May 2006

Computer Simulation of Crowd Dynamics and Destructive Crowd Behavior

Bruce Bassi
University of Connecticut

Computer Simulation of Crowd Dynamics and Destructive Crowd Behavior*

Bruce Bassi – University of Connecticut
Advisor: Dr. C. Wesley Younts– University of Connecticut

Paper submitted in fulfillment of the requirements for graduation as a University Scholar, University of Connecticut.

April 28, 2006

* This research would not have even begun if it was not for C. Wesley Younts' openness to new ideas. Dr. Younts has guided this research from start to finish (well, a temporary finish, since we think that this simulation can always be expanded). I would like to recognize Ruth Ungar (Computer Science) and Peter Turchin (Environmental and Evolutionary Biology) for their advice and help in editing. Graduate student in computer science, Alan Wong, has also provided me with advice in planning the simulation. I would also like to acknowledge the support provided by the University of Connecticut Honors Program in the form of a SURF grant during the summer of 2005. This paper is dedicated to my father, who has paid over $40,000 for it. That would make it the most expensive paper I have ever made, at about $800 per page.

Computer Simulation of Crowd Dynamics and Destructive Crowd Behavior

Abstract

The social processes that lead to destructive behavior in celebratory crowds can be studied through an agent-based computer simulation. Riots are an increasingly common outcome of sports celebrations, and pose the potential for harm to participants, bystanders, property, and the reputation of the groups with whom participants are associated. Rioting cannot necessarily be attributed to the negative emotions of individuals, such as anger, rage, frustration and despair. For instance, the celebratory behavior (e.g., chanting, cheering, singing) during UConn's "Spring Weekend" and after the 2004 NCAA Championships resulted in several small fires and overturned cars. Further, not every individual in the area of a riot engages in violence, and those who do, do not do so continuously. Instead, small groups carry out the majority of violent acts in relatively short-lived episodes. Agent-based computer simulations are an ideal method for modeling complex group-level social phenomena, such as celebratory gatherings and riots, which emerge from the interaction of relatively "simple" individuals. By making simple assumptions about individuals' decision-making and behaviors and allowing actors to affect one another, behavioral patterns emerge that cannot be predicted by the characteristics of individuals.

The computer simulation developed here models celebratory riot behavior by repeatedly evaluating a single algorithm for each individual, the inputs of which are affected by the characteristics of nearby actors. Specifically, the simulation assumes that (a) actors possess 1 of 5 distinct social identities (group memberships), (b) actors will congregate with actors who possess the same identity, (c) the degree of social cohesion generated in the social context determines the stability of relationships within groups, and (d) actors' level of aggression is affected by the aggression of other group members. Not only does this simulation provide a

systematic investigation of the effects of the initial distribution of aggression, social identification, and cohesiveness on riot outcomes, but also an analytic tool others may use to investigate, visualize and predict how various individual characteristics affect emergent crowd behavior.

Computer Simulation of Crowd Dynamics and Destructive Crowd Behavior

Introduction

Celebratory riots, unlike other forms of civil unrests (e.g., political protest, social protest), are commonly associated with unruly college students who want to join other students in recognizing a recent victory, to express grief on a loss, or to just spread school camaraderie. The opposite result, including physical damage to the community, personal humiliation, and worse, defamation to the institution, is an unfortunate and all-too-common occurrence. The negative consequences that are associated with riots are indeed important, and should not be overlooked. The intention of the simulation, however, is to help explain the cause of a riot, not the aftermath. While a celebratory disturbance at a college campus may involve the typical crimes (e.g., possession of alcohol by a minor, or public drunkenness), not all riots begin in the same manner.

On any given day, students at the University of Connecticut are unlikely flip a car. However, after a national sports victory or during Spring Weekend[1] history has shown that students are more likely to riot. Given some rare and significant event with the right ingredients in the crowd, a riot is likely to break out. Students have rioted after winning the national basketball championship in 1998 and 2004, and have been in a near-riot state during on Spring Weekend of the last five years. In 1998, students caused nearly $150,000 worth of property damage and police in riot gear made over 60 arrests. In 2004, there were 60 arrests, including 23 students following NCAA basketball championship [6]. During the years in between, there were

[1] Students typically use Spring Weekend to relieve stress before final exams. School sponsored events include free massages, ice cream socials and concerts. The non-school sponsored events have become such a part of the school's history that it is a tradition to gather at certain locations around campus on Thursday, Friday, and Saturday nights. Crowds can be expected to reach sizes of 25,000 or more [9].

between 29 and 105 arrests during Spring Weekend [9]. Although the creation of this simulation has been prompted by these riots around the University of Connecticut, celebratory riots are not unique to just this university [1,2]. The number of institutions that must deal with this relatively recent (since 1990) trend is increasing (see *Figure 1* of the Appendix). For a list of the universities which have had at least one celebratory riot between 1998 and 2003, please see the Appendix.

Student unrest is nothing new, and is actually as old as the oldest universities. Historically, student unrest has been triggered by social and political issues. For some perspective, some examples of the earliest student protests have been over the American Revolution in the 1770's, institutional policies in Virginia in 1828, and bad food at Harvard 1834. In the 1960's and 1970's student rioting took on a more political connotation—typically with anti-war protests. Recently, however, it has been closely associated with celebrations related to athletic events [24]. Rioting about something as frequent as a sporting event victory can be much more unnerving than a riot about a war. Championships are so frequent (given the number of NCAA and professional sports) that this raises a serious issue if rioting is just as common.

In response, universities typically develop task force groups which submit recommendations to school officials to help thwart these celebratory riots. Task forces, such as those developed in 1999 at the University of Connecticut, appear to be unable to prevent such riots, in light of the riots of 2004. School officials will continue to try to curb celebratory riots around college campuses, but given the inevitable underdog upset, the annual NCAA basketball national championship, and the routine use of alcohol during such celebrations, it may be in our best interest to continue to study riots.

On the bright side, students usually do not gather with the intentions of rioting. Acting alone, students would not become as violent as they do in a group [22] . As violent celebrations become more commonplace, however, students may sometimes feel that it is traditional to gather at a particular location in hopes of finding the must-see images: the overturned car, the dumpster fire, or the flash of exhibitionism. Over the past ten years, universities around the United States have become accustomed to dealing with such situations. Through a history of previous riots, students at a particular university may become predisposed to rioting. But relative to what could potentially happen later, students gather initially peacefully. Although not *necessary* for a riot, a fuel of alcohol, anonymity, and a school "tradition" of rioting can help tip a large gathering to a riot. Instead of focusing on these common factors, the simulation developed for this project examines initial conditions which are *sufficient* to cause riots.

The characteristics of a crowd can be generalized into a few adjectives that describe the interaction between its participants. A volatile crowd is one which contains a large proportion of rebels (i.e., those who lead others or otherwise go against the behavior of those around them), as opposed to a stable crowd, which contains a large proportion of sheep (i.e., followers). A convenient advantage of agent-based modeling is that the properties of individuals' interactions (e.g., relationship strength, identification), not just the properties of the individuals themselves, affect the chances of a riot occurring. The simulation was developed to manipulate these characteristics in order to see their effects on the chances of a riot occurring. The interaction of individuals in a crowd helps to propagate emotions and aggression. The more the individuals can relate to each other, the more influential they are and the more easily emotion and aggression can spread A crowd in a university setting implies the existence of a variety of categories of individuals—categories in which its members can easily relate. Members who can relate to one another very easily are likely to move in synchronization (i.e., form a cluster). The amount of interpersonal attraction defines how often individuals would want to cluster with a neighbor. Once the cluster is formed, the chances of the individuals staying together depends on their cohesiveness. The cohesiveness of a crowd can be used to describe the sticking force between group members. The greater the cohesiveness, the more likely participants will remain together. The simulation models the characteristics of the crowd through various initial conditions which the program uses in an algorithm to evaluate each actor's change in aggression. The user needs to be notified when a riot occurs so that there is a distinct and clear outcome. When a riot does not occur, the simulation should not run forever, so the simulation ends after a predetermined amount of time, after which it is assumed that a riot will not occur (e.g., the party is over). The main goal of the simulation is to investigate the relationships between the initial conditions of the simulation and the likelihood of a riot occurring. It was found, just as in real-life, that riots can develop through the interaction of a set of initially non-violent individuals, depending on the initial conditions (which define the characteristics of the crowd).

The paper first explores the theories that were used in developing the simulation. Different theories are implemented in the simulation so I also explain how each theory is relevant to the code. The code that was written in JAVA is transformed into easy-to-read equations in the methods section. In the discussion, I provide a more detailed discussion of the flexibility and future expansion of the program (which is arguably the most important part). For those interested in using the simulation, I provide a user's manual in the appendix which explains the graphical user interface.

Theory

The definition of the word *riot*, taken out of a dictionary, is relatively ambiguous. Dictionaries typically describe “a large number of people” who engage in a “wild or turbulent disturbance” [26]. Exactly how large and how turbulent is open for debate. This is why I will first define all key terms that are used in the model. Some terms will be defined later, as it is sometimes necessary to define the word in terms of some property of the simulation. The paper will then review relevant sociological theories in the order that they are implemented in the simulation. The simulation begins with a set of individuals who have been assigned an action choice and identity. The actors then interact and create small groups according to how easily they can relate with each other. This depends on their identity and aggressiveness. Individuals interacting with a cluster of individuals are affected differently than if they interact with other individuals, so the paper will then examine the effects of clustering. Individuals can effectively become more similar or more dissimilar to their neighbors. In other words, there exists a continuum over which an agent can change its behavior. This range of possible new aggression levels is dependent on the aggression levels of nearby actors. Therefore, physical proximity plays an important role in how an agent changes its behavior. The dynamic of every actor changing according to its nearby agents creates emergent properties which would not have been discovered solely from the rules of the agents. In particular, the chances of a riot occurring is function of the interpersonal attraction between agents, the cohesiveness within a group, and the proportion of rebels within the gathering.

Definitions of Relevant Terms

A large crowd of people does not form instantaneously, rather it starts from multiple smaller gatherings that join together. A *gathering* refers to the collection of two or more actors at the same socially defined time and place (i.e., an event). Gatherings and collective behavior are not synonymous, but gatherings provide the necessary medium for collective behavior. *Demonstrations* refer to collective behavior in the form of protest or celebration. *Riots* are a form of demonstrations with an additional element of violence against a person or property [3]. The media usually defines whether a gathering has rioted or not. A headline claiming that a riot has occurred implies that there will be catchy images of burning, looting, violence, etc. Violence is an activity which results in the deliberate physical damage of persons or property [21]. In the context of a riot, violence is seen as a collective action (e.g., multiple people are needed to flip a car). Therefore, in order for a number of distinct, aggressive individuals to riot, they must first collaborate and cluster.

Individual Motivation and the Distribution of Individuals in a Crowd

Crowds begin heterogeneously, and then can cluster and homogenize or polarize through interaction. In the case of a celebratory riot, the distribution of actors' aggression affects the extent of action, which is why sometimes aggression can be generalized into an actor's *action choice*. The term is self descriptive: it refers to the behavior that an actor is most likely to choose. It helps determine the milling time required for action and the probability of cluster formation [20].

Although no human characteristics are particularly crucial for violent crowd behavior, it is noted that the distribution of certain personalities is important. In turn, the consensus (i.e., the final outcome) of the crowd changes as the individuals who constitute the crowd change. The composition of the individuals in the crowd affects how the crowd would react. If most individuals are moderates, the modal action of the crowd would be as such unless someone more militant persuades the moderates to act. Johnson concurs with Berk, claiming that crowd members arrive on scene willing to engage in a variety of actions from extreme to moderate [20].

The motivations underlying the assembly phase of a gathering may differ for each individual, group of individuals, and gathering of people. In a celebratory riot such as those on a college campus, the group could be comprised of a wide variety of individuals. For example, at UConn there are undergraduates, graduate students, and non-students. The categories themselves are arbitrary; what really matters is that individuals can classify themselves in relation to other social categories. Similarly, sociologists stress the importance of the *process* in which individuals identify with each other rather than the entity with which individuals identify themselves [27].

Burke [28] explores how commitment helps explain the ways individuals take on certain roles and become a function of the surrounding social structure. Through commitment, people internalize expectations of a particular identity and deal with the implications of those particular roles. Through the concept of commitment, Burke’s theory describes how an actor deliberately chooses a particular behavior, as opposed to other theories that treat behavior as an uncontrollable (by the individual) byproduct of external forces. Commitment helps an individual to keep a particular identity because it plays into that role. A role that is, in the actor’s mind, consistent with that particular identity yields commitment. Commitment is the total driving force that causes people to maintain a consistency between their own identity and opposing appraisals (i.e., others' reactions to the identity one presents publicly). The empirically supported conclusion was that commitment helps mediate between identity and role performance [28].

Individuals who belong to the same social category often adopt a common set of beliefs. Stets has shown that individuals who hold membership to a group will have greater commitment to the group and will not want to leave the group, regardless of the group's status. Through the processes of depersonalization and self-verification, members within a group will behave similarly [28]. Depersonalization is a social process in which one sees the self as an embodiment of the in-group's norms [29]. Depersonalization accounts for group cohesiveness, emotional contagion (e.g., the spread of aggression from one actor to another) and collective action [22]. This process is realized in the simulation through a specific equation that accounts for the group's cohesiveness. When an identity becomes activated, self-verification occurs. If this happens, one will see the self as an embodiment of one's perception of a role which contains the in-group's meanings and norms [30, 31]. As a result, individuals within a group will behave similarly. The simulation uses this idea by having all individuals within a group move in a similar direction. The discussion section explores ways in which group behavior can be incorporated even more in future developments of the simulation, through having all other individuals in a group change aggression level when a single individual in a group changes aggression level.

Individuals, groups of friends, as well the student body as a collective, have different intentions before participating in an event. The gathering of the crowd may raise expectations and focus attention on a set of goals, creating a hierarchy known as a "goal-gradient" [19]. For example, some celebratory intentions (e.g., chanting, cheering, singing) during Spring Weekend and the NCAA Championship resulted in pounding upon, rocking and overturning cars, breaking bottles and windows, and setting trash cans on fire. Although people's motives (whether at the individual or group level) vary widely in celebratory gatherings, motives supporting aggressive behavior can be conceptualized in terms of the one's preference for violent, aggressive behavior. *Action choices* refer to the actors' varying preferences for a behavior at a given time, such as cheering, setting a fire, rioting, etc. This general aggressiveness index is labeled the action choice of an actor. It is assumed that each actor's behavior in a situation depends on her/his own action choices and those of actors with which they congregate. An actor's behavior in a situation depends on each actor's own action choices and those of actors with which they congregate [4].

Berk describes how there exist different categories of motivated people. There are the militants who are likely to become violent and there exist the moderates who are not. If the crowd members are primarily moderates, then the crowd is not likely to turn into a riot. If the crowd is largely militant, then a participant could possibly riot while anticipating group support. At the beginning of the simulation, the user can specify the aggressiveness for each identity group in the crowd, thus controlling the number of militants. Different groups are likely to react in a manner such that they maximize fulfillment of their interests. What is important about this is that people's decision making process is dynamic— it depends on many things so that everyone's interests are met (ideally). Outcomes depend on the actions of other actors. Before one makes a decision, he or she weighs different factors. These different factors will be the inputs of the algorithm that each actor will use in order to make a decision. The probability that an individual will act is a function of the amount of support and the pay offs between acting and not acting [19]. Although this slightly enters the realm of game theory, the simulation does not compare the differences between acting and not acting. Rather, it calculates the amount of support that it has and whether this is enough to act in a certain manner.

Gould offers the idea that group violence occurs because of tension between what a group wants and what an individual wants. Individuals in a dispute call upon their allies to show that they are not alone. In the simulation, an agent's change in behavior will depend on how much support it has. Nondisputants (bystanders) tend to become less involved if the disputants involved are strangers rather than kin. It is the fragility of solidarity not its strength that leads to the intensification of conflict [17]. Chwe suggests that theories of collective action should take into account structure *and* individual strategy together rather than separately [16]. This raises another interesting area of expansion for the program. The physical area in which the agents roam can be weighted so that the leaders mill toward the center, and the bystanders toward the periphery. As a result the leaders will interact more and the bystanders less.

As another constraint for this simulation, the radical leaders (i.e., those with extremely high action choices) will be initially dispersed throughout the crowd in a random pattern, mimicking the propagation of aggressive behavior in a celebratory riot similar to Spring Weekend or the NCAA Championships. This initial condition represents a departure from previous simulations by mathematical sociologists. For example, William Feinberg's model uses concentric circles to represent the crowd with "a core of activists at the center, a set of supporters beyond the core, and a ring of spectators or observers at the fringe" [4]. Feinberg's model is more appropriate for a highly structured protest riot, while my model will begin with an even spread of radical leaders, which is more representative of a celebratory riot.

The assembly phase, with respect to the individual, is relatively simple. Actors cluster with other individuals which can relate with each other very well. On a group level, the assembly phase is the outcome of a complex combination of individual and group processes. There is a sequence of interpersonal exchanges whereby people learn of an event transpiring and can exchange information. The repetition of this action leads to group formation. This period brings individuals together and primes them for participation. We must account for both individual predispositions and interactions in order to accurately simulate collective behavior [18].

Principles of Group Formation

Actors within a gathering interact with one another and thereby create small group structures [5]. These social structures result from individual and collective pedestrian movements (i.e., milling) and the emergence of clusters of individuals within a gathering (i.e., congregating), and they have direct consequences for dynamics within a gathering [3]. The parameters of movement affect the propagation of action choices because an actor can only affect other actors with whom it interacts. Once the actor interacts with others, it becomes susceptible to change in its action choice, as do the others. In addition to movement parameters, the structure of social ties is relevant since the actions of others to which an actor is tied shape its behavior directly. People are assumed to react to the actions of their neighbors in the network. In Chwe's network, an actor learns about its neighbors' preferences and their willingness to revolt; actors must be able to communicate, however, in order for such emotions to propagate. Each person's choices depends on the relationship with other neighbors [16]. In this simulation, actors know their neighbor's identity and current action choice.

Effects of Clustering: Social Identity and Influence, Rebels versus Sheep

Through spatial clustering, group-level phenomena begin to appear. Consolidation identifies the situation in which diversity within a group is reduced as the proportion of people who hold the minority position decreases. Consolidation occurs whenever minorities are more exposed than majorities to adverse social influence [14]. A crowd consisting of a large proportion of *sheep* (i.e., agents who are likely to change behavior to be more similar to nearby actors) will quickly destroy any minority position. At the onset of the simulation's execution, each agent is assigned a "susceptibility to influence" parameter indicating whether it is a sheep or a rebel (i.e., an agent who either does not change behavior based on the behavior of nearby actors or who changes behavior to be less similar to nearby actors). Controlling the number of sheep and rebels likely changes the vitality of the minority position. Continuing diversity describes how a minority opinion can survive due to a perceived local majority. It occurs when a minority cluster tends to conform to an illusory local majority rather than to the group majority [12].Continuing diversity results when individuals in minority clusters are able to resist adverse influence. Clustering leads to continuing diversity because clusters protect minorities from majority influence even when they are eager to succumb to the majority. Individuals are typically stubborn, weighing their own opinions highly even when faced with unanimous opposition. All of these group behaviors are dependent on the geometry of social space [14]. A discussion of how the simulation can take on different social geometries can be found later, but for now the simulation takes on a free space in which each actor has an equal probability of any location in the space.

Just as identity affects how easily one can relate to another person, physical proximity is also a limiting factor. The number of people located at any given physical distance from an individual should increase in approximate proportion to that distance. Thus if we were all equally influential, then people would be more affected by distant strangers than by their neighbors, since there are more people further away. Fortunately individuals respond to sources in close proximity rather than the entire physical space. This idea is the basis for a concept of social space, which is a product of social influence, but also constrained by the physical dimensions of space and time [11].

The greater the distance between people, the less influence one will have over the other; there is an inverse relationship between distance and influence. Therefore social influence is very much a local phenomenon. The amount of influence that one has on another also depends on the strength and number of sources. The greater the number of sources within a close physical proximity, the more influence they have over the target. We are most affected by the people closest to us, even if we do not know them. These strangers provide a basis for comparison, which is sometimes more effective than conversation. Distances within a social space can be measured by the degree to which social positions (age, sex, race, status, religion or occupation) are shared. People tend to associate with others of similar status. Therefore spatial relationships are a function of both physical distance and status [13].

The number of sources also affects how much an individual will change. It has been shown that impact grows in proportion to the square root of the number of influences [11]. As seen in *Figure 2* of the appendix, the slope of the square root function decreases with increasing sources of influence. The addition of another source at a high group size will not make as much impact as the addition of an additional source at a low group size. Since the simulation rarely has cluster sizes above 10, a logarithmic relationship was used instead, which has essentially the same properties as a square root function, but is more exaggerated at lower levels. It was also found that group members became more similar to their neighbors, but minorities did not completely disappear. The group members who interacted the most were most likely to change. But not everyone was perfectly rational; people switched positions when receiving very few opposing messages, and some people were stubborn and did not change when there was a majority telling them to do so [12].

People perceive regularities, mimic them, and social norms develop, are legitimated, and take on a life of their own [32]. These norms reinforce social representations and promote self-organization. Dynamic social impact theory states that the more different the individuals, the more likely that social representations will emerge. In other words, spatial clustering and similarities among groupings increase with increasing randomness [10]. As a result of clustering, attributes possessed by individuals within a cluster become correlated. Individuals influence each other with social impacts which correspond to the sum of the persuasive impact (the total force to change, coming from individuals with opposing opinions) and a supportive impact (total force to change, coming from individuals with the same position). The methods section makes specific and clear how the simulation uses this summation. The amount of social impact on an individual is put on a continuum that takes into account the influence of its neighbors. When neighbors are very influential, they have a very high persuasive impact, and vice versa. Incremental influence processes lead to convergence; nonlinear influence processes lead to continuing diversity. This simulation allows for continuing diversity through assigning each individual a property whether they will assimilate or will dissimilate. Actors who have been assigned rebellious characteristic will provide for continuing diversity. These two impacts are a function of strength and distance of each persuader or supporter. An individual will change only if the net persuasive influence is greater than the net supportive influence plus the individual bias. High strength individuals tend to anchor the borders of minorities, rather than to continually change and eventually unify. Despite strong pressures to unity, diversity can continue to exist (continuing diversity).

Clustering has positive effects on the society as a whole. Clustering provides for continuing diversity because the clusters protect each other from external influences. Inside a cluster, members experience agreement and validation for attitudes and behaviors that may face condemnation or other negative sanctions outside that cluster. On the boarder of a cluster, people are exposed to opposition and are able to change positions. In one-dimensional geometry, the minorities are no more exposed than the majority and so are no more likely to decrease in number [13]. Most of the qualities that people use to characterize each other (representations) are communicated verbally or visually (and often subconsciously) through physical cues [11]. The simulation does not take into account a line of sight however, and assumes that actors influence each other when they are within a certain distance.

Conditions Sufficient for a Riot

Collective violence involves social contagion, where individuals are instigated/inhibited by the information that they receive [21]. The fact that we are most influenced by the people closest to us allows for organization of our social structure. But the fact is that individuals differ (one assumption of the dynamic social impact theory). Therefore, there exists a spectrum of influence among the general population, from not very influential to very influential. Some individual factors include physical size, intellect, wealth, social status, or belonging to the same group [11]. The simulation cannot, and should not, take into account all of these characteristics, but each actor does take into account the amount of influence that a neighbor has. This depends on how well the agent can relate to the neighbor, and how large the neighbor's cluster is.

The occurrence of suggestions (e.g., "Hey guys, I think we should…") varies according to the extremity of the actions involved. Reaction to these suggestions can be analyzed using a modified version of Berk's game theory. He says that the ultimate crowd consensus is based on rational deliberation of potential rewards and costs to individuals. The more actors that are involved in a situation, the more likely it is for a non-participating actor to join, since it has less cost to this individual. This is why the simulation takes into account the size of the neighbor's crowd in order to determine how influential it is. All members of the crowd seek support, and those who propose a course of action are those who believe that they will receive support. Perception of support will determine whether the individual will attempt to influence the crowd. Influence can be viewed as a function of a group factor (the crowd distribution) and individual factors (the intensity with which individuals hold their opinions). The crowd does not always follow the suggestion of an actor, it first assesses his or her suggestion and then reacts to it [20]. In this way, actors can either become more similar or less similar to its neighbors. This is why the simulation assigns to each actor a characteristic of being rebellious or not.

Understanding the dynamics of collective behavior is prerequisite to understanding the propagation of aggressive behavior [7]. The goal of the simulation is not to predict the behavior of a single person, but to understand the collective outcome of a riot and how aggressive behavior can propagate through patterns of interactions amongst the crowd. The iterative, recursive outcome of individual influence processes will lead to the global self-organization of socially influenced attributes and the emergence of group-level phenomena [11]. When individuals cannot accurately assess the most popular position of the whole group, then emergent group-level behavior such as clustering and incomplete consolidation appears [14].

Methods

A computer simulation was chosen for a variety of reasons, and there are many benefits to agent-based models for the investigation of large-scale collective behavior. Foremost I can explicitly control how the actors evaluate each other. A computer makes it possible to evaluate nearly one hundred actors, two thousand times each, in a matter of minutes. The JAVA language provides for helpful graphical output, a simple user interface, easy expandability, and customized output according to the user's or the programmer's desires.

Utility and Appropriateness of Computer Simulations for Studying Collective Behavior

Computer simulations are helpful because they can take a single algorithm and evaluate it with different initial conditions and display an emergent property. Simulations can evaluate exactly what the author deems is important, and then run the same routine on each agent. In other words, each agent can use the same algorithm but with different values for the parameters for different others and different points in time. The fact that actors' choices are dependent on previously made choices also is ideal for a computer simulation; the choice made at time t depends on the conditions at (t-1). Properties of the crowd emerge that could not be predicted from the algorithm itself.

The simulation uses the principles of agent-based modeling, and in this instance the "agents" in the simulation represent participants at an event (i.e., the same socially defined place and time). In accordance to Gilbert and Troitzsch's suggestions, the agents have the typical properties: autonomy, social ability, reactivity, proactivity. Each agent can make its own decisions (autonomy) by communicating with others (social ability). When group clusters are made, they are able to perceive various qualities about their environment (reactivity) and can take the initiative to start another group if so desired (proactivity) [23]. The outcome of each agent depends on the user-defined initial conditions, its environment and its own “proclivities”. The computer simulation recreates the basic processes of each actor in order to (1) demonstrate how they combine to generate the complex phenomenon of a riot, (2) understand the effects of individual characteristics (e.g., action choices) on these processes and the action consensus of the crowd (e.g., dispersion or riot), and (3) provide an analytic tool others may use to investigate, visualize and predict the effects of varying characteristics of individuals and groups on the action consensus of the crowd [4,8].

Graphical User Interface

The simulation has been written entirely in JAVA and therefore can be run on any Windows or Macintosh operating system. Running the simulation reveals a window such as that in *Figure 2* of the Appendix. The simulation consists of three main parts. The large window with the circles is where each actor is represented visually. To the bottom left there is an information pane where information about each actor appears when a circle (actor) is clicked. To the bottom right is where the initial conditions are set and the simulation can be stopped, started or restarted.

In the large window, each circle represents one actor. The circles have different characteristics that help visualize their properties. The size of the circle represents their action choice, or how aggressive that particular actor is. The larger the circle, the more aggressive the actor is. Likewise, the smaller the circle, the less aggressive. The aggression levels are dynamic and change throughout the simulation. The color of the circle represents the actor’s identity, which is a constant set once the simulation starts. Actors with the same shade of blue represent actors with the same social identity.

Each time the program is opened, every parameter is set to the default values. The parameters, which are set in the initial condition pane, can be changed by the user. The number, *n*, represents how many actors will be involved in the simulation. Each actor is named from 0 to *n*-1 and organized into an array. Therefore when a circle is clicked, text in the following form appears:

Actor: *x* Agg: *y* Id *z*. Cluster: -1 Population 0.

If an actor is not part of a cluster, its cluster name is represented by the value *-1*. Of course when the actor is not in a cluster, the population of its current cluster is zero since the actor does not belong to any cluster. Once an actor joins a group, not only will the border of the circle change to a different color, but when you click on a circle within that group, its cluster identity and population will no longer read *-1* and *0*, respectively.

As mentioned, the number of actors within the simulation can be changed. The values range from 10 to 130 in increments of 10. The default value is set at 60 so that the average processor can smoothly run the simulation. Additionally, the aggressiveness of each identity can be changed. Since there are 5 different categories of identities in the simulation, five sets of numbers appear for the mean aggression level, and five sets of numbers for the standard deviation. The mean aggression level sets the average aggression level for the group of actors which has that particular identity. For example if the first spinner is set at 10, actors with an identity equal to zero (since values range from 0 to 4) will, on average, have an aggression level of 10. The standard deviation spinner that is directly below the mean level spinner corresponds to the same identity group. The greater the standard deviation, the more likely that an actor will have an aggression level further from the group's mean. For example if the standard deviation is 1, then it is very likely that every actor in that identity will have an aggression level close to the mean aggression level (which ranges from 1 to 100).

The three sliders below provide more variables that the user can set as initial conditions. Setting the interpersonal attraction level to high makes it more likely that when two actors meet, one will either join the other's cluster, or the two will want to start a new cluster together. The two actors will join each other but then quickly disband if they do not have any group cohesiveness. The amount of cohesiveness is set in the next slider, which is appropriately labeled. If this slider is set to *high*, then actors are more likely to stay in a cluster once they join.

The last slider determines the percentage of actors within the simulation that have a disposition to assimilate (while concurrently defining the percentage that dissimilates as the complement of assimilators). An actor who is defined as an assimilator will always become more similar to a neighbor with whom it is interacting. Conversely, an actor who rebels will change its action choice to be more different than its neighbor. These characteristics are set at the beginning of the simulation and do not change; one who starts off a rebel will end a rebel. These initial conditions each affect, but do not *directly* or *completely* determine the outcome of the simulation.

When two actors interact, the program runs a very important algorithm which decides how each actor will change. The algorithm says nothing of how the entire crowd should act but rather defines the changes for the one actor in question. Each actor uses the same algorithm, although each has different initial values of the variables in the algorithm based on the initial conditions established at the outset and encounters different neighbors with different values of these same variables. As the simulation runs, this algorithm determines how much each actor's action choice changes, and whether or not they join a cluster. The simulation goes through each actor iteratively, from actor 0 to actor *n*-1. When each actor's *evaluate* method is invoked, the actor has the potential to change the neighbor's action choice as well, and it can tell the other actor to join a cluster. In other words the actors can communicate with each other. This satisfies a key property of agent based modeling.

The first step in the *evaluate* method is to determine how much the actor, actor *A*, wants to be in its own cluster. This consists of lines 12 – 25 in *Code Fragment 1* in the appendix. The amount that an actor wants to remain in its group is quantified using a point system. The higher an actor's cohesion points, the greater the actor's allegiance to the group. If actor *A* is not part of a cluster, then the actor will receive no points for cohesiveness with its own group. The closer the actor's identity is to the mean identity level of the cluster, the more the actor is similar to the group, and the more the actor wants to stay in the group. The following equation formally represents the relationship between identity and cohesion felt towards one's current in-group:

$$IDCohesion_T^{A/GroupA} = IDCohesion_{T-1}^{A/GroupA} + 10 - \left| avgID_{T-1}^{GroupA} - ID_{T-1}^{A} \right| * 2.5 .$$

Where $IDCohesion_T^{A/GroupA}$ represents the cohesiveness (due to identity) of actor *A* with *Group A* (its own group) at time *T*. The absolute value of the difference between group *A*'s average identity and *A*'s average identity is represented by $\left| avgID_{T-1}^{GroupA} - ID_{T-1}^{A} \right|$. Since this difference can range from 0 to 4, the number of points ranges from 0 to 10. When actor A has the exact same identity as the mean identity of the group, it will be assigned 10 cohesion points, while an actor who is 4 identity categories away would obtain 0 points. Additionally, the greater the size of the group, the more difficult it is for the actor to leave the group. The amount of points that are assigned depends on the logarithm of the size of the group:

$$SizeCohesion_{T}^{A/GroupA} = SizeCohesion_{T-1}^{A/GroupA} + \log(population_{T-1}^{GroupA}) * 10 .$$

The variable $SizeCohesion_{T}^{A/GroupA}$ represents the cohesiveness (due to cluster size) of actor *A* with *Group A* (its own group) at time *T*. The variable $population_{T-1}^{GroupA}$ represents the population of *A*'s cluster at time *T*-1. This would be equal to zero if *A* is not in a cluster, and equal to two if *A* has one other actor in its cluster, and so on. The logarithm of the cluster population is multiplied by a scaling factor of 10 so that the value has more weight. The logarithm in this equation is to the base 2, so that a cluster size of 10 will receive 10 points. The logarithm function was used since the addition of an actor in *A*'s cluster will have a greater effect if the size of actor *A*'s cluster is small than if it is large [11]. This is also described in the theory section. The total cohesion is given by the following equation:

$$Cohesion_{T}^{A/GroupA} = IDCohesion_{T}^{A/GroupA} + SizeCohesion_{T}^{A/GroupA} .$$

The total cohesiveness of actor *A* with *Group A* is a function of the cohesiveness due to identity and the cohesiveness due to group size. The actor now has a sense of its loyalty to its group.

It should be noted that the evaluate method is called whenever the simulation moves an actor and there is another actor, actor *B*, within a certain radius. The radius dimension is a set value and is defined by the number of pixels from the upper left corner of one circle's bounding rectangle to the upper left corner of another circle's bounding rectangle. JAVA represents each actor as a single point; the size of the circle is for visualization purposes only. Therefore the larger circles are not any more likely to interact with more people just because they are larger. Since the *evaluate* method is being called according to physical proximity, most of the time the evaluation method is called with actors who are in the same cluster because actors who are in the same cluster also move in the same direction. In order to prevent actors from changing their action choice incessantly, an actor can only change its action choice when it encounters another of either (1) a different cluster or (2) of no cluster at all. This is consistent with sociological research on social networks that demonstrates that while local groups have the greatest amount of influence on an actor, distal groups are more capable of providing novel information and therefore producing changes in the local group [33].

The next statements in the evaluate method checks to see whether the two actors are in the same cluster (see lines 29 – 31, *Code Fragment 1*). If the actors are in the same cluster and the amount of cohesiveness is less than the amount specified by the slider on the application window, the actor will leave the cluster (line 34, *Code Fragment 1*). The program gives the actor enough time to leave the cluster without it being evaluated again in the immediate successive iteration (line 33, *Code Fragment 1*).

In order to find out whether actor *A* should leave its group and join actor *B*'s group, or vice versa, it is necessary to find out how much loyalty actor *B* has toward its own group. Therefore the program calculates actor *B*'s cohesiveness just as it evaluated actor *A*'s (see lines 38 – 49 *Code Fragment 1*). Actor *A* cannot tell actor *B* that it does not belong in its group. However if the two are attracted to each other more than actor *B* wants to be in its group, actor *B* will leave its group and join actor *A.*

As already mentioned, if and only if the two actors are in different groups, can they change their action choices. The program then figures out how much the two actors are attracted to each other (lines 54 – 56 *Code Fragment 1*). This is done by using the points system similar to how the group cohesiveness was calculated. If the two actors are very similar to each other, they are more likely to join together. The difference in aggression level is calculated by the following:

$$Attraction_{T}^{A/B} = Attraction_{T-1}^{A/B} + 10 - \frac{\left| AvgAgg_{T-1}^{GroupA} - AvgAgg_{T-1}^{GroupB} \right|}{10}.$$

Since the difference in aggression levels range from 0 to 100, the absolute value is divided by 10 in order to be scaled to a points range of 0 to 10. Aggression levels that are identical will receive 10 points, and those that are completely different will receive no points. The amount of joining points takes into account difference in identity in a similar manner:

$$Attraction_{T}^{A/B} = Attraction_{T-1}^{A/B} + 10 - \left| AvgID_{T-1}^{GroupA} - AvgID_{T-1}^{GroupB} \right| * 2.5.$$

Since the difference in identities range from 0 to 4, the value of $\left| AvgID_{T-1}^{GroupA} - AvgID_{T-1}^{GroupB} \right|$ also ranges from 0 to 4. The equation is scaled such that $Attraction_{T}^{A/B}$ can range from 0 to 10. An actor that has the exact same identity as the mean identity of the group would be assigned 10 points, while an actor who is 4 identity categories away would obtain 0 points.

As actors encounter others, four outcomes are possible. If neither actor is a part of a cluster, the two actors can (1) join each other or (2) both walk away from one another. If one or more actors is in a cluster, the other actor can either (3) join that actor’s cluster or (4) both can leave their respective clusters and form a new cluster. All of these scenarios are included in *Code Fragment 1*, and there are comments explaining exactly what code handles what situation. As in situation 4, if both actors can leave their respective groups (i.e., their cohesiveness is greater than the amount specified on the simulation’s control panel) and both actors want to join each other (i.e., their attraction to each other is greater than the amount specified on the simulation’s control panel) then both actors will leave their clusters and form a new cluster. This situation is coded in lines 58 to 60 of *Code Fragment 1*. If situation 4 is not possible but actor *A* is able to leave its cluster (note, as previously stated, if an actor is not in a cluster $Cohesion_T^{A/GroupA}$ = 0) and its $Attraction_T^{A/B}$ is greater than the amount set on the application's control panel, then actor *A* will join actor *B*'s cluster. This situation is coded in lines 62 to 64 of *Code Fragment 1*. The opposite situation in which actor *B* will join actor *A*'s cluster is described in lines 66 to 68 of *Code Fragment 1*. All of the aforementioned situations exhaust the possibilities of what could happen between the two actors in regards to their clusters. However there is one more dynamic aspect to each agent: their action choices.

Actor *A*'s new aggression is equal to its old aggression plus a change in aggression. Each actor's change in aggression is a function of the influence of the other actor's cluster. The change in aggression is dependent first on whether or not the actor was assigned the status of rebel or not. If the actor is an assimilator, then its aggression level will *always* change toward the level of its neighbor with whom it is interacting. For example, if actor *A* is an assimilator with an aggression level of 20 (out of 100) and it encounters actor *B* with an aggression level of 60, then actor *A*'s aggression level can range between 20 and 60. If actor *A* had been a rebel, then its potential for change would be between 0 and 20. The following equation describes how this works, and its implementation appears from lines 71 to 74 of *Code Fragment 1*:

$$Agg_T^A = Agg_{T-1}^A + direction * \left(AvgAgg_{T-1}^{GroupB} - Agg_{T-1}^A\right) * \frac{\left(Cohesion_{T-1}^{B/GroupB} + 2\right)}{30}.$$

The variable *direction* refers to whether the actor is an assimilator or rebel. This is assigned as soon as the agent is constructed, and is equal to 1 if the agent is an assimilator and -1 if the agent is a rebel. How much actor *A* changes depends on the amount of influence that actor *B* and its cluster has (if it has one). The variable $Cohesion_T^{B/GroupB}$ was defined in the same way that $Cohesion_T^{A/GroupA}$ was defined. It ranges from 0 to 30 (on an arbitrary scale) which is why the value is divided by 30. If the cluster is extremely influential, the actor will change completely to what the neighbor's cluster's average aggression level is. An arbitrary value of 2 was added in case $Cohesion_T^{B/GroupB} = 0$, so there will be some change in the actor's aggression level when no clusters have formed yet. The neighbors change in aggression is also computed similarly, and appears in lines 77 to 81 of *Code Fragment 1*:

$$Agg_T^B = Agg_{T-1}^B + direction * \left(AvgAgg_{T-1}^{GroupA} - Agg_{T-1}^B\right) * \frac{\left(Cohesion_{T-1}^{A/GroupA} + 2\right)}{30}.$$

Actor *B*'s change in aggression is a function of how influential actor *A*'s group is, if it is in one. Actor *A* handles this computation and then tells the actor *B* how much it should change.

Meanwhile, during each iteration, the application is also checking to determine whether a riot has occurred or not. A riot is defined by any point in the simulation when there is a cluster which has a population of 5 or more with an average aggression level of 90. The program will abort and alert the user, in the messages box, that a riot has occurred. If either this occurs, or there have been 2,000 iterations, the program will stop running and the output data will be written to file. This number of iterations, although somewhat arbitrary, was chosen because on average, if no riot has occurred by this point, none will occur at all. These two points can be thought of as a steady state equilibrium representing the consensus of the crowd. The time to consensus and whether consensus is reached will depend on the distribution of aggression and the initial conditions that were chosen for the simulation. The output file records the values of all the initial conditions, the number of iterations that have occurred, and each cluster and its average identity, average aggression and standard deviation of its identity and aggression as well as the population of the cluster and who is in it. Other files that are created record the actors' cluster name and identity at each iteration.

Once the program is over, the program can then be run exactly as the first time, however there is a random component in the location of each agent and also the direction in which they move. The data files can be examined easily for analysis.

Discussion

It is important to remember that not every individual in the riot area engages in violence, and those who do, do not do so continuously. Instead, small groups carry out the majority of violent acts; riots are not a gathering of widespread and all inclusive violence [7]. The probability that each small group will commit a violent act depends on the composition of that group. As a result, the action consensus of the crowd will not represent unanimous riot behavior by all actors, but rather riot behavior by small groups within the crowd. The simulation shows signs of emergent behavior, since crowd behavior develops from an algorithm defined for just each actor. The agents are designed so that they know how to move, get information about their neighbors, and make decisions regarding their action choice. Once the actor is in a cluster, their properties depend on the cluster's size, movement, aggression level and average identity. The simulation eventually reaches an equilibrium at which the outcome will no longer change. The outcome, whether it riots or not, depends heavily upon the initial conditions.

Basic findings have been discovered after some preliminary trials. By manipulating the initial conditions on the interface of the simulation, it is possible to make some estimates of the sufficient conditions for a riot. The proportion if rebels in the simulation must be sufficiently high. If there are no rebels but all sheep, all of the actors will converge onto one average aggression level. On the contrary, if all actors are rebels, then the peaceful will end up getting more peaceful, and the aggressors will get more aggressive. Eventually the entire group will polarize and a riot will be inevitable. A figure depicting how this the proportion of rebels changes the likelihood of a riot is in *Figure 5* of the Appendix. It was found that the group cohesiveness needs to be at an optimal value in order for a riot to occur. If the cohesiveness is too low, the actors will join but will quickly disband and therefore will never reach first condition of a riot (that the cluster's population is five or more). If the cohesiveness is too high, then large groups will form and there won't be enough neighbors to influence the large group. This relationship can be found in *Figure 6* of the Appendix. The interpersonal attraction seems to follow a square root relationship such that once the interpersonal attraction level is high, then increasing the value will barely increase the chances of a riot. This makes sense because when the interpersonal attraction is low, groups will not form, so the first condition of a riot will not be met. But if the interpersonal attraction level is higher, then the actors are more likely to cluster, and therefore the likelihood of a riot increases. The results were logical and coincided with current literature on the conditions sufficient for a riot, however the methods I used were unique. Most simulations of riots do not focus specifically on a celebratory riot, and this is one reason why what I use varies slightly from past simulations.

My simulation differs slightly from current theory in the following ways. All of the agents in this simulation move completely randomly, but in actuality this is not the case. Current theory suggests that certain types of actors interact more than others. In the simulation however, all actors have equal chances of interacting with others. The simulation could be changed so that certain actors have a greater tendency to remain around the center of the simulation, while other actors revolve around the periphery.

The simulation additionally does not take into account crowd density. The number of actors within an area may change actors' dispositions and likelihood of change. The size of the simulation window was chosen arbitrarily, but the density can be controlled by changing the number of actors. A density factor can be added fairly easily since there are methods that calculate if there is an actor present within a certain radius. The number of actors can be divided by the area of the circle determined by that radius. There is plenty of room for expansion.

It has also been thought that the homogeneity of a cluster can affect the loyalty of its members. This can be quantified by the standard deviation of the crowd, which the simulation currently calculates for each cluster. One way in which this could be incorporated into the simulation is by affecting the amount of cohesiveness than an actor has for a particular group. In some preliminary analysis, the outcomes do not seem to be affected by density.

The simulation was coded in a way that it could be easily expanded. A lot of code has been commented out for the sake of simplicity, but can be included in the future if a programmer so chooses. The code was written in a general way where new variables can be introduced quite easily. For example, to coincide with current theory about group behavior, an entire cluster can change its action choice when a member of the cluster changes its action choice, with just a couple lines of code. Another possible area of further study could be the effects of social geometry on the chances of a riot occurring. Right now the actors move in a random fashion, but this could be changed so that certain actors mingle near the center of the space, and certain actors mill around the periphery. This could be used to study non-celebratory riots where there are leaders, such as protest riots.

Studying the process through which this equilibrium is reached, as well as the final action consensus, will provide invaluable insights into crowd dynamics and the factors leading to destructive behavior in celebratory crowds. The results may also be significant to policy makers, administrators and public safety personnel who must deal with the uncertainty and potentially dangerous outcomes of crowd behavior. Although sociological literature also addresses the consequences of gatherings for both society and individuals the purpose of the simulation is to recreate the important elements of collective crowd behavior.

Appendix

Clemson University
Colorado State University
Indiana University
Kansas State University
Michigan State University
Ohio State University
Ohio University
Pennsylvania State University
Purdue University
University of California Los Angeles
University of Colorado
University of Connecticut
University of Dayton
University of Maryland
University of Michigan
University of Minnesota
University of Oregon
University of Wisconsin
Washington State University

List 1: Universities Which Have Had At Least One Celebratory Riot Between 1998 and 2003 [25]

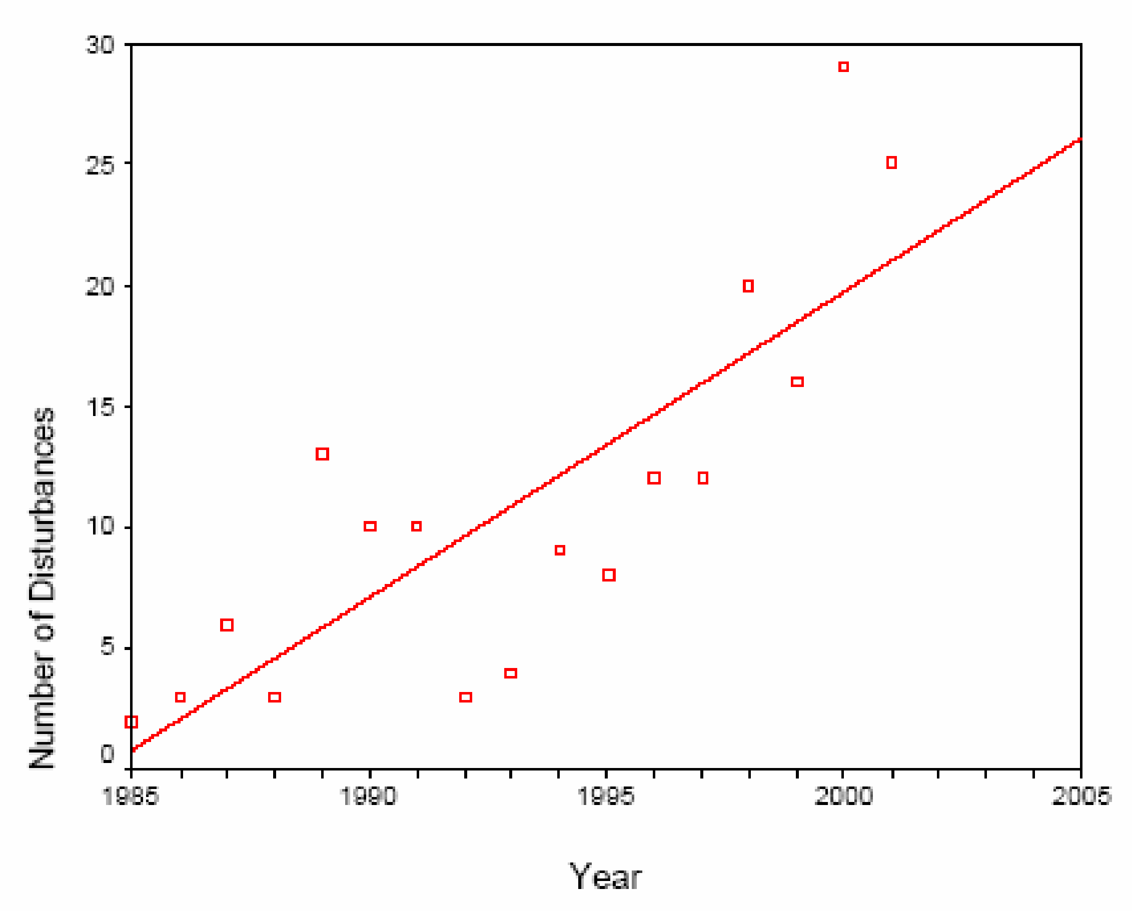

Figure 1: Frequency of U.S. Campus/Community Disturbances Not Associated with Protests, 1985 – 2002 [1].

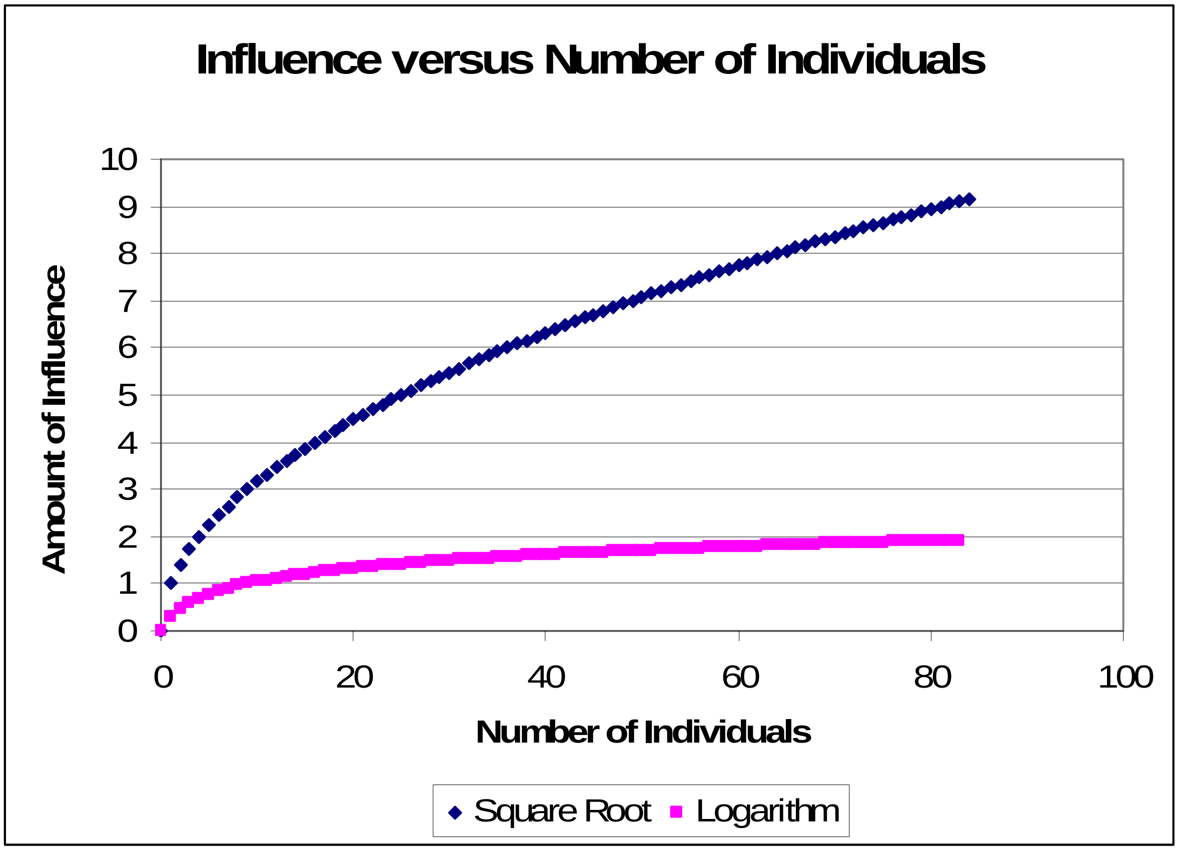

Figure 2: Influence versus Number of Individuals using a Square Root Relationship and a Logarithmic Relationship.

User Instruction Manual

1. As soon as you start the program, it automatically creates three output files called:
character_data.txt
cluster_data.txt, and
final_data.txt

*character_data.txt records the actors name and his identity and aggression level for each iteration.

*cluster_data.txt records each actors name and which cluster he was in for each iteration. A value of "-1" indicates that the actor was not in a cluster.

*final_data.txt records the final information once the Stop button is pressed and the trial is over. See (6) below.

2. If you wish to change the location of the output, click "Output Options" then "Save As" and choose the destination of your choice.

3. All of the controls at the bottom right of the panel should be considered settings for initial conditions. Although these controls can be changed during trial, this should be avoided in order to get consistent results.

4. If the aggression levels, standard deviation levels, or the number of actors is changed, then you should press the restart button in order to register this new information. Moving the sliders does not require you to press the restart button.

5. For more information about a certain actor, you may either click on the circle as the simulation is running, or you may press the stop button to obtain this information easier.

6. When the trial is done, press the stop button so that it does not continue writing data. Upon pressing the stop button, the program will calculate the following final data:

Before the simulation is started, one should change the conditions accordingly and then press the restart button, which sets these values to the program. Without pressing the restart button, the actors will continue to use the default values.

It is necessary to close it and re-execute (re-open) it to assure that all the memory has been cleared properly.

Be careful when the program is re-run because all the old data files will be erased and used for the output again unless the name is not changed. This can be done manually or can be done by clicking on *Output Options* and then *Save As*.

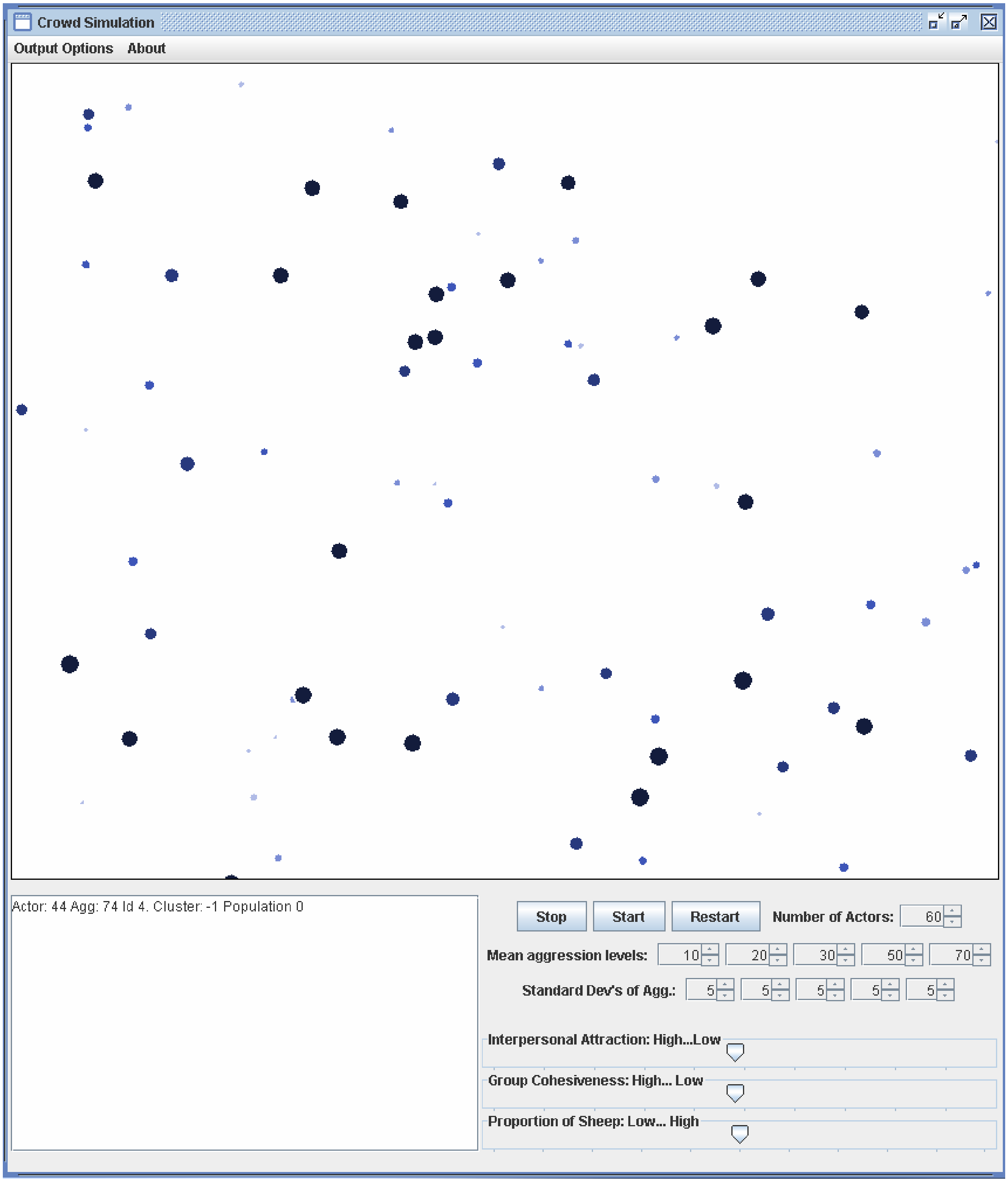

Figure 2: Crowd Simulation Program

FLOW CHART

"How to cluster…"

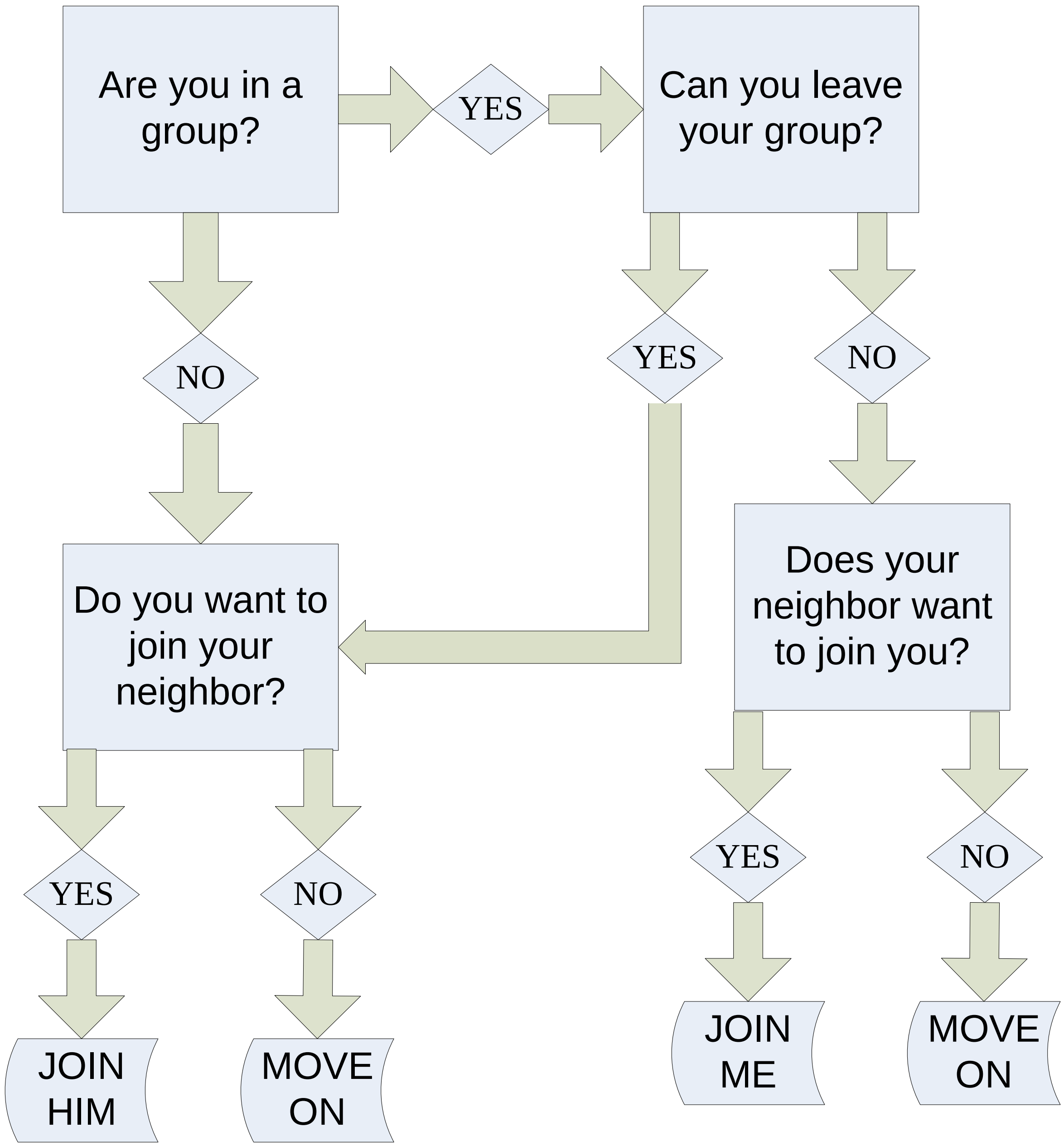

Figure 3: How An Individual Clusters

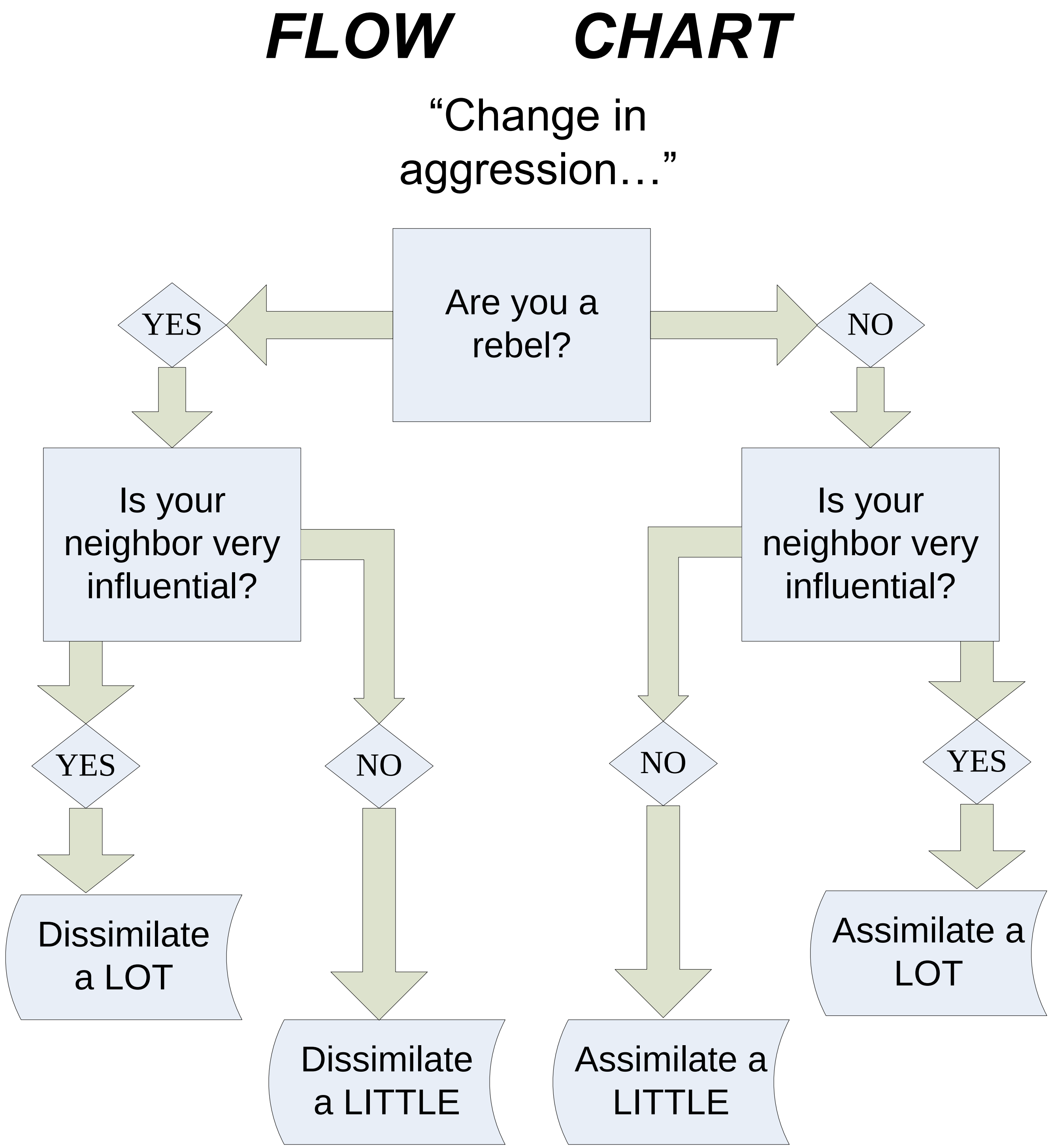

Figure 4: How An Individual Changes Aggression

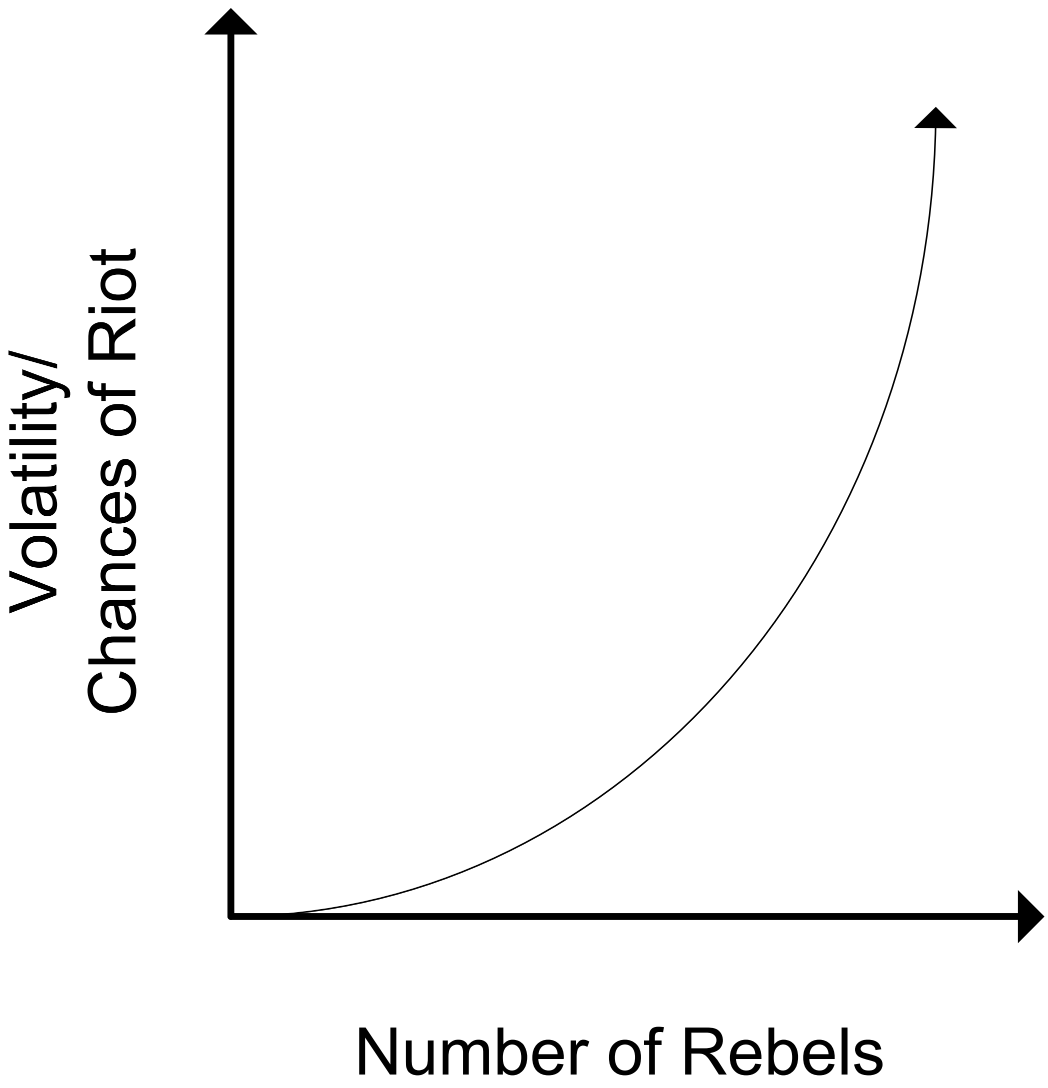

Figure 5: Chances of Riot vs. Number of Rebels

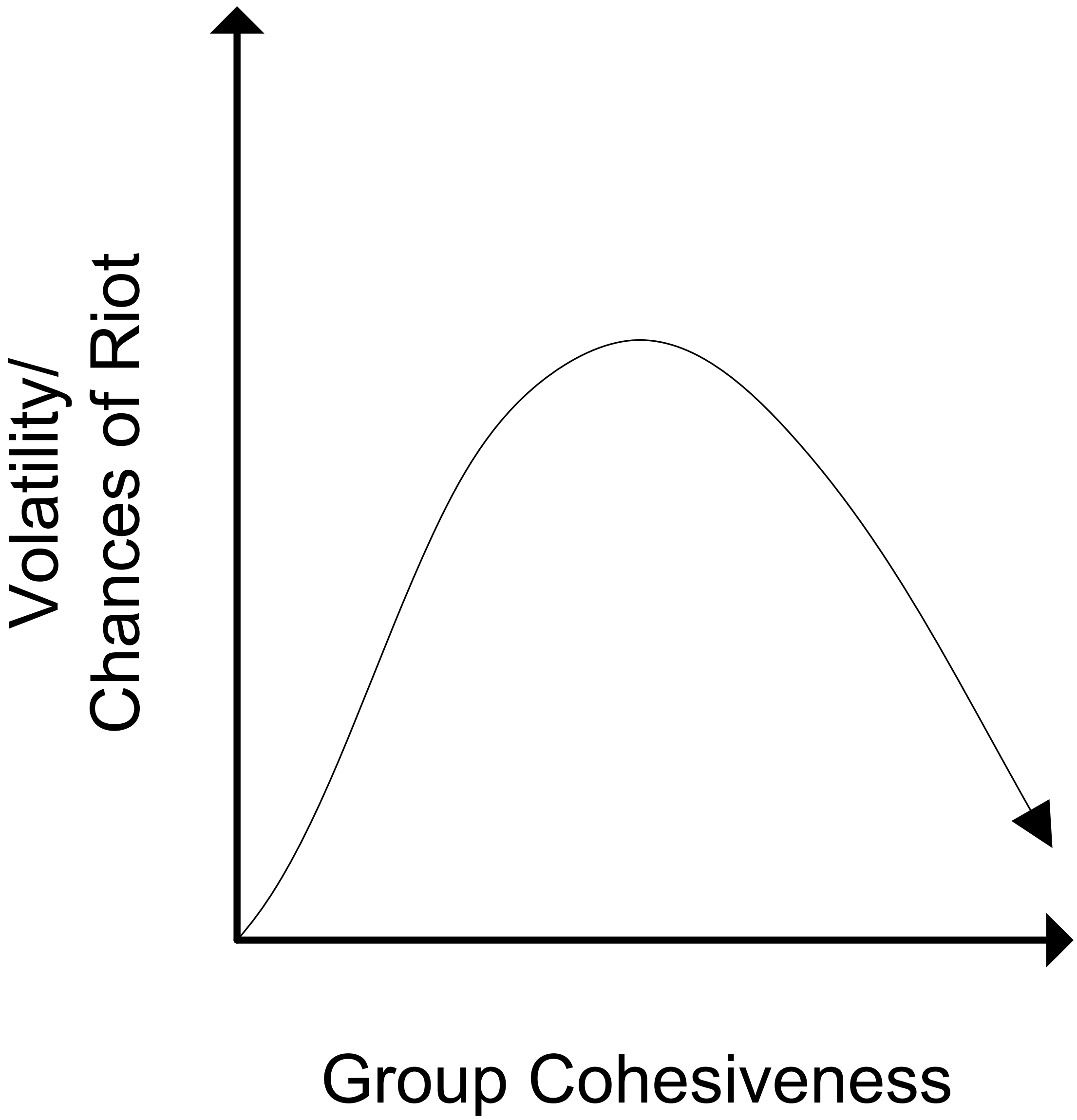

Figure 6: Chances of Riot vs. Group Cohesiveness

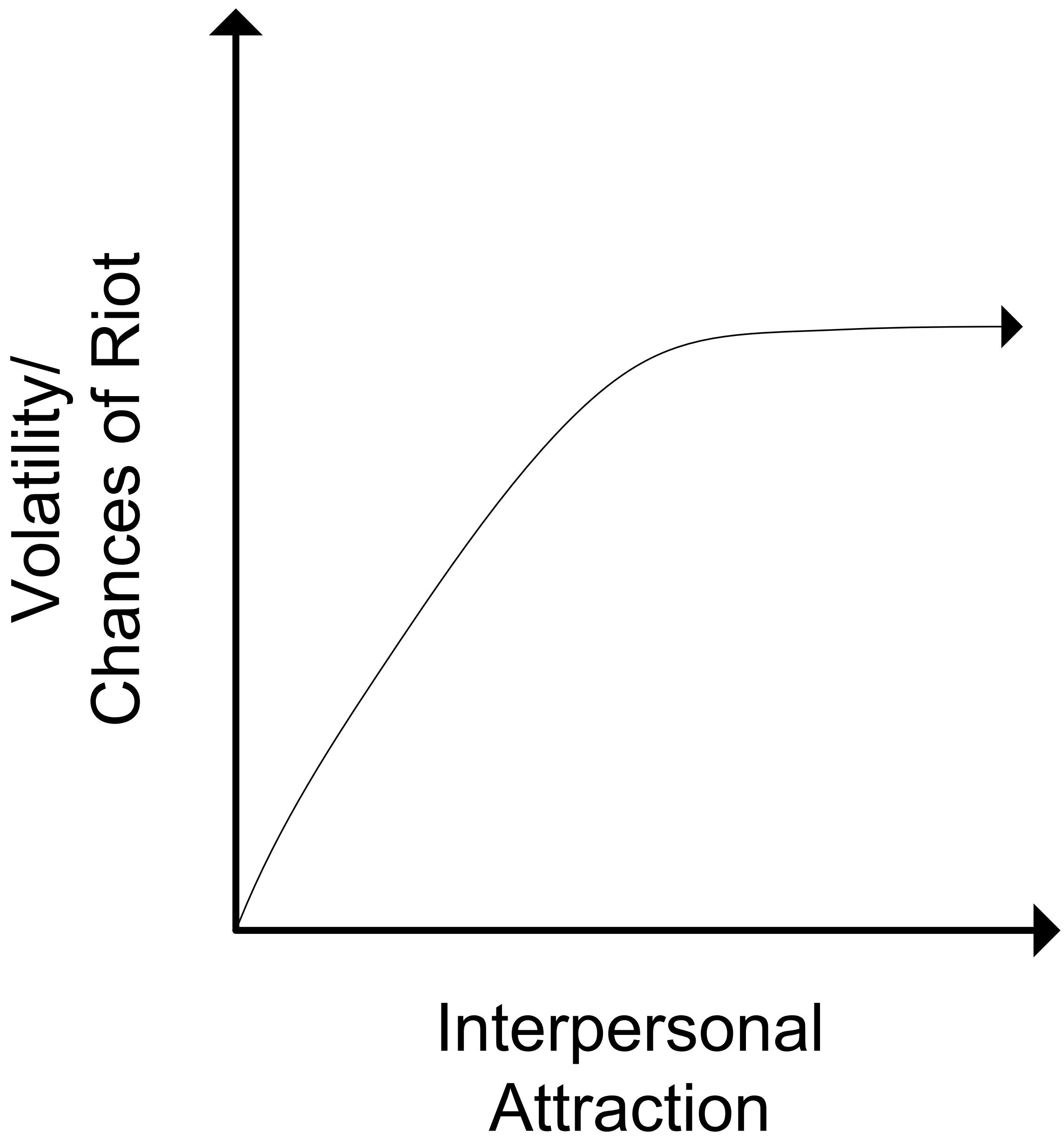

Figure 7: Chances of Riot vs. Interpersonal Attraction

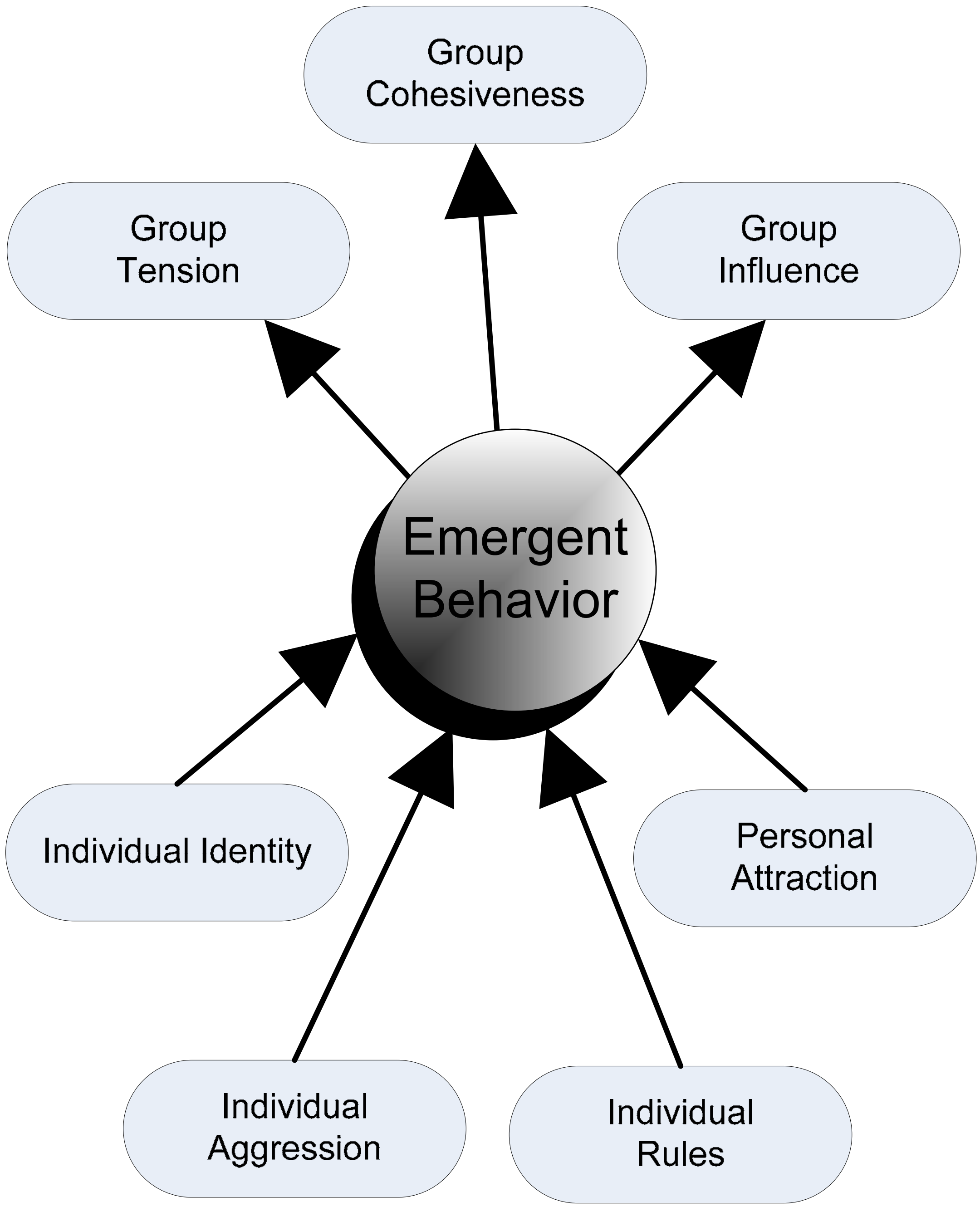

Figure 8: Basics of Emergent Behavior

Code Fragment 1: Evaluate Method

```
public void evaluate(int aNeighbor) {
int Neighbor_ClusterSize = _theMediator.ClusterSize(aNeighbor);
int Own_ClusterSize = _theMediator.ClusterSize(_actorsName);
double selfsCohesivenessWithOwnGroup, ownClusterAvgAggression, ownClusterAvgIdentity;
double neighborsCohesivenessWithHisGroup, neighborClusterAvgAggression, neighborClusterAvgIdentity;
double interpersonalAttractionPoints;
int neighborClusterName = _theMediator.getPersonsClusterName(aNeighbor);
boolean sameCluster = false;
// *********************************************************** //
// See how much the actor wants to be in its own cluster.
// Own_ClusterSize = 0 if not in a cluster
if(Own_ClusterSize > 0 && _myCluster >= 0) {
ownClusterAvgAggression = _theMediator.getAverageAggressionLevel(_myCluster);
ownClusterAvgIdentity = _theMediator.getAverageIdentity(_myCluster);
selfsCohesivenessWithOwnGroup = 0;
//Difference in ID ranges from 0 to 4.
selfsCohesivenessWithOwnGroup += 10 - abs(ownClusterAvgIdentity - _identity)*2.5;
// the larger the group, the more weight it has.
selfsCohesivenessWithOwnGroup += log(Own_ClusterSize)*10;
}

else {
selfsCohesivenessWithOwnGroup = 0;
ownClusterAvgIdentity = _identity;
ownClusterAvgAggression = _aggression;
}
// *********************************************************** //
// If they are in the same cluster, the clustering points must be over a certain number for the actor to remain with that person.
// Change direction of each person if they both leave.
if(_theMediator.sameCluster(_myCluster, aNeighbor)) {
sameCluster = true;
}
if(sameCluster && selfsCohesivenessWithOwnGroup < _theMediator.getGroupCohesionLevel()) {
this.LeaveMeAlone();
this.leaveCluster();       // Delete the actor from the cluster.
}
// *********************************************************** /
// See how much the other person wants to be in that cluster.
if(Neighbor_ClusterSize > 0 && neighborClusterName >= 0) {
neighborClusterAvgAggression = _theMediator.getAverageAggressionLevel(neighborClusterName);
neighborClusterAvgIdentity = _theMediator.getAverageIdentity(neighborClusterName);
neighborsCohesivenessWithHisGroup = 0;
neighborsCohesivenessWithHisGroup += 10 - abs(neighborClusterAvgIdentity - _identity)*2.5;
neighborsCohesivenessWithHisGroup += log(Neighbor_ClusterSize)*10;
}

else {
neighborsCohesivenessWithHisGroup = 0;
```

```
neighborClusterAvgIdentity = (double) _theMediator.getNeighborsIdentity(aNeighbor);
neighborClusterAvgAggression = (double) _theMediator.getNeighborsAggression(aNeighbor);
}

// ******************************************************** //
// Whoever wants to be in their cluster MORE will stay and the other person will join him.
if(!sameCluster) {
//Find the amount of joining points, see how much they want to be with each other.
interpersonalAttractionPoints = 0;
interpersonalAttractionPoints += 10 - (abs(ownClusterAvgAggression -
neighborClusterAvgAggression)/10);
interpersonalAttractionPoints += 10 - abs(ownClusterAvgIdentity -
neighborClusterAvgIdentity)*2.5;

// If both self and neighbor are able to leave group, make a new group.
if(selfsCohesivenessWithOwnGroup < _theMediator.getGroupCohesionLevel() &&
neighborsCohesivenessWithHisGroup < _theMediator.getGroupCohesionLevel() &&
interpersonalAttractionPoints > _theMediator.getInterpersonalAttractionLevel()) {
this.makeNewCluster(aNeighbor);
}

// If self is able to leave group, join Neighbor's Group
else if(selfsCohesivenessWithOwnGroup < _theMediator.getGroupCohesionLevel() &&
interpersonalAttractionPoints > _theMediator.getInterpersonalAttractionLevel() ) {
this.joinACluster(neighborClusterName);
}

// If neighbor is able to leave group, tell the neighbor to join this group.
else if(neighborsCohesivenessWithHisGroup < _theMediator.getGroupCohesionLevel() &&
interpersonalAttractionPoints > _theMediator.getInterpersonalAttractionLevel() ) {
_theMediator.joinMyCluster(aNeighbor, _myCluster);
}

// ******************************************************** //
// Change in Own Aggression. Function of how INFLUENTIAL the other person's cluster is.
double newAgg = _aggression + direction*(neighborClusterAvgAggression -
_aggression)*(neighborsCohesivenessWithHisGroup+5)/30;
if(newAgg > 100) newAgg = 100;
else if(newAgg<0) newAgg = 0;
this.setAggression(newAgg);
// ******************************************************** //
// Change in OTHER's Aggression. Function of how INFLUENTIAL the other person's cluster is.
newAgg = _theMediator.getNeighborsAggression(aNeighbor) +
direction*(ownClusterAvgAggression -
_theMediator.getNeighborsAggression(aNeighbor))*(selfsCohesivenessWithOwnGroup+5)/30;
if(newAgg > 100) newAgg = 100;
else if(newAgg<0) newAgg = 0;
_theMediator.setAnAggression(aNeighbor, newAgg);
}}
```

Works Cited

1. McCarthy, J. D., Martin, A. W., McPhail, C., & Cress, D. (2002, August). *Mixed-issue campus disturbances, 1985-2001: Describing the thing to be explained*. Paper presented at the annual meeting of the American Sociological Association, Chicago, IL.

2. Jeffrey MacDonald After the big game, why is there a riot going on? USA Today

3. McPhail, Clark. "Individual and Collective Behavior within Gatherings, Demonstrations and Riots." Annual Review of Sociology (JSTOR) 9 (1983): 579-600.

4. Feinberg, William E. "Radical Leaders, Moderate Followers: Effects of Alternative Strategies on Achieving Consensus for Action in Simulated Crowds." Journal of Mathematical Sociology 15.2 (1990): 91- 115.

5. Feinberg, William E. "Elementary Social Structure and the Resolution of Ambiguity: Some Results from a Computer Simulation Model." Sociological Focus 23.4 (1990): 315-331.

6. Barabasi, Albert-Laszlo. Linked: How Everything Is Connected to Everything Else and What It Means. New York: Penguin Group, 2003.

7. McPhail, Clark. "Presidential Address – The Dark Side of Purpose: Individual and Collective Violence in Riots." Sociological Quarterly 35.1 (1994): 1-32.

8. Oliver, Pamela E. "Formal Models of Collective Action." Annual Review of Sociology 19 (1993), 271-300.

9. Staff Editorial, The Daily Campus, April 20, 2005 "Spring Weekend Plan An Inflated Cause."

10. Huguet, P., & Latane, B. (1996). Social representations as dynamic social impact. Journal of Communication, 46(4), 57-63.

11. Latane, B. (1996). Dynamic social impact: The creation of culture by communication. Journal of Communication, 46(4), 13-25.

12. Latane, B., & Bourgeois, M. J. (1996). Experimental evidence for dynamic social impact: The emergence of subcultures in electronic groups. Journal of Communication, 46(4), 35-47.

13. Latane, B., & Liu, J. H. (1996). The intersubjective geometry of social space. Journal of Communication, 46(4), 26-34.

14. Latane, B., & L'Herrou, T. (1996). Spatial clustering in the conformity game: Dynamic social impact in electronic groups. Journal of personality and social psychology, 70(6), 1218-1230.

15. Musse, S. R., Thalmann, D. "A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis." Swiss Federal Institute of Technology (EPFL).

16. Chwe, M. S.-Y. (1999). "Structure and Strategy in Collective Action." American Journal of Sociology 105-156(1): 128.

17. Gould, R. V. (1999). "Collective Violence and Group Solidarity: Evidence from a Feuding Society." American Sociological Review 64(3): 356-380.

18. McPhail, C. (1971). "Civil Disorder Participation: A Critical Examination of Recent Research." American Sociological Review 35(6): 1058-1073.

19. Berk, R. A. (1974). "A Gaming Approach to Crowd Behavior." American Sociological Review 39(3): 355-373.

20. Johnson, N. R. and W. E. Feinberg (1977). "A Computer Simulation of the Emergence of Consensus in Crowds." American Sociological Review 42(3): 505-521.

21. Pitcher, B. L., et al. (1978). "The Diffusion of Collective Violence." American Sociological Review 43(1): 23-35.

22. Stets, J., Burke, P. (2000). "Identity Theory and Social Identity Theory." Social Psychology Quarterly 63(3): 224-237.

23. Gilbert, N., Troitzsch, K. Simulation for the Social Scientist. New York: Open University Press, 2005.

24. Lipset, S. M., & Schaflander, G. M. (1971). Passion and Politics: Student Activism in America. Boston: Little, Brown.

25. Ohio State University. Task Force Report. *Task Force on Preventing Celebratory Riots.* April 7, 2003.

26. http://www.answers.com/topic/riot

27. Bales, Robert F. 1950. "A Set of Categories for the Analysis of Small Group interaction," *American Sociological Review.* 15:257-263.

28. Burke, Peter J. and Donald C. Reitzes. 1991. "An Identity Theory Approach to Commitment." *Social Psychology Quarterly* 54(3):239-251.

29. Hogg, Michael A., Deborah J. Terry, and Katherine M. White. 1995. "A Tale of Two Theories: A Critical Comparison of Identity Theory With Social Identity Theory." *Social Psychology Quarterly* 58:255-269.

30. Burke, Peter J. 1991. “Identity Process and Social Stress.” *American Sociological Review*. 56:836-49.

31. McCall, George J. and J. L. Simmons. 1978. *Identities and Interactions.* New York: Free Press.

32. Berger, Peter and Thomas Luckman. 1966. *The Social Construction of Reality*. Doubleday.

33. Granovetter, Mark. 1973 "The Strength of Weak Ties." *American Journal of Sociology* 78(6):1360-1380.