Title
Refining the Breeding of Hybrid Strategies

Authors
Marks, Albert E
Midgley, David F
Cooper, Lee G

Publication Date
2023-03-31

Refining the breeding of hybrid strategies

by
Robert E. Marks[1],
David F. Midgley[1], and
Lee G. Cooper[2]

Paper presented to the Third International Conference on Computing in Economics and Finance, Stanford, CA, June 30-July 2, 1997.

---

1 Australian Gradute School of Management, University of New South Wales, Sydney, Australia.
2 Anderson Graduate School of Management, UCLA.

Introduction

We are interested in the effects of asymmetic market response on the competitive actions of managers. In particular, how the managers responsible for the marketing of brands in an oligopoly will compete in a multiperiod game where each brand elicits a different response from consumers and each firm faces different costs. This is the typical situation faced by brand managers for many of the product categories sold in supermarkets. Through a substantial literature built on the analysis of scanner data much is known about consumer response in these situations and methods for modeling this response are well established (Cooper and Nakanishi 1988). However, far less is known about the competitive actions of managers and there have been few attempts to model the repeated game that is evident in these product categories.

In earlier work (Midgley, Marks and Cooper 1997), we showed how the actions (prices and promotional activities) of each brand manager in an oligopoly could be modelled as an outcome of several finite automata playing in a repeated game. Using the Axelrod/Forrest representation (Axelrod 1987) of the artificially adaptive economic agents as bit strings, we used a genetic algorithm and a market-response model to coevolve better artificial agents for the three major brands of canned, ground coffee in a regional U.S. market.

These agents were specified as partitioning the previous actions of competitors into a small number of bands and selecting an action that would be profitable for them in the next period of the game from a similarly small number of available actions. This process can be thought of as defining perceptions of possible states of the market and developing mappings from these perceptions to an action for the next period. Depending on how many previous states of the market are defined, and how many actions are available to an agent, these mappings can be more or less complex.

In our earlier work we exogenously determined both perceptions and actions from an analysis of historical data and chose a simple specification for our agents. These agents were limited to a set of four actions and their perceptions restricted to an equivalent four bands for a single previous period of the game. This resulted in agents whose mapping from perceptions to actions could be specified by 134 bits. The genetic algorithm was then used to evolve mappings that maximised profits over a number of multi-period games in which various agents were pitted against two of their competitors (simulating the three main brands in the chosen market). Profits were computed by reference to a market response model that estimates brand sales given the actions of the competing brands.

The best of these agents performed well both in these games and when a single agent was pitted against the historical actions of human brand managers. While the latter test is “unfair” -- there being no opportunity for the human managers to respond to the agent -- our best agents outperformed the managers of two of the main brands and came close to equalling the performance of the market leader. Given the simplicity of our agents this was in many ways a surprising result. Moreover, in developing these agents we learned that retail store policies and demand saturation were also important to the realism of our results. Store policies are important because these can act to constrain the frequency of temporary price reductions and promotional displays, and demand saturation is important because there is a limit to the amount of any product that can be consumed or stored in a given period. Both these constraints set the environment within which managers and agents can select their competitive actions.

However, while these results are encouraging we recognise that the agents and procedures we use are capable of futher development. There are two important limitations to our earlier work, the first methodological and the second concerning the sophistication of our agents.

First, the genetic algorithm (GA) used a single population of 25 artificial agents, scoring the profitability of each string differently, depending on which of the three brands the agent was designated as in a particular simulation game. This was done because the GA software available at the time only addressed the single population case. However, for in our situation where consumer response and costs differ by brand it would be more desirable to have a multi-population GA. There is also a secondary issue around the size of the population. Populations of size 25 are commonly recommended in the GA literature but appear to us quite small when compared with natural evolution.

Second, we only modeled the three main players in a market that has nine brands and we only allowed our agents four actions when human managers used a far greater number. These choices were made partly because of the early stage of our research but also because the high computing demands of GA applications made it difficult to complete more complex simualtions. In the intervening period computing power has increased and we have learned to make GA software more efficient. This permits more realistic representations of brand management to be modeled.

This paper reports the consequences of relaxing these restrictions. In particular, how the profit performance of the agents improves as we allow brand-specific responses with separate GA populations, as we coevolve four artificial agents instead of three, and as we use a set of eight possible actions instead of four. We also address the issue of population size and the potential for “genetic drift”.

Finally, we begin the task of understanding the learning inherent in the actions of human managers. While these managers use more actions than our agents the actions they use are only a subset of all possible actions. Just how they came to select this subset, and how optimal it is compared with other possibilities, are important research questions. By examining agents at different stages of their “evolution” and by using different sets of exogenously determined actions we can begin to address these issues. Perhaps more importantly, our results here identify a clearer development path for artificial agents.

The structure of the paper is as follows. We first describe some necessary improvements to our modeling procedures that facilitate more sophisticated agents, speed up optimization, assess the robustness of our results and allow multiple populations. These improvements concern demand saturation, agents that do not conform to store policies, Monte Carlo methods and rewriting the GA software. We then present the results from a series of experiments. Experiment 1 in which we examine how optimization can be accelerated by filtering bit strings and Experiment 2 in which we examine the impact of multiple populations.. After these methodological considerations we turn to specifying more complex agents. In Experiment 4 we increase the number of players from 3 to 4 and in Experiment 5 we increase the number of actions from 4 to 8. Finally, we discuss Experiments 5 on co-evolution, population size and genetic drift, and Experiment 6 onthe nature of managerial learning. We conclude by identifying issues of concern and areas for future research.

Methodology

We have made a number of improvements to the basic methodology described in Midgley, Marks and Cooper (1997). These improvements relate to the demand saturation model, store policy constraints, Monte Carlo methods and multiple population genetic algorithms.

Demand saturation

Saturation of demand occurs when the one-week market response model allows sales volume to greatly exceed the capacity of the market to buy and consume the weekly amounts of coffee. Previously, we had modelled this by keeping a running seven-week average of total sales, and pro-rating each brand's sales to keep total weekly sales within the historical long-run average. This meant that for the first seven weeks, we did not constrain the sales, in order to derive the degree of over-saturation of the market on the eighth week. With our earlier four action results we never observed saturation within the first seven weeks -- partly because the four actions chosen were not extreme. These actions were based on common pricing strategies employed by human managers and presumably managers learn that over-saturating the market is futile (e.g. when demand is inelastic to continue lowering prices simply lowers profits).

However, when we moved to eight actions we included more aggressive pricing strategies and in particular the lowest prices observed in the market. While these had occurred historically they were not common and their repeated use within the simulation would lead to saturation unless constrained. Our eight action artificial agents quickly learned to exploit the loophole in our constraints -- essentially learning to price low and build up large profits in the unconstrained first seven weeks. Our solution was to consider the first seven weeks as a calibration period, and not count the profits in the agent's performance. This eliminates the incentive for the agents to over-saturate the market. While this is a relatively minor improvement to our modeling it does illustrate the importance of the correct environment in producing realistic agents. This theme continues in our next improvement.

Store policies

Previously, we imposed two exogenous requirements on the actions of agents: (a) they could not price “low” two weeks in a row, and (b) only one brand per week could price “low”. The definition of “low”depended on the particular brand's actual pricing behaviour in the historical data. The reason for these restrictions on the agents' behaviour is that, in their absence, all agents price at the lowest of the four prices each was allowed. Given that demand is not over-saturated there is an incentive for agents and brand manangers to price low -- provided their competitors are not also pricing low. But it is difficult for the agents to predict competitive behaviour in an unrestricted simulation and they become locked into continuous low pricing and low profit strategies. This is not the behaviour observed in the historical data, since the chain store imposed similar restrictions on the behaviour of competing manangers. These store policies appear to organise price promotions over time -- they may also serve an economic purpose for the store itself.

As is common we commenced our genetic optimization from populations of random bit strings, many of which violated these store requirements. With only three players and four actions per player, despite a large proportion of initial agents that did not obey the two requirements, these agents were scored sufficiently badly by our constraint penalties that the GA soon learnt to avoid the genotypes which gave rise to such illegal behaviour. With the refinements of eight actions, or four strategic brands as players, the bit strings are very much longer, which, we found, meant that such learning took far longer. The population would start off with 100% illegal strings, and the emergence of legal strings would only occur through the mutation operator, not through the recombination of crossover. Given that the added complexity of the simulations meant that the simulations were taking much longer (up to 36 hours on a DEC Alpha station 5,000/50), we have been prescreening the strings in the initial population to cull illegal genotypes. We term this procedure “filtering” and it greatly accelerates the simulations. The results of this improvement are discussed under Experiment 1 below.

Monte Carlo simulations

In order to assess the robustness and generalizability of our results we now perform Monte Carlo simulations for certain experiments. These are easily achieved as the GA needs a random number seed to generate its initial population of strings. We simply start each Monte Carlo simulation with a different seed. However, it should be noted that Monte Carlo methods have been greatly facilitated by the steps we have taken to speed up simulations and increases in computing power. Previously running such repeated simulations would have taken an excessive amount of time.

Multiple population simulations

The earlier work -- in common with all published use of GAs in economics -- relied on a single population of strings (agents): we used different payoff matrices, depending on which brand a string was designated as during the competitive simulations. This meant that the profitability (fitness) of a string varied according to situation and hence the GA itself was subjected to greater variance in its search for higher-performing strings (genotypes). So long as the profit surfaces of the three strategic brands had similar geometries (similar points of inflexion, local maxima and minima, etc.), then the variance should not have created much difficulty for the GA: the single population would behave as though the payoffs were noisy, but not pathological, in the way that opposite slopes of profit functions might induce.

Nonetheless, with heterogenous players (brands), and the possibility of fitness surfaces which exhibited opposite slopes, we determined to develop a simulation with multiple populations. This would allow us to reflect the differing responses brands invoke from consumers, and the differing costs they face, more accurately in our simulations. Table 1 shows just some of the parameters of the consumer response model and demonstrates the heterogenity evident in this market3.

3 The response model comprises a market share sub-model, a category volume sub-model and a profitability sub-model. In all it conatins over 180 parameters and explains the historical volume and share levels very well.

Table 1. Some parameters of the response model.

Brand	**Price elasticities**	**Variable costs per lb**
Folgers	-4.4	$1.39
Maxwell House	-3.9	$1.32
Chock Full O'Nuts	-4.7	$1.19

Source: Cooper and Nakanishi 1988.

Developing a multiple population GA was not a trivial exercise, since we have three different players competing at the same time in each week, and necessitated getting under the hood of our GA engine, the UCSD version of GENESIS. The net effect of Hossein Shirazi's recoding is that the simulations are much faster: using all information generated in each interaction of players: with three populations of players, the new code is almost as fast per trial as the old code was with a single population.

Because of the stochastic nature of the simulations, we have performed some Monte Carlo simulations to compare the convergence and outcomes (in terms of best and average phenotypic behaviour) of moving from a single population with three brands to three separate populations, one per brand. These results are described under Experiment 2 below.

Having made these methodological improvements we now tunr to developing more sophisticated agents.

Four strategic players

A natural extension of the earlier work has been to increase the number of strategic players. With the new multi-population code, it has been relatively easy to extend the simulations to a fourth player, at some cost in complexity of the bit-string chromosomes (which grow in length from 134 bits to 520 bits for one-week memory and four possible actions per player, including bits for the phantom memory of the first week's play[4]).

The increased complexity of the bit strings underlines the advantage of pre-screening to eliminate illegal strings before play occurs: otherwise the emergence of a full population of 25 legal strings occurs after a much longer period.

The choice of the fourth player is not obvious. The fourth ranked brand by market share -- *Master Blend* -- belongs to the same company as one of the three main brands and was relatively inactive in terms of price promotions for the period for which we have data. Instead we chose a smaller player -- *Hills Bros*. -- that is known to be a strong competitor for one of the major brands. This choice seemed to us to add a more

---

4 . 4 actions requires 2 bits per action; 4 actions, 4 players, and 1-week memory implies 44 = 256 possible states; phantom memory is 4 x 2 = 8 bits. So 2 x 256+8 = 520 bits per string.

interesting dimension to the competitive game. The results of this game are reported under Experiment 3 below.

Eight actions per player

The use of four actions per player was an exogenously determined number in the earlier work. Choice of the four actions meant that the artificial agents were more constrained than their historical counterparts had been, but none the less the artificially bred agents were able to outperform the historical agents, as described in the earlier work.. By choosing four actions which had frequently been used in the historical data, we denied our artificial agents the opportunity to learn what the historical brand managers must have learnt through experience and corporate memory: the boundaries of extreme behaviour. By increasing the number of actions to eight, we hoped to give our agents the opportunity to demonstrate that the four actions used earlier were robust, and that our assumption of a mature oligopoly were correct, at least in terms of the combinations of prices and other marketing actions encountered.

The eight actions meant an increase in the complexity of the simulations: retaining one-week memory and three players meant that the string length rose from 134 bits in the earlier work to 21,545 bits, an increase of more than an order of magnitude. As before we identified the apprpriate actions for each brand by a process of cluster analysis and visual inspection of historical data. We also used historical data from three chains rather than simply the one chain that is the focus for our simulations. This provided us with a broad sample of actions. Six actions for each brand were chosen by this process -- to which we added the highest price and the deepest price promotion observed for the brand in the period. The final eight actions allow the agent representing a brand manager to have a much more varied and wider ranging set of actions than in our earlier work. This can be seen in Table 2 which presents the previous and new sets of actions. The results of simulations based on eight actions are discussed under Experiment 4 below.

Table 2. Four and eight action sets.

	Folgers			Maxwell House			Chock Full O'Nuts		
Action	Price	Feature	Display	Price	Feature	Display	Price	Feature	Display
1	$1.87	95	69	$1.96	95	69	$1.89	100	77
2	$2.07	83	0	$2.33	83	0	$2.02	100	65
3	$2.38	0	0	$2.46	0	0	$2.29	0	0
4	$2.59	0	0	$2.53	0	0	$2.45	0	0
1	$1.62	67	67	$1.60	97	97	$1.64	0	0
2	$1.83	97	96	$1.87	94	91	$1.89	97	97
3	$1.96	0	0	$2.06	88	76	$1.89	98	29
4	$2.03	79	77	$2.33	79	0	$2.01	0	0
5	$2.04	85	0	$2.38	54	0	$2.02	97	62
6	$2.22	96	33	$2.52	0	0	$2.31	0	49
7	$2.57	0	0	$2.53	0	53	$2.33	0	0
8	$2.78	0	0	$2.59	0	13	$2.49	0	0

The numbers in the table represent the prices per pound of coffee and the percentage of stores in the chain "featuring" and/or "displaying" the brand. The shaded areas represent the actions defined as "low" for the purposes of implementing store policy. In the four action case this is simply the deepest price cut (which also always involves a feature and display). In the eight action case "low" prices are defined as prices below $2.20 (determined from analysis) combined with a majority of stores featuring the brand. "Featuring" involves an arrangement for the store to promote the brand -- for which they charge the manufacturer and which normally invokes a stronger response from consumers than a price cut alone. A featured low price is thus the type of promotion that is controlled by store policy. Note that one low price action for *Chock Full O'Nuts* is not classified this way because it has no associated feature and hence significantly less impact on the market.

Coevolution and genetic drift

The artificial agents learn through application of the evolutionary techniques of the GA. This is clear when the agents are solutions to a static problem, as has been the most usual application of GA techniques in, say, engineering. It is also the case that the first application of GAs in economics (Axelrod 1987) was static, even if stochastic: Axelrod and Forrest used GAs against a non- evolving but mixed-strategy niche of algorithms derived from the early Prisoner's Dilemma tournaments (Axelrod 1984). But Marks

(1989, 1992) and others following have bred artificial agents against each other, a process that Marks called bootstrapping and biologists term coevolution5.

Against a static environment, progress of the artificial agents is readily revealed by their improving fitness scores, but against a dynamic environment comprised of like artificial agents scores may not rise from generation to generation6.

In our earlier work we attempted to show the greater competence of our artificial agents by pitting them against the historical actions of managers, but some criticism has been made that this overstates the skills of the artificial agents and understates the skills of the managers, who have no opportunity to respond to the actions of the artificial agent: their plays are given.

Here we attempt to show how the artificial agents have learnt by taking agents after 2,500 trials (100 generations) and playing them against not the frozen moves of their historical opponents, but the agents after only 200 trials (8 generations): a process we have termed pitting "sophisticated" agents against "naive" agents. The results of these competitions are given under Experiment 5 below.

Managerial learning

In our final experiment -- Experiment 6 -- we investigate an analogous learning process -- that of the brand managers. We did this by contrasting the performance and behaviour of agents using the eight actions derived from analysis of historical data with agents using eight randomly determined actions. The actions we see in the historical data are presumably the end result of several decades of competition and managerial learning. By contrasting agents using these actions with those using random actions we attempt an assessment of the impact and consequences of this learning.

Results

Experiment 1 -- filtering for illegal strings in the first generation

An initial population of strings chosen at random will exhibit illegal behaviour: pricing low two weeks in succession. Because it is easy to screen the genotype (the bit string

---

5 Nothing is absolutely predictable about the direction of coevolution. How an interaction coevolves depends not only on the current genetic makeup of the species involved but also on new mutations that arise, the population characteristics of each species, and the community context in which the interaction takes place. Under some ecological conditions, an antagonistic interaction between two species can coevolve to enhance the antagonism; the species build up methods of defense and attack, much like an evolutionary arms race. Under other ecological conditions, however, the antagonism may be lessened, resulting in reduced antagonism or cooperation or mutual dependence. *Encyclopaedia Britannica*, CD97, Propaedia: The biosphere and concepts of ecology, the coevolutionary "arms race" versus reduced antagonism.

6 In a recent book, Stephen Jay Gould has argued that although baseball skills have improved tremendously this century, this is not revealed in the scores of professional baseball clubs, since these scores are relative not absolute measures of performance (Gould 1996). Incidentally, in first-class cricket, there has been improvement in scores (NYRB 1996).

structure) for this behaviour, and because (as we shall see) starting the simulation with a random, unfiltered initial population slows learning down, we have been eliminating illegal strings from the initial population.

Because of the stochastic nature of the processes, we have used a Monte Carlo technique, in which we did 50 runs of the random-strings model, and 50 runs of the legal-strings model. These models are our original three brand, four action specification. For each run, we saved two statistics every 100 trials (4 generations): the mean of the best-ever strings' scores and the mean of all strings' scores. These are plotted in Figures 1 and 2.

[Figures 1 and 2 about here]

Figure 1 shows the performance of Folgers in the simulation with random initial strings. The higher solid line is the mean of the best-ever strings (it thus can never have a negative slope). The lower solid line is the mean of all strings' scores (which can slope down). Because many strings exhibit illegal behaviour (which in this case is easily identified from examination of their structures or genotypes), the lines do not always start at the left of the graph: only when the illegal strings have disappeared from the category being plotted will the plot commence.

Note that the mean of all strings converges towards the mean of the best-ever strings after 1,800 trials (72 generations), although never completely. The mean of all strings is just over $600 per week for Folgers.

Figure 2 shows the same two statistics in the case where the initial population is filtered so that all individuals are legal. Comparison of the two figures shows that there is much less to be learnt by the strings. Since with filtering there are no illegals from the first trials, all statistics start at the beginning and convergence occurs sooner, after 500 trials (20 generations) or so. Note that since saturation is not a problem with the four action model, we have included the profits of the first seven weeks trading for Figures 1 and 2. It is clear that filtering speeds up learning and convergence considerably.

Experiment 2 -- multi-population simulations

As above, because of the stochastic nature of the processes, we have used a Monte Carlo technique, in which we did 50 runs of the single-population model, and 50 runs of the three-population model. Again this is for the three brand, four action model. For each run, we saved two statistics every 100 trials (4 generations): the mean of the best-ever strings' scores and the mean of all strings' scores. These are plotted in Figures 2, 3, 4, and 5.

We have already seen Figure 2: it shows the results for *Folgers* of the 50 simulations from different but legal initial generations, in which a single population of strings did yeoman service as the supply of genotypes for all three competing brands.

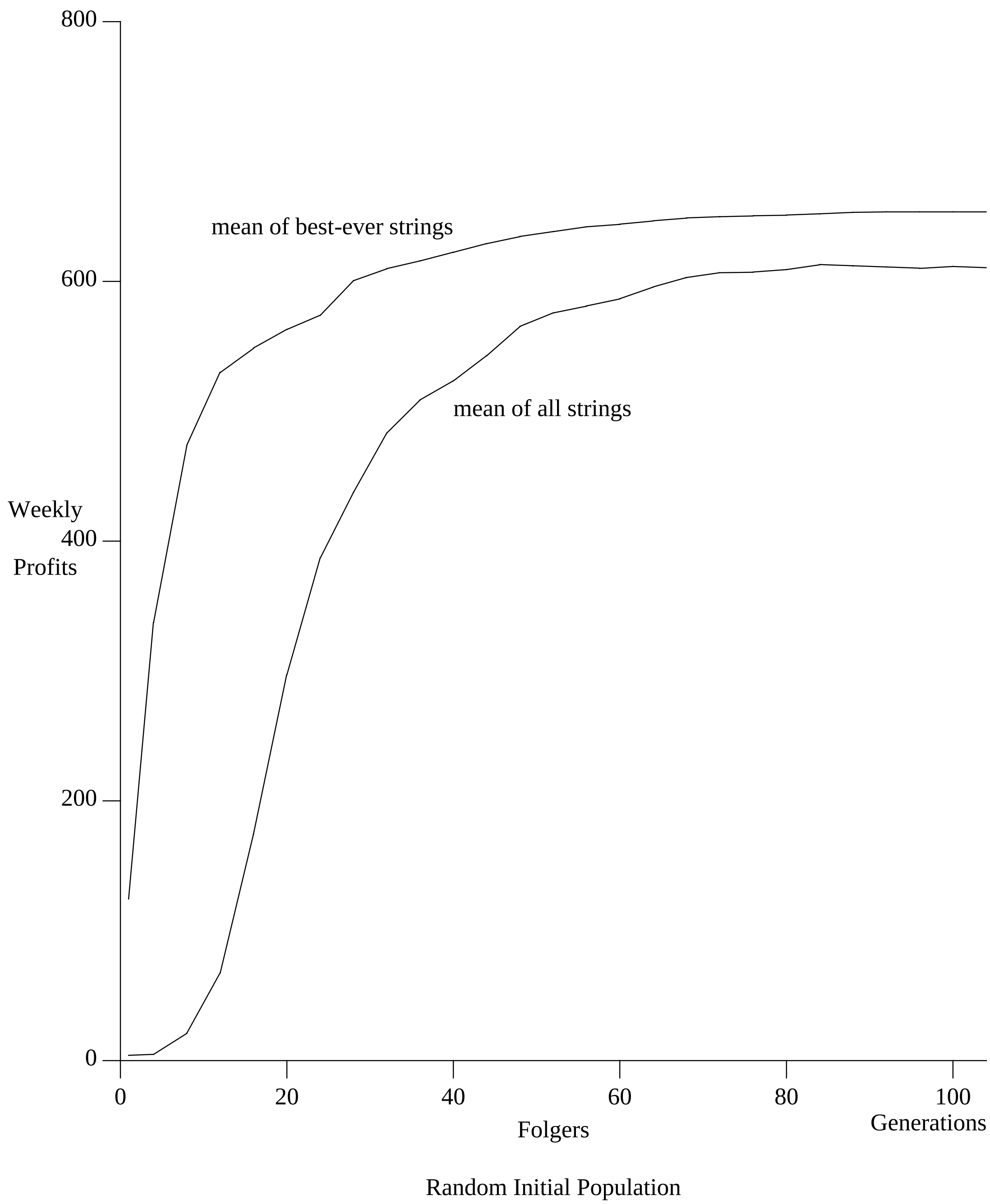

Figure 1: Learning with Random Initial Population

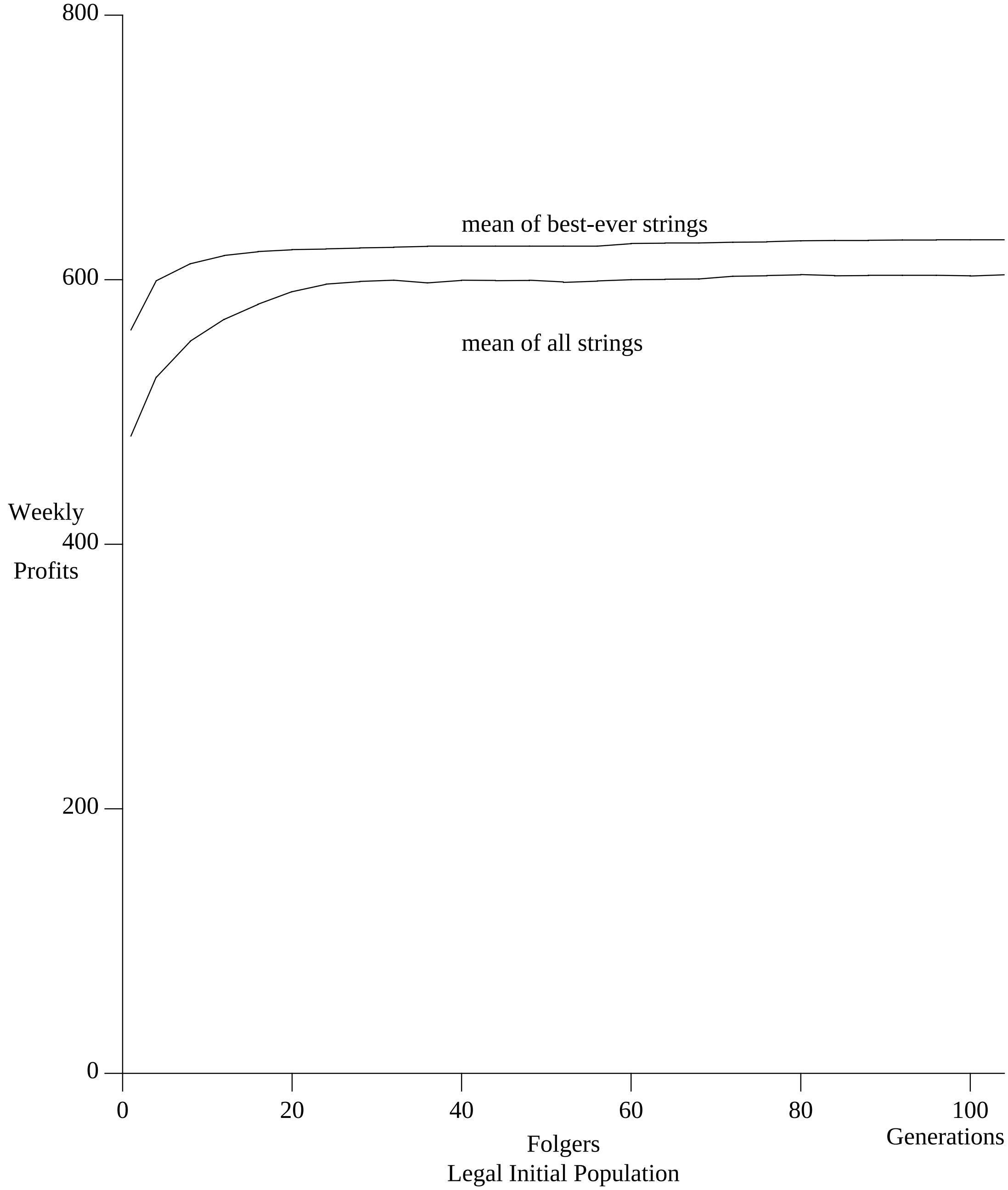

Figure 2: Learning with Filtering: Legal Initial Population

It includes the profits from the first seven weeks. The other two brands would have similar plots.

[Figures 3, 4 and 5 about here]

Figures 3, 4, and 5 are the results of the Monte Carlo simulations when using separate populations for the three strategic brands. The same statistics are plotted, and we have filtered the initial generation so that only legal strings begin. They also include profits from the first seven weeks.

Comparing Figures 2 and 3, we see that the three- population model results in more satisfactory performance in three characteristics. First, with the three-population model there is faster convergence after 500 trials (20 generations), compared to 600 trials (24 generations) with the single-population model. Second, there is higher mean performance of both best-ever strings and all strings: the mean string performance is almost $700 per week, compared to less than $600 per week; and the mean performance of the best-ever strings is higher with the three-population model. Third, as we might expect, with a separate population for each player, and hence a less noisy environment for the GA, comparison of Figure 2 with Figure 3 shows that the mean performance across all strings in the three-population model is not just higher, but is closer to the mean performance of the best-ever strings. Figures 4 and 5 show the performance of the other two brands (*Maxwell House* and *Chock Full O Nuts*) under the three-population model. We see the early convergence and closeness of the performance of the mean string and the best-ever string seen in Figure 3 and discussed above. It is clear that moving from simulation with a single-population model to a three-population model results in better performance on several dimensions.

Experiment 3 -- four strategic players

The results of expanding from three to four strategic players -- by adding a niche player in *Hills Bros*. -- are shown in Figure 6. Here we use our four action specification. These results are as we might expect in that the profits of the three main players are reduced quite significantly, even though *Hills Bros* itself generates quite small profit levels. The main reason for this reduction is that *Hills Bros* is a frequent price promoter and takes up promotional opportunities that would otherwise have gone to one of the main brands. With the store policies set the way they are there is only one opportunity for a featured price promition each week. Obviously the more brands competing for these promotions the fewer opportunities any one brand is going to have. Moreover, as these price promotions usually generate significant profits the overall profitability of the brand is consequently reduced.

[Figure 6 about here]

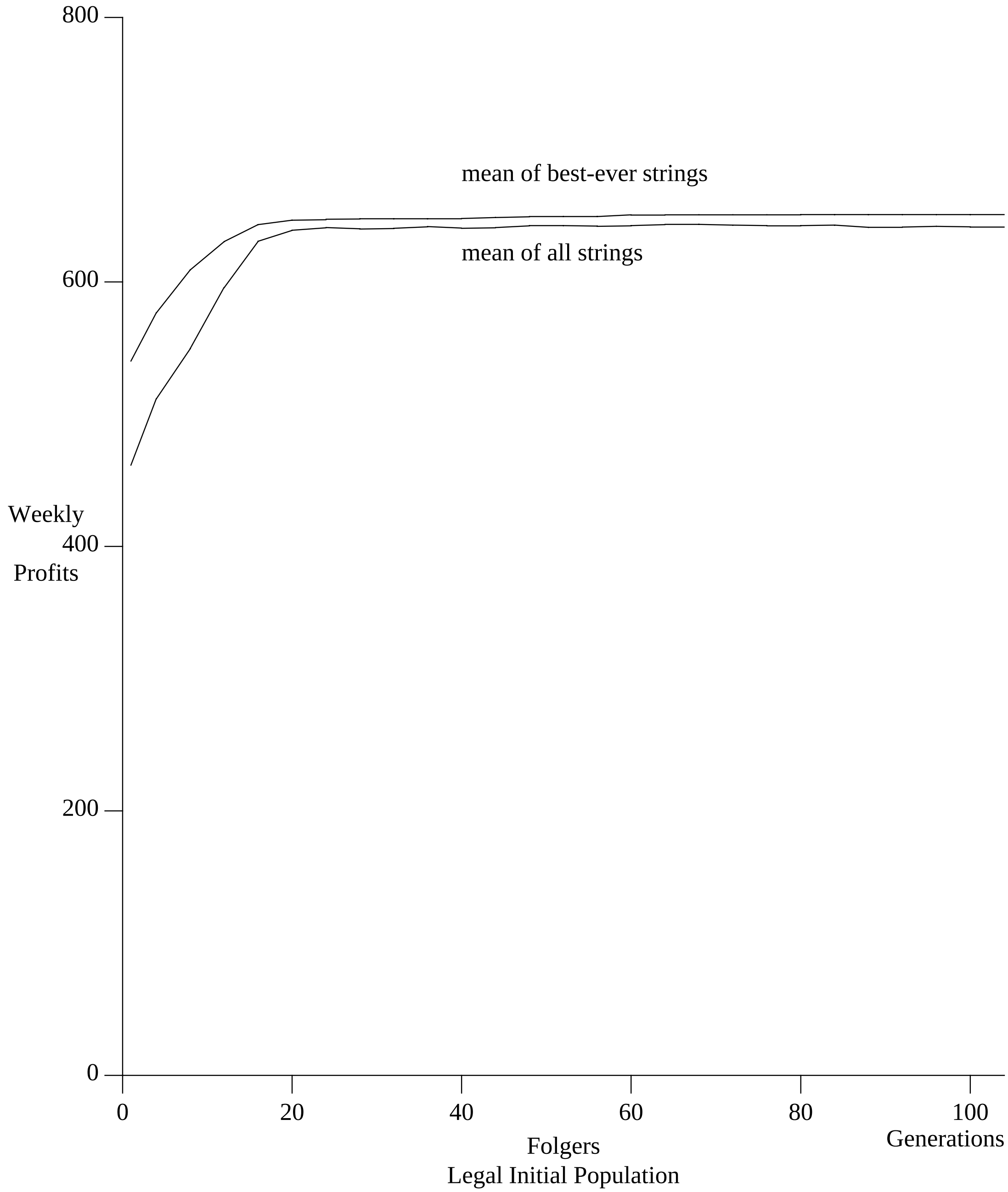

Figure 3: Three-Population Simulation, Folgers

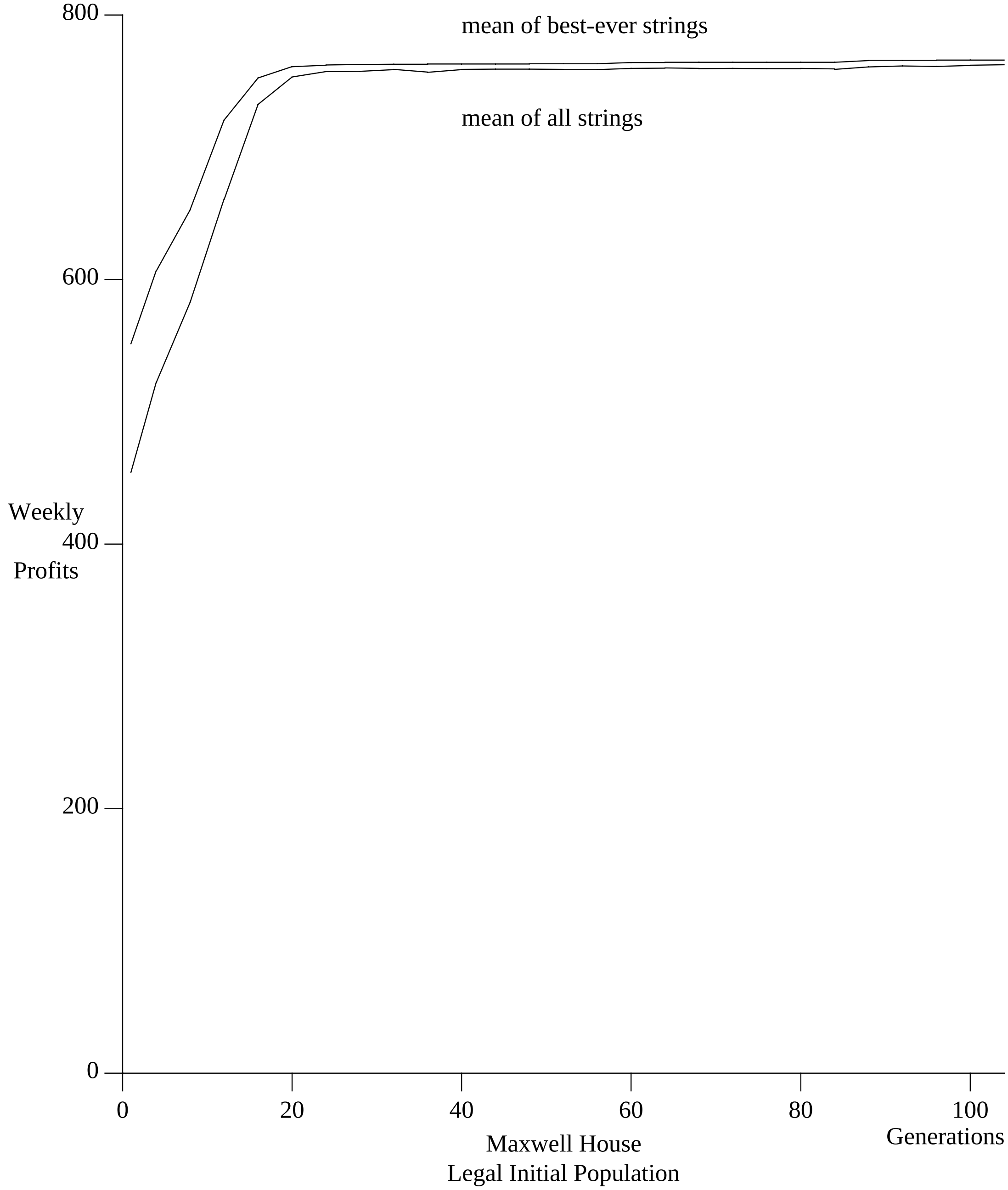

Figure 4: Three-Population Simulation, Maxwell House

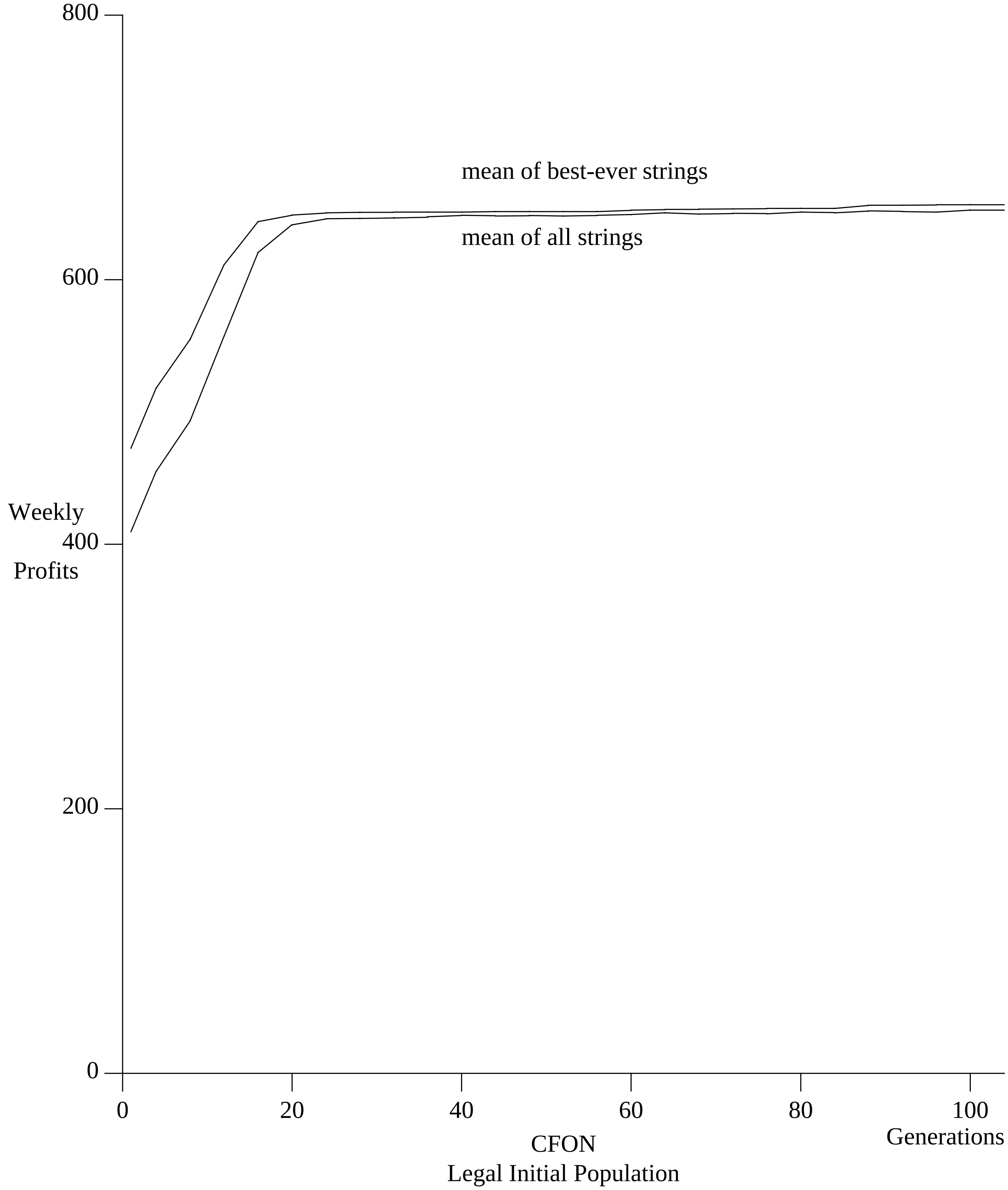

Figure 5: Three-Population Simulation, CFON

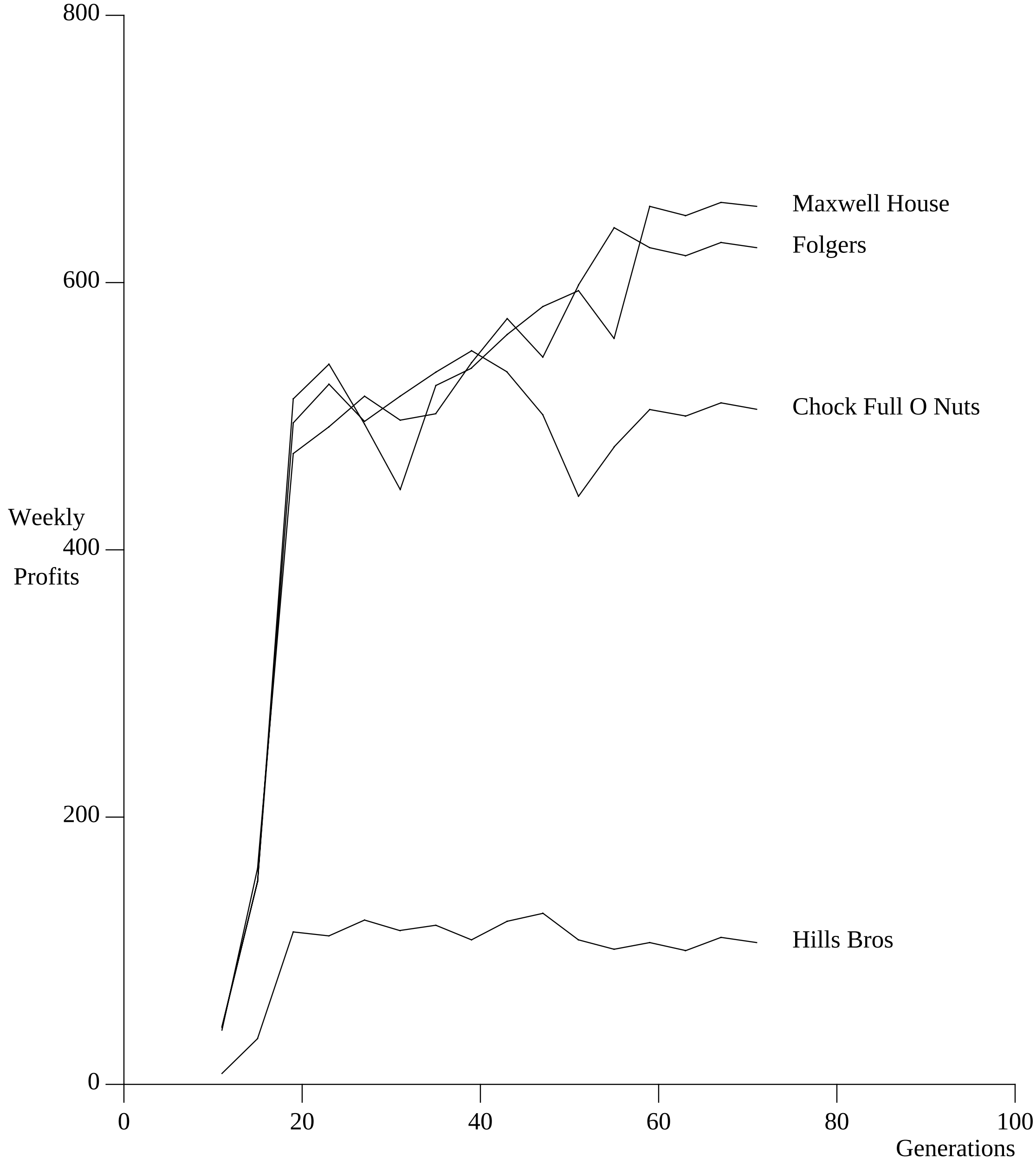

Figure 6: Four Strategic Players, Means of All Strings.

Experiment 4 -- eight possible actions per player

For this experiment we revert to three players and use the actions shown in Table 2. As before 50 Monte Carlo simulations were run to assess the stability and generalisability of the results. As the focus of the experiment is on the increase in the number of actions we captured not simply just the profits of the strings but also the frequency with which they were using each action. Early in the evolution of the GA the strings tend to use each of the eight actions with a roughly equal frequency. Table 3 shows the frequencies with which the actions are used over the first 100 trials (4 generations) for one of our simulations. While there is some preference for the highest price (Action 8) the remaining actions are evenly distributed.

Table 3. Frequency of actions over the first four generations

	Actions (percent)								
	Low price							**High**	
	Price								
Brand	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Total**
Folgers	8	7	11	8	6	13	11	36	100
Maxwell House	6	7	6	15	12	13	12	29	100
Chock Full O'Nuts	11	7	6	13	7	13	12	31	100

However, as the strings evolve they typically begin to favour some actions at the expense of others. In other words they learn which actions maximise profits given the various constraints and the behaviour of their competitors. Over the 50 Monte Carlo simulations different patterns of actions can emerge as populations of agents evolve from different starting points. We have used cluster analysis to identify the major patterns in our 50 simulations and it turns out that three patterns of competitive interaction account for 44 of the 50 simulations7. These three patterns of interaction are shown in Table 4 and are derived from action frequencies during the 100 generation (2,500th trial). As such they are based on 31,250 competitive interactions per agent and for one Monte Carlo simulation. Table 4 is based on the best agent observed in the 100th generation for each simulation. However, by the 2,500th trial the population within a simulation is typically very homogenous and essentially similar frequencies would be obtained by taking any other agent.

In the Table we indicate the actions that invoke the store policy constraint ("low") by shading and to simplify the data we indicate the dominant actions in **bold**. It can be seen that by the 100th generation the agents are using far fewer than eight

7 We cluster analysis the action frequencies of all three brands jointly because the issue is one of identifying the patterns of interaction between the brands as well as the action frequencies for each brand.

actions. In fact the two or three bolded actions for each brand account for between 63% and 92% of all actions (average of 77%). The agents have learned the two or three actions that work best for them.

Table 4. Patterns of competitive interaction at the 100th generation

	Actions Low price High Price								
Pattern 1, 26 simulations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Profit**
Folgers	13	4	7	**20**	1	**24**	3	**25**	$881
Maxwell House	2	2	4	**43**	8	**20**	6	12	$880
Chock Full O'Nuts	2	**26**	1	12	2	9	8	**37**	$646
Pattern 2, 11 simulations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Profit**
Folgers	**29**	8	4	5	0	**46**	1	6	$897
Maxwell House	0	0	0	**92**	1	1	1	3	$1085
Chock Full O'Nuts	0	**40**	0	4	0	10	2	**42**	$735
Pattern 3, 7 simulations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Profit**
Folgers	10	0	**27**	**28**	0	**21**	2	9	$931
Maxwell House	0	2	5	**49**	0	0	1	**40**	$1120
Chock Full O'Nuts	3	**31**	0	8	4	1	0	**49**	$622

However, these learned actions vary significantly between brands and across patterns. *Chock Full O'Nuts* uses a similar strategy across all three competitive patterns -- in essence "pulsing" between a high price and a featured low price. In Pattern 2 *Folgers* also pulses but in the other simulations it exhibits more diverse behaviour with three frequent actions. *Maxwell House* also has diverse behaviour. In Pattern 1 it oscillates between a featured medium price and various higher prices, in Pattern 2 it is essentially static on the featured medium price (which does not invoke the store policy constraint) and in Pattern 3 it pulses between this price and a high price. In all cases Maxwell House uses promotions that invoke the constraint very sparingly (and in Pattern 2 it avoids them altogether). Overall these results indicate the importance of store policy. They also indicate that different patterns of competitive interaction can appear over long periods of evolution -- which raises the question of the nature of this coevolution.

Experiment 5 -- coevolution: sophisticates against naives

As discussed above, we want to demonstrate that the natural selection of the GA is improving the performance of the artificial agents, which start off as filtered (and hence legal) but otherwise random strings. The improvements of the mean and extreme scores of the total population and the best-ever strings, as shown in Figures 1 through 5, are one measure. In this section we consider another: pitting the best-performing string after 2,500 trials or 100 generations (the 'sophisticate' agent) against all combinations of the other two brands after 100 trials or 4 generations (the 'naive' agents).

What should we expect? Since the sophisticates have had ten times as many generations to learn and change than have the naives, we should expect them to score better against the naives than against the sophisticates. The procedure followed was:

. After 200 trials (8 generations), identify the best individual strings (in the 8th generation) from each of the three (brand) populations.
. Play these three against each other for a 50-week repeated game, and note their average weekly profits.
. Allow the three populations to continue coevolving via the GA.
. After 2,500 trials (100 generations), match the best strings (in the 100th generation) from the three populations and note their average weekly profits.
. Replace the best *Folgers* string after 200 trials (8 generations) by the best *Folgers* string after 2,500 trials or 100 generations (e.g. replace the best naive string by the best sophisticated string).
. Play all combinations of three strategic brands, and consider string-by-string the change in average weekly profits with the sophisticated player and without the sophisticated player in one brand.
. Repeat steps 5 and 6 for the remaining two brands.

Because of the stochastic nature of the simulation, this process was repeated 50 times. The results are given in Table 5.

Table 5. Mean changes in average weekly profits with best sophisticate

Best Sophisticate	**Change in *Folgers***	**Change in *Maxwell House***	**Change in *Chock Full O'Nuts***
Folgers	-15.01	41.42	42.03
Maxwell House	2.03	-20.04	37.77
Chock Full O'Nuts	13.93	-28.99	82.34

We should expect positive diagonal elements of the table: that the sophisticate's performance would be better than that of the replaced naive string, but despite our expectation, only with one brand (*Chock Full O'Nuts*) does this occur; the other two brands show falls in the average weekly profits of between 2.5% and 5%.

What is going on? Consider a variation on the above where we identify not the best sophisticate but the worst sophisticate, and replace the best naive by the worst sophisticate. The results are given in Table 6.

Folgers' performance improves marginally, and *Chock Full O'Nuts'* improves significantly (about 20%), but *Maxwell House's* suffers a fall (about 15%). From these two tables we are left to conclude that, at least for two of our three brands, the sophisticated agents do not compete effectively with naive agents. We believe that this might be being caused by genetic drift over the 100 generations of coevolution.

Table 6. Mean changes in average weekly profits with worst sophisticate

Worst Sophisticate	**Change in *Folgers***	**Change in *Maxwell House***	**Change in *Chock Full O'Nuts***
Folgers	1.23	-11.82	54.94
Maxwell House	13.29	-94.44	20.22
Chock Full O'Nuts	54.94	-197.10	127.20

Genetic drift is the change in the gene pool of a small population that takes place strictly by chance. Genetic drift can result in genetic traits (genes, or patterns on our bit strings) being lost from a population or becoming widespread in a population without respect to the survival value of the alleles involved. As a random statistical effect, genetic drift can occur only in small, isolated populations in which the gene pool is small enough that chance events can change its makeup substantially. In larger populations, any specific allele is carried by so many individual strings that it is almost certain to be transmitted to the next generation by some of them, unless it scores very badly. (An example of the latter are the alleles that code for two weeks with low prices, and our penalty scoring soon eliminates these alleles.)

The magnitude of gene frequency changes due to genetic drift is inversely related to the effective size of the population (the number of individuals selected to produce offspring), since only parent strings transmit their genes to the following generation. In our case we have a very small population to begin with, only 25 individuals, and the number of parents each generation is smaller still, so that genetic drift over 100 generations (2,500 trials) may become significant. This could explain the poor showing that the *Folgers* and *Maxwell House* sophisticates make against the naives; the *Chock Full O Nuts* sophisticates consistently do better against the naives than against the sophisticates.

One way to further explore the notion that genetic drift is causing the anomaly would be to repeat the Monte Carlo simulations, but with population sizes much larger than the 25 individuals used above. If it is the case that the sophisticated strings have lost the alleles that led to higher weekly profits when the naives were competing, or, in the vernacular, that the sophisticates have become “flabby” in competition against similar sophisticates8, then the question arises of how to create pressure against the disappearance of this information as the strings coevolve. One method might be to continually inject the coevolving populations with strings from earlier generations, which might allow a sort of parallel evolution9. Another thought is that a diploid genotype (Goldberg & Smith 1987; Mitchell 1996) (which allows the emergence of “recessive genes”, such as that for blue-eyed human beings) might allow once and future genes of value to persist, even in an ever sophisticated population. Mitchell (1996, pp.21-27) discusses work by Hillis, in which he used a diploid representation: chromosomes in pairs, rather then the single chromosomes of haploid represention that we use here10.

In order to test the conjecture that the reason the sophisticates were not doing as well as the naives is genetic drift, we decided to increase the size of each population from 25 to 250. The tenfold increase in the population sizes means that (1) testing takes much longer, since each individual now has to compete against 250^2 combinations (instead of 25^2) and there are ten times more individuals. That is, a thousand-fold increase in the number of three-way interactions per generation. And (2) convergence is much slower: it's not just that each of the three populations takes longer to converge (as it would by itself) but that the slowness of the opponents to converge means that any convergence that does first occur is likely to be premature. With the three populations converging roughly at the same rate, *Folgers*, for instance, may find a small set of strings whose patterns mean good profits, against the current populations

---

8 An analogy in game theory is the issue of how a strategy will perform against an irrational strategy that moves off-equilibrium.

9 Compare this with Hillis (1992), who bred his artificial agents as hosts against coevolving problems, or parasites.

10 Since there are pairs of chromosomes, at each position on the first chromosome (or genotype string) the string segment (or allele) is compared with the corresponding string segment on the second chromosome. If they encode the same action (are homozygous), then that action is the contingent action for the corresponding state; if they encode different actions (are heterozygous), then one of the actions will be dominant, and the other recessive. The coding for the recessive action may survive the recombination of the chromosomes during the GA's evolution, and eventually may prove valuable. Recombination with diploid organisms is different from that of the simple haploid crossover we used earlier (Midgley et al. 1997). See Mitchell (1996, p.25). But homozygosity can protect crucial contengent actions: if a crucial contingent action is at a heterozygous position on its chromosome (genotypic string), then it can be lost under crossover, whereas such actions at homozygous positions cannot.

of *Maxwell House* and *Chock Full O'Nuts*. But when further convergence in, say, *Maxwell House* occurs, the *Folgers* strings may be further from optimal, which means that some alleles may become unconverged, and convergence may slightly reverse, until the *Folgers* population adapts to the new topology in strategy space, and evolutionary convergence continues. Meanwhile, *Chock Full O'Nuts* will have adapted to the earlier patterns of play of *Folgers* and *Maxwell House*, but this adaptation's performance will worsen in response to the new patterns of play of *Folgers* and *Maxwell House*. We can think of this as a spiralling towards a non-central node: closer and closer but then further away, and then closer, and so on. We have observed this in lengthy simulations we have run on populations of 250 (these simulations take weeks rather than days to complete).

Table 7 shows the changes in the averages across the 250 strings in each of the three populations when the best of the strings in early, naive population is replaced by the best from a later population. In Table 7 the later population occurs after 20,000 trials (80 generations) and the early population is after 500 trials (2 generations); in Table 8 the later population is that after 40,000 trials (160 generations), and the earlier one is still after 500 trials.

Table 7. Mean changes in average weekly profits with best sophisticate after 80 generations, population of 250

Best Sophisticate	**Change in *Folgers***	**Change in *Maxwell House***	**Change in *Chock Full O'Nuts***
Folgers	-28.00	-12.21	-50.22
Maxwell House	-86.36	-518.29	166.09
Chock Full O'Nuts	30.58	11.82	-191.32

Although the falls in the results for the sophisticates' populations (the diagonal figures) are surprising, since the sophisticates are doing worse than the naives they replaced, the falls in the off-diagonals, where they occur, show that the introduction of the sophisticate (one string in each of the three populations, seriatim) is reducing the performance of the naive populations, as we might expect.

Table 8. Mean changes in average weekly profits with best sophisticate after 160 generations, population of 250

Best Sophisticate	**Change in *Folgers***	**Change in *Maxwell House***	**Change in *Chock Full O'Nuts***
Folgers	-87.11	75.13	-55.66
Maxwell House	-101.87	-512.51	155.45
Chock Full O'Nuts	-63.19	-42.08	-23.77

After 20,000 trials (Table 7) *Folgers* sophisticate lowers the performance of the naive *Maxwell House* and *Chock Full O'Nuts*, but doesn't improve its own; after 40,000 trials (Table 8) *Folgers* sophisticate does worse: since its performance falls and *Maxwell House's* performance rises.

There is little change between the tables for the *Maxwell House* sophisticate: the performance of *Folgers* and *Chock Full O'Nuts* naive agents worsens slightly and while the performance of the sophisticate improves slightly it still does far worse than the naive agent it replaced.

For the *Chock Full O'Nuts* sophisticate, the performance after 20,000 trials is woeful (Table 7) its own performance falls, while the two naive populations' rise. After 40,000 trials, however, the two naive populations' performances worsen, which is to be expected, and its own performance, while it is still worse than the naive agent it replaced, has improved.

The lengthy simulations with populations of 250 per brand are continuing, and it may be that as learning and convergence continue, the performance of later sophisticates against the naives will be closer to that expected. With such relatively large populations, any genetic drift should be much reduced, which may rule out that as an explanation of why the sophisticates do not perform better themselves, without, however, providing a good explanation of the performances reported.

Experiment 6 -- managerial learning

Thus far we have looked at how agents learn in competitive simulations. However, our methodology also allows us to look at how managers may have learned over decades of actions in the cofffee market. In Experiment 4 we used eight actions that were developed from analysis of the historical data, that is from managers' historical actions. These eight actions are thus highly learned ones and exogenous to the agents that we evolve in our simulations. All the agents detailed above are similarly restricted to a repertoire of actions we specify for them. The question arises of how these agents might perform with a different repertoire of actions -- one developed without reference to the historical actions of managers. Such a repertoire is shown in Table 9 below. These actions are common to each brand and are developed from a random experimental design where price is stepped in 10 cent increments between $1.60 and $2.80. Feature and display are allowed to take on the value of 0 or 100 and can only apply to prices below $2.20.

Table 9. Eight randomly generated or 'naive' action sets.

	Folgers			**Maxwell House**			**Chock Full O'Nuts**		
Action	Price	Feature	Display	Price	Feature	Display	Price	Feature	Display
1	$1.60	100	100	$1.60	100	100	$1.60	100	100
2	$1.70	100	0	$1.70	100	0	$1.70	100	0
3	$1.80	100	0	$1.80	100	0	$1.80	100	0
4	$1.90	0	100	$1.90	0	100	$1.90	0	100
5	$2.00	0	0	$2.00	0	0	$2.00	0	0
6	$2.60	0	0	$2.60	0	0	$2.60	0	0
7	$2.70	0	0	$2.70	0	0	$2.70	0	0
8	$2.80	0	0	$2.80	0	0	$2.80	0	0

In Table 10 we show the patterns of competitive interaction that are observed across the 50 Monte Carlo simulations we ran with these eight random actions. In this case there are two major patterns that account for 31 out of 50 simulations. These are presented in the same format as Table 4 earlier. There are two striking facts we can note about Table 10. First, the profit levels are much higher than with the earlier, learned actions. Second, these profit levels are achieved because the agents are very sparing in their use of featured, low price promotions and maintain high prices throughout most interactions. This is certainly true for *Chock Full O'Nuts* which stays at high prices most of the time. *Folgers* "pulses" in both patterns but also maintains the highest prices more frequently than in earlier results. Unlike before *Maxwell House* has a definite high/low pulsing strategy but also stays at the highest price more frequently. In essence the level of competition is much lower with these random actions. While our analysis of these results is continuing, and we intend to try other sets of random actions, it would appear as though managers have "learned" to become overly competitive.

Table 10. Patterns of competitive interaction at the 100th generation for random actions

	Actions Low price High Price								
Pattern 1, 18 simulations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Profit**
Folgers	**24**	14	1	1	0	8	2	**47**	$2,009
Maxwell House	14	**24**	2	1	1	8	3	**45**	$2,801
Chock Full O'Nuts	4	1	1	0	2	6	**29**	**52**	$1,151
Pattern 2, 13 simulations	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Profit**
Folgers	5	**34**	0	4	0	4	3	**47**	$1,916
Maxwell House	**41**	2	0	2	0	3	2	**48**	$3,244
Chock Full O'Nuts	4	1	1	1	1	2	3	**85**	$1,192

Discussion

Discussion

In summarizing our findings and conclusions we have chosen to touch on three main areas; coevolution, the specification of agents and managerial behaviour. After discussing these we will briefly outline our future research plans.

Coevolution

The methodological contribution we make in this paper is the rewriting of the GA to coevolve multiple populations of agents. This is clearly a more valid approach for our application. Each brand evokes a different response from consumers, and faces different cost structures, and these facts should be reflected in the agent's mapping from perceptions to actions. While a single-population GA can produce a solution, it is a "one size fits all" solution, whereas a multiple-population GA allows a unique population of solutions for each brand. The empirical confirmation of these arguments is that the agents evolved from our three- and four-population simulations outperform the single-population agents of our earlier paper.

But coevolution of multiple populations is not without costs. It requires more computing time to reach convergence, and this is not solely because of the extra computations involved in the multiple populations. We also incur extra computing time because the multiple populations do not evolve in lock-step. Bit strings for one population can converge but then find their competitive environment (the other populations) has changed. They then need to evolve again to catch the moving target. Overall convergence is thus approached in a "spiraling" manner and takes much longer than with a single population.

We also have an unresolved problem with the results of coevolution -- our finding that "sophisticated" agents do not necessarily compete well with "naive" agents. In one sense this is not a problem. Agents perform well against other agents at the same level of sophistication. And nearly all of our agents perform well against the historical actions of human managers. But eventually we would like to breed agents that are robust to changes in competition -- for example, the introduction of a new "naive" player into a market. We therefore need bit strings which retain the capacity to compete with agents other than the ones they currently face.

Our initial supposition that the loss of this capacity was due to genetic drift in a small population is not borne out by our results. We have increased the population size tenfold but still obtain similar results to before -- sophisticates from a population of 250 do not perform well against naives from earlier in evolutionary process. Of course, 250 is still not a large population by the standards of the natural world, but our intuition is that increasing this population yet further will not solve the problem. We may need to look at other methods for retaining information in the bit strings -- such as a diploid mechanism rather than the haploid we have used this far.

Specification of agents

While we still have an unresolved issue with our basic genetic mechanism, the results presented here do allow more definite conclusions to be made about the form and structure of our agents. We have learned that our agents do not need as many actions as perhaps we thought or as perhaps managers use.

Our earlier work used four actions. At the time we thought that this was a reasonable number with which to start our work.. It is also true that the state of development of our methods and the available computing hardware had some impact on this choice -- it would have been very time consuming to investigate a larger number. Here advances in our methods (for example, “filtering”), and in the computing power available to us, have allowed us to investigate eight actions. But we find that the agents end up using far fewer than eight. In fact they most often end up using two or three with high frequency and the rest with very low frequencies. This raises the interesting question of whether we need to equip our agents with as many actions as we do in this paper.

Of course, we need to be careful in this assessment -- a low-frequency action may be important to the overall performance of the agent. And we have not yet investigated these matters in detail. But it does suggest that eight actions is a reasonable upper bound and we need not be concerned with equipping our agents with more than eight actions.

We have conceptualized the structure of our agents in three components: 1) a mechanism for perceiving the previous state of the market, 2) a mechanism for mapping from this state to future actions, and 3) a mechanism for determining an appropriate set of actions. Our earlier paper addressed simply the mapping mechanism using exogenously determined perceptions and actions (which were in fact equivalent). This paper goes some way to addressing the third component by suggesting an upper bound of eight. Our intuition is that the most productive areas for future research are perceptions, and methods for endogenising the development of perceptions and actions. We will discuss these subsequently.

Managerial behaviour

In most of our work we have exogenously determined the actions of the agents by analysis of the historical actions of managers. Since we also equivalence perceptions to actions these too are based on the historical data. In that sense our agents have “learned” from, or at least been informed by, what managers think are appropriate actions, and how managers react to the actions of their competitors, etc. The agents are thus performing well because the GA evolves effective mappings between perceptions and actions. And we have some evidence that these mappings are at least as effective as those of managers'.

In our last experiment we broke from this path -- by randomly drawing a set of actions for the agents. Our prior hypothesis was that agents with these random actions would not perform as well as those with exogenously determined actions; in other words, that the managerial learning incorporated in the exogenous actions had value. We were therefore disconcerted to find that the agents with random actions are extracting much higher profits from this market. Is this because we have been lucky with our random draw? Or is there some misspecification in our model? Our results here are preliminary and we intend to do much more work in this area. But they are at least provocative. Could it be that managers learn bad patterns of competition in some markets? Our agents faced with a menu of random actions make most of their profits at prices above the shelf prices which were the highest prices used by the historical barnd managers. Such outcomes are what one might have expected before the data used in our earlier paper had been examined: that the highest profits earned would occur at collusive, high prices. But those data show that the greatest profits earned by the historical brand managers occur, not at high, above-shelf-price levels, but at low, sale prices. This suggests that one reason for the high-price profits of the agents with a menu of random actions to choose from is that the demand for coffee at high prices is not constrained by the sales from other supermarket chains. We have not modelled the effects of these sales on the demands faced by our agents.

Future research

A conclusion we draw from all the above is that it would be better if our agents endogenously determined their own perceptions and actions. Then we could compare the performance and behaviour of these agents with that of managers without the confounding of the two which comes from our current selection of actions and perceptions. But endogenising perceptions and actions is a non-trivial task. One of us has done some general thinking in the area of partitioning perceptions (Marks 1997) but we lack a specific algorithm to implement. We have not, as yet, done much thinking on what would be a basis for selecting a set of actions. Perhaps parts of the bit string could represent a price or promotional level. Similarly perceptions might be represented as partitions -- the coordinates of which were encoded in the bit string. The evolutionary process (GA) would then exert selective pressure on these bits, resulting in the selection of partitions and action sets that maximized profits.

The other area we wish to work on is that of the interaction between store policy and brand management. It is clear from our results that store policy has a major impact on brand competition. We need to consider joint models whereby store managers seek to impose policies that maximize their profits whilst brand mangers seek to maximize profits within these policies. Since brand managers also seek to influence store policy by various incentives, these will be complex models.

References

[1] Axelrod R. 1987, The evolution of strategies in the iterated Prisoner's Dilemma. In: Genetic Algorithms and Simulated Annealing, L. Davis (ed.) (Pitman, London) pp.32-41.

[2] Goldberg D.E. & Smith R.E. (1987) Nonstationary function optimization using genetic algorithms with dominance and diploidy. In: Grefenstette J.J. (ed.) Genetic Algorithms and their Applications, Proceedings of the 2nd. International Conference on Genetic Algorithms. Lawrence Erlbaum, Hillsdale, N.J.

[3] Gould S.J. 1996, Full House: The Spread of Excellence from Plato to Darwin, New York: Harmony Books. (Also, for some reason, published in London as Life's Grandeur.)

[4] Midgley D.F., Marks R.E., Cooper L.G. 1997, Breeding competitive strategies, Management Science, Volume 43, No. 3 March, pp.257-275.

[5] Mitchell M. 1996, An Introduction to Genetic Algorithms, Cambridge: MIT Press.

[6] 1996, New York Review of Books,