agent in the model makes its own decisions at every relevant time step based on its encoded strategy. Currently, EC has spawned into various fields varying from machine reasoning (Maxion, 1990) to ecology (Stone, 1990). However, within EC there are many subsets and a vast array of different simulation approaches.

These approaches are reviewed beginning with the conceptual framework on which EC models are built and the main categories of artificial intelligence: cellular automata, genetic algorithms, classifier systems, and neural nets. The review will be followed by a description of the Complex Organization and Bifurcation Within Environmental Bounds (COBWEB) software. The COBWEB software is an EC model and is used to simulate the system response to different changes within a variable environment.

2. Conceptual Background of Emergent Computation Models

Olson and Sequeira (1995a) define an emergent computer system as

“A system composed of independent agents, which behave according to explicit instructions. The system exhibits implicit spatial and/or temporal patterns that arise as a result of interactions between these subcomponents and/or between them and their environment. The patterns are apparent at a higher level than the agents, and are not explicitly coded in their specifications.”

The key factor in EC is that the local interactions of agents cause the development of observed patterns and are important in determining the properties of the system. Consequently, these models allow for a bottom-up effect not a top-down effect. In other words, the properties of the system emerge from the interaction between all of the components. The resulting system properties are not deducible from any one agent, but in fact emerge over time. This phe-

nomenon of “emergence” is central to ecology and to the field of EC.

The nonlinearities that result from interaction between agents and between agents and the environment is achieved with EC by simulating general as opposed to specific systems, although models such as COBWEB can be constructed around specific environments and applications. EC models also tend to be non-equilibrium models, which are more realistic for simulating the behavior of any complex systems. These models allow for pockets of order and disorder to coexist in space and multiple attractors over time.

Model building requires trade-offs between generality, reality and precision (Nijkamp & van der Bergh, 1997), and only two of these properties could be included into any model (Guisan & Zimmermann, 2000). Empirical or statistical models are centered on precision and reality as they are used to condense experimental observations not to explain the underlying functions and mechanism. Analytical models are centered on generality and precision and are used to predict responses within a simplified environment. The last model group, centered on reality and generality, are mechanistic or causal models and incorporate EC models to some extent. These models are also used to predict, on the basis of real cause-effect relationships, but it is the theoretical validity of the response that is the main objective.

EC models also tend to focus on resource and direct gradients as opposed to indirect gradients. Resource gradients refer to matter and energy consumed by living organisms (e.g., nutrients). Direct gradients are those natural factors that organisms do not consume but are important for their survival (e.g., pH). Indirect gradients are environmental factors with no direct importance on an organism's survival (e.g., elevation). The effect of indirect gradients will often supersede the effect of different combinations of resource and direct gradients. However, indirect gradient modelling has limited applicability due to the varying mixtures of direct and resource gradients in most regions. Moreover, using direct and resource gradients results in a more general model. EC models that incorporate an environment focus on resource gradients, but indirect gradients can be incorporated to some extent through energy requirements for movement and barriers in the environment.

Ecological models are also concerned with the described niche of the organism and are either focused on the fundamental or the realized niches (Guisan & Zimmermann, 2000). The fundamental niche combines the physiological performance and ecosystem restraints while the realized niche includes biotic interactions and competitive exclusion. Models incorporating the fundamental niche tend to simulate the realized response while static models based on empirical data are used to predict the realized niche. EC models incorporate the fundamental niche and tend to be better suited than empirical models in simulating responses to changing environmental situations.

3. Classes of Emergent Computation

The first and earliest EC research was with John von Neumann's work on self-reproducing automata (Kier & Cheng, 2000). In the 1940s, von Neumann formulated conditions for self-replication in a system with the computational abilities of a full Turing machine (Olson & Sequeira, 1995b). He used a large number of agents, each with 29 possible states and treated pieces of data like biological cells (Taylor & Jefferson, 1995). Just as external factors affect biological cells, the cells in the von Neumann model changed their state in response to the state of surrounding cells (Crutchfield & Mitchell, 1995).

Von Neumann's research led to the development of Cellular Automata (CA). A CA is a dynamic computational system that is discrete in space, time and state (Kier & Cheng, 2000). The rules governing the system-wide behavior of automata emerge on a grid based on a set of rules for each individual grid cell. CA track the progress of ever-changing events and have been applied to various fields of research including gas phenomenon and biological systems (Kier & Cheng, 2000). The rules that the automata are governed by can be extremely simple yet result in quite complex system dynamics (Kier & Cheng, 2000). The work of Garder in 1970 with his CA model known as LIFE is a typical example of simplistic rules encoded by individual agents (Olson & Sequeira, 1995a). In LIFE, the rules encoded by the cellular automata were straightforward IF-THEN situations (e.g., IF a certain situation arose, THEN the automata would react in a particular way).

The next type of EC is based on genetic algorithms (GA). Central to the growth of this field is the concept of evolution by natural selection (Mitchell & Forrest, 1995). GAs are computational models of evolution and grew principally through the work of John Holland (Mitchell & Forrest, 1995). They consist of bit strings of 0's and 1's that encode an agent's solution to a particular problem (Crutchfield & Mitchell, 1995). The bit strings act as search algorithms that evolve solutions to optimization problems and pass the most suitable solutions to the next generation (Crutchfield & Mitchell, 1995). A GA simulation mimics biological natural selection. The artificial environment will select the agents with fitter bit strings, or "genes" to produce more offspring.

In various models, three situations can arise when agents produce offspring by using genetic operators. The first scenario is crossover between two bit strings that resembles sexual recombination. As in natural systems, this crossover introduces variability into the system and is often programmed to be more probabilistically favored. The second situation is that mutation could change 0's into 1's or vice versa. Mutations are beneficial in a model as it can retain or regenerate strategies that are lost by the probability of the artificial environment. This re-emergence is analogous to the function of biodiversity (Bass et al., 2002). Finally, although rarely implemented nowadays, inversion could reverse small sections of the bit string to change the

order of the 0's and 1's (Mitchell & Forrest, 1995).

The advantage of using GAs over conventional search algorithms is that the GA search is parallel (Olson & Sequeira, 1995a); a population of solutions is tested simultaneously increasing the probability to finding the best, although not necessarily the most optimal solution. GAs have been applied in economic models, machine learning, social system models and ecological models (Mitchell & Forrest, 1995). Although recent research with GAs resembles Holland's original work, there are notable divergences, and the term evolutionary computation has been introduced recently to encapsulate the various different members (Mitchell & Forrest, 1995).

A prominent example of an EC model using GA is Holland's model, Echo (Mitchell & Forrest, 1995). Echo is an ecological system model simulating agent-agent and agent-environment interactions to determine high level system phenomenon (e.g., flow of resources, cooperation). Each agent's genome encodes all of the instructions, which the agent follows to interact with other agents and the artificial environment. In addition, Mitchell and Forrest (1995) pointed to four key advances Echo contributed to the field of GA. They are:

- 1) the explicit encoding of resources into the system,
- 2) the possibility that an agent's fitness is a function of geographical location,
- 3) the possibility of various types of agent interactions, and
- 4) that fitness is endogenous to the model.

In addition to its normal functions, GAs can also be combined with a CA to form another EC field of research often referred to as Artificial Life (ALife) (Crutchfield & Mitchell, 1995). Spearheaded in the 1980s by Chris Langton, the ALife field is extremely broad and has been used to study various subjects from the origins of life to nanotechnology (Olson & Sequeira, 1995b). Despite its broad usage, different models within the field are linked by the concept of emergence focusing on patterns such as self-sustaining loops, food webs, and cooperation (Downing, 1998) and are used in the study of evolution, and social structures and behaviour (Olson & Sequeira, 1995b). An example of an ALife simulation is Booth's GECKO program. In this model, based on biological interactions, plants are able to convert resources to carbohydrates (Downing, 1998).

The third class of EC is the classifier system, which can be considered an advancement of the GA subtype. In simplest terms, a classifier can be thought of as one super agent combining several different GAs. Subsequently, a bidding process is required for the selection of the strategy that the super agent will use at any given time, and classifier systems also require a performance system and credit assignment system (Westerdale, 2001). Eventually, those strategies that are not used are discarded.

Classifier systems are composed of three layers; the

lowest being the performance system. This first system refers to the agent interaction with the environment and the bidding of the various strategies within the super agent to be the strategy that the super agent chooses to follow at that time step. (Westerdale, 2001; Olson & Sequeira, 1995a). Next the credit assignment system, the second layer, uses a bucketbrigade algorithm to assess the strength of the classifier, based on the agent's previous experiences (Olson & Sequeira, 1995a). The classifier with the greatest strength is interpreted as having the best fitness and consequently will be chosen as the classifier of choice. Finally, the highest level of the classifier system is the genetic algorithm that, over time, will delete those classifiers with low strength and replace them by applying genetic operators to modify high strength classifiers to yield processes such as mutation or cross-over (Westerdale, 2001; Olson & Sequeira, 1995a). Because of these properties, classifier systems are often used in complex environments that present new situations periodically over time.

Krebs and Bossel (1997) used a classifier system to simulate the learning of a single animat (artificial animals) in different environments. The animats exist as single organisms in an ecological context and are encoded by classifiers to link perception with suitable action. They gain energy through consuming food and lose it for movement, collisions and learning. The animats learned to survive by recognizing scenarios and acting appropriately to increase long-term fitness. After training the animats, Krebs and Bossel (1997) then placed them in a different environment to measure their development.

The last category of EC to be reviewed is a neural net. This EC model is a complex array of layers designed to resemble the neural pathways in the brain (Long et al., 1998). Similar to GAs and classifier systems, the neural network works in parallel and due to its overall construction, it is most often used in pattern-recognition and classification devices (Olson & Sequeira, 1995a). It incorporates the basic principles of neuron in the brain has the capability to "learn" from experience. Some of the benefits to using neural nets in ecological simulations are that neural nets tend to be flexible and can respond to new stimuli. A downfall to using neural nets is the need to balance the time needed for a simulation and the size of the network (Long et al., 1998).

Currently, there are two types of popular neural nets. The first is the well-known multi-layer feed-forward neural network (a.k.a. using backpropagation algorithms (BPN)) while the second to be discussed is self-organizing and are formulated using adaptive resonance theory (ART) (a.k.a. using Kohonen self-organizing mapping) (Foody, 1999). The second is closer to an EC model.

The self-organizing neural network takes into account the greater complexity of neural functions, circumvents the need for supervised learning, and incorporates competition and associative learning. The generic set-up of an ART net is very different to the BPN and allows grouping of similar objects. To begin, a self-organizing/self-learning neural net-

work consists of only 2 layers, the input layer and the output layer (Lek & Guegan, 1999). The input layer represents the input features and the output layer represents the categories.

The external stimuli are received by a node in the input layer, which sends a bottom-up weighted signal to the output layer. Subsequently, the bottom-up signal is complemented by top-down signals to prevent coding instabilities that may arise when the same stimuli is presented to the input nodes constantly. The purpose of the top-down signals is to reference the bottom-up signal to previously learned categorizations. If the input resembles a particular category with high strength, the input is classified with that node. It in turn will inhibit the other nodes from categorizing the signal. The adaptability of the system comes from its capability to change classifications with time and exposure to stimuli.

Foody (1999) tested the abilities of a self-organizing feature map neural network to be used for ecological classification and ordering samples. Using vegetative survey data and the unsupervised neural network, he found that his model was in general agreement with observations. Although further validation is required, his findings support the drive to use self-learning systems in ecological modeling.

4. Complex Organization and Bifurcation within Environment Bounds (COBWEB)

4.1 The COBWEB Architecture

COBWEB is a general simulation platform used to study agent adaptation to a wide variety of environmental conditions. Similar to the Echo model, each agent is encoded by a GA. COBWEB's GAs are made up of a random sequence of 0s and 1s that determine the agent's strategy as well as an additional 2-bit memory for storing environmental information. The platform supports multiple heterogeneous agents in the same grid. This heterogeneity allows for competition and physical interactions between the various agents.

In the initial version of COBWEB, the agents move around the grid, consume energy, expend energy, replicate, die and possibly anticipate the future through its use of memory. As soon as enough energy is accumulated, the agent replicates. In addition, the agent cannot see past one square and its interactions are limited to collisions with other agents, immovable rocks and food resources. Consequently, differences between parent and offspring in this first version arise solely through mutation.

The novelty in COBWEB is its construction; it is an object-oriented program and the agents possess memory. The object-oriented structure allows for a separation between the GA of the agents and the other parts of the program. Thus it is possible to replace the GA with a neural net or a classifier system. This also allows for the creation of different types of agents with different properties in addition to their genetic code.

The memory allows the agent to incorporate nongenetic information into its sequence of 1's and 0's allowing it to go beyond its genetics. This is accomplished by giving additional input and output bits not essential for agent survival. At the beginning of the simulation both input and output bits are set to zero but the extra output bit is moved to the extra input bit at each time step, essentially allowing the agent to send memos of its last action to itself at each time step. In a broader context, this acts as a prediction mechanism and allows the agent to alter its future behavior. The additional data also produces a Turing complete GA that is anticipatory in nature (Bass et al., 2002).

The agents move around the grid, consume energy, expend energy, replicate, die and possibly anticipate the future through its use of memory. As soon as enough energy is accumulated, the agent replicates. In addition, the agent cannot see past one square and its interactions are limited to collisions with other agents, immovable rocks and food resources.¹

The agents exist on a two-dimensional grid that contains rocks and resources. The grid's boundaries are hard or can be turned off so that agents can pass through a boundary and emerge on the other side of the grid, a wrapping effect. Resources are dropped randomly into one grid square of the environment at each time step but the spread of the food is controlled by a non-deterministic cellular automaton. Any resource that is not consumed by an agent is able to grow to adjacent grid squares at a rate determined by the experimenter prior to the beginning of the simulation. Since the location of the resources is constantly changing, the environment is in a constant state of flux, and can be used to simulate environmental variability.

COBWEB has 18 variables that the experimenter is able to alter. In extremely broad terms, the variables can be separated into four main categories. The first category deals with physical characteristics of the grid (e.g., dimensions of the grid, hard boundary/environment wraps on itself, number of stones and the number of agents). The second category deals with characteristics of the resources (e.g., initial amount of resources, the rate of resource growth, energy obtained from resources and the probability of a new resource growing in an empty space). The third category deals with the energy requirements and expenditures of the agents (e.g., initial energy at birth, energy required to breed, step, step into a rock, and to turn). Lastly, the fourth category deals with miscellaneous aspects of the agent characteristics (e.g., mutation rate and agent memory). The output of the simulation is the number of grids occupied by food, the number of agents, the average agent energy and the total agent energy.

The constraint on the agents in COBWEB is energy, which determines if and when an agent replicates. Those agents that do not have strategies that select for sufficient accumulation of energy will, therefore, be removed from the simulation. Thus, those strategies that are left by the system are a system-wide response to anticipated future events. Despite the apparently simple construction of the program, COBWEB exhibits non-linear behaviour, semi-stability and multiple attractors.

By far the most common attractor is a predator-prey cycling between a dominant agent, representing one particular strategy, and resources. However, other attractors include elimination of all agents, unlimited growth of agents, predator-prey cycling with no dominant agent and unpredictability. The emergence of the first three attractors can be predicted with approximately 10-20% uncertainty from a specific range of initial parameter values whereas the emergence of the fourth and fifth attractors cannot be predicted from the initial parameter settings. The fifth attractor is characterized by bifurcation points, where the system trajectory can be vastly different from one simulation to the next without any parameter changes.

COBWEB is visual, and there are multiple attractors in space although these have not been classified. However, visual observation has detected that pockets of local spatial clusters emerge with a relative stability for several hundred time steps and then dissipate. In some instances the patterns of agents are closely tied to the location of resources and may reflect adaptation to the environment, but this will require further testing.

4.2 Methodology

COBWEB is used to assess how a system responds to environmental change and variation. Variation is modelled with the CA. Environmental change is introduced through changing a select set of parameters. The methodology consisted of varying the selected parameters individually and in combination. During each simulation, the following conditions were held constant: the dimensions of the environment, no hard boundaries (an agent hitting a wall emerges on the other side of the grid) and agent memory was turned on.

High, low and intermediate values were chosen to act as reference guides during the systematic increase and decrease of other parameters. The high and low values were chosen by arbitrarily estimating extreme limits, which could theoretically be observed in a simple environment. The intermediate values were chosen if the values displayed the common predator-prey fluctuations observed in the majority of runs and were between the high and low values.

¹In the most recent version of COBWEB, the agents also have a communication buffer to store information from other agents and both buffers can be varied in size. In addition to communication, the agents can also engage in sexual reproduction with each other.

Table 1. High, Intermediate and Low Values Used as Reference Values during PHASE 1

PHASE 1 Values	# Stones	# Agents	Resource amount	Resource Rate	Resource Growth	Mutation Rate	Initial Energy	Food Energy	Breed Energy	Step Energy	Step Rock Energy	Turn Energy	Turn Right Energy	Turn Left Energy
High	1000	2000	3000	2	1	2	2000	200	200	100	100	100	50	50
Intermediate	300	1000	1000	1	0.5	0.05	1000	200	10	5	5	5	2.5	2.5
Low	1	2	5	0.5	0.5	0.03	2	2	1	1	1	1	0.5	0.5

Table 2. High, Intermediate and Low Values Used as Reference Values during PHASE 2

PHASE 2 Values	# Stones	# Agents	Resource amount	Rate of Resource Growth	Prob. of new Resources	Mutation Rate	Initial Energy	Energy from Resources	Breed Energy	Step Energy	Step Rock Energy	Turn Energy	Turn Right Energy	Turn Left Energy
High	2000	1000	2000	3	2	1	3000	500	100	30	30	30	15	15
Intermediate	1000	100	500	1	1	0.01	2000	100	50	10	10	10	5	5
Low	10	10	20	0.2	0.2	0.01	30	3	0.2	0.5	0.5	0.5	0.25	0.25

During various trials, the intermediate values were the default values while the each parameter was altered between the low and high values at a regular interval. Two experimental phases were developed, representing different initial conditions (Tables 1 and 2), to investigate the temporal effect of altering each parameter individually on three outcomes: agent population, energy and resource availability. The increase, no change or decrease of the long-term agent population and average agent energy for each run were compared to assess the effect of each environmental change on each outcome. In addition, other experiments were conducted to test the temporal effect of altering multiple parameters at the same time. Each simulation lasted for 3000 time steps unless all the agents die prematurely.

4.3 Results and Discussion

4.3.1 Manipulation of Individual Parameters

In both Phases, various parameters were altered individually while the other parameters remained at their default values. The most common characteristic of all experiments was a dramatic decrease in the number of agents at 200 time steps. Once the grid was cleared of vast numbers of agents, the survivors moved more frequently and for longer distances. In addition, after the significant agent mortality, the system exhibited simple, repetitious, upward and downward fluctuations in population and resources characteristic of the first attractor. In the vast majority of cases, certain agents were consistently present, occasionally dominating the grid for short periods. Long-term stability in agent population is representative of the first attractor, while a collapse or crash of the agent population is the second attractor, indicated by an agent population of zero in Table 3.

The results of varying the individual parameters on the long-term agent population and the average agent energy are presented in Table 3. The results that are presented are

representative of a large number of simulations. The two phases reveal many similarities and some startling differences. Overall, a higher number of initial agents was a disadvantage in terms of long-term agent population, average energy and long-term stability. For example, the second attractor, represented by a zero for agent population, emerged several times during Phase I but not during Phase II. However, the third attractor, continual increase in population emerged twice in Phase II.

The results in Table 3 indicate that there are parameter settings that often lead to the emergence of the second and third attractors. A continuously increasing agent population in some Phase II runs could lead to a third attractor, or the population could stabilize. This could not be predicted. There are also some clear indications of adaptation in Table 3. For example, in Phase II, when the energy from resources increased, the average energy per agent decreased. Visually, the agents remained in one spot for long periods of time, usually a strategy that leads to death. However, with the additional energy, the agents could afford to reduce the search for resources, and the system could allow for strategies that normally would be detrimental to long-term survival. The mutation rate may be more favorable to agent population than indicated in Table 3. A subsequent run indicated that at agent populations between 300 and 400, a mutation of rate of 0.65 is most favorable.

4.3.2 Manipulation of Multiple Parameters

In the second stage of experimentation several variables were altered at the same time (Table 4). Following these tests, two different sets of variables were altered in opposite directions to assess the impact. During this section, the 14 possible variables were divided into related groups. In the first alteration all the initial parameters (i.e. number of stones, agents and resources) were raised to their high val-

Table 3. Effects of Individual Parameter Manipulation

	PHASE I			PHASE II		
	Parameter	# Agents	Avg. Agent Energy	Parameter	# Agents	Avg. Agent Energy
# Stones	300-600	80-90	540	10	440	1300
	≥ 700	0		2000	325	1175
Resource Growth Rate	0.5	0		0.5	325	1200
	1.0	80	530	3.0	525	1300
	2.0	170	550			
Mutation Rate	0-0.06	80-90	540	0.25	1300	1000
	0.07	0		0.75	100	1000
Initial Energy	0-800	0		30	110	275
	1000	90	540	3000	400	1875
	2000	140	540			
Energy from Resources	2-100	0		50	350-400	1270
	200	80-90	540	500	350-400	1090
Step Energy	0-2	260-270	540	0.5	450	1340
	2-4	130-140	540	10.0	350-400	1200
	5	80-90	540	30.0	350-400	1200
	>5	0				
Turn Energy	0	Increasing		0.5	Increasing	1240
	5	80-90	670	10-30	300-450	1240
	20	50	540			
	30	0	560			

ues. In the second alteration the resource parameters (i.e. resource growth rate and the probability of new growth) were raised and lowered to their extreme values. In the third alteration the energy parameters (i.e. energy at birth and from resources and energy required to breed, step, step into a rock, turn, turn right and turn left) were raised and lowered to their extreme values.

Interesting trends emerged when the groups of parameters were altered in opposite directions for the Phase I & II settings. The alternating experiments in Phase I indicated that the system could survive, but the agent population was low, if the resource parameters were sufficiently high or if the energy requirements and energy expended was sufficiently low. In Phase II, the second attractor emerged, when all of the resource parameters were set to their low values. However, the emergence of the third attractor is predictable when the energy parameters are set to their low values. The last experiment suggests that the energy parameters hold more weight in determining the course and outcome of runs as opposed to the initial parameters.

4.3.3. Unlimited Growth and the Limits of Prediction

A third set of experiments was conducted to investigate the attractors leading to unlimited growth and unpredictable results. The parameter settings in Table 5 represent two runs with seemingly unlimited growth. The runs were set to 10,000 time steps instead of the usual 3,000 time and the third attractor eventually lead to a system collapse due to

overpopulation. There were no free spaces to accommodate birth of new agents. The system literally collapsed as this situation forced a closure of the program.

Table 4. The Effects of Altering Multiple Parameters

Parameter	PHASE I		PHASE II		
	# Agents	Avg. Agent Energy	# Agents	Avg. Agent Energy	
Initial Parameters	High	90-0	540	450	120
	Low	90	550	275	1300
Resource Parameters	High	180	550	800	1300
	Low	40-0	530	0	
Energy Parameters	High	Survival	530	275	1670
	Low	Low-0		Fills up grid	
Alternating Directions					
Initial Parameters High	Low	- 0		0	
Resource Parameters Low					
Energy Parameters High	Low	- 0		0	
Resource Parameters Low					
Initial Parameters Low	Survival	530	800	1325	
Resource Parameters High					
Initial Parameters Low	Low	- 0		225	1750
Energy Parameters High					

Table 6 shows one example of a set of parameters where program was on a bifurcation point leading to an unpredictable outcome. For the particular set of parameters used in this discussion, there were two possible outcomes that could occur. The first was system survival without any agent dominating the grid (the fourth attractor) while the other possible result was system death. The agents survived only 50% of the time. However, it should be noted that altering certain parameters in small increments nudged the system to either the first or second attractor. For example, when the initial energy was altered to 300, ten consecutive runs produced death. On the other hand, when initial energy was raised to 800, ten consecutive runs produced system survival. These findings are indicative of a fifth attractor that appears to conform to the behavioral properties of a chaotic attractor.

4.3.4. The Effect of varying grid size

The effect of grid size was tested by replicating the runs where the step energy parameter was altered between its high and low values but a 50 × 50 grid size was used in place of the usual 75 × 75 grid. In this initial analysis, the grid size does have an effect on the equilibrium agent number and average agent energy, which is expected given the smaller area of the environment. However, the general trend of increasing or decreasing average agent energy did not appear to be constant. Average agent energy tended to be higher than observed in the larger grid and no regular trend could be ascertained in the smaller grid. In addition, the smaller grid runs did not have the characteristic high rise and fall in agent population and average agent energy that were typical on the large grid, and even small fluctuations were observed only occasionally at the beginning of the run, within the first 200 time steps.

4.3.5. Spatial Clustering

In most simulations, agents with similar strategies tend

to aggregate for fixed periods of time, and then the cluster dissipates. These clusters appear with a large, localized growth in resources. The increased availability of energy appears to fuel the growth of the cluster so that it is maintained for a longer period of time. However, the timing of the clusters appears to be random and do not favor one strategy over another

There are six hypotheses to explain the spatial aggregations of agents:

1. The rapid proliferation of a single agent in rich environmental conditions creates a cluster of similar agents. The agent's strategy is not significant; any agent placed in similar conditions would cluster in the same manner.
2. The grouping of similar organisms favors further clustering of similar agents because it prevents other, mobile, invading organisms from taking away the food in the area.
3. Some combination of environmental factors in the area makes the region favorable for a particular type of agent strategy over the others, i.e. geographic isolation, large food growth, etc. Consequently, that particular strategy tends to be more abundant in that area.
4. Barriers to mobility through a higher number of stones, agents or food or larger energy expenditures to move favored the aggregation of agents. In this case, the formation of spatial clusters appears to be a system adaptation to the barriers to mobility.
5. A lower mutation rate can reduce the variety of agent strategies that survive in the long-term. Lower mutation rates favor single-strategy clusters. A larger variety reduces the chance of one type of agent exhibiting clustering behavior since it is more likely that a different type of agent will invade the cluster.
6. Clustering behavior is enhanced when the energy required to breed is low, initial agent energy is high or energy from food resources is high. Conversely, if the

Table 5. Parameter Values Leading to Unlimited Growth Results, the Third Attractor

Trial #	# Stones	# Agents	Resource amount	Rate of Resource Growth	Prob. of new Resources	Mutation Rate	Initial Energy	Energy from Resources	Breed Energy	Step Energy	Step Rock Energy	Turn Energy	Turn Right Energy	Turn Left Energy
1	100	100	500	1	1	0.01	2000	100	50	10	10	0.5	0.25	0.25
2	300	1000	1000	1	20	0.05	1000	200	10	5	5	5	2.5	2.5

Table 6. Parameter Values Leading to the Fourth and Fifth Attractors

Trial #	# Stones	# Agents	Resource amount	Rate of Resource Growth	Prob. of new Resources	Mutation Rate	Initial Energy	Energy from Resources	Breed Energy	Step Energy	Step Rock Energy	Turn Energy	Turn Right Energy	Turn Left Energy
1	300	50	15	0.05	1	0.15	400	50	10	20	10	5	5	5

energy required to breed and move is high and the other inputs are relatively too low, aggregation is less likely to occur.

Note that these explanations are not mutually exclusive and any combination of these may occur.

4.3.6. The Advantages of Memory and Communication

The more recent versions of COBWEB allow for agents with memory and communication. These characteristics are important in that they allow the agents to utilize information that is not genetic, and in the case of memory, create agents that are anticipatory in nature. An experiment was performed to test the impacts of memory and communication. In tracking a single agent through different simulations, it was found that it did make different decisions with memory and communication when faced with the same decision (Table 7).

Table 7. The Impact of Memory and Communication on Movement

Memory	Communication with other Agents	Facing	Distance to Object	Action
Yes	Yes	south	3	step
No	Yes	south	3	do nothing
Yes	No	south	3	do nothing
No	No	south	3	turn Right

The presence of memory was examined from the perspective of longevity and the number of descendants. In both cases, over 10 simulations, an agent with memory lived longer only 50 % of the time (Figure 1) and where the impact on the number of descendants was significant, it more often favored agents with memory (Figures 2 and 3). In many cases, the superior agents significantly outperformed their counterparts. The only explanation is that the environment developed in such a way as to highly favor either anticipatory agents or agents without memory.

5. Conclusions

COBWEB is a general simulation program where each agent is encoded by a genetic algorithm and the rate of resources is controlled by a non-deterministic cellular automata. During trials to test the boundaries of the program, COBWEB exhibited non-linear behavior as well as five main attractors. The first three attractors could be predicted from the initial parameter values with a relatively high probability (80-90%). The system trajectory of the fourth and fifth attractors could not be predicted from the initial parameter values. In fact, the fourth attractor often emerged

when the system trajectory was characterized by a bifurcation, i.e. the fifth attractor. When parameter values were found that caused the fifth attractor emerge, the system trajectory was very sensitive to minor changes in these parameter values.

An interesting observation periodically found in COBWEB runs was the clustering of the agents. The clustering observed in COBWEB resembled the clustering often observed in self-organized neural networks; thus, the clustering in COBWEB may be indicative of learning by the agents. Agents also appeared to modify their spatial behavior perhaps being indicative of the capacity to anticipate the location of food.

As an EC model, COBWEB exhibited behavior resembling other EC simulations such as sensitivity to resource levels, re-emergence of lost agent strategies, and in keeping with the definition of EC, the system dynamics of COBWEB were found to result from the interactions of low-level factors. COBWEB also allowed for multiple agent strategies and exhibited evolutionary behavior as agents consumed and expended energy. As in other models, most maladapted strategies are removed from the system.

The properties that COBWEB has demonstrated indicate its utility for modelling ecological adaptation to environmental change. It features mobile agents and immobile resources, environmental variability and environmental change. High level properties emerge from the interactions of many individual agents, the system behavior is non-linear and it allows for the existence of multiple attractors and important characteristic of ecological adaptation (Bass, 1998). More recent versions of COBWEB have multiple resources that can be programmed to provide varying amounts of energy to different agents. This version of COBWEB has been used to simulate long-term responses to climate change, as represented by the changing length of different seasons, invasive species and the advantages of impact of memory and communication. Memory and communication with other agents changes the decision that an agent will make in a similar circumstance, and agents with memory are more likely to live longer and have more descendants.

Future work on COBWEB will involve more analysis on the threshold values that trigger different attractors, the advantages of memory and communication, the evolution of cooperation, spatial analysis of the system and agent behavior with distinct preferences. COBWEB is a general simulation platform constructed in the Java computer language. The code is available from Environment Canada by contacting Brad Bass.

Acknowledgments. The authors would like to acknowledge the contributions of past members of the COBWEB team: Jeff Hill, Jin Kang, Ruta Wad, Eric Morgen, James Martens, Kenny Chng, John Choy and Peter Lavelle.

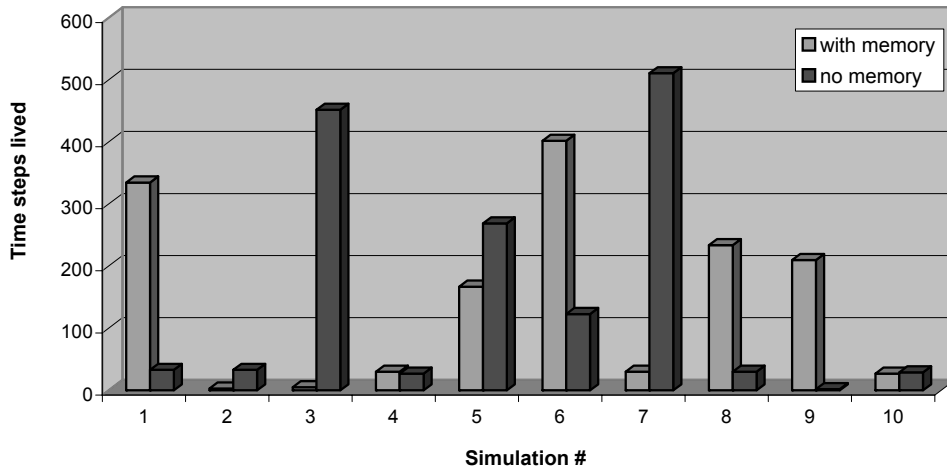


Figure 1. Longevity of Agent with and without Memory.

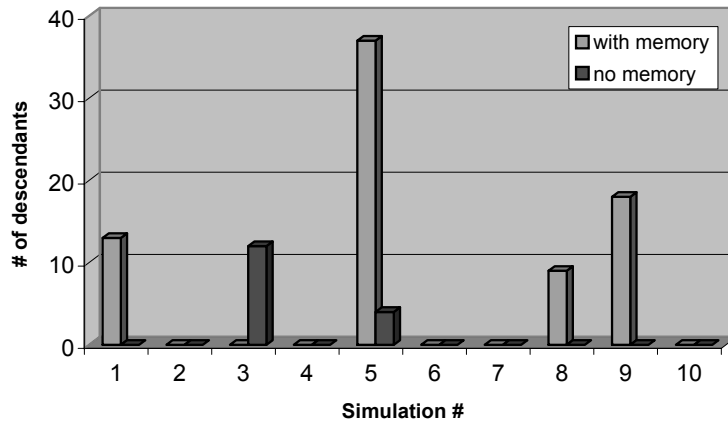


Figure 2. Impact of Memory of Descendants.

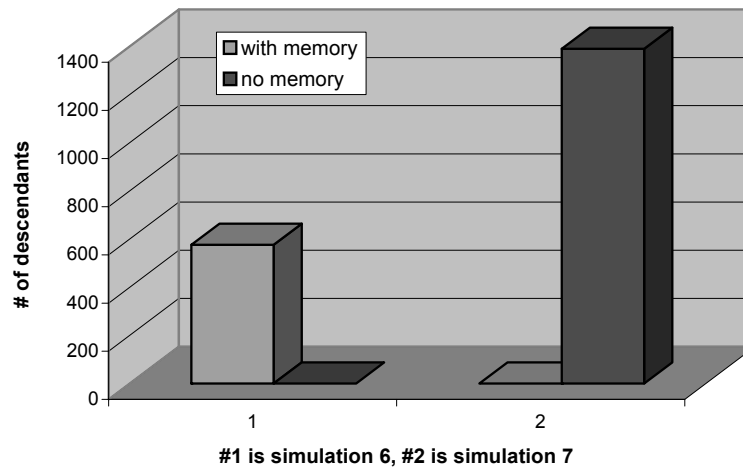


Figure 3. Impact of Memory on Descendants.

References

- Bass, B. (1998). Applying Thermodynamic orientors, in F. Müller and M. Leupelt (Eds.), *EcoTargets, Goal Functions and Orientors*, Springer-Verlag, Berlin, Heidelberg, pp. 193-208.
- Bass, B., Hill, J. and Suh, N. (2002). Simulating adaptation to environmental change: Complexity and orgenetic algorithmized behaviour within environmental bounds (COBWEB). *J. CASYS*, 13, 25-38.
- Bonabeau, E.W. and Theraulaz, G. (1995). Why do we need artificial life?, in C.G. Langton (Ed.), *Artif. Life*, MIT Press, Cambridge, MA, USA, pp. 303-326.
- Crutchfield, J.P. and Mitchell, M. (1995). The evolution of emergent computation. *Proc. Natl. Acad. Sci.*, 92, 10741-10746.
- Deadman, P.J. (1999). Modeling individual behaviour and group performance in an intelligent agent-based simulation of the tragedy of the commons. *J. Environ. Manage.*, 56, 159-172.
- Downing, K. (1998). Using evolutionary computational techniques in environmental modelling. *Environ. Model. Soft.*, 13, 519-528.
- Foody, G.M. (1999). Applications of the self-organising feature map neural network in community data analysis. *Ecol. Model.*, 120, 97-107.
- Guisan, A. and Zimmermann, N.E. (2000). Predictive habitat distribution models in ecology. *Ecol. Model.*, 135, 147-186.
- Harris, G. (2002). Integrated assessment and modelling: An essential way of doing science. *Environ. Model. Soft.*, 17, 201-207.
- Kier, L.B. and Cheng, C.K. (2000). A cellular automata model of an anticipatory system. *J. Mol. Graphics Model.*, 18, 29-32.
- Krebs, F. and Bossel, H. (1997). Emergent value orientation in self-organization of an animat. *Ecol. Model.*, 96, 143-164.
- Lek, S. and Guegan, J.F. (1999). Artificial neural networks as a tool in ecological modelling, an introduction. *Ecol. Model.*, 120, 65-73.
- Long, D.L., Parks, R.W. and Levine, D.S. (1998). An introduction to neural network modeling: merits, limitations, and controversies, in D.L. Long, R.W. Parks, D.S. Levine (Eds.), *Fundamentals of Neural Network Modeling: Neuropsychology and Cognitive Neuroscience*, MIT Press, Cambridge, MA, USA, pp. 3-31.
- Maes, P. (1995). Artificial life as a tool for biological inquiry, in C.G. Langton (Ed.), *Artif. Life*, MIT Press, Cambridge, MA, USA, pp. 135-162.
- Maxion, R.A. (1990). Toward diagnosis as an emergent behaviour in a network system. *Physica D.*, 42, 66-84.
- Mitchell, M. and Forrest, S. (1995). Genetic Algorithms and artificial life, in C.G. Langton (Ed.), *Artif. Life*, MIT Press, Cambridge, MA, USA, pp. 267-289.
- Moss, S., Pahl-Wostl, C. and Downing, T. (2001). Agent-based integrated assessment modelling: The example of climate change. *Integr. Assess.*, 2, 17-30.
- Nijkamp, P. and van der Bergh, J.C.J.M. (1997). New advances in economic modelling and evaluation of environmental issues. *Eur. J. Oper. Res.*, 99, 180-196.
- Olson, R.L. and Sequeira, R.A. (1995a). Emergent computation and the modeling and management of ecological systems. *Comput. Electron. Agric.*, 12, 183-209.
- Olson, R.L. and Sequeira, R.A. (1995a). An emergent computational approach to the study of ecosystem dynamics. *Ecol. Model.*, 79, 95-120.
- Recknagel, F., French, M., Harkonen, P. and Yabunaka, K.L. (1997). Artificial neural network approach for modelling and prediction of algal blooms. *Ecol. Model.*, 96, 11-28.
- Stone, N.D. (1990). Chaos in an individual-level predator-prey model. *Nat. Resour. Model.*, 4, 539-553.
- Taylor, C. and Jefferson, D. (1995). Artificial life as a tool for biological inquiry, in C.G. Langton (Ed.), *Artif. Life*, MIT Press, Cambridge, MA, USA, pp. 1-13.
- Villa, F. and Constanza, R. (2000). Design of multi-paradigm integrating modelling tools fro ecological research. *Environ. Model. Soft.*, 15, 169-177.
- Westerdale, T.H. (2001). Local reinforcement and recombination in classifier systems. *Evol. Comput.*, 9, 259-281.