Using Genetic Algorithms to Breed Competitive Marketing Strategies

1998 IEEE International Conference on Systems, Man, and Cybernetics
La Jolla, California, USA
October 11–14, 1998
Intelligent Systems for Humans in a Cyberworld

G. M. Shiraz
Dept. of Artificial Intelligence
University of New South Wales
Sydney NSW 2052
Australia
hossein@cse.unsw.edu.au

Robert E. Marks
Australian Graduate School of Management
University of New South Wales
Sydney NSW 2052
Australia
bobm@agsm.unsw.edu.au

David F. Midgley
Australian Graduate School of Management
University of New South Wales
Sydney NSW 2052
Australia
davidm@agsm.unsw.edu.au

Lee G. Cooper
Anderson Graduate School of Management
University of California Los Angeles
Los Angeles CA 90095
USA
lcooper@agsm.ucla.edu

Using Genetic Algorithms to Breed Competitive Marketing Strategies

G. M. Shiraz
Dept. of Artificial Intelligence
University of New South Wales
Sydney NSW 2052
Australia
hossein@cse.unsw.edu.au

Robert E. Marks
Australian Graduate School of Management
University of New South Wales
Sydney NSW 2052
Australia
bobm@agsm.unsw.edu.au

David F. Midgley
Australian Graduate School of Management
University of New South Wales
Sydney NSW 2052
Australia
davidm@agsm.unsw.edu.au

Lee G. Cooper
Anderson Graduate School of Management
University of California Los Angeles
Los Angeles CA 90095
USA
lcooper@agsm.ucla.edu

ABSTRACT

Genetic algorithms (GAs) have been used extensively in engineering and computer science to optimize specific functions, especially those which exhibit non-convexities and so are not amenable to calculus-based methods of optimization. A parallel use of GAs has been to solve algorithmic problems. A third domain in which GAs have been used is that of searching for mappings which optimize a repeated procedure. An offshoot of this has been their use in what has been called co-evolution of mappings. This paper reports results from a project in which GAs have been used to, first, to derive mappings which may explain the behavior of brand managers in an oligopolistic retail market for coffee, and, second, to attempt to improve on the historical profits of these brand managers, pitted in weekly competition with each other, vying for sales and profits with their different brands of ground, sealed coffee on the supermarket shelves. As well as advancing the practice of GAs, with separate populations competing, the work also advances our understanding of modeling players in repeated oligopolistic interactions, or games.

1. INTRODUCTION

The theory of oligopolistic behavior (that is, the behavior of sellers in a market with a small number of sellers, but many buyers, so that one seller’s actions will affect the profits of other sellers, and vice versa) has mainly been approached from the point of view of searching for Nash equilibria in players’ actions, that is, a combination of actions, where each player’s actions are the best he can do for himself, given that the other players’ actions are the best they can do for themselves. Such a combination is self-reinforcing, since no single player has an incentive to alter his actions.

The project reported here, however, has been concerned with trying to explain and to improve upon the historical behavior and profits of a group of sellers, as recorded in supermarket scanner data, and using a market model to predict one-shot (weekly) profits of each player, given the marketing actions of all players. The data have been described in a recent article [1]. Briefly, each player has a choice of weekly actions: price per pound, coupons, in-store promotional displays, and featured local advertising. The CASPER market model [2], estimated from historical data, is used to identify each of the several firms’ weekly profits, given all brand managers’ actions.

We modeled the brand managers, the players, as stimulus-response automata [3], where the response is the player’s marketing actions for the next week, and the stimulus is the state of the market this week, which we took to be a function of all players’ actions this week and last week and several weeks past. The reason we believe that managers remember past actions is that this means they can respond to movements (aggressive or conciliatory) in other players' pricing.

For instance, it turns out that historically most prices and most sales have been made when prices are low. So if one brand, say Folgers, were pricing aggressively low last week, and raises its price this week, this could be a signal that it is becoming less aggressive, and might like reciprocation from its rival brands. If the brand managers are able to remember more than two weeks of marketing actions, then they may respond not just to rising or falling prices of their rivals, but also how quickly these prices are rising or falling. These issues are explored at greater length in [4].

2. MODELING THE MANAGERS

We model each manager as a finite automaton that responds to the state of the market with a set of marketing actions. To do this we need a set of rules, which are here represented by a binary string, following the Axelrod/Forrest representation [5]. Each string becomes an individual in a population of strings as artificial brand managers, and each string's average profit after a series of repeated interactions with the other artificial brand managers can be used as its "fitness" for the GA [6].

To be specific, say there are *p* players, each with *a* possible actions per week, and *m* weeks of memory, then the total number of possible states is given by

$$\text{number of possible states} = a^{mp}. \qquad (1)$$

This number increases rapidly: with three players, four actions, and one week of memory there are 64 possible states, but increasing memory to two weeks increases the number of possible states to 4,096.

Moreover, the length of the bit-string is only equal to the number of possible states in the unlikely event that a player can choose only from two possible actions, which can then be coded as zero or one. If, however, the player can choose from four actions, then the bit length doubles, and from eight actions it trebles, so that each possible state corresponds to three bits, which code for eight possible actions.

We are modeling the brand managers as boundedly rational: bounded in terms of their perceptions of reality, which is really saying that it is costly to perceive reality finely [4]; bounded in terms of their memory (which is another way of saying that their perception is limited because costly); and bounded in terms of the possible actions they can make. None the less, we found that our simple finite-automaton artificial brand managers could outperform their historical flesh-and-blood forbears [1]. In showing this, we were able to develop strings (using the GA to search through the space of possible mappings from history to actions) that represented real strategies in asymmetric markets (asymmetric because the brands historically faced different costs, evoked different responses from customers, and chose from different sets of possible actions).

This line of research does not merely pit each bit-string against a complex and sometimes noisy environment, as had been done by others, in looking at artificial players in repeated games [5]. We co-evolved the players, so that each string was being tested for its fitness against the consequences of other strings, which in turn were being tested for their fitness [7]. This may be a good example of Szpiro's "surfing in a seascape" [8].

2.1 The Agents' Choices

Given the problem of the curse of dimensionality, with rapid growth in the length of the bit strings modeling the agents, the question at first was how could we model the market interactions with the smallest sacrifice of realism? We focussed on the three most active brands in the market: Folgers, Maxwell House, and Chock Full O' Nuts, although later we have increased the number of strategic players.

We assumed that the decision to use coupons was equivalent to a reduction in price. Moreover, we chose at first to use only four possible prices, instead of the range available to the historical managers (from $1.50 per pound to about $3.00 per pound). For each of the three players we examined the historical pricing decisions to arrive at the brand-specific sets of four possible prices per player. At the same time, realising that other marketing actions (advertising feature and aisle display) were highly correlated with price, we factored those into the four pricing actions. (Only when the price is low did the historical players use feature or display, presumably to move more stock at an attractive price; see [1] for more discussion.)

To begin with, we modeled the players as remembering the actions of all three players of only one week ago, although this was relaxed later. With three players, each with four possible actions per week, and one week's memory, equation (1) tells us there are 64 possible states. With four possible actions, each state must map to two bits on the player's string. When, following [5], we use six bits for the phantom memory used in the first round (effectively endogenising the initial conditions of the simulation), each player is modeled with a 134-bit string. Not only are 134-bit strings easy to simulate, but the 75 weeks of historical data provided sufficient to evolve effective strings of this length.

Although it would have been possible to link the CASPER market model (which derives each brand's weekly profit, given the other brands' actions) to the GA, we found that computing the market response functions for each iteration of the game took an excessive time, and we had problems in marrying the compiled CASPER model with the compiled evaluation function of the GA. Moreover, with only 64 possible states, it seemed more elegant to derive three $4 \times 4 \times 4$ payoff matrices off-line (one per asymmetric brand), and to compile them into the GA as look-up routines. This was done, although later we would have to increase the dimensions of this array quite considerably.

2.2 The Genetic Algorithm

There is no need to describe the workings of GAs in 1998. There are many books [6][9] and articles doing this. Suffice it to say that in our earlier work [1] we adapted GAucsd, the U.C. San Diego version of John Grefenstette's GENESIS [10]. We describe below the extensions that we have made to it in order to examine the phenomena under review.

3. EXPERIMENTS

The results of the experiments described below are reported in more detail in [1] and [11]. Our purpose here is to discuss the extensions made to the GAucsd to accommodate our models.

3.1 Unconstrained Agents

Despite some expectations that collusion would occur at a high price (price is the most powerful of the several marketing actions available to the sellers, and we concentrate on it here), we found convergence, with all brands pricing at their lowest historical prices. This result was consistent with the historical observation that most sales and most profits occur at low prices with promotions, because of such behavior as stockpiling and brand-switching. Ground coffee in vacuum sealed cans has a storage life of up to seven weeks. Moreover, the historical market was mature, with no external shocks on either the supply side or the demand side.

3.2 Institutional Constraints

Unfortunately, these results were unrealistic since historically only one brand a week has priced at the low promotional level to which all brands had converged. The supermarket chain whose scanner data we were using had managed to maximize its profits while not exhausting demand. Its policy was to constrain the brands: only one brand promoting with low prices in any week, and no brand promoting with low prices in two successive weeks.

We mimicked this. Ties in which two or more brands respond to the state of the market via their mapping strings by both promoting at low prices were broken by random choice, the loser pricing arbitrarily high. In order to speed up the simulations, we determined that we could examine the genotype (the structure of each artificial brand's bit-string) to see whether that string's low promotion price this week would be followed by a similar price next week, rather than waiting for the simulation to reveal the particular realization of the player's phenotype (its response behavior). This "filtering" of strings greatly speeded up the simulation, since strings whose structures revealed illegal successive promotions were given arbitrarily low fitness, and their characteristics were excluded from future generations of strings by the GA. After 20 generations (with a population size of 25), most illegal strings had vanished, and the last had usually disappeared by generation 44.

Although the brands' behavior was closer to that seen historically [1], we found that, because the market model CASPER had been written and estimated for a single week's interaction, the overall levels of low, promotional prices were leading, with brand switching, to demand saturation.

3.3 Demand Saturation

While the retail coffee market is very volatile in the short run, it is very stable in the long run [1]. We pro-rated the weekly total by the degree of over-saturation of the past seven weeks, chosen to approximate the average interpurchase interval for this product. We first calculated the total sales volume per week, a function of the actions of the three strategic brands and the remaining non-strategic brands (whose behavior was assumed to be static). We then calculated the average total sales volume over the previous seven weeks and with a figure for the historical average total sales volume in this market calculated the percentage degree of saturation. If this was above 100%, then the total sales volume for the latest week was reduced by the degree of saturation. (In steady state, this procedure means that total sales volume must equal the historical average.) Then the profits of the three strategic brands were reduced from the limits now placed on each brand's sales volume.

The results of this experiment are to be seen in Figure 2 of [1]. The experiment results in a greater degree of competition than observed historically, owing to the immediacy of the simulation laboratory, in which brands immediately respond to others' actions last week. The artificial brand managers thus generated average weekly profits from 3.5 to 9.7 times higher than did the historical brand managers.

3.4 Tests Against History

How well had our best artificial agents learned (or evolved) to play the game which models the oligopolistic market for coffee we are examining? In order to answer this question, we took the most profitable agents from the previous series of experiments (after 100 generations of the GA) and tested each in turn against the historical actions of their two strategic rivals. The historical actions of the five non-strategic brands were also used, but our artificial agents as modeled were blind to these actions.

This was achieved by taking a string, designating it as a particular brand, say Maxwell House, and allowing it to respond to the historical actions of the two rivals brands over a 52-week period of history. Since the historical brand managers had had a much larger range of prices and other actions to choose from (although the artificial player's range spanned the historical range), we used a rough partitioning of the historical actions into four intervals, to which the artificial agent responded [4]. Its performance was measured by its average profits over this period, calculated weekly by CASPER, with the historical actions of the other strategic and non-strategic players as input. Since the GA's population size was 25, there were 25 possible strings: only later did we separate the players into distinct populations to be evolved in parallel by the amended GA.

The results are detailed in [1]. For two brands (Folgers and Chock Full O' Nuts) most of the strings performed better than their historical counterparts did; for Maxwell House only two of the 25 strings did (although they were 20% more profitable, none the less). Maxwell House historically was the most profitable of the three brands, so perhaps the artificial agents faced a higher performance hurdle.

A criticism of this experiment is that it is an "open-loop" regime: although the artificial agent responds to the historical actions, week by week, as it had been bred to do by the GA, the historical actions are fixed, with no possibility of responding to the artificial agent's action last week.

Another criticism, which we address in Section 3.5 below, is that we were using a single population of strings in the GA. When the problem is static, a single population of strings provides many possible solutions (Holland's "implicit parallelism" [12]), but when we engage in coevolution with asymmetrical players, as here, there is no reason to believe that "one size fits all", especially since the same state may best trigger quite different responses in different brands.

Because of these concerns, we concluded that what was impressive about these results was not that our artificial agents could outperform their historical counterparts, but that very simple agents (with only four possible actions and one week's memory) could generate reasonable performance in the noisy coevolutionary environment.

3.5 Multiple Population Simulations

As mentioned, despite the fact that we were coevolving asymmetric agents, we — in common with all other users of the GA — had been using a single population. As well as making it much harder for the GA to search for fitter mapping strings (consider: a single string might perform well as one brand but badly as another), a single population means that, through the genetic recombination of the GA, strings may be communicating genotypically, as well as phenotypically via their fitness (profitability) in the repeated interaction. Tony Curzon Price has called this "incest," in a personal communication.

We have extended GAucsd to include multiple populations of bit strings, so that the fitness of any string is dependent upon all strings in the other strategic players' populations. As well as making things less noisy for the GA, having distinct populations means that the strings are interacting only via their phenotypic behavior, and not at the genotypic structural level, since the populations are entirely separate, as far as the GA knows.

Amending the GAucsd software was not a trivial exercise, since three or four players may be interacting many times in determining each string's fitness (its average weekly profits). One of us (Shiraz) took the opportunity to streamline the logic of the fitness evaluation functions, by recording the other strings' performances during the round-robin interactions, so that the new code with three populations is almost as fast as the old code with a single population.

Because of the stochastic nature of the simulations, we have performed Monte Carlo simulations (50 runs each) to compare the convergence and profits of the common-population GA (25 strings, 50 simulations each) with those of the distinct-population GA (three populations of 25 strings each, 50 simulations each).

The distinct-population GA generates more profitable strings and converges faster than does the common-population GA. In aggregate, the improvements to average weekly profit are only about 4%, but this summary statistic masks interesting brand-specific outcomes: with distinct string populations, Folgers' profits increase by 3% and Maxwell House's by 24%, while Chock Full O' Nuts' profits fall by 16%. Distinct populations allow the Maxwell House strings to better capitalize on that brand's strengths.

The distinct-population GA allows the brands to differentiate themselves more in terms of the patterns of weekly response, as [11] reports. Moreover, when testing strings from the distinct-population GA against history (see Section 3.4 above), we found that strings coevolved using the distinct-population GA did better against history than did strings coevolved using the common-population GA.

Indeed, we conclude that moving to distinct populations has generally resulted in higher-performing strings, both when coevolving and when competing against the historical actions of brand managers, and that distinct populations also result in greater heterogeneity in the performance of each brand's artificial agents.

3.6 Four Strategic Players

With the rewritten, multi-population GA code, it was relatively easy to extend the simulations to a fourth strategic player, at some cost in terms of the complexity of the bit strings, which grew in length from 134 bits (three players, four actions, one-week memory) to 520 bits (including the initial week's phantom memory).

Although Hills Bros., the fourth player, was a niche player, with smaller profits than the other brands, its inclusion results in significant and complex changes in the behavior and profitability of the three major brands. The details can be read in [11]. The impacts were greater than we had anticipated, but our approach allows us to analyze the changes using a methodology based on a detailed, realistic, and empirically grounded model of consumer response.

3.7 Eight Actions per Player

We had chosen the number of four possible actions per player for convenience in our initial work, but were pleased with the results we obtained with our constrained strings none the less. But rather than exogenously imposing our decisions on the artificial managers, we would prefer them to learn which actions were most profitable, given the actions of their rivals. By increasing the number of possible actions to eight, we hoped to give the artificial managers the opportunity of demonstrating that the four actions used previously were robust, and that our assumption of a mature oligopoly was correct.

Doubling the number of possible actions implies further complexity: from 520 bits per string to 12,312 bits per string. Of each brand's eight actions, we chose six from an historical analysis, to which we added the brand's highest observed price and lowest promotional price, thus providing each artificial manager with a much richer set of possible actions than previously.

Although in early generations of the GA simulation each of the eight actions is used with a similar frequency, by the hundredth generation (25 individuals per population) the artificial managers fall into one of two patterns of competitive interaction, as revealed by 50 Monte Carlo runs, both of which employ many fewer than eight actions. The managers have learnt the two or three actions that are most profitable for them, given the behavior of their rivals. Against the historical actions of actual brand managers, the artificial managers do at least as well as their historical counterparts. See [11] for details.

3.8 Co-evolution: Sophisticates against Primitives

Unlike the use of GAs to solve static problems, where the fitness scores of the simulation improve as generations pass, when the strings model artificial managers competing against other evolving artificial managers — co-evolution — fitness scores may not improve from generation to generation. Rather than engaging an evolved string in the open-loop competition against the frozen patterns of behavior of its historical rivals, as reported in Section 3.4 above, we take a string (the "sophisticate") from the hundredth generation and play it against rival strings (the "primitives") from the eighth generation.

Since the sophisticates have had many more generations to learn and adapt than have the primitives, we should expect them to score better against primitive than against sophisticated rivals. But, using the original three brands and 50-run Monte Carlo simulations, we found that for two of the three brands the sophisticates do not compete effectively with the primitives, a phenomenon that Bernhard Borges has dubbed the Holyfield-Tyson effect.

Is this due to genetic drift, where the gene pool of a small population may change randomly, when specific genes (positions on our strings) are not useful in scoring well? To test this conjecture, we increased the size of each population from 25 to 250, which means that each string now has to compete against 250^2 combinations, instead of 25^2, and there are ten times as many strings to test, a thousand-fold increase in the number of three-way interactions per generation. Convergence is also likely to be much slower. We did not attempt Monte Carlos: a single simulation run took weeks rather than days to complete.

The results of our large-population simulations [11] appear to eliminate genetic drift as an explanation, but, given the length of the cycles of convergence, we cannot rule out the emergence of higher-performing sophisticates after the hundredth generation. Moreover, we were able to in the time available to examine a model with three players and four possible actions. Would an eight-action model, allowing the artificial agents greater degrees of freedom as discussion in Section 3.7 above, demonstrate genetic drift? Our prior is no.

4. CONCLUSIONS

Although we believe that our papers provide much insight into the historical patterns of oligopolistic rivalry in a mature market, as well as revealing how historical brand managers might learn to improve their profitability and competitiveness by consideration of the patterns and strategies learnt by the artificial brand managers via the GA simulation of coevolution, we have focussed here on our contributions to the use of GAs in competition analysis.

We have shown that it is possible and appropriate to use multi-population GAs when co-evolving asymmetric artificial agents. We have shown that the GA can effectively used for bit-string agents of very high complexity. We have shown the potential of GAs to be used in exploring the patterns and strategies of asymmetrical rivals in a mature oligopoly.

5. REFERENCES

[1] D.F. Midgley, R.E. Marks, and L.G. Cooper, “Breeding competitive strategies,“ *Management Science*, Vol. 43, No. 3, March 1997, pp. 257–275.

[2] L.G. Cooper and M. Nakanishi, *Market Share Analysis: Evaluating Competitive Marketing Effectiveness,* Boston: Kluwer, 1988.

[3] R.E. Marks, “Repeated games and finite automata,” in *Recent Developments in Game Theory,* ed. by J. Creedy, J. Eichberger, and J. Borland, London: Edward Elgar, 1992, pp. 43–64.

[4] R.E. Marks, “Evolved perception and behavior in oligopolies,” *Journal of Economic Dynamics and Control*, Vol. 22, No. 8–9, pp. 1209–1233, July 1998.

[5] R. Axelrod, “The evolution of strategies in the iterated Prisoner’s Dilemma,” *Genetic Algorithms & Simulated Annealing,* L. Davis (ed.), London: Pittman, 1987.

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

[7] R.E. Marks, “Breeding optimal strategies: optimal behavior for oligopolists,” *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June 4–7, 1989, ed. by J. David Schaffer, San Mateo, Calif.: Morgan Kaufmann Publishers, pp.198–207, 1989.

[8] G. Szpiro, “The emergence of risk aversion,” *Complexity*, Vol. 2, pp. 31–39, 1997.

[9] D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence,* New York: IEEE Press, 1995.

[10] N.N. Schraudolph and J.J. Grefenstette, A User’s Guide to GAucsd 1.4, Technical Report CS92-249, UCSD CSE Department, La Jolla, CA, 1992.

[11] R.E. Marks, D.F. Midgley, and L.G. Cooper, Refining the Breeding of Hybrid Strategies, Australian Graduate School of Management Working Paper, Sydney, 1998.

[12] J.H. Holland, *Adaptation in Natural and Artificial Systems*, second edition, Cambridge: MIT Press, 1992.