Smart Frequentists Drink Alone:

PSCS 530 W98 (Riolo) Project 1 Writeup

David M. Pennock
University of Michigan AI Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109-2110 USA
dpennock@umich.edu

April 29, 1999

1 Introduction and Background

In Santa Fe, New Mexico there is a bar called *El Farol*. It has all of the usual features that draw people to bars: beer, wine, drinks, food, people and atmosphere. Residents especially enjoy Thursday evenings, which feature live music. That is, unless too many people crowd the bar, and it becomes uncomfortable, smoky, and difficult to hear friends and music. Every Thursday each resident, including Maria, must make an excruciating decision: risk the crowds and go to El Farol, or sacrifice the potential of invigorating live music and sit at home. If Maria believes that everyone else is going, than she would prefer to stay at home. On the other hand, if many people reason this way and thus stay home, then she would prefer to go to the bar. There are many analogous situations where agents are rewarded for being in the minority: drivers prefer to avoid heavy traffic and computer network users prefer to access the least congested connections.

To explore the dynamics of minority competition, Savit, Manuca, Li, and Riolo [1] propose a simple model that captures the essential flavor. At each time step, each of N agents makes one of two choices: either go to the bar (choice "1") or stay at home (choice "0"). The bar is considered too crowded (bar state "0") if more than half of the agents go, and comfortable (bar state

"1") otherwise. Agents receive one reward point if they go to the bar and it is comfortable, or if they stay home and it is crowded. Thus they are rewarded if and only if their choice is the minority decision.

It is hard, if not impossible, for an agent to formulate a completely rational strategy. Certainly there is no dominant strategy which is always best regardless of what everyone else chooses. Savit, Manuca, Li, and Riolo [1] propose what I call the *lookup* agent. This agent has available to it nS strategies and can remember the outcome of only the previous $m0$ time steps. Each strategy is a lookup table that specifies a choice for every possible combination of the previous $m0$ outcomes. For example, if $m0 = 2$, a strategy assigns a choice of 0 or 1 to each of the four possible histories 00, 01, 10, and 11. There are $2^{2^{m0}}$ possible strategies. A lookup agent has access to only nS of these, assigned at random with replacement. The agent chooses from among its available strategies by determining which would have done the best in the past.

It is unclear whether formal, equilibrium analysis of this model is meaningful or realistic since, by definition, at each time step more than half of the agents would prefer to change groups. Given the current limits of out-of-equilibrium mathematical tools, empirical simulations may be better suited for providing insights into the nature of minority competition. Savit, Manuca, Li, and Riolo carry out a series of experiments with populations of lookup agents. The model is implemented in SWARM, a powerful simulation environment written in Objective C. By following the instructions written by Brian Mulloy, and with help from Rick Riolo, I was able to copy, compile, and run the existing code fairly easily. The instructions also explained how to generate a *MatLab* plot that recreates one the authors' main results. Figure 1 graphs the standard deviation of bar attendance versus $m0$. The standard deviation can be thought of as a proxy for system inefficiency. If the standard deviation is very high, then the bar attendance swings wildly between nearly empty and overcrowded. If the standard deviation is low, then almost half of the agents are rewarded every round (assuming that, on average, about half of the agents go to the bar). As illustrated in Figure 1, the system is inefficient for small $m0$, is most efficient around $m0 = 6$, and is moderately efficient for $m0 \geq 9$.

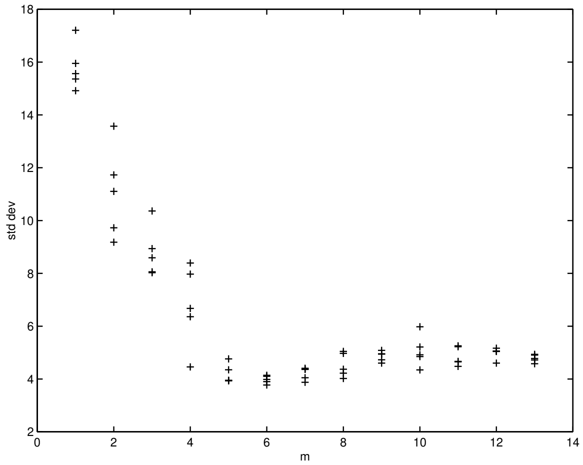

Figure 1: Standard deviation of bar attendance versus $m0$, as reported in [1]
.

2 Frequentist Agents and Implementation

Figure 1 suggests that inefficiencies exist in a system of lookup agents with small $m0$. A natural question is whether other types of agents can recognize and exploit this inefficiency. Toward this end, I propose a *frequentist* agent, which remembers the outcomes of the previous $m1 = m0 + 1$ time steps, and which chooses the option that looks most favorable according to the current conditional probabilities. For example, suppose that $m0 = 1$, $m1 = 2$, and in the previous two time steps the bar has been crowded (state sequence 00). Then a frequentist agent will go to the bar if $\Pr(1|00) > 0.5$ and will stay at home if $\Pr(1|00) < 0.5$. The agent computes this probability by counting the number of times in the past that the bar has been comfortable immediately following two days of crowds (hence the term frequentist). If this is the first time the sequence 00 has ever occurred, then the agent behaves as a lookup agent.

I modified the existing simulation code to admit frequentist agents. The program already had facilities to support two types of lookup agents with two different memory lengths, $m0$ and $m1$. The parameter $m\mathrm{R}$ specifies the fraction agents with memory length $m0$. The biggest change was made in the function `makeChoice` in the file `Agent.m`. If the agent's memory length is $m0$, then it behaves as a lookup agent; if the agent's memory length is $m1$, then it behaves as a frequentist. Thus if $m\mathrm{R} = .99$ and there are 101 agents, then 100 of these are lookup agents and one is a frequentist.[1] If $m\mathrm{R} = .2$, then 20% are lookup agents and 80% are frequentists. The array `histogram1` in the `Bar` class contains the information necessary to calculate the conditional probabilities. The remaining changes in `Parameters.m` and `ModelSwarm.m` ensure that $m1 = m0 + 1$ and otherwise support the new agent type.

By modifying a `drone.ctr` program written by Brian Mulloy, I used `drone` to carry out a series of experiments. I wrote two short C++ programs to extract relevant information from the data files generated by `drone`, and to output the results in a format easily interpreted by *Mathematica*. Most of the graphs in Section 3 were created in *Mathematica*. I also modified some of the *MatLab* scripts written by Brian Mulloy to produce the graphs of standard deviation versus $m\mathrm{R}$. In general, I found the SWARM and *MatLab* code fairly well written and accessible. More documentation would have sped up the process, but what documentation exists is clear and helpful.

3 Results

One of my initial goals was to determine whether frequentist agents are able to exploit inefficiencies in the system. To measure their performance, I calculated what I call the *average wealth ratio*. This quantity is the average reward per agent among the frequentists divided by the average reward per agent among the lookup agents. The first experiment measures the average wealth ratio for a single frequentist agent in a population of 100 lookup agents (i.e., $N = 101$ and $m\mathrm{R} = 0.99$). Figure 2 plots the results for the same range of values of $m0$ as in Figure 1. Each point is an average over five runs. Note the striking similarity between Figures 1 and 2. A single frequentist's success is highly correlated with the standard deviation of attendance: when the standard deviation is high, frequentists fare well. Average wealth ratio can also be thought of as a measure of system inefficiency. A single frequentist accumulating more than its share of wealth is a signal that some exploitable regularities exist in the system.

Figure 3 shows that frequentist agents work best alone. This figure plots the average wealth ratio versus $m\mathrm{R}$ when $m0 = 1$. When there is only one

[1] I modified the semantics of $m\mathrm{R}$ very slightly to round to the nearest agent, rather than always rounding down.

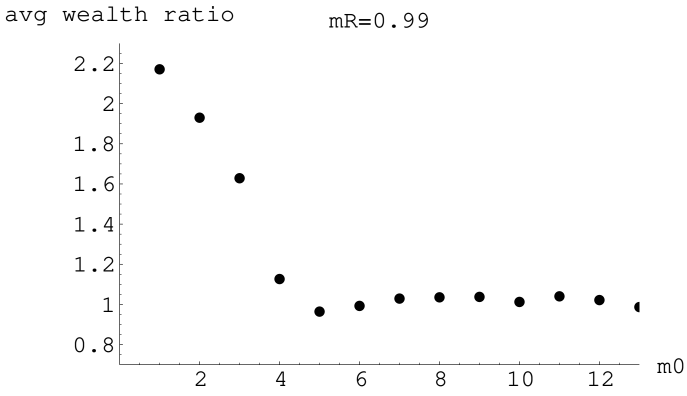

Figure 2: Average wealth of a single frequentist agent divided by the average per-agent wealth of 100 lookup agents for various values of $m0$. For small $m0$, the frequentist agent can exploit inefficiencies in the system. This graph mirrors the shape of Figure 1.

frequentist ($m\mathrm{R} = 0.99$), it performs more than twice as well as the lookup agents. As the percentage of frequentists grows, their average performance drops rapidly. When the population consists of more than 40% frequentists ($m\mathrm{R} < 0.6$), their average reward is almost zero. This is because all frequentists are created equal. Each computes exactly the same conditional probabilities, and each makes exactly the same decisions (except the first time a particular series of outcomes is encountered). Thus frequentists behave as a herd, all rushing to the bar or remaining at home in unison. Whatever their choice, it is bound to be the majority decision.

Figure 4 graphs the standard deviation of attendance versus $m\mathrm{R}$, again for $m0 = 1$. The figure has some of the same qualitative characteristics as Figure 1 (although the absolute magnitudes are higher here). The standard deviation is highest when the system is dominated by frequentists and their herding mentality. The standard deviation is also relatively high when the population consists almost solely of lookup agents. There is a minimum in standard deviation (and thus a maximum in system efficiency) when the fraction of lookup agents is between 0.8 and 0.9. Note that this is roughly the same point at which the average wealth ratio equals one. The system appears to be most efficient when the population maintains a "fair" balance between frequentist and lookup agents, such that each group earns roughly

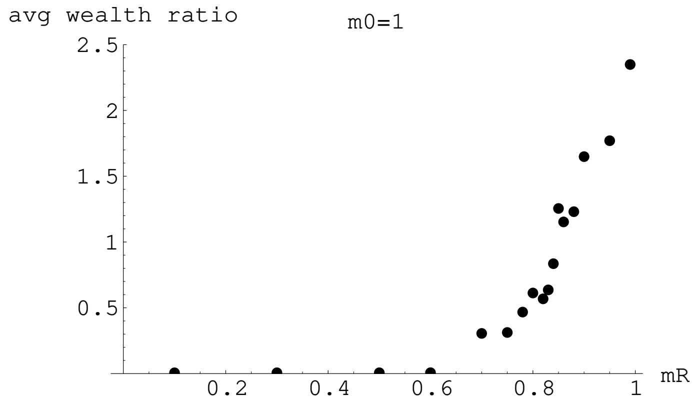

Figure 3: Average wealth ratio versus the fraction of lookup agents for $m0 = 1$.

the same average rewards.

The histogram in Figure 5 is essentially a replication of a result in [1]. It demonstrates that, in a system dominated by lookup agents with $m0 = 2$ (with only one frequentist), almost no predictive information is available to agents with memory length two. The conditional probability that the bar will be comfortable is almost exactly 0.5, given any of the four combinations of outcomes in the previous two time steps.

Figure 6 is also similar to a result in [1]. It shows that, although there is no predictive information available to agents with $m0 = 2$, there is considerable information visible within a three time step window. The conditional probability that the bar will be comfortable varies widely across the eight different combinations of outcomes in the previous three time steps. It is this structure which the frequentist agents are able to exploit. Indeed, the single frequentist in the system portrayed in Figures 5 and 6 collects just under twice as much reward on average as one of the 100 lookup agents.

As more and more frequentists enter the system, any surplus available to agents with $m1 = 3$ is quickly exhausted. Figure 7 displays the situation when there are 20% frequentists and 80% lookup agents. All eight conditional probabilities are now nearly 0.5.

As $m0$ increases, and the system becomes more efficient, there is less structure for frequentist agents to exploit. Once $m0 = 4$, a single frequentist agent among 100 lookup agents garners only about 1.2 times as much wealth.

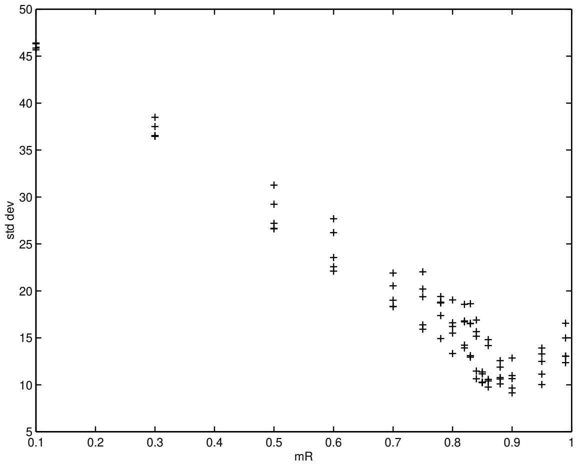

Figure 4: Standard deviation of attendance versus the fraction of lookup agents (mR) for $m0 = 1$. The value of mR where the standard deviation is a minimum is approximately the value for which the wealth ratio equals one.

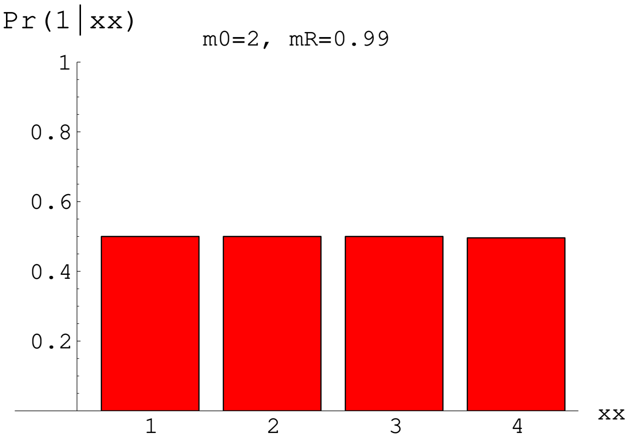

Figure 5: Conditional probability that the bar is not crowded, given its state in the previous two time steps. The system consists of one frequentist agent and 100 lookup agents with $m0 = 2$.

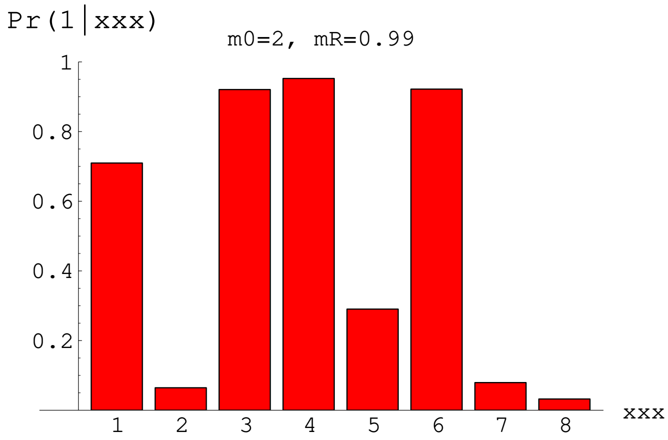

Figure 6: Conditional probability that the bar is not crowded, given its state in the previous *three* time steps, under the same conditions as Figure 5. The extreme probabilities suggest that exploitable inefficiencies exist in the system.

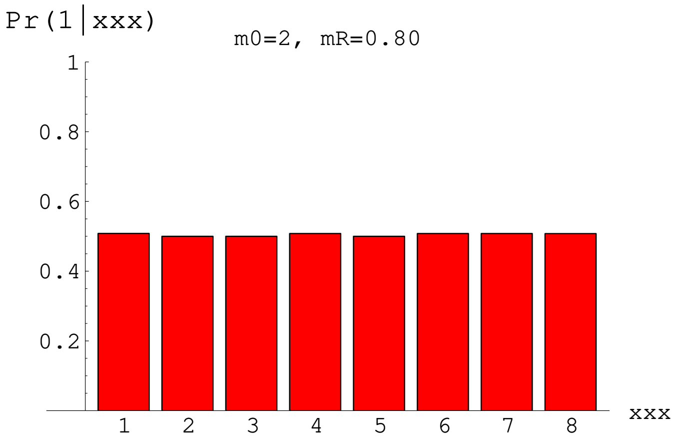

Figure 7: Conditional probability that the bar is not crowded, given its state in the previous three time steps. The system consists of 20% frequentist agents and 80% lookup agents with $m0 = 2$. The frequentist agents have exhausted all inefficiencies that are observable in a three time-step window.

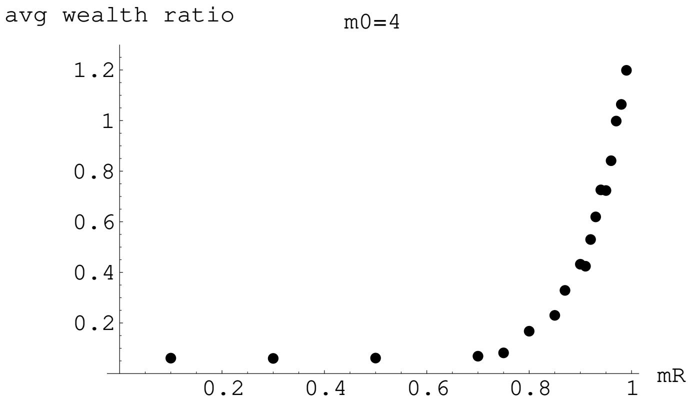

Figure 8: Average wealth ratio versus the fraction of lookup agents for $m0 = 4$.

As we see in Figure 8, the drop off in performance as the fraction of frequentists grows is much faster than in Figure 3. This drop off is a fairly smooth exponential, as can be seen in Figure 9. The relationship between wealth ratio and $m\mathrm{R}$ is roughly linear in this log-log plot. A similar relationship is evident for other values of $m0$.

Figure 10 plots the standard deviation of attendance versus $m\mathrm{R}$ under the same conditions as Figures 8 and 9. Although it is hard to tell conclusively, it appears that there is again a minimum in standard deviation at about the value of $m\mathrm{R}$ where the wealth ratio equals one. This phenomena also is observable for other values of $m0 < 4$.

Figures 11 and 12 show that, as $m0$ increases beyond 4, a lone frequentist agent fares about equally as well as lookup agents. At the same time, populations dominated by frequentists perform better and better. When $m0 = 10$, the wealth ratio is close to one for all values of $m\mathrm{R}$. I believe that this is simply due to the fact that, as memory lengths grow, more combinations of outcomes has never been observed before. When this happens, as described in Section 2, frequentist agents behave just like lookup agents.

4 Conclusions and Future Possibilities

Savit, Manuca, Li, and Riolo [1] examined the behavior of populations of *lookup* agents competing in a so-called *minority game*, in which rewards are

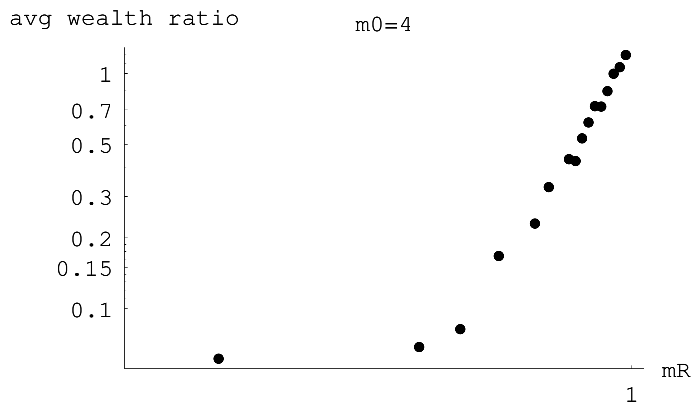

Figure 9: Log-log plot of Figure 8.

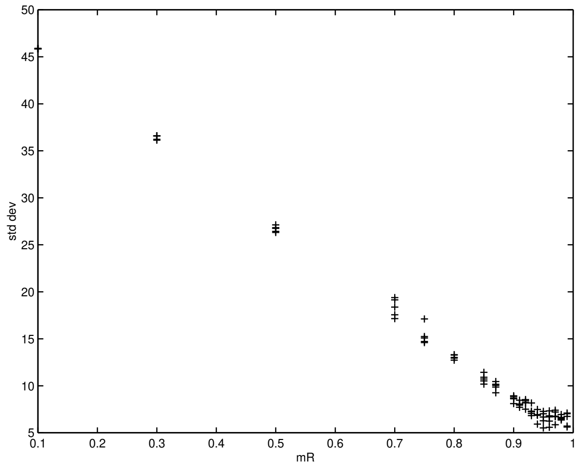

Figure 10: Standard deviation of attendance versus the fraction of lookup agents for $m0 = 4$. Again, the minimum in standard deviation appears to occur approximately at the point where the wealth ratio equals one.

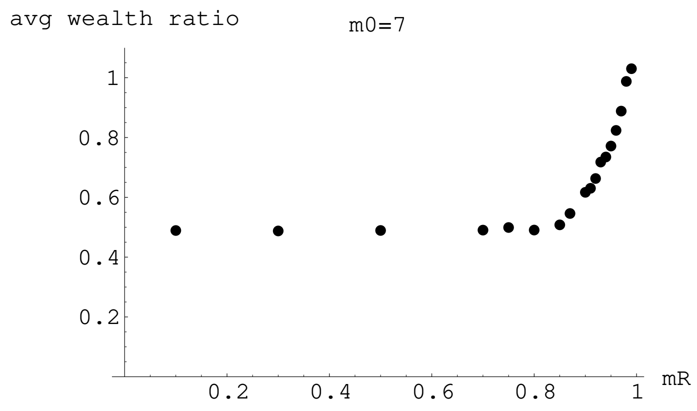

Figure 11: Average wealth ratio versus the fraction of lookup agents for $m0 = 7$.

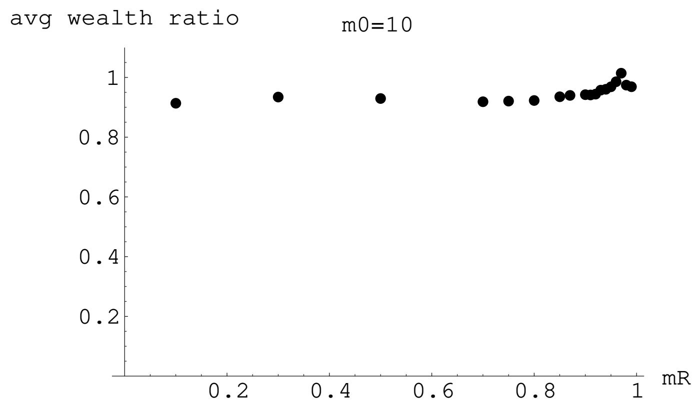

Figure 12: Average wealth ratio versus the fraction of lookup agents for $m0 = 10$.

doled out to those who choose the least popular of two options. Such a reward structure is common in many realistic scenarios, from Internet traffic to bar attendance.

I added a new type of agent to the model, called a *frequentist*. These agents simply choose the option that is most likely to be in the minority, according to past statistics. Frequentist agents are able to exploit some inefficiencies, when they exist. In fact, the success of these agents is closely correlated with the inefficiency of the system. Frequentist agents work best alone: when too many coexist, their herd-like behavior hurts the entire group. A mixture of both frequentist and lookup agents is most efficient when each group earns roughly the same average reward. The "optimal" mixture percentage is about 15% frequentists for $m0 = 1$ and decreases rapidly as $m0$ grows.

Frequentists clearly can take advantage of some inefficiency when $m0$ is small. However, even at the optimal mixture percentages, the standard deviation of attendance for $m0 = 1$ is greater than for $m0 = 6$. Thus some inefficiency still remains. It would be interesting to determine what other types of agents can be successful in this regime. One simple change would have frequentist agents choose randomly, weighted according to the conditional probabilities. Another idea is to introduce an *anti-frequentist* that always chooses the opposite of what the current conditional probabilities suggest. This type of agent could exploit the herding nature of frequentist agents, but would themselves be herders if their number grew too large.

It would be intriguing to investigate the evolutionary dynamics of minority competition. In a system of lookup and frequentist agents, would the population evolve to consist of the optimal mixture of agents? This report provides some evidence in the affirmative, since a less than optimal fraction of frequentists are rewarded as a whole, and a larger fraction are punished, relative to lookup agents. Another interesting evolutionary setting to consider would be a mixture of frequentist and anti-frequentist agents, given their symbiotic nature.

Acknowledgments

Thanks to Brian Mulloy for his part in writing the minority game and *MatLab* code, and for his clear instructions on setting up and running the programs. Thanks to all other developers of these programs, and of SWARM and SWARM-related utilities, including `drone`. Thanks to Rick Riolo for allowing me a little extra time (1 year!) to complete this project, and for helping me to get the simulation up and running.

References

[1] Robert Savit, Radu Manuca, Yi Li, and Rick Riolo. The dynamics of minority competition. University of Michigan Program for the Study of Complex Systems preprint no. 98-12-002, 1998.