ELSEVIER

Journal of Economic Dynamics & Control
24 (2000) 1–19

JOURNAL OF Economic Dynamics & Control

www.elsevier.nl/locate/econbase

An illustration of the essential difference between individual and social learning, and its consequences for computational analyses

Nicolaas J. Vriend*

Department of Economics, Queen Mary and Westfield College, University of London, Mile End Road, London, E1 4NS, UK

Received 22 July 1998; accepted 9 November 1998

Abstract

Drawing a conclusion from recent insights in evolutionary game theory, we show that a so-called spite effect implies that there is an essential difference between individual and social learning. We illustrate its consequences for the choice of computational tools in economics and social settings in general by analyzing two variants of a Genetic Algorithm. © 2000 Elsevier Science B.V. All rights reserved.

JEL classification: C72; D43; D83

Keywords: Cournot oligopoly; Individual learning; Social learning; Spite effect; Genetic Algorithm

1. Introduction

Learning and adaptive behavior are studied frequently in the social sciences nowadays. And for various reasons not spelled out here (but see, e.g., Vriend (1994), or Vriend (1996)), this is often done following a computational approach, using a variety of different algorithms. One dimension in which these can be distinguished is the level at which learning is modeled. The two basic possibilities are the individual and the population level. We will make these learning processes more precise below, but the basic idea is the following. With individual

* E-mail: ⟨n.vriend@qmw.ac.uk⟩, http://www.qmw.ac.uk/ ~ ugte173/.

PII: S0165-1889(98)00068-2

learning an agent learns *exclusively* on the basis of his own experience, whereas population or social learners base themselves on the experience of other players as well (see, e.g., Ellison and Fudenberg, 1993). The difference between these two approaches to modeling learning is often neglected, but we show that for a general class of games or social interactions this difference is essential. Hence, we will argue in this paper that the choice of the computational tools to be used should be influenced by an understanding of the underlying dynamics of the processes to be modeled.

As Blume and Easley (1993) pointed out, when there is learning in an interactive setting, there are actually two underlying processes. On the one hand, there will be a change in the perception of the underlying circumstances, i.e., learning, but on the other hand, there will also be a change in these circumstances themselves. And in general, the dynamics of learning and the dynamics of the underlying forces as such will interact with each other. In an economic setting, we will show that the effect of the economic forces as such might be that an identical learning algorithm will lead to sharply different results when applied as a model of individual learning and when applied as a model of social learning.

The phenomenon causing this is called the '*spite effect*'. The spite effect occurs when choosing an action that hurts oneself but others even more. The term '*spiteful behavior*' goes back, at least, to Hamilton (1970). There are two aspects of the spite effect. First, purely spiteful behavior, related to the preferences of the agents. Some players simply like to beat other players (see, e.g., Fouraker and Siegel (1963), or Levine (1998)). Second, spiteful behavior related to the limited perception of the agents (related in turn to their bounded rationality, learning, lack of information, etc.). It is this second aspect, stressed in the recent evolutionary game theory literature (see Rhode and Stegeman (1995), and Vega-Redondo (1997)), on which we will focus. It goes back to the casual side-remarks of Friedman (1953) about selection and optimizing behavior. In order to lend some plausibility to his 'as if' argument, Friedman gave the example of top billiard players who do not know the laws of physics, but are likely to play as if they were optimizing using those rules, because otherwise they would not have survived a competitive selection process. Schaffer (1988, 1989) understood the importance of the spite effect in this respect. Suppose players are boundedly rational, and do not know how to optimize. Instead they look around to see what other players achieve, with the probability of choosing a certain strategy observed in the population being a monotonically increasing function of the payoff realized by that strategy. As we will see in greater detail below, the spite effect influences such a selection process through the different effects it has on the payoffs of different strategies.

In order to illustrate the essence of the spite effect, consider the bimatrix game in Fig. 1 (see Palomino, 1995), where T and B are the two possible strategies, and a, b, c, and d are the payoffs to the row and column player, with $a > b > c > d$. Clearly, (T, T) is the only Nash equilibrium since no player can

	T	B
T	a, a	c, b
B	b, c	d, d

Fig. 1. Bimatrix game, with payoffs $a > b > c > d$.

improve by deviating from it, and this is the only combination for which this holds. Now, consider the strategy pair (B, T), leading to the payoffs (b, c). Remember that $a > b > c > d$. Hence, by deviating from the Nash equilibrium, the row player hurts herself, but she hurts the column player even more. We could also look at it from the other side. Suppose both players are currently playing strategy B, when the column player, for one reason or another, deviates to play strategy T instead, thereby improving his payoff from d to c. But the row player, simply sticking to her strategy B, would be 'free riding' from the same payoff of d to a payoff c that is even higher than b. The question, then, is what effects these outcomes will have on what the players learn.

This simple example immediately highlights the two issues that we will analyze in greater detail in this paper. On the one hand the interaction between the players' actual behavior (and the outcomes thus generated), and on the other hand their learning about actions and outcomes. As we will see, it is the way in which these processes interact with each other that causes the essential difference between individual learning and social learning.

The remainder of this paper is organized as follows. In Section 2 we present a computational example illustrating this difference, which we will analyze in Section 3 in relation to the spite effect. Section 4 draws our example into a broader perspective by discussing some specific features of the example, and Section 5 concludes.

2. A computational example

Consider a standard Cournot oligopoly game. There is a number n of firms producing the same homogeneous commodity, who compete all in the same market. The only decision variable for firm i is the quantity q_i to be produced. Once production has taken place, for all firms simultaneously, the firms bring their output to the market, where the market price P is determined through the confrontation of market demand and supply. Let us assume that the inverse demand function is $P(Q) = a + bQ^c$, where $Q = \Sigma q_i$. Making the appropriate assumptions on the parameters a, b, and c ensures that this is a downward-sloping curve, as sketched in Fig. 2. Hence, the more of the commodity is supplied to the market, the lower the resulting market price P will be. We

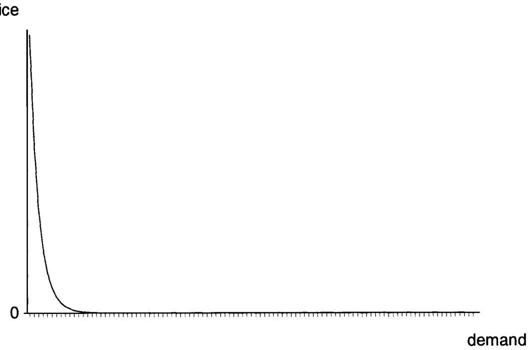

Fig. 2. Sketch demand function.

assume that the production costs are such that there are negative fixed costs K, whereas the marginal costs are k. Imagine that some firms happen to have found a well where water emerges at no cost, but each bottle costs k, and each firm gets a lump-sum subsidy from the local town council if it operates a well. Given the assumptions on costs, each firm might be willing to produce any quantity at a price greater or equal to k. But it prefers to produce that output that maximizes its profits. The parameters for the underlying economic model can be found in Appendix A.

Assume that each individual firm (there are 40 firms in our implementation) does not know what the optimal output level is, and that instead it needs to learn which output level would be good. Let us model this with a Genetic Algorithm (GA). Then, there are two basic ways to implement a GA.[1] The first is as a model of *social or population learning*. Each individual firm in the population is characterized by an output rule, which is, e.g., a binary string of fixed length, specifying simply the firm's production level.[2] In each trading day, every firm produces a quantity as determined by its output rule, the market price is determined, and the firms' profits are determined. After every 100 trading days, the population of output rules is modified by applying some reproduction, crossover, and mutation operators.[3] The underlying idea is that firms look around, and tend to imitate, and re-combine ideas of other firms that appeared to be successful. The more successful these rules were, the more likely they are to

[1] The only essential point for our story is here the difference between individual and social learning. For a good introduction of GAs as such see, e.g., Goldberg (1989), or Mitchell (1996).

[2] Hence, the output rules do not have the conditional '*if*… *then*…' form.

[3] In each of the 100 periods between this, a firm adheres to the same output rule. This is done to match the individual learning GA (see below), and in particular its speed, as closely as possible.

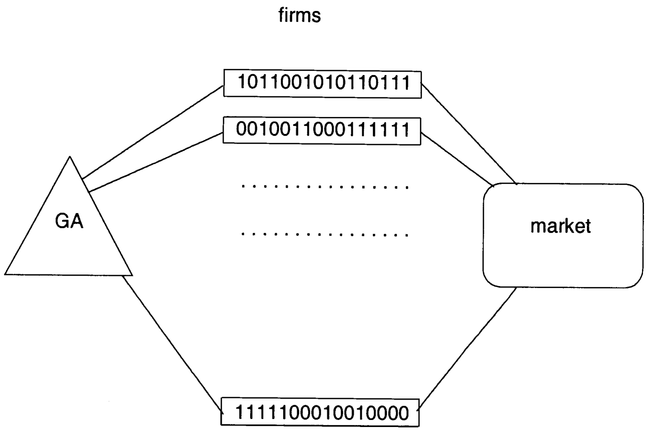

Fig. 3. Social learning GA.

be selected for this process of imitation and re-combination, where the measure of success is simply the profits generated by each rule. Fig. 3 shows both the Cournot market process, and the social learning process with the GA.

The second way to implement a GA is to use it as a model of *individual learning*. Instead of being characterized by a single output rule, each individual firm now has a set of rules in mind, where each rule is again modeled as a string, with to each rule a fitness measure of its strength or success attached i.e., the profits generated by that rule when it was activated. Each period only one of these rules is used to determine its output level actually supplied to the market; the rules that had been more successful recently being more likely to be chosen. This is known as a Classifier System (see, e.g., Holland (1986) or Holland (1992)). The GA, then, is used every 100 periods to modify the set of rules an individual firm has in mind in exactly the same way as it was applied to the set of rules present in the population of firms above. Hence, instead of looking how well other firms with different rules were doing, a firm now checks how well it had been doing in the past when it used these rules itself.[4] Fig. 4 shows the underlying economic market process, and the individual learning process. The parameter specification of the GA plus the pseudo-code can be found in Appendix B.

[4] Hence, an alternative way to obtain the match in the speed of learning of the individual and the social learning GA would have been to endow the individual learning GA with the capability to reasoning about the payoff consequences for every possible output level in its set, updating all strengths every period.

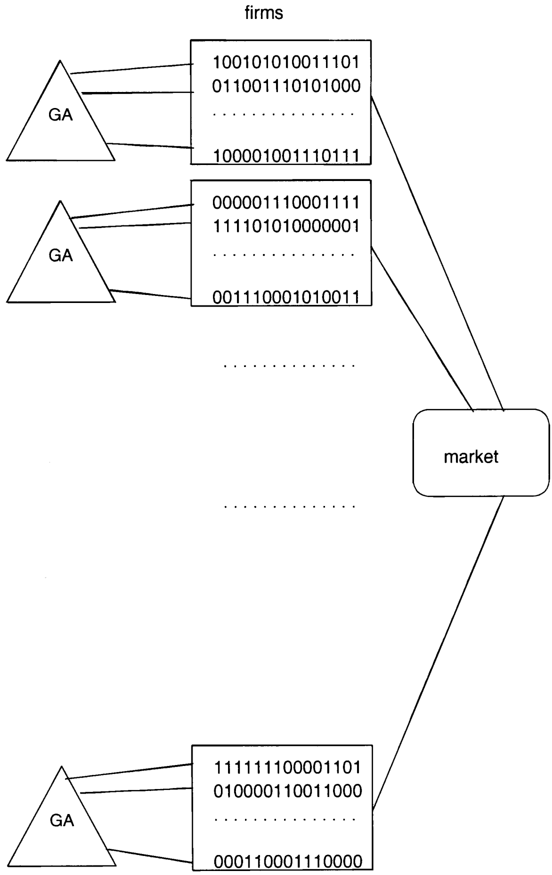

Fig. 4. Individual learning GA.

Fig. 5 presents the time series of the average output levels for each of the 25 runs of the two algorithms. As we see, they approach a different level. Whereas both series start around 1000, the social learning GA quickly 'converges' to a level of 2000, but the individual learning GA keeps moving around a level just below 1000.[5] Table 1 shows the average output level and the standard deviation

[5] The 5000 periods here presented combined with the GA rate of 100 imply that the GA has generated 50 times a new generation in each run. Each single observation in a given run is the average output level for that generation. We did all runs for at least 10,000 and some up to 250,000 periods, but this did not add new developments.

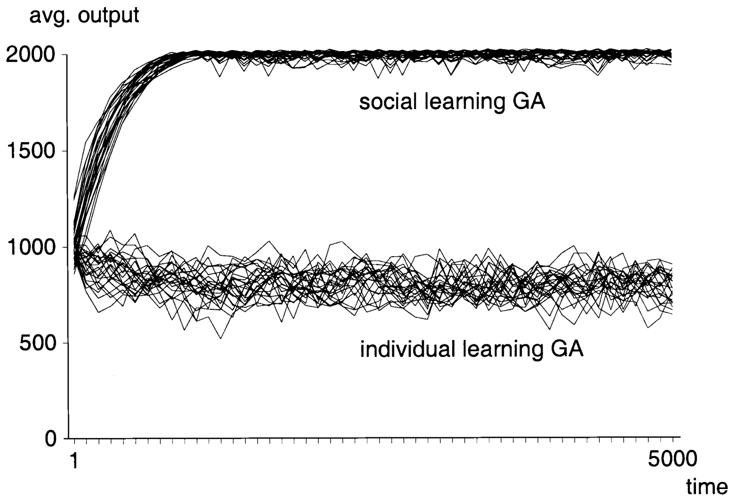

Fig. 5. Average output levels individual learning GA and social learning GA.

Table 1
Output levels individual learning GA and social learning GA, periods 5001–10,000

	Indiv. learning GA	Social learning GA
Average	805.1	1991.3
Standard deviation	80.5	24.7

for the periods 5001 to 10,000 in the two variants of the GA. We want to stress that these data are generated by exactly the same identical GA for exactly the same identical underlying economic model.

3. Analysis

We first compute two equilibria of the static Cournot oligopoly game specified above for the case in which the players have complete information. The GAs do not use this information, but the equilibria will serve as a theoretical benchmark that helps us understanding what is going on in the GAs. Besides the parameters of the underlying economic model, Appendix A also presents the formal derivation of the two equilibria.

If the firms behave as *price takers* in a competitive market, they simply produce up to the point where their marginal costs are equal to the market price

P. Given the specification of the oligopoly model above, this implies an aggregate output level of $Q^{W} = 80{,}242.1$, and in case of symmetry, an individual Walrasian output level of $Q^{W}/n = 2006.1$. If, instead, the firms realize that they influence the market price through their own output, still believing that their choice of q does not directly affect the output choices of the other firms, they produce up to the point where their marginal costs are equal to their marginal revenue. This leads to an aggregate Cournot–Nash equilibrium output of $Q^{N} = 39{,}928.1$, and with symmetry to an individual Cournot–Nash output of $Q^{N}/n = 998.2$.

As we see in Fig. 5, the GA with individual learning moves close to the Cournot–Nash output level, whereas the GA with social learning 'converges' to the competitive Walrasian output level. The explanation for this is the spite effect.

In order to give the intuition behind the spite effect in this Cournot game, let us consider a simplified version of a Cournot duopoly in which the inverse demand function is $P = a + bQ$, and in which both fixed and marginal costs are zero (see Schaffer, 1989). The Walrasian equilibrium is then $Q^{W} = -a/b$, as indicated in Fig. 6. Suppose firm i produces its equal share of the Walrasian output: $q_i = Q^{W}/2$. If firm j would do the same, aggregate output is Q^{W}, the market price P will be zero, and both make a zero profit. What happens when firm j produces more than $Q^{W}/2$? The price P will become negative, and both firms will make losses. But it is firm i that makes less losses, because it has a lower output level sold at the same market price P. What happens instead if firm j produces less than $Q^{W}/2$? The price P will be positive, and hence this will increase firm j's profits. But again it is firm i that makes a greater profit, because

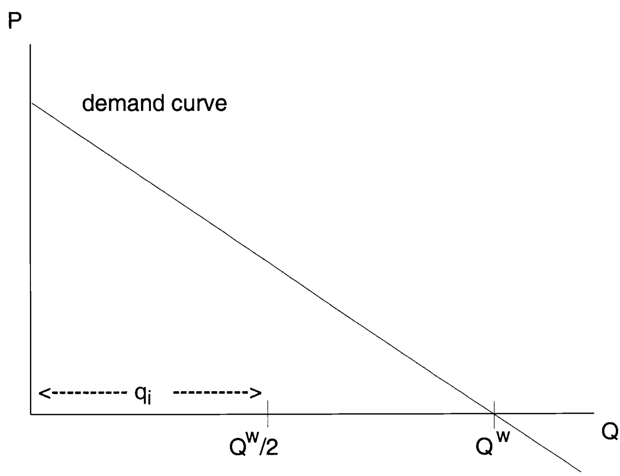

Fig. 6. Example simple Cournot duopoly.

it has a higher output level sold at the same market price P. In some sense, firm i is free riding on firm j's production restraint. Hence, in this Cournot duopoly the firm that produces its equal share of Q^W will always have the highest profits. Note that this implies in particular the following. If firm i produces its share of the symmetric Walrasian output, while firm j naively chooses the symmetric output level to maximize its absolute profits (i.e., its equal share of the Cournot–Nash output), it is firm i that realizes the highest profits. Moreover, even if firm j is aware of the fact that firm i is producing at the Walrasian output level, and maximizes its profits taking this into account, it is firm i that realizes the highest profits.

In case we consider oligopolies with more than two firms, matters become slightly more complicated, but the following holds. Whenever the aggregate output level is below Walras, i.e., on average an individual firm produces less than its share of the Walrasian output level, the price will be positive, and it is the firms with the higher output levels that generate the higher profits. Notice that in particular cases this might imply that a firm producing well above its share of Walras generates higher profits than a firm producing exactly its share of Walras. Exactly the reverse holds when aggregate output exceeds the Walrasian output level; the lower a firm's output level, the higher its profits will be.

How do these payoff consequences due to the spite effect explain the difference in the results generated by the two GAs? As we saw above, the spite effect is a feature of the underlying Cournot model, and is *independent* of the type of learning applied. The question, then, is how this spite effect is going to influence what the firms learn. It turns out, this depends on *how* the firms learn.

In the *social learning GA*, each firm is characterized by its own production rule (see Fig. 3). The higher a firm's profits, the more likely is its production rule to be selected for reproduction. Due to the spite effect, whenever aggregate output is *below* Walras, this happens to be those firms that produce at the *higher* output levels. And whenever aggregate output is *above* Walras, the firms producing at the *lowest* output are most likely to be selected for reproduction. As a result, the population of firms tends to converge to the Walrasian output.[6]

In the *individual learning GA*, however, the production rules that compete with each other in the learning process do not interact with each other in the same Cournot market, because in any given period, an individual firm actually applies only *one* of its production rules (see Fig. 4). Hence, the spite effect, while still present in the market, does not affect the learning process, since the payoff generated by that rule is not influenced by the production rules that are used in *other* periods. Clearly, there is a spite effect on the payoffs realized by the other

[6] This is not due to the specifics of this simple example, but it is true with great generality in Cournot games (see Vega-Redondo, 1997; Rhode and Stegeman, 1995). In particular, it also holds when all firms start at the Cournot–Nash equilibrium (as long as there is some noise in the system).

firms' production rules, but those do not compete with this individual firm's production rules in the individual learning process. We would like to stress that it is these learning processes that is the crucial feature here, and not the objectives of the agents. Both the individual and the social learners only try to improve their own *absolute* payoffs. The only difference is that their learning is based on a different set of observations.

Fig. 7 further illustrates the different consequences of the spite effect in the two variants of the GA by showing the average utility levels generated by the output levels reported in Fig. 5. In the social learning GA the spite effect drives down the utility levels, while the performance of the individual learning GA improves over time. In other words, the dynamics of learning and the dynamics of the economic forces as such interact in a different way with each other in the two variants of the GA, and this explains the very different results generated by the two GAs.

There is one additional issue to be analyzed. As we saw above in Fig. 5 and Table 1, 'convergence' with the individual learning GA is not as neat as with the social learning GA. Some numerical analysis shows that this is *not* a flaw of the individual learning GA, but related to the underlying economic model. The formal analysis of the Cournot model shows that there is a unique symmetric Cournot–Nash equilibrium. But in our numerical model we use a discrete version of the model, as only integer output levels are allowed. As a result, there turn out to be 1637 symmetric Cournot–Nash equilibria; for any average output level of the other firms from 1 to 1637, the best response for an individual firm is

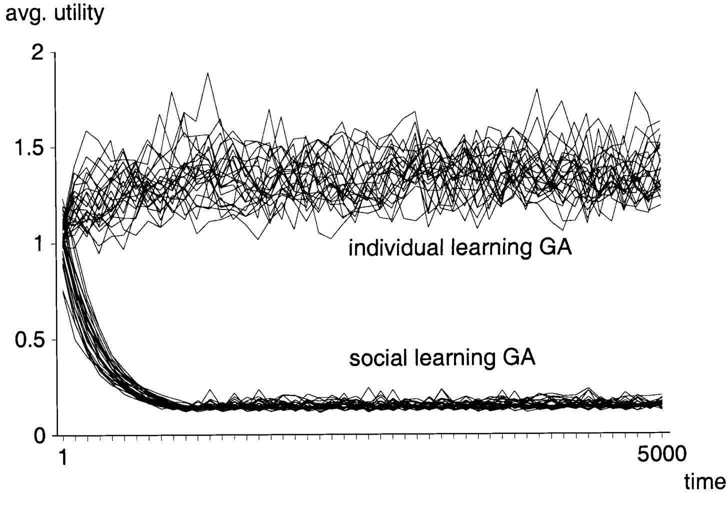

Fig. 7. Average utility levels individual learning GA and social learning GA.

to choose exactly the same level. Hence, the outcomes of the individual learning GA are determined by the underlying economic forces, but convergence can take place at any of these Cournot–Nash equilibrium levels. As a result, the output levels actually observed in the individual learning GA depend in part on chance factors such as the initial output levels, the length of the bit string, and genetic drift. Notice that although there are multiple Cournot–Nash equilibria, they are still distinct from the Walrasian equilibrium.

4. Discussion

Before we draw some general conclusions, let us discuss some specific issues in order to put our example into a broader perspective. First, the spite effect we presented occurs in finite populations where the agents 'play the field'. The finite population size allows an individual agent to exercise some power, and influence the outcomes of the other players. For example, in a Cournot model, when the population size n approaches infinity, the Cournot–Nash output level converges to the competitive Walrasian output. Finite populations are typically the case in computational analyses; certainly those applying GAs. To see why the 'playing the field' aspect is important, suppose there are many separate markets for different commodities, such that the actions in one market do not influence the outcomes in other markets, whereas firms can learn from the actions and outcomes in other markets. Since the spite effect does not cross market boundaries, if all firms in one market produce at the Cournot–Nash level, they will realize higher profits than the firms in another market producing at a Walrasian level. 'Playing the field' is typically the case in, e.g., economic models where the players are firms competing in the same market. There are also some results concerning the spite effect with respect to, e.g., 2-person games in infinite populations, but matters become more complicated (see Palomino, 1995).

Second, individual learning and social learning are two extreme forms of learning. One of the referees suggested that there is also an interesting intermediate '*type learning*' case, in which agents of several distinct types interact with each other, but they only learn from successful agents of their own type. Individual learning in this paper is in effect a special case of type learning, with each agent type being a singleton. And, obviously, there are also other dimensions to distinguish learning models from each other.

Third, although, as we have seen, the spite effect may influence the outcomes of a coevolutionary process, one should not confuse the spite effect with the phenomenon of coevolution as such. In fact, as the bimatrix game in the introduction showed, the spite effect can occur in a static, one-period game, and is intrinsically unrelated to evolutionary considerations.

Fourth, one of the referees suggested that some more basic issues concerning evolutionary modeling are touched on here. Individual learning might be the counterpart to individual rationality, while social learning could be the equivalent of competitive behavior or market forces. We do not attempt to provide any explanation as to *why* individual players adhere to individual or to social learning.[7] Interestingly, in the Cournot model social learning leads to the socially desirable competitive outcome. It is an open question whether this reflects a more general phenomenon, and whether the spite effect plays a fundamental role in this respect.

Fifth, the simple Cournot model we considered is not a typical search problem for a GA; not even if the demand and cost functions were unknown. The appeal of the Cournot model is not only that it is convenient for the presentation because it is a classic discussed in every microeconomics textbook, but the fact that we can derive formally two equilibria providing us also with two useful benchmarks for the analysis of the outcomes generated by the learning algorithms. Hence, the Cournot model is just a vehicle to explain the point about the essential difference between individual and social learning, and for any model, no matter how complicated, in which a spite effect occurs this essential difference will be relevant.

Sixth, one could consider more complicated strategies than the simple output decisions modeled here. For example, the Cournot game would allow for collusive behavior. However, as is well-known from the experimental oligopoly literature, dynamic strategies based on punishment and the building up of a reputation are difficult to play with more than two players. Moreover, considering more sophisticated dynamic strategies would merely obscure our point, and there exists already a large literature, for example, on GAs in Iterated Prisoners' Dilemma (see, e.g., Axelrod (1987), Stanley et al. (1994), or Miller (1996)).

Seventh, we are sure that the GAs we have used are too simple, and that much better variants are possible. However, bells and whistles are not essential for our point. The only essential aspect is the level at which the learning process is modeled, and the effect this has on the convergence level.

Eighth, we would even argue that there is nothing intrinsically linked to GAs in our story. For *any* learning algorithm that is based on a selection mechanism which is monotonic in the payoffs it will make an essential difference whether one applies it at the population or at the individual level.[8] The main reason to

[7] Nevertheless, our analysis immediately suggests one possibility. If players practise social learning in a Cournot model, their profits will go down. Hence, even without much knowledge of their environment, one would expect the players to resist this form of learning. Experimental evidence (Bosch-Domènech and Vriend, 1998), suggests this is indeed the case.

[8] For example, reinforcement learning as in Roth and Erev (1995) takes place at the individual level, whereas replicator dynamics (see, e.g., Weibull, 1995) are a form of social learning.

focus on a GA in this paper is that this algorithm allows us to use exactly the same leaning mechanism at both the individual and the social level. And indeed both variants of GAs have been used in the literature (see, e.g., Axelrod (1987), Marimon et al. (1990), Arifovic (1993), Stanley et al. (1994), Vriend (1995), or Miller (1996)).

5. Conclusion

The general conclusions are twofold. The first one concerns the *interpretation* of outcomes generated by a numerical model in which learning takes place. We showed that the presence of the spite effect implies that there is an essential difference between an individual learning and a social learning algorithm. In other words, when interpreting outcomes of computational models, one needs to check which variant is used, and one needs to check whether a spite effect driving the results might be present. That is, one needs to understand both the dynamics of learning, the dynamics of the underlying forces, and the way these two interact with each other.

The second conclusion concerns the choice of learning variant. In the social sciences people tend to choose the social learning variant. As a matter of routine, the justifications given are a combination of the following. The fact that it is simpler to program seems to mean that it serves the scientific principle of '*parsimony*'. By referring to a list of other papers making the same choice, it is argued that the social learning version is '*standard*' in the literature. And this is supported by "*authority*' in the form of reference to some of the seminal works in the field. The lesson to draw here is that the computational modeling choice between individual and social learning algorithms should be made more carefully, since there may be significant implications for the outcomes generated.

Ultimately it seems an empirical issue whether and when people tend to learn individually or socially (see, e.g., Huck et al. (1999), Offerman et al. (1997), or Bosch-Domènech and Vriend (1998)). But until this has been sorted out, at least one should be aware of what processes are actually being modeled when dealing with learning in an interactive situation.

Acknowledgements

I wish to thank Melanie Mitchell for looking at an earlier sketch of these ideas, and to Tony Curzon Price, Herbert Dawid, Bruce Edmonds, Steffen Huck, Blake LeBaron, three anonymous referees, and conference participants in

Ancona and Cambridge for helpful comments and discussions. The usual disclaimer applies.

Appendix A. The Cournot model: Parameters and analysis

Table 2 presents the parameter specification of the underlying Cournot model.

We derive two equilibria for a standard Cournot oligopoly where the inverse demand function is $P(Q) = a + bQ^c$, and the total costs for an individual firm are given by $TC(q) = K + kq$. Hence, the profits for an individual firm are $\Pi(q) = [a + bQ^c]q - [K + kq]$, where $Q = \sum_{i=1}^{n} q_i$. We first determine the optimal output for a firm that believes it cannot influence the market price P. The first-order condition is:

$$\frac{\mathrm{d}\Pi(q)}{\mathrm{d}q} = [a + bQ^c] - k = 0.$$

If a firm behaves as a *price-taker* in a competitive market, it simply produces up to the point where its marginal costs are equal to the market price P. Hence, the aggregate Walrasian equilibrium output is

$$Q^W = \left(\frac{k-a}{b}\right)^{1/c},$$

which in case of a symmetric equilibrium implies that the individual Walrasian output is

$$q_i^W = \frac{Q^W}{n}.$$

Table 2
Parameters Cournot oligopoly model

Inverse demand function	$P(Q)$	$a + b \cdot Q^c$
Demand parameter	a	-1×10^{-97}
Demand parameter	b	$\times 1.5 \times 10^{95}$
Demand parameter	c	-39.99999997
Fixed production costs	K	-4.097×10^{-94}
Marginal production costs	k	0
Number of firms	n	40

Next, we determine the optimal output for a firm that knows its own output will influence the market price P. The first-order condition is

$$\frac{\mathrm{d}\Pi(q)}{\mathrm{d}q} = P + \frac{\mathrm{d}P}{\mathrm{d}q}q - k = [a + bQ^c] + \frac{\mathrm{d}[a + bQ^c]}{\mathrm{d}q}q - k = 0.$$

If a firm realizes that it influences the market price through its own output, still taking the output of the other firms as given, it produces up to the point where its marginal costs are equal to its marginal revenue. Hence, the aggregate Cournot–Nash equilibrium output is

$$Q^{\mathrm{N}} = \left[\frac{k - a}{b\left(\frac{c}{n} + 1\right)}\right]^{1/c},$$

which in case of a symmetric equilibrium implies that the individual Cournot-Nash output is $q_i^{\mathrm{N}} = Q^{\mathrm{N}}/n$. When $a < 0, b > 0, c < 0$, and $c - 1 > -2n$, the second-order condition will also be satisfied.

Appendix B. The genetic algorithm

In Table 3 we give the parameter specification of the GA, while Table 4 presents the pseudo-code of the Genetic Algorithm. A further explanation follows below.

Table 3
Parameters Genetic Algorithm

Minimum individual output level	1
Maximum individual output level	2048
Encoding of bit string	Standard binary
Length of bit string	11
Number rules individual GA	40
Number rules social GA	40·1
GA-rate	100
Number new rules	10
Selection	Tournament
Prob. selection	Fitness/Σ fitnesses
Crossover	Point
Prob. crossover	0.95
Prob. mutation	0.001

Table 4
Pseudo-code Genetic algorithm

```
program MAIN;                                              {initialization firms}
for each firm do for each rule do
begin
    make random bit string of length 11 with standard binary encoding;
    fitness:= 1.00;
end;                                                       {start main loop}
for each period do
begin
    for each firm do                                       {Classifier Systems's actions}
    begin
        active_rule:= CHOOSE_ACTION;                       {see function below}
        output level:= action of active_rule;
    end;
    determine market price;
    for each firm do                                       {Classifier Systems's outcomes}
    begin
        profit:= (market price)·(output level)-costs;
        utility:= monotonic transformation of profit;
        with active_rule do fitness:= utility;
    end;
    if period is multiple of 100 then                      {application Genetic Algorithm}
    begin
        if individual learning GA then for each firm do
            GENERATE_NEW_RULES                             {see procedure below}
        else if social learning GA then
        begin
            create set of 40 rules taking the 1 rule from each firm;
            GENERATE_NEW_RULES;                            {see procedure below}
            re-assign 1 rule to each of the 40 firms
        end;
    end;
end;
```

```
function CHOOSE_ACTION;
begin
    for each rule do
    begin
        linearly rescale the firm's actual fitnesses to [0,1];
        bid:= rescaled_fitness + ε;                        {with ε≅N(0, 0.075)}
        with probability:= 0.025 the bid is ignored;
    end;
    determine highest_bid;
end;
choose_action:= highest_bid;
```

Table 4 (Continued)

```
44  procedure GENERATE_NEW_RULES;
45  linearly rescale the actual fitnesses to [0,1];
46  repeat;
47      choose two mating parent rules from 30 fittest rules by roulette wheel
        selection;
48          (each rule with
            probability := rescaled_fitness/sum_rescaled_fitnesses)
49      with probability := 0.95 do
50      begin
51          place the two binary strings side by side and choose random crossing
            point;
52          swap bits before crossing point;
53          choose one of the two offspring at random as new_rule;
54      end;
55      with new_rule do
56      begin
57          fitness := average fitnesses of the two mating parent strings;
58          for each bit do with prob. := 0.001 do mutate bit from 1 to 0 or other
            way round;
59      end;
60      if new_rule is not duplicate of existing rule
61          then replace one of weakest 10 existing rule with new_rule else throw
            away;
62  until 10 new rules created;
```

1–32 The main program.
2–6 Initialization of the firms.
2 In the individual learning GA there are 40 rules per firm, in the social learning variant there is only 1 rule per firm.
4 The initial rules are randomly drawn from a uniform distribution.
5 The initial fitness of all rules is 1.00.
7–32 The main loop of the program.
9–13 The firms actions are decided by a Classifier System.
11 The active rule is decided in the CHOOSE_ACTION procedure.
12 The firm's supply to the market is determined by the active rule.
14 The market price is determined through the inverse demand curve.
15–20 The market outcomes are reported to the Classifier System.
19 The fitness of the active rule is made equal to the utility just generated by that rule.
21–31 Every 100 periods the GA generates 10 new rules.
23–24 With the individual learning GA, the procedure GENERATE_NEW_RULES is applied to the set of rules for each firm separately.
25–30 With the social learning GA, the procedure GENERATE_NEW_RULES is applied to the set of rules for all firms combined, with each firm contributing just one rule.
33–43 The active rule is chosen by a stochastic auction in the Classifier System. Clearly, in the social learning variant this is trivial, as there is only one rule to choose from.
39 Through a 'trembling hand' some experimentation is added.
43 The rule making the highest bid will be the active one in this period.
44–62 The Genetic Algorithm as such.
47–48 Only the 30 fittest rules can be selected for reproduction.
49–54 The crossover operator.
58 The mutation operator.
60 To prevent complete convergence of the rules, no duplicate rules are allowed.
61 The 10 weakest rules are replaced by the newly created rules.

References

Arifovic, J., 1993. Genetic algorithm learning and the cobweb model. Journal of Economic Dynamics and Control 18 (1), 3.

Axelrod, R., 1987. The evolution of strategies in the iterated prisoner's dilemma. In: Davis, L. (Ed.), Genetic Algorithms and Simulated Annealing. Pitman, London.

Blume, L.E., Easley, D., 1993. What has the learning literature taught us?. In: Kirman, A.P., Salmon, M. (Eds.), Learning and Rationality in Economics. Blackwell, Oxford.

Bosch-Domènech, A., Vriend, N.J., 1998. Do boundedly rational people imitate? Working Paper No. 379. Queen Mary and Westfield College, University of London.

Ellison, G., Fudenberg, D., 1993. Rules of thumb for social learning. Journal of Political Economy 101, 612–643.

Fouraker, L.E., Siegel, S., 1963. Bargaining Behavior. McGraw-Hill, New York.

Friedman, M., 1953. The methodology of positive economics. In: Friedman, M. (Ed.), Essays in Positive Economics. University of Chicago Press, Chicago, pp. 3–43.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading MA.

Hamilton, W., 1970. Selfish and spiteful behavior in an evolutionary model. Nature 228, 1218–1225.

Holland, J.H., 1986. Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (Eds.), Machine Learning: An Artificial Intelligence Approach, vol. 2, Morgan Kaufmann, Los Altos, CA, pp. 593–623.

Holland, J.H., 1992. Adaptations in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. 2nd ed., MIT Press, Cambridge, MA.

Huck, S., Normann, H.T., Oechssler, J., 1999. Learning in Cournot oligopoly – an experiment. Economic Journal.

Levine, D.K., 1998. Modeling altruism and spitefulness in experiments. Review of Economic Dynamics.

Marimon, R., McGrattan, E., 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.

Miller, J.H., 1996. The coevolution of automata in the repeated prisoner's dilemma. Journal of Economic Behavior and Organization 29, 87–112.

Mitchell, M., 1996. An Introduction to Genetic Algorithms. MIT Press/Bradford, Cambridge, MA.

Offerman, T., Potters, J., Sonnemans, J., 1997. Imitation and belief learning in an oligopoly experiment (mimeo) University of Amsterdam, Department of Economics.

Palomino, F., 1995. On the survival of strictly dominated strategies in large populations. Institute for Economic Analysis WP 308.95. Universitat Autonoma de Barcelona.

Rhode, P., Stegeman, M., 1995. Evolution through Imitation (with applications to duopoly). Working Paper E95-43. Virginia Polytechnic Institute and State University, Department of Economics.

Roth, A.E., Erev, I., 1995. Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior 8, 164–212.

Schaffer, M.E., 1988. Evolutionary stable strategies for a finite population and a variable contest size. Journal of Theoretical Biology 132, 469–478.

Schaffer, M.E., 1989. Are profit maximizers the best survivors?. Journal of Economic Behavior and Organization 12, 29–45.

Stanley, E.A., Ashlock, D., Tesfatsion, L., 1994. Iterated prisoner's dilemma with choice and refusal of partners. in: Langton, C.G. (Ed.), Artificial Life III, vol. XVII, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Redwood, CA, pp. 131–175.

Vega-Redondo, F., 1997. The evolution of Walrasian behavior. Econometrica 65, 375–384.

Vriend, N.J., 1994. Artificial intelligence and economic theory. In: Hillebrand, E., Stender, J. (Eds.), Many-Agent Simulation and Artificial Life. IOS, Amsterdam, pp. 31–47.
Vriend, N.J., 1995. Self-organization of markets: an example of a computational approach. Computational Economics 8, 205–231.
Vriend, N.J., 1996. Rational behavior and economic theory. Journal of Economic Behavior and Organization 29 (2), 263–285.
Weibull, J., 1995. Evolutionary Game Theory. MIT Press, Cambridge, MA.