12 Social Learning in a Model of Adverse Selection*

JASMINA ARIFOVIC AND ALEXANDER KARAIVANOV

It is well known from the mechanism design literature that the optimal contracts that arise in environments with asymmetric information can take complicated forms due to the need to satisfy various, typically nonlinear, constraints – such as participation, incentive compatibility, and/or self-selection. Further, these optimal contracts crucially depend on the participating agents' preferences, the properties of the endowment process or production technology, and various elements of the institutional environment – such as the degree of contractibility, contract enforcement, and agents' ability to commit. Usually, this literature solves for the best possible contract assuming that certain actions, states, or types are unobservable to the contract designer – typically, the principal in a principal-agent setting. At the same time, however, it assumes that this designer has perfect knowledge of objects that are likely much harder to know or observe, such as the agent's preferences or decision-making process in general.

In this paper, we explicitly model the principal's learning process about what contracts to offer based only on observable information such as the principal's profits. Our main objective is to investigate whether and under what conditions this learning process converges to the optimal mechanism design contract. If the learning process does converge, we are also interested in how quickly convergence occurs. On the other hand, if the learning process does not converge to the theoretically optimal contract, we are interested in what the (suboptimal) contracts might look like.

In our previous related work (Arifovic and Karaivanov 2009), we study and compare the relative performance of two major learning paradigms from the literature, social and individual learning – that is, 'learning from others' versus 'learning by doing' – in the context of a principal-agent model of output sharing under moral hazard. The principal tries to learn the optimal agency contract without knowing the agent's preferences and the production technology. We find that learning is hard in this type of environment due both to its stochastic nature and to the discontinuity of the payoff space in the neighbourhood of the optimal contract.[1] In this difficult learning environment, we find that 'learning from others' (social learning) is much more successful in reaching the optimal contract.

It is worth noting that Curtis Eaton and Jasmina Arifovic have studied social learning in a context of a sender/receiver game of common interest with cheap talk in which players use messages to communicate their type (Arifovic and Eaton 1995, 1998). The 'order' of an equilibrium in the game is given by the number of player types that is successfully communicated. In addition, there are multiple equilibria of each order, and equilibria of different orders can be Pareto-ranked. The evolutionary social learning process results in interesting dynamics in which a population of genetic algorithm players climbs an equilibrium payoff ladder, moving from equilibrium of one order to equilibrium of the next-highest order. Eventually, this process results in convergence to the Pareto-optimal equilibrium of the highest order.

In this paper, we further examine the social learning process in an environment of adverse selection and monopolistic screening (see Maskin and Riley 1984). Specifically, we study the problem of a principal who needs to learn how to design an optimal wage contract, based only on observable information, when facing two types of agents who differ in effort costs. Agents' effort and output are observable but their type is unknown to and unobserved by the principal, which results in an adverse selection problem. We model the principal's learning process using the 'social evolutionary learning' (SEL) algorithm – see Arifovic (2000) for a discussion of the social learning paradigm – in which players (a set of principals) update their strategies based on imitating strategies of those players who have performed better in the past and occasionally experiment with new strategies.

The optimal contract (wage-effort pair) in our theoretical setting is (for any cost parameters) always a separating contract. Thus, a fully rational profit-maximizing principal who is aware of the preferences of both agent types – but does not know who is who – optimally will offer two types of contracts that the agents optimally will self-select. Naturally, we find that such a separating outcome – that is, when the two types accept different wage-effort pairs – is one possible result of the learning process. However, we also find that it is possible that the principal is unable to learn the optimal separating contract and instead ends up offering contracts such that one of the types (the high cost) is excluded – that is, that both contract offers violate her participation constraint. It can also happen that the learning process converges to a 'pooling' outcome, whereby both agent types take one of the two contracts offered.

We find that, although learning the optimal separating contract in the screening environment is very hard, even when using social learning, the learning process and outcomes are rich and interesting. Our results show that the SEL algorithm we implement always converges to one of three outcomes: separating, pooling, or excluding. That is, our principals might need to learn three different types of contracts. Our main finding is that, conditional on the type of contract reached in the final simulation period, the principals achieve payoffs that are quite close to the maximum possible payoff within that contract class. That is, if the principals were somehow able to restrict agents' choices so to stay within a given class, their learning process would lead very closely to the best possible contract in the class. Such a restriction is, of course, not optimal for the agents who 'shop' for the best contract offered, which might result in any of the three outcomes depending on the particular offers. Because of this, if we look at the average deviation to the maximum possible payoff[2] achieved across all runs (and, hence, across the three outcome types), it is quite high, indicating poor performance of the learning algorithm, although, at the same time, convergence within each outcome type is much better.

The main impediment to the learning process is not one of local stability – indeed, if we introduce a principal who offers the optimal separating contract in the initial population, all principals converge to the optimal contract in 99 per cent of the runs. Rather, the main impediment is one of global stability – that is, getting into the 'right' region of the contract space to ensure that adaptation is to the separating contract, not to one of the two alternative outcomes. The pooling and excluding cases do not maximize profits in the global sense (that is, among all feasible contracts) but can be locally optimal and stable within certain regions of the contract space.

More specifically, over all simulations with our baseline SEL specification,[3] our principals learn or converge with the highest frequency to an excluding contract (63 per cent), followed by a separating contract (25 per cent), and, finally, a pooling contract (12 per cent). The outcome types to which the learning algorithm converges critically depend on the agents' effort cost parameters and on the learning algorithm parameters, including the random seed. In each case, the convergence is not necessarily to the optimal (profit-maximizing) contract within that class. We find that the profit-maximizing excluding and pooling contracts are much easier to learn (where the profits achieved are close to the theoretical maximums within the corresponding class of contracts) than the optimal separating contract (where the profits achieved are, on average, a smaller fraction of the theoretical optimum). We investigate the factors that affect the adaptation process and the types of contracts to which social learning leads, and we discuss the intuition behind the observed outcomes. We also perform a wide range of additional simulations to analyze the robustness of our findings.

The rest of the paper is organized as follows. In the next section, we describe the theoretical model, solve for the optimal separating contract, and derive the profit-maximizing wage-effort pairs within the classes of excluding and pooling contracts. We then describe our implementation of the social learning algorithm, and report the results from the baseline simulations and a range of robustness checks. Finally, we discuss the theoretical and computational issues relevant for the learning process performance in our environment, and offer some concluding comments.

The Model

Consider a model of monopolistic screening with hidden information (second-degree price discrimination). Suppose that there are types of agents who differ in their costs of supplying a given level of labour effort, x, that is used to produce output, y, using the constant returns to scale technology, $y = x$. Specifically, for simplicity, let there be two agents[4] and let $c_i(x)$ be the cost of agent $i = L, H$ supplying effort level x. Each agent knows her effort costs.

Assumption A1: The cost function c is increasing and strictly convex with $c_H(0) = c_L(0) = 0$ and satisfies the single-crossing property – that is, $c'_H(x) > c'_L(x)$ for all $x > 0$.

A risk-neutral principal wants to design an optimal wage contract that maximizes his profits by employing one or both agents. A wage contract specifies an amount of output (or, equivalently, effort/work to be done by the agent), x, and a wage, w. Output is perfectly observable and contractible but the worker's type is not – that is, the principal does not know the preferences of the agent he employs but is aware that there are only two types of agents and knows perfectly their preferences and fractions in the population; he just cannot tell who is who. This asymmetric information creates an adverse selection problem in the spirit of Akerlof (1970) or Stiglitz and Weiss (1981): the contracts the principal offers affect which agent(s) apply for the job and the total profits the principal receives as a result. The principal has full commitment – that is, he cannot renege on or modify *ex post* a contract once an agent has accepted it.

Let (w, x) be a contract – a wage/effort pair the principal offers. As is standard in such settings, the principal optimally will offer two such contracts to 'screen' the two types. Assume the agents' utility is equal to the wage, w_i, minus the cost of effort, $c_i(x_i)$, $i = L, H$. The outside option of each agent is normalized to 0. There is no capacity constraint – that is each offered contract might be accepted by neither agent, a single agent, or both agents. The employer's problem is:

$$\max_{w_L, x_L, w_H, x_H} x_L - w_L + x_H - w_H, \tag{1}$$

subject to

$$w_L - c_L(x_L) \geq 0, \tag{pcL}$$
$$w_H - c_H(x_H) \geq 0, \tag{pcH}$$
$$w_L - c_L(x_L) \geq w_H - c_L(x_H), \tag{ssL}$$
$$w_H - c_H(x_H) \geq w_L - c_H(x_L). \tag{ssH}$$

The first two constraints, (pcL) and (pcH), are the participation constraints stating that the agent must obtain at least her reservation utility from a contract she takes. The latter two constraints, (ssL) and (ssH), are the self-selection constraints ensuring that agent type $i = L, H$ is screened correctly and takes her intended contract (x_i, w_i).

The Optimal Contract

Applying the revelation principle and standard arguments, Assumption A1 implies that, in the optimal separating contract,

- the participation constraint for the low-marginal-cost type (agent L), (pcL) is not binding;
- the self-selection constraint for the high-marginal-cost type (agent H), (ssH) is not binding; and
- the participation constraint for the high-cost type (pcH) and the self-selection constraint for the low-cost type (ssL) are binding.

Therefore, the optimal contract extracts all surplus from the high-cost type and gives the low-cost type just enough surplus to discourage her from pretending that she is a high-cost type. Plugging in the binding constraints (pcH) and (ssL), the first-order conditions for the principal's problem are:

$$1 - c_L'(x_L) = 0, \tag{2}$$

$$1 + c'_L(x_H) - 2c'_H(x_H) = 0, \quad (3)$$

which, together with the binding constraints (pcH) and (ssL), can be solved for the optimal separating contract (x^*_H, w^*_H, x^*_L, w^*_L). Denote by π^* the principal's profits from offering this contract. By assumption A1, equations (2) and (3) imply that $x^*_L > x^*_H$ and $w^*_L > w^*_H$ – the low-cost agent is assigned higher output and receives a higher wage. This implies that the principal would never optimally offer a pooling contract – that is, a contract with $x_H = x_L$ and $w_L = w_H$, despite the fact that such a contract satisfies the self-selection constraints.

Note that, in the optimal contract, the low-cost agent obtains utility strictly higher than her reservation value (an information rent). That is, the monopolist cannot extract all the surplus from the contractual relationship and, hence, his profits from the low-cost type are lower than those the employer could achieve if the high-cost agent were not present. If this reduction in profits needed to satisfy the self-selection constraint (ssL) is large enough, it might instead pay off for the employer to exclude the high-cost agent[5] by offering a contract that violates (pcH) and maximizes the principal's profits from agent L. Clearly, it is not optimal to exclude agent L instead, since she works more than agent H at any wage level. The optimal contract excluding agent H solves:

$$\max_{w_L, x_L} x_L - w_L, \quad (4)$$

subject to

$$w_L - c_L(x_L) \geq 0,$$
$$w_L - c_H(x_L) < 0.$$

It is easy to see that the first constraint must bind at optimum and, then, by assumption A1, it is clear that the second constraint is satisfied. Thus, the profit-maximizing excluding contract, (x^e_L, w^*_L) solves

$$1 - c'_L(x_L) = 0 \text{ and } w_L = c_L(x_L).$$

Note that x^e_L takes the same value as in the optimal separating contract, x^*_L, but the corresponding wage is lower – that is, $x^e_L < w^*_L$. Call the employer's profits associated with this contract π^e.

A Computable Example

Assume a quadratic effort cost function, $c_i(x_i) = a_i \frac{x_i^2}{2}$ for $i = L,H$, with $a_H > a_L > 0$. Clearly, assumption A1 is satisfied. The first-order conditions, equation (2) – (3) become:

$$1 - a_L x_L = 0,$$
$$1 + a_L x_H - 2a_H x_H = 0.$$

Solving those, together with the binding constraints, gives the optimal separating contract:

$$x_L^* = \frac{1}{a_L},\; x_H^* = \frac{1}{2a_H - a_L},$$

and

$$w_H^* = c_H(x_H^*) = \frac{a_H}{2(2a_H - a_L)^2},$$
$$w_L^* = c_L(x_L^*) + w_H^* - c_L(x_H^*) = \frac{1}{2a_L} + \frac{a_H - a_L}{2(2a_H - a_L)^2}.$$

Further, the contract that maximizes profits from agent L if agent H is excluded solves problem (4) above,[6] which, in our example, yields:

$$x_L^e = \frac{1}{a_L} \text{ and } w_L^e = \frac{1}{2a_L}$$

with profits $\pi^e = \frac{1}{2a_L}$. It is easy to verify that, for any $a_H > a_L > 0$, the firm makes larger profits by keeping both agents contracted – that is, $\pi^* > \pi^e$ – so, with our quadratic cost function, it is never optimal to exclude the high-cost agent. In the simulation runs, we always assume that the principal offers two contracts (wage-effort pairs), but in the exclusion scenario, both contracts fail to satisfy agent H's participation constraint.

Our main objective is to examine whether social learning via a genetic algorithm can be used to learn the optimal contract. That is, we study how hard or easy it is for boundedly rational players to learn what the optimal contract looks like. As a first pass, we assume one-sided learning – that workers are perfectly rational, know their preferences, and optimally pick the better of the two contracts offered. In contrast, employers know only that they must offer effectively four numbers: two wages and two output levels (w_L, w_H, x_L, x_H). They observe the resulting profits from their choices each period and use those to update the offered contracts according to the social learning algorithm (see the next section for details). This implies that, along the learning path, suboptimal contracts inevitably would be offered, and sometimes both agents could pick the same contract (wage-output pair) or a contract might not be picked by any agent. Assume that, if an offered contract, (w_L, x_L) or (w_H, x_H), is not taken by any agent, the principal obtains zero profits from it. Similarly, if both offered contracts violate some agent's participation constraint, this agent obtains a payoff of zero.

We classify the contracts to which the social learning process can converge as three types: separating contracts (when both agents accept an offered wage-output pair and these pairs are not the same); excluding contracts (when agent L accepts a wage-output pair from the two offered but agent H accepts neither); and pooling contracts (when both agents accept the same wage-output pair of the two offered and the other pair is accepted by no one).[7] Indeed, in our simulations, we observe the occurrence of all three outcomes as a result of using the social learning algorithm. This implies that the learning algorithm performance is not perfect since we know from theory that the only optimal contract is the separating one. We investigate how the fraction of runs (for various cost parameters and random seeds) converging to each of the three outcomes depends on the learning algorithm parameters and also how close we get to the optimal contract or to the optimal profit level. We compare both the absolute/overall performance of the learning algorithm (relative to the optimal separating contract) and its relative performance – that is, conditional on converging to an excluding or a pooling contract, how close we get to the best contract within that class.

For the latter exercise, we need to solve for the profit-maximizing pooling contract – that is, when the principal offers the same x,w pair:

$$\max_{w,x} x - w,$$

subject to

$$w - c_H(x) \geq 0.$$

Since for the same (x,w), the (pcH) always binds before (pcL), we do not need to include (pcL). The profit-maximizing pooling contract is then

$$x^p = \frac{1}{a_H} \text{ and } w^p = \frac{1}{2a_H},$$

with profits $\pi^p = \dfrac{1}{2a_H}$.

The Social Evolutionary Learning Algorithm

We implement the SEL paradigm to investigate whether and/or how quickly a population of principals that can learn from each other adapts to the optimal screening contract. Specifically, the population of principals learns collectively over time, imitating the strategies (contracts) of more successful principals (those earning higher profits) and occasionally experimenting with new contracts.

A *contract* for principal i, $i \in \{1,\ldots,N\}$ at time t is a pair of real numbers, $z_t^i \equiv \{x_t^i, w_t^i\} \in Z \equiv X \times W$. The sets X and W are uniformly spaced grids[9] for outputs and wages of sizes $\#X$ and $\#W$, respectively, defined on the intervals $[0, 1.1\overline{x}]$ and $[0, 1.1\overline{w}]$, where $\overline{x}$ is largest level of x_L^* and $\overline{w}$ is the largest level of w_L^* achieved across all cost parameterizations $\{a_{L,}\ a_H\}$ that we consider.[10]

A *strategy* for principal i at time t is a pair of contracts, $s_t^i \equiv (z_{1,t}^i, z_{2,t}^i) \in G \equiv Z \times Z$, where G is the strategy space. Finally, for any t, the *strategy set*, $S_t \equiv \{s_t^i,\ldots,s_t^N\}$, where N is the size of the population of principals, consists of all principals' time t strategies. There are $\#X(\#W)$ possible output-wage pairs in Z and, hence, $(\#X)^2(\#W)^2$ feasible strategies for each principal, who chooses to play a single one each period.

Every period, each principal meets two agents (one low cost and one high cost) but does not know who is what type. Each principal, $i = 1,..,N$, then offers the two contracts prescribed by his strategy, s_t^i, to both agents, and each agent either accepts a contract or rejects both contracts. An agent cannot accept more than one contract. We assume that both the high-cost and the low-cost agents always respond optimally, by accepting or rejecting the contracts offered by comparing the utilities they would obtain in each contract as well as their reservation utility. Then, the principals compute their profits based on the outcome: total output produced and total wages paid out, according to the selected contracts. In theory, this implies that there are eight possible acceptance-rejection outcomes (four for each contract); in practice, however, some of these outcomes violate agents' self-selection constraints and would never be observed.

The principals are assumed to be boundedly rational and thus unable to solve directly the maximization problem in equation (1) – that is, they do not have the computational ability to set up first-order conditions and maximize. Moreover, they do not have any information about the preferences of each type of agent. By endowing the principals with strategies that consist of a pair of contracts, we assume only that they know that they face two different types of agents. The principals also lack information on the cost function and parameters that are part of the solution for the optimal contract.

Once the principals' payoffs are computed, updating of the principals' strategies takes place. The first step of the SEL algorithm is *replication*, which allows for potentially better-paying alternatives to replace worse ones. As our baseline replication operator, we use proportionate (biased 'roulette wheel') replication. Specifically, each strategy, s_t^i, $i = 1,\ldots,N$, in the current strategy set has the following probability of obtaining a replicate:

$$prob_t^i = \frac{\exp(\lambda \pi_t^i)}{\sum_{j=1}^{N} \exp(\lambda \pi_t^j)}, \tag{5}$$

where π_t^i is the profit that principal i earned at time t and λ is a parameter governing the relative fitness weight. Replication is thus used to generate (drawing at random with the above probabilities) a population of N replicates, r_{t+1}^j, $j = 1,\ldots,N$, of the strategies that were used in the population at period t.

The new replicate, r_{t+1}^j, however, replaces the strategy s_t^j that was previously implemented only if it yields a higher payoff. If this is not the case, we assume that the principal keeps his previous strategy. More formally, for each principal, $j = 1,\ldots,N$, the payoff of using strategy, r_{t+1}^j, $j = 1,\ldots,N$, is compared to the payoff of using the existing strategy, s_t^j – the j-th member of the strategy collection at time t. That is, the strategy at location j that has a higher payoff between r_{t+1}^j and s_t^j becomes the member of the set S_{t+1} at $t + 1$:

$$S_{t+1}^j = \max\{s_t^j, r_{t+1}^j\} \text{ for } j = 1,\ldots,N. \tag{6}$$

As a robustness check, we also did simulation runs with what we call *simple replication*, whereby each newly selected replicate, r_{t+1}^j, *always* replaces the existing strategy, s_t^j, independent of whether it yields a higher or lower payoff.

Once the replication stage is complete, *experimentation* takes place. That is, we subject each strategy, s_{t+1}^i, in the new strategy pool obtained after replication is completed to random experimentation (mutation) with probability μ. Specifically, if experimentation takes place for some principal, *j*, his strategy, s^j_{t+1}, is replaced by a new strategy randomly drawn from the strategy space *G*.[11] Both strategy elements, $z^j_{1,t+1}$ and $z^j_{2,t+1}$, undergo experimentation simultaneously. For each $k = 1, 2$, the new strategy is drawn from a square centred at $z^j_{k,t+1}$ with sides of length $2r_m$. We refer to the parameter *rm* as the 'experimentation distance,' as it determines the size of the area in the contract space[12] in the neighbourhood of the current strategy within which experimentation can take $z^j_{k,t+1}$. The strategy set, S_{t+1}, is thus updated with the new experimental strategies as applicable. The strategy-updating process continues for *T* periods. After some threshold period, $\hat{T} < T$, the experimentation rate (constant until then) is subject to decay governed by the parameter χ.[13]

To summarize the learning process in words, in each period a 'biased roulette wheel' is spun for each principal, which yields as an outcome (a replicate) one of the strategies from the current strategy set consisting of all *N* principals' strategies. Strategies with higher payoffs have higher probabilities to be replicated (copied). This replication step is the heart of the SEL algorithm. Then, for each principal, *i*, we compare the payoff of his replicate strategy with that of his own time *t* strategy, s^i_t, and, if the former payoff is larger than the latter, s^i_t is replaced with the replicate. Finally, after the replication stage, each principal's strategy can be also subject to 'experimentation,' which occurs with probability μ. That is, the strategy that the principal intended to play in period $t + 1$ is replaced (mutates) with another, (locally) randomly drawn strategy from *G*.

The learning process models the interactions in a population of principals that learns 'collectively' through gathering information about the behaviour of others and the imitation of successful strategies. Via the replication process, those strategies that yield above-average payoffs tend to be used by more principals in the following period. The experimentation stage incorporates innovations by principals, done either on purpose or by chance.

Results

Computational Implementation of the Learning Algorithm

In this section, we describe the computational procedure we followed to initialize and implement the SEL algorithm in our theoretical setting.[14] Table 12.1 shows the baseline parameters we use to initialize the learning algorithm.

Table 12.1
Baseline Parameter Values

Parameter	Values Used
Cost parameter for agent L, a_L	10 uniformly spaced points on [1, 2]
Cost parameter for agent H, a_H	10 uniformly spaced points on [3, 4]
Random seeds	70 random integers on [1, 10,000]
Population size, N	30
Simulation run length, T	2,400
Experimentation rate, μ	0.05
Experimentation distance, r_m	0.1
Experimentation decay factor, χ	0.9998
Weighting factor, λ	1
Number of grid points, $\#X$, $\#W$	100

The cost parameters are ten uniformly spaced points on the interval [1,2] for the low-cost agent L and ten on [3, 4] for the high-cost agent H. For each pair of cost parameters (100 in total), we conduct 70 runs using a different random generator seed. Thus, for a given variant of social learning (baseline or robustness run), we conduct a total of 7,000 runs. We also perform various robustness and comparative statics runs varying the parameters in Table 12.1 (see below).

The contract space from which the contracts $z_{k,t}$ are chosen is composed of all possible output-wage (x, w) pairs such that x belongs to a uniformly spaced linear grid, X, of 100 points on the interval [0, 1.1] and w belongs to a uniformly spaced linear grid, W, of 100 points on the interval [0,0.594]. Thus, there are $\#X(\#W) = 10{,}000$ distinct feasible contracts yielding 108 distinct feasible strategies in the strategy space G. The grid bounds $1.1x$ (= 1.1) and $1.1w$ (= .594) are implied by the cost parameters in Table 12.1. To start with, N strategies are randomly chosen from G at $t = 1$ and assigned an initial payoff of zero. Each run continues for $T = 2{,}400$ periods. At period $T = 2{,}000$, the experimentation rate, μ, (constant until then) begins to decay at rate χ.

Baseline Simulation Results

As described in the theory section, there are three possible outcomes to which the social learning process might converge. Since the experimentation rate decays to zero by period T, all runs converge to a single contract offered by all principals – that is, all N strategies in the time T strategy pool, S_T, are the same and correspond to one of three outcome types:

1 If each agent accepts a different contract, then the outcome is classified as a ‘separating’ outcome.
2 If the low cost agent accepts a contract and the high cost agent rejects both contracts, then the outcome is classified as an ‘excluding’ outcome.
3 If both agents accept the same contract, then the outcome is classified as a ‘pooling’ outcome.

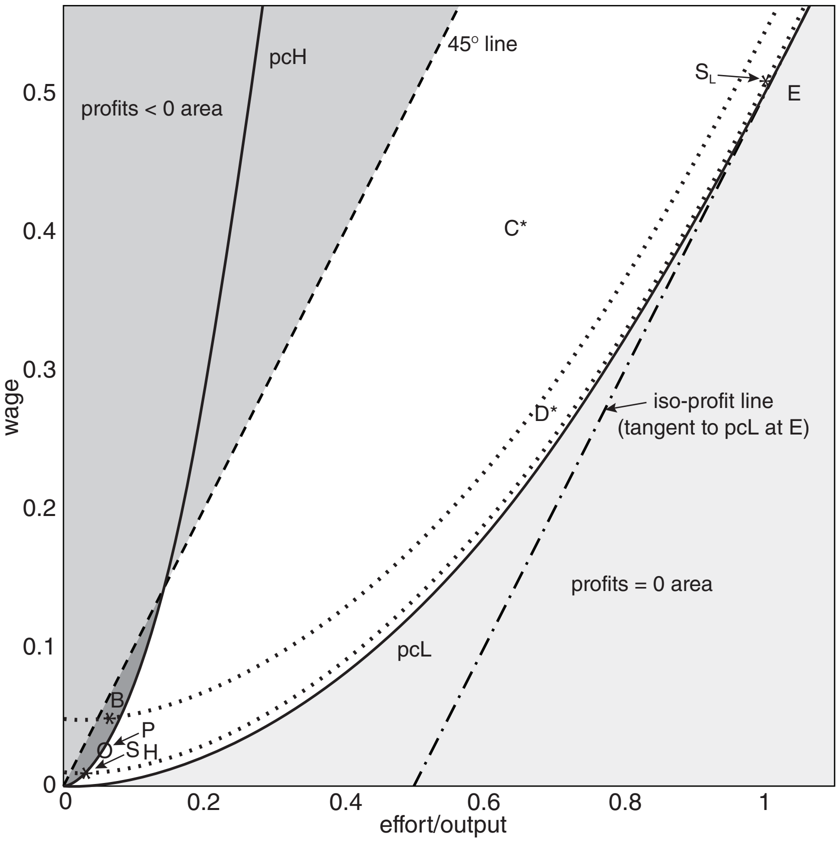
The results from the baseline simulation runs show that 24.9 per cent of all runs converge to outcomes falling into the separating category, 63.4 per cent converge to an excluding outcome, and the remaining 11.8 per cent converge to a pooling outcome.

We begin our discussion of the baseline results by looking at the payoff space that the principals face. Figure 12.1 illustrates the various possibilities in terms of wages and effort levels that matter in determining the principal’s profits from a given contract (it is hard to illustrate on a two-dimensional figure what happens in terms of total profits as the two offered contracts interact). The principal’s iso-profit curves are upward-sloping straight lines with slope 1. Profits increase moving in the east direction.

The ‘\’-pattern shaded area on the upper-left side of the figure indicates the region that generates negative profits for the principal. If a contract is offered below the participation constraint for the low-cost agent (the horizontal-line-pattern area on the right), profits from that contract are zero; note that zero is also the principal’s outside option, so profits can never be nonpositive at optimum. All other areas on Figure 12.1 generate positive profits. The crescent-shaped shaded area at the bottom left between the zero profits iso-profit line (the 45-degree line through the origin) and the PC constraint for agent H indicates the set of contracts that the high-cost agent would accept and that, at the same time, generate positive profits for the principal. Points S_L and S_H indicate the output-wage pairs corresponding to the optimal separating contract (S_L for agent L and S_H for agent H). By theory, points S_L and S_H lie on the same indifference curve of agent L. Finally, point E denotes the profit-maximizing excluding contract (which occurs at the tangency point between the line labelled pcL and an iso-profit line), and point P denotes the profit-maximizing pooling contract, as derived in the theory section.

Figure 12.2 plots the frequencies of the three possible outcomes (separating, excluding, and pooling) that the learning process results in as functions of the a_L and a_H cost values we consider. We can see from the figure

Figure 12.1: Principal's Payoff Space



that the excluding outcome is by far the most frequent. However, an interesting observation is that the different outcomes achieve their highest frequencies for distinct regions of the cost parameter space. Specifically, the pooling outcomes' highest frequencies are observed in the area where there is the smallest difference between a_L and a_H. This is also exactly the area where the frequency of 'excluding' outcomes is the lowest. The intuition for this is that, for relatively low a_H and relatively high a_L, the cost functions of both agents are most similar and the chances that agent H will accept a contract are the highest. A question that remains at this point is why such pooling outcomes occur at all – that is, why the principal does not move one of the offered contracts below pcL and also move the contract for agent L toward the optimal excluding contract that lies on that agent's participation constraint, which would result in higher profits. A possible reason is that, as we move in an easterly direction, the principal's profits can drop locally. Similarly, why is the separating contract so hard to learn so that, as a result, most of the runs converge to an excluding outcome? These issues are further illustrated and discussed in detail below.

Next, we report results on the rate of convergence in our baseline simulation runs as defined by the parameters in Table 12.1. We define and examine the behaviour of several measures that reflect both qualitative and quantitative aspects of the learning dynamics. These measures refer both to the overall performance of the learning process relative to the optimal (separating) contract and to its performance within each of the three possible outcome classes (separating, excluding, and pooling).

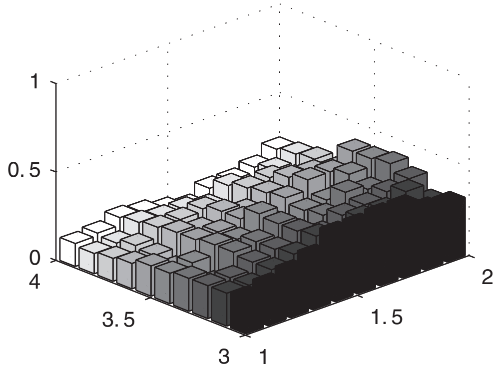
One measure is the time paths of the fractions of all runs with average payoffs within a given 'distance' from the profit-maximizing contract. Each point on the plotted line equals the average fraction over all strategies over the 7,000 runs. Three distance criteria are considered: 0 per cent, 5 per cent, and 10 per cent, which refer to the percentage deviations from the theoretically maximum profits (overall or conditional on the outcome class). Another measure is the frequency distribution over all simulations of the differences between simulated and profit-maximizing[15] payoffs of all strategies in the final period.

The four panels of Figure 12.3 display the time paths of the fractions of all runs with offered contracts that result in average principals' total profits (at each t we take the average over N) within a given percentage deviation of the theoretically maximal profits overall (the top panel) and conditional on each of the three outcome classes. The top panel represents the fractions over time of all runs attaining average profits within 0, 5, and 10 per cent of the maximum profits. These fractions remain quite low over the entire 2,400 periods, indicating that overall convergence to the optimal separating contract does not occur in the majority of cases. The lower three panels help clarify the reason for the low overall convergence rates – namely, because the 7,000 runs are actually split among the three possible convergence outcomes (separating, excluding, and pooling). Thus, even if convergence turns out to be quite good within each outcome class, the overall performance, as measured relative to the optimal separating contract, is quite unsatisfactory.

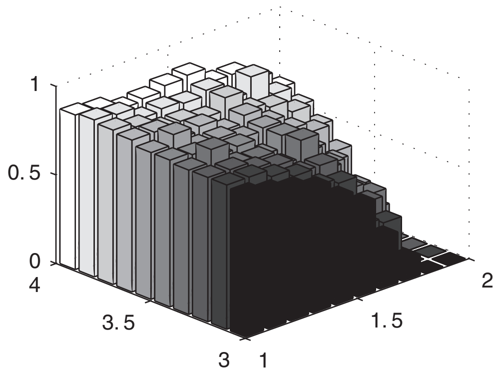
To address the latter point further, let us look more closely at the algorithm performance within each outcome class. The second panel of Figure 12.3 shows the time paths of the fraction of runs that fall into the

Figure 12.2: Frequencies of the Three Possible Outcomes

Baseline: frequency of convergence to a separating contract as function of agents' costs



Baseline: frequency of convergence to an excluding contract as function of agents' costs



Baseline: frequency of convergence to a pooling contract as function of agents' costs

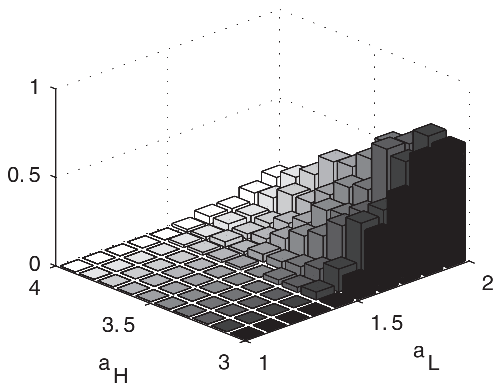
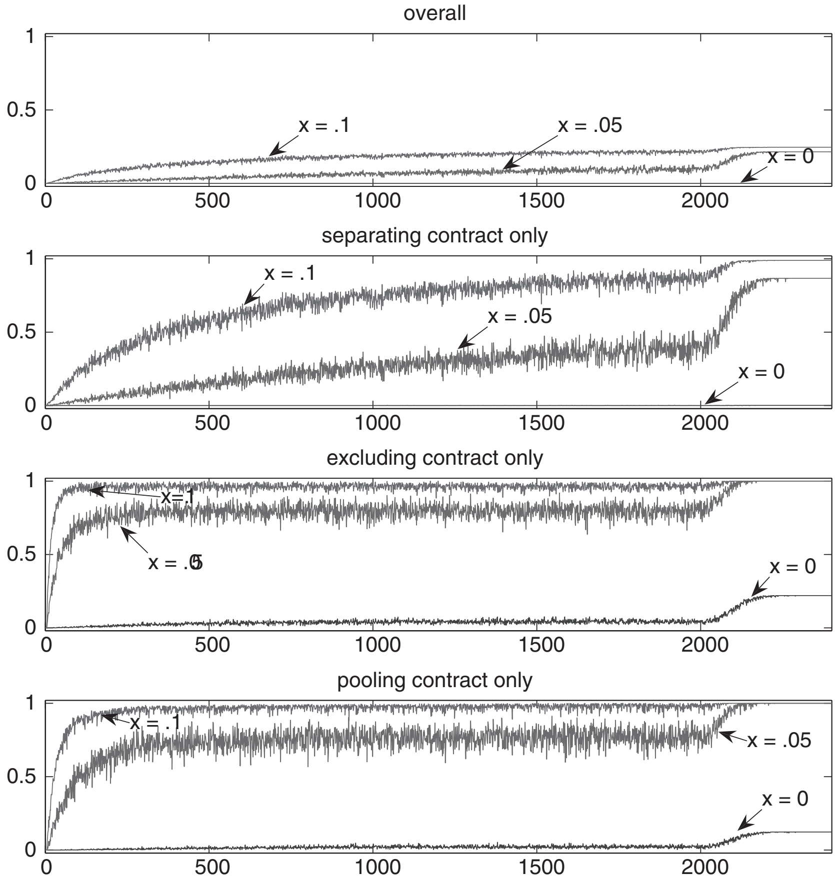


Figure 12.3: Time Paths of the Fractions of Runs with Average Payoffs within x% of the Payoff of the Profit-Maximizing Outcome



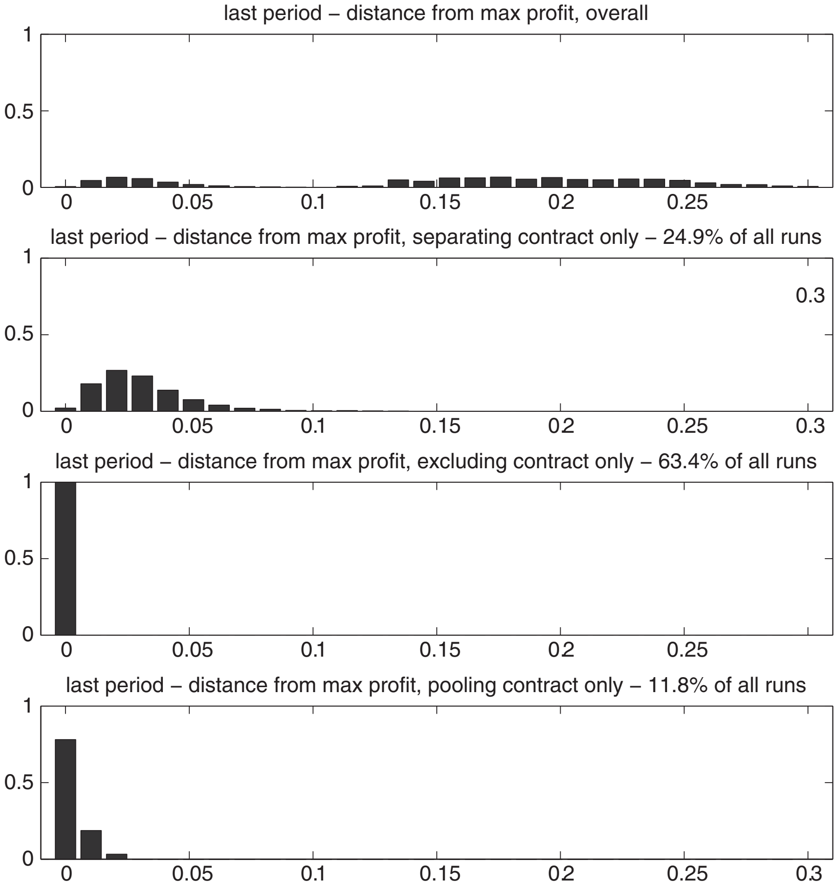
'separating contract' category, based on what type of contract (as defined in the beginning of this section) they converge to at time T. The fraction of runs with profits equal to those in the optimal contract remains very close to zero over time. However, the fraction of runs coming within 5 per cent of the optimal separating contract profits increases over time and reaches more than 85 per cent at period 2,400. The fraction within 10 per cent of the theoretically maximum profits increases even more substantially over time and almost reaches the maximum value of 1 at the end.

The remaining two panels show the same graphs for the runs that reach excluding and pooling outcomes at time *T*. The fraction of runs that reach average profits within 0 per cent – that is, that coincide with the best pooling or excluding contracts – naturally increases sharply at the end of the runs as the rate of experimentation goes to zero.[16] Further, in contrast to the separating case, the 'within 5 per cent' and 'within 10 per cent' profits deviation time paths reach the maximum value for both outcomes, indicating that the separating contract is the hardest to learn. As the figure shows, the payoff fractions exhibit some volatility. This is a result of the fact that small changes in the offered contracts could result, because of the rugged payoff landscape, in relatively large fluctuations in payoffs (see more on this below).

The four panels of Figure 12.4 show the distribution (histogram), computed over all 7,000 runs, of the distance between the average realized profit and the profit-maximizing outcomes in the last period. The first panel illustrates the distribution of the distance in terms of profits from the optimal separating contract over all 7,000 runs – that is, these are the overall profit 'errors' or absolute differences relative to the optimum. The panel illustrates that only a tiny fraction of all runs comes close to the payoffs associated with the optimal separating contract. This reflects two facts. First, a minority (about 25 per cent) of all runs converges to a separating outcome. All other runs converge to other outcomes that have strictly lower profits. Second, even among the runs that converge to a separating outcome, only a small percentage gets close to the optimal separating contract. This latter point is evident in the second panel of Figure 12.4, which displays the distribution of profit distances from the optimal separating contract profit only within runs with separating learning outcomes. Still, cumulatively, more than 60 per cent of these runs achieve payoffs within 5 per cent of the optimal contract's payoff.

Finally, panel 3 of Figure 12.4 shows the distribution of percentage differences between the realized and theoretically maximal profits only for runs classified as excluding outcomes based on the values of $(z_{1,T}, z_{2,T})$. Evidently, all simulations that converge toward excluding outcomes reach payoffs equal or very close to the payoff from the profit-maximizing excluding contract. Finally, the last panel of the figure shows the profit errors distribution for runs classified as 'pooling' outcomes. As in the excluding case, a large fraction of these runs reaches the payoff of the profit-maximizing pooling contract, while the rest come within 2.5 per cent of it.

Figure 12.4: Histograms of Last-Period Outcomes (distance from maximum payoff)



Robustness Runs

In addition to the baseline runs defined by the parameters in Table 12.1, we also conducted numerous robustness simulation runs (see Table 12.2). Apart from the baseline run as the benchmark, the table lists numerous variations of the learning algorithm and its parameters that we implemented to test the robustness of our results: 'selective experimentation,'

Table 12.2
Types of Learned Contract Outcomes

	Fraction of All Runs Converging to:		
	Separating Contract	Excluding Contract	Pooling Contract
		(*percent*)	
Baseline	24.9	63.4	11.8
Robustness			
Selective experimentation	22.8	64.7	12.5
Normally distributed experimentation	24.9	61.5	13.6
Tournament selection	23.1	64.1	12.7
SEL weighting factor, $\lambda = 3$	25.9	62.6	11.5
Simple replication	12.8	73.4	13.8
Experimentation distance, $r_m = 2$	77.8	14.7	7.5
$N = 100$	28.2	63.5	8.3
Coarse grids ($\#X = \#W = 10$)	27.5	61.6	10.9
Optimal contract in initial pool	99.9	0.1	0.0
Optimal contract H only in initial pool	31.8	51.8	16.4

experimentation with normally distributed draws, a 'tournament selection' replication operator, 'simple replication,' a 'large' experimentation distance ($r_m = 2$), an increase in the size of the pool of principals to $N = 100$, and the use of a coarser strategy space. We also conducted runs with both optimal separating contracts (x_L^*, w_L^*) and (x_H^*, w_H^*) or only the high-cost agent's optimal contract (x_H^*, w_H^*) included in the initial strategy pool, S_0.

Table 12.2 demonstrates that the excluding contract as an attractor for the largest fraction of learning outcomes remains robust across most of the different variants of our social learning algorithm. The only exceptions are the treatments where we explicitly include the optimal separating contracts for both agents in the initial pool, and when we increase the experimentation radius to 2 so that the area of experimentation covers the whole contract space (global experimentation). In the treatment with the separating contract in the initial strategy pool, virtually all runs (99.9 per cent) converge to separating outcomes. This is reassuring with regard to the local stability of the optimal contract (once in the pool it is not knocked out during the learning process). It also shows, however, how hard it is to learn the optimal screening contract in our setting: if the principals are somehow 'shown' what is optimal, they stick to this strategy; otherwise they seem to struggle to find it and instead adapt to other outcomes (most likely excluding, but sometimes pooling), depending on the cost parameters and random seeds. In contrast, the runs in which we included (x_H^*, w_H^*) but not

(x_L^*, w_L^*) in the initial strategy pool still feature mostly excluding outcomes, although the fraction of runs with separating outcomes is significantly higher relative to the baseline and most other treatments.

The second-highest percentage of separating outcomes (77.8 per cent) occurs in the runs conducted within a large experimentation range parameter, $r_m = 2$, that covers the whole strategy space, G – that is, where experimentation is not local, as in the baseline, but global. Intuitively, this should help break away from the local maximums associated with the best excluding and pooling contracts, and so it does: the fraction of runs converging to a separating outcome more than triples relative to the baseline. Below we also show that the 'separating contract in the pool' runs converge very close to the optimal separating contract. Finally, observe that the lowest fraction of outcomes (16.4 per cent) across all of the treatments are pooling.

Next, we study the SEL algorithm performance in terms relative to the optimal contracts overall and within each outcome class. Table 12.3 shows how close the contracts to which SEL converges are, in terms of principals' profits, to the profit-maximizing contracts, both overall and within the sets of runs classified in each of the separating, excluding, and pooling categories. The first three columns report the fractions of runs with final-period profits within a given distance of the optimal (separating contract) profits – that is, the SEL algorithm's overall performance. We report these fractions for runs with profits within 0, 5, and 10 per cent deviation from the theoretical maximum. As Figure 12.4 also shows, these fractions are quite low for the baseline specification: 0 per cent for the 0 per cent deviation criterion, 21.6 per cent for the 5 per cent deviation criterion, and 24.6 per cent for the 10 per cent deviation criterion. As the rest of Table 12.3 shows, however, the main reason for this bad performance overall is the high incidence of convergence to other (suboptimal) contractual outcomes, pooling or excluding, and not necessarily bad convergence to a separating outcome when such is learned.

Looking at the profit differences with the theoretical maximum within or, conditional on, each outcome type, the percentage of runs with average final-period profits exactly equal to those in the profit-maximizing contract in each category (the $x = 0$ per cent columns in Table 12.3) is fairly low. Based on this 'zero distance' criterion, the baseline runs with excluding outcomes at T yield the highest fraction (22.0 per cent) of runs converging exactly to the profit-maximizing excluding contract, compared with 12.4 per cent for the class of pooling outcomes and only 0.2 per cent within the runs with separating outcomes.

These percentages, however, increase significantly when we allow for a tolerance of within 5 per cent or 10 per cent of the theoretically maximum

Table 12.3
Social Learning Performance

Run	Percentage of Last-Period Payoffs within x% of Optimal Payoff (overall and per contract type converged to)											
	Overall			Separating Contract Only			Excluding Contract Only			Pooling Contract Only		
	x = 0	x = 5	x = 10	x = 0	x = 5	x = 10	x = 0	x = 5	x = 10	x = 0	x = 5	x = 10
Baseline	0.0	21.6	24.6	0.2	86.7	98.9	22.0	100.0	100.0	12.4	100.0	100.0
Robustness												
Selective experimentation	0.0	19.8	22.6	0.1	87.1	99.4	22.9	100.0	100.0	12.1	100.0	100.0
Normally distributed experimentation	0.0	20.7	24.5	0.1	83.2	98.6	23.6	100.0	100.0	11.3	100.0	100.0
Tournament selection	0.0	20.1	22.9	0.1	86.9	99.0	22.7	100.0	100.0	12.5	100.0	100.0
SEL weighting factor, $\lambda = 3$	0.1	22.5	25.7	0.2	87.0	99.3	22.3	100.0	100.0	11.3	100.0	100.0
Simple replication	0.0	1.1	3.9	0.0	8.7	30.4	0.1	33.8	62.5	0.2	23.6	49.2
Experimentation distance, $r_m = 2$	0.0	4.0	23.6	0.0	5.1	30.3	6.3	100.0	100.0	2.7	98.5	100.0
$N = 100$	0.4	27.9	28.2	1.4	99.1	100.0	23.7	100.0	100.0	9.5	100.0	100.0
Coarse grid ($\#X = \#W = 10$)	8.0	16.5	18.2	29.0	60.1	66.1	0.0	38.6	78.5	0.0	0.0	43.1
Optimal contract in initial pool	99.2	99.8	99.9	99.2	99.9	100.0	0.0	100.0	100.0	0.0	100.0	100.0
Optimal contract H only in initial pool	0.1	27.8	31.5	0.2	87.4	99.1	22.6	100.0	100.0	12.8	100.0	100.0

profits within each outcome category. Overall, the optimal separating contract is the hardest to adapt to, even conditionally, within the runs with separating outcomes, with 86.7 per cent reaching profits within 5 per cent of the maximum payoff in the baseline and 98.9 per cent reaching profits within 10 per cent of the theoretical maximum. In contrast, all of the runs with excluding or pooling outcomes yield profits within 5 per cent of the theoretically maximum profits within their corresponding class – that is, convergence to the profit maximizing contracts in the excluding and pooling cases is much better than in the separating case.

The remaining rows of Table 12.3 report results from various robustness runs in which we vary the parameters of the baseline social learning algorithm or modify the learning operators. The 'selective experimentation,' experimentation drawing from a normal distribution, 'tournament selection,' and the $\lambda = 3$ (where we place a larger weight on the best-performing strategy) treatments all result in numbers that are very close to those in the baseline, demonstrating the robustness of our findings in these dimensions of the SEL algorithm.

In contrast, using 'simple replication' leads to uniformly much lower fractions of runs that yield profits within a given distance of the theoretically maximum profits, both overall and within each class (compare with the baseline row of Table 12.3), as some good strategies can be replaced by worse ones during the learning process.

Interestingly, the 'large experimentation distance' specification also performs quite poorly in terms of its overall adaptation to the optimal contract. While we saw in Table 12.2 that, under this treatment, many more runs than in the baseline converge to a separating outcome, because of the extra volatility introduced by the large experimentation range, these runs on average do not get near the optimal contract, as demonstrated by the low numbers in Table 12.3. For example, for $r_m = 2$, only 5.1 per cent of the runs with separating outcomes are within 5 per cent of the theoretically maximum profits compared with 86.7 per cent in the baseline. That is, while allowing for global experimentation, we can achieve better performance in the qualitative sense – the 'correct' type of contract is learned – but this comes at the expense of worse quantitative performance in terms of getting close to the theoretically optimal profit level.

Increasing the number of strategies/principals, N, from 30 to 100 naturally leads to better performance than the baseline but the differences are not large. Using coarser grids ($\#X = \#W = 10$) helps the principals adapt to the optimal contract more often (in 8 per cent of the runs overall), but with only 100 feasible points in total in the strategy space G, the chances are simply mechanically quite higher than are the case with 10,000 points, as in the baseline. Nevertheless, the overall percentage of runs within 5 per cent or 10 per cent of the theoretically maximum profit in this treatment is lower than in the baseline.

Finally, the performance results for the treatment in which we include the optimal separating contract in the initial pool are very different. Here, 99.9 per cent of all simulation runs converge to a separating outcome (see Table 12.2) and, furthermore, almost all (99.2 per cent) converge to the optimal separating contract even though we do nothing to 'protect' this strategy from being extinguished by replication or changed by experimentation. On the other hand, the simulations with only one of the elements of the optimal contract (x^*_H,w^*_H) in the initial strategy pool result in performance outcomes and convergence rates quite close to those in the baseline.

Discussion

Why do our principals have such a hard time figuring out what the optimal separating contract is? The principals modify and adapt their strategies in response to the payoffs they receive. However, for the problem at hand, the payoff landscape as a function of x and w is extremely rugged and shifts shape over time as a result of the interaction between the principals' actions (the specific contract offers) and the agents' optimal responses. It is important to point out here that, despite the ruggedness of the payoff landscape for the principals, this is not due to the agents' behaviour changing over time – agents are fully rational and always pick the best (for them) of the two contracts offered – but instead is due to the discrete shifts in agents' choices as the offers are varied.

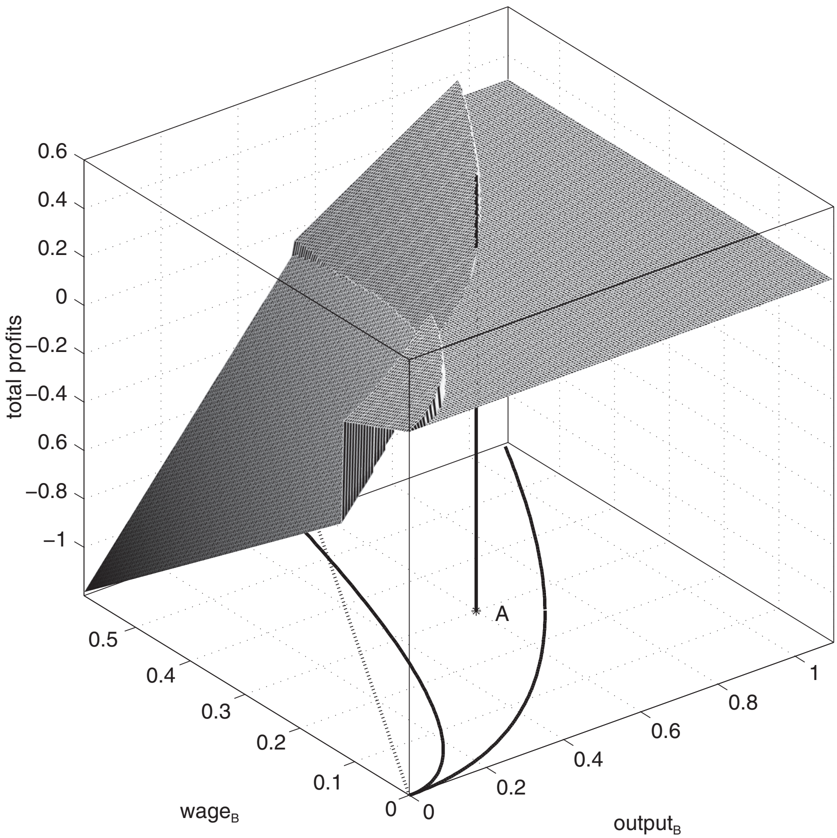
To illustrate this point further, look back at Figure 12.1, which shows why learning the optimal separating contract (S_L and S_H) might be quite hard. For example, suppose one of the two wage-effort pairs offered is (by chance or design) at S_L. Then, any point such as B would cause agent L to switch to it (the dotted line through B is agent L's indifference curve). In fact, both agents would choose B if B and S_L were offered simultaneously. Clearly, as a result, we would move away from the optimal contract for agent L, (S_L), even if we start quite close to it, and only a point such as C on the graph (and not, for example, D or, in general, any point between the indifference curves through S_L and B, no matter how close to S_L) might attract agent L away from the 'wrong' area of the wage-effort space near B. Thus, in a given run, even if a principal 'discovers' a separating contract, even one near the optimal one, it is relatively easy, due to the randomness in the replication and experimentation process, for this contract to disappear after being replaced by a pooling or excluding outcome that generates higher profits.

In contrast, suppose one of the offered contracts is point E – the optimal excluding contract. Then, the principal would not choose any other points nearby and above the line pcL as that would reduce his profits (since agent L would switch). Even if we explicitly introduce point S_H into the pool, profits would be lower than those at E (as both agents would switch to point S_H), and the principal would experience this and go back to offering E. Thus, point E can be locally stable and robust to experimentation in terms of profits generated, unless a principal manages to deviate from it in the direction of S_L and simultaneously offer another contract close to S_H (but such that agent L does not take it), which seems quite hard.

Figure 12.5 helps to explain further the difficulties with converging to the optimal contract by illustrating the principal's total profits, plotted as a function of varying one of the offered contracts over the whole wage-output space while fixing the other contract[17] (point A on the figure, chosen between pcH and pcL). Note first the flat area on the bottom right of the figure (bounded by the axes, pcH, and agent L's indifference curve through point A). For any contract offered in that area, agent L chooses contract A, while agent H chooses not to participate. Note also the two local maximums: one at the bottom left (where agent H chooses such a contract while agent L stays at A) and one at the top right (where the maximum profits are achieved by an excluding contract). These local extrema of the joint-profit function might make learning the optimal separating contract quite hard. For example, if the other contract (given A is offered) is in the northeast region of the (x, w) space, it is likely that the principal's strategy pool would converge to an excluding contract. The same is true if we start in the flat area to the right of pcL. Fixing point A, it might seem possible that, if the other contract is nearby, the optimal separating contract for agent H (the local peak at the bottom left 'ridge') can be learned, but if we allow point A to move as well (which it does in reality), agent L at some point might switch to that contract, too (see the discussion on Figure 12.1 above), which would reduce the principal's profits.

Finally, as we show in Arifovic and Karaivanov (2009), given that the maximum profits are achieved by contracts located right on the constraints – that is, on the 'cliffs' of the two elevated 'ledges' – profits drop discontinuously for a small mutation to the right, which again creates problems for the learning process.[18]

Figure 12.5: Principal's Total Profits from Fixing One Contract (A) and Varying the Other (B)



Concluding Remarks

In this paper, we have investigated the dynamics of social evolutionary learning in the context of a model of optimal monopolistic screening. Principals learn about what wage/work effort contracts to offer based only on observable information – namely, principals' profits. They have no other information about the costs of two types of agents they face or the agents' preferences in general. In addition, the principals do not have the computational ability to solve for the optimal screening contract. Instead, they learn over time through a process of trial and error by imitating the contracts of other principals who have made larger profits in the past and by occasional experimentation with new contracts.

Our results suggest that it is rather hard to learn the optimal separating contract in this environment. Instead, most frequently (63.4 per cent of the 7,000 baseline runs), the principals' adaptation process converges toward a contract that excludes the high-cost agent. Next in terms of frequency is the separating outcome (24.9 per cent). A pooling outcome (where a contract is offered that both types of agents accept) is also possible (11.8 per cent of the baseline runs). Even if a separating outcome is achieved, exact convergence to the optimal contract is rare. In contrast, when the adaptation process leads to a pooling or an excluding outcome, the resulting offered contract gets very close to the profit-maximizing contract.

In terms of possible extensions, it might be interesting to study the performance of our learning algorithm in a competitive screening setting where principals offer contracts that compete for the same agents. In that case, only the two contracts in the strategy pool that yield the best payoffs for the agents would be chosen, in contrast to our current 'local monopoly' setting in which various types of contracts are chosen by some agents at the same time. *Ex-ante* it is not clear how this would affect the learning algorithm performance. An extra layer of strategy reinforcement would be introduced by the competition, but there also would be many more strategies that no agents would take and that would produce zero payoffs, which might 'stall' the learning dynamics, especially given that the optimal contract must also feature zero profits for the principals.

NOTES

1 We thank Greg Dow, an anonymous referee, and the participants in the 2008 Conference in Honour of B. Curtis Eaton for their extremely helpful comments and suggestions. Nick Kasimatis and Sophie Wang provided excellent research assistance. Both authors acknowledge the support of the Social Sciences and Humanities Research Council of Canada.

2 The theoretically optimal contract satisfies the incentive compatibility and participation constraints with equality, so, for a very small perturbation to the contract that violates the constraints, the principal's payoff drops discontinuously.

3 That is, the payoff achieved by the optimal separating contract.

4 In total, there were 7,000 runs, with different cost parameters and random seeds. The random seeds determined the random number draws that were used in the probabilistic operators of the social learning algorithm.

5 This is not essential. As long as the principal knows the fractions of each type in the population, our results generalize easily, assuming that the principal maximizes (population-weighted) total expected profits.

6 Comparing the profits in each case, it is easy to verify that such exclusion would pay off if $c_L(x_H^*) - 2c_H(x_H^*) + x_H^* < 0$, where x_H^* solves $c_H^{'}(x_H) = 1$. This inequality would hold if the profit from agent H, $x_H^* - c_H(x_H^*)$ is not high enough to compensate for the cost differential, $c_L(x_H^*) - c_H(x_H^*)$.

7 The constraint $w_L - c_H(x_L) < 0$ is implied by $w_L - c_L(x_L) = 0$ and thus is omitted.

8 Clearly, it is impossible to offer a contract that excludes only agent L. Excluding both agents yields a zero payoff and never occurs as an outcome of the learning process (each of the three other outcomes has a positive payoff).

9 In principle, we can use continuous sets in the numerical implementation of the algorithm. However, learning is likely to be even harder in that setting.

10 Note that, by the results from the second section, all other contracts of interest, (x_H^*, w_H^*), (x^P, w^P), and (x^e, w^e), feature lower x and w compared with (x_H^*, w_H^*) and so always lie strictly within the space $X \times W$.

11 Our baseline simulation uses uniformly distributed draws. As a robustness check, we also implement experimentation with normally distributed draws.

12 We conducted sets of simulations using both 'small' experimentation distance ($r_m = .1$) and 'large' distance ($r_m = 2$). The latter covers the entire strategy space G. If the drawn contract falls outside the edges of the strategy space, it is replaced by the nearest element on the edge. In addition, we conducted sets of simulations with 'selective experimentation,' where the new randomly drawn experimental strategy replaces the existing one only if it yields a higher payoff, similarly to that the replication operator.

13 That is, the experimentation rate at time t follows: $\mu_t = \mu_{t-1}\chi^{t-\bar{T}}$ for all periods, $t \in [T, T]$.

14 The MATLAB code for all simulations is available from the authors upon request.

15 Profit maximizing refers to the respective of the three contract outcomes.

16 This indicates that, in those cases, it is the continued experimentation that prevents earlier convergence and, if we start decreasing the experimentation rate earlier, adaptation would obtain even sooner.

17 It is impossible to illustrate on a single figure the total profits when varying both offered contracts simultaneously.

18 A possible extension that could alleviate this problem is to allow the agent's participation decision to be probabilistic – that is, to have the principal's payoff go down continuously near the constraint. This would be the case, for example, if the agent needed to learn about her optimal action and the location of the participation constraint.

REFERENCES

Akerlof, G., 1970. 'The Market for 'Lemons': Quality Uncertainty and the Market Mechanism.' *Quarterly Journal of Economics* 84 (3): 488–500.

Arifovic, J. 2000. 'Evolutionary Algorithms in Macroeconomic Modeling: A Survey.' *Macroeconomic Dynamics* 4 (3): 373–414.

Arifovic, J., and B.C. Eaton. 1995. 'Coordination via Genetic Learning.' *Computational Economics* 8 (3): 181–203.

– 1998. 'Evolution of Communication in a Sender/Receiver Game of Common Interest with Cheap Talk.' *Journal of Economic Dynamics and Control* 22 (8-9): 1187–1207.

Arifovic, J., and A. Karaivanov. 2009. 'Learning by Doing vs. Learning from Others in a Principal-Agent Model.' Working paper, Simon Fraser University.

Maskin, E., and J. Riley. 1984. 'Monopoly with Incomplete Information.' *RAND Journal of Economics* 15 (2): 171–96.

Stiglitz, J., and A. Weiss. 1981. 'Credit Rationing in Markets with Imperfect Information.' *American Economic Review* 71 (3): 393–410.