Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)

A Reinforcement Learning Model for Economic Agent Specialization

Denton Cockburn
University of Windsor
School of Computer Science
Windsor, Ontario, Canada

Ziad Kobti
University of Windsor
School of Computer Science
Windsor, Ontario, Canada

Timothy A. Kohler
Washington State University
Department of Anthropology
Pullman, WA, U.S.A

Abstract

We model a framework whereby agents decide how to allocate their time among available tasks. Agents learn from their previous experiences, adjusting the weights given to each task as a result. These social agents are also influenced by the experiences of others in their social networks, including kin and trading partners. Agents are allowed to trade surplus goods and to request goods that they need using various trading methods. We demonstrate this model using the Agent-Based Model of the Village Ecodynamics Project, which simulates the life of Pueblo farmers of the central Mesa Verde region between A.D. 600 and 1300.

Introduction

Specialization, or division of labour, allows individuals to maximize their productivity by exploiting their skill set and environment (Murciano 1997). It does this by cooperating with other individuals in a community of mutual interest (Spencer, Couzin, and Franks 1998). While there are many definitions of specialization, we use a definition derived from "the production of substantial quantities of goods and services well beyond local or personal need, and whose production is generally organized, standardized and carried out by persons freed in part from subsistence pursuits" (Arnold and Munns 1994). In our definition, specialization is the choice to produce quantities of some goods in excess of a level needed for subsistence, while simultaneously underproducing some goods. When agents specialize, if they do not produce all their subsistence goods, then they must acquire some through trade with other agents (Evans 1978). Specialization can be viewed as a spectrum. Agents can be fully specialized, whereby they perform one task to the exclusion of all others, or they can be partially specialized, whereby they perform all tasks to varying degrees. In our system, our agents are expected to be partially specialized, but it's also possible for some agents to become fully specialized.

Economically, specialization is expected to increase the level of productivity in a market system (Murciano 1997). A productive individual will simultaneously increase the demand for some resources while increasing the supply of others (Young 1928), leading to an increase in wealth (Lavezzi 2003b). Young posits that an increase in specialization leads to more specialization, limited only by the size of the market. Eventually, the market reaches a state of equilibrium (Young 1928). This state is not assumed to happen in dynamic human societies, as the birth and death of individuals keeps the situation in a state of flux. Other changes in agent states also have this effect, such as an agent no longer being able to perform a certain task (Lavezzi 2003a). In this paper we create a mechanism by which human agents can adjust their specialization among given tasks.

The amount of specialization and level of output depends on agent competition, trade networks, and even initial conditions (Lavezzi 2003a). If many agents are already performing the same task and outputting the same resource, then the supply for that resource is likely to surpass the demand, making it being illogical for more agents to supply the same resource. It has been shown in complex systems that the level of specialization in a system is affected by the size of the system (Bonner 2004), which is a property also found in human societies (Bonner 1993). The behaviour of cognitive agents has been modeled using motivation networks (Krink, Mayoh, and Michalewicz 1999). Agents would choose between moving, eating and breeding based on conditions within the environment. Our human agents are not specialized to this degree, and only use specialization to determine what jobs to perform (and how to divide time among those tasks).

Case Study: Village Ecodynamics Project

The Village Ecodynamics Project (VEP) is a multidisciplinary project involving many institutions. The model (Kohler 2000; Kohler et al. 2007) simulates the Pueblo settlement location and size, and farming practices based on archaeological survey and excavation data. The object is to understand the settlement, subsistence, and exchange behavior of the inhabitants and the reasons leading to their eventual disappearance from the region around A.D. 1300 based upon modern archaeological knowledge of the region. The study area is rich in ruins and artifacts of pre-Hispanic Pueblo peoples from AD 600-1300. Kobti et al investigated the role of social interactions in this system, using cultural algorithms and trading networks (Kobti, Reynolds, and Kohler 2004; Reynolds, Kobti, and Kohler 2004; Kobti and Reynolds 2005).

The model creates agents, each representing a household, that live, work, and reproduce based on the data collected in the region, or based on analogues from societies that are similar in scale. Agents are responsible for gathering resources while feeding their families and trading with other agents. Agents can farm maize, hunt for protein (which includes cottontail, jackrabbit, and mule deer), obtain water from rivers, springs, and other water sources, and also gather wood for the purposes of fuel from forests. A more detailed explanation of the model is outside the scope of this work and can be found in (Kohler et al. 2007). The work reported here significantly modifies and expands the simulation as reported in these earlier references.

Approach

Task definition

A concept is an idea or technology and an action is a verb that tells what may be done. CT is the set of all concepts, V is the set of all actions, and T is the set of all tasks. A Task t is defined as a tuple (C_t, A_t), where $C_t \subseteq$CT and $A_t \in$V. Tasks are therefore a combination of prerequisites and actions required to perform each task. It is also possible that Task t $\in$CT. This means the knowledge of a task may serve as a prerequisite for knowledge of another task. Let $K_{Ag} \subseteq$CT such that c $\in K_{Ag}$ implies that c is a concept known to individual Ag. Given a Task t, for all c$\in$CT, c$\in K_{Ag}$ iff Task t is available to agent Ag. This simply states that if an agent meets all the prequisites for performing a task, then they are able to perform this task. We therefore also define the set $T_{Ag} \subseteq$T, as the set of all tasks available to be performed by agent Ag.

Problem Description

Given agent Ag, the set T_{Ag} and a resource R_{Ag}, how does an agent allocate its R_{Ag} among each task t in T_{Ag}?

So: Σx_i= S(R_{Ag}), where i is each task in T_{Ag}, S(R_{Ag}) refers to the amount of the resource R_{Ag} available, and $x_i$refers to a fraction of S(R_{Ag}).

The problem also involves the following conditions:

The problem is continuous over a period of iterations

S(R_{Ag}) changes between iterations

x_i is allowed to change over iterations

Each agent Ag also has a set REQ_{Ag}, such that a resource r $\in$REQ implies that Ag needs some amount of r for subsistence between iterations.

Weight-based model for time allocation among tasks

For each agent Ag, we propose a set EC $\subseteq T_{Ag}$, where $e_i \in$EC implies there is a task i in T_{Ag} and e_i represents the weight of task i.

Task weights in EC are relative, therefore for the given a task i and a resource to be allocated R_{Ag}, the amount of R_{Ag} to be allocated to task i is:

$\frac{e_i}{S(EC)}$ x S(R_{Ag}), where S(EC) is the sum of all elements in EC. We make no assumptions about the initialization of the weights in EC. They can be randomly assigned, or initialized by some other method. A task having a weight of 0 will result in the task being allocated none of R_{Ag}.

Ag must possess some evaluation function P(t) for each task t in EC. P(t) is assumed to be a composite function, assumed to be an economic performance function. P(t) is applied to each task in EC after the performance of that task, therefore representing the result of performing the task. If P(t) > 0, the task is assumed to have had a positive result, in which case e_t is increased by some factor, which is domain dependent. In the case of P(t) < 0, e_t is similarly decreased by some factor. The result of this process is the updating of the weights in EC, which in turn determine how each agent allocates the resource in question.

Our weight adjustment model is a reinforcement learning model, as households learn and adjust based on previous experiences. Note also that agents are not concerned with the results or experiences of their neighbours.

Environment

The VEP environment consists of 4 resources: water, wood, maize and meat; all but wood are needed for survival. Maize is considered the only caloric source, but protein is still required. Households allocate the amount of time to spend on a task based on the collective needs of its members. Each resource is associated with a task that produces that resource. A farmer produces maize, a hunter acquires protein, a woodsman gathers wood, and a water carrier retrieves water. Each task also has constraints and requirements for the performance of that task. Farmers require plots of land to plant their maize. There are a limited number of productive plots on the landscape and each plot varies in productivity, both across the landscape in any year, and through time. Hunting requires the presence of animals within the hunting range (a parameter set in the simulation). Gathering wood requires the presence of trees, and carrying water requires that there are water sources that the agent can travel to. Note that for wood and water, agents are not bound by the distance to these resources. They can travel as far as they need to in order to obtain these resources. All tasks require energy to perform, and thus require the agent to have enough calories to perform the task. The amount of energy required is already a feature of VEP, the explanation of which is outside the scope of this work.

Agents must allocate their family's total calories available for the year among the given tasks. The number of calories available to each household is determined by the number of adults in the household, the number of children in the household, as well as how many hours per day each is willing to work. In our simulation, the number of hours willing to work was set to the same value of 7 hours/day for every family. Agents are able to spend any amount of their calories on any specific task. While not measured, agents also have a secondary goal, the accumulation of resources that increase its economic security for times when they cannot procure additional resources, such as during a famine or drought. To prevent agents from dying before they have time to procure resources, all households are given an initial allocation of two year's supply of maize and meat, as these resources can only be gathered, in the model, in autumn and summer respectively.

It is not feasible to initialize an agent's allocation among tasks randomly, since a low allocation for farming, for example, may result in the agent starving. To address this problem, we have households calculate how much of each resource they need and allocate enough time to meet these needs. The only way for a new agent to be introduced to the system is for a household to survive long enough to produce offspring.

Updated Simulation

In our updated simulation, agents do not allocate their time/resources based on needs (except during initialization). Instead, agents all have a set amount of time each member of the household is willing to work in one day. This is a parameter to the simulation, the value of which we currently have set to 7 hrs/day. This results in a certain amount of calories that a household has available to expend during the year. Households then have the ability to allocate their available calories among the 4 tasks. While agents must sustain needs to survive, their focus is on maximizing their productivity given their abilities. All agents have the same skill level, so ability is delineated by the productivity of an agent at performing a task. Thus an agent having more productive farming plots would get a higher return on the energy expended on those plots, and thus can be claimed to be a "better" farmer than an agent with a less productive plots.

As the weight system cannot be initialized randomly, we have each agent perform their tasks in the first year based solely on the family's needs. We then use the results of this to determine the initialization of our weight system. If an agent spent 25% of their first year farming, then farming will have a 25% weight for that agent during the second year. After this, agents rely on a performance and feedback function to update their weights. Agents may not be able to provide themselves with all the subsistence goods they need, and may thus rely on trading and begging to obtain those resources.

There are many other processes that agents perform that are outside of this work. New births and age-related deaths are examples of this. We also acknowledge that some of the changes we've made to our simulation are not historically accurate. One such example is that agents in our simulation can store infinite amounts of a resource, such as water. At this point in the development of this simulatuion our goal is not to be fully accurate historically, but to create a framework in which domain experts may implement historically accurate constraints.

Agent steps

In this model, the agent procures resources based on the following steps:

1. Perform first year based on family needs
2. Use allocations from previous year to initialize weights
3. Perform tasks and expend energy (eating, performing those tasks)
4. Exchange resources if needed
5. If still alive, update weights
6. If agent location is not sustainable, then move to new location.
7. Goto 3

The logic used by the agent to plan their first year, perform tasks, and change locations are not affected by our changes.

Agent states

Agents have 5 states for each resource as it relates to health and trading. Calculations for each state depend on the size and makeup of each family. The calculations do not include usage of the resources for the purposes of working or performing other tasks. The states are based on how long the agents estimate the amount of the resource they possess will be able to meet their family's needs.

- WEALTHY - More than 2 years supply
- TRADING - 1.5 to 2 years supply
- SATISFIED - 6 months to 1.5 years supply
- CRITICAL - less than 6 months supply (but above 0)
- STARVING - When an agent doesn't have any of the resource needed, and needs to immediately obtain some via trading or begging.

Update function

We use a uniform update function for each task in our weight system. This update function is applied at the end of each year, and determines how the agent will allocate time for the upcoming year. All agents strive to reach/maintain a WEALTHY state.

Given task t that provides resource r and x amount of resource r:

1) If the agent is in a WEALTHY state, then assume y = x - the threshold for WEALTHY. The agent will then reduce the weight of task t proportionally, resulting in the agent producing y less of r than it produced in the previous year. In other words, if the agent has 200 Kg too much maize, then it will reduce the weight it applies to farming so that the agent expects to produce 200 Kg less maize next year. We also define a maximum that we allow the weight of a resource to decline, this prevents any task from reaching a weight of 0.

2) If an agent is not in a WEALTHY state, then the agent will attempt to increase the weight for the task t so that it expects to produce enough additional resources in the following year to get it to a WEALTHY state. For an example, if WEALTHY is defined as 800 Kg of maize, an agent has produced 1000 Kg in the current year, the agent has 200 Kg stored, and the weighting for farming is 0.1, then the agent will increase the weight so that it would expect to produce 600 Kg more maize in the upcoming year. That would result in the agent increasing the weight for farming to 0.1 + 0.1 * (800 - 200) / 1000, or 0.16. It should be noted here that when an agent adjusts the weight for one task, it does not factor in changes that may have happened with its weights for other tasks.

To prevent agents from oscillating between sharp increases and decreases in weights, we limit the percentage by which agents can modify their weights. Weights are then normalized so that the sum of all weights among all tasks is 1.

Exchange

Exchange, though restricted by a trading distance, is necessary when an agent cannot provide for its own needs, for any of the 4 resources modeled.

Barter Exchange We introduce a barter exchange system into the simulation to allow agents to trade one or more resources in exchange for another resource. We use a simplified barter exchange system in which agents trade goods based on a fair valuation system. Prices are therefore not negotiated between agents. To determine values for resources, we use the agent's cost of production. We accept that this does not result in the level of inequality that one would expect in barter system where prices are negotiated. For instance, in a barter system, we'd expect that if an agent has the sole supply of a desired resource, this would inflate the price of that resource much higher than the agent's cost of production. We did not include such a mechanism as it would increase the computational complexity of the system beyond what we were willing to tolerate currently. Moreover, in this world most resources are distributed fairly evenly.

If an agent (rAG) is in a state of CRITICAL or STARVING for a resource, it tries to obtain enough of that resource to get back to a SATISFIED state. First it must identify agents that it can possibly trade with for the resource. It does so by the following process:

1. Ask each agent tAG within trade range if they are willing to trade the needed resource and what they are willing to accept in exchange.
2. Call the set of resources that tAG is willing to accept RWA(tAG)
3. If tAG has enough of the resource being requested by rAG (tAG is in a TRADING or WEALTHY state for that resource)
   (a) If rAG has enough of one of the resources being demanded (in a SATISFIED state or better) by tAG, then add tAG to a list of trade partners, which we can call TList.
4. Sort TList in order of price for the resource being sought.

After finding out which agents within its trade range are potential trade partners, rAG must then ask these agents to trade in exchange for what it can offer them. That process is as follows:

1. For each agent tAG in TList
   (a) Calculate how much of the required resource tAG is willing to sell. tAG is willing to sell any amount as long as it would not fall below the TRADING state.
   (b) Filter RWA(tAG), removing resources where rAG is not above the CRITICAL threshold for that resource. The resulting set can be called TRADE_SET.
   (c) Calculate how much of the required resource tAG is willing to offer (so that it doesn't fall below TRADING), we can call this set OFFER
   (d) Limit OFFER to the amount desired by rAG
   (e) Calculate an amount for each resource in TRADE_SET that is equivalent in value to OFFER. rAG is not allowed to fall below SATISFIED for any of these resources.
   (f) If we can find a combination of such resources, then trade that combination of resources with tAG in exchange for the required resource.
   (g) If we cannot find such a combination, then calculate the maximum total value of resources that we are willing to trade with tAG.
      i. Calculate the amount of the required resource that tAG is willing to give for that value.
      ii. Trade the selected amount of resources in exchange for the equivalent amount of the required resource that tAG is willing to give.
   (h) If rAG is now in a SATISFIED state for the resource, then stop, otherwise move to the next agent tAG in TList

As stated above, the value of a resource is determined by the cost to the agent to acquire that resource. So if it costs an agent 1000 calories to acquire 10 Kg of protein, then the value of that protein is 100 calories/Kb. Agents do not question the value of resources as determined by other agents. Note also that agents are able to sort through those providing resources. This means that agents know who in their neighbourhood can provide the resource at the cheapest prices. This factor results in the requesting agent having an advantage in trade relationships, as it can sort selling agents by price, but selling agents will accept the cost to rAG to produce the goods being given in exchange.

Generalized Reciprocal Exchange An agent household will know the households of its parents (the male and female's parents), as well as those of its siblings. This leads to the introduction of the generalized reciprocal network (GRN) (Kobti, Reynolds, and Kohler 2004; Reynolds, Kobti, and Kohler 2004; Kobti and Reynolds 2005). This network operates over the kinship network between households. In GRN, agents are able to make requests for resources from these other closely related households. This provides a social safety net. Agents are not expected to repay resources that they obtain in the GRN. In addition to responding to requests, agents in a WEALTHY state will donate some of their resources to a household to which they are linked in the GRN. All trading and donation in GRN are limited to a certain geographical distance, which is a parameter set in the simulation. Kin will not put themselves below the SATISFIED state to help, as this may put their own household at risk.

Balanced Reciprocal Exchange The balanced reciprocal network (BRN) (Kobti and Reynolds 2005) is a reputation-based borrowing/loaning network. Agents are willing to loan resources to non-kin neighbours within its trading range. Agents are able to improve their reputation by loaning resources. If a neighbour loans an agent a resource, his reputation with that agent goes up. This means that later if this neighbour is in need of another resource that the agent is able to provide, they will more likely do so. Resource transaction in the BRN is like-for-like. This means that if an agent is loaned some maize, they are expected to repay in maize. They cannot repay any equivalent debt in a different resource such as protein. A neighbour agent has to be in a TRADING or WEALTHY state before they are willing to consider lending through the BRN. After this they are then willing to consider the reputation of the asking agent. The asking agent needs a positive or neutral reputation before the neighbour will proceed. After these two requirements, the neighbour will consider whether they are in a loaning mood, based on a probability formula within the simulation. Note also that a neighbour will not allow itself to fall below a TRADING state in loaning a resource to an agent. These loaned resources do not accrue interest.

Trading process Agents first seek to obtain the needed amount of a resource via the barter network. If the agent still has not obtained enough of the resource it needs from its trading partners and it's in a STARVING state, it attempts to use one of the other trading networks. First the agent uses the GRN trading network to ask up to 4 kin (this is another parameter set in the simulation) to give it the amount it's short. If an agent is still unable to meet its resource requirements, it then proceeds to try to borrow the resource on the BRN. There are no consequences if agents are not able to obtain all the resources they need, unless they starve as a result. The balanced and reciprocal networks are limited to protein and maize exchanges only. This means that an agent can only obtain water and wood via barter or by procuring it on their own.

Experiments

We present our simulation using standard parameter settings. An explanation of all the paramters used in the VEP simulation is outside the scope of this work. All agents have resource needs that are dependent upon the size of their household. Our primary expectation is that agents will adapt their behaviour to changing resource availability. Agents will still die as some resources may not be available to them. Agents that are unable to adjust sufficiently to their environment will die.

As viewed in Figure1, agents' allocation of their time changes considerably during the initial years. This occurs as agents adapt to the environment they are in. We note that agents increase the amount of time they spend hunting significantly during 3 periods in the simulation. This is due to the change in the deer population 2. Deer serves as the most efficient form of protein (it has the highest protein return to effort). When the deer population decreases, agents need to spend more time searching for deer, or get smaller protein returns from hunting rabbits and hares. Another result of the shortage of deer is that the agent population decreases as agents died from starvation and malnutrition 3. The decline in the agent population in turn led to an increase in the deer population, which also allowed the agent population to eventually recover. This result was repeated several times over the run of the simulation, as can be seen in Figures 1 and 2.

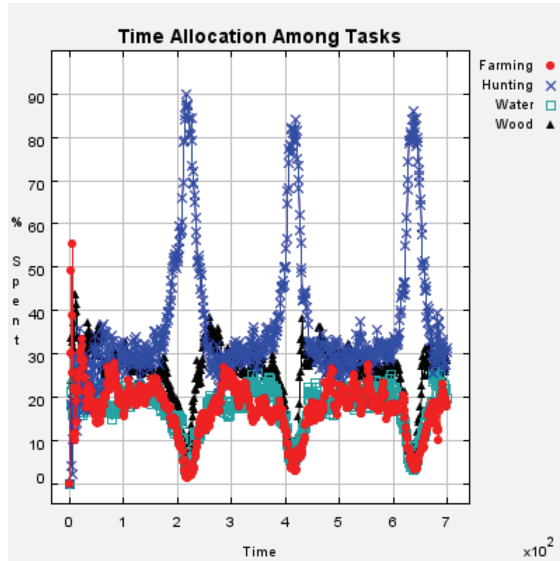

Figure 1: Average agent time allocation among the 4 tasks

In Figure 2 we can see that the decline in the deer population bottoms out at around 1000 deer. This is a failsafe in the VEP simulation that prevents all the deer from being hunted.

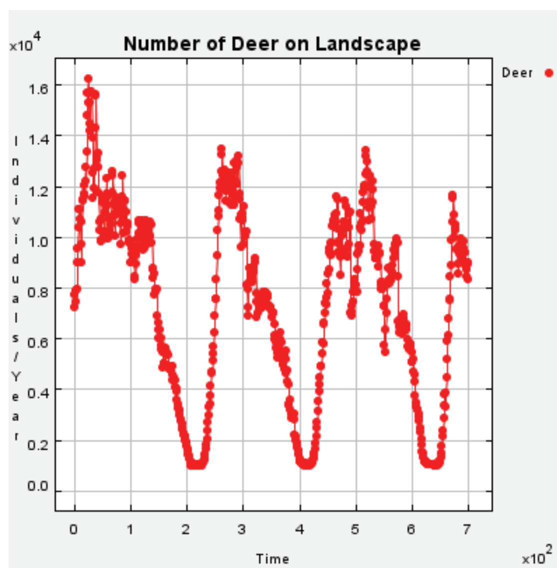

Figure 2: Average annual number of deers over time

We can see in Figure 3 that in the early years of the simulation (first 3 years), there is a sharp decrease in the population. This is not a result of the changes to the simulation explained in this paper. When the simulation is initialized, agents are placed randomly across the map. A lot of these locations do not have enough resources for an agent to survive. The initial drop in population reflects this.

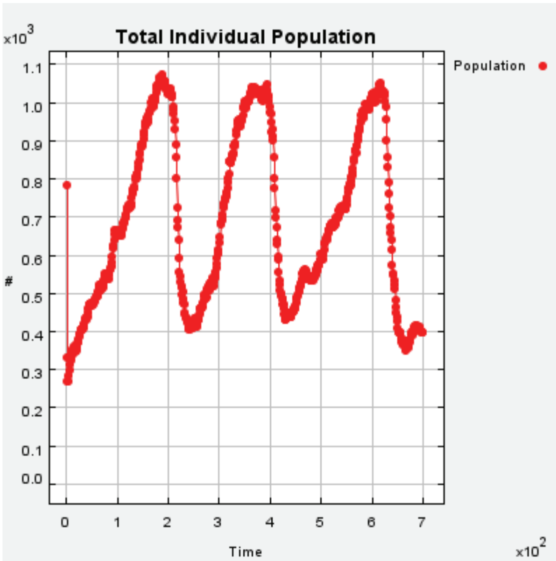

Figure 3: Population counts over time

Conclusion and future work

In this paper we implemented a weight-based system for agent time allocation in the Village Ecodynamics Project simulation. We demonstrated that agents are able to survive in a system where they attempt to maximize their own productivity and rely on trade relationships to provide subsistence goods they do not produce on their own. In addition, we saw that agents are able to survive the drought and famine periods that we know to have existed in the region based on the archaeological record. In future work we intend to explore biological and social influence factors on agent specialization. Biological preferences should allow agents to prioritize resource procurement based on resource importance and decay rate.

Two significant long-term trends in human social evolution are for human groups to become larger in population and more internally specialized. These additions to the VEP simulation are an important step towards understanding how one aspect of these changes may be modeled. Comparison of the model and the empirical record helps archaeologists to understand the processes that may be responsible for the patterns they see, and encourages the modelers to be realistic in their representation of the target system.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant DEB-0816400. Additional funding was provided by a Discovery grant from the Natural Sciences and Engineering Research Council of Canada.

References

Arnold, J., and Munns, A. 1994. Independent or attached specialization: The organization of shell bead production in california. *Journal of Field Archaeology* 21(4):473–489.

Bonner, J. 1993. Dividing the labor in cells and societies. *Curr Sci* 64:459–466.

Bonner, J. 2004. Perspective: The size-complexity rule. *Evolution* 58:1883–1890.

Evans, R. 1978. *Early Craft Specialization: An Example for the Balkan Chalcolithic. In Social Archaeology: Beyond Subsistence and Dating.* New York: Academic Press.

Kobti, Z., and Reynolds, R. 2005. Modeling protein exchange across the social network in the village multi-agent simulation. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, 3197–3203 Vol. 4.

Kobti, Z.; Reynolds, R.; and Kohler, T. 2004. The effect of kinship cooperation learning strategy and culture on the resilience of social systems in the village multi-agent simulation. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, 1743–1750 Vol.2.

Kohler, T.; Johnson, C. D.; Varien, M. D.; Ortman, S.; Reynolds, R.; Kobti, Z.; Cowan, J.; Kolm, K.; Smith, S.; and Yap, L. 2007. *The Model-Based Archaeology of Socionatural Systems*. Santa Fe, New Mexico: SAR Press. chapter Settlement Ecodynamics in the Prehispanic Central Mesa Verde Region., 61–104.

Kohler, T. 2000. The final 400 years of pre-hispanic agricultural society in the mesa verde region. *Kiva* 66:191–264.

Krink, T.; Mayoh, B.; and Michalewicz, Z. 1999. A patchwork model for evolutionary algorithms with structured and variable size populations. In *Genetic and Evolutionary Computation Conference*.

Lavezzi, A. M. 2003a. Complex dynamics in a simple model of economic specialization. Technical report, University of Pisa.

Lavezzi, A. M. 2003b. Smith, marshall and young on division of labour and economic growth. *European Journal of the History of Economic Thought* 10:81–108.

Murciano, A. Millán, J. Z. J. 1997. Specialization in multi-agent systems through learning. *Biological Cybernetics* 76(5):375–82.

Reynolds, R.; Kobti, Z.; and Kohler, T. 2004. The effects of generalized reciprocal exchange on the resilience of social networks. *Computational and Mathematical Organizational Theory* 9:227–254.

Spencer, A. J.; Couzin, I. D.; and Franks, N. 1998. The dynamics of specialization and generalization within biological populations. *Journal of Complex Systems* 1(1):115–127.

Young, A. A. 1928. Increasing returns and economic progress. *The Economic Journal* 38:527–542.