Agent-Based Modeling of Ambidextrous Organizations:

Virtualizing Competitive Strategy

Nicholas S.P. Tay, *University of San Francisco*

Robert F. Lusch, *University of Arizona*

Agent-based modeling, a rich platform for studying complex evolving systems, is used to model a market where ambidextrous and nonambidextrous organizations compete for buyers.

Turbulence, uncertainty, dynamic processes, and networks increasingly characterize competitive markets and business strategies. Consequently, there's a need to model such markets and strategies as dynamic, evolutionary processes—that is, as *complex adaptive systems*.

There are three common approaches to investigating competitive markets and strategies: cross-sectional or longitudinal empirical studies with relatively short-term data; analytical modeling based on neoclassical economic foundations and profit maximization or game theoretic paradigms; and economic experiments with real subjects. None of these methods lets us adequately model and investigate how things unfold over long periods of time. Competitive strategy, like all evolutionary processes, occurs relatively slowly over long periods of time. So, even a time-series analysis of 20 or 30 years might not reveal the true dynamics underlying a competitive marketplace, especially if the data is coarse grained and includes only ending quarters or years.

However, with *agent-based modeling* (ABM), you can construct a virtual competitive market that gives business strategists a Petri dish for investigating competitive strategy. To accomplish this task, we built a basic virtual marketplace that others could refine to introduce more realism. Our purposes were twofold:

- to illustrate ABM's value for studying competitive-market processes and strategies and how the complexities evolve over time and space, and
- to use ABM to assess the extent to which an organization should develop *ambidexterity*.[1]

Ambidextrous organizations have a dual focus:

- *exploitation*, or making small, incremental changes to what they're good at and what they know, and
- *exploration*, or making revolutionary changes that characterize innovation.[1,2]

Balancing exploitation and exploration is important to the study of product innovation. This is an issue of national concern as more countries try to become globally competitive via innovation.

Strategy modeling: ABM vs. traditional methods

One characteristic distinguishes ABM from virtually all other current methods for modeling competitive strategies. Most traditional research and statistical methods focus on the relationship between constructs or variables.[3] Perusing contemporary research in economics, management, and marketing science reveals that scientists are trying to model how construct A influences, causes, or relates to construct B. On the other hand, ABM models the microscopic actions and interactions of agents or actors (in our case, sellers and buyers).

Viewing competitive market and business processes not as the interaction of variables but as interactions among agents who mutually influence each other reduces economics to its most microscopic level. In doing so, it tries to get below the surface of

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

what's often observed—the macroscopic outcome of microscopic actions.

For instance, in economics, an analyst could estimate demand models as the responsiveness of sales to price, advertising, or product quality. The analyst would fit the organization or industry data to an equation (linear or nonlinear) described by a set of parameters that relate independent variables to dependent variables.

On the other hand, with ABM, we would model how organizations set their decision variables in relation to competing sellers' actions and how customers react to offers from the organization and its competitors. From these actions and interactions among the organization, competitors, and customers, a relationship or pattern would emerge between the organization's decision variables and sales. However, because the agents' interactions would comprise continuing processes, the relationships or patterns between the variables would regularly change as the agents themselves changed. This is the evolutionary, complex adaptive-systems feature of many agent-based models.

The ambidextrous organization

Researchers have written about ambidextrous organizations. James March suggested that exploitation focuses on efficiency, production, refinement, implementation, and execution, while exploration focuses on discovery, innovation, play, experimentation, risk taking, search, and flexibility.[2]

Michael Tushman and Charles O'Reilly compared corporate survival to evolutionary biology, where adaptation to change occurs slowly as a result of environmental change.[1] This is consistent with evolutionary biology's view of species change—that species gradually adapt over time via variation, selection, and retention. However, Tushman and O'Reilly noted that Darwinian theory was altered to reflect the occurrence of *punctuated equilibria*, which is when long periods of gradual change are interrupted by dramatic environmental shifts. When this occurs, and the species continues to change gradually, the species is often subject to extinction. Consequently, when organizations meet periods of rapid environmental change, they must do more than exploit what they know or do best. They must innovate and explore new ways of coping and surviving. You never know when a major environmental discontinuity will occur, so organizations should become ambidextrous.[1]

Because evolutionary biology has been a theoretical framework for speculating about ambidextrous organizations and how they use resources to enhance their survival, we have an excellent setting for demonstrating ABM. Exploitation and exploration strategies are both learning strategies. Using a biological analogy, learning via exploitation is like learning via crossover, where the entity combines its knowledge in different ways to improve what it's doing (that is, it recombines existing genes in novel ways). It applies selection and reproduction to imitate or reproduce what works and then discards what doesn't work. On the other hand, learning via exploration can be accomplished with mutation, where the entity does something different (for example, obtains new genetic material by adopting a new technology), then uses selection and reproduction to keep and replicate what works and discards what doesn't work. So, both exploitation and exploration use selection and reproduction, but they differ in how they obtain variation.

> Viewing competitive market and business processes as interactions among agents who mutually influence each other reduces economics to its most microscopic level.

Testing competitive strategies in a virtual market

We created a virtual world for testing various competitive strategies. Our theory of ambidextrous organizations would suggest adopting a different competitive strategy in a stable environment than in an unstable environment. We tested these hypotheses:

- In a stable environment, organizations with high exploitation and low exploration will perform best over time.
- In a turbulent environment, organizations with high exploitation and moderate or high exploration will perform the best over time.

Essentially, these hypotheses suggest that the environment's turbulence moderates the effect between learning strategies and the organization's performance.

Basic model structure

To make the agents in our model as lifelike as possible, we made them autonomous and interdependent. That is, each agent can make his or her own decisions, and there's no central authority or institution to coordinate the system. However, the agents are interdependent. Each agent influences and is influenced by other agents. Together, the agents help create a dynamic environment that they must adapt to. The agents' decision making is driven by a set of "if A, then B" rules. These are not static rules; these rules can evolve on the basis of how well they have helped the agent performed in the past. The agents learn by studying what worked in the past and constantly update their rules according to their experience. But while they rely on past experience for guidance, there's a forward-looking element in how they adapt. They modify old rules to create new ones to test in the marketplace. Therefore, they have the ability not only to learn but also to create new knowledge based on their past experience and information. They move into the future by induction, not deduction.

In addition, we made our agents even more realistic by making them heterogeneous, with memory and bounded rationality but lacking complete information about the issues they confront. The agents don't have the superhuman abilities common to such analytical models as those from the neoclassical microeconomic paradigm.

In developing the virtual competitive world, the key was not to model all the details. Instead, we explored the simplest set of assumptions to allow the virtual world to generate the pattern of explanatory interest (that is, the appropriate strategy to adopt in a stable versus turbulent environment).[3,4]

Our competitive market consists of N_b (80) buyers and N_s (4) sellers. The sellers compete for customers. Each seller adjusts price (P), product attribute (PA), and production level (L) to improve profits. Each buyer buys one unit per period, and all buyers have the same preferred product attribute. However, sellers don't know the buyer's preference or that the buyers all have the same preference. Each seller can set the price from \$0 to \$10 [$P = (0, 10)$], the product attribute from 0 to 10 [$PA = (0, 10)$], and the production level from 0 to 80 [$L = (0, 80)$]. Furthermore, each seller has fixed costs (Fc) of \$10 and variable costs (Vc) of \$0.50. The seller can't hold the unsold units in each period for future sale. Each buyer will purchase from the seller that offers the lowest score value (S), defined as $S = P + \alpha D$, where D is the absolute distance between a buyer's preferred attribute and the actual attribute a seller offers. D measures the extent of disagreement between the product attribute the seller offers and the attribute the buyer wants. α controls D's importance relative to P in determining S. In the simulation, we set α as 2. When more than one seller offers the same S, they split the units sold evenly.

We also needed to model the environment's turbulence as a market characteristic. In a stable environment, the 80 buyers have the same preferred product attribute throughout the simulation. In a turbulent environment, the buyers' preferred product attribute varies randomly every 1,500 periods.

Modeling decision making and learning

In determining P, PA, and L, each seller must forecast what the other sellers will offer and how the buyers will respond. To do well, the sellers must learn to outguess each other.

So, how do sellers make decisions in such an environment? Psychologists observe that when people must make decisions and learn in complex, ill-defined environments, they rely on their innate ability to reason inductively and analyze in fuzzy terms. Fuzzy logic lets us efficiently process an immense amount of complex information. We gain efficiency from compressing the information into a few fuzzy notions that we can handle more economically using fuzzy logic. By being less precise in our description, we can be more relevant in our decision making. Doing so improves our likelihood of survival in a complex, ever-changing environment.

Furthermore, we must rely on inductive reasoning because the lack of information causes gaps in our reasoning and makes our logical deductions fail.[5–8] So, we sensibly fill those gaps with tentative hypotheses. Specifically, induction comprises two steps:

- *Possibility elaboration* uses our experience and available information to create a spectrum of plausible alternatives or rules of thumb.
- *Possibility reduction* tests those alternatives to see how well they connect our incomplete premises to explain the data observed. We then accept the best fit connection as a viable explanation for the data observed. Subsequently, when new information becomes available or when the underlying premises change, the current connection might not be a good fit anymore. When this happens, a new alternative will take over.

To model inductive reasoning, we let each seller continually create multiple "market hypotheses" or rule bases (possibility elaboration). Each hypothesis is a collection of fuzzy if-then rules that determines the action to take under all contingencies. These hypotheses represent the seller's subjective expectation model of why customers purchase from a particular seller. The seller uses a fuzzy inference system to model how these hypotheses interact with the data observed in each period to determine the actions to take. The seller continually tests these hypotheses in the market and tracks their effectiveness using the *fitness function*, a mathematical function of profits. In the end, sellers will retain and act on hypotheses that can better predict market behavior, and they'll drop poorly performing hypotheses (possibility reduction). They'll then replace the dropped hypotheses with new ones.

When people must make decisions and learn in complex, ill-defined environments, they rely on their innate ability to reason inductively and analyze in fuzzy terms.

To make our agents more realistic and human, we employed a fuzzy logic inference system that can evolve and learn using a genetic algorithm. The GA generates the new hypotheses, tests the existing hypotheses in the marketplace, and weeds out the bad hypotheses.

Additionally, the system's design lets us capture some aspects of the associative-reasoning system identified by psychologists. The associative system can help us understand new information by associating it with the knowledge already in our minds. The associative system processes information in the same way that police officers compose a sketch of a suspect by combining different facial features from eyewitness accounts. This process lets police officers create a picture that closely resembles a suspect even though the officers haven't seen that suspect. Likewise, a similar process helps us interpret new information that we've never encountered before.

Our genetic fuzzy system uses decision rules that are coded as strings of numbers. In our system, there are two levels of association at play. At a macro level, agents in the model attempt to associate the rules to the market states they've observed by matching the conditions in the fuzzy rules with the current market state. However, more important is the next association level, which occurs in the background (agents don't even notice it). Here, the GA performs schema-level association.

A schema is a similarity template that we can generalize from a collection of strings representing the rules. For instance, a possible schema for the strings {00110, 00111, 01110} is (0*11*), where * represents a wild card. Schemas are useful because they let a GA quickly find general patterns that lead to better decision making. Furthermore, in a GA, the searches occur in parallel among all existing schemas. This phenomenon is known as *implicit parallelism*. (Holland's Schema theorem shows that a GA's implicit parallelism lets it process on the order of N^3 schemata per generation, where N is the number of strings in the population. So, if we have 100 strings, a GA will process on the order of one million schemata.)

Our mind also appears to order and reorder concepts in a successively more abstract form, like a GA's schema representations. Such subconceptual representations have two advantages. Like a GA's schemata, these subconceptual representations in our mind allow more efficient storage and processing of new information. Instead of having to store every piece of new information as is, our mind can instead represent the new (perhaps complex) information as subconceptual models. These subconceptual models serve as building blocks for bigger, more complex notions. In addition, such subconceptual models not only symbolize a concept but also represent some of its internal structure. They constitute an analysis of a concept. The advantage of including such analyses in a representation is to permit simpler, faster processing of reasoning.

Making fuzzy inferences

Our model's fuzzy rule looks like

If $(P_i / \overline{P})$ is A1, $(Q_i/\overline{Q})$ is A2,
and $(1 - Q_i/L_i)$ is A3,
then P_i' is C1, PA_i' is C2,
and ϕ_i' is C3

where $\overline{P}$ is the maximum price offered in the last period and $\overline{Q}$ is the maximum units demanded from a seller in the last period. The input variables $[P_i, Q_i, L_i]$ refer to the price seller i offered, the units buyers demanded from seller i, and the production level seller i adopted in the last period, respectively. If $Q_i > L_i$, the seller won't have enough units to meet the buyer's demand; consequently, the most units the seller can offer is L_i. Conversely, if $Q_i < L_i$, the seller will have excess inventory after meeting the buyer's demand. Because our model assumes that the product is perishable and can't be carried over to the next period, it's costly for the seller to carry excess inventory. However, it's also costly in terms of lost sales for the seller to not have sufficient inventory to meet the buyer's demand. The output variables $[P_i', PA_i', \phi_i']$ refer to the new price, new product attribute, and new production parameter value for the seller i to set for the next period. The actual production level is determined as a product of this parameter value and the number of buyers in the market, which we've set to 80. For example, if the new production parameter's value is 0.5, the actual production level will be 0.5×80, or 40 units.

The antecedents A1, A2, and A3 and the consequences C1, C2, and C3 are fuzzy sets[9,10] that depict the states of the input and output variables. We characterize the possible states that each antecedent can assume as "low," "moderately low," "moderately high," and "high," and we code them as "1," "2," "3," and "4." As long as the prevailing state matches these conditions, the rule's conditional part will be fulfilled, and the rule will be activated. However, because these variables are intrinsically fuzzy, the conditions they describe will likely match many market states. So, what really matters is the degree to which each condition is fulfilled. We set the *universe of discourse* (that is, the range) for the input variables $(P_i/\overline{P})$ and $(Q_i/\overline{Q})$ to [0, 1]. For the third input variable $(1 - Q_i/L_i)$, we set the universe of discourse to [–1, 1].

On the other hand, the possible states for C1, C2, and C3 are represented by seven fuzzy sets, which are coded as "1," "2," "3," "4," "5," "6," and "7." We set the universe of discourse for the new price and product attribute to [0, 10] and the universe of discourse for the new production parameter to [0, 1]. In our model, the fuzzy inference system needs 64 fuzzy rules to guide decision making under all contingencies. This is because there are three antecedents that can assume four possible fuzzy states (low, moderately low, moderately high, and high), giving rise to 4^3 or 64 possible states. Collectively, these 64 rules form a rule base, which we consider to be a seller's hypothesis about how consumers and other sellers will respond to his or her business decisions. Each seller maintains 10 rule bases at any moment. We can economically code each fuzzy rule base as $[A_1, A_2, A_3 \mid C_1, C_2, C_3]$, which is a 64×6 matrix, where $A_1, A_2, A_3 \in \{1, 2, 3, 4\}$ and $C_1, C_2, C_3 \in \{1, 2, 3, 4, 5, 6, 7\}$.

Figure 1 uses a simple example to clarify how the fuzzy inference system evaluates the input variables in the rules and transforms the input values to decisions on a new P, PA, and L. We assume that the fuzzy rule base only has 10 fuzzy rules. Using our coding convention, the fuzzy rule base looks like this:

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 4 & 7 & 5 \\ 1 & 2 & 4 & 1 & 4 & 2 \\ 1 & 3 & 1 & 2 & 3 & 2 \\ 2 & 1 & 2 & 1 & 3 & 5 \\ 2 & 1 & 3 & 3 & 2 & 6 \\ 2 & 1 & 4 & 3 & 3 & 1 \\ 3 & 2 & 1 & 4 & 1 & 3 \\ 3 & 2 & 3 & 7 & 3 & 6 \\ 3 & 2 & 4 & 7 & 3 & 1 \\ 4 & 1 & 2 & 4 & 4 & 1 \end{array}\right)$$

Values assumed in the example

	Price (P)	Demand (Q)	Prod. Level (L)
Seller 1	1.25	10	22.22
Seller 2	1.00	50	50.00
Seller 3	1.10	15	40.00
Seller 4	5.00	0	30.00

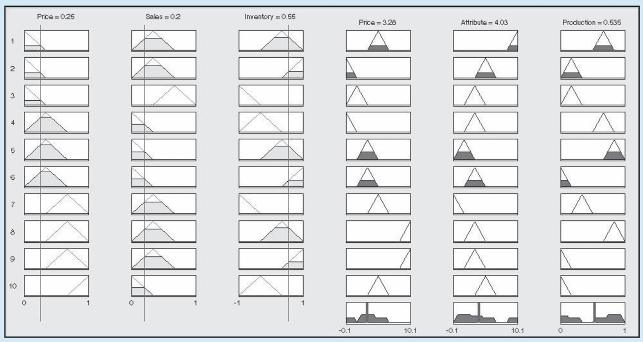
(a)

Seller 1's inputs to the fuzzy rules

$$\left(\frac{P_1}{\overline{P}}\right) = \left(\frac{1.25}{5}\right) = 0.25, \quad \left(\frac{Q_1}{\overline{Q}}\right) = \frac{10}{50} = 0.2, \quad \left(1 - \frac{Q_1}{L_1}\right) = 1 - \frac{10}{22.22} = 0.55t$$

(b)

How the input states are transformed to output decisions



1. *Feeding the inputs to the fuzzy inference system creates several lightly shaded truncated fuzzy sets in the three columns on the left of figure 1C.*
2. *The inputs are connected with the logical connective AND. We model the effects of AND by transferring for each row the minimum height of the truncated fuzzy sets in the previous step to the fuzzy sets for the corresponding row in the three columns on the right. This creates the darker truncated fuzzy sets.*
3. *To obtain a numerical value for each output decision, the darker truncated fuzzy sets in each column are aggregated and the centroid of the aggregated area is computed. This is depicted by the picture at the bottom of each of the three columns on the right. The results are [3.28, 4.03, 0.53] for the price, product attribute, and production decisions, respectively.*

(c)

Figure 1. The fuzzy inference system.

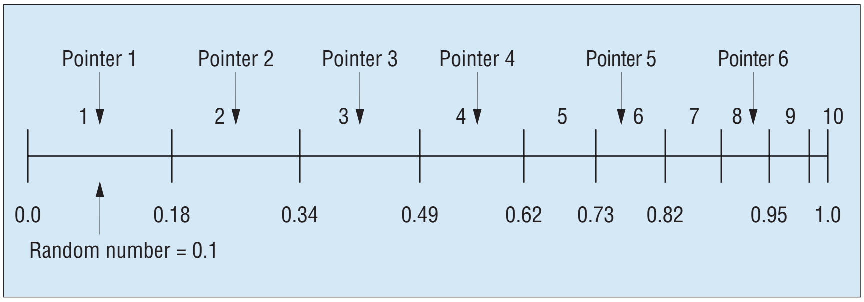


Figure 2. Integers represent objects of interest.

Inductive learning

This section provides the relevant pseudocode to illustrate how we model inductive learning. We constructed our model using Matlab and its companion Fuzzy Logic Toolbox. Although we wrote the code for our model from scratch, it's not necessary for anyone interested in implementing ABM to do so. In recent years, numerous (mostly free) software packages have become available that let people implement ABM without having to code the model from scratch. For a comprehensive list of software tools for programming agent-based models, see www.econ.iastate.edu/tesfatsi/acecode.htm.

Pseudocode. The pseudocode in figure 3 illustrates how we use a GA to model inductive learning and how this fits in the main program. We've omitted details in the program that aren't directly relevant to our task at hand. To denote the program's variables, we use these notations:

- S = score value
- P = product price
- D = absolute distance between the product attribute offered and the desired attribute
- N_s = number of sellers
- N_b = number of buyers.

We alter the probability of crossover and mutation to reflect varying degrees of ambidexterity. The probability of crossover is [0.5, 0.5, 1.0, 1.0] and the probability of mutation is [0.5, 0.25, 0.25, 0.05] for sellers 1 to 4, respectively. Consequently, sellers 1, 2, and 3 are more ambidextrous, and seller 4 is less ambidextrous.

Stochastic universal sampling. Our model uses the SUS method. In our experiments, we desire a random selection of N objects (rule bases) from a population of size T. To implement SUS, we first define a $1 \times T$ vector where each cell holds an identifier for each object, from the most fit object to the least fit object. (In our case, the identifier is the fitness value, which measures how effective the rule base has been in the past. T is 10 because each seller can have 10 rule bases.) We label this vector $\mathbf{T}$. We define the identifier for object i as d_i and define $S_\tau = \sum_{i=1}^{\tau} d_i / \sum_{i=1}^{T} d_i$ so that the vector $\mathbf{S}$ = $[S_1\ S_2\ S_3 \ldots 1]$ contains the accumulations of the normalized fitness values for the T objects (see figure 2). The SUS method selects the desired N objects of interest[11] in one step, using N equally spaced pointers separated by the distance $1/N$. As in figure 2, a random number $\hat{t}$ equal to 0.1 is drawn from $U[0, 1/T]$, followed by the computation of the N equally spaced pointers, which are placed in the vector

$$\underline{\hat{p}} = \begin{matrix} \hat{t} & \hat{t}+\dfrac{1}{N} & \hat{t}+\dfrac{2}{N} \\ \hat{t}+\dfrac{3}{N} & \ldots & \hat{t}+\dfrac{N-1}{N} \end{matrix}$$

The N desired objects are then chosen by whether the pointers in $\underline{\hat{p}}$ lie within the intervals defined by the points in $\mathbf{S}$. In figure 2, the objects [1, 2, 3, 4, 6, 8] are selected.

The virtual world of ambidextrous organizations

We simulated our virtual world over 25,000 periods. You could consider a period to be one day, so 25,000 periods would be approximately 70 years. Because crossover and mutation can occur every 30 periods, this would equate to one month. This could coincide with a monthly meeting where employees and management share ideas about past and future decisions, mimicking crossover. At the same time, the organization could decide to randomly try new approaches, mimicking mutation.

A stable environment

As we predicted, the seller who pursued a strategy of high exploitation and low exploration (seller 4) performed better than the more ambidextrous sellers. In fact, seller 4 was the only highly profitable seller in the stable environment. Because the buyer preferences were stable, sellers had no incentive to explore new product offerings at a moderate or high level. So, this level of exploration was often costly. Figure 4 shows the sellers' average cumulative profits or losses over 20 simulations, with each simulation running for 25,000 periods. Seller 4 had average cumulative profits of $253,491 (see figure 4a). The ambidextrous sellers (sellers 1, 2, and 3) experienced average cumulative losses of $318,995, $224,272 and $74,883, respectively. These sellers used a strategy opposite of what you would expect an informed organization to use. Given the stable environment, they should have heavily exploited or refined the successful product offerings and should have avoided exploration. Instead, they explored at a moderate to high level. Notably, seller 3 lost the least among the ambidextrous sellers. Seller 3 wisely had high exploitation, but its moderate exploration didn't serve it well given the unchanging environment.

A turbulent environment

As we mentioned earlier, we let buyers randomly change their product preferences every 1,500 periods (that is, every 50 months or approximately every four years) to model a turbulent environment. Figure 4b shows the four sellers' average cumulative profits for a turbulent environment. As we hypothesized, the more ambidextrous sellers performed best. However, much as in the stable environment, seller 3 performed best among the ambidextrous organizations. It appears that in an environment characterized by major discontinuities, it's best to have a high level of exploitation and a moderate level of exploration. Seller 3 earned $199,667, while sellers 2 and 4 earned $50,326 and $51,529, respectively. As we expected, seller 1 did poorly because the organization innovated too frequently and failed to adequately exploit its innovations as it adjusted to the major environmental discontinuities. So, it made a meager average cumulative profit of $12,815—less than one-fifteenth of seller 3's earnings.

Proper use of ABM for virtual competitive markets

It's important to use ABM appropriately

MAIN PROGRAM
Initialize the model.
WHILE time < 25,000,
    FOR each of the four sellers,
        Use data from previous period to compute the states of the three input variables to the fuzzy rules.
        On the basis of the fitness values of the rule bases, use the stochastic-universal-sampling (SUS) method to select one rule base out of the 10 rule bases to guide the decision making.
        Feed the states of the input variables to the chosen rule base to compute the decisions.
        On the basis of the seller's decisions, buyers compute their score values ($S = P + 2D$) for this seller and store results in a score matrix ($Ns \times Nb$).
    ENDFOR

    FOR each of the 80 buyers,
        Check the score matrix to determine which seller has the minimum score value and therefore got the sales from this buyer. The seller that got the sales will have a "1" recorded in a sales matrix ($Ns \times Nb$).
    ENDFOR

    FOR each of the four sellers,
        Aggregate the values in the sales matrix to determine the total sales units secured by each seller.
        IF sales units > units produced by the seller, THEN set sales = units produced.
        Compute profits made by each seller and their new wealth ($\text{profit}_i = \text{price}_i \times \text{sales}_i - 0.5 \times \text{production}_i - 10$. The variable cost and fixed cost per unit is 0.5 and 10, respectively. $\text{wealth}_{t,i} = \text{wealth}_{t-i,i} + \text{profit}_i$).
        Update the fitness value ($\text{fit}_{t,i,j} = (1 - 0.005)\text{fit}_{t-1,i,j} + 0.005 \times \text{profit}_i$) of the rule bases that had been used for decision making.
    ENDFOR

    IF 30 periods have transpired since the last crossover and/or mutation, **THEN** execute the crossover and mutation routine.
    ENDIF
    INCREMENT time by 1.
ENDWHILE

CROSSOVER AND MUTATION ROUTINE
  FOR each of the four sellers,
        Save the two rule bases that have the highest fitness values in "bestRB" to avoid being altered.

    CROSSOVER SUBROUTINE
    IF uniform random number <= crossover probability, **THEN**
        Use the SUS method to identify six of the most fit rule bases for crossover. (Rule bases with higher fitness values are more likely to be selected.)
        Extract the fuzzy rules from the selected rule bases and store half in a matrix labeled "Parent1" and half in a matrix labeled "Parent2."
        Create two new matrices by setting Child1 = Parent1 and Child2 = Parent2.
        Create another matrix that has the same size as Parent1 and fill this matrix at random with zeros and ones with 50/50 probability. Call this the "Mask" matrix.
        Locate the coordinates of the ones in the Mask matrix.
        Use these coordinates to extract the values in similar positions from Parent1 and Parent2.
        Insert the values extracted from Parent2 into similar positions in Child1.
        Insert the values extracted from Parent1 into similar positions in Child2.
        Use SUS to identify six of the least fit rule bases for elimination.
        Replace the rules in the rule bases identified for elimination with the new rules stored in Child1 and Child2.
    ENDIF

    MUTATION SUBROUTINE
    IF uniform random number <= mutation probability, **THEN**
        Use the SUS method to identify six of the most fit rule bases for mutation.
        Extract the fuzzy rules from the selected rule bases and subject these rules to mutation by randomly switching with 50 percent probability each fuzzy set in the output variables to any of the seven allowed fuzzy sets ($C1, C2, C3 \in \{1,2,3,4,5,6,7\}$).
        Use SUS to identify six of the least fit rule bases for elimination, and replace the rules in these rule bases with the rules that are created from the mutation process in the previous step.
    ENDIF
    Reinsert the two best rule bases stored in bestRB.
ENDFOR

Figure 3. Pseudocode illustrating how a genetic algorithm can model inductive learning.

to grow virtual competitive worlds. As ABM software becomes more accessible, you'll be tempted to build models and draw conclusions that might be misleading unless you understand ABM's proper use.

Don't use to make precise quantitative predictions

Although agent-based models can provide qualitative insights, you shouldn't use them to produce highly calibrated quantitative predictions. If you try to use ABM to forecast with precision, you'll be disappointed. ABM is a highly simplified model of basic processes that characterize the microactions of agents that result in macrostructures. So, you

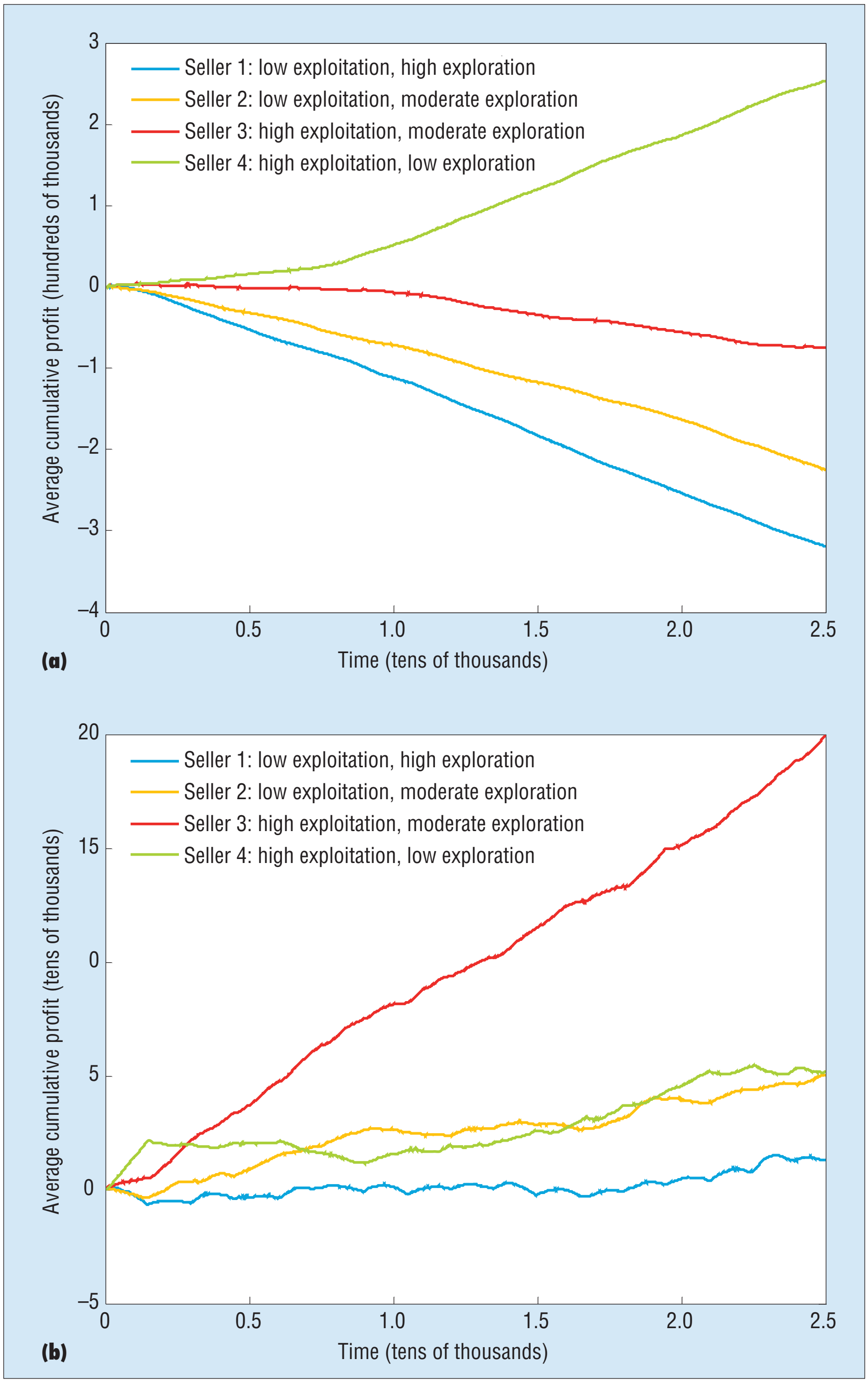


Figure 4. The average cumulative profit payout for four sellers in (a) a stable environment with fixed buyer preferences and (b) a turbulent environment with varying buyer preferences.

should use ABM to show how and why certain qualitative patterns emerge.

Do use to test strategies or theory

You can use ABM in many cases to test strategies or theory. How is it that ABM can't make precise, reliable quantitative predictions but can test theory? Virtually all hypotheses dealing with competitive strategy are qualitative. They generally take the form "Construct or variable A is positively (or negatively) related to construct or variable B." They don't take the form "If A rises by X percent, then B will rise (or decline) by Y percent." Most research in economics, marketing, and management science is basic and not intended to provide precise metrics on response functions or relations between variables or constructs. In this regard, ABM is capable of validating qualitative hypotheses. In fact, it can go further to help identify the microactions of agents that would produce the type of qualitative predictions that most hypotheses make.

Do use for thought experiments

You can use ABM to conduct thought experiments. If you want to understand fundamental processes, you can model those using simple rules and examine the results to increase your understanding. Remember to keep the rules or assumptions simple. If you try to make the models too realistic, you might undermine their usefulness in thought experiments. These models don't necessarily "aim to provide an accurate representation of a particular empirical application. Instead, ABM's goal is to enrich our understanding of fundamental processes that may appear in a variety of applications."[4]

Alter a key contingency or connection and watch what happens. Assume your organization is operating in a turbulent market with four sellers, and each seller has the same level of ambidexterity. You find that being ambidextrous doesn't offer your organization a relative competitive advantage. So, you engage in a thought experiment and reason that if you could learn more quickly than the other organizations, you would perform better. To test this idea, you could use the previous ABM and give all sellers a 100 percent chance of crossover. Then, have three sellers experience crossover every 30 periods and one seller every 24 periods. At the same time, give each seller a 25 percent chance of mutation. Have three sellers experience mutation every 30 periods and one seller every 24 periods. Essentially, one seller can learn, via exploitation and exploration, 20 percent faster than the other sellers (that is, every 24 periods versus every 30 periods).

Do use to bridge micro and macro models

Agent-based models can bridge the gap between macromodels and micromodels. Although agent-based models begin with autonomous agents' actions and interactions, they create macrostructures. So, they can show how a macrosystem emerges and suggest a focus for understanding microlevel mechanisms. To see if you can explain a macrostructure, you might try to grow it in a virtual world. Essentially, you want to uncover what processes produce a result, pattern, or outcome that you observe in a macrostructure.

Organizations can modify our model to reflect their industry, including the number of organizations, level of environmental turbulence, and supply and demand factors. Simulations using the recalibrated model can help organizations decide what effort and resources they should devote to exploitation versus exploration.

As economics, marketing, and management science advances and attempts to better capture competitive markets' processes, complexities, and evolutionary and adaptive nature, we believe ABM will become more popular as a research method. ABM has diverse potential applications. For instance, interactions and actions of buyers and sellers can influence the overall level of price movements, such as rapid cascades of price inflation or deflation. This example also illustrates ABM's value to public policy. For instance, changes in tax policy, patent policy, and advertising regulations could impact how quickly innovation or changes occur in policies on price controls (such as rent control) or taxing of certain goods (such as taxes on owning or renting a house).

A growing area of investigation in economics is networks. Networks inherently involve the interactions of entities, making them ideal for ABM. Social networks such as the Internet have attracted much research attention because of the rise in stock fraud on the Internet. The emergence of Internet technology has made it feasible for a market manipulator to manipulate many investors at lightning speed with negligible costs, while remaining anonymous. The dual advantages of low mass-distribution costs and anonymity has resulted in hundreds of Internet fraud cases. This prompted the US Securities and Exchange Commission to create the Internet Enforcement Office to deal solely with Internet fraud.

You can also use ABM to investigate how a social network's characteristics make stocks more vulnerable to cyber manipulations. There's a growing interest in supply and value networks and the processes that they manage. These can range from traditional procurement and logistical processes to collaborative processes focused on product development or other complex issues.

Finally, the emerging field of service science yields other opportunities for applications, such as in modeling service flows in a hospital or local police and fire protection. One area where competitive strategy is always pervasive is military combat. Here, the agents can be the soldiers and their equipment, and various interaction patterns over space and time can be modeled using ABM. ■

The Authors

Nicholas S.P. Tay is an associate professor of finance in the University of San Francisco's school of Business and Management. His research interests lie in building and applying agent-based models to understand the implications of business decisions in an environment that's constantly evolving due to the perpetually novel and complex interactions among economic agents, institutions, and markets. He received his PhD in finance from the University of Oklahoma. He's a member of the Financial Management Association and the American Finance Association. Contact him at the School of Business and Management, Univ. of San Francisco, San Francisco, CA 94117; tay@usfca.edu.

Robert F. Lusch is the Lisle & Roslyn Payne Professor of Marketing in the University of Arizona's Eller College of Management. His research interests include marketing strategy and service science management and engineering, particularly using agent-based models to understand these areas. He received his PhD in business administration from the University of Wisconsin-Madison. He was the dean of the college of business at the University of Oklahoma and Texas Christian University. He's a member of the American Marketing Association and the Academy of Marketing Science. Contact him at the Eller College of Management, Univ. of Arizona, Tucson, AZ 85745; rlusch@eller.arizona.edu.

References

1. M. Tushman and C.A. O'Reilly, "Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change," *California Management Rev.*, vol. 38, no. 4, 1996, pp. 8–30.
2. J.G. March, "Exploration and Exploitation in Organizational Learning," *Organizational Science*, vol. 2, no. 1, 1991, pp. 71–87.
3. M.W. Macy and R. Willer, "From Factors to Actors: Computational Sociology and Agent-Based Modeling," *Ann. Rev. Sociology*, vol. 28, 2002, pp. 143–166.
4. R. Axelrod, *The Complexity of Cooperation*, Princeton Univ. Press, 1997.
5. B.W. Arthur, "Designing Economic Agents That Act Like Human Agents: A Behavioral Approach to Bounded Rationality," *American Economic Rev.*, vol. 81, no. 2, 1991, pp. 353–359.
6. L.E. Blume and D. Easley, "Evolution of Market Behavior," *J. Economic Theory*, vol. 58, no. 1, 1992, pp. 9–40.
7. N. Rescher, *Induction: An Essay on the Justification of Inductive Reasoning*, Univ. of Pittsburgh Press, 1980.
8. H.A. Simon, "On the Concept of Organizational Goal," *Administrative Science Quarterly*, vol. 9, June 1964, pp. 1–22.
9. G.J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, 1995.
10. B. Kosko, *Fuzzy Thinking: The New Science of Fuzzy Logic*, Hyperion, 1993.
11. H. Pohlheim, "Evolutionary Algorithms: Selection," *GEATbx: Genetic and Evolutionary Algorithm Toolbox for Use with Matlab Documentation*, 2005; www.geatbx.com/docu/algindex-02.html.