Evolving Blackjack Strategies Using Cultural Learning in Multi–Agent Systems

Dara Curran and Colm O'Riordan

Dept. of Information Technology
National University of Ireland, Galway.

Abstract. This paper examines a new approach to the evolution of blackjack strategies, that of cultural learning. Many traditional machine learning approaches have concentrated on reinforcement learning approaches and report satisfactory results. Populations of neural network agents evolve using genetic algorithms (population learning) and at each generation the best performing agents are selected as teachers. Cultural learning is implemented through a hidden layer in each teacher's neural network that is used to produce utterances which are imitated by its pupils during many games of blackjack. Results show that the cultural learning approach outperforms previous work as well as the best known non–card counting human approaches.

1 Introduction

The game of blackjack has been the subject of much research, particularly in the reinforcement learning domain. This paper introduces a cultural learning approach for the evolution of high quality blackjack playing agents. Using a combination of genetic algorithms and neural networks, we evolve a population of neural network agents which play games of blackjack against an automated dealer. Cultural learning is introduced by taking a percentage of the population and allowing it to teach the following generation through specialised verbal output nodes. Three experiments are performed, each one increasing the information available to each agent. We compare the evolved strategies with bench–marks obtained from a blackjack simulator.

The remainder of this paper is arranged as follows: Section 2 discusses related work, including a summary of the learning models employed in this set of experiments and evolutionary computation approaches to the game of blackjack. Section 3 presents the results of bench–marking several popular blackjack strategies, Section 4 introduces the artificial life simulator employed in the experiments, Section 5 presents the experimental results and Section 6 concludes and suggests future work.

2 Related Work

2.1 Learning Models

Popular learning models for evolving populations of neural network based agents can be roughly classified into two distinct groups: population and life-time learning.

Population Learning Population learning refers to the process whereby a population of organisms evolves, or learns, by genetic means through a Darwinian process of iterated selection and reproduction of fit individuals. Learning takes place at a purely genetic level and the agent itself does not contribute to its survival through any independent learning or adaptation process during its lifetime. Population learning is typically simulated through the use of genetic algorithms[1, 2].

Life–ime Learning Agents that are capable to adapt to environmental changes and novel situations during their life-time can be said to be employing lifetime learning. The population as a whole continues to evolve by population learning and lifetime learning further enhances the population's fitness through its adaptability and resistance to change[3–9].

A phenomenon related to life–time learning, first reported by Baldwin[10] and simulated by Hinton and Nowlan[7], occurs when certain behaviour first evolved through life–time learning becomes imprinted onto an individual's genetic material through the evolutionary processes of crossover and mutation. This individual is born with an innate knowledge of such behaviour and, unlike the rest of the populations, does not require time to acquire it through life–time learning. As a result, the individual's fitness will generally be higher than that of the population and the genetic mutation should become more widespread as the individual is repeatedly selected for reproduction.

Cultural Learning Culture can be succinctly described as a process of information transfer within a population that occurs without the use of genetic material. Culture can take many forms such as language, signals or artifactual materials. Such information exchange occurs during the lifetime of individuals in a population and can greatly enhance the behaviour of such species. Because these exchanges occur during an individual's lifetime, cultural learning can be considered a subset of lifetime learning.

Artificial cultural evolution, or synthetic ethology[11], has been extensively researched and a number of approaches were considered for its implementation for this set of experiments. These included fixed lexicons[12, 13], indexed memory[14], cultural artifacts[15, 16] and signal–situation tables[17]. The approach chosen was the increasingly popular teacher/pupil scenario[18, 19, 13, 20] where a number of highly fit agents are selected from the population to act as teachers for the next generation of agents, labelled pupils. Pupils learn from teachers by

observing the teacher's verbal output and attempting to mimic it using their own verbal apparatus. As a result of these interactions, a lexicon of symbols evolves to describe situations within the population's environment.

2.2 The Game of BlackJack

Blackjack or twenty–one begins with the dealer dealing two cards face–up to each player and two to his/herself, with one card visible (the *up–card*) and the other face down. Cards are valued by their face value (10 for all picture cards) except for the ace which can be counted either as 11 or 1. The object of the game is to obtain a higher score (the sum of all card values) than that of the dealer's without exceeding 21. Each player can *draw* additional cards until they either *stand* or exceed 21 and go *bust*. Once all players have obtained their cards, the dealer turns over the hidden card and draws or stands as appropriate. Should the dealer's hand bust, all players win.

The dealer is at considerable advantage because he/she only enters the game once all players have fully completed their play. Thus, it is probable that some players will have bust even before the dealer reveals the hidden card. In addition, the fact that only one of the dealer's cards is visible means that players must make judgements based on incomplete information. As a rule, the dealer follows a fixed strategy, typically standing on a score of 17 or more and drawing otherwise.

All aspects related to betting such as doubling down and splitting have been removed from this implementation in order to facilitate comparison with previous work which employs a similar approach.

In a casino setting, between 3 and 6 six full decks of cards are shuffled at the start of the first hand and the game is played until the cards run out. Up to six players and one dealer may play at a blackjack table. Again for simplicity, this implementation considers only a single player playing against the dealer using a single deck of cards which is shuffled at the start of each hand.

A number of very successful card–counting strategies inspired by Thorp[21] have been developed but are not considered in this set of experiments.

Several attempts have been made to develop high performing blackjack strategies with populations of neural networks using reinforcement learning techniques[22, 23]. The nature of the game means that there is no perfect set of neural network outputs from which to perform back–propagation. It is for this reason that we wish to show that the introduction of cultural learning can generate superior strategies than reinforcement learning methods and provides the learning framework required without knowledge of the perfect strategy.

3 Bench–marking

In order to assess the performance of any evolved strategy, a set of bench–marks must be obtained for comparison purposes. While there have been many attempts to calculate the performance of blackjack strategies using simulation and probabilistic techniques[24, 25, 21], the values produced tend to vary by a

rather large margin. For instance, the success of a player employing the standard dealer strategy is reported at between 39% and 44% wins. As a result of these discrepancies, it was felt that it may be more meaningful to calculate the values for various strategies using our own simulation. These values will be more readily comparable to the performance of evolved strategies, since a large proportion of the blackjack simulator will also be used by the evolving populations to play games.

The blackjack simulator consists of a dealer, who employs the traditional dealer strategy of standing on 17 or greater, and a single player whose strategy can be set at the beginning of the simulation. As in previous work, both dealer and player hand values are calculated by adding card values where each ace is counted as 11 unless it would cause a bust.

Several strategies were considered:

- Dealer's (Stand on 17 or more, Draw on less)
- Random
- Always stand
- Hoyle's
- Uribe Evolved Strategy

The Hoyle strategy[26] is based on the dealer's up card and the possession of an ace. It can be summarised as:

```
if (dealer card < 6)
    if (ace is held)
        stand on 15
    else
        stand on 13
else
    stand on 17
```

The Uribe Evolved strategy is taken from the work of Uribe and Sanchez[23] and can be summarised as:

```
if (score> 9) or [(score> 13)and(score< 19) and(an ACE is held)]
    stand
else
    hit with 50% of probability
```

In order to produce statistically meaningful results, we performed 1000 runs of 1000 games for each strategy. The results presented in the table below are average wins for each strategy.

Strategy	Percentage Wins	Standard Deviation
Hoyle	43.70	1.587
Dealer	41.55	1.576
Uribe et al	38.76	1.505
Always Stand	37.91	1.531
Random	30.41	1.511

We can see from these results that most strategies are quite poor against the dealer and that Hoyle's strategy performs best. This is most likely due to its inclusion of ace and dealer up–card information.

4 Simulator

Each agent in the population contains a neural network that allows it to play blackjack. Once cards are dealt to the agents the value of the hand is shown to the network using thermometer encoding[22]. In this first set of experiments, there are 18 input nodes, representing a scale of hand values of 4-21 (Figure 1). An initial set of trial experiments using a single input node with 18 activation values was shown to perform poorly, motivating the decision to employ 18 input nodes.

The agent's decision is determined by rounding the output of the single output node, where draw and stand are represented by 0 and 1 respectively. The number of hidden layers and nodes therein is unrestricted and is determined by the evolutionary process. The agents are evolved using a previously developed artificial life simulator[8, 9], outlined below.

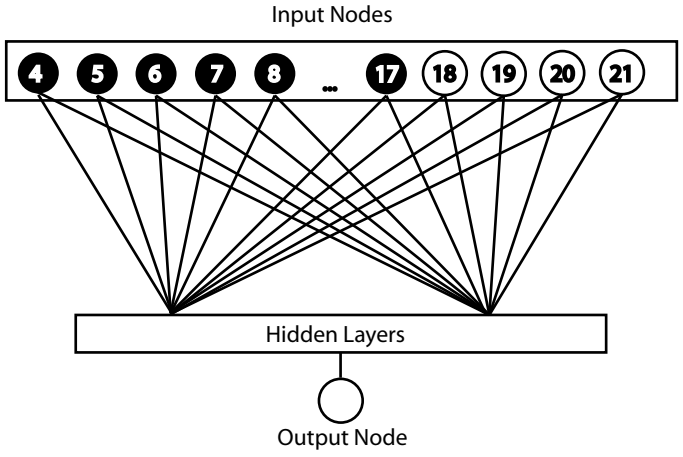

Fig. 1. *Thermometer Encoding*

The architecture of the artificial life simulator can be seen as a hierarchical structure. At the top-level of the simulator is a command interpreter which allows users to define an experiment's variables including the number of networks, the number of generations to run the experiment, mutation and crossover rates and the actual problem set which the population will be attempting to solve.

The neural network layer takes the variables set using the command interpreter and initialises a given number of neural networks. The layer then performs training and testing of the networks according to the parameters of the experiment. These network memory structures are then passed to the encoding layer which transforms them into genetic code structures for use in the genetic algorithm. The encoding mechanism used for this set of experiments is a modified version of marker based encoding.

The genetic algorithm layer uses the genetic codes and the data retrieved from the neural network layer's testing of the networks to perform its genetic operators on the population. A new population is produced in the form of genetic codes. These are passed to the decoding layer which transforms each code into a new neural network structure. These structures are then passed up to the neural network layer for a new experiment iteration. Once the required number of generations has been reached, the experiment finishes.

Two-point crossover is employed and weight mutation is employed which takes the weight value and increases/decreases the value according to a random percentage (200%). This approach was found, empirically, to be more successful and was adopted for this set of experiments.

4.1 Encoding Scheme

An encoding scheme is necessary to map each agent's neural network structure to a genetic code. Many schemes were considered in preparation of these experiments, prioritising flexibility, scalability, difficulty and efficiency. The scheme chosen is based on Marker Based Encoding which allows any number of nodes and interconnecting links for each network giving a large number of possible neural network permutations.

Marker based encoding represents neural network elements (nodes and links) in a binary string. Each element is separated by a marker to allow the decoding mechanism to distinguish between the different types of element and therefore deduce interconnections[27, 28]. A gene code produced using this scheme is treated as a circular entity. Thus, the code parsing mechanism reading the end of the gene code will begin reading the start of the gene code once the end is reached until all available information is correctly retrieved.

4.2 Simulating Cultural Evolution

In order to perform experiments related to cultural evolution, it was necessary to adapt the existing simulator architecture to allow agents to communicate with one another. This was implemented using an extended version of the approach

adopted by Hutchins and Hazlehurst. The last hidden layer of each agent's neural network functions as a verbal input/output layer (figure 2).

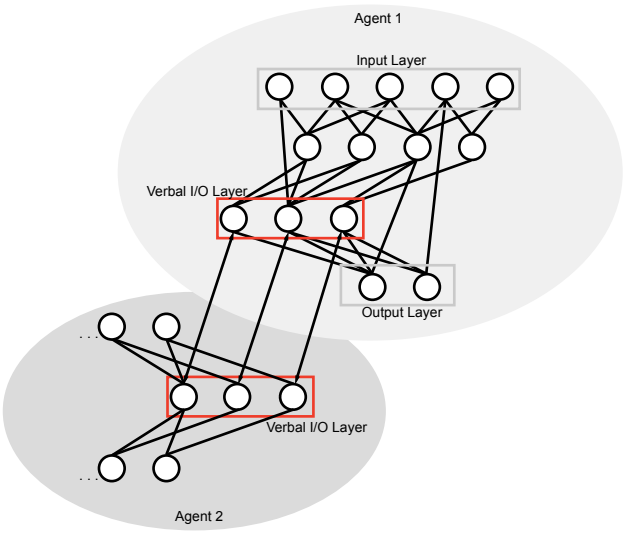

Fig. 2. *Agent Communication Architecture*

At end of each generation, a percentage of the population's fittest networks are selected and are allowed to become teachers for the next generation. The teaching process takes place as follows: a teacher is stochastically assigned n pupils from the population where $n = \frac{N_{pop}}{N_{teachers}}$, where N_{pop} is the population size and $N_{teachers}$ is the number of teachers. Each pupil follows the teacher in its environment and observes the teacher's verbal output as it interacts with its environment. A teaching cycle occurs when the pupil attempts to emulate its teacher's verbal output using back-propagation. Once the number of required teaching cycles is completed, the teacher networks die and new teachers are selected from the new generation.

Unlike previous implementations, the number of verbal input/output nodes is not fixed and is allowed to evolve with the population, making the system more adaptable to potential changes in environment. In addition, this method does not make any assumptions as to the number of verbal nodes (and thus the complexity of the emerging lexicon) that is required to effectively communicate.

5 Experiments

Each experiment allows 100 agents to evolve over 500 generations. At each generation, agents play 100 games against a dealer strategy and an agent's fitness is determined by the percentage of wins obtained scaled to [0.0,1.0]. Agents

are linearly ranked and selected for reproduction using roulette wheel selection. Crossover and mutation are applied with probabilities 0.6 and 0.02 respectively. When cultural learning is applied, 10% of each generation are selected to act as teachers for the next. Teaching cycles are set at 2 and a noise value in the range [-0.5,0.5] is added to the each interaction with probability 0.01. Each of these settings were empirically determined to be suitable.

5.1 Basic Experiment

The results for these experiments are presented in Figure 3. In addition to the average fitness of the population, both Dealer and Hoyle strategy levels are shown for comparison purposes.

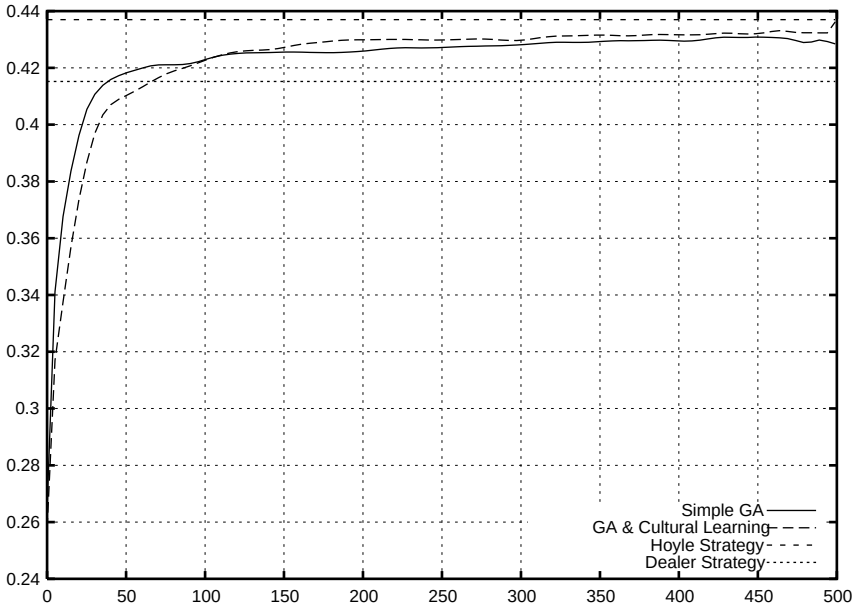

Fig. 3. *Experiment 1*

Both populations begin at fitness (define) levels below 0.3 (worse than even a random strategy) but quickly improve until generation 200, where both stabilise. The introduction of cultural learning provokes a slight improvement in fitness over the simple genetic algorithm but is not able to bring the population much above the standard Dealer strategy.

The strategies employed by the population were extracted by feeding all possible card values to each network in the population at generation 500 and noting the resulting agent decisions. On examination of these strategies, we discovered that the learning population is infact using the dealer strategy of standing on

17 or more while the simple genetic algorithm population is using a similar but less effective strategy of standing on 16 or more.

Given that the agents receive such little information about the game, it is unsurprising that they should only be able to discover the Dealer strategy since this is the best possible strategy given only the current hand value as a guide.

5.2 Introducing the Ace

Of all cards in the game of blackjack, the Ace is the most versatile since it can be used both as an 11 or as a 1, improving the chance of a better hand if used wisely. Intuitively, better strategies should evolve as a result of ace information being introduced.

This set of experiments is identical to the first except that each agent's neural network is shown information regarding its possession of an ace. To implement this, an additional input unit was created bringing the total number of input nodes for each network to 19. The new ace node is set to 1 if the agent's hand has an ace, and to 0 otherwise. All other variables remain equal to the first experiment.

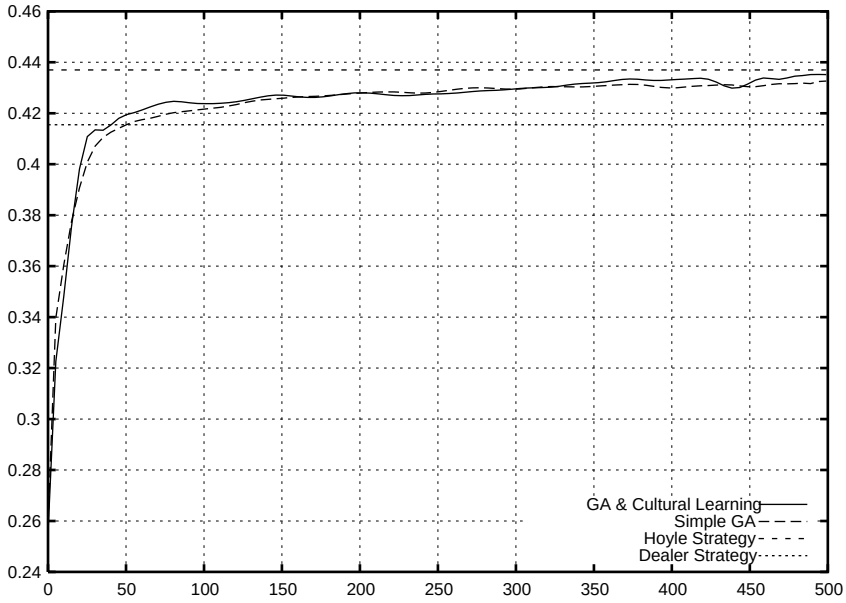

Fig. 4. *Experiment 2*

The results illustrated in Figure 4 show that the introduction of ace information gives rise to a very slight improvement in overall fitness. Once again, the population employing cultural learning performs modestly better than that

using the simple genetic algorithm (average of 0.427 or 42.7% wins for the GA and 0.431 or 43.1% wins for cultural learning) and overall, results have improved since the introduction of ace information.

The evolved strategy of the learning population consists of standing on 14 or greater if no ace is present and standing on 18 otherwise. The strategy employs the ace information to determine when and when not to risk drawing another card. Since an Ace can be valued at both 11 or 1, a hand with an ace and value 17 can also be valued at 7 posing little risk if another card is drawn. On the other hand, if no ace is held, it is prudent to stand on a lower value for the risk of busting is greater.

5.3 Dealer Information

The addition of ace information provides a vary slight improvement on the previous experiment and does not yet achieve the level of the Hoyle strategy. Most advanced strategies, including Hoyle's, take the dealer's up–card into account when determining a course of action. Therefore, we introduce the dealer's card to the agent population, by adding an additional 10 input nodes, one for each of the dealer's possible up–cards (2-9, 10 for all picture cards and the ten, Ace). All other variables remain equal to the last experiment.

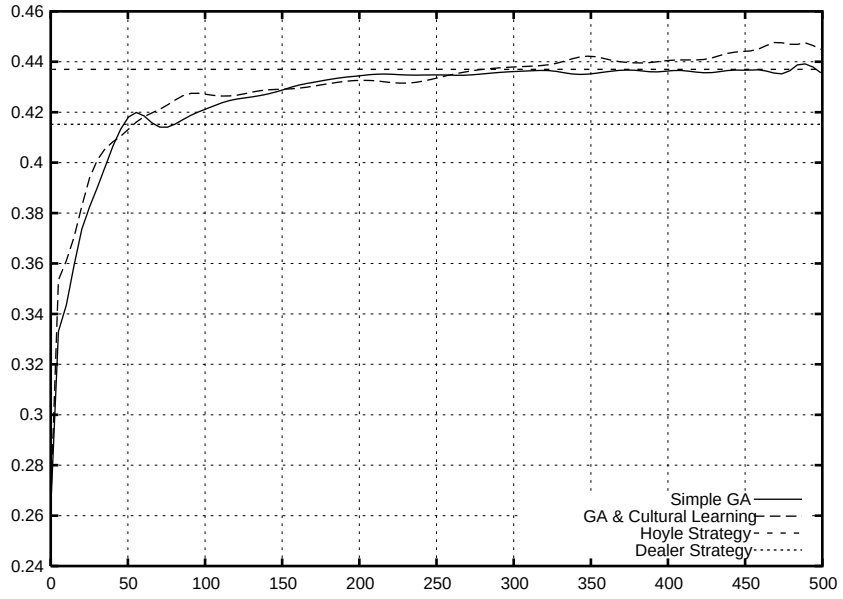

Fig. 5. *Experiment 3*

The graph in figure 5 shows that the addition of dealer information dramatically improves the performance of both populations but particularly that of the

population employing cultural learning which achieved highs of nearly 0.46 (46% wins) versus 0.44 for population learning. The resulting strategy is significantly more complex than in the last experiments:

```
if (an Ace is held)
{
    if (dealer has a 6 or higher)
        stand on 16
    else
        stand on 17
}
else
{
    if (dealer has a 7 or higher)
        stand on 17
    else
        stand on 13
}
```

It is clear from the strategy that the evolved agents are employing the new dealer and ace information to the full extent and have identified a threshold value for the dealer up–card. The strategy is tested in the next section to ascertain its performance with respect to the bench–marked strategies.

5.4 Strategy Testing

As a result of the inherent random nature of blackjack, it is necessary to test the strategies over a number of runs to observe whether they are successful. The final evolved strategy was taken and hard–coded into the blackjack simulator and 1000 runs of 100 games were played. The averaged results are displayed in the table below:

Strategy	**Percentage Wins**	**Standard Deviation**
Hoyle	43.70	1.587
Evolved Strategy	**43.67**	**1.582**
Dealer	41.52	1.576
Sanchez et al	38.43	1.505
Always Stand	38.00	1.531
Random	30.67	1.511

The results of the simulation show that the evolved strategy does not quite reach the level of Hoyle's strategy but is very close. On examination of the standard deviations, it is clear that the top two strategies are relatively similar, suggesting that the population has evolved an optimum strategy given the information available. It is likely that in order to out–perform Hoyle's strategy it is necessary to keep track of cards that have been played during a game, something which would only become truly useful if the number of players was increased.

6 Conclusion

We have shown that the addition of cultural learning to a population of neural network agents evolving using a genetic algorithm can produce robust blackjack strategies that out–perform those evolved thus far using reinforcement learning. As in previous work, the addition of dealer information to the population significantly improves performance. Through the bench–marking process we have shown that the evolved strategy is practically equivalent to the best human strategy which does not incorporate card–counting. Future work will introduce more players per game in the with the expectation of evolving agents capable of card counting strategies such as those developed by human experts.

References

1. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, Addison-Wesley, 1989.
2. J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor MI: The University of Michigan Press, 1975.
3. G. Mayley. Guiding or hiding: Explorations into the effects of learning on the rate of evolution. In *Proceedings of the Fourth European Conference on Artificial Life*. MIT Press, 1997.
4. James Watson and Janet Wiles. The rise and fall of learning: A neural network model of the genetic assimilation of acquired traits. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, pages 600–605, 2002.
5. F. B. Pereira and E. Costa. How learning improves the performance of evolutionary agents: A case study with an information retrieval system for a distributed environment. In *Proceedings of the International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models (ISAS 2001)*, pages 19–23, 2001.
6. D. Parisi S. Nolfi, J. L. Elman. Learning and evolution in neural networks. In *Adaptive Behavior*, volume 3, pages 5–28, 1994.
7. G. E. Hinton and S. J. Nowlan. How learning guides evolution. In *Complex Systems*, volume I, pages 495–502, 1987.
8. D. Curran and C. O'Riordan. On the design of an artificial life simulator. In V. Palade, R. J. Howlett, and L. C. Jain, editors, *Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2003)*, University of Oxford, United Kingdom, 2003.
9. D. Curran and C. O'Riordan. Artificial life simulation using marker based encoding. In *Proceedings of the 2003 International Conference on Artificial Intelligence (IC-AI 2003)*, volume II, pages 665–668, Las Vegas, Nevada, USA, 2003.
10. J.M. Baldwin. A new factor in evolution. In *American Naturalist 30*, pages 441–451, 1896.
11. L. Steels. The synthetic modeling of language origins. In *Evolution of Communication*, pages 1–34, 1997.
12. H. Yanco and L. Stein. An adaptive communication protocol for cooperating mobile robots, 1993.
13. A. Cangelosi and D. Parisi. The emergence of a language in an evolving population of neural networks. *Technical Report NSAL–96004, National Research Council, Rome*, 1996.

14. L. Spector and S. Luke. Culture enhances the evolvability of cognition. In *Cognitive Science (CogSci) 1996 Conference Proceedings*, 1996.
15. E. Hutchins and B. Hazlehurst. Learning in the cultural process. In *Artificial Life II, ed. C. Langton et al.* MIT Press, 1991.
16. A. Cangelosi. Evolution of communication using combination of grounded symbols in populations of neural networks. In *Proceedings of IJCNN99 International Joint Conference on Neural Networks (vol. 6)*, pages 4365–4368, Washington, DC, 1999. IEEE Press.
17. B. MacLennan and G. Burghardt. Synthetic ethology and the evolution of cooperative communication. In *Adaptive Behavior 2(2)*, pages 161–188, 1993.
18. A. Billard and G. Hayes. Learning to communicate through imitation in autonomous robots. In *7th International Conference on Artificial Neural Networks*, pages 763–738, 1997.
19. D. Denaro and D. Parisi. Cultural evolution in a population of neural networks. In *M.Marinaro and R.Tagliaferri (eds), Neural Nets Wirn-96.New York: Springer*, pages 100–111, 1996.
20. E. Hutchins and B. Hazlehurst. How to invent a lexicon: The development of shared symbols in interaction. In N. Gilbert and R. Conte, editors, *Artificial Societies: The Computer Simulation of Social Life*, pages 157–189. UCL Press: London, 1995.
21. E. O. Thorp. *Beat the Dealer.* Random House, 1966.
22. D. K. Olson. *Learning to Play Games from Experience: An Application of Artificial Neural Networks and Temporal Difference Learning.* Pacific Lutheran University, 1993.
23. Andrs Prez-Uribe and Eduardo Sanchez. Blackjack as a test bed for learning strategies in neural networks. In *International Joint Conference on Neural Networks, IJCNN'98*, 1998.
24. E. O. Thorp. *The Mathematics of Gambling.* Lyle Stuart, 1984.
25. H. Maisel R. R. Baldwin, W. E. Cantey and J. P. McDermott. The optimum strategy in blackjack. In *Journal of the American Statistical Association*, 1956.
26. A. H. Morehead and G. M. Smith. *Hoyle's Rules of Games.* Plume, 1963.
27. H. Kitano. Designing neural networks using genetic algorithm with graph generation system. In *Complex Systems, 4, 461-476*, 1990.
28. G. F. Miller, P. M. Todd, and S. U. Hedge. Designing neural networks using genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pages 379–384, 1989.