# No-Limit Texas Hold'em Poker Agents Created with Evolutionary Neural Networks

Garrett Nicolai and Robert J. Hilderman

***Abstract*— In order for computer Poker agents to play the game well, they must analyse their current quality despite imperfect information, predict the likelihood of future game states dependent upon random outcomes, model opponents who are deliberately trying to mislead them, and manage finances to improve their current condition. This leads to a game space that is large compared to other classic games such as Chess and Backgammon. Evolutionary methods have been shown to find relatively good results in large state spaces, and neural networks have been shown to be able to find solutions to non-linear search problems such as Poker. In this paper, we develop No-Limit Texas Hold'em Poker agents using a hybrid method known as evolving neural networks. We also investigate the appropriateness of evolving these agents using evolutionary heuristics such as co-evolution and halls of fame. Our agents were experimentally evaluated against several benchmark agents as well as agents previously developed in other work. Experimental results show the overall best performance was obtained by an agent evolved from a single population (i.e., no co-evolution) using a large hall of fame. These results demonstrate an effective use of evolving neural networks to create competitive No-Limit Texas Hold'em Poker agents.**

## I. Introduction

In the field of AI, games have presented a challenging area for research [4], [16], [22]. Games have a well-defined set of rules, and also have a distinguishable goal state. The rules of any particular game remain constant, and can be learned by an artificial agent. Games also present conditions amenable to agent evaluation, as agents developed to play a particular game can be placed in direct competition with other agents of the same game to objectively determine their worth.

Numerous successes have been achieved in the field of computer-developed game-playing agents. Chess agents have been developed that are capable of playing at the grandmaster level (e.g., DEEP BLUE [7] and HYDRA [8]). Champion Checkers agents have also been developed [18], [19]. Agents for Backgammon, a stochastic game, have been developed that are capable of playing above the level of grandmasters [22]. The use of evolutionary algorithms has contributed to this success, and they have been used to develop agents for Backgammon [16], [22], Go [12], Checkers [18], [19], Chess [9], [11], [23], and Poker [1], [2].

The game of Poker has recently attracted attention as a challenging area of games research [1], [2], [4], [10]. The game is difficult for a computer agent to play well, as it has a very large state space, some information is hidden from the players, there is an element of chance due to the order in which the cards are dealt, and opponents often provide false information [4]. The most often studied variant is Limit Texas Hold'em [1], [4], [10]. Another variant, called No-Limit Texas Hold'em [2], [6], changes only one rule, namely the amount that can be bet in each betting round, but this one change leads to several differences in how the game is played.

In this paper, we create No-Limit Texas Hold'em Poker playing agents using evolving neural networks. We evaluate the effectiveness of using evolution to promote learning of successful No-Limit Texas Hold'em Poker strategies. Betting strategies that are capable of winning large-scale tournaments are promoted, while unsuccessful strategies are replaced by slight variations of the successful strategies. We further test the effectiveness of evolutionary methods applied to No-Limit Texas Hold'em Poker strategies by implementing the evolutionary heuristics of co-evolution, and halls of fame, as suggested in [12], [16], [17]. For a comprehensive discussion of our work on No-Limit Texas Hold'em, see [14].

## II. RELATED WORK

Research into computer Poker has progressed slowly in comparison with other games, so Poker does not have as large an established literature.

### A. Limit Texas Hold'em Poker

The Computer Poker Research Group at the University of Alberta is the largest contributor to Poker research in AI. The group recently created one of the best Poker-playing agents in the world, winning the 2007 Poker Bot World Series [10].

Beginning with Loki [5], and progressing through Poki [4] and PsOpti [3], the University of Alberta has concentrated on creating Limit Texas Hold'em Poker players. Originally based on opponent hand prediction through limited simulation, each generation of Poker agents from the UACPRG has modified the implementation and improved upon the playing style of the predecessors. The current agents [10], [20] are mostly game theoretic players that try to minimize loss while playing, and have concentrated on better observation of opponents and the implementation of counter-strategies. The current best agents are capable of defeating weak to intermediate human players, and can occasionally defeat world-class human players.

### B. No-Limit Texas Hold'em Poker

No-Limit Texas Hold'em Poker was first studied in [6], where a rule-based system was used to model players. The

Both authors are with the Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada. E-mail: nicolaig@uregina.ca, Robert.Hilderman@uregina.ca

earliest agents were capable of playing a very simple version of two-player No-Limit Texas Hold'em Poker, and were able to defeat several benchmark agents. After modifying the rules used to make betting decisions, the agents were again evaluated, and were shown to have maintained their level of play, while increasing their ability to recognize and adapt to opponent strategies.

No-Limit Texas Hold'em Poker agents were developed in [2], and were capable of playing large-scale games with up to ten players at a table, and tournaments with hundreds of tables. Evolutionary methods were used to evolve two-dimensional matrices corresponding to the current game state. These matrices represent a mapping of hand strength and cost. When an agent makes a decision, these two features are analysed, and the matrices are consulted to determine the betting decision that should be made. The system begins with some expert knowledge (what we called a head-start approach). Agents were evolved that play well against benchmark agents, and it was shown that agents created using both the evolutionary method and the expert knowledge are more skilled than agents created with either evolutionary methods or expert knowledge.

### C. Games and Evolutionary Neural Networks

Applying evolutionary algorithms to games is not without precedent. As early as the 1950's, the concept of self-play (i.e., the process of playing agents against themselves and modifying them repeatedly) was being applied to the game of Checkers [18]. In [22] evolutionary algorithms were applied to the game of Backgammon, eventually evolving agents capable of defeating the best human players in the world. In [19], an algorithm similar to that described in [22] was used in conjunction with self-play to create an agent capable of playing small-board Go.

Evolutionary methods have also been applied to Poker. In [1], agents are evolved that can play a shortened version of Limit Texas Hold'em Poker, having only one betting round. Betting decisions are made by providing features of the game to a formula. The formula itself is evolved, adding and removing parameters as necessary, as well as changing weights of the parameters within the formula. Evolution is found to improve the skill level of the agents, allowing them to play better than agents developed through other means.

In [23], temporal difference learning is applied to Chess in the NeuroChess program. The agent learns to play the middle game, but plays a rather weak opening and endgame. In [11], a simplified evaluation function is used to compare the states of the board whenever a decision must be made. Evolution changes the weights of various features of the game as they apply to the decision formula. The evolutionary method accorded values to each of the Chess pieces, similar to a traditional point system used in Chess. The final agent was evaluated against a commercially available Chess program and unofficially achieved near expert status and an increase in rating of almost 200% over the unevolved agent. In [9], the endgame of Chess was the focus, and the opening and midgame were ignored. For the endgame situations, the agents started out poorly, but within several hundred generations, were capable of playing a grandmaster level engine nearly to a draw.

## III. OUR APPROACH

### A. Evolutionary No-Limit Texas Hold'em Poker

The main considerations of an evolutionary algorithm are generally representation, selection, recombination, and mutation. Representation is key, as it is used to define the meaning of recombination and mutation. The agent must be represented in such a manner that small changes can be exercised upon it, leading to differences in behaviour, but also in a manner such that multiple agents can be combined to create offspring with characteristics of all parents.

The Poker agents are implemented as evolving neural networks. The networks are 35-20-5 feed-forward neural networks, with a sigmoidal function applied at each level. The inputs are the set of features shown in Table III-A. Agents participate in tournaments consisting of up to 2,000 agents for 500 generations. After each generation, the agents who performed best in the tournaments are selected and retained.

One or more of these superior agents are selected as the parents of a newly created agent, known as a child agent. A normally distributed random bias is given to each of the parents. The weights of the child's neural network are a biased sum of the weights of the parents, according to equation 1, where $C_x$ is the $x^{th}$ weight of the Child agent's neural network, P0 through Pn are the $0^{th}$ through $n^{th}$ parent agents, and $B_0$ through $B_n$ are the biases given to the parents.

$$C_x = B_0 * P0_x + B_1 * P1_x + ... + B_n * Pn_x \qquad (1)$$

Learning is applied through a mutation operator. After each generation of evolution, normally distributed random noise is applied to a percentage of the weights of the networks of the new agents; the best agents from the previous generation remain the same.

TABLE I
INPUT FEATURES OF THE EVOLUTIONARY NEURAL NETWORK

| Feature | Description |
|---|---|
| Pot | Chips available to win |
| Call | Amount needed to call the bet |
| Opponents | Number of opponents |
| Win Percent | The percentage of hands that will win |
| Chips | The chip counts for each player |
| Total | The overall aggressiveness of each player |
| Current | The recent aggressiveness of each player |

The winning percentage is based upon a simulation of all possible future hands. Given the cards that are showing, this simulation can determine the number of times that an agent's hand will win, lose, or tie if the hand goes to a showdown. The percentage of hands that win or tie is a compromise of a hand's current strength and its potential to improve. The end result of a hand is felt to be more important than its current standing, and thus this compromise is made. The aggressiveness of an opponent is determined using a simple formula, where an agent receives an aggressiveness score of 0 for a fold, a 1 for a call, and a score of ($x$ / amount to call) for a raise, with $x$ being the amount raised. The special case of a *check* (i.e., a call of cost 0) is interpreted as being very unaggressive, and is also given a score of 0. The scores are averaged and kept as both an overall and recent aggressiveness score. The overall agressiveness keeps track of every decision since a player began playing against a particular opponent, while the recent aggressiveness is concerned with the most recent pass of 10 hands around the table, typically involving 15 to 40 betting decisions.

A selection method is needed to promote learning. In Poker, players are ranked according to how they perform in tournaments, and thus a selection method was developed to model this tournament ranking. For each generation, agents are placed in a playing population, and play numerous tournaments. They are then rated according to their average ranking across these tournaments. The best agents are segregated into a separate, elite population, and chosen as the parents for the next generation. Parents are combined using a weighted average of the weights of their neural networks, and the children become the new population. The weighting of the neural networks is randomly assigned, and after the combination, a small amount of normally distributed random noise is added to the weights of the neural networks, with a likelihood of 10%, a mean of 0, and standard deviation of 0.1. This mutation allows the children to diverge from the parents, allowing further exploration of the state space. The elite population is then inserted into the new population with no change made to their networks. This final step is made to ensure that the skill level of the previous generation is maintained.

### B. Additional Heuristics

Several heuristics were added to the evolutionary algorithm, to improve upon the evolution process. In [17], it is suggested that evolution by itself is not enough to promote a successful learning environment. Here, the use of co-evolution and a hall of fame structure is recommended to create a better environment.

Co-evolution uses multiple, genetically distinct populations in direct competition in an effort to increase the genetic diversity of the overall playing population. Competing populations play games against each other, but are evolved separately. In this way, agents that perform at a level that would be deemed poor at the global level may still survive within their respective population. Furthermore, as the generations progress, agents enter a so-called *arms race*, where each population orients itself towards evolving a strategy that can defeat opponents from the other population. As these opposing strategies meet and pass each other, the overall skill level of the entire population increases.

A hall of fame is a structure that is intended to preserve previously acquired and successful strategies. Although evolution is normally progressive, there are cases where a certain strategy can fall into disuse and be deleted from the population. The hall of fame contains a set of good strategies from previous generations, and is used to try to preserve these successful strategies. The members of the hall of fame compete against the regular populations, but are not allowed to pro-create. The strategies are not allowed to be lost, simply because they do not fall into disuse, as they are being used in competition against the members of the hall of fame.

## IV. EXPERIMENTAL RESULTS

In this section, we present the results of an experimental evaluation of the No-Limit Texas Hold'em Poker agents generated using our evolutionary neural network approach. A series of experiments were conducted to evaluate the quality of the agents generated when evolutionary neural networks were used alone, and when evolutionary neural networks were used in combination with co-evolution and a hall of fame. The software was implemented in Visual C++ using Microsoft Visual Studio 2005. All the experiments were run under Windows XP, on two IBM-compatible PCs, one with a 3.4 GHz Pentium D processor and 512 MB of memory, and the other with a 2.8 GHz Pentium D processor with 2 GB of memory.

The skills of the evolved agents were tested against a group of five static agents and three dynamic agents. The five static agents were developed to utilize a specific fixed playing strategy, and consist of an *auto-folding/calling/raising* (i.e., *Random*) agent that is equally likely to fold, call, or raise at each decision point, an *auto-raising* (i.e., *Raiser*) agent that always raises by some random value at each decision point, an *auto-folding* (i.e., *Folder*) agent that always lays down its cards at each decision point, an *auto-calling* (i.e, *Caller*) agent that always calls any bet made, and an *auto-calling/raising* (i.e., *CallOrRaise*) agent that is equally likely to call or raise at each decision point. The three dynamic agents were agents generated as part of some previous work we completed on evolving No-Limit Texas Hold'em Poker agents [2]. These agents are designated *OldBest* (the best agent evolved using our head-start approach), *OldScratch* (the best agent evolved from scratch without the head-start), and *OldStart* (created using our head-start approach, but no evolution).

A *duplicate tables tournament* format was used to rank the skills of the evolved agents versus the five static agents and the three dynamic agents. In a duplicate tables tournament, nine agents are seated at a single table. The tournament continues until one agent emerges as the winner (i.e., all the other agents have been eliminated from the tournament). The agents are ranked according to the order in which they are eliminated. That is, the first agent eliminated is given a rank of 10, the second a rank of 9, and so on. The winner is given a rank of 1. The agents are then re-seated at the table in the same position as at the start of the tournament just completed, and then each is shifted one seat to the left for the next tournament. The same sequence of cards is dealt to each duplicate table. The duplicate tables tournament is over once each agent has sat in every seat. Upon completion, the mean rank for each agent is determined, equal to the mean of the rankings obtained at each duplicate table of the duplicate tables tournament, to quantify their skill level. The objective of the duplicate tables tournament is to try to remove some of the variance from the rankings. That is, to reduce the effect of an agent's position at the table and the cards the agent sees as a result of being in a particular position. The idea is that a highly skilled agent will tend to win more frequently than a less skilled agent, regardless of its position at the table and the cards it might see.

### A. *Heuristic-Free Evolution*

The objective of the first experiment was to establish a set of baseline rankings for the nine agents, as shown in Figure 1. In Figure 1 (and in all other figures), the Rank value described along the vertical axis corresponds to the skill level of an agent relative to the other agents, where better agents receive higher (but numerically lower) ranks. *Control* was evolved over 500 generations, with a population of 1,000 agents competing in 500 tournaments per generation. In the first generation, the population consists of 1,000 randomly created agents. At the completion of the tournaments for each generation, the 100 most highly ranked agents (i.e., the elite agents) are used as the initial population for the next generation, and the other 900 agents in the population are created through the combination and mutation of these elite agents. At the completion of the tournaments for the 500-th generation, the most highly ranked agent was selected as the Control agent. The Control agent then played in an evaluation tournament consisting of 100,000 duplicate tables tournaments against the eight other agents.

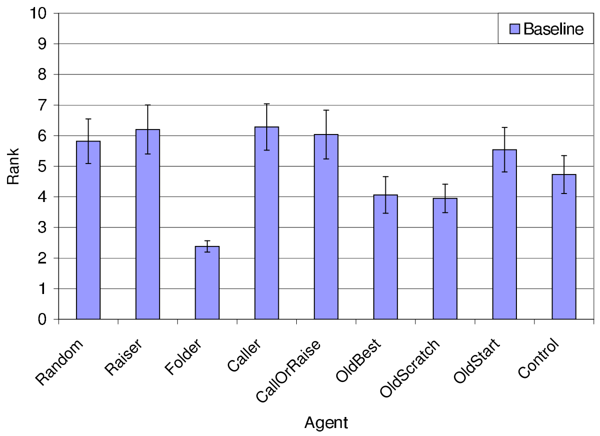


Fig. 1. Results of duplicate tournament testing control agent

In Figure 1, the lowest rank was 6.28, the highest rank was 2.38, the median rank was 5.54, and the standard deviations of the agent ranks varied from 0.19 to 0.80, with a median value of 0.73. Random, Raiser, Caller, and CallOrRaise (i.e, four of the static agents) performed poorest, receiving 6.28, 6.04, and 6.17 as the minimum, maximum, and mean ranks, respectively. Folder performed best with a rank of 2.38. Note however, that the strategy of Folder has an interesting characteristic within the context of No-Limit Texas Hold'em Poker. That is, it would seem that folding every hand and waiting for opponents to eliminate one another is a reasonable strategy, although the Auto-Fold agent will never win a hand, will never win a tournament, and, at best, will finish in second place. OldBest and OldScratch (i.e, two of the dynamic agents) performed similarly, receiving a mean rank of 4.01. OldStart received a rank of 5.54, while Control received a rank of 4.73, placing it slightly better than the median rank.

Finally, even though the duplicate tables tournament format contributed to a reduction in variance, the standard deviation is still relatively high, approximately 20% of the ranks obtained. Although agents will see roughly the same cards, a different decision at the same decision point can lead to an agent seeing cards that an opponent would not have seen in the same situation. Likewise, an agent may fold its cards where an opponent didn't, and miss seeing certain cards. These differing cards can lead to different results, although agents see roughly the same cards as their opponents. A two-tailed paired t-test was conducted on the null hypothesis that the ranks of any two distinct agents were equal. In all cases, and for all experiments, the null hypothesis was rejected with 99% confidence.

In the graphs that follow in Sections B through D, the baseline results are included for the reader's convenience.

### B. *Evolution with Multiple Populations*

The objective of the second and third experiments was to evaluate the effectiveness of co-evolution. The results of these experiments are shown in Figures 2 and 3, where the Control agents were evolved over 500 generations from a population of 1,000 agents split into two sub-populations (known as *2Pop-1* and *2Pop-2*) and four sub-populations (known as *4Pop-1*, *4Pop-2*, *4Pop-3*, and *4Pop-4*) of 500 and 250 agents, respectively. In the two and four sub-populations, agents are only ranked with respect to the other agents in their corresponding sub-population. At the completion of the tournaments for each generation, the 50 and 25 most highly ranked agents, respectively, are used as the initial population for the next generation, and the other 450 and 225 agents, respectively, in each sub-population are created through the combination and mutation of the elite agents in the corresponding sub-population. At the completion of the 500-th generation, the most highly ranked agents from each of the two and four sub-populations, respectively, were selected as the Control agents. The Control agents then played in their own evaluation tournaments against the eight other agents.

In Figure 2, the rankings for the five static agents and three dynamic agents are similar to those obtained in the baseline results of Figure 1. The median rank is 5.50 for 2Pop-1 and 5.64 for 2Pop-2. However, the rankings for co-evolved Control agents from 2Pop-1 and 2Pop-2 are higher than the baseline ranks, representing an increase of 3.7% and 21.0%, respectively. 2Pop-1 was also able to obtain a higher rank than OldBest and OldScratch, higher by 14.6% and 15.9%, respectively. The only difference between the

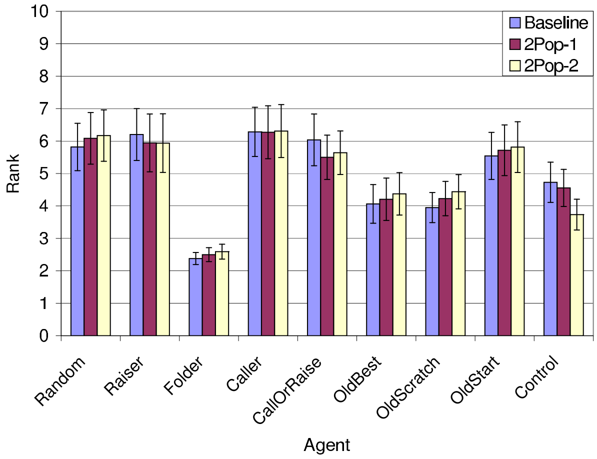


Fig. 2. Results of using 2 separate populations in the evolutionary process

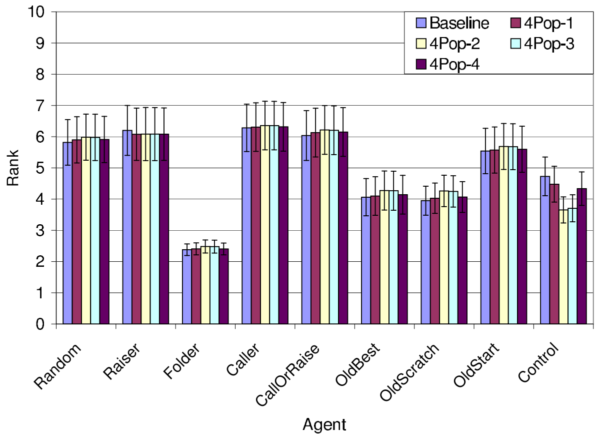


Fig. 3. Results of using 4 separate populations in the evolutionary process

Control agent in Figure 1 and the Control agents in Figure 2 is that the Control agents in Figure 2 were co-evolved. The results in Figure 3 are similar to those in Figure 2, with 4Pop-1, 4Pop-2, 4Pop-3, and 4Pop-4 showing an increase in rank of 5.3%, 22.7%, 21.6%, and 8.3%, respectively, and median rankings of 5.57, 5.69, 5.68, and 5.60, respectively. Furthermore, 4Pop-2 and 4Pop-3 also obtained a higher rank than OldBest by 14.5% and 13.2%, respectively, and OldScratch by 14.2% and 12.7%, respectively.

### C. Evolution with a Hall of Fame

The objective of the fourth experiment was to evaluate the effectiveness of the hall of fame structure. The results of this experiment are shown in Figure 4, where the Control agents were evolved over 500 generations. The *LargeHOF* and *SmallHOF* Control agents were evolved from a population of 1,000 agents (using a hall of fame of 1,000 agents) and 500 agents (using a hall of fame of 500 agents), respectively. Results obtained for LargeHOF and SmallHOF had 2,000 and 1,000 competing agents, respectively. That is, the agents in the population plus the agents in the hall of fame were competing, but the agents in the hall of fame could not be selected as parents for the next generation. At the completion of the tournaments for each generation, the 100 and 50 most highly ranked agents, respectively, are used as the initial population for the next generation, and the other 900 and 450 agents, respectively, are created through the combination and mutation of the elite agents. In addition, if any of the elite agents from LargeHOF and SmallHOF have ranked more highly than the 100 or 50 lowest ranked agents from their respective hall of fame, these elite agents replace the lower ranked agents in the hall of fame. At the completion of the 500-th generation, the most highly ranked agents from LargeHOF and SmallHOF, respectively, were selected as the Control agents. The Control agents then played in their own evaluation tournaments against the other eight agents.

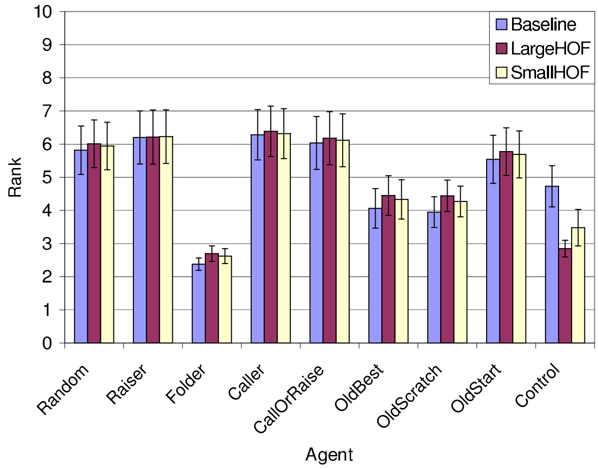


Fig. 4. Results of using 1 population of 1000 agents, along with a Hall of Fame of 1000 agents, and 1 population of 500 agents, along with a Hall of Fame of 500 agents

In Figure 4, the rankings for Random, Raiser, Caller, CallOrRaise, and OldStart for both LargeHOF and SmallHOF are similar to the baseline. For LargeHOF, Folder, OldBest, and OldScratch have a 13.3%, 9.5%, and 12.5% lower rank, respectively, than the corresponding baseline rank, and the median rank is 5.78. The Control agent from LargeHOF, with a rank of 2.85, has a rank 39.8% higher than the rank for the baseline Control agent, and has a higher rank than all other agents except Folder, which has a rank of 2.70. For SmallHOF, Folder, OldBest, and OldScratch have a 10.1%, 6.7%, and 8.1% lower rank, respectively, than the corresponding baseline rank, and the median rank is 5.69. The Control agent from SmallHOF, with a rank of 3.48, has a rank 26.4% higher than the rank for the baseline Control agent.

### D. Evolution with Multiple Populations and Multiple Halls of Fame

The objective of the final experiment was to evaluate the effectiveness of utilizing both a hall of fame and co-evolution in combination. The results of this experiment are shown in Figure 5, where the Control agents were evolved over 500 generations from a population of 1,000 agents split into two sub-populations (known as *HOF2Pop-1* and *HOF2Pop-2*) of 500 agents, each with its own hall of fame of 500 agents drawn from the corresponding sub-population. At the completion of the tournaments for each generation, elite agents are selected in the same manner as the two population experiment described in Section B, and these agents are inserted into their respective halls of fame in the same manner as the hall of fame experiments in Section C. At the completion of the 500-th generation, the most highly ranked agents from each of the two sub-populations were selected as the Control agents. The Control agents then played in their own evaluation tournaments against the other eight agents.

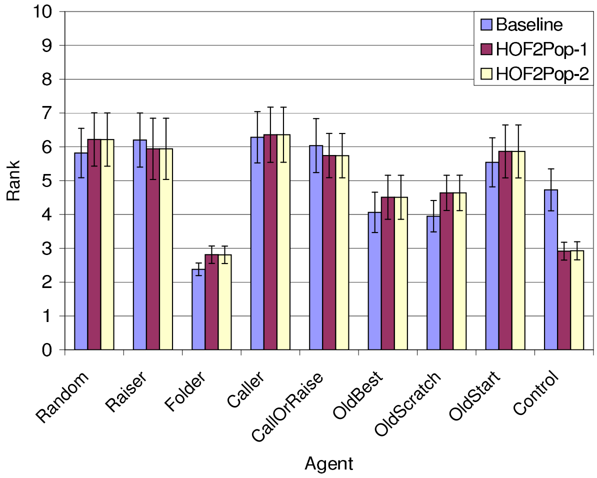


Fig. 5. Results of using 2 populations of 500 agents apiece, along with two Halls of Fame of 500 agents each

In Figure 5, the rankings for all of the agents are similar to those obtained in the hall of fame results shown in Figure 4. The Control agents from HOF2Pop-1 and HOF2Pop-2 have a 38.3% and 38.1% higher rank, respectively, than the baseline Control agent, and the median ranks are 5.74 and 5.73, respectively.

## V. CONCLUSION AND FUTURE WORK

Figure 6 shows a summary of the ranks obtained for the best Control agents from all five experiments. In Figure 6, the rank obtained for the baseline Control agent was lowest overall. The ranks obtained for the Control agents evolved using multiple populations (i.e., 2Pop-1, 4Pop-1, 4Pop-4, 2Pop-2, 4Pop-3, and 4Pop-2) showed some gradual improvement over the baseline Control agent, although results were mixed (i.e., the two and four population results were quite similar). More improvement was obtained using co-evolution of two populations combined with halls of fame (i.e., HOF2Pop-2 and HOF2Pop-1). However, the overall best result was obtained by the Control agent evolved from a single population using a large hall of fame (i.e., LargeHOF).

Future work will focus on attempts to further improve the skill levels of the evolved agents, particularly those evolved using a hall of fame. The hall of fame is currently finite,

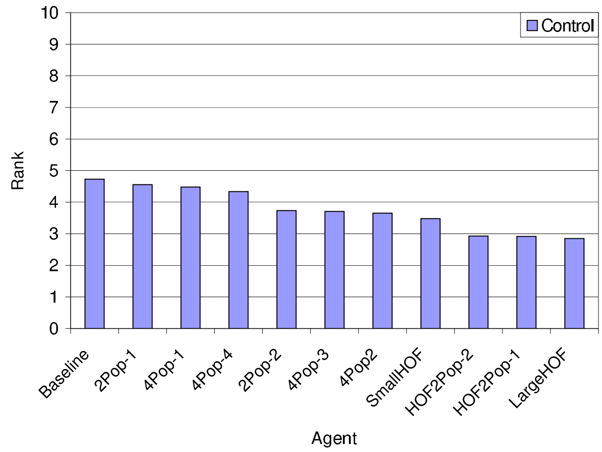


Fig. 6. Summary of all results

and while it improves the rank of the evolved agents, [17] suggests the use of an infinite hall of fame with random selection. Furthermore, the hall of fame is currently not capable of any form of reproduction. Optimization of the neural network and evolution should also be considered. Varying the mutation rate, number of hidden nodes, and mean and standard deviation of the noise applied for mutation has not been evaluated. Finally, the aggressiveness function needs further study. It is likely that a different aggressiveness function will lead to better results.

## REFERENCES


[1] L. Barone and L. While. An adaptive learning model for simplified poker using evolutionary algorithms. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 153–160, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[2] B. Beattie, G. Nicolai, D. Gerhard, and R. J. Hilderman. Pattern classification in no-limit poker: A head-start evolutionary approach. In *Canadian Conference on AI*, pages 204–215, 2007.

[3] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker, 2003.

[4] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.

[5] D. Billings, D. Papp, L. Pena, J. Schaeffer, and D. Szafron. Using selective-sampling simulations in poker, 1999.

[6] L. R. Booker. A no limit texas hold'em poker playing agent. Master's thesis, University of London, 2004.

[7] M. Campbell, A. J. H. Jr., and F. hsiung Hsu. Deep blue. *Artificial Intelligence*, 134:57–83, 2002.

[8] C. Donninger and U. Lorenz. The hydra project. *Xcell Journal*, 53:94–97, 2005.

[9] A. Hauptman and M. Sipper. Gp-endchess: Using genetic programming to evolve chess endgame players. In M. Keijzer, A. Tettamanzi, P. Collet, J. van Hemert, and M. Tomassini, editors, *Proceedings of the 8th European Conference on Genetic Programming*, 2005.

[10] M. B. Johanson. Robust stategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta, 2007.

[11] G. Kendall and G. Whitwell. An evolutionary approach for the tuning of a chess evaluation function using population dynamics. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, 2001.

[12] A. Lubberts and R. Miikkulainen. Co-evolving a go-playing neural network. In *Proceedings of the GECCO-01 Workshop on Coevolution: Turning Adaptive Algorithms upon Themselves*, 2001.

[13] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[14] G. Nicolai. Evolutionary methods for learning no-limit Texas hold'em poker. Master's thesis, University of Regina, 2008.

[15] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[16] J. B. Pollack and A. D. Blair. Co-evolution in the successful learning of backgammon strategy. *Mach. Learn.*, 32(3):225–240, 1998.

[17] C. D. Rosin. *Coevolutionary Search among adversaries*. PhD thesis, University of California, San Diego, 1997.

[18] A. L. Samuel. Some studies in machine learning using the game of checkers, 1959.

[19] J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, and D. Szafron. A world championship caliber checkers program. *Artif. Intell.*, 53(2-3):273–289, 1992.

[20] T. Schauenberg. Opponent modelling and search in poker. Master's thesis, University of Alberta, 2006.

[21] P. Straffin. *Game Theory and Strategy*. Mathematical Association of America, 1993.

[22] G. Tesauro. Programming backgammon using self-teaching neural nets. *Artif. Intell.*, 134(1-2):181–199, 2002.

[23] S. Thrun. Learning to play the game of chess. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 1069–1076. The MIT Press, Cambridge, MA, 1995.