BREEDING HYBRID STRATEGIES:

Optimal Behaviour for Oligopolists

Robert E. Marks

Australian Graduate School of Management,
University of New South Wales,
P.O. Box 1,
Kensington NSW 2033
Australia

Telephone: +61 2 931–9271
Fax: +61 2 662–2451
Internet: `bobm@agsm.unsw.oz.au`

Abstract

OLIGOPOLISTIC pricing decisions—in which the choice variable is not dichotomous as in the simple Prisoner's Dilemma but continuous—have been modeled as a Generalized Prisoner's Dilemma (GPD) by Fader and Hauser, who sought, in the two MIT Computer Strategy Tournaments, to obtain an effective generalization of Rapoport's Tit for Tat for the three-person repeated game. Holland's genetic algorithm and Axelrod's representation of contingent strategies provide a means of generating new strategies in the computer, through machine learning, without outside submissions.

The paper discusses how findings from two-person tournaments can be extended to the GPD, in particular how the author's winning strategy in the Second MIT Competitive Strategy Tournament could be bettered. The paper provides insight into how oligopolistic pricing competitors can successfully compete, and underlines the importance of "niche" strategies, successful against a particular environment of competitors.

Bootstrapping, or breeding strategies against their peers, provides a means of examining whether "repetition leads to coöperation": we show that it can, under certain conditions, for simple and extended two- and three-person GPD repeated games. The paper concludes with a discussion of the relationship between Selten's trembling-hand perfect equilibrium and Maynard Smith's evolutionarily stable strategies, with practical simulations of successful and unsuccessful "invasions" by new strategies.

Keywords: REPEATED GAMES; GENETIC ALGORITHM; STRATEGIES

Appeared in the *Journal of Evolutionary Economics*,
2, 1992, pp 17-38.

CONTENTS

1. Competition among the Few 1
2. Game Theory and Strategic Behaviour 2
3. Modeling Oligopolistic Behaviour 4
   3.1 Modeling Strategies in Repeated Games 4
   3.2 Strategies as Bit-String Mappings 5
4. Niche Strategies 6
5. Bootstrapping Evolution 8
   5.1 The Simple Prisoner's Dilemma 10
   5.2 Extended Prisoner's Dilemma Games 10
   5.3 The MIT Competitive Strategy Tournaments 12
6. Invasion and the Trembling Hand 14
   6.1 Invasion of an Extended Prisoner's Dilemma 15
   6.2 Invasion of the Three-Person Game 16
7. Conclusions 16
Appendix: Genetic Algorithms[8] 18
References 22

LIST OF FIGURES

Figure 1. The Simple Prisoner's Dilemma 7
Figure 2. An Extended Two-Person Prisoner's Dilemma 11
Figure 3. Profits: Monopoly L, Cournot M, and Competitive H 11
Figure 4. First Player's Payoff Matrix 13
Figure 5. First Player's Payoff Matrix 14

Breeding Hybrid Strategies: Optimal Behaviour for Oligopolists

Robert E. Marks
Australian Graduate School of Management,
University of New South Wales,
Kensington, NSW 2033,
Australia.
Internet: bobm@agsm.unsw.oz.au *

1. Competition among the Few

THE OLIGOPOLY problem can be stated as: with a small number of competitive sellers, what is the equilibrium pattern of price and quantity across these sellers, if any? Cournot, in his celebrated example of mineral-water producers (1838), envisaged that competing firms would decide their production levels, and that a market-clearing price would occur from the aggregate of their supply facing a market demand. He characterized equilibrium in this market as occurring when the output of each firm is the best response to the other firms' outputs; that is, the equilibrium level of output for each firm depends on the actions of its competitors, and no single firm can increase its profit by using a different output level. This *strategic* feature distinguishes oligopolistic equilibria from those of pure competition and monopoly. Cournot's analysis was explicitly for static, one-shot markets.

In a review of Cournot's book fifty years later, Bertrand (1883) argued that price—rather than quantity—was the variable set by firms, in which case, as he demonstrated, competition between sellers of homogeneous goods results in the competitive price and quantity, even if there are only two sellers. If the products are differentiated, then the seller who quotes a higher price still sells some quantity, and the price-setting equivalent of Cournot's equilibrium is an example of a Nash equilibrium in a non-coöperative game, where the firm's output level is its (pure) strategy. Any strategy combination is a Nash equilibrium if each player's optimal strategy belongs to the appropriate strategy set (that is, is attainable by the player) and if it is impossible that any single player can obtain a higher payoff through the use of a different strategy, given the strategy choices of the remaining players.

A market of three sellers, each facing an elastic demand and selling a differentiated output, can be modeled as a three-person Generalized Prisoner's Dilemma (GPD): each seller's profits would be maximized by a

* The author would like to thank Robert Axelrod for his comments and assistance, John Grefenstette for his GENESIS algorithm and advice, David Midgley for several valuable electronic discussions, and David Schaffer, John Koza, and the anonymous referees for their comments. Earlier versions of this paper were presented at the December 1988 meetings of the American Economics Association in New York under the auspices of the Econometric Society, and at the Third International Conference on Genetic Algorithms, at George Mason University, June, 1989.

coöperative, high price, but competition drives the price down towards the Pareto-inferior, non-coöperative, Nash price and hence each seller's total profits down, even though with elastic demand the market grows. Will repetition break this logic? Three-person GPD tournaments at MIT (Fader and Hauser 1988) and the AGSM were run to see whether entrants' strategies could generalize Rapoport's Tit for Tat (coöperate on the first round and then mimic one's opponent's previous move) from the two-person to the three-person repeated game.

In this paper we revisit the price wars of the GPD tournaments armed with the techniques of machine-learning known as the genetic algorithm (GA), which can—with the appropriate modeling of strategy selection—obviate the need for submitted strategies. Section 2 gives a brief history of research into games and oligopoly behaviour. Section 3 discusses Axelrod and Forrest's method of modeling strategies in repeated games as bit-string mappings between each player's state and that player's next move or action, which enables the GA to search for solutions efficiently. Section 4 reports results of niche strategies against unchanging environments of strategies. Section 5 introduces the idea of *bootstrapping* to obtain an optimum optimorum, given the implicit constraints of the particular model, and reports results of this in two- and three-person games. Section 6 discusses concepts of stability, and examines the stability of stable strategies when confronted with invaders. Section 7 concludes with a discussion of future research work on market strategies. The Appendix provides a brief overview of genetic algorithms.

2. Game Theory and Strategic Behaviour

MICRO-ECONOMICS has recently been enriched by studies of strategic behaviour among small numbers of competitors, in oligopolistic markets (see J.W. Friedman 1983, for instance). Following Cournot, these have been in terms of the dynamic adjustment of the competitors' behaviours, and have been facilitated by the insights from game theory (Schelling 1984; Ulph 1987). Strategic behaviour is important because in competition among few agents the individual agent is neither powerless (pure competition) nor powerful (monopoly), and the interaction among competitors cannot be readily described in a closed form.

The strategic behaviour of two competitors has been extensively studied in simple two-person games, the most productive of which has been the Prisoner's Dilemma (PD) (Diekmann and Mitter 1986)—although, as Rapoport (1988) reminds us, there are hundreds of other strategically unequivocal ordinal 2 × 2 games to be explored. In its one-shot version the PD demonstrates how the logic of self-interest, in the absence of trust or enforceable pre-commitment, results in a Cournot–Nash solution of non-coöperation that is Pareto-inferior to the coöperative solution.

In a single PD game, the dominant (pure) strategy is to defect, despite a higher payoff for coöperation, because of the reward of cheating and the penalty of being cheated. In a repeated PD game of unknown length, however, the higher payoff to coöperation may result in strategies different from the Always Defect of the single game, because of the possibility of punishing defection provided by later rounds. By breaking the logical imperative of mutual defection inherent in the static, one-shot PD, the repeated PD—in which the players repeatedly face each other in the same situation—can admit the possibility of *learning* on the part of the players, which may result in mutual coöperation or some mixed strategy on their part, as they learn more about the type of behaviour they can expect from each other and build up a set of commonly held norms of behaviour.

An early analysis of successful strategies in the repeated PD (Luce and Raiffa 1957, pp.97–102) suggested that continued, mutual coöperation might be a viable strategy, despite the rewards from defection, but for twenty years no stronger analytical results were obtained for the repeated PD.

In the late 1970s, political scientist Robert Axelrod, in an investigation of the emergence of coöperative behaviour and social norms in Hobbesian societies, hit upon the idea of exhaustively pitting strategies for the repeated PD by coding them into computer algorithms. He called for entries of strategies (for the repeated PD) coded as computer algorithms, and ran successive tournaments that attempted to reveal the "best" (highest scoring) strategy (Axelrod 1984, Axelrod and Dion 1988). In essence the tournaments were an attempt to search the strategy space by asking researchers in diverse disciplines to devise and submit strategies.

As is now widely known, Axelrod's tournaments revealed that one very simple strategy is very difficult to better in the repeated PD: Rapoport's Tit for Tat. When pitted against a "nasty" strategy, such as Always Defect, it does almost as well, itself defecting on every round but the first, but at the cost of the aggregate score. When played against itself, each player's aggregate score is a maximum, since every round will then be mutual coöperation, a result which resembles collusion, although each player's decisions are made independently of the other's.

Axelrod's tournaments and later tournaments modeling a three-person price war (Fader and Hauser 1988) were an attempt to pit as wide a variety of strategies against each other as possible, in order to derive more robust results and insights than would follow with a small set of strategies, although knowledge of Tit for Tat's success in the two-person tournaments may well have conditioned later strategies, as Nachbar (1988*a*) argues, questioning the robustness of the results.

Mathematically, the problem of generating winning strategies is equivalent to solving a multi-dimensional, non-linear optimization with many local optima. In population genetic terms, it is equivalent to selecting for fitness. Indeed, in a footnote, Cohen and Axelrod (1984, p.40) suggest that

> One possible solution may lie in employing an analogue of the adaptive process used in a pool of genes to become increasingly more fit in a complex environment. A promising effort to convert the main characteristics of this process to an heuristic algorithm is given by John Holland (1975). This algorithm has had some striking preliminary success in the heuristic exploration of arbitrary high dimensionality nonlinear functions.

Such a research program was also suggested—albeit in more general terms—by Aumann (1985, pp.218–219) and by Binmore and Dasgupta (1986, pp.6–7, 12–14).

Axelrod then used the GA to "breed" strategies in the two-person

repeated PD game (Axelrod 1987, Forrest and Mayer-Kress 1991). Axelrod reported that the GA evolved strategy populations whose median member was just as successful as Tit for Tat, whom they closely resembled. (In 95% of the time, the evolved rules make the same choice as would Tit for Tat in the same situation.) In some cases the GA was able to evolve highly specialized adaptations to a specific environment of strategies which perform substantially better than does Tit for Tat in that situation. Miller (1989) and Marks (1989) have both extended Axelrod's recent work, and examine how the GA can be used in the breeding of strategies to such problems as the two-person PD with uncertainty ("noise") (Nalebuff 1987). This paper examines examples of oligopolistic markets, such as the three-person PDs of the price war (Fader and Hauser 1988).

The advent of GAs (and machine learning) means that a much more exhaustive set of potentially winning strategies can be generated by a single researcher, without the combined efforts of many competitors. This is because, within any given degree of "strategic complexity", *any* potential strategy is grist to the GA's mill, and will eventually be tested if it is a contender for best strategy, given the environment of competitors.

3. Modeling Oligopolistic Behaviour

3.1 Modeling Strategies in Repeated Games

IN ORDER to use the GA, we follow Axelrod and Forrest in modeling strategic behaviour as bit-string mappings, by first determining what the possible actions of the players are for any round; let us assume that the finite set of actions, S_i, for player i is unchanging and identical across players. Then player i's decision before each round is to choose an action (which may be a scalar or a vector) s_i from the set S_i of possible actions. If the competitive interaction among players is strategic, then each player's performance in each round is a function of his opponents' moves as well as his own. In the absence of information about the other players' decisions for the next round of play until they reveal their hands, their previous moves—which are known with certainty since we assume a game of perfect and complete information—provide the best information about their forthcoming moves. It is possible to look back at as many rounds as desired; we shall designate strategies that look back only one round as *one-round-memory strategies*, and so on.[1] Note that Tit for Tat is a one-round-memory strategy, and hence evidence that against many different strategies a long memory is not necessary for profitability.

In a strategic competition the action s_i must be contingent upon what

1. Aumann (1985, p.218) speaks of "states" of mind that depend "only on the previous state and the previous action of the other player"; the player's action then depends only on the new state. He also speaks of limiting the complexity of a strategy by limiting its memory, and presents (Appendix 5.5) a zero-round-memory strategy, called "memory zero". Marks (1990) discusses the relationship between complexity, bounded rationality, and finite-memory strategies.

the player expects his opponents to do themselves in the next round, an expectation which is a function of their moves in the past. So long as the action set for each player is finite, there is a finite set of events, Q_i, defined by the past actions of players over a given number of rounds.

If we denote the event or state of history that player i experiences before round t as $q_i(t) \in Q_i$, then we can model the decision of which action $s_i(t)$ to make at round t as a mapping from state $q_i(t)$ to action $s_i(t) \in S_i$. The state of player i one round later, $q_i(t+1)$, will include all the actions made in round t, and the consequent action of player i in round $t+1$ will be a function of $q_i(t+1)$.

These two steps for player i can be written as

$$s_i(t) = f_i[q_i(t)], \tag{1}$$

where f_i is the action function f_i: $Q_i \rightarrow S_i$ and,

$$q_i(t+1) = g_i[q_i(t),\ s_j(t)], \quad j \neq i, \tag{2}$$

where g_i is the next-state (or transition) function g_i: $Q_i \times S_j \rightarrow Q_i$, $j \neq i$. The next-state function is conceptually simple enough: in a one-round-memory strategy the previous state $q_i(t)$ ($= q_j(t)$) was simply the set of all players' actions in round $t-1$; $q_i(t+1)$ ($= q_j(t+1)$) is simply the set of all players' actions in round t. For strategies with longer memories, the state $q_i(t)$ can be thought of as a stack: the most recently occurring round's actions at the bottom, the oldest round's actions at the top; the next-state function pushes in the latest actions at the bottom and discards the forgotten round's actions.

Of course, we could include elements other than the players' actions in each player's state $q_i(t)$. For instance, the cumulative scores (undiscounted) were available to the programmers in the MIT tournaments. One possible strategy might be simply to ape the last round's action of the player with the highest cumulative score—to ride on his coat tails. The state might simply be the move of the most successful player last round, in which case the action function would be a simple one-to-one mapping.

What is described in this paper is a search in strategy space for a mapping from historic state to next action—the action function $f_i(\bullet)$—which results in the highest score in a repeated game. This corresponds to determining the most successful solution to a repeated oligopolistic game. One question to be answered will be the extent to which the dynamic nature of the game results in a coöperative solution, when the one-shot PD dictates the non-coöperative Cournot–Nash solution.

3.2 Strategies as Bit-String Mappings

Consider a game in which each player has to choose one of four possible courses of action (they could be prices themselves or they could be more complicated procedures for determining a price).[2] We can represent these 4 possibilities with a binary number of length 2 bits (where 00 is 0, 01 is 1, 10 is 2, 11 is 3). Now, the action function $f_i(\bullet)$ is a mapping from historical state to next-round action, so for each possible state there must correspond a 2-bit length of the binary string. If there are two other players, each also facing 4 possible actions (leading to 4 possible moves per player), then a one-round-memory strategy must allow for a possible number of states equal to $4^3 = 64$. The general rule is that the number of contingent states equals m^{rp}, where there are p players each with m possible moves per round and where each individual looks back r rounds.

In the example above, the complete mapping $f_i(\bullet)$ must associate a 2-bit length of binary string with each of 64 possible contingent states. If we are to have a unique correspondence, with no overlapping segments, then the minimum length of a binary representation of the action function mapping $f_i(\bullet)$ must be $64 \times 2 = 128$ bits. This mapping string will not alter through the repeated game, but the lengthening history will result in varying contingent moves. Depending on the competitive environment (the opponents' strategies), each mapping string $f_i(\bullet)$ will result in a score: the cumulative profit resulting from the outcomes of the rounds of the game. Thus, we are using the undiscounted limit-of-means criterion to select with—with a finite game, we are simply comparing the means of the strategies.

Although the mapping strings lend themselves to computer simulation and machine learning, they do not readily reveal to the human eye the class of strategies they model (nice, nasty, grudging, forgiving, generous, etc.). As soon as a string gets much over 16 bits long, such recognition is difficult. Perhaps using constructs from the theory of finite automata will assist (Miller 1989, Marks 1990); perhaps there is no way to characterize a complex strategy: the only way to understand it is to watch its behaviour in a repeated game.

This representation allows us to use the GA to develop what is in effect a machine-learning process to search for strings which are ever more successful at playing the repeated game. As explained by Schaffer and Grefenstette (1988), our process can be classified as an example of the Pitt approach to machine learning (Smith 1981), in which each string is evaluated for evolutionary fitness (its score in the repeated game), and this score is used to control the selection of strings used to generate a new set of strings. The particular GA we use is Grefenstette's GENESIS (1987). (See a discussion of genetic algorithms in the Appendix.)

4. Niche Strategies

2. The action function maps from contingent state to next-round action. We have spoken of the action as identical with the player's next-round move, but this is not necessarily so; the action could be a procedure to calculate the next-round move. For example, there could be two actions: (1) next-round move = arithmetic mean of all players' last-round prices, or (2) next-round move = geometric mean of all players' last-round prices. The number of possible actions is less than the number of possible moves in this case. Of course, this is simply a model of a two-state decision process: use the binary string to determine the second-stage process for determining the next-round move. It demonstrates, however, that there is no reason why the number of a player's possible moves should be equal to the number of his possible actions. Indeed, the action could be a vector: a numbered process plus a parameter, for instance.

CONSIDER a repeated PD game: each player has two choices: coöperate C or defect D, so the choice can be represented by a single bit: 0 for C and 1 for D. With a one-round memory, and only considering the moves of each player, the event space contains four possibilities (CC, DC, CD, or DD), where XY means that one's last move was X, one's opponent's was Y. With four events, each mapping to a single bit to determine the next move, the bit string will be 4 bits long, resulting in $2^4 = 16$ possible strategies. For instance, the string (0111) means that one will coöperate if both players coöperated last round, otherwise one will defect; the string (0011) is Tit for Tat: one mimics one's opponent's last move; (1111) means always defect, whatever one's opponent's last move.

It can be shown that a game of a fixed number of rounds N will result in a score equal to that obtained in a game of uncertain length, where w is the probability that any round is *not* the last, in a risk-neutral world. This probability can in turn be shown to be equivalent to an implicit discount rate r. This enables us to relate the length N of a fixed-rounds game to an implicit discount rate r. A payoff of R units for an N-round game equals NR units. The expected payoff of a game in which the probability of continuing is w is $R(1 + w + w^2 + w^3 + \cdots) = R/(1 - w)$. Equating these, we see that $w = (N-1)/N$. If we think of w as a discount factor, then the implicit discount rate r is given by $r = (1-w)/w$ or $r = 1/(N - 1)$. (This can also be obtained by considering NR as the present value of an infinite flow of R discounted at r per round.)

Marks (1989) reports on machine-learning solutions of a repeated two-person PD (Figure 1) with three-round memory, replicating Axelrod (1987). The values are constrained by $T > R > P > S$, and in general by $2R > T + S$. In this model, there are 64 ($= 4^3$) possible states, corresponding to the 4 possibilities of action in each of the last 3 rounds. Since the action (C or D) can be modeled by a single bit, the complete mapping can be modeled by a string of length 64, corresponding to $2^{64} \simeq 10^{20}$ possible combinations. An additional 6 bits are used to model the "phantom memory" of unplayed rounds for the first three rounds, following Axelrod and Forrest.

R, R	*S, T*
T, S	*P, P*

Figure 1. The Simple Prisoner's Dilemma

We used Axelrod's "niche" environment of five rules (Axelrod 1984, p. 199) and sought a "better" (higher scoring) strategy than Tit for Tat. We used his values of $T = 5$, $R = 3$, $P = 1$, and $S = 0$. With 151-round games ($w = 0.9934$, $r = 0.67\%$ per round),[3] our benchmark scores in this niche were:

3. We used 151-round games because Axelrod (1987) did so, following the mean length of the games in his second (1984) tournament, in which the number of rounds per game was a random variable, to preclude end-game plays, as discussed below.

Always Defect: 223.980
Always Coöperate: 369.768
Tit for Tat: 382.392

Tit for Tat outscored both the ultra-nice Always Coöperate and the ultra-nasty Always Defect. After breeding a population of fifty 70-bit-string strategies for 2,000 generations—a total of 100,000 trials, each trial resulting in a weighted average of the scores of 151 rounds of the repeated symmetric PD against each of the five "niche" strategies—the best individual strategy scored 394.0348, and appeared after trial 79,083, in the 1,581st generation. Since two of Axelrod's five niche strategies were non-deterministic, the apparent superiority of the new strategy may not be statistically significant, but nonetheless provides an insight into the structure of a possibly Tit-for-Tat-dominating strategy.

On examination the winning structure was very similar to a three-round-memory Tit for Tat. (Recall that Tit for Tat requires only a single-round memory.) The difference is that a coöperative C on the part of the opponent following two defections D in the immediately preceding rounds is not sufficient to elicit a coöperative C from the strategy: two successive Cs are required. "Grudging Tit for Tat"—as the strategy was dubbed—would forgive a single defection by its opponent if it was followed by a C, as would Tit for Tat, but two Ds would require two Cs before it would also coöperate again.

Miller (1989) also reports an attempt to use the GA to breed niche strategies against Axelrod's environment; he models strategies as finite automata, rather than the action functions described above, which can model "trigger" strategies, but are difficult to extend to three-person games. Miller also considers bootstrapping evolution.

5. Bootstrapping Evolution

AXELROD (1987) pointed out that any strategies bred using the GA would be highly adapted to the particular "niche" defined by the rules of their competitors. Thus, each simulation session would be unique, up to the level of definition of the niche rules. And yet the literature of repeated games has been concerned with examining the extent to which repetition results in coöperation. The GA can be used to explore the extent to which this is true, for particular games as models of market interactions.

In the one-shot PD game the Cournot–Nash non-coöperative equilibrium dominates the Pareto-superior coöperative solution. This result generalizes to n-player games and provides a rationale for price wars when there are a small number of sellers of differentiated products, as the MIT tournaments modeled. With a simple game played between two opponents for more than a single round, the possibility of responding to an opponent's defection in the previous round with a defection in this and later rounds raises the possibility that the threat of defection may induce mutual coöperation. But for games of finite duration with low discount rates (we can use the mean of all rounds for the game score or the discounted present value of the rounds' results) this hope is dashed by the end-game behaviour, or what Selten (1975) called the "chain-store paradox". In his second open computer tournament, Axelrod (1984) chose the number of rounds per game probabilistically, with a 0.00346 chance of ending with each given move (or $w = 0.99654$), in order to eliminate submission of strategy algorithms which might exhibit end-game behaviour. In the genetic-algorithm tournaments, so long as the number of rounds played is greater than the number of rounds remembered, there can be no such behaviour. We used 22-round games ($w = 0.9545$) for convenience, except when the implicit discount rate ($r = 4.76\%$ per round) was too low, as discussed below.

There is a discontinuity for infinitely repeated games (or supergames): the Folk Theorem (Aumann 1989) tells us that any individually rational payoff vector can be supported in infinitely repeated games, for sufficiently low discount rates. (For high discount rates the threat of future punishment may not be sufficiently great to offset the gain from defecting now.)

In order to explain the apparent evidence of coöperative behaviour among oligopolists in the real world, among experimental subjects in clinical trials, and among strategy simulation tournaments—all of them examples of finite repetitions—researchers have sought relaxation of the underlying assumptions in the finite game. Radner (1980, 1986) assumed a type of bounded rationality similar to satisficing. Kreps et al. (1982) assumed incomplete information—they relaxed the assumption that rationality is "common knowledge" (Aumann 1976) among the players. Neyman (1985) and Radner (1986) argued that limited complexity of players' strategies, and Harrington (1987) argued that limited complexity of players' beliefs, could result in the emergence of coöperation. J.W. Friedman (1971) and Sorin (1986) showed that a sufficiently high discount rate was sufficient. Fudenberg and Maskin (1986) extended the proofs in the infinitely repeated case to games of three or more players. See Marks (1990) for further discussion of this isse.

As Binmore and Dasgupta (1986) suggest, an evolutionary competition among game-playing programs provides an avenue for linking prescriptive game theory with descriptive game theory: in the long run not quite all of us are dead, only those who were unsuccessful in the repeated game—some genes (combinations of zeroes and ones in the binary string) of those who scored well survive in their descendents. This provides a *learning* model in which it is the generations of populations of strategies that learn, not individuals, which are immutable strings of bits. Samuelson (1988) provides a theoretical framework for examining the processes of the evolution of strategies, at least for finite, two-person, normal-form games of complete information. He proves that, under certain properties of the evolutionary process, equilibrium strategies will be supported that are "trembling-hand perfect" (Selten 1975, 1983; Binmore and Dasgupta 1986), a subset of Cournot–Nash equilibrium.

Our results support the contention that "repetition breeds coöperation", at least for two-person games with unique Nash–Cournot equilibria. Our method is to "breed" populations of strategies (our binary mapping strings), where each individual strategy in a population of strategies is pitted against all other strategies (or combinations of strategies in three-person games) to obtain a "fitness" score for each strategy. This *bootstrap* breeding, together with the GA's search properties, should result in "evolutionary" convergence to the optimum optimorum of all possible strategies. (There is some doubt

whether all loci will be optimally selected for: an individual emerging into a population of similar strategies will not experience much opportunity to respond to hugely different strategies, and over time there may be genetic drift, as the descendents lose some traits previously strongly selected for.[4] The consequences of this for the possibility of invasions are discussed below.)

As a consequence of the GA's processes, in bootstrapping we speak of convergence to *behaviour*, not to *structure*: when, amongst themselves, the population of strategies all play the same action for the duration of each repeated game and for all possible combinations, we say that the population has *converged*. We examine in Section 6 the resistance of these converged populations to the introduction or invasion of new strategies from outside.

5.1 The Simple Prisoner's Dilemma

In a simple, two-person, symmetric PD with perfect information and the Axelrod payoffs in a repeated game amongst one-round-memory strategies (using a 6-bit string: 4 bits for the contingent states, and 2 bits for the phantom memory used in the first round of a 22-round game, that is, $w = 0.9545$, $r = 4.76\%$ per round), the population of 50 individuals converged[5] from a random distribution of bit strings to a population supporting the coöperative equilibrium (C,C) in 22 generations.

The bootstrap evolution was repeated for two-round-memory strategies, with their more subtle strategic possibilities. (They use a 20-bit string: 16 bits for the 4×4 contingent states, and 2×2 bits for the phantom memory.) The population of 100 individuals converged from a random distribution to the uniform coöperative behaviour of (C,C) in 61 generations.

5.2 Extended Prisoner's Dilemma Games

A more realistic PD might allow players to shade their coöperation or defection (To 1988) by choosing actions with payoffs between the two extremes of the simple game. For instance, the row player's payoff matrix of Figure 2 is a superset of the simple PD payoff matrix, and allows a greater variety of strategies, even only with one-round memory, which is reflected in the longer mapping string. With 4 possible actions, each action must be coded with a 2-bit segment, and there are 16 possible states with one-round memory, corresponding to the 4×4 payoff matrix above. The strings must be 36 bits long: 2×4^2 for the next action, plus 2×2 strings for the phantom memory. Starting from random strings, a population of 50 strategies converged to coöperative behaviour (A, A) after 419 generations of bootstrapping, using 22-round games ($w = 0.9545$, $r = 4.76\%$ per round).

4. It has been suggested (Goldberg and Smith 1987) that the recessive genes of diploid genotypes are a reservoir of stored information that proved fit in earlier environments, and that GAs which included diploidy and dominance will thus perform better in "noisy" environments than does GENESIS, which utilizes haploid genotypes.
5. By *converging* to a uniform population, we mean that the process first attains a uniform population of strings—there may be subsequent generations which are non-uniform, as the recombinant operators generate new individuals, as further bits and bit combinations are tested—not the time of apparent stability, some generations later.

	A	*B*	*C*	*D*
A	27, 27	18, 33	9, 39	0, 45
B	33, 18	23, 23	13, 28	3, 33
C	39, 9	28, 13	17, 17	6, 21
D	45, 0	33, 3	21, 6	9, 9

Figure 2. An Extended Two-Person Prisoner's Dilemma

Following Sonnenschein (1989), we consider a more interesting two-person game with three alternative actions, call them L, M, and H. The payoff matrix of Figure 3 reveals that (M, M) is the unique Nash equilibrium: given that one's opponent plays M, the best that one can do oneself is to play M too. (Note that because we must use two bits to code for action, we have to include payoffs for a fourth possibility, O—large negative payoffs will select against this possibility.)

	L	*M*	*H*	*O*
L	15, 15	5, 21	3, 10	5, –50
M	21, 5	12, 12	2, 5	12, –50
H	10, 3	5, 2	0, 0	5, –50
O	–50, 5	–50, 12	–50, 5	–50, –50

Figure 3. Profits: Monopoly L, Cournot M, and Competitive H

A bootstrapped population of 25 one-round-memory strategies (36-bit strings) converged from random to the coöperative solution (L, L) after 33 generations of 22-round games, showing that repetition can lead to the *L*ow output–high profit equilibrium.

An alternative score to the average payoff per game is the discounted present value of the payoffs. Sonnenschein (1989) shows that when the discount rate per round is high—he uses ¾—the present value of the future costs imposed by one's opponent in response to one's preëmptive defection is less than the immediate gains from a defection. That is, a defection from (L, L) by one player will garner him 6 units now, at the cost of 3 units per future round forgone indefinitely if his opponent defects in the next and succeeding rounds. The present value of an annuity of 3 units discounted at ¾ is 4 units. So the present value of the payoff is increased by defection, and the coöperative equilibrium of (L, L) cannot be supported.

As reported above, in a 22-round game with an implicit discount rate of 4.76% per round and using average scores, a population of 25 random 36-bit strings converged to the coöperative solution of (L, L) in 33 generations. With an explicit additional discount rate of 80% per round in the 22-round games, the coöperative equilibrium was not supported: an identical population of 25 random strings converged to the Pareto-inferior Cournot–Nash solution of (M, M) in 31 generations. This is in accord with Sonnenschein's argument.

5.3 The MIT Competitive Strategy Tournaments

The simplest three-person game is a repeated PD with one-round memory. Each round corresponds to one of eight possible states (*CCC*, *CCD*, *CDC*, *CDD*, *DCC*, *DCD*, *DDC*, *DDD*), where *XYZ* means that one's last move was *X*, one's first opponent's was *Y*, and one's second opponent's *Z*. Since, in the simple PD, one's choice is dichotomous, the mapping string from state to action need only be 8 bits long. An example is the string (01110111), which models one's strategy of coöperating (0) only when both of one's opponents coöperated in the last round (whatever one did oneself), otherwise defecting (1).

In November 1984, the MIT Marketing Center in the Sloan School of Management announced a three-person repeated game, in which participants were invited to submit a strategy, the outcome of which was one's price in the next round of the repeated game, given complete knowledge of one's own previous moves (prices), one's own cumulative score (total profits, undiscounted), and the previous moves and scores of both other players (Fader and Hauser 1988). One reason was to explore how Axelrod's two-person results generalize to more complex and managerially relevant situations.

The model is of sales of differentiated goods. With prices of P_i, P_j, and P_k, the payoff π_i for any player i is given by:

$$\pi_i = 3375\,(P_i - 1)\,P_i^{-3.5}\,P_j^{0.25}\,P_k^{0.25} - 480, \qquad (3)$$

where $i \neq j \neq k$. This corresponds to a constant-elasticity-of-demand, constant-returns-to-scale differentiated triopoly (Fader and Hauser 1988). The payoff of equation (3) results in a "coöperative" price of $P^o = \$1.50$, the joint maximization price (Shubik 1980), which would result from collusion, and in a "defect" price of $P^* = \$1.40$, the non-coöperative Cournot–Nash price, which maximizes the payoff independent of the others' prices. Two other prices are the two-player coalition price, P^c, which maximizes the profits of two colluding firms independent of the third firm's price, $P^c \simeq \$1.44$; and the "envious" price, P^e, which maximizes the firm's share of total profits, $P^e \simeq \$1.36$.

In 1986, a new tournament was announced, in which with the prices P_i, P_j, and P_k the profit function for any player i was:

$$\pi_i = 200\,(8 - 6\,P_i + P_j + P_k)(P_i - 1) - 180, \qquad (4)$$

with $i \neq j \neq k$. This profit function corresponds to a linear-demand, differentiated triopoly. This new function was chosen because it does not yield unique values of the Cournot–Nash price, P^*, or the two-person coalition price, P^c, which means that creating and maintaining two-person coalitions will be harder.

Both contests may be considered as oligopoly markets, with three sellers who compete with price: if they could collude, then the joint maximization price, P^o, maximizes the profit of each, but independent profit-maximization results in the Cournot–Nash price, P^*. The price war has strong elements of the PD: defection (price cutting) dominates collusion (pricing at the joint-maximization level), at least in the one-shot game, but if the game continues, then the threat of a continuing price war may result in all three choosing the joint maximization price, or near to it.

We could model each player's action as choosing a price in cents between \$1.36 and \$1.51, 16 possible actions. Each action can be coded by a 4-bit binary number, 0000 $\rightarrow$ \$1.36 and 1111 $\rightarrow$ \$1.51. We could model the possible states as the triple of prices from the previous round: a one-round memory. With three players and 16 possible prices, there are $16^3 = 4{,}096$ possible states. The mapping string for this model would be $16^3 \times 4 = 16{,}384$ bits long, which models $2^{16{,}384} \simeq 10^{4{,}932}$ possible strategies.

In practice, we might want to use some knowledge of the structure of the payoffs to reduce both the number of possible actions and the number of distinct states. For instance, we might reduce the number of actions from 16 to, say, the 4 price points: P^o, P^*, P^e, and P^c, perhaps with a further "shading" action: up a little, down a little, or steady. This would require 2 bits for the price, and another 2 bits for the shading. (This would provide enough bandwidth for $2^4 = 16$ actions again, even though we only use $4 \times 3 = 12$—could this redundancy be used?) We could similarly look at 12^3 possible states: for each player, is he in (at, above, below) one of the four price-point regions? This gives 4×3 possibilities per player, so the number of possible states is $12^3 = 1{,}728$. This means a string of length $1{,}728 \times 4 = 6{,}912$, which models $2^{6{,}912} \simeq 10^{2{,}081}$ possible strategies, still a heap of possibilities!

If we abandon shading, then there are 4 possible actions (which require 2 bits), and $4^3 = 64$ possible states or events, resulting in a string of length $64 \times 2 = 128$ bits, which models $2^{128} \simeq 10^{39}$ possible strategies. To simplify the problem, we consider only a dichotomous three-person game, in which each player has to choose whether to price at the coöperative level (\$1.50: *C*) or at the non-coöperative level (\$1.40: *D*). In this case the bit string will be 8 bits long, plus 3 bits for the phantom memory.

The payoff matrix of Figure 4 has been calculated from the payoff function of equation (4), from the second MIT tournament—although the payoffs for the same price combinations are very similar for the function of equation (3).

	C	***D***	***C***	***D***
C	20	10	10	0
D	28	20	20	12
	C		***D***	

Figure 4. First Player's Payoff Matrix

In Figure 4 the payoff to defecting with one other player results in the same payoff (\$20) as does the three-way collusion of (*C*,*C*,*C*); in Figure 5 this has been reduced slightly. For both Figures, Player 1 chooses the row, Player 2 the column, and Player 3 the matrix.

A bootstrapped population of 25 one-round-memory strategies (11-bit strings) playing 22-round ($w = 0.9545$, $r = 4.76\%$ per round), three-person PD games converged from random to the one-shot Nash non-coöperative behaviour of (*D*, *D*, *D*): this took 31 generations for the payoffs of Figure 4 and

	C	***D***	***C***	***D***
C	20	10	10	0
D	28	15	15	12
	C		***D***	

Figure 5. First Player's Payoff Matrix

23 for Figure 5. A bootstrapped population of 50 two-round-memory strategies (70-bit strings) playing 22-round, three-person PD games also converged from random to (*D*, *D*, *D*), taking 27 generations for the payoffs of Figure 5.

These results were at first disturbing: the GA was apparently not finding the global optimum. The implicit discount rate *r* of a 22-round game is 4.76% per round. Perhaps this rate is too high to support the coöperative (*C*,*C*,*C*) behaviour as an equilibrium? The length of the repeated game was increased three-fold to 66 rounds, increasing *w* to 0.9848 and lowering the implicit discount rate *r* to 1.54% per round, and the answer was: yes. An identical random population of the 25 one-round-memory strategies converged to the coöperative equilibrium after 35 generations, again using the payoffs of Figure 5. Apparently both the theory of repeated games and the GA are vindicated.

6. Invasion and the Trembling Hand

EARLY work by biologists on the emergence of coöperation in animal populations (Maynard Smith 1982) was also concerned with the *evolutionary stability* of strategies (or genetically determined behaviour traits): their ability to survive in the face of an "invasion" by other strategies. Our formulation allows precise and unambiguous simulations to be made of such occurrences by use of a non-random initial population of strategies that has been seeded with any desired ratio of incumbents to specific invaders. The invaders can be any of the strategies possible within the particular formulation used.

Moreover, the "convergence" spoken of above is related to the general concept of the ability of a population, over several generations, to respond to the emergence of new strategies—by the genetic recombinations of mutation and crossover or by exogenous invasion—either by successfully out-competing the new strategies, which will die without issue, or by interbreeding with the successful newcomers, so that, over several generations, the successful genes spread through the new generations of offspring.

Binmore and Dasgupta (1986, pp.16–19) argue that the equilibrium concept that Selten (1975) calls *perfect* equilibrium but that they call *trembling-hand* equilibrium[6] is relevant to the discussion of stability to invasion—it is also relevant to the convergence of the GA's evolutionary process towards a uniform payoff (that is, uniform behaviour in the present generation). Roughly speaking, a Nash equilibrium for any game is a trembling-hand equilibrium if each of its component strategies remains optimal even when the opponents' hands "tremble" as they select their equilibrium strategies.

Consider a two-person symmetric game, the scores of which will be inputs to the GA in generating the new set of offspring strategies. Suppose a population of individual strategies A is invaded by a small number of strategies B. Let ε be the proportion of Bs in the total population of Bs and As. It will then be as though each player were facing an opponent using a mixed strategy—choosing A with probability $(1 - \varepsilon)$ and B with probability ε—as Binmore and Dasgupta argue, the opponent may be regarded as a player who selects A "but with a trembling hand". We discuss the results of deliberate introductions of Bs into populations of As below.

Maynard Smith's evolutionarily stable equilibrium demands that

$$U[A,\ (1 - \varepsilon)A + \varepsilon B] > U[B,\ (1 - \varepsilon)A + \varepsilon B],$$

for all sufficiently small ε, where $U(X, Y)$ is the score of X against Y. If A is an evolutionarily stable strategy (ESS), then a population of As is immune to invasion by a small group of Bs. Boyd and Lorberbaum (1987) argue that no *pure* strategy can be evolutionarily stable in a repeated PD. They argue that no strategy whose behaviour during the nth round is uniquely determined by the history of the game up to that point is evolutionarily stable (that is, has a higher expected fitness than any rare invading strategy) if w, the probability of *not* ending the game, is sufficiently large. That is, if w is sufficiently large—if the game continues for a sufficient number of rounds without discounting—then no strategy can be best against all opponents. Nachbar (1988*b*) and D. Friedman (1991) have examined the relationship between ESS and Nash equilibrium in both static and dynamic processes. A closer study of the convergence and stability of our evolutionary processes demands a closer study of the mechanisms of the GA, and perhaps an acceleration of convergence through better selection mechanisms. This must await a later paper.

6.1 *Invasion of an Extended Prisoner's Dilemma*

We have reported above the bootstrapping of the repeated game whose payoff matrix is shown in Figure 3, and its sensitivity to the number of rounds per game (that is, to the probability w of not stopping before the next round). As reported, with the high explicit discount rate of 80% per round, the coöperative equilibrium was not supported in a bootstrap evolution of 25 random one-round-memory strategies (36-bit strings). After convergence to

6. They prefer *trembling hand* to *perfect* in order to clearly distinguish the concept from another of Selten's: *subgame-perfect* (Binmore and Dasgupta 1986, fn.18). All trembling-hand equilibria are subgame perfect, but the converse is not true. See also Selten (1983).

the non-coöperative equilibrium of (M, M), the explicit discount rate was switched to zero. For 20 further generations the non-coöperative equilibrium appeared stable.

6.2 *Invasion of the Three-Person Game*

Several simulations of invasion were performed with the three-person games of Figures 4 and 5, using one-round-memory strategies (11-bit strings). In both cases, with 22-round games (w = 0.9545, r = 4.76% per round), an initial population of 24 ultra-nice Always Coöperate strategies (00000000000) was successfully invaded by a single ultra-nasty Always Defect (11111111111); it took 39 generations for convergence to (D, D, D) with the payoffs of Figure 4, and only 23 generations with those of Figure 5.

Using the payoffs of Figure 5, an initial population of 24 Always Coöperates was seeded with a single strategy (10000000000) that defects when everyone has coöperated on the previous round, but otherwise coöperates. After 7 generations all strategies were alternating between C and D, but after 18 generations the invasion was complete: all strategies were at the non-coöperative equilibrium, (D, D, D).

As remarked above, we found that lengthening the number of rounds from 22 to 66 (w = 0.9848, r = 1.54% per round) permitted support of the coöperative equilibrium (C, C, C), but when we repeated the invasion of 24 Always Coöperates by one Always Defect we obtained convergence to the non-coöperative equilibrium, which demonstrates again that although a larger number of rounds (and hence a higher value of w) enables support of the coöperative solution, it does not *ensure* its emergence, the final convergence still being a function of the initial population of strategies.

7. Conclusions

THERE has been much recent theoretical work on the outcomes of repeated games, both infinite and finite. In particular, researchers have sought to answer the question: does repetition in the finite game result in a greater degree of coöperation than will occur in a one-shot game? This work does have policy implications: if the conditions for coöperative behaviour to emerge as a stable equilibrium of repeated encounters ("games") in the absence of outside enforcement are not too restrictive, then apparent market collusion may be just that: apparent. If the conditions are not too restrictive, then coöperative or collusive behaviour in markets will not be sufficient evidence of clandestine agreements to collude. These conclusions are not new. What is new with this paper is a method for cutting through the theoretical (and very technical) expositions: we have provided a means of modeling strategies in repeated games, the degree of complexity of which is only limited by the imagination of the modeler. The GA of machine learning provides a technique for efficiently selecting from the immense number of possible strategies those that perform best, as scored in the repeated game. Moreover, the number of players is no conceptual limit—as computing speeds increase, the complexity of problems amenable to solution will become ever more realistic. Optimal solutions to the MIT tournaments will be published in a future paper.

The GA is a parametric optimization technique, but the parameters are coded as binary strings, and so permit a much greater range of possible solutions than are possible with calculus-based methods. Indeed, a program for future research is to characterize families of strategies parametrically, so that optimal strategies can be sought, with fewer constraints than traditional techniques. The contingent-action strategies we have considered here is one possibility, Fujiki and Dickinson's LISP "grammar" is another.[7]

Utilizing these modeling techniques and the GA for solving the selection of optimum strategies, we have reported results similar to Axelrod's for the simple symmetrical two-person PD. We have used "bootstrapping"—breeding strategies against their peers—to examine the emergence in repeated games of stable equilibria, whether coöperative (as in infinite supergames) or non-coöperative (as in the one-shot PD). We find that, so long as discounting is sufficiently low (both explicitly in the scoring function and implicitly from the game length), coöperative equilibria are supported, as theory would suggest.

To what extent is the convergence of the machine-learning search process of the GA a model of real-world equilibrating behaviour? Given the simplicity of our models, probably only slight, but this convergence is related to theoretical notions of stability in evolutionary processes and to the susceptibility of populations of strategies to the invasion of new strategies, whether spontaneously generated by the recombinant operations, or exogenously introduced. Since explicit examination of the changing membership of the population of strategies is possible, the convergence process—including the fitness scores of specific strategies—can be closely monitored. This awaits future study. This paper reports on successful and unsuccessful invasions of exogenously introduced strategies.

The author's hope is that this paper demonstrates that the research programs outlined by both Aumann (1985) and Binmore and Dasgupta (1986) are underway—Holland's GA provides a powerful tool for the continued study of strategies for repeated games. The strategies cannot collude since they are nothing more than stimulus–response machines, and yet even in three-person interactions, such as oligopoly markets, coöperative behaviour can occur.

There is no conceptual reason—although CPU time may provide a constraint—why these modeling and solution techniques cannot be used for examining games which are closer to market situations. Nalebuff (1987) has asked how robust Tit for Tat would be in the case in which there was not perfect information about one's opponent's past moves, or at least in which the players mistakenly believed that they possessed perfect information about each other's past moves. So long as the simulation provided sufficient statistical power for selection of bits in the mapping string, and given that the model could accommodate strategies of sufficient subtlety, the machine-

7. Fujiki and Dickinson (1987) describe using the GA to generate programs written in Lisp to "solve" the repeated PD—this is much more complex than our binary strings. Using a "grammar" of possible strategies, they found that against Axelrod's environment (Axelrod 1984) the strategy known as Tit for Two Tats scored best, and that when bootstrapping the best strategy was the "trigger" strategy of coöperating until first defected against, and then always defecting.

learning techniques should provide an answer to Nalebuff's challenge. Indeed, Miller (1989) has bred finite automata strategies in noisy games.

We have discussed above relaxing three of the features of simple models: (*a*) strategies with longer than one-round memories, (*b*) games with more than two possible actions per player, and (*c*) games with more than two players. With each of these relaxations our models (slowly) come closer to modeling reality. A further relaxation might (Midgley 1988) be (*d*) partitioning the players into two or more groups, each with a distinct payoff matrix or function—this raises the interesting question of how the market system will behave with each group facing a changing set of opponents (in a three-person game) and facing a distinct payoff function. There may not be convergence to a stable equilibrium, but then that may model the real world.

Appendix: Genetic Algorithms[8]

IN SECTION 3.2 we have seen how strategies (sets of rules) for playing repeated games of the Prisoner's Dilemma can be represented as bit strings of zeroes and ones, each locus or substring (or gene) along the string mapping uniquely from a contingent state—defined above by all players' moves in the previous round or rounds of the repeated game—to a move in the next round, or a means of determining this next move.

We describe these strings as "chromosomes" because GAs use selection and recombinant operators—crossover and mutation—derived by analogy from population genetics in order to generate new sets of strings (a new generation of "offspring") from the previous set of strings. Brady (1985) notes that "during the course of evolution, slowly evolving genes would have been overtaken by genes with better evolutionary strategies," although there is some dispute about the extent to which such outcomes are optimal (Dupré 1987). The GA can be thought of (Bethke 1981) as an optimization method which overcomes the problem of local fitness optima, to obtain optima which are almost always close to global. Moreover, following biological evolution, it treats many candidate solutions (individual genotypes) in parallel, searching along many paths of similar genotypes at once, with a higher density of paths in regions (of the space of all possible solutions) where fitness is improving: the "best" individual improves in fitness and so does the average fitness of the set of candidates (the population).

Hereditary models in population genetics define individuals solely in terms of their genetic information: the genetic structure of an individual—or genotype—is represented as strands of chromosomes consisting of genes, which interact with each other to determine the ultimately observable characteristics—or phenotype—of the individual. A population of individuals can be viewed as a pool of genetic information. If all individuals in the population have equal probability of mating and producing offspring, and if

8. For an introduction to GAs, see Goldberg (1988). Also see Davis (1991) and Rawlins (1991). For another application of genetic algorithms to economics, see Marimon et al. (1990).

the selection of mates is random, then the information in the gene pool will not change from generation to generation. But environmental factors affect the fitness of phenotypes of individuals, and hence affect the future influence of the corresponding genotypes in determining the characteristics of the gene pool—the principle of natural selection, which results in a changing gene pool as fitter genotypes are exploited. Natural selection can be viewed as a search for coädapted sets of substrings which, in combination, result in better performance of the corresponding phenotype (the individual's behaviour) in its environment.

Schaffer and Grefenstette (1988) argue that the theory of GAs derived by Holland (1975) predicts that substrings associated with high performance will spread through the new populations of bit strings. Paraphrasing Holland (1984), a GA can be looked upon as a sampling procedure that draws samples from a potential set T. With each sample is associated a value, the fitness (or score) of the corresponding genotype (or fundamental hereditary factors). Then the population of individuals at any time is a set of samples drawn from T. The GA uses the fitness (scores) of the individuals in the population at each generation to "breed" and test a new generation of individuals, which may include the best individuals from the previous generation. The new generation is "bred" from the old using genetic operators: selection of parents according to their fitness, crossover of genetic material from both parents, and random mutation of bits. This process progressively biases the sampling procedure towards the use of combinations of substrings associated with above-average fitness in earlier generations (that is, sample individuals characterized by higher scores because their behaviours are "better"), so the mean score of successive generations rises owing to selective pressures. A GA is all but immune to some of the difficulties that commonly attend complex problems: local maxima, discontinuities, and high dimensionality.

The GA is an answer to the problem of obtaining a robust and efficient use of information contained in a limited amount of experience: the difficulty is the low level of confidence that can be placed on inferences from limited samples. The GA, so Schaffer and Grefenstette (1988) argue, results in a near-optimal trade-off between exploration (gaining more reliability from more experience) and exploitation (using the information from past experience to generate new trial structures). The GA can be described in essence as (1) producing the initial population $P(0)$ (randomly or using some seeding method); (2) evaluating each trial structure of the population $P(t)$ at any time t; (3) selecting the fittest trial structures, as measured by their scores (in our case in the repeated games); (4) applying the genetic recombination operators to the selected parent structures to produce the offspring generation, $P(t+1)$; (5) testing against a stopping rule—if YES, then stop, if NO, then return to stage (2).

The initial population $P(0)$ is usually chosen at random, but can be constructed to contain heuristically chosen initial strings. In either case, the initial population should contain a wide variety of structures. The structures of the population $P(t+1)$ are chosen from the population $P(t)$ by a randomized "selection procedure" that ensures that the expected number of times a structure is chosen to be a "parent" is proportional to that structure's performance, relative to the rest of the population. That is, if unique

structure x_j has twice the average performance of all the structures in $P(t)$, then x_j is expected to appear twice in the mating pool. At the end of the selection procedure, the mating pool contains exact duplicates of the selected structures in population $P(t)$, in proportion to their share of the aggregate of fitness scores.

Although realizations of the GA differ in their methods of survival selection, of mate selection, and of determining which structures will disappear, and differ in their size of population and their rates of application of the different genetic operators, all exhibit the characteristic known as *implicit parallelism*. Any structure or string can be looked at as a collection of substring components or schemata which together account for the good or bad performance of the individual structure. Then Holland's Schema Sampling Theorem (Holland 1975, Davis 1991) demonstrates that schemata represented in the population will be sampled in future generations in relation to their observed average fitness, if we can assume that the average fitness of a schema may be estimated by observing some of its members. (Note that many more schemata are being sampled than are individual structures of the population being evaluated.) Genetic algorithms gain their power by searching the space of all schemata and by quickly identifying and exploiting the combinations which are associated with high performance.

This can be formalized. A population of binary structures of length L bits can be viewed as points in an L-dimensional space. Genetic algorithms search for better structures by focusing on partitions (hyperplanes) of this space associated with good performance or high fitness. A kth order hyperplane $(0 \le k \le L)$ is defined as an $(L-k)$-dimensional subspace, and is specified by assigning values to only k of the L string positions or loci, the rest being filled with the # symbol. Let H be a hyperplane in the representation space. Let $M(H,t)$ denote the number of individual structures in $P(t)$, the population at time t, that are members of the hyperplane H. For example, y_2 below is a member of the hyperplane #1001###. Holland (1975) has shown that the effect of selection alone (in the absence of other genetic operators) is that

$$M(H,t+1) = \frac{\mu(H,t)}{\mu(P,t)} M(H,t), \qquad \text{(A1)}$$

where $\mu(H,t)$ is the average fitness of the structures or schemata that are in both $P(t)$ and in H, and where $\mu(P,t)$ is the average fitness of all structures in $P(t)$. Thus, the number of samples allocated to a hyperplane H changes exponentially over time, growing for the above average and dwindling for the below average. The relationship in equation (A1) applies to each hyperplane represented in the population. In general, in a population of N binary structures of length L bits, between 2^L and $N 2^L$ distinct hyperplanes are available for sampling. Genetic algorithms test and search for these high-performance substrings or schemata, which far outnumber the individual structures in the population at any time, since a single structure is an instance of 2^L distinct hyperplanes $(\Sigma_{k=0}^{L} {}_L C_k)$.

The most important recombination operator is *crossover*. Under the crossover operator, two structures in the mating pool exchange portions of their binary representation. This can be implemented by choosing a point on the structure at random—the crossover point—and exchanging the segments to the right of this point. For example, let two “parent” structures be

$$x_1 = 100{:}01010, \quad \text{and}$$
$$x_2 = 010{:}10100.$$

and suppose that the crossover point has been chosen as indicated. The resulting “offspring” structures would be

$$y_1 = 100{:}10100, \quad \text{and}$$
$$y_2 = 010{:}01010.$$

Crossover serves two complementary search functions. First, it provides new strings for further testing within the structures already present in the population. In the above example, both x_1 and y_1 are representatives of the structure or schema 100#####, where the # means “don’t care, because the value at this position is irrelevant.” (If 1001 is a point, then 100# is a line, and 10## is a plane, and 1### is a hyperplane.) Thus, by evaluating y_1, the GA gathers further information about this structure. Second, crossover introduces representatives of new structures into the population. In the above example, y_2 is a representative of the structure #1001###, which is not represented by either “parent.” If this structure represents a high-performance area of the search space, the evaluation of y_2 will lead to further exploration in this part of the search space. The GENESIS package (Grefenstette 1987), which we use, implements two crossover points per mating.

A second operator is *mutation*: each bit in the structure has a chance of undergoing mutation, based on an interarrival interval between mutations. If mutation does occur, a random value is chosen from {0,1} for that bit. Mutation provides a mechanism for searching regions of the allele space not generated by selection and crossover, thus reducing the likelihood of local optima over time, but mutation is capable only of providing a random walk through the space of possible structures.

The basic concepts of GAs were developed by Holland (1975) and his students (De Jong 1980; Bethke 1981; Goldberg 1988; and others). Theoretical considerations concerning the allocation of trials to structures (Holland 1975; De Jong 1980) show that genetic techniques provide a near-optimal heuristic for information gathering in complex search spaces. A number of experimental studies (De Jong 1980; Bethke 1981; Grefenstette 1986) have shown that GAs exhibit impressive efficiency in practice. While classical gradient search techniques are more efficient for problems which satisfy tight constraints (e.g., continuity, low-dimensionality, unimodality, etc.), GAs consistently out-perform both gradient techniques and various forms of random search on more difficult (and more common) problems, such as optimizations involving discontinuous, noisy, high-dimensional, and multimodal objective functions (Holland 1984; Brady 1985).

The GAs do not require well-behaved, convex objective functions—indeed, they do not require closed objective functions at all—which provides an opportunity for an exhaustive study of the solution to repeated games. This is possible because to use the GA to search for better solutions it is sufficient that each individual solution can be scored for its “evolutionary

fitness:” in our case the aggregate score of a repeated game provides that measure, but in general any value that depends on the particular pattern of each individual chromosome will do.

References

Aumann R. (1976) Agreeing to disagree. *Annals of Statistics* **4**: 1,236–1,239.

Aumann R. (1985) Repeated games. In: Feiwel G.R. (ed.) *Issues in Contemporary Microeconomics and Welfare.* Macmillan, London.

Aumann R. (1989) Game theory. In: Eatwell J., Milgate M., Newman P., (eds.) *The New Palgrave: Game Theory.* Macmillan, London.

Axelrod R. (1984) *The Evolution of Coöperation.* Basic, New York.

Axelrod R. (1987) The evolution of strategies in the iterated Prisoner’s Dilemma. In: Davis L. (ed.) *Genetic Algorithms and Simulated Annealing.* Morgan Kaufmann, Los Altos.

Axelrod R. and Dion D. (1988) The further evolution of coöperation. *Science* **242**: 1,385–1,390.

Bertrand J. (1883) Review of *Théorie Mathématique de la Richesse Sociale* and of *Recherches sur les Principes Mathématiques de la Théorie des Richesses. Journal des Savants* **68**: 499–508.

Bethke A.D. (1981) Genetic algorithms as function optimizers. (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International* **41**(9): 3503B. (University Microfilms No. 81-06101)

Binmore K. and Dasgupta P. (1986) Game theory: a survey. In: Binmore K. and Dasgupta P. (eds.) *Economic Organizations as Games.* Basil Blackwell, Oxford.

Boyd R. and Lorberbaum J.P. (1987) No pure strategy is evolutionarily stable in the repeated Prisoner’s Dilemma game. *Nature* **327**: 58–59.

Brady R.M. (1985) Optimization strategies gleaned from biological evolution. *Nature* **317**: 804–806.

Cohen M.D. and Axelrod R. (1984) Coping with complexity: the adaptive value of changing utility. *American Economic Review* **74**: 30–42.

Cournot A. (1838) *Recherches sur les Principes Mathématiques de la Théorie des Richesses.* Hachette, Paris.

Davis L. (1991) A genetic algorithms tutorial. In: Davis L. (ed.) *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

De Jong, K.A. (1980) Adaptive system design: a genetic approach. *IEEE Transactions on Systems, Man, and Cybernetics* SMC–**10**: 566–524.

Diekmann A. and Mitter P. (eds.) (1986) *Paradoxical Effects of Social Behavior: Essays in Honor of Anatol Rapoport.* Physica-Verlag, Heidelberg.

Dupré J. (ed.) (1987) *The Latest on the Best: Essays on Evolution and Optimality.* M.I.T. Press, Cambridge.

Fader P.S., and Hauser J.R. (1988) Implicit coalitions in a generalized Prisoner’s Dilemma. *Journal of Conflict Resolution* **32**: 553–582.

Forrest S. and Mayer-Kress G. (1991) Genetic algorithms, nonlinear dynamical systems, and models of international security. In: Davis L. (ed.) *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Friedman D. (1991) Evolutionary games in economics. *Econometrica* **59**(3): 637–666.

Friedman J.W. (1971) A non-coöperative equilibrium of supergames. *Review of Economic Studies* **38**: 1–12.

Friedman J.W. (1983) *Oligopoly Theory.* Cambridge University Press, Cambridge.

Fudenberg D., and Maskin E. (1986) The Folk Theorem in repeated games with discounting or incomplete information. *Econometrica* **54**: 533–554.

Fujiki C. and Dickinson J. (1987) Using the genetic algorithm to generate Lisp source code to solve the Prisoner's Dilemma. In: Grefenstette J.J. (ed.) *Genetic Algorithms and their Applications,* Proceedings of the 2nd. International Conference on Genetic Algorithms. Lawrence Erlbaum, Hillsdale, N.J.

Goldberg D.E. (1988) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass.

Goldberg D.E. and 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.

Grefenstette J.J. (1986) Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, & Cybernetics* SMC–**16**: 122–128.

Grefenstette J.J. (1987) A User's Guide to GENESIS. Navy Center for Application Research in Artificial Intelligence, Naval Research Laboratories, mimeo., Washington D.C.

Harrington J.E., Jr. (1987) Finite rationalizability and coöperation in the finitely repeated Prisoner's Dilemma. *Economics Letters* **23**: 233–237.

Holland J.H. (1975) *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

Holland J.H. (1984) Genetic algorithms and adaptation. In: Selfridge O., Rissland E., and Arbib M.A. (eds.) *Adaptive Control of Ill-Defined Systems.* Plenum, New York.

Kreps D., Milgrom P., Roberts J., and Wilson R. (1982) Rational coöperation in the finitely repeated Prisoner's Dilemma. *Journal of Economic Theory* **27**: 245–252.

Luce R.D. and Raiffa H. (1957) *Games and Decisions: Introduction and Critical Survey.* Wiley, New York.

Marimon R., McGrattan E., and Sargent T.J. (1990) Money as a medium of exchange in an economy with artificially intelligent agents. *Journal of Economic Dynamics and Control* **14**: 329–373.

Marks R.E. (1989) Niche strategies: the Prisoner's Dilemma computer tournaments revisited. *AGSM Working Paper 89–009.*

Marks R.E. (1990) Repeated games and finite automata. *AGSM Working Paper 90–045.*

Maynard Smith J. (1982) *Evolution and the Theory of Games.* Cambridge University Press, Cambridge.

Midgley D.F. (1988) personal communication.

Miller J.H. (1989) The coevolution of automata in the repeated Prisoner's Dilemma. *Santa Fe Institute Working Paper 89–003.*

Nachbar J.H. (1988*a*) The evolution of coöperation revisited. Mimeo., RAND Corporation, Santa Monica.

Nachbar J.H. (1988*b*) An ecological approach to economic games. Mimeo., RAND Corporation, Santa Monica.

Nalebuff B. (1987) Economic puzzles: noisy prisoners, Manhattan locations, and more. *Journal of Economic Perspectives* **1**: 185–191.

Neyman A. (1985) Bounded complexity justifies coöperation in the finitely repeated Prisoner's Dilemma. *Economics Letters* **19**: 227–229.

Radner R. (1980) Collusive behavior in noncoöperative epsilon-equilibria of oligopolies with long but finite lives. *Journal of Economic Theory* **22**: 136–154.

Radner R. (1986) Can bounded rationality resolve the Prisoner's Dilemma? In: Hildenbrand W. and Mas-Colell A. (eds.) *Contributions to Mathematical Economics In Honor of Gérard Debreu.* North Holland, Amsterdam.

Rapoport A. (1988) Editorial comments on the article by Hirshleifer and Martinez Coll. *Journal of Conflict Resolution* **32**: 399–401.

Rawlins G.J.E. (1991) Introduction. In: Rawlins G.J.E. (ed.) *Foundations of Genetic Algorithms.* Morgan Kaufmann, San Mateo.

Samuelson L. (1988), Evolutionary foundations of solution concepts for finite, two-player, normal-form games. In: Vardi M. (ed.) *Proceedings of the Second Conference on the Theoretical Aspects of Reasoning About Knowledge.* Morgan Kaufmann, San Mateo.

Schaffer J.D. and Grefenstette J.J. (1988) A critical review of genetic algorithms. Mimeo.

Schelling T.C. (1984) What is game theory? In his: *Choice and Consequences.* Harvard University Press, Cambridge.

Selten R.C. (1975) Reëxamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory* **4**: 25–55.

Selten R.C. (1983) Evolutionary stability in extensive two-person games. *Mathematical Social Sciences* **5**: 269–363.

Shubik M., with Levitan R. (1980) *Market Structure and Behavior.* Harvard University Press, Cambridge.

Smith S.F. (1981) A learning system based on genetic adaptive algorithms. (Doctoral dissertation, University of Pittsburgh). *Dissertation Abstracts International* **41**(12): 4582B. (University Microfilms No. 81-12638)

Sonnenschein H. (1989) Oligopoly and game theory. In: Eatwell J., Milgate M., Newman P., (eds.) *The New Palgrave: Game Theory.* Macmillan, London.

Sorin S. (1986) On repeated games with complete information. *Mathematics of Operations Research* **11**: 147–160.

To T. (1988) More realism in the Prisoner's Dilemma. *Journal of Conflict Resolution* **32**: 402–408.

Ulph A. (1987) Recent advances in oligopoly theory from a game theory perspective. *Journal of Economic Surveys* **1**: 149–172.