

# Using Evolutionary Models with Mutations to Predict Long-Term Trends in Simple Social Interactions

Barry Webster and Suja Ramakrishnan

**Abstract**—Predicting the outcomes of social interactions between humans is notoriously difficult. Variations within the experiences, beliefs, and actions of individual humans (and even within a particular given human from one situation to the next) render accurate predictions of the outcome of individual interactions problematic at best. However, it is somewhat easier to make predictions regarding the expected outcome of interactions involving large groups of humans over an extended period. This paper presents a series of studies where simple social interactions between humans of different personality types were modeled over a long term, and where the behavior patterns of individuals within the population were allowed to change. The results of these studies provide predictions for how groups of humans would likely behave in similar situations.

**Index Terms**—Agent-based simulation, evolutionary modeling, predictive modeling, social interactions.

## I. INTRODUCTION

One of the primary things that one would like to accomplish when using a model is to be able to use that model to predict future events. While it is folly to expect such a model to be 100% accurate under all circumstances, any reasonably accurate predictive model can still be very useful on a variety of fronts, from scheduling to contingency planning. Some fields have had great success in developing predictive models. For example, our ability to launch spacecraft and guide them successfully to a specific body within the solar system is due in no small part to the ability to create highly accurate predictive orbital mechanics models.

Other fields, though, have had much more of a struggle to produce accurate predictive models that can be relied upon consistently. One such field is in the area of predicting human behavior. Humans have demonstrated a great capacity for frustrating those who would attempt to predict what they are going to do in various situations. Each individual has a unique combination of experiences, beliefs, expectations, biases, etc., which results in a highly volatile mix rendering specific predictions regarding behavior difficult at best. Even particular individuals whose general behavior is well known can surprise us. Who among us has been perfectly consistent in the manner in which we have dealt with a particular situation, or has not found themselves in a position where we are wondering what in the world we were thinking when we performed some action? As Jim Morrison and The Doors famously (and accurately) put it, “people are strange.” [1]

Manuscript received January 10, 2015; revised May 25, 2015.

Barry Webster is with the Florida Institute of Technology, Melbourne, FL 32901 USA (e-mail: bwebster@fit.edu).

Suja Ramakrishnan was formerly with the Florida Institute of Technology, Melbourne, FL 32901. She is now employed by a private company in South Africa (e-mail: suja.ramakrishnan@gmail.com).

Having said that, we can still use models to provide some degree of predictive capability for gauging human behavior. One way of doing this is to attempt to model such predictions in the aggregate rather than at the individual level. That is, instead of attempting to predict what specific individuals are going to do in a given situation, an attempt is made to predict what groups of people as a whole are likely to do in that situation. In this way, an *expected* behavior is produced, which can then be treated as the expected action that a given individual will take in the situation. This expected behavior acts much like the result of a linear regression equation. Whereas it is highly unlikely that an individual data point predicted by the equation will have zero error, the equation itself is still useful in providing predictions that minimize the average amount of error in the data space [2].

This paper presents a series of studies conducted to produce just such a model – that is, a model of human behavior that determines an expected aggregate behavior pattern that can be used to predict the actions of persons in a given situation. The studies utilized a well-known situation involving simple interactions between two individuals, where the actions available to each individual were limited and easily measurable. The results of these studies yielded some interesting insights into human behavior patterns, not so much for the actual results themselves, but more so because of the way in which those results came about.

## II. BACKGROUND FOR THE STUDIES

As mentioned, the studies for this paper involved a well-known behavioral situation, one that has been used in a wide variety of studies of human behavior, known as the *Ultimatum Game*. This game requires two persons, one of whom is designated the *Proposer*, and another who is designated the *Disposer* (the names used to designate the two participants in the game vary somewhat, but their actual titles matter little – it is their roles that are important). The game includes one interaction between the two participants, where they are tasked with dividing some asset. The Proposer demands some particular division of the asset, and then the Disposer either agrees with or disagrees with the demanded division. If the Disposer agrees, the asset is divided according to the Proposer’s demand, and each participant receives their share of the asset from that division. If the Disposer disagrees, then the asset simply goes away and neither participant receives anything. Results of this game are easily measured; we simply record what the Proposer’s demand was, whether or not it was accepted by the Disposer, and what portions of the asset were given to each participant. This simplicity makes the Ultimatum Game a popular and easy-to-use choice for observing and measuring simple human interactions, and

many variations of it have appeared in the literature of several disciplines.

For the purposes of the studies for this paper, the Ultimatum Game was implemented as an agent-based simulation. This type of simulation involves the creation of a number of *agents*, or independent entities. Each of the agents is programmed with a particular set of behaviors that it is capable of performing, and it will interact with other agents within the simulation according to those defined behaviors. A simulation typically involves defining an environment in which the agents can maneuver and interact, and the experimenter observes what develops over time as a result of these interactions.

Agent-based simulation provides a useful medium for implementing *evolutionary modeling*. Evolutionary modeling is a sub-field of game theory, and also closely related to genetic programming within computer science. In an evolutionary model, a *population* is defined consisting of a collection of entities, each of which is of a particular *phenotype*. The phenotype of an individual entity determines its behavior, which is to say that the action(s) that an entity will perform during an encounter with another entity are “hard-wired” according to the entity’s phenotype. Thus, an entity will take whatever courses of action are allowed by its phenotype, and it does not have the option of deviating from these allowable actions. For example, if an entity is of phenotype “always pick heads”, then if it is engaged in an encounter with another entity in which it will be required to participate in a coin flip, it will always choose heads as its pick for the flip, and does not have the option of choosing tails under any circumstances.

An entity’s phenotype is determined at the time of its creation, and it does not have the option of choosing a different phenotype. Phenotypes may be as simple as specifying a single action that an entity will take under the appropriate circumstances, or as complicated as specifying an entire set of actions to be performed within a given set of circumstances. They can even be probabilistic, performing particular actions according to a given distribution of likelihoods. Returning to the example of the coin flip, an entity’s phenotype might dictate that it select heads 40% of the time and tails 60% of the time. This may sound like the entity is altering its behavior, but in actuality it is not. It will always behave according to the dictates of its phenotypes; it just so happens that in this case those dictates are determined by a probability distribution.

Entities within the population are allowed to interact, and their ability to perform well in encounters with others is determined by their relative *fitness* for handling the types of encounters that may occur. It may be the case that entities of phenotype A perform better than entities of phenotype B in encounters between individuals of those two phenotypes. In such a case, entities of phenotype A are considered more fit than entities of phenotype B. The fitness of an entity determines its likelihood of reproducing; those that are more fit will produce a relatively higher number of other entities of that phenotype than those that are less fit.

Experimenters utilizing evolutionary models are typically going to be looking at one or both of two facets of the model’s performance. In the *dynamic* view, the experimenter will be looking at what happens to the ratios of the

phenotypes of the entities within the experimental population over time, in order to observe whether the ratios will change and a stable state will be reached. Differences in the relative fitness of phenotypes may result in one particular phenotype being more fit than all others. If so, over time it is expected that the ratios of the numbers of entities of the non-dominant phenotypes to the numbers of entities of the dominant phenotype will decrease (due to the relative reproduction rates resulting from the differences in fitness), and eventually all entities of the non-dominant phenotypes will disappear from the population altogether. Once this happens, the population will have achieved a stable state in which there is only one phenotype present. Such a population is said to be *monomorphic*. It could also be the case that the population reaches equilibrium in a state where multiple phenotypes have nonzero percentages of entities present. This configuration of population is categorized as *polymorphic*.

A second facet of the population’s performance is the *static* view. Here, the experimenter is looking to see if it is possible for a given stable population to be successfully “invaded” by entities of one or more other phenotypes. Again, due to differences in relative fitness it may be possible for entities of a particular phenotype to enter a population and take it over. Their fitness is greater, and consequently they will perform better in interactions with existing entities within the population. As such, their numbers will eventually cause the other phenotypes to disappear from the population. For the purposes of the studies for this paper, it was the dynamic view of population performance that was of interest, since the intent was to attempt to discover how patterns of human behavior would evolve over time.

It should be noted that encounters within a population do not take place only between entities of different phenotypes. Encounters can and do occur between entities of the same phenotype, and the result is not necessarily a draw. For example, suppose that when two entities of phenotype A meet, there will be some sort of contest. It may be the case that half the time the first entity will win the contest, and half the time the second entity will win. It could also be the case, though, that both entities suffer damage as a result of the encounter, and either or both may become unable to produce any additional entities to perpetuate the phenotype.

In addition to running the evolutionary model as an agent-based simulation, a mathematical process can be used to derive what the stable state of the population should be. This works by comparing the relative fitness of the phenotypes in the population to determine the ratios of those phenotypes at which the expected fitness of each phenotype is the same. Recall that the expected value of a function is the value of an outcome of that function multiplied by the frequency of that outcome’s occurrence, summed across all possible outcomes of the function. For an evolutionary model, the expected fitness of a phenotype will be the fitness value of that phenotype multiplied by the frequency of the phenotype’s occurrence within the population. When the expected fitness for all phenotypes within the population is the same, the population becomes stable, and the proportion of each of the phenotypes within the population is no longer expected to change. This is referred to as the *replicator dynamics* of the population [3].

To illustrate this process, consider a very simple example

given in the matrix shown in Fig. 1, produced using a game design and analysis tool called Gambit [4]. This matrix shows the possible outcomes of an encounter between two entities from a population. One entity is represented by the rows of the matrix, the other by the columns. The capital letters A and B represent two phenotypes within the population. The numbers represent the results of an encounter, where the first number in a cell of the matrix represents the result to the “row” entity, and the second number the result to the “column” entity. Thus, to find the results of an encounter between one entity of phenotype A and another of phenotype B, we can examine the cell in the matrix positioned at the intersection of the row marked “A” and the column marked “B”. Here we see that the result to the A entity is 50, and the result to the B entity is 0. Alternatively, we could look at the cell positioned at the intersection of the row marked “B” and the column marked “A”, where we would find the same results. The matrix is symmetric, so it does not matter which method is used to locate the encounter results; for the purposes of this example, we will use the former method (row first, then column).

		A		B	
A	-15	-15	50	0	
B	0	50	15	15	

Fig. 1. Example fitness matrix.

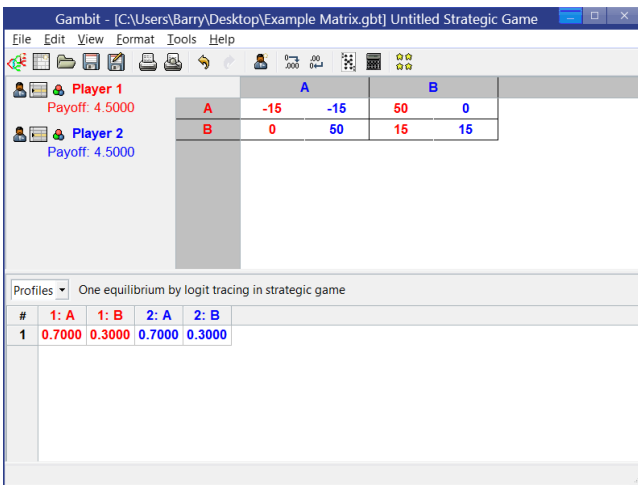


Fig. 2. Gambit solution to example fitness matrix.

If we let  $p$  be the percentage of entities of phenotype A in the population, then the percentage of entities of phenotype B in the population will be  $(1 - p)$ , since A and B are the only phenotypes present. Using the definition of expected fitness (call it  $EF$ ) given earlier, the equation for the EF of phenotype A is given in (1), and the equation for the EF of phenotype B is given in (2).

$$EF(A) = -15p + 50(1 - p) = 50 - 65p \quad (1)$$

$$EF(B) = 0p + 15(1 - p) = 15 - 15p \quad (2)$$

To find the point of equilibrium, we need to set (1) and (2) equal to each other and solve for  $p$ . This is given in (3).

$$50 - 65p = 15 - 15p$$

$$35 = 50p$$

$$p = 35/50 = 0.7 \quad (3)$$

Thus, we see that when the population consists of 70% phenotype A and 30% phenotype B, the expected fitness of each phenotype is the same. We would expect, then, that regardless of the starting percentages of each phenotype, the population would evolve until it reaches the state of 70% A, 30% B, at which point it would become stable. When we use Gambit to find the equilibrium point, we get the same result, as shown in Fig. 2, [4].

The replicator dynamics for a population can be used as a reference point when running an agent-based simulation of an evolutionary model, to observe whether the simulation is converging to the same equilibrium point as calculated by the replicator dynamics.

### III. THE STUDY EXPERIMENTS

The studies for this paper consisted of a series of agent-based simulations of evolutionary models, where the environment for the encounters within the population was an Ultimatum Game. The simulations were constructed using a tool called NetLogo [5]. NetLogo is a utility that allows the building of agent-based simulations within a self-contained development environment, using its own user interface design facility and simulation programming language.

The interface for the NetLogo environment created for the studies is shown in Fig. 3, [5]. The “world” of the simulation is the square in the middle of the interface, shown containing icons of different colors representing the individual entities of particular phenotypes within the simulation population. These entities can move around within the square, initiating encounters with each other when they occupy the same position.

To the left of the “world” is a set of parameter controls for customizing the simulation. This set includes:

- Buttons for initializing and running the simulation
- A slide control for setting the initial population size
- A switch to generate a random initial distribution of phenotypes or allow manual setting
- Slide controls to manually set the initial distribution of phenotypes if this mode of initialization was selected
- Drop-down menus to set the initial and maximal “energy” of individual entities within the population
- Controls for setting simulation stop conditions (threshold for minimum change of proportion between phenotypes per generation to qualify as significant, threshold for maximum number of consecutive generations allowed without any significant change in phenotype proportions, and maximum number of generations allowed)
- Controls for setting the manner (globally in each generation or as a result of individual encounters) and degree (frequency of occurrence) in which mutations appear during the simulation

Finally, a set of counters, histograms, and line graphs for monitoring simulation progress appears to the right and bottom of the interface (some cannot be seen in Fig. 3 since the complete interface did not fit on the screen all at the same time). These objects are used to provide both current and running tallies of the counts and percentages of entities of each phenotype within the population.

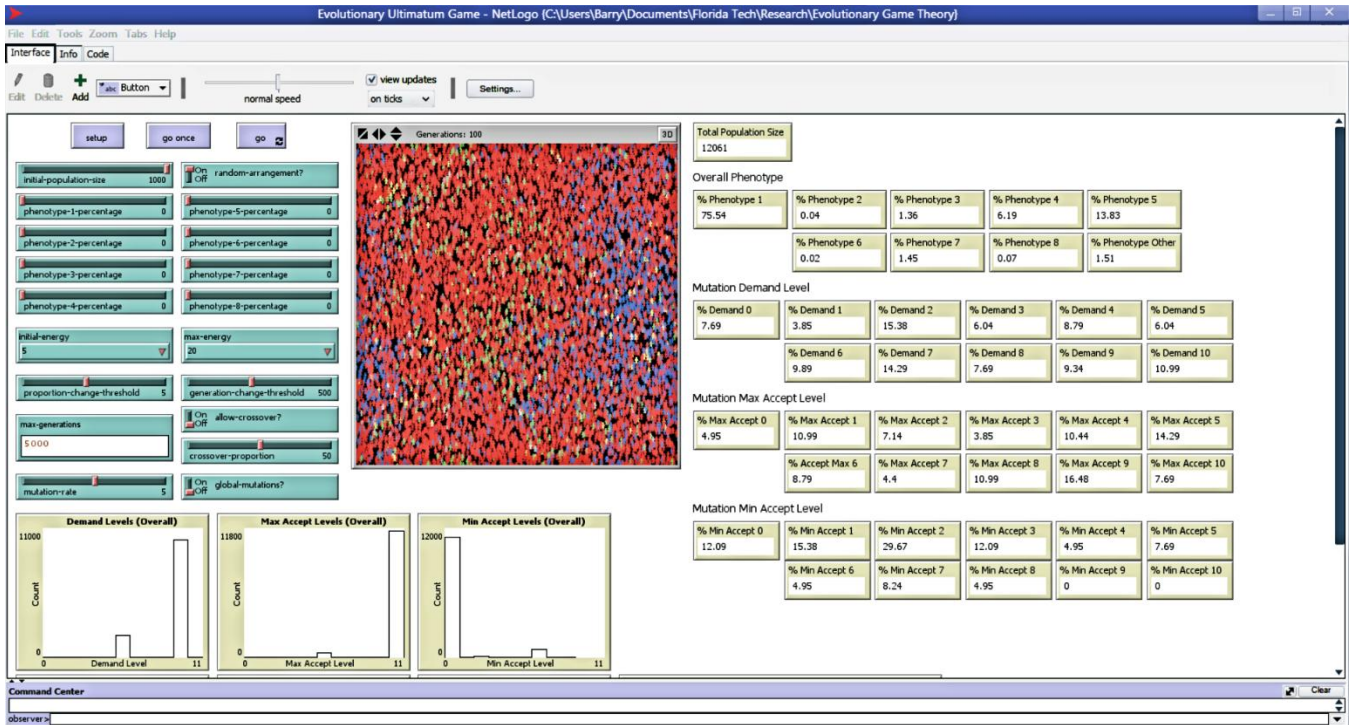


Fig. 3. NetLogo interface for the experimental studies.

The particular implementation of the Ultimatum Game used for the simulations was to postulate cakes divided into 10 slices, one cake for each encounter between two entities. The entity designated the Proposer would demand some number of slices of the cake, and the entity designated the Disposer would then accept or veto the demand. If the demand was accepted, both entities received their allotted number of slices of the cake. If the demand was vetoed, neither entity received anything. Each slice of cake received increased the “energy” of an entity, allowing it to continue to exist within the population.

Entities to be included within the initial configuration of the population could belong to one of eight phenotypes. These eight phenotypes were designed to mimic those defined by Brian Skyrms, who provided the phenotypes as part of his discussion of the evolution of social interactions [6]. In his definitions of the phenotypes, Skyrms allowed for only two levels of demand when entities acted as Proposers – either five slices or nine slices of the cake. Consequently, his definitions only specified that entities would either accept or reject demands of five or nine slices when acting as Disposers. However, in the simulations for these studies it was possible for mutated phenotypes to be introduced. These mutated phenotypes could result in any demand level, from none of the slices of cake to all of them. In order to be able to deal with all of the possible demand levels, the acceptance level for a phenotype was defined in terms of a minimum accept level and a maximum accept level. This meant that when acting as a Disposer, entities of a given phenotype would accept any demand greater than or equal to the specified minimum accept level, and less than or equal to the specified maximum accept level. The resulting initial phenotype definitions are shown in TABLE I.

Encounters occurred whenever two entities attempted to occupy the same section of grid within their navigable “world”. During an encounter, one of the entities would randomly be designated as the Proposer, and the other entity

would then be designated as the Disposer (each entity had an equal probability of being selected as the Proposer). Once an encounter had been initiated, the entity designated as the Proposer would make its demand according to its phenotype. If the demand fell between the minimum and maximum accept levels (inclusive) of the entity designated as the Disposer according to its phenotype, the encounter was deemed successful. The Proposer entity would receive its demanded number of slices of the cake, which resulted in a matching increase in its energy level. Likewise, the Disposer entity would receive the remaining slices of the cake and matching increase in its energy level. Successful encounters also resulted in the creation of one additional entity having the same phenotype as the Proposer entity, and one additional entity having the same phenotype as the Disposer entity. Unsuccessful encounters resulted in no increase to the energy level of either entity, nor the creation of any additional entities.

TABLE I: DEFINITIONS OF PHENOTYPES AVAILABLE FOR INITIAL POPULATIONS

Phenotype	Demand Level	Min. Accept Level	Max. Accept Level
1	9	0	10
2	9	0	0
3	9	0	5
4	9	6	10
5	5	0	10
6	5	0	0
7	5	0	5
8	5	6	10

As mentioned, there was also the possibility that mutations could arise during the simulation. Following successful encounters, a mutation could occur randomly according to a rate that could be set as a parameter by the experimenter. If a mutation occurred, the new entity created for the Proposer would have its demand level, minimum accept level, and maximum accept level each set to a random integer value

between zero and 10 (with the restriction that the maximum accept level be at least as high as the minimum accept level). The same would then happen for the new entity created for the Disposer. Mutations could also be set (again, by parameter) to occur at the global level. If this was the case, mutations would occur within the existing population rather than when new entities were created, and would occur randomly according to the mutation rate set. If a mutation occurred for an entity, its current demand level, minimum accept level, and maximum accept level would each be replaced by a random integer between zero and 10 (again, with a maximum accept level at least as high as the minimum accept level). The mutation mechanism allowed for the possibility of a substantial number of new phenotypes to be introduced into the population in addition to the eight original predefined phenotypes. This in turn allowed for the possibility that one or more phenotypes other than the one(s) present at the beginning of the simulation could become a sizable presence within, or even dominate, the population.

Simulations were run in a series of turns that represented “generations” of the population. For each turn, entities would move one unit of distance in a random direction and their energy level would decrease by one. Once all entities had moved, encounters and mutations would be processed. If an entity’s energy level reached zero, it would “die” and be removed from the population. Of course, an entity’s existence could be prolonged through successful encounters, where its energy level would be increased by receiving slices of cake (up to a maximum energy level defined by parameter). A simulation would end when any of four conditions were met:

- No entities remained within the population
- All entities within the population were of the same phenotype (i.e. the population had become monomorphic)
- No sufficiently large change had occurred in the percentages of any of the phenotypes represented in the population for a certain number of consecutive generations (as mentioned earlier, both the threshold defining a “sufficiently large change” and the necessary number of consecutive generations were parameters that could be set by the experimenter)
- A maximum number of turns had elapsed (again, as mentioned earlier this also was a parameter that could be set by the experimenter)

To conduct the actual simulations for the studies, a number of preliminary test simulations were first performed. This was done for three purposes: to test the NetLogo environment itself to ensure that it was functioning properly, to get a feel for the various parameter settings and the effects they would have on the operations of the simulations, and to provide a basis for estimating the number of sample runs of each simulation that would be required in order to allow for accurate statistical analysis of the results. Skyrms discussed several scenarios involving different configurations of all eight of his phenotypes [6]. As part of the preliminary test simulations, the scenarios that Skyrms discussed were executed. These scenarios were:

- 1) Begin with a population where all eight phenotypes have equal percentages of entities (12.5%). This

represents a population where all behaviors are equally likely at first.

- 2) Begin with a population where 30% of the entities are of phenotype 7, with the remaining phenotypes having equal percentages of entities (10% each). This represents a population where a plurality of people begin by behaving in a manner that would typically be considered fair and rational, with all other behaviors being equally likely.
- 3) Begin with a population that is 32% phenotype 1, 2% phenotype 2, 10% phenotype 3, 2% phenotype 4, 10% phenotype 5, 2% phenotype 6, 40% phenotype 7, and 2% phenotype 8. This represents a population that Skyrms considered to be of a more “plausible” nature, with a plurality of people still behaving fairly and rationally (with some smaller groups behaving fairly but perhaps irrationally), but also with a sizable number of people behaving in a greedy (and perhaps irrational) manner [6].

Following the preliminary test simulations, a set of parameter values was decided upon for the actual study simulations. These settings are given in TABLE II:

TABLE II: SIMULATION PARAMETER SETTINGS FOR STUDY EXPERIMENTS

Parameter	Setting
Initial population size for each simulation	1,000
Random arrangement of phenotypes	Off
Initial energy level for new entities	5
Maximum energy level allowed for entities	20
Global mutations	Off
Mutation rate (expected number of mutations per 1,000 opportunities)	5
Percentage change in phenotype counts between turns considered to be significant	5
Number of consecutive turns without a significant percentage change triggering a stop condition	500
Maximum number of turns (generations)	5,000

Using these parameter settings, the Skyrms scenarios were simulated. Since Skyrms did not allow for mutations in his scenarios, mutations were disabled for these experiments. However, another complete set of the Skyrms scenarios were then run, this time allowing for mutations using the mutation rate given in TABLE II. In all cases, for each scenario the experiment was repeated 100 times to achieve statistical significance and the average results for each experiment (scenario) were recorded.

#### IV. EXPERIMENTAL RESULTS

The experimental results for the Skyrms scenarios without mutations were as follows:

- Scenario 1 – the replicator dynamics for this scenario expect that the population will stabilize when it achieves the state of having 87% phenotype 1 and 13% phenotype 4. The experimental simulations resulted in populations with an average of 86.8% phenotype 1 and 13.2% phenotype 4.
- Scenario 2 – the replicator dynamics for this scenario expect that the population will stabilize at 35.1% phenotype 5 and 64.9% phenotype 7. The experimental simulations resulted in populations with an average of 97.34% phenotype 1, 0.65% phenotype 4, 0.62%



phenotype 5, and 1.39% phenotype 7.

- Scenario 3 – the replicator dynamics for this scenario expect that the population will stabilize at 43.5% phenotype 5 and 56.5% phenotype 7. The experimental simulations resulted in populations with an average of 98.75% phenotype 1, 0.07% phenotype 4, 0.56% phenotype 5, and 0.62% phenotype 7.

As can be easily seen from these results, in the case of scenario 1 the experimental results matched nearly exactly with the replicator dynamics expectations. However, in the cases of scenarios 2 and 3 the experimental results were entirely different from the replicator dynamics expectations. The replicator dynamics expect that when populations start out consisting of equal numbers of people who are greedy and fair, rational and irrational, it will eventually be the case that the population will consist mostly of people who will be greedy in their demands but amenable to making all kinds of deals, and the experimental simulations agreed. The differences come when populations start out consisting of a plurality of people who are fair and rational. In these situations, the replicator dynamics expect that it will eventually be the case that the population will consist entirely of people who are fair in their demands, with a majority being rational in the deals that they will accept and the remainder just wanting to make deals without being concerned with whether those deals are fair to them. In the experimental simulations, though, it was by far the case that the populations ended up consisting almost exclusively of people who are greedy in their demands but wanting to make deals, with a smattering of people who are greedy but irrational, fair and deal-making, and fair and rational.

These differences became even more pronounced when mutations were introduced into the models. When this was done, there were a few cases where mutated phenotypes never really gained a foothold, but in the vast majority of cases one or more of the mutated phenotypes came to dominate the populations. It took a long time for this to happen, usually over 3,000 generations, and it also was the case that the precise mutation that would come to dominance varied from time to time. However, what was consistent was the fact that phenotype 1 would usually become dominant over the other original phenotypes, and then a mutation would evolve to dominate phenotype 1. Recalling that phenotype 1 demands nine slices of the cake and accepts any level of demand, the mutation that would evolve to defeat it was one that would demand all 10 slices of the cake but would accept only demands of less than nine slices. What would happen then is that when the mutant entities acted as Proposers and the phenotype 1 entities acted as Disposers, the mutant entities would demand all 10 slices and the phenotype 1 entities would accept, leaving the mutant entities with 10 slices of cake and the phenotype 1 entities with nothing. When the situation was reversed, the phenotype 1 entities would demand nine slices of the cake and the mutant entities would reject the demand, leaving them both with nothing. On average, then, the mutant entities would end up with some slices of the cake and the phenotype 1 entities would end up with nothing, meaning that the mutant entities were more fit, and thus the phenotype 1 entities would slowly die out. This property is what allowed a variety of mutant phenotypes to

dominate; all that was required was that the mutant phenotype demand 10 slices of the cake and accept any demand less than nine.

Of note here is the fact that the simulation was designed to stop once the entire population consisted of a single phenotype. This was due both to the fact that the population had become monomorphic (indicating that one particular phenotype had achieved supremacy), and also to the fact that simulations running into the thousands of generations took a substantial amount of time, so the desire was to complete each simulation as soon as any reasonable stopping state was reached. In this case, that meant that the simulation stopped when the mutant entities overwhelmed the last of the phenotype 1 entities. However, had the simulation continued, eventually all of the mutant entities would have died out as well. This is because with all of the phenotype 1 entities gone, the only possible encounters would be between the mutant entities themselves. Since the mutant entities demand 10 slices of cake and accept only demands for less than nine, successful encounters could not occur and they would start to die out, paving the way for perhaps a different mutant phenotype to become dominant and carry on the population.

## V. CONCLUSION

It was quite clear from the results of the experimental studies that the agent-based simulation evolutionary models did not reach the same conclusion in the majority of cases as the traditional replicator dynamics method. The replicator dynamics suggest that in most cases, populations will evolve to a state where the majority of the people within those populations are fair and rational. Conversely, the experimental models suggest that in most cases, populations will evolve to a state where the majority of people within those populations are greedy, ambitious, and not completely rational, even if the population begins with a plurality of people who want to be fair and rational.

So how do we explain these differences? The preliminary tests all showed that the models were functioning as designed, but could it be the case that the designs themselves were flawed? It must be acknowledged that this is a possibility; the operating parameters were set to values that allowed the simulations to run smoothly and in reasonable amounts of time, but whether they were realistic is a different question. The stop condition thresholds were set such that the models were not too sensitive to changes within the population nor too insensitive, but perhaps in reality populations are more sensitive to some types of changes than to others. A mutation rate of five per 1,000 entities is certainly not unrealistic from a biological standpoint, but single mutations seldom result in changes as dramatic as those that occurred in the simulations. It should be noted, too, that there were quite a few occasions when the simulations did match the replicator dynamics, even though on average they did not. Skyrms himself also offered a possible explanation. He showed that under certain circumstances, populations could enter states of extreme unfairness, even when they begin in states of fairness [6].

One other explanation (and one that the authors of this paper tend to favor) is that perhaps the models were not so far off after all. The reader is invited to take a look at the society surrounding us. Is it not often the case that we encounter

people who want to work with us, yet are also quite greedy in the demands they make of us? It is true that these observations are anecdotal, and subject to a host of biases that are common to human observations, yet the fact remains that what we see happening around us tends to match better with what the models are telling us rather than what the traditional replicator dynamics are telling us.

Another finding with tantalizing implications is related to the aforementioned fact that if the simulations had been allowed to continue beyond the point where the mutant entities took over, those entities would have themselves eventually died out or been taken over by yet another mutant variety having more flexibility. This suggests that populations consisting entirely of people who are greedy and uncompromising are unsustainable, and that at least a certain amount of fairness and rationality is necessary in order for the population to survive.

One other finding that was interesting was that there was no built-in intelligence in the models, yet the mutations almost always became configured in such a way that they came to dominate the population. Whenever a particular original phenotype started to rise to dominance, mutations also arose that would dominate, or at least compete favorably with, that phenotype. This compares very well to how mutations operate in nature. Many natural mutations are useless if not downright harmful, and tend to vanish, but a few prove to be advantageous and become plentiful. It was actually quite fun to watch the models as they ran, and observe the mutations as they gradually "figured out" how to configure themselves such that they could rise to prominence.

It is not so much the specific results of the studies, but rather the models themselves that are the most promising. The models were able to take a classic measuring tool for human behavior and reach conclusions which mirror well what we see happening in everyday interactions, without requiring excessive computational resources (the typical simulation was handling up to 18 different phenotypes and over 12,000 entities simultaneously, on an office computer, without overtaxing it). This suggests that the models can be used for other situations involving human behavior. They can be expanded to handle more complex interactive environments and more options for the types of encounters that can occur. In doing so, some very interesting predictions may be developed.

#### REFERENCES

- [1] The Doors, Composer, *People Are Strange*. (Sound Recording). Elektra, 1967.
- [2] R. T. Clemen and T. Reilly, "Chapter 10 - Using Data," in *Making Hard Decisions with DecisionTools*, 3rd ed., Mason, OH, South-Western, Cengage Learning, 2014, pp. 447-471.

- [3] J. N. Webb, "Chapter 9-Replicator dynamics," *Game Theory: Decisions, Interaction and Evolution*, Springer Science & Business Media, 2007, pp. 165-185.
- [4] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software Tools for Game Theory, Version 14.1.0. [Online]. Available: <http://www.gambit-project.org>.
- [5] U. Wilensky. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. [Online]. Available: <http://ccl.northwestern.edu/netlogo/>.
- [6] B. Skyrms, *Evolution of the Social Contract*, 1st ed., New York, NY: Cambridge University Press, 1996.



**Barry Webster** was born in Elmira, NY in 1963. He earned a B.S. in computer science from the Pennsylvania State University in University Park, PA in 1984. He went on to earn an M.S. in computer science in 1995, a Ph.D. in computer science in 2004, and an M.S. in systems engineering in 2005, all from the Florida Institute of Technology in Melbourne, FL.

He began his career in 1985 working as a systems engineer for Grumman Aerospace Corporation on Long Island, NY. He spent a total of 21 years with the company (which eventually became Northrop Grumman Corporation), holding positions in systems, software, and support engineering departments, ultimately becoming the senior database administrator for the engineering directorate. In this capacity, he was responsible for managing the operation of over 50 databases and associated applications and tools, used at installations around the world. After relocating to Melbourne, FL in 1987 he became associated with the Florida Institute of Technology, first as a student, earning three graduate degrees while working full time, then as a researcher on a sponsored program, and then as an adjunct professor. In 2007 he made the transition to Florida Tech as a full-time member of the faculty within the Department of Engineering Systems (though still consulting for Northrop Grumman part-time for another three years), where he remains to the present day. As part of his Ph.D. studies, he was also accepted for a doctoral internship at NASA's Ames Research Center in Mountain View, CA, where he worked in the Autonomous Systems Department on automated telescope control systems and algorithms for solving NP-Hard telescope scheduling problems. His research interests include artificial intelligence, decision theory, and game theory.

Dr. Webster is a member of the Association for Computing Machinery (ACM), the Institute for Operations Research and the Management Sciences (INFORMS), and the Game Theory Society (GTS). He has chaired conference sessions in database management and artificial intelligence, received several citations for his work in database management, and two Best Paper/Best Presentation awards for works on game theory.



**Suja Ramakrishnan** was born in India. She received her B.Eng. degree from the Visvesvaraya Technological University in Jnana Sangama, Machhe, Belgaum, Karnataka, India. She also holds an M.S. degree in operations research from the North Carolina State University in Raleigh, NC, and a Ph.D. in operations research from the Florida Institute of Technology in Melbourne, FL.

She has held positions as a graduate teaching assistant and research assistant at the Florida Institute of Technology, and is now employed by a private company in South Africa. She has published several papers in the areas of mathematics and operations research. Her research interests include solving linear systems, decision theory, and game theory.