Coevolution with the Genetic Algorithm: Repeated Differentiated Oligopolies

Robert E. Marks
David F. Midgley
Australian Grad. Sch. of Management
Univ. of New South Wales,
Sydney NSW 2052
Australia.
bobm@agsm.edu.au
davidm@agsm.edu.au

Lee G. Cooper
Anderson Grad. Sch. of Management
UCLA
Los Angeles CA 90095
USA
lcooper@agsm.ucla.edu

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

Abstract

This paper reports results from a project in which Genetic Algorithms (GAs) have been used, first, to derive mappings which may explain the behavior of brand managers in an oligopolistic retail market for coffee; 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; and, third, to reveal how the artificial agents' performance is positively related to their complexity. As well as advancing the practice of GAs, with coevolving 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 each seller's actions will affect the profits of the other sellers) 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, is 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 are described in a recent article (Midgley et al., 1997). 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 (Cooper & Nakanishi, 1988), estimated from historical data, is used to identify each of the several firms' weekly profits, given all brand managers' actions.

We model the brand managers, the players, as stimulus-response automata (Marks, 1992), 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 take 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 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 Marks (1998).

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 (Axelrod, 1987). Each string becomes an individual in a population of 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 (Mitchell, 1996).

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

$$\text{number of possible states} = a^{mp}. \tag{1}$$

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 model 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 (Marks, 1998); 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 find that our simple finite-automaton artificial brand managers can outperform their historical flesh-and-blood forbears (Midgley et al., 1997). In showing this, we are able to develop strings (using the GA to search through the space of possible mappings from history to actions) that represent 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 has been done by others, in looking at artificial players in repeated games (Axelrod, 1987). We coevolve the players, so that each string is being tested for its fitness against the consequences of other strings, which in turn are being tested for their fitness (Marks, 1989). This may be a good example of "surfing in a seascape" (Szpiro, 1997).

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 is how can we model the market interactions with the smallest sacrifice of realism? We focus on the three most active brands in the market: Folgers, Maxwell House (MH), and Chock Full O' Nuts (CFON), although later we increase the number of strategic players.

We assume that the decision to use coupons is equivalent to a reduction in price. Moreover, we choose 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 examine 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 factor those into the four pricing actions, as seen in Table 3 below. (Only when the price is low did the historical players use feature or display, presumably to move more stock at an attractive price (Midgley et al., 1997).)

To begin with, we model the players as remembering the actions of all three players of only one week ago, although this is 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 Axelrod (1987), 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 provide sufficient to evolve effective strings of this length.

Although it is 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 is 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 is done, although later we increase the dimensions of this array quite considerably.

2.2 THE GENETIC ALGORITHM

There is no need in 1999 to describe the workings of GAs. There are many books (Mitchell, 1996; Fogel, 1995) and articles doing this. Suffice it to say that in our earlier work (Midgley et al., 1997) we adapted GAucsd, the U.C. San Diego version of John Grefenstette's GENESIS (Schraudolph & Grefenstette, 1992). 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 Midgley et al. (1997) and Marks et al. (1998). Our purpose here is to discuss the extensions made to the GAucsd to accommodate our models and the performance of the artificial agents.

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 — see Table 3), we find convergence, with all brands pricing at their lowest historical prices. This result is 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, over the period considered.

3.2 INSTITUTIONAL CONSTRAINTS

Unfortunately, these results are unrealistic, since historically only one brand a week priced at the low promotional level to which all brands converged. The supermarket chain whose scanner data we use 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 mimic this. Ties in which two or more brands respond to the state of the market via their mapping strings by each promoting at low prices are broken by random choice, the loser pricing arbitrarily high. In order to speed up the simulations, we examine the genotype (the structure of each artificial brand's bit-string) to see whether that string's low promotion price this week will 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 speeds up the simulation, since strings whose structures reveal illegal successive promotions are given arbitrarily low fitness, and their characteristics are excluded from future generations of strings by the GA. After 20 generations (with a population size of 25), most illegal strings vanish, and the last usually disappear by generation 44.

Although the brands' behavior is closer to that seen historically (Midgley et al., 1997), we find that, because the market model CASPER was written and estimated for a single week's interaction, the overall levels of low, promotional prices lead, 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 (Midgley et al., 1997). We pro-rate 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 calculate the total sales volume per week, a function of the actions of the three strategic brands and the remaining non-strategic brands (whose behavior is assumed to be static). We then calculate the average total sales volume over the previous seven weeks and, with a figure for the historical average total sales volume in this market, calculate the percentage degree of saturation. If this is above 100%, then the total sales volume for the latest week is 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 are reduced from the limits now placed on each brand's sales volume.

The results of this experiment are seen in Figure 2 of Midgley et al. (1997). 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 generate average weekly profits from 3.5 to 9.7 times higher than did the historical brand managers.

3.4 TESTS AGAINST HISTORY

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

This is achieved by taking a string, designating it as a particular brand, say MH, 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 a much larger range of prices and other actions to choose from (although the artificial player's range spans the historical range), we use a rough partitioning of the historical actions into four intervals, to which the artificial agent respond (Marks, 1998). Its performance is 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 is 25, there are 25 possible strings: only later do we separate the players into distinct populations to be coevolved in parallel by the amended GA.

The results are detailed in Midgley et al. (1997). For two brands (Folgers and CFON) most of the strings perform better than did their historical counterparts; for MH only two of the 25 strings do (although they are 20% more profitable, none the less). MH historically was the most profitable of the three brands, so perhaps the artificial agents face 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 has 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 below, is that we are using a single population of strings in the GA. When the problem is static, a single population of strings provides many possible solutions ("implicit parallelism" [Holland, 1992]), 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 conclude that what is impressive about these results is not that our artificial agents outperform their historical counterparts, but that very simple agents (with only four possible actions and one week's memory) can generate reasonable performance in the noisy coevolutionary environment.

3.5 MULTIPLE-POPULATION SIMULATIONS

As mentioned, despite the fact that we coevolve asymmetric agents, we — in common with all other users of the GA — have 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 (pers. com.) has called this "incest."

Koza (1992) was the first to propose coevolution as a general procedure, although others (Husband & Mill, 1991; Hillis, 1992) have used it casually. Angeline and Pollack (1993) argue that coevolution with separate populations will cause the GA to converge faster to an optimum. Because of the asymmetries across brands in our market — asymmetric costs, asymmetric perceptions, asymmetric market responses — it makes sense to coevolve the brands using distinct populations.

We extend 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 is 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 perform 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).

Comparing Table 1 with Table 2, we see that the distinct-population GA generates more profitable strings and converges faster than does the common-population GA. In aggregate, the improvements to average weekly profits 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 MH strings to better capitalize on that brand's strengths.

The distinct-population, coevolutionary GA allows the brands to differentiate themselves more in terms of their patterns of weekly response (Midgley et al., 1997). Moreover, when testing strings from the distinct-population GA against history (see Section 3.4 above), we find that strings coevolved using the distinct-population GA do better against history than do strings evolved using the common-population GA.

Indeed, we conclude that moving to distinct populations generally results 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 is relatively easy to extend the simulations to a fourth strategic player, at some cost in terms of the complexity of the bit strings, which grow 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, is 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 (Marks et al., 1998). The impacts are greater than we anticipated, but

TABLE 1. Patterns of Competition Among Evolved Agents—Common Population and 4 Actions

	Actions: Low price			High price	Average Profit ($)
Pattern 1 21/50 runs a	1	2	3	4	
Folgers	1* b,c	98	0	1	1,022
MH	32*	7	14	47	631
CFON	0*	100	0	0	633
Pattern 2 11/50 runs	1	2	3	4	
Folgers	0*	97	2	1	1,011
MH	33*	4	10	53	625
CFON	0*	98	0	2	630
Pattern 3 1/50 runs d	1	2	3	4	
Folgers	46*	52	0	2	1,082
MH	30*	0	34	36	623
CFON	0*	50	0	50	707

a. Patterns of competition are computed during the hundredth generation from all combinations of 25 agents playing 52-week games.
b. Row percentages total 100%.
c. Asterisks * identify the actions constrained by store policy.
d. Best performing of remaining 18 patterns.

TABLE 2. Patterns of Competition Among Evolved Agents—3 Distinct Populations and 4 Actions

	Actions: Low price			High price	Average Profit ($)
Pattern 1 25/50 runs a	1	2	3	4	
Folgers	1* b,c	92	3	4	1,093
MH	47*	0	3	50	804
CFON	2*	91	3	4	527
Pattern 2 16/50 runs	1	2	3	4	
Folgers	1*	94	2	4	1,092
MH	47*	1	3	48	804
CFON	1*	91	3	4	527
Pattern 3 1/50 run d	1	2	3	4	
Folgers	2*	92	0	6	1,045
MH	46*	0	4	50	830
CFON	48*	44	4	4	580

a. Patterns of competition are computed during the hundredth generation from all combinations of 25 agents playing 52-week games.
b. Row percentages total 100%.
c. Asterisks * identify the actions constrained by store policy.
d. Best performing of remaining 9 patterns.

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 chose 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 are most profitable, given the actions of their rivals. By increasing the number of possible actions to eight, we hope to give the artificial managers the opportunity of demonstrating that the four actions used previously are robust, and that our assumption of a mature oligopoly are correct. Table 3 shows the four and eight possible actions by specific player.

TABLE 3. Sets of Four and Eight Possible Actions.

	Folgers			Maxwell House			CFON		
A	P ($)	F (%)	D (%)	P ($)	F (%)	D (%)	P ($)	F (%)	D (%)
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

* Asterisked actions are subject to store policy. A is Action, P is Price/lb., F is advertising Feature, D is aisle Display.

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 choose six from an historical analysis, to which we add 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, both of which employ many fewer than eight actions, as revealed by 50 Monte Carlo runs. See Tables 4 and 5. The managers learn 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

TABLE 4. Frequency of Actions Over the First Four Generations

	Actions							
	Low Price							High Price
Pattern a	1	2	3	4	5	6	7	8
Folgers	8* b,c	7*	11	8*	6*	13	11	36
MH	6*	7*	6*	15	12	13	12	29
CFON	11	7*	6*	13	7*	13	12	31

a. Patterns of competition computed over the first four generations of one simulation.
b. Row percentages total 100%.
c. Asterisks * identify the actions constrained by store policy

TABLE 5. Frequency of Actions During the Hundredth Generation

	Actions							
	Low Price							High Price
Pattern 1 27/50 runs a	1	2	3	4	5	6	7	8
Folgers	20* b,c	3*	11	20*	1*	25	1	20
MH	3*	1*	3*	61	2	12	2	15
CFON	3	34*	0*	10	0*	8	4	40
Pattern 2 14/50 runs	1	2	3	4	5	6	7	8
Folgers	27*	7*	11	5*	0*	31	1	18
MH	1*	1*	3*	66	9	8	5	8
CFON	1	30*	0*	7*	1	11	4	46

a. Patterns of competition are computed during the hundredth generation from all combinations of 25 agents playing 52-week games.
b. Row percentages total 100%.
c. Asterisks * identify the actions constrained by chain policy.

as well as their historical counterparts (Marks et al., 1998).

3.8 COEVOLUTION: 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 — coevolution — 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. Table 6 presents the results.

TABLE 6. Mean Changes in Average Weekly Profits with Best Sophisticate

Best Sophisticate	Change in Folgers	Change in MH	Change in CFON
Folgers	–15.01	41.42	42.03
MH	2.03	–20.04	37.77
CFON	13.93	–28.99	82.34

Since the sophisticates have had many more generations to learn and adapt than have the primitives, we expect them to score better against primitive than against sophisticated rivals. But, using the original three brands and 50-run Monte Carlo simulations, we find that for two of the three brands the sophisticates do not compete effectively with the primitives, a phenomenon that Bernhard Borges (pers. com.) 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 increase 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 will be much slower. We did not attempt Monte Carlos: a single simulation run takes weeks rather than hours to complete. Table 7 presents the results.

TABLE 7. Mean Changes in Average Weekly Profits with Best Sophisticate After 160 Generations, Population of 250

Best Sophisticate	Change in Folgers	Change in MH	Change in CFON
Folgers	–87.11	75.13	–55.66
MH	–101.87	–512.51	155.45
CFON	–63.19	–42.08	–23.77

The results of our large-population simulations (Marks et al., 1998) 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 only. 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 focused 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 coevolving 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.

Acknowledgments

This work was partly supported by the Australian Research Council and the Australian Graduate School of Management. We thank Bernhard Borges, Tony Curzon Price, and four anonymous referees for their comments and support.

References

P. J. Angeline and J. B. Pollack (1993). Competitive environments evolve better solutions for complex tasks. In S. Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, 264–270. San Mateo, CA.: Morgan Kaufmann.

R. Axelrod (1987). The evolution of strategies in the iterated Prisoner's Dilemma. In L. Davis (ed.), *Genetic Algorithms and Simulated Annealing.* London: Pittman.

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

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

W. D. Hillis (1992). Co-evolving parasites improve simulated evolution as an optimization procedure. In J. D. Farmer, C. G. Langton, C. Taylor, and S. Rasmussen (eds.), *Artificial Life II*, 313–323. Menlo Park, CA., Addison-Wesley.

J. H. Holland (1992). *Adaptation in Natural and Artificial Systems*, second ed., Cambridge: MIT Press.

P. Husband and F. Mill (1991). Simulating co-evolution as the mechanism for emergent planning and scheduling. In R. Belew and L. Booker (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms,* 264–270. San Mateo, CA.: Morgan Kaufmann.

J. R. Koza (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

R. E. Marks (1989). Breeding optimal strategies: optimal behavior for oligopolists. In J. David Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, 198–207. San Mateo, CA.: Morgan Kaufmann.

R. E. Marks (1992). Repeated games and finite automata. In J. Creedy, J. Eichberger, and J. Borland (eds.), *Recent Developments in Game Theory*, 43–64. London: Edward Elgar.

R. E. Marks (1998). Evolved perception and behavior in oligopolies. *Journal of Economic Dynamics and Control* **22**(8–9): 1209–1233.

R. E. Marks, D. F. Midgley, and L. G. Cooper (1998). Refining the Breeding of Hybrid Strategies, Australian Graduate School of Management Working Paper 98–017, Sydney.
<http://www.agsm.edu.au/~bobm/papers/wp98017.html>

D. F. Midgley, R. E. Marks, and L. G. Cooper (1997). Breeding competitive strategies. *Management Science* **43**(3): 257–275.

M. Mitchell (1996). *An Introduction to Genetic Algorithms.* Cambridge: MIT Press.

N. N. Schraudolph and J. J. Grefenstette (1992). A User's Guide to GAucsd 1.4, Technical Report CS92–249, UCSD CSE Department, La Jolla, CA.

G. Szpiro (1997). The emergence of risk aversion, *Complexity* **2**: 31–39.