Cross-Validation in Multiagent-based Simulation: Analyzing Evolutionary Bargaining Agents

Keiki Takadama
ATR Human Information Science Labs.
2-2-2 Hikaridai
Seika-cho, Soraku-gun
Kyoto 619–0288 Japan
keiki@atr.co.jp

Yutaka L. Suematsu
National University of Engineering
Av. Tupac Amaru 210 - Rimac
Apartado 1301 - Lima 100
Lima, PERU
suematsu@terra.com.pe

Norberto E. Nawa
ATR Human Information Science Labs.
2-2-2 Hikaridai
Seika-cho, Soraku-gun
Kyoto 619–0288 Japan
eiji@atr.co.jp

Katsunori Shimohara
ATR Human Information Science Labs.
2-2-2 Hikaridai
Seika-cho, Soraku-gun
Kyoto 619–0288 Japan
katsu@atr.co.jp

Abstract

This paper addresses *cross-validation* in multiagent-based simulation by analyzing evolutionary agents in a bargaining game in game theory. In particular, this paper focuses on analyzing different *learning mechanisms* and *knowledge representation* capabilities applied to agents for cross-validation. To investigate these issues, we compare the following two cases: (1) agents employing an evolutionary strategy (ES) and agents employing a learning classifier system (LCS) as different learning mechanisms; and (2) agents handling an ordinary explanation of numbers and agents handling a limited explanation of numbers as different knowledge representation capabilities. An intensive comparison of simulation results reveal the following implications: (1) simulation results by ES-based agents show the same tendency in game theory but those by LCS-based agents do not; and (2) even simulation results by ES-based agents become strange when the agents are restricted to handling only a real number with two decimal digits instead of an ordinary real number in a negotiation process between the agents in the bargaining game.

Keywords: multiagent-based simulation, cross-validation, modeling agents, bargaining game, learning mechanism, knowledge representation

* Paper submitted to the 2002 Genetic and Evolutionary Computation Conference (GECCO'02)

1 Introduction

Recently, research based on multiagent-based simulation (Moss and Davidsson, 2001) or agent-based simulation (Axelrod, 1997) has attracted a lot of attention owing to not only improvements in computational power and simulation techniques but also to the availability of an alternative way of understanding complex social phenomena. Such research has employed computational techniques to provide tools for analyzing social dynamics and has contributed to the creation of theories by clarifying vague, intuitive, or under-specified issues in conventional approaches. Although, many useful implications have been found through computer simulation, the *validation*[1] of simulation results and computational models remains an open issue. To overcome these problems, Axtell and his colleagues claim the importance of investigating whether two different models can produce the same results in terms of validating the results and models (Axtell et al., 1996).[2]

It should be noted, however, that results derived by computer simulation are sensitive to how agents are modeled. This indicates that two models may not show the same results even if a different part between two models is very small. Examples include two models which difference is only *learning mechanisms* or *knowledge representation capabilities* of agents. To clarify such parts, this paper starts by comparing two computational models that are implemented differently in terms of the learning mech-

[1] Burton argues that computational validity is a balance of three elements: (1) the question or purpose, (2) the computational model, and (3) the experimental design (Burton and Obel, 1995).

[2] They call this concept the "alignment of computational models" or "docking".

anisms and knowledge representation capabilities of agents.[3] Here, we call this type of investigation *cross-validation*,[4] which means to validate the results and models against different implementation of agents. Note that this differs from the usual meaning of cross-validation that checks the results of a single learning algorithm or a single knowledge representation on a reserved set of data. As a concrete domain, we employ a bargaining game (Muthoo, 2000) for this comparison, because this game is one of the fundamental examples and because rational behaviors of agents have already been analyzed in game theory (Osborne and Rubinstein, 1994).

This paper is organized as follows. Section 2 starts by explaining the bargaining game. A concrete implementation of agents is described in Section 3. Section 4 presents computer simulations and Section 5 discusses the cross-validation of results and models. Finally, our conclusions are made in Section 6.

2 Bargaining game

A bargaining game (Muthoo, 2000) was studied in the context of game theory (Osborne and Rubinstein, 1994). This study addressed the situation where two or more players (or agents) try to reach a mutually beneficial agreement through negotiations, and investigated when and what kinds of offers of an individual player can be accepted by other players.

To understand the bargaining game, let's give an example. In Rubinstein's work (Rubinstein, 1982), he illustrated a typical situation using the following scenario: two players, A_1 and A_2, have to reach an agreement on the partition of a "pie". For this purpose, they alternate offers describing possible divisions of the pie, such as "A_1 receives x and A_2 receives $1-x$ at time t", where x is any value in the interval $[0,1]$. When a player receives an offer, the player decides whether to accept it or not. If the player accepts the offer, the negotiation process ends, and each player receives the share of the pie determined by the concluded contract. Otherwise, the receiving player makes a counter-offer, and all of the above steps are repeated until a solution is reached or the process is aborted due to some external reason (*e.g.*, the number of negotiation processes is finite or one of the players leaves the process). In the case that the negotiation process is aborted, both players can no longer receive any share of the pie.

Here, we consider the finite-horizon situation, where the maximum number of steps (`MAX_STEP`) in the game is fixed and all players know this information as common knowledge (Ståhl, 1972). In the case where `MAX_STEP` $= 1$ (also known as the *ultimatum game*), agent A_1 can make the only offer and then A_2 can accept or refuse it. If A_2 refuses the offer, both agents receive nothing. Since a rational agent is based on the notion of "anything is better than nothing", a rational A_1 tends to keep most of the pie to herself by offering only a minimum share to A_2. Since there are no further steps to be played in the game, a rational A_2 inevitably accepts the tiny offer. By applying a backward induction reasoning to the situation above, it is possible to perform simulation for `MAX_STEP` > 1. For the same reason of the ultimatum game, the agent who can make the last offer is better positioned to receive the larger share by offering a minimum offer (Ståhl, 1972). In this case, the last offer is granted to the agent that does not make the first offer if `MAX_STEP` is even, because each agent is allowed to make at most `MAX_STEP`/2 offers. On the other hand, the last offer is granted to the same agent that makes the first offer if `MAX_STEP` is odd.

After this session, we use the terms "payoff" and "agent" instead of the terms "share" and "player" for their wide meanings in the bargaining game.

3 Modeling agents

To implement agents in the framework of the bargaining game, this section starts by modeling a basic part of agents and then modeling the parts of our focus.

3.1 Modeling of a basic part

For a basic part of modeling agents, we implement the following components of agents as shown in Figure 1. Note that each agent has the same architecture.

< Memory >

- **Strategies memory** stores a set of strategies (the number of strategies is n in this figure), which consist of fixed numbers of pairs of offers (O) and thresholds (T), and the worth of the strategies (w). These strategies are similar to those used in (Oliver, 1996). The offer and threshold values are encoded by floating point numbers in the interval $[0,1]$, while the worth values are calculated as averages of acquired payoffs. In this model, agents independently store different strategies, which are initially generated at random.

[3] Moss classifies the validation issues as (1) predictions, (2) agent and mechanism designs, and resulting outputs as descriptions (Moss, 2001). From this viewpoint, our focus is related to the second point in his classification.

[4] Carley also claims the same point using the term *cross-model validation* (Carley and Gasser, 1999).

- **Selected strategy memory** stores the one strategy selected to confront the strategy of an opponent agent. Figure 1 shows the situation where agent A_1 selects the xth strategy while agent A_2 selects the yth strategy.

< **Mechanism** >

- **Learning mechanism** varies both offer and threshold values in order to generate good strategies that acquire a large payoff. The detailed mechanism is described later.

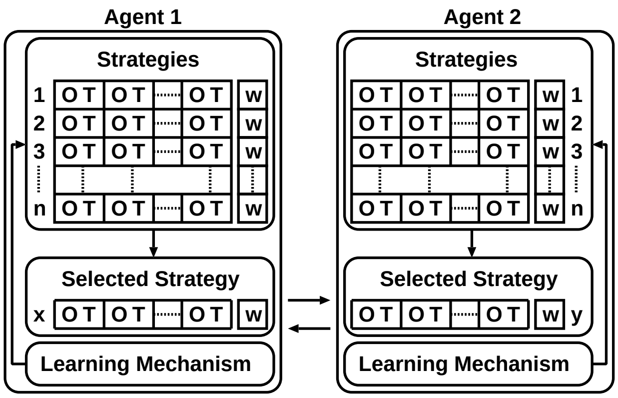


Figure 1: **Agent architecture**

As a concrete negotiation process, agents proceed as follows. Defining $\{O,T\}_i^{A_{\{1,2\}}}$ as the ith offer or threshold value of agent A_1 or A_2, agent A_1 starts with the first offer $O_1^{A_1}$. Here, we count one *step* when either agent makes an offer. Then, A_2 accepts the offer if $O_1^{A_1} \geq T_1^{A_2}$; otherwise, it makes a counter-offer $O_2^{A_2}$, *i.e.*, the offer of A_2. This cycle is continued until either agent accepts the offer of the other agent or the maximum number of steps (`MAX_STEP`) is exceeded. To understand this situation, let's consider the simple example where `MAX_STEP`$=10$, as shown in Figure 2. Following this example, A_1 starts by offering 0.01 to A_2. However, A_2 cannot accept the first offer because it does not satisfy the inequality of $O_1^{A_1}(0.01) \geq T_1^{A_2}(0.99)$. Then, A_2 counter-offers 0.01 to A_1. Since A_1 cannot accept the second offer from A_2 because of the same reason, this cycle is continued until A_1 accepts the 10th offer from A_2 where the offer satisfies the inequality of $O_{10}^{A_2}(0.01) \geq T_{10}^{A_1}(0.01)$. If the negotiation fails, which means exceeding the maximum number of steps, both agents can no longer receive any payoff, *i.e.*, they receive 0 payoff. Here, we count one *confrontation* when the above negotiation process ends or fails.

Furthermore, the worth of each strategy is calculated by the average of payoffs acquired in a fixed number

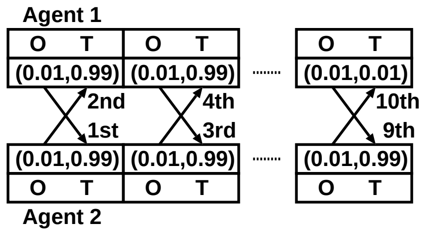


Figure 2: **An example of a negotiation process**

of confrontations (`CONFRONTATION`), where the strategies of the other agents are randomly selected in each confrontation. For example, the xth strategy of A_1 in Figure 1 confronts the randomly selected strategies of the other agents in the `CONFRONTATION` number of confrontations and then the worth of the xth strategy is calculated by the average of payoffs acquired in these confrontations. Since each agent has n number of strategies, the (`CONFRONTATION` $\times n \times 2$) number of confrontations is required to calculate the worth of all strategies of two agents. Here, we count one *iteration* when the worth of all strategies of two agents is calculated.

3.2 Modeling of parts of our focus

For our focus of modeling agents, we address the following two parts in modeling agents: (1) learning mechanisms and (2) knowledge representation capabilities.

3.2.1 Learning mechanisms

When implementing learning mechanisms of agents, we can consider several mechanisms. Among the many useful learning mechanisms, we employ the following: (1) evolutionary strategy (ES) (Back et al., 1992) and (2) learning classifier system (LCS) (Goldberg, 1989, Holland et al., 1986). The reasons for this employment are summarized as follows: (1) the ES mechanism performs well with a real number required to represent offer and threshold values in the bargaining game; and (2) the LCS architecture is implemented by modeling human beings (Holland et al., 1986) and several conventional research works employing LCS have already investigated social phenomena (*e.g.*, an artificial stock market (Arthur et al., 1997)). In detail, we employ the conventional $(\mu + \lambda)$ evolution strategies (ES) (Back et al., 1992) for ES and a Pittsburgh-style (Smith, 1983) classifier system instead of a Michigan-style (Holland, 1975) classifier system for LCS.

Under these learning mechanisms, a strategy, the worth of a strategy, and the strategies of an agent as shown in Figure 1 correspond to a gene, a fitness, and a population in evolutionary computation (EC) literature, respectively. Based on these learning mechanisms, EC-based agents acquire good strategies by varying the numerical values of offer and threshold as shown by the following ordinary procedure: (1) a fixed number (μ or `GENERATION_GAP`$\times n$) of the best strategies (*parents*) remains in the set from one iteration to the next; (2) a fixed number (λ or `GENERATION_GAP`$\times n$) of new strategies (*offspring*) is produced from the set of parents at each iteration by applying the mutation operation in $(\mu+\lambda)$-ES and the crossover, mutation, and inversion operations in the Pittsburgh-style LCS; (3) new strategies replace the same number of strategies with low worth values. The detailed implementation of both learning mechanisms is described below.

- **Evolutionary strategy:** Two bargaining agents are equipped with their own $(\mu + \lambda)$-ES; the framework is based on the works of (Gerding et al., 2000, Bragt et al., 2000). When producing offspring strategies, the mutation operation adds to or subtracts from the offer and threshold values. These added and subtracted values are calculated from a Gaussian distribution with standard deviation σ, which is kept by each strategy. After these offspring strategies are produced, the standard deviations of the offspring are set as the averages of those in the parents at each iteration, while the standard deviations of the parents are maintained. Note that if an offer or threshold value becomes inappropriate (*e.g.*, a minus value or a value more than 1), it is reset to 0 or 1, whichever is closer to the current value.

- **Learning classifier system:** Two bargaining agents are equipped with their own Pittsburgh-style LCS. To conduct computer simulations in the same framework of ES, we set or modify each LCS as follows: (1) the LCS in our simulations applies the crossover operation at each iteration to produce offspring strategies the every iteration. Mutation and inversion operations are probabilistically applied to the offspring strategies generated by the crossover operation; (2) although the pair of offer and threshold can be considered as one *if-then* rule from the viewpoint of LCS, the selected order of these rules (pairs) is determined in advance; and (3) the concept of *don't care* is not employed in our simulations. For the second point, in particular, a preliminary research found that LCS-based agents cannot learn good strategies if they are allowed to select the rules (the pair of offer and threshold) in the ordinary LCS way.

3.2.2 Knowledge representation capabilities

When implementing agents, we have to consider their knowledge representation capabilities. In the bargaining game, in particular, a representation of strategies of agents must be considered, though there are no standard guidelines. From this fact, we start by employing the following two types of knowledge representation capabilities:[5] (1) an ordinary explanation of numbers (*e.g.*, $0.01\cdots$) and (2) a limited explanation of numbers, which are restricted to a real number with two decimal digits (*e.g.*, 0.01) in this simulation. We focus on this knowledge representation because (1) social scientists may take the latter case for a concise representation (Indeed, we have met such situations); and (2) a real number in offer and threshold values is critical in the bargaining game.

4 Simulation

4.1 Simulation design

Computer simulations are conducted to compare the following two cases. Note that the first simulation is performed without any restriction of a real number represented in strategies of agents (*i.e.*, the simulation with an ordinary real number).

- **ES vs. LCS:** An investigation on the influence of different learning mechanisms of agents.

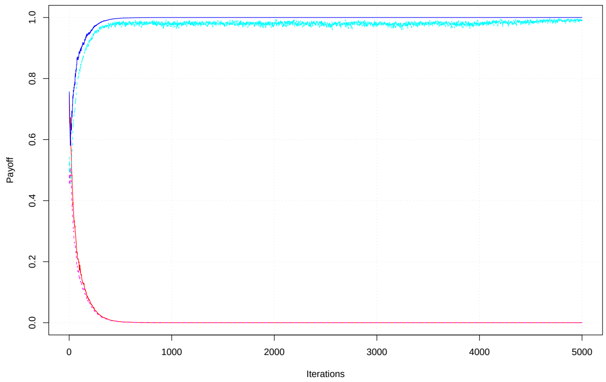
- **An ordinary real number vs. a real number of two decimal digits in ES:** An investigation on the influence of different knowledge representation capabilities of agents.

In each simulation, the following three cases are investigated. Note that all simulations are conduced until 5000 iteration and their results show average values over 10 runs.

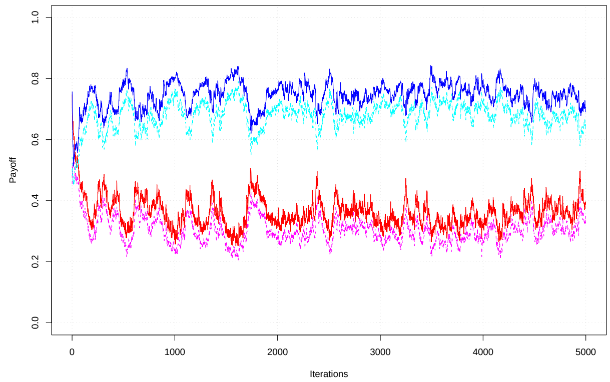
- **Case (a):** A payoff
- **Case (b):** An average negotiation process size
- **Case (c):** An accumulated number of each negotiation process size at the final (5000) iteration

As the parameter setting, the variables are set as follows. Note that preliminary examinations found that the tendency of the results does not drastically change according to the parameter setting.

[5] In addition to these issues, we should also investigate an influence of a modeling of strategies that are currently composed of the combination of offer and threshold as shown in Figure 1.

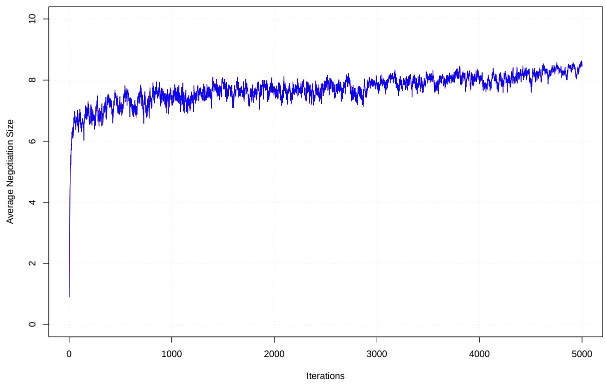


ES (a payoff)

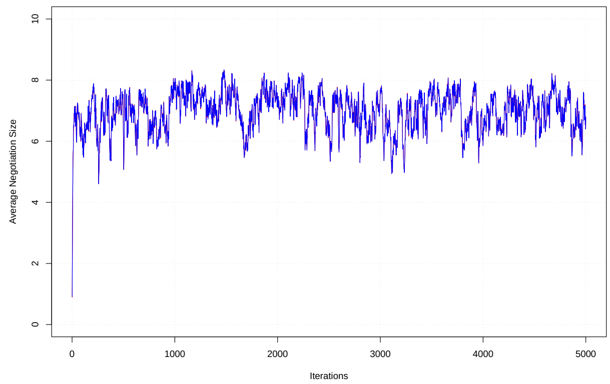


LCS (a payoff)

Case (a)

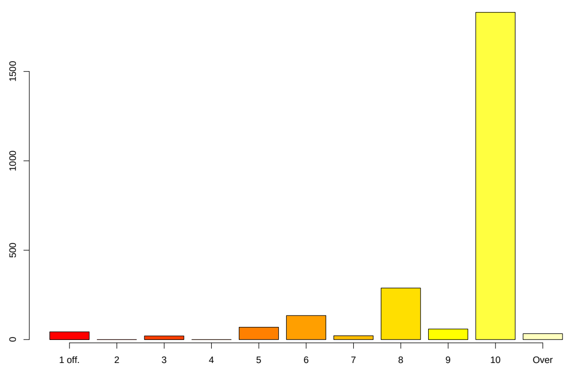


ES (average negotiation process size)

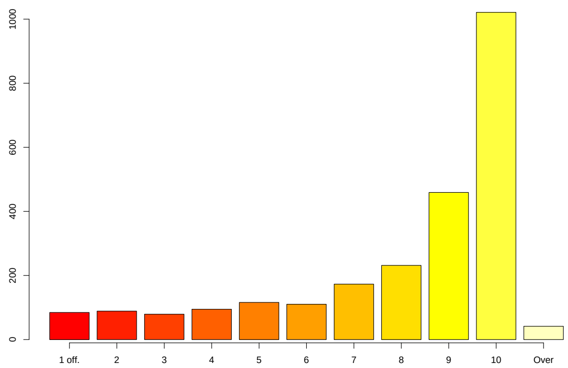


LCS (average negotiation process size)

Case (b)



ES (accumulated number of each negotiation process size at the 5000 iteration)



LCS (accumulated number of each negotiation process size at the 5000 iteration)

Case (c)

Figure 3: **Simulation results of ES vs. LCS: Average values over 10 runs at the 5000 iteration.**

- **Common parameters:** n (the number of strategies) is 50; `MAX_STEP` (the maximum number of steps in one confrontation) is 10; and `CONFRONTATION` (the number of confrontations for each strategy) is 20.
- **ES parameters:** μ (the parent population size) is 25; λ (the offspring population size) is 25; and σ (the initial standard deviation of a Gaussian distribution) is 0.5.
- **LCS parameters:** `GENERATION_GAP` (the percentage of replaced strategies) is 50%; `CROSSOVER_RATE` (the percentage of crossover operations) is 100%; `MUTATION_RATE` (the percentage of mutation operations) is 5%; and `INVERSION_RATE` (the percentage of inversion operations) is 5%.

4.2 Simulation results

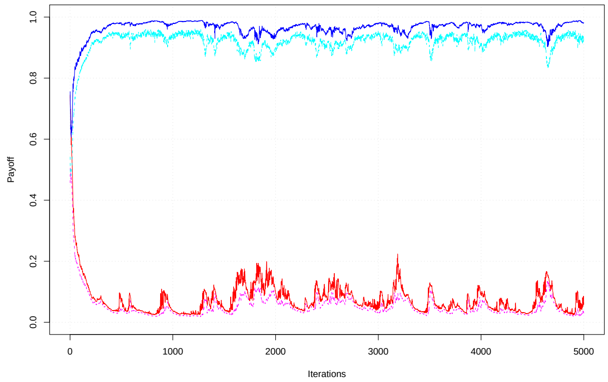
Figure 3 shows simulation results of both ES and LCS. In detail, figures (a), (b) and (c) indicate the results of the payoff, the average negotiation process size, and the accumulated number of each negotiation process size at the final (5000) iteration, respectively. The vertical axis in all of the figures indicates the indexes in the above three cases, while the horizontal axis in figures (a) and (b) indicates the iterations and the horizontal axis in figure (c) indicates the negotiation process size with negotiation failure represented as "Over" at the most right side. In particular, Figure 3(a) shows the payoff of agent A_1 in the lower lines and that of A_2 in the upper lines. The difference between the solid and light dash lines indicates that the former shows the best results and the latter shows the average results over 10 runs. Furthermore, Figure 4 shows simulation results of ES restricted to a real number with two decimal digits. Cases (a), (b) and (c) in Figure 4 has the same meaning of those in Figure 3.

From these results, we find that simulation results do not show the same tendency when different learning mechanisms or knowledge representation capabilities are applied to agents.

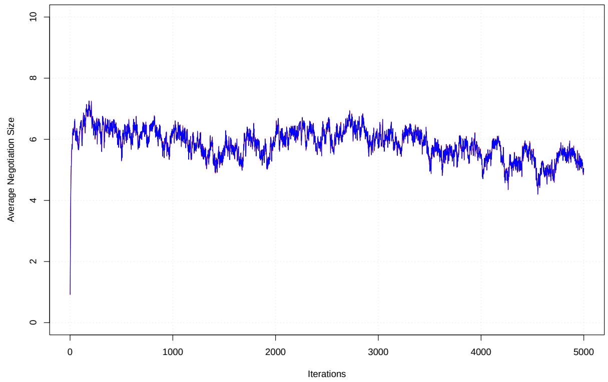
5 Discussion

5.1 Learning mechanisms

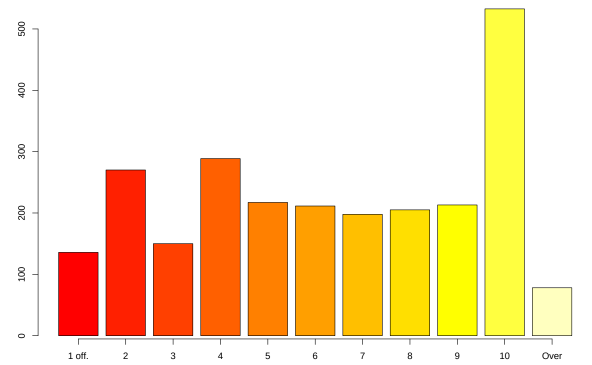
First, when focusing on the simulation results on different learning mechanisms of agents in Figure 3, the following implications are revealed: (1) the payoff of ES-based agents finally converges at the mostly maximum or minimum value (*i.e.*, 1 or 0), while that of LCS-based agents neither converges at a certain value



ES (a payoff)
Case (a)



ES (average negotiation process size)
Case (b)



ES (accumulated number of each negotiation process size at the 5000 iterations)
Case (c)

Figure 4: **Simulation results of ES with two decimal digits: Average values over 10 runs at the 5000 iteration.**

nor close to the maximum or minimum value; (2) the average negotiation size of ES-based agents increases, while that of LCS-based agents does not but simply oscillates; and (3) although the accumulated number of the 10th negotiation process size at the 5000 iteration is high both in ES-based and LCS-based agents, the accumulated number of another negotiation process size of ES-based agents is mostly close to 0, while that of LCS-based agents is not.

The reasons for the above results are summarized as follows: (1) the standard deviation of a Gaussian distribution in ES decreases as the iterations become large, while the crossover, mutation and inversion operations in LCS are constantly performed. Since most of these operations work as a divergent or explored factor, the decrease of such influence makes simulation results converge; (2) the offer and threshold values in all offspring are modified at every iteration in ES, while they are modified only by a mutation operation executed in a low probability in LCS. Furthermore, ES modifies such values like in a gradient search, while LCS modifies them randomly.

Here, we consider that game theory proves that rational agents A_1 and A_2 receive the maximum and minimum payoffs at the final negotiation process, respectively. This is because A_1 in our simulations has to accept any small offer proposed by A_2 at the 10th negotiation process; otherwise, A_1 cannot receive any payoff, *i.e.*, it receives 0 payoff. We therefore expect the following simulation results: (1) learning agents can acquire the maximum and minimum payoffs; (2) the average negotiation size increases if agents learn strategies appropriately; and (3) learning agents complete their negotiation process at the final offer of A_2 and the acceptance of A_1. In analyzing the simulation results according to the above three assumptions, we can consider that the ES-based agents show the same tendency in game theory but that LCS-based agents cannot. From this analysis, we can first conclude that simulation results are sensitive to the learning mechanisms applied to agents.

5.2 Knowledge representation capabilities

Next, when focusing on the simulation results on different knowledge representation capabilities of agents in Figure 3 (the normal case employing an ordinary real number) and Figure 4 (the restricted case employing a real number with two decimal digits), the following implications are revealed: (1) the payoff in the normal case finally converges at the mostly maximum or minimum value (*i.e.*, 1 or 0), while that in the restricted case does not completely converge; (2) the average negotiation size in the normal case increases, while that in the restricted case decreases; and (3) the accumulated number of each negotiation process size except the final (*i.e.*, the 10th) process in the normal case is mostly close to 0, while that in the restricted case is not.

To seek the reasons for the above different results, let's move our focus on to the 10th offer in Figure 2, where the values of offer and threshold are set here as 0.012 and 0.011, respectively. In this case, the agent who receives the offer from the opponent agent cannot accept it when employing an ordinary real number because the inequality of $O(0.012) \geq T(0.011)$ described in Section 3.1 is not satisfied. In contrast, the same agent accepts the offer when employing a real number with two decimal digits because the inequality of $O(0.01) \geq T(0.01)$ is satisfied. The same story can be told for other steps where both the offer and threshold values are close to each other. Due to this fact, agents have a possibility of accepting offers in each negotiation process. For this reason, agents with restricted knowledge representation capabilities cannot learn good strategies appropriately and thus they may accept unwilling (*i.e.*, small) offers in each negotiation process size. This increases the accumulated number of each negotiation process size except the final (the 10th) process in Figure 4(c) in comparison with that in Figure 3(c).

This finding indicates that simulation results can become strange when agents are restricted to handling only a real number with two decimal digits instead of an ordinary real number. Considering the fact that the previous section indicates that ES-based agents show the same tendency in game theory, we can secondary conclude that simulation results are sensitive to the knowledge representation capabilities of agents, even employing the same mechanism.

5.3 Essential factors for modeling agents

From the above analysis, both (1) learning mechanisms and (2) knowledge representation capabilities are important factors for cross-validating simulation results and computational models. This indicates that it is necessary to investigate such factors before investigating social phenomena arising from learning agent interaction.

6 Conclusion

This paper addressed cross-validation in multiagent-based simulation by analyzing evolutionary agents in a bargaining game. In particular, this paper focused on analyzing different learning mechanisms and knowledge representation capabilities applied to agents for cross-validation. To investigate the importance of the above issues, we compared the following two cases: (1) agents employing an evolutionary strategy (ES) and agents employing a learning classifier system (LCS) as different learning mechanisms; and (2) agents handling an ordinary explanation of numbers and agents handling a limited explanation of numbers as different knowledge representation capabilities. Through an intensive comparison of the above simulation results, we found that both learning mechanisms and knowledge representation capabilities are important factors for cross-validating simulation results and computational models.

However, the results obtained in this paper do not cover all factors for cross-validation, and thus further careful qualifications and justifications such as experiments in other domains are needed to generalize our results. Such important directions must be pursued in the near future, but the following implications are potentially suggested from the current results: (1) simulation results by ES-based agents show the same tendency in game theory but those by LCS-based agents do not; and (2) even simulation results by ES-based agents become strange when the agents are restricted to handling only a real number with two decimal digits instead of an ordinary real number in a negotiation process between agents in the bargaining game.

Future research will include the following: (1) many simulations in other domains to generalize our results; (2) a comparison with other learning mechanisms such as reinforcement learning or other knowledge representations such as discrete numbers; (3) a validation of results and models with more than two agents; and (4) an investigation on the influence of the discount factor in ES- and LCS-based agents.

Acknowledgements

The research reported here was supported in part by a contract with the Telecommunications Advancement Organization of Japan entitled, "Research on Human Communication."

References

Arthur, W. B. et al. (1997). "Asset Pricing Under Endogenous Expectations in an Artificial Stock Market," in W. B. Arthur et al. (Eds.), *The Economy as an Evolving Complex System II*, Addison-Wesley, pp. 15–44.

Axelrod, R. M. (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*, Princeton University Press.

Axtell, R. et al. (1996). "Aligning Simulation Models: A Case Study and Results," *Computational and Mathematical Organization Theory (CMOT)*, Vol. 1, No. 1, pp. 123–141.

Bäck, T. et al. (1992). "Evolutionary Programming and Evolution Strategies: Similarities and Differences," *The 2nd Annual Evolutionary Programming Conference*, pp. 11–22.

van Bragt, D. D. B. et al. (2000). "Equilibrium Selection in Alternating-Offers Bargaining Models: The Evolutionary Computing Approach," *The 6th International Conference of the Society for Computational Economics on Computing in Economics and Finance (CEF'00)*.

Burton, R. M. and Obel, B. (1995). "The Validity of Computational Modes in Organization Science: From Model Realism to Purpose of the Model," *Computational and Mathematical Organization Theory (CMOT)*, Vol. 1, No. 1, pp. 57–71.

Carley, K. M. and Gasser, L. (1999). "Computational and Organization Theory," in Weiss, G. (Ed.), *Multiagent Systems – Modern Approach to Distributed Artificial Intelligence –*, The MIT Press, pp. 299–330.

Gerding, E. H. et al. (2000). "Multi-Issue Negotiation Processes by Evolutionary Simulation: Validation and Social Extensions," *CWI, Centre for Mathematics and Computer Science*, No. SEN R0024.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.

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

Holland, J. H. et al. (1986). *Induction*, The MIT Press.

Moss, S. and Davidsson, P. (2001). *Multi-Agent-Based Simulation*, Lecture Notes in Artificial Intelligence, Vol. 1979, Springer-Verlag.

Moss, S (2001). "Editorial Introduction: Messy Systems – The Target for Multi Agent Based Simulation" in (Moss and Davidsson, 2001), pp. 1–14.

Muthoo, A. (2000). "A Non-Technical Introduction to Bargaining Theory," *World Economics*, pp. 145–166.

Oliver, J. R. (1996). "On Artificial Agents for Negotiation in Electronic Commerce," Ph.D. Thesis, University of Pennsylvania.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*, MIT Press.

Rubinstein, A. (1982). "Perfect Equilibrium in a Bargaining Model", *Econometrica*, Vol. 50, No. 1, pp. 97–109.

Smith, S. F. (1983). "Flexible Learning of Problem Solving Heuristics through Adaptive Search," *The 8th International Joint Conference on Artificial Intelligence (IJCAI '83)*, pp. 422–425.

Ståhl, I. (1972). *Bargaining Theory*, Economics Research Institute at the Stockholm School of Economics.