"Protests, Elections, and Other Forms of Political Contagion"

Paul E. Johnson
Dept. of Political Science
University of Kansas
Lawrence, Kansas 66045
pauljohn@ukans.edu
http://lark.cc.ukans.edu/~pauljohn

Abstract

This paper explores agent-based models of the formation of individual opinions and collective judgments. Models of information transmission between individuals through political communication are contrasted with models in which individuals adjust their opinions through observations of social aggregates. Several variants of the culture model originally proposed by Robert Axelrod are offered. All models are built with the Swarm Simulation System, an open source toolkit written in Objective-C.

Prepared for delivery at the 1999 annual meeting of the American Political Science Association, Atlanta, Georgia. Special thanks to Rick Riolo of the University of Michigan Center for the Study of Complex Systems.

Statement of the Problem

Large scale social events result from the conglomeration of many micro events. The linkage between macro level stimuli and macro level outputs is uncertain because the macro level inputs are filtered down to the individual level through a complex system in which individual decisions are made.

Theories of individual interaction in the formation of individual judgment, and eventually social judgment, can typically be categorized in two ways. The first holds that individual judgments are affected by their observations of social aggregates. Judgment is not driven by personal conversation, but rather by observation of mass behavior. It seems that descriptions of this sort are especially prevalent in models of mass political protest. Granovetter’s threshold model links individual behavior to observations of aggregate levels of action (1978). Lohmann’s model of revolutionary behavior presents a richer model of the way individuals react to the aggregates they observe, but for many purposes, the essential idea is the same.

A second sort of model about the formation of collective public opinion does not focus on aggregates, but rather details of individual interaction. Theories of this sort are often describes public opinion formation during an election or another kind of political decision. Following some insights of early work by McPhee and Smith (1962) and later explored by Robert Huckfeldt and John Sprague (1995), individuals are seen as parts of loosely knit, flexible networks in which information transmission occurs through political discussions. Individuals adjust their opinions on the basis of the perceived quality of the information from individual discussants and other factors. Adapting Huckfeldt’s analogy, the formation of public opinion is like collecting the conclusions of thousands of individuals serving on different juries (1999).

This paper reports on preliminary effort to develop a perspective from which to design models of these processes and compare them. Along the way, some insights into the modeling process are afforded. The general modeling problems that seem most pressing are the following.

1. How can one differentiate "data" from that which is to be "reproduced?" In other words, how does one know what is to be assumed and what is to be derived? McPhee and Smith proposed to begin their model with known qualities of individuals. However, it may not be safe to assume that the results of empirical research about individuals reflect the actual underlying laws that guide individual action. Instead, the observed patterns reflect the outcome of a sifting-out process that may obscure evidence of some individual traits or create illusory indications of other traits.

2. How can one distinguish specification errors from emergent properties? The term "emergent property" arose from modern complex systems theory and it has captured the imagination of many writers. While the term is elusive and difficult to pin-down, it is meant to convey the sense that a model produces a pattern that is unexpected, but is somehow relevant or important. There is always the danger that these patterns might be artifacts of computer model design, rather than the substance of the problem itself. Hegselmann (1996, p. 222) notes, for example, that it can make a difference if a model is written so that the information state of all agents is temporarily frozen in order for all to adjust their behavior (synchronous updating) or if the effect of each agent’s action is immediately known to each following agent (asynchronous updating). Choices in the structure of updating can produce some surprising patterns, which some might call emergent properties and others will call mistakes (Huberman and Glance, 1993).

Technical Details

The simulations discussed here were done with the Swarm Simulation System (http://www.santafe.edu/projects/swarm). Swarm is a toolkit, meaning it is a programming library that contains a variety of software components that are intended to work together. It is written in Objective-C, an object-oriented computer language. An object is a self-contained entity that is capable of maintaining data (the values of variables) as well as executing methods (sequences of commands). The Swarm toolkit supplies classes that can be used and, in many cases, subclassed to form new variants.

In the software development effort of which this paper is a part, a few important technical hurdles had to be overcome. The most important of these were the development of tools for iterative processing of Swarm programs and development of an enhanced class to manage record-keeping on the positions of the agents.

1. Dealing with batch mode processing.

The Swarm toolkit is designed to allow users to run their programs in either of two modes. The first is the GUI (Graphical User Interface) mode in which the user directly interacts with the program through a point-and-click interface. By the convention, the object that manages the GUI is called the observer swarm. A "snapshot" of the screen from one simulation is shown in Figure 1. The Swarm GUI uses the tcl/tk toolkit (http://www.scriptics.com/software), which is a combined scripting language (tcl) and widget set (tk) that can be used to manage user interaction with a running program. The observer swarm creates a control panel object that has "start" and "stop" buttons that regulate events, and the user can design the code in the observer so that certain user clicks bring up "probes" that display information about the internal states of the agents and instruct them to perform particular tasks. In a GUI simulation, the observer is the top level swarm, which means that it not only manages the GUI, but also it initiates the creation of lower level objects and monitors their progress. The observer swarm can contain code to create various kinds of graph objects. The Swarm library contains methods that can be used to take "snapshots" of particular window displays (called "widgets") at predetermined times and save them in the png format (png is an open source pixmap format similar to the proprietary gif format familiar to internet users).

Swarm also has a batch mode. However, the usage of the batch mode is not so well standardized as the GUI mode. Batch mode processing is required in order to conduct many replications of a single experiment or to conduct a "parameter sweep." The top level object is referred to as the batch swarm and it can create lower level objects and write reports on their activities to files. The Swarm development team has made significant strides in 1999 in the so-called "serialization" of Swarm models, tools that can save summaries of the state of objects and make them available is a standardized format for statistical analysis. (Swarm supports saving of records in the HDF5 format, which is an open-source data storage format supported by a number of advanced packages for graphical visualization of data).

However, within Swarm itself there is no library with which to manage the repetition of simulations. Furthermore, in Swarm’s batch mode, there is no capability of making graphical snapshots to summarize the state of events. In order to circumvent these limitations, two additional steps were taken in the present project. First, a generic simulation management tool called Drone, which was developed at the University of Michigan for the "CAR Group" (Cohen, Axelrod, Riolo), was used (http://aaron.physics.lsa.umich.edu/Software/Drone/). Because Drone was completed before the Swarm project reached maturity, some special effort is required to make the two work together. Second, a user-contributed class called BatchRaster (developed by Nelson Minar and Benedikt Stefansson in 1977: ftp://ftp.santafe.edu/pub/swarm/users-contrib/anarchy/batchraster-movies.tar.gz) was updated to work with Swarm 1.4.1 (ftp://ftp.santafe.edu/pub/swarm/users-contrib/anarchy/BatchRaster-990811.tar.gz). The BatchRaster class can create a portable pixmap record (in ppm format) of a two dimensional space. It records, pixel by pixel, the coloring of a two dimensional grid. Through this approach, pictures can be saved while the program runs in batch mode, without accessing the GUI interface that slows the processing significantly. Incidentally, because of the limitations imposed by Swarm's approach to batch processing, I've explored the development of a new class that sits above the observer swarm and can spawn one GUI simulation after another, recording data and taking pixmap snapshots of any Swarm widget (graph or raster). (An example of such a model can be found at this site: http://lark.cc.ukans.edu/~pauljohn/Swarm/RepeatingHeatbugsParameters-1.4.1-2.tar.gz.)

2. Managing artificial agents in space.

Whether a model is run in the GUI or not, the lower level implementation of the artificial society and the agents within it is the same. By custom, the term "model swarm" an intermediate level object that creates instances of various subclasses of agents, schedules their interactions, and maintains records which can be made available to agents within the simulation or to the record-keeping activities of the observer swarm.

As previously mentioned, the Swarm toolkit contains a number of libraries that help the user to design a simulation. There are libraries to manage collections of objects, to create random number generators and objects capable of generating observations from statistical distributions, to analyze and summarize observations, and also there is a space library which contains a set of classes that can be used to maintain records on the position of agents within two dimensional spaces. One limitation of the space library in the Swarm toolkit is the assumption that only one object can occupy a given position in a space. This is, of course, a standard premise in models based on cellular automata, and the assumption that only one agent can occupy a given location is perhaps logical in a model in which the cells in the grid are very finely grained. In such a model, it might be plausible to argue that it is physically impossible for more than one agent to be in one position at one time. (In fact, in the Heatbugs model, one of the "swarmapps" distributed to illustrate the usage of the Swarm toolkit, the impossibility of positioning multiple agents on a single cell is one of the critical elements of the model.)

If each cell in the grid is thought of as a household or village, the restriction on multiple occupancy becomes untenable. As a result, for the purposes of this project, a new class was developed that allows both multiple occupancy and user interaction with locations through the graphical user interface (GUI). The first steps toward this goal were taken by Sven Thomassen, who released a class he called MoGrid2d (ftp://ftp.santafe.edu/pub/swarm/users-contrib/anarchy/SvenSpaces-970805.tar.gz). MoGrid2d creates a subclass of the Swarm class Grid2d in which there is a Swarm List object at every position. After an instance of the MoGrid2d class is created, user code can then tell the space to add (or remove) objects at a particular position in the space and MoGrid2d handles the dirty work. If an agent tries to move onto a cell that has no current occupants, a new list object is created for that cell, and when the last agent leaves a cell, the list is destroyed.

The MoGrid2d class, however, does not completely solve the problem because it does not allow the user to interact with the entities in the grid, except to add or extract agents. It is not possible, for example, to ask a cell to report the average of agent opinions within the cell, or to describe the conditions of neighboring cells. The ability to dynamically create probes that reveal the internal states of objects is one of the key features of the Swarm library. MoGrid2d does not exploit that ability because it does not make it possible for users to interact with agents within these cells. Since the Swarm List class does not allow subclassing by users, it was necessary to redesign the multiple occupancy grid.

The solution used in this paper is a new class called WrappedListGrid2d. Like MoGrid2d, this class is based on the Swarm Grid2d class. The objects placed on the grid have more functionality than the Swarm List class. These objects are designed to inherit certain generic features, such as the ability to add and remove agents from lists that they manage, but in addition they can respond to application-specific commands, which, for example, can ask the cell to gather information about the current members of the list or draw on graphs to represent a summary of the events within the cell. The WrappedListGrid2d class also contains some code designed to improve speed and efficiency. When a position in the grid is left empty by the movement of agents, the object in that cell is not destroyed, but rather it is removed and placed into a queue for recycling when agents move onto previously unoccupied cells.

Applications

This paper presents two applications of the multiple occupancy class. The first describes the formation of public opinion through information transmission in individual interactions. This example explores variants of a model originally proposed by Robert Axelrod. The second example studies public opinion formation in a setting where the individual actors do not base their judgments on opinions they exchange directly, but rather by observing social aggregates. This model describes the updating of beliefs about the nature of a political regime and the decision to protest.

1. Modeling Interaction and Public Opinion

Huckfeldt and Sprague describe political interaction in terms that go something like this. People move around within their spheres of activity and, depending on where they are and what they are doing, they may meet people who may be inclined to discuss politics. The people are able to sort through their opportunities for political interaction. And, possibly on the basis of that interaction, people may alter their opinions.

I have taken steps to develop a model in which individual agents can formulate their opinions on the basis of their experiences. The agents are capable of selecting discussion partners and moving about in space. The simulations discussed here take Axelrod's culture model (Axelrod, 1997, Chapter7) as the baseline and then consider variations. There are a number of ways to implement the culture simulation in Swarm (two examples that I found most useful were by Rick Riolo, (ftp://www.pscs.umich.edu/pub/software/Swarm/axelcult-990712a.tar.gz) and Lars-Erik Cederman (http://www.sscnet.ucla.edu/polisci/faculty/cederman/S99/p209-1/code/culture.tar.gz)). The code that created the simulations described in this section is available (http://lark.cc.ukans.edu/~pauljohn/Swarm/CultureWrappedGridBatchRaster.tar.gz).

1. Axelrod's culture model (ACM). Axelrod's model conceptualizes interaction among agents in the following way. A square grid of agents, described as villages, is created. Each village has a culture that is assigned randomly at the outset. The number of cultural issue dimensions or topics, called "features" in the culture model, and the number of positions on each topic, called "traits" in his model, can be varied. The agents are like villages in the sense that they are not allowed to move about on the grid, but and each village is conceived of as a unitary actor. An agent (village) is randomly selected and given the opportunity to interact with a randomly chosen neighbor. The set of neighbors is the so-called von Neumann neighborhood, which

consists of the four agents on the east, west, north, and south borders. Cells that lie on the outside boundaries of the grid have fewer neighbors with which to interact. Unlike some other simulation models that use a toroidal representation to make the grid wrap around and connect the opposite edges (Epstein and Axtell, 1996), Axelrod simply reduces the interaction opportunities for actors on the edge. In Axelrod's model, one of the neighbors is selected at random, and then an interaction occurs with probability equal to the similarity of the opinions of the two agents. If the interaction occurs, then an issue on which the two agents disagree is selected at random and the agent's opinion on the issue is changed to match the other.

Axelrod made a number of observations on the basis of his model, the most striking being that, over the long run, there is not likely to be very much cultural diversity. While the tendency toward homogeneity is greater for some parameter settings than others, it is powerful in all cases. This conclusion poses a challenge at the outset for the current modeling exercise. If we are to formulate a useful model of political interaction and formation of small networks, and we want that model to be faintly realistic, we do not want the major implication of the model to be that diversity is unlikely to exist. One solution is to write agents who are individually resistant to environmental influences, but that is not the route explored here. Rather the emphasis is on developing a more intricate understanding of the formation of networks on formulation of public opinion.

2. Generalization of the culture model. The Swarm model described here embeds the Axelrod culture model (ACM) in a more general simulation model that allows multiple occupancy, varying definitions of the neighborhood of each agent, as well as movement among agents within a given grid or movement to another grid altogether. This report provides indications of the effect of changes in the introduction of multiple occupancy as the ability of agents to form "networks" by opening discussions with neighbors in a selective process.

Axelrod used the principles of "zone" and "region" to summarize the development of patterns across the grid. A zone is a continguous region of cells that have at least one feature in common, while a region is a grouping of cells that are completely alike. In the model with multiple occupancy and mobility of agents, these summary indicators are not useful. Instead, I propose to use the summary measures of the perceptions of the individual agents as well as global indicators of diversity.

To summarize the perceptions of the agents, three measures were collected. Each agent keeps a "running tally" of its experiences. For each prospective discussant, the agent records whether there was an interaction. The proportion of possible interactions that actually materialize is referred to as the perceived level of acquaintance. This is recorded as a 20 period moving average. The second and third measures summarize the similarity of an agent and the others with which it has had an interaction. The "harmony" moving average indicates the proportion of opinions that the two agents share, while the "identical" indicator is the proportion of people that the agent finds are identical to itself.

In addition to the subjective experiences of the agents, an objective measure of diversity is also calculated after each time the list of citizens is processed. This measure, known as entropy, or Shannon's information index, is a normed measure that is equal to 0 if all objects in a set are identical and 1 if there is an equal number of every kind (Shannon, 1949; Baltch, 1998). If there are F different features (issue dimensions) and there are T different traits (positions) for each feature, then the number of possible issue stances is T^F. If the proportion agents holding a given set of positions is p_j, then the normed entropy is given by:

$$NormedTotalEntropy = \frac{\sum_{j=1}^{T^F} p_j * \log_2(p_j)}{\log_2(1/T^F)}$$

The normed entropy measure depends on both the number of traits and the number of features. As a result, it can be tricky to compare across models where those variables differ. Furthermore, although the entropy depends only on the proportions of agents holding particular opinions, those proportions can be affected by the number of agents in the simulation. For example, if a distribution has 10 features and 10 types, then the number of possible observed types is a huge number, 10^{10}. If there are 100 agents, there are at most 100 different observed preferences, and so the calculated value of normed total entropy will be lower than a model that has 10000 agents.

To allow comparison across models that differ in the number of features, but have the same number of traits, I propose to calculate the entropy observed for each specific feature.

$$NormedFeatureEntropy = \frac{\sum_{i=1}^{T} p_i * \log_2(p_i)}{\log_2(1/T)}$$

An average of feature specific entropy measures can be compared across experiments, as long as the number of traits is the same.

One other technical issue is the scheduling of the actions of the agents. Axelrod's simulation was designed so that one agent is chosen at random for a possible interaction in each time period. As noted by Axtell, et al (1996, p. 190), the outcome of the simulation over the long term can depend on seemingly unimportant assumptions such as this one. Because this project is intended to investigate the simultaneous interaction of many agents in a complex system, the scheduling of actions is done by randomly sorting the agents at each time period and then processing all of them in sequence. Each interaction reflects the impact of all previous interactions within the time step of the model.

3. What if they really were villages? Using the WrappedListGrid2d class described above, this model represents a square grid in which each cell can contain several individual agents. As the simulation proceeds, the list of all citizens is randomly sorted and each agent is given the instruction to select a possible discussant and then initiate an interaction. As in the original Axelrod culture model, an interaction occurs with probability equal to the similarity of the cultures of the two agents.

To reproduce Axelrod's original results, one set of models is designed with one agent per cell and prospective discussant are selected at random from the four adjacent cells. The snapshot in Figure 1 shows a simulation of the Axelrod culture model (ACM) in a 10 by 10 cellular grid with 5 features (issues) and 5 traits (positions per issue). The "raster" displays on the right hand side (labeled "feature00" and so forth) use color to indicate the trait (opinion) of the agent in that cell on that feature. The simulation stops when ten cycles through the list of agents are completed without any agent changing its opinion. This snapshot is taken at the end of the simulation. Each raster indicates that a single viewpoint is dominant and the normed entropy indicator is zero.

A replication of this model for various values of the parameters matches the substantive highlights of Axelrod's results (see Table 1). For each setting of the parameters, the table indicates the averages of the model's duration (number of iterations), the total and feature-specific entropy values at the stop time, as well as averages of the perceptions of the individuals. The perceived acquaintance value represents the proportion of the time that the agent has tried to initiate a conversation and succeeded. The other two measures summarize interactions. Harmony is the percentage of features that the agent shares with acquaintances. The Identical percentage is the percentage of acquaintances that are identical with the agent on all features. To summarize the results, there is not much diversity in models with a small number of features and adding to the number of features reduces diversity still further. Increasing the number of traits increases diversity somewhat, but not in a striking way.

The entropy figures are abstract and perhaps a brief illustration of their meaning will help. Consider a single feature that has 5 possible values. If we have 100 agents and they are distributed (20,20,20,20,20), then the feature-specific entropy value is 1.0. Some other distributions, and their entropy values, are provided for comparison purposes.

Distribution of opinions	Entropy Value
(97,1,1,1,0)	0.12
(99,0,0,0,1)	0.034
(70,0,0,0,39)	0.37
(50,0,0,32,18)	0.63

When the entropy level is so small as in the runs in Table 1, one is probably safe in concluding that only a trivial amount of diversity remains after the model reaches its resting state.

Does increasing the number of agents in each cell change the prediction of the model? When there are many agents in each cell, then each agent chooses a prospective discussant in the following way. First, with a given probability, the agent decides whether to select the discussant from its own village. If the agent decides to look for a discussant elsewhere, then one of the four neighboring villages is chosen at random, and a random occupant is selected from that village. The "parochialism" factor, the probability that the agent will choose a discussant from its own village, can vary from 0 to 1. If it is set at 1, then each village is autonomous and there is no homogenization across villages. On the other hand, if the parochialism factor is 0, then nobody ever talks to anybody in their own village. If it is set at 0.2, then the agent is equally likely to stay in the village as to initiate an interaction with a person from one of the four neighboring cells.

It is not known how high the parochialism factor must be in order to sustain diversity across cells. Certainly 1 is sufficient. Experimentation found that 0.2 is too small, in the sense that entropy declined to 0, as in the original ACM. By raising the parochialism factor to 0.5, the homogenization of opinion is somewhat reduced, but certainly not blunted. Figure 2 shows what happens with 5 features, 5 traits, and 10 agents per cell. The model takes a significantly larger amount of computer time (duration), but the result is the same, complete homogeneity.

A summary of 100 replications of these computations is presented in Table 1, along side the original culture model. The left side of the table presents results for the "no parochialism" model, which means that an agent is equally likely to look for a discussion partner in its own cell or in a neighboring cell. The right side of the table summarizes results for a model in which each agent looks within its own cell with probability 0.5, and looks in the neighbors with equal likelihood after that. One can see that after raising the parochialism factor, there is only a sight increase in long-run diversity. Entropy in a model with five agents per cell (features=5, traits=5) is increased from 0 to 0.0077, a statistically significant increase by any test, but certainly not a substantively significant increase.

4. What if there are small endogenous networks? In a second set of simulations, the agents are given the power to be selective about their discussion partners. The model is supposed to represent a part of the logic of interaction outlined by Huckfeldt and Sprague. At a given moment, people have an opportunity to discuss issues with people that are physically accessible to them. At work, church, home, or elsewhere, one finds a group of possible discussion partners and then chooses among them. The agent in this model has a limited ability to choose the most appealing discussion partner, which in this case is the expected similarity of opinion with the agent (but it could be enriched to include other personal traits, such as expertise).

The world and the agents are created through the same random process as described for the culture model. If there is only one agent per cell, then the agent simply makes a list of the four neighbors and chooses to discuss with the one who is expected to be the most agreeable. If there are many agents per cell, then the agent builds a list of possible discussants by taking a random selection from its own cell and one from each of the four neighbors. (Parochialism was not investigated in this model.) The agent then sorts them according to their agreeability and considers the most agreeable one (or a randomly chosen one among those equally most-agreeable) for a possible interaction. As in Axelrod's original model, an interaction occurs with probability equal to the similarity of the two agents. This is operationalized by selecting a single feature (issue) and testing to find if the two agents have the same trait (position). If they do, then one of the issues on which they disagree is randomly selected and the agent copies that position from the discussion partner.

At beginning of the simulation, no two agents have met, and so all of them assume the others are equally attractive and choose randomly among them. As the simulation proceeds, each agent keeps an internal record summarizing its experiences with other individuals. If they interact, which means they share opinions on all of the issues, then the agent can make a record of the similarity of the other's opinions with its own. In the future, when an agent is considering a list of possible discussants, these records are consulted and used to rank them potential discussants.

If another agent is not in an agent's diary, then the agent has to make a guess about how agreeable the other is likely to be. Intuition might lead one to expect that this assumption will play a pivotal role in the development of the model. If each agent assumes that any stranger is completely disagreeable, then the agent will choose to interact with a stranger only when the list of possible discussants includes only strangers or acquaintances who are totally disagreeable. On the other hand, if the agent assumes that strangers are quite likely to be agreeable, then the agent may choose to interact with them when there are in fact more agreeable acquaintances.

One of the surprises to be found in the numerical analysis of this model is that, as long as one does not assume strangers are highly likely to be agreeable, then it makes little difference what one assumes about them. Consider the model depicted in Figure 3. This shows the same 10 by 10 grid as in the culture models. The only difference is that the agents are selective and, furthermore, they assume that an unfamiliar agent is completely disagreeable. In contrast to the total homogeneity induced by interaction in the ACM, in this model a fairly high amount of diversity is preserved. Lest one think that the distrust of strangers is a driving force in this model, compare it with Figure 4, which shows what happens with the same agents when it is assumed that a stranger will agree on one half of the issues (when there are 5 features, half means 2 because of rounding).

Curiously, when agents are added to the model, the diversity-enhancing effect of individual selection is reduced. When there are 5 others per cell, diversity is significantly reduced, as shown in Figure 5. These calculations used a parochialism level of 0.5. The level of diversity drops to a lower level, but it does not vanish over the long run. The effect of increasing the number of people in each village is to make it more likely that any given agent will be confronted with a list of strangers or undesirable acquaintances. This exposes the agent to the buffeting influences that cause homogeneity in the ACM. The summary statistics for a batch of these models are presented in Table 2.

In these small network simulations, the level of diversity is significantly higher than in the different variants of the culture model. Adding agents to each cell increases the chance of "unselected" interactions and thus allows the homogenizing tendencies of the culture model to operate.

There is one highly interesting pattern that is illustrated in these models. Even when their society is in fact quite diverse, the individual agents perceive a great deal of homogeneity within their sphere of activity. The perceived rates of acquaintance and harmony with acquaintances rise much more quickly than the entropy level declines.

2. A Model of Protest.

In contrast to the individual-level networking described in the previous model, there is a class of models in which individual opinion and behavior are keyed to social aggregates, rather than opinions of particular individuals. In the "threshold" model of protest, Granovetter describes individuals who may decide to take action only if a sufficiently high number of other individuals are currently acting. This model has a number of implications that have been explored (Granovetter, 1978).

Lohmann argues that the Granovetter model falls short in at least two respects. One weakness is the uninteresting nature of the model's prediction about the dynamics of the protest itself. A protest can grow up to a point, but it cannot shrink. The second weakness, which is no doubt the source of the first, is the model's simplistic view of individual level. It seems there is no doubt that both of these weaknesses are serious. The question is how we ought to remedy them.

Lohmann presents a game theoretic model that imposes quite high demands on the information processing abilities of individual agents. Agents observe a signal at the outset about the desirability of the government. Individual protest action creates an "information cascade" through which the actions of other agents are changed due to their observations. The sequential equilibrium approach that is used to represent this process imposes extremely high requirements on the individual rationality of the actors. Each actor formulates a plan of action that describes her willingness to protest at time t if the number of protestors at t-1 is X_{t-1}. These personal strategies are optimal in the following sense. At any point in time, suppose each agent were given a copy of each of the other agent's strategy (the complete "contingency plan" showing what it would do at all future timepoints under all possible conditions). On the basis of that detailed information, each agent sees no reason to deviate from its announced plan. Someone who believes the government is bad will protest if she believes her action is likely to topple the government or that it will cause somebody else to topple the government.

The agent-based alternative need not impose such a demanding set of standards. In particular, it does not suppose that each agent is privy to the long-term action plan of each other agent. Rather, it supposes that each agent has rather limited information obtained by observing the level of protest in society at large. Furthermore, there is no need to assume that any agent believes their action is likely to play a pivotal role. Rather, people protest when they believe with a sufficiently high likelihood that the government must go. The agent uses its personal condition as an anchor in evaluating the current government. When there is discontent in a neighboring vicinity, then the individual might be more inclined to think the government is bad. And if that belief becomes sufficiently strong, then the person may join the protest in their areas.

The protest model, a snapshot of which is shown in Figure 6, has the following ingredients. First, agents are created and randomly (uniformly) placed in a square grid. Each agent is assigned initial values for a number of variables, most importantly their "personal condition." Personal condition varies from 0 to 1, as determined by a Beta distribution with coefficients adjusted so that the agents in the top left region of the display have the lowest expected values for personal condition and the agents in the bottom right have the highest expected value. The model also assigns a threshold for each agent. The threshold, distributed uniformly on the interval [.5,1], is an element in the decision to protest.

As the simulation proceeds, each agent does three actions in each time step. These are:

1. Update beliefs.
2. Consider joining the protest.
3. Choose a location.

The updating of beliefs is based on the personal observation of discontent within the agent's "vision radius." If the vision radius is 1 unit, then the agent is able to discern the level of protest activity only in its own cell and the 8 adjacent cells. If the vision radius is 2, then the agent can observe its own cell as well as the "Moore neighborhood" formed by making a square that extends 2 cells on all sides of the agent's cell. The agent is able to estimate the percentage of other agents who are currently protesting. Unlike the culture model, this models uses a toroidal model of the world that "wraps around" to join the extreme edges of the model.

A number of functions can be considered with which agents might calculate the probability that the government is bad. A logistic function is one appealing alternative:

$$probabilityRulerIsBad = \frac{1}{1+e^{(-0.2+(1.0-personalCondition)+0.01*localDiscontent}}$$

The higher an agent's personal condition, the less likely it is to think the government is bad. If the perceived protest percentage within the range of vision, then the probability is increased. After the agent updates its beliefs, it then decides whether to join the protest. If the government is believed to be "bad" with probability higher than the threshold, then the agent will protest. A number of additional specifications and variables have been considered.

When the agent decides whether or not it will protest, then it can choose a new location in the grid. Agents can move only one square at a time. Agents that are not protesting look in their vicinity for the least tumultuous cell and move there in order to be safe. Agents who want to protest look around for the cell with the highest percentage of people currently protesting, so they can be in the company of as many like-minded people as possible.

There are a number of purely-technical modeling complications that arise in the scheduling of agent actions. In particular, the choice of synchronous or asynchronous updating is significant. For example, should each agent's decision to protest and move be observed by all of the other agents while they are making their decisions? The computer code has been written to compare these two alternatives, but so far no significant difference between them has been found. The models reported here use asynchronous updating.

In Figure 6, a screen shot of the protest model is presented. The raster graph, labeled "Protest Grid," is intended to convey a number of pieces of information. The background is black in a cell that is not occupied and is not near any protest activity. A cell is blue if it is occupied by people who are not protesting. As the percentage of people who are protesting in a cell increases, the color of that cell turns to a dull green and then a bright green color. To illustrate the "contagion effect" of protest activity, the unoccupied cells are tinted by shades of red to indicate the intensity of protest that is occurring within 2 cells in any direction that cell. If a cell is bright red, it means that one hundred percent of the visible citizens are protesting. The accompanying graph shows the average of the agent’s beliefs that the government is bad and ought to go, along with the percentage of citizens who are protesting at a given time.

Because the people who are not protesting look for quiet places to occupy, they tend to collect and, as shown in Figure 6, form "firebreaks" that appear to dull the spread of the protest movement. Since the agents have limited vision, there are some agents who might protest, but do not do so because they are unable to "see past" a packet of calm.

As shown in Figure 6, the model produces a slow, steady growth of protest activity. There are slight changes in the specification that can make the protest grow more suddenly. For example, if we cause agents to protest when protest within their vicinity surpasses some pre set level, then the conflagration will sweep through most of the population. Figure 7 is produced from the same model (including the stream of random numbers) as Figure 6, except in 7 a person will join the crowd as soon as more than 50 percent of the people within view are protesting. After 150 periods, only a small spot of tranquility remains in the lower-right (well-off) section of the graph.

So far, the model makes no special assumptions that could cause the level of protest to rise and then fall. Agents can withdraw from the protest if their beliefs change, however, that is unlikely. Since protesters move to "hot spots," seeking other protesters, their beliefs can change only if there is an influx of nonprotesters. Additional assumptions about the behavior of agents over a period of time can, of course, change this. One example to be considered is that agents can become exhausted, and after a number of periods of protesting, they find they would rather go back to school or their jobs. There is empirical support for this characterization (Francisco, 1999). With all conditions the same as in Figure 7, add the assumption that every agent is assigned an endurance parameter which varies randomly from 3 to 10. The agent will protest only if the total number of periods spent protesting does not exceed the endurance parameter. In Figure 8, the damping effect of exhaustion is quite clear. Instead of enveloping the whole population, the protest movement grows and then dies in a small number of periods.

The ability to investigate "what if" questions about slight changes in the model specification is one of the major advantages of the agent-based modeling enterprise.

Comments and Conclusion

In the first part of the paper, it was observed that it can be difficult to tell a surprising result from a specification error. It is also difficult to know what to assume while building one of these models, and perhaps it is more difficult to justify these decisions. Unlike a more well-established modeling approach, agent-based modeling lacks a core of "tropes" and standard rationalizations that can serve to facilitate research and the exchange of ideas.

In this paper, two sorts of models have been considered. Models in which people learn about the state of the world though detailed conversations with individuals shown to have implications that depend on how individuals are linked together through discussion networks. If people are exposed in a completely unstructured way to interaction with ideas that may differ from theirs in a significant way, it seems that the total homogenization of public opinion is to be expected. On the other hand, by supposing that people are able to recognize potential discussion partners and select among them, we arrive at a more satisfying conclusion that the following things can coexist.

1. Agents are able/willing to change their minds in response to input from outside.
2. Substantial social diversity exists.
3. Individual agents believe that most people think like they do about most things.
4. In a long-running social system that is not subjected to external shocks, agents are seldom observed to change their opinions.

In the so-called protest model, the conclusions are not so clear and, to be honest, the model is not so well investigated. The first cut at the model indicates that the original description of the model was incomplete. The effort to explain the growth of a protest movement with detailed individual models of adjustment in beliefs and comparison against personal thresholds falls short. In order to make the model generate the desired aggregate results, new elements must be introduced at the individual level, such as personal exhaustion.

References

Axelrod, Robert. 1997. The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton, NJ: Princeton University Press.

Axtell, Robert, Robert Axelrod, Joshua M. Epstein, and Michael D. Cohen. 1996. Aligning Simulation Models: A Case Study and Results. *Computational and Mathematical Organization Theory* 1: 123-41.

Baltch, Tucker. 1998. Social Entropy: An Information Theoretic Measure of Robot Team Diversity. Unpublished Dept of Computer Science, Carnegie Mellon University.

Epstein, Joshua M. and Robert Axtell. 1996. Growing Artificial Societies. Washington, DC: Brookings Inst.

Francisco, Ronald. 1999. Fast, Punctuate, and the Red Queen: Time in the World of Protest & Coercion. Paper presented at the Annual Meeting of the Midwest Political Science Association, Chicago, IL.

Granovetter, Mark. 1978. Threshold Models of Collective Behavior. American Journal of Sociology, 83: 1420-1433.

Huberman, Bernardo A., and Natalie S. Glance. 1993. Evolutionary Games and Computer Simulations. Proceedings of the National Academy of Sciences U.S.) 90: 7716-7718.

Hegselmann, Rainer. 1996. Cellular Automata in the Social Sciences. In Rainer Hegselmann, Ulrich Mueller, and Klaus G. Troitzsch, eds., Modeling and Simulation in the Social Sciences from the Philosophy of Science Point of View. Dordrecht, Kluwer Academic Publishers, pp. 209-233.

Huckfeldt, Robert, and John Sprague. 1995. Citizens, Politics, and Social Communication. New York: Cambridge University Press.

Huckfeldt, Robert. 1999. The Social Communication of Political Expertise: Individual Judgments Regarding Political Informants. Unpublished Ms.

Lohmann, Susanne. 1994. The Dynamics of Informational Cascades: The Monday Demonstrations in Leizig, East Germany, 1989-91. World Politics 47: 42-101.

McPhee, W., and R. Smith. 1962. A Model for Analyzing Voting Systems. In W. McPhee and W. Glaser,eds. Public Opinion and Congressional Elections. New York: Free Press, pp. 123-179.

Nowak, Andrzej, and Maciej Lewenstein. 1996. Modeling Social Change with Cellular Automata. In Rainer Hegselmann, Ulrich Mueller, and Klaus G. Troitzsch, eds., Modeling and Simulation in the Social Sciences from the Philosophy of Science Point of View. Dordrecht, Kluwer Academic Publishers, pp. 209-233.

Shannon. C.E. 1949. The Mathematical Theory of Communication. Champaign-Urbana, University of Illinois Press.

Table 1

Culture Simulation (Averages of 100 Runs)

Features=5	No Parochialism, Traits=5, Ppl/Cell=1	Ppl/Cell=5	Ppl/Cell=10	Parochialism=0.5, Traits=5, Ppl/Cell=5	Ppl/Cell=10
Total Entropy	549.38	2300.4	4814.6	2209.28	4805.66
Agv.FeatureEntropy	0.001	0	0	0.0077	0
Pct.Acquainted	0.0053	0	0	0.009	0
Pct. Harmony	0.99	0.99	0.99	0.99	0.99
Pct.Identical	0.99	0.99	0.99	0.99	0.99
	0.97	0.99	0.99	0.96	0.99

Features=5	Traits=10, Ppl/Cell=1	Ppl/Cell=5	Ppl/Cell=10	Traits=10, Ppl/Cell=5	Ppl/Cell=10
	628.12	2518.4	5665.53	2525.02	5515.57
	0.028	0.000012	0	0.000012	0
	0.13	0.000062	0	0.000078	0
	0.92	0.99	0.99	0.99	0.99
	0.98	0.99	0.99	0.99	0.99
	0.96	0.98	0.99	0.98	0.99

Features=10	Traits=5, Ppl/Cell=1	Traits=5, Ppl/Cell=5
	715.2	3540.13
	0	0
	0	0
	0.99	0.99
	0.99	0.99
	0.96	0.98

Features=10	Traits=10, Ppl/Cell=1	Traits=10, Ppl/Cell=5
	854.3	4136.2
	0.00036	0
	0.0036	0
	0.99	0.99
	0.99	0.99
	[illegible]	[illegible]

Table 2

Voluntary Networks Model (Averages of 100 Runs)

Features=5	Traits=5 Ppl/Cell=1	Traits=5 Ppl/Cell=5	Traits=5 Ppl/Cell=10	Traits=10 Ppl/Cell=1	Traits=10 Ppl/Cell=5	Traits=10 Ppl/Cell=10
Duration	60.67	2657.77	1675.5	68.75	2205.6	2470.45
Tot.Entropy	0.42	0.2	0.044	0.39	0.16	0.097
Avg.FeatureEntropy	0.94	0.67	0.17	0.96	0.63	0.36
Pct. Acquaintance	0.88	0.99	0.99	0.11	0.98	0.97
Pct.Harmony	0.9	0.99	0.99	0.35	0.99	0.98
Pct. Identical	0.88	0.99	0.99	0.11	0.99	0.97

Features=10	Traits=5 Ppl/Cell=1	Traits=5 Ppl/Cell=5	Traits=10 Ppl/Cell=1	Traits=10 Ppl/Cell=5
Duration	69.78	11,648.03	103.51	6285.84
Tot.Entropy	0.19	0.13	0.18	0.085
Avg.FeatureEntropy	0.94	0.78	0.95	0.63
Pct. Acquaintance	0.98	0.99	0.34	0.99
Pct.Harmony	0.98	0.99	0.46	0.99
Pct. Identical	0.98	0.99	0.34	0.99

Figure 1
Culture Model: One Person Per Cell

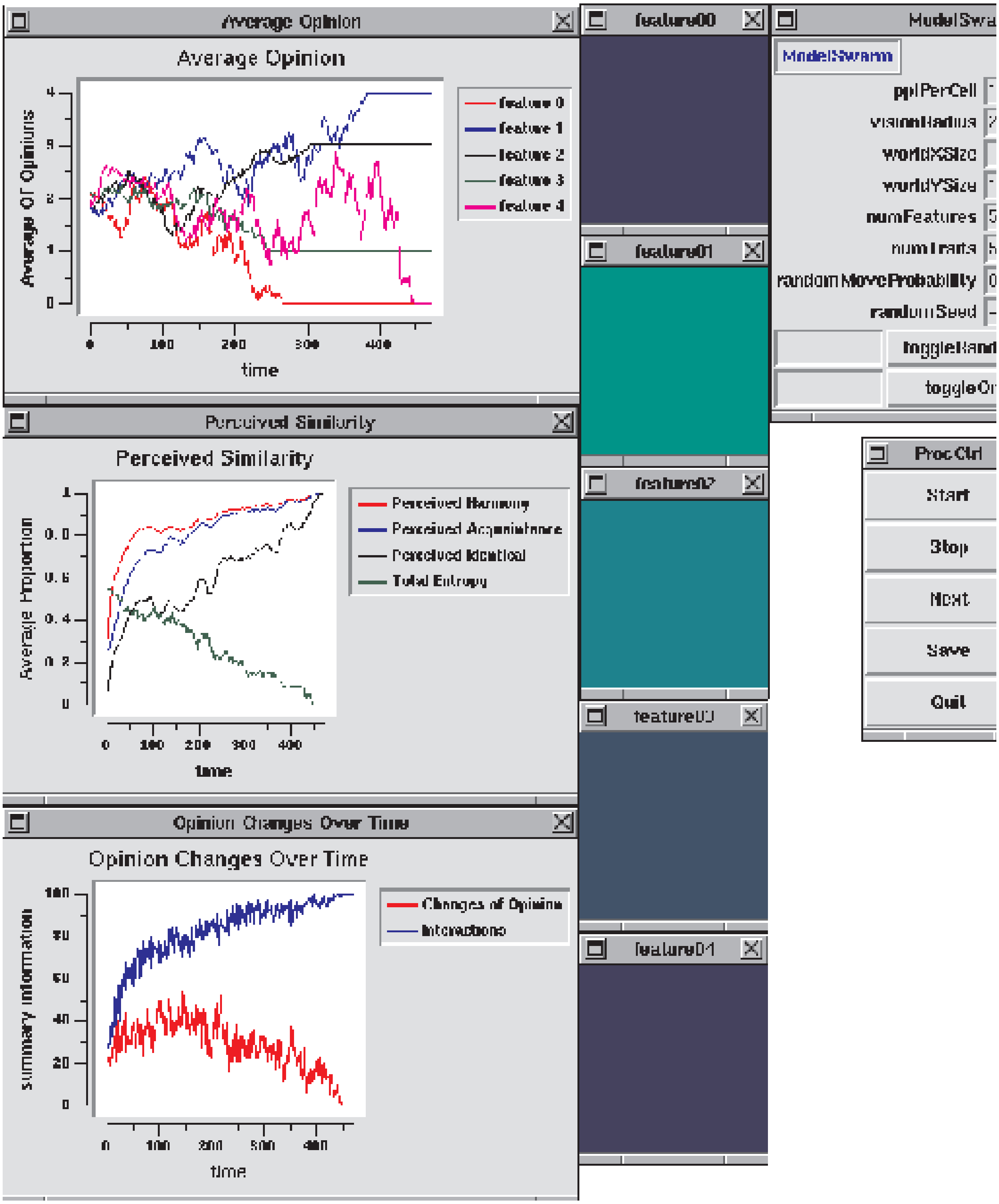

Figure 2
Culture Model: Ten Agents Per Cell

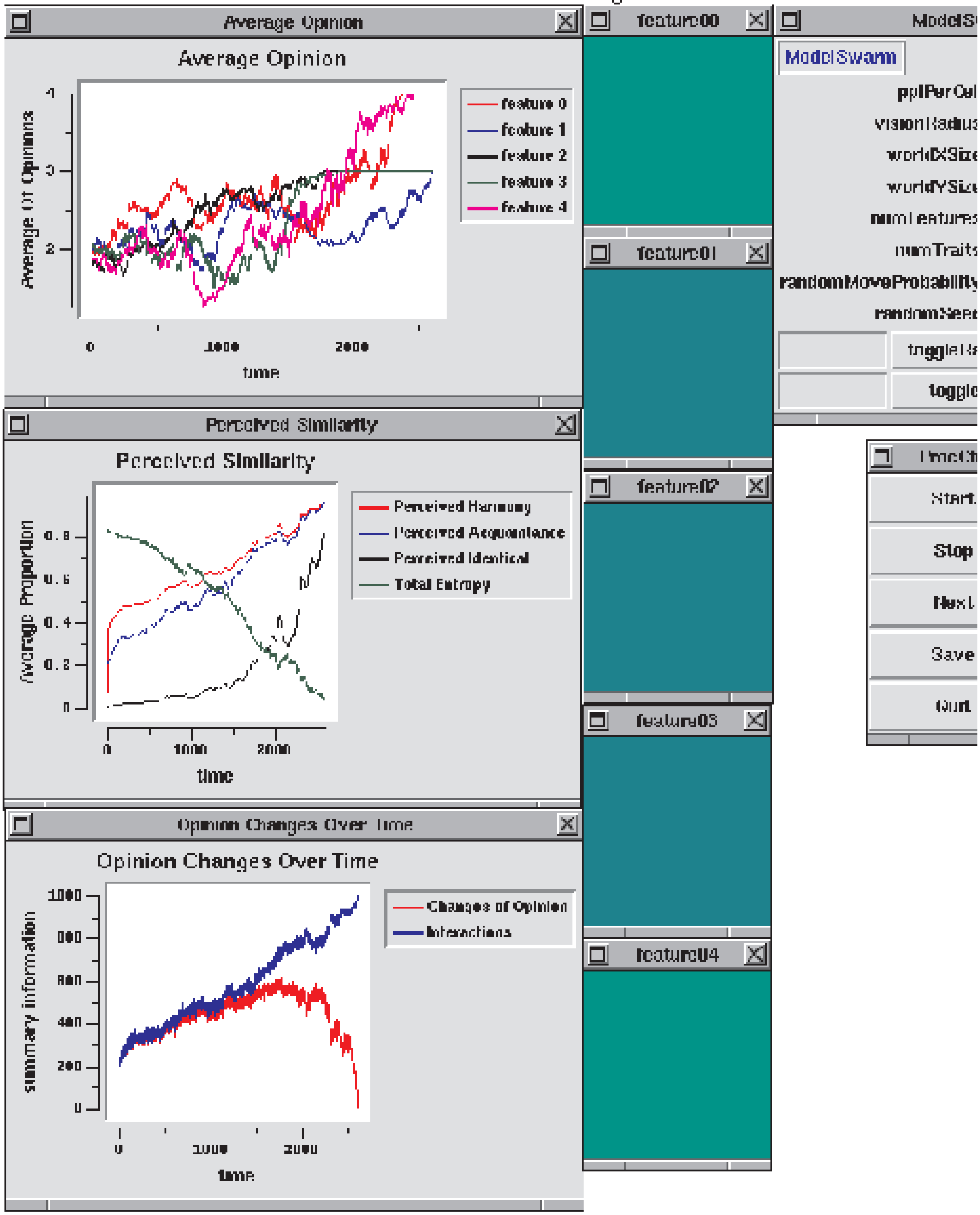

Figure 3
Small Endogenous Networks with One Selective Agent Per Cell

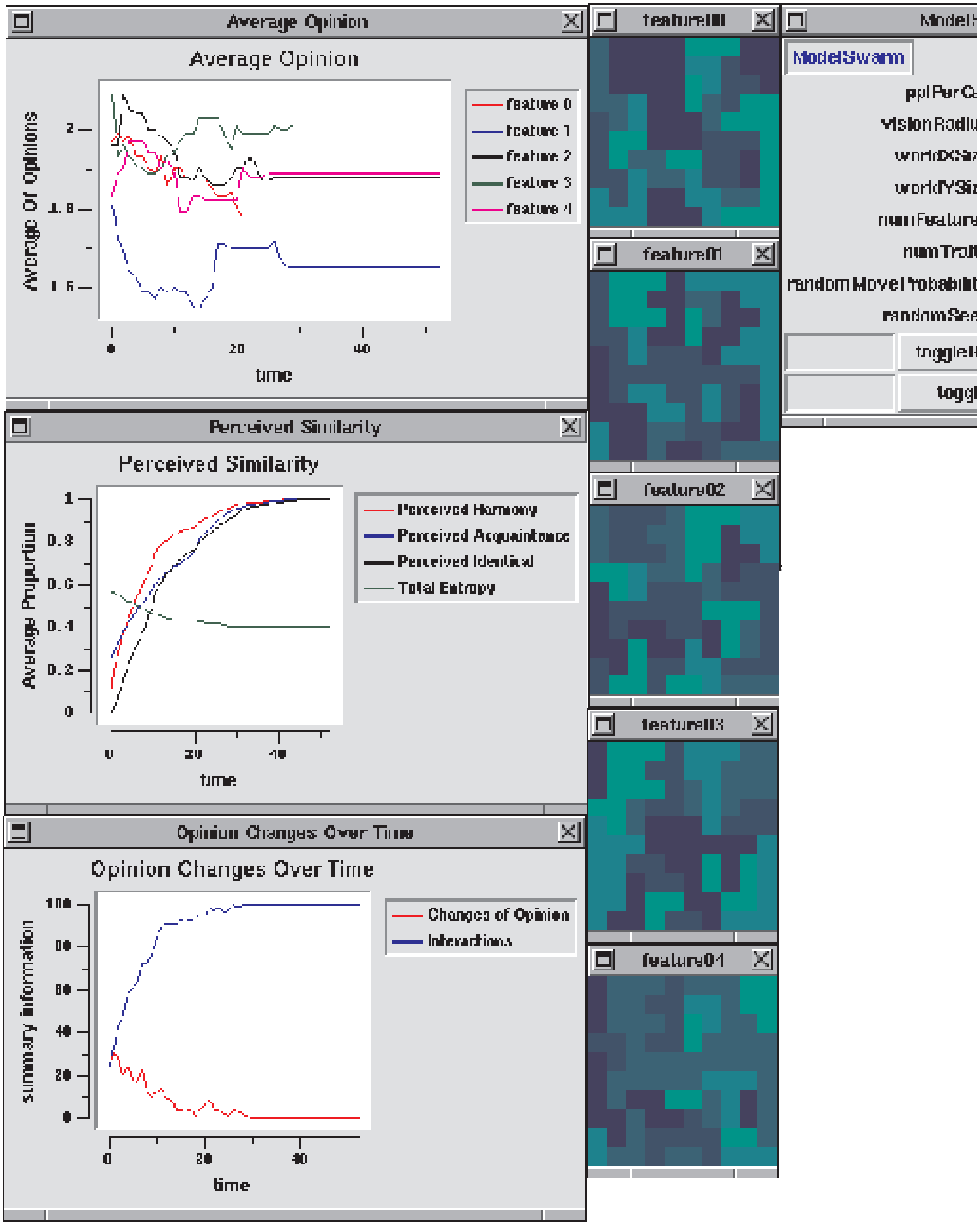

Figure 4
Small Endogenous Networks with Strangers Assumed Agreeable

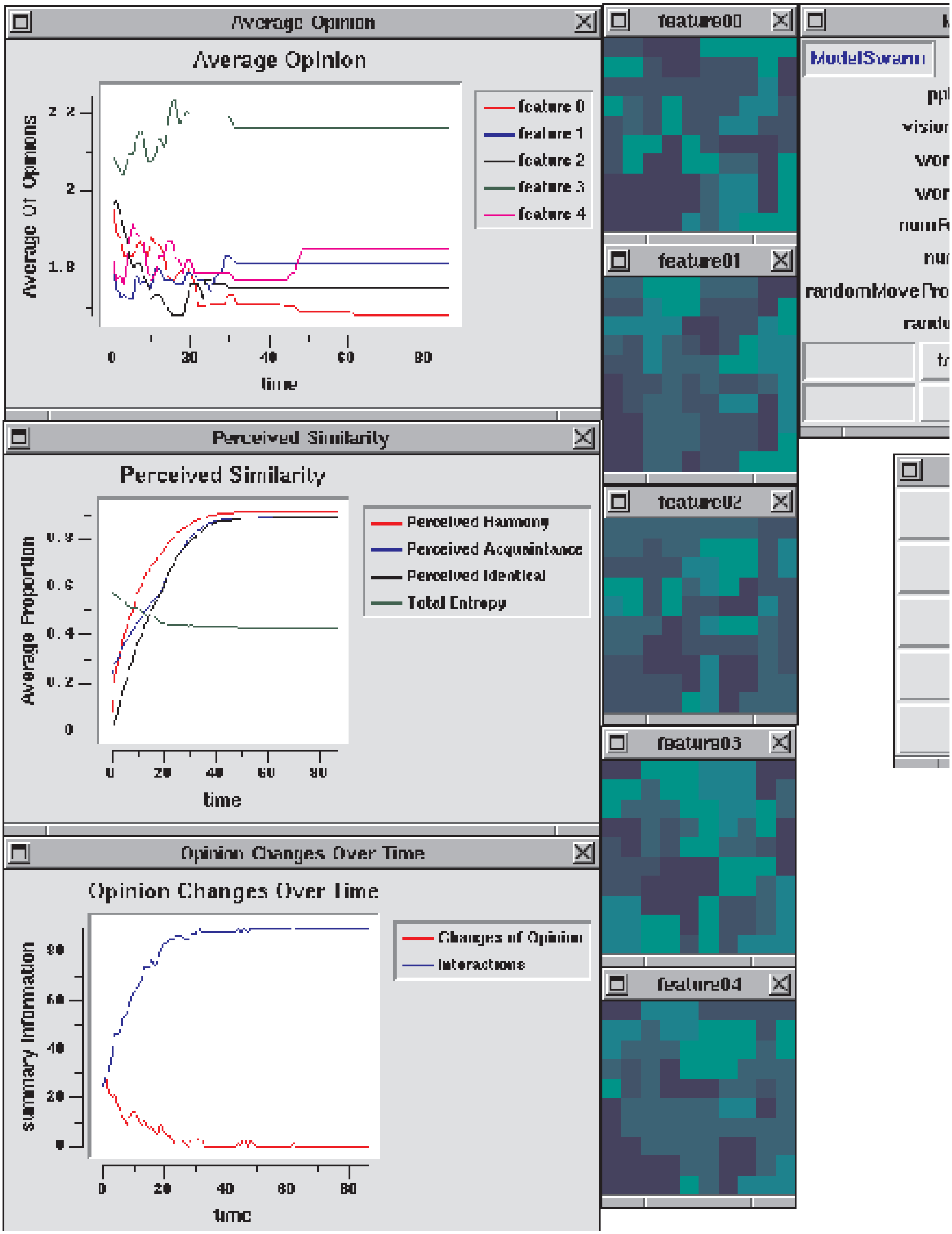

Figure 5
Small Endogenous Networks with Strangers Assumed Agreeable (5 Agents Per Cell

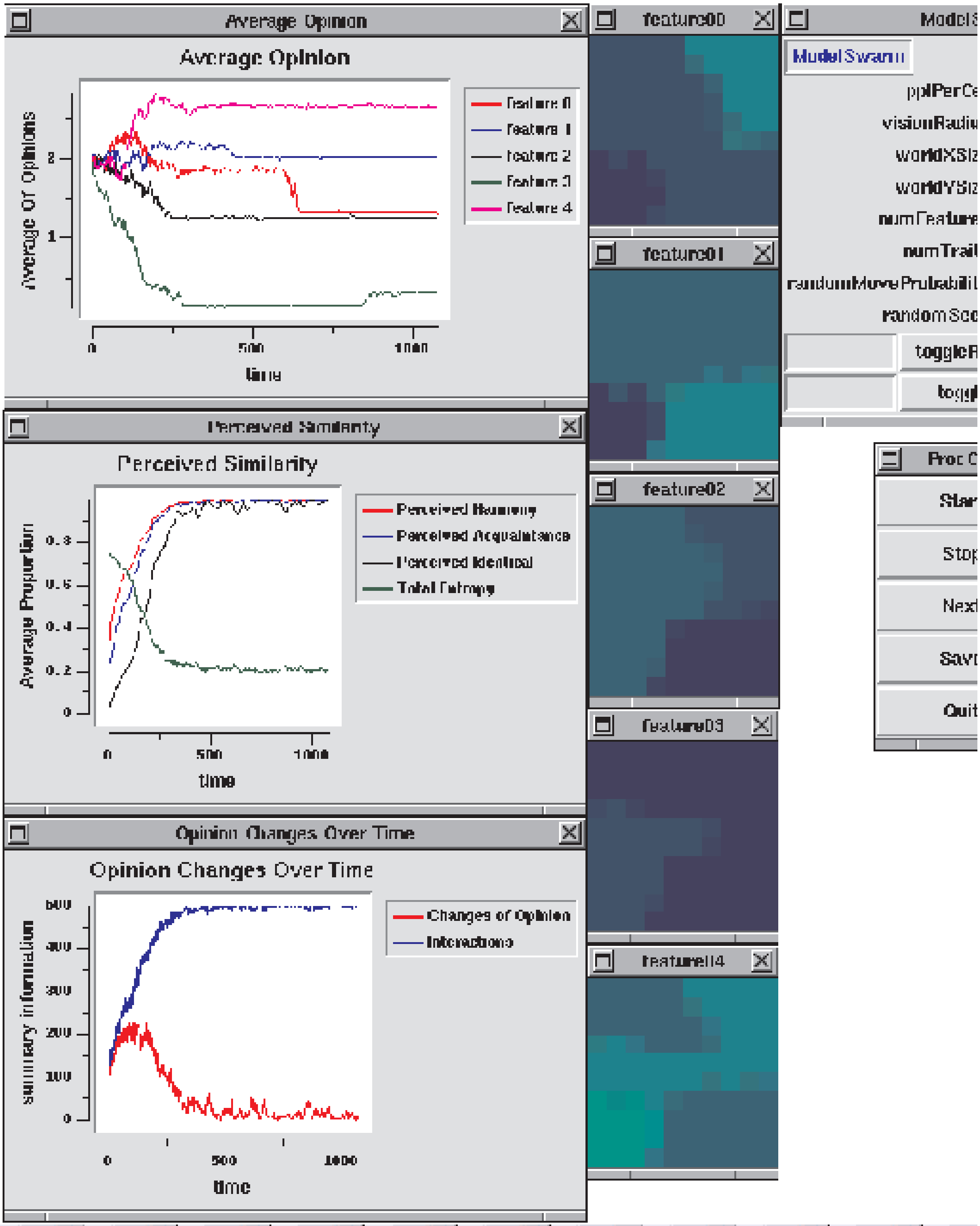

Figure 6

Contagious Protest

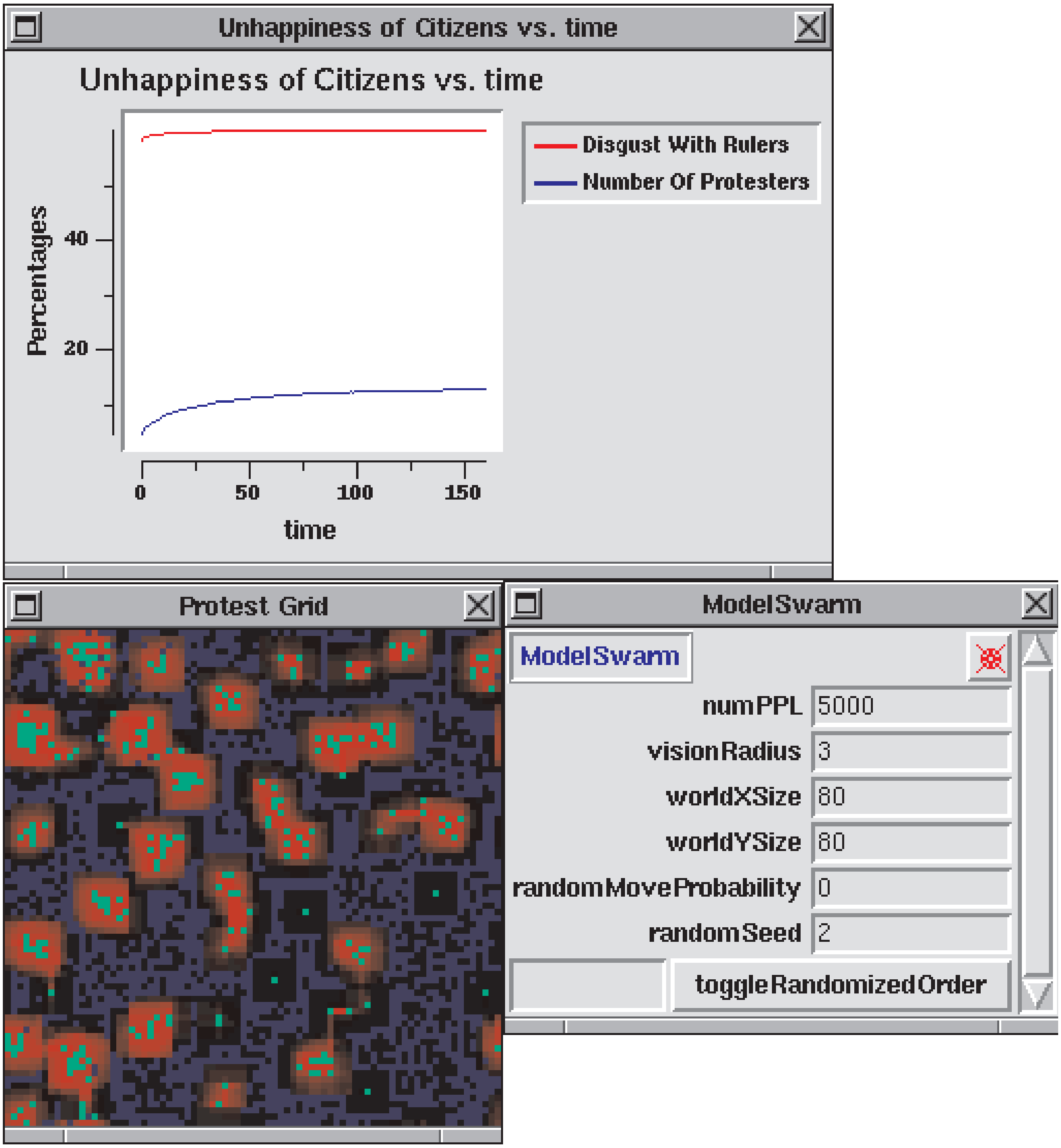

Figure 7

Protest Model with an Added Threshold Component

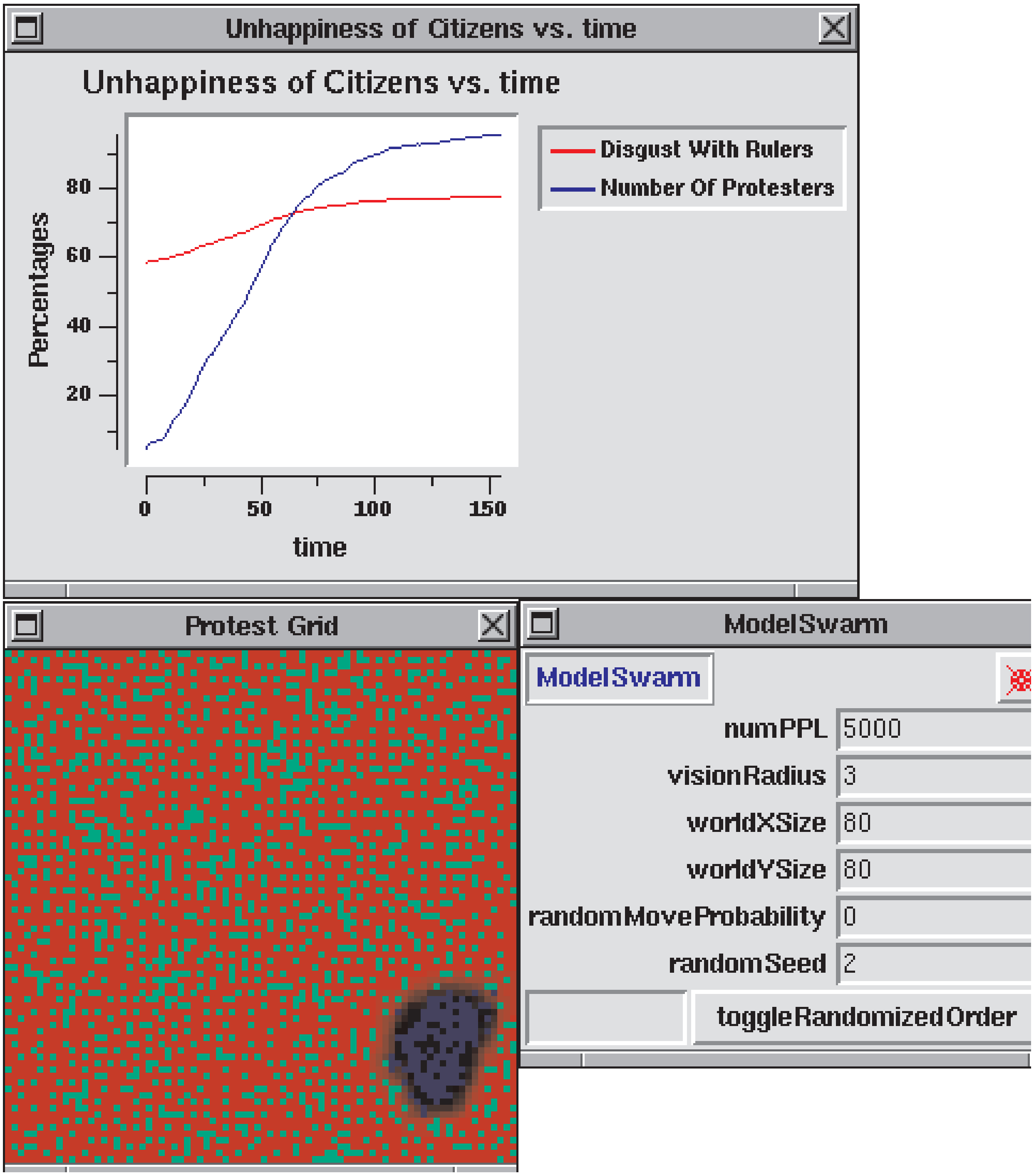

Figure 8

Protest Model with Exhaustion Factor