DNA Starts to Learn Poker

David Harlan Wood[1⋆], Hong Bi[2⋆], Steven O. Kimbrough[3⋆⋆], D. J. Wu[4], and Junghuei Chen[2⋆]

[1] Computer Science, University of Delaware, Newark DE, 19716
[2] Chemistry and Biochemistry, University of Delaware, Newark DE, 19716
[3] The Wharton School, University of Pennsylvania, Philadelphia, PA, 19104
[4] Bennett S. Lebow College of Business, Drexel University, Philadelphia,PA, 19104

Abstract. DNA is used to implement a simplified version of poker. Strategies are evolved that mix bluffing with telling the truth. The essential features are (1) to wait your turn, (2) to default to the most conservative course, (3) to probabilistically override the default in some cases, and (4) to learn from payoffs. Two players each use an independent population of strategies that adapt and learn from their experiences in competition.

1 Introduction

The long-term goal is to use DNA to construct special purpose computers. Their special purpose is to learn game-playing strategies adapting to the strategies of opponents, even while opponent's strategies are also changing and adapting. It is clear that many real-world problems have this nature—and it is equally clear that no general solution method is known for these problems. The ultimate payoff for our research is a method of searching for adaptive game-theoretic strategies.

The ultimate aim is to use DNA to encode game strategies that improve over time and adapt to the strategies of other players. In the long term, this is to be addressed for the game of poker. In the medium term, a simplified 3-person poker [?] that has no "equilibrium" is to be addressed.

In the near term, this paper demonstrates the necessary DNA laboratory techniques for an example from a textbook on game theory [?]. This game is a simplified version of poker, but it still involves probabilistic strategies of bluffing versus truth-telling and calling versus folding. The essential features are to wait your turn, to default the most conservative course, and to probabilistically override this default in some contexts, and to learn from the payoffs obtained. Each of two players competes using a large population of strategies that adapt and learn from their experiences in competition.

We employ laboratory evolution of DNA [?,?,?,?,?], DNA hairpin extension [?,?,?], and the evolutionary computation paradigm from conventional computing [?,?,?]. All three of these techniques been used before, but have never been combined.

⋆ Partially supported by NSF Grant No. 9980092 and DARPA/NSF Grant No. 9725021.
⋆⋆ Partially supported by NSF Grant No. 9980092

1.1 The Advantages of DNA For Computing

Computations of evolving strategies seem particularly well suited to DNA implementation.

1. Estimated answers for a particular problem can be encoded in DNA molecules using binary representation.
2. Selection by fitness, and breeding via mutation and crossover, can be implemented by laboratory procedures, as demonstrated in [?,?].
3. Evolutionary computation, like natural evolution, benefits from tolerance of error [?,?], requiring only that selection be correlated with fitness.
4. Massive parallel processing of up to 10^{18} independent bytes of data is a characteristic of DNA laboratory processes (about one milligram). This is comparable to projected next-generation silicon computers [?].
5. A very large amount of information storage is available using DNA. For example, the entire Internet contains about the same amount of data as a milligram of DNA [?].
6. DNA laboratory procedures can multiplex many simultaneously evolving populations at no extra cost. Multiplexing permits large-scale sampling of the distribution of possible population evolutions.

2 Where Do Game Strategies Come From?

A game is a situation in which two or more players make moves (or plays). The reward received by a player for its moves depends in part on the moves made by the other player(s). The broad applicability of game theory not only ensures its importance, but also explains why game theory is unlikely to produce general methods for finding good strategies. While we admire the accomplishments of game theory, we regret that equilibrium and hyper-rationality are so often unrealistically assumed [?].

Playing competitive poker, for example, seems to be (1) a dynamic process of adapting one's strategies while (2) exploiting the mistakes of opponents. Regrettably, neither of these two features is usual in game-theoretic analysis. We depart from what has been the mainstream of game theory in that we focus on the dynamics of play and strategy creation, rather than on the statics and the various equilibrium concepts. See [?].

A recent commentary in Nature [?] nicely captures our perspective:

> "Of course, the main problem with Nash equilibria is still there: they may exist, but how does one reach them? . . . .We are in a situation akin to the beginning of mechanics: we can do the statics, but we don't have the dynamics."

Some of our prior research provided application of evolutionary computation to competitions [?,?,?,?]. We are led to using evolutionary computation because: (1) it is a general paradigm for exploring large search spaces. (2) its robustness under change or uncertainty is important since the very meaning of "good strategies" dynamically changes as opponents evolve their own strategies.

2.1 Complexity of Seeking Game Strategies

The complexity of the problem of finding good strategies can be indicated in the following way. Roughly speaking, all interesting games have exponentially many possible strategies. As for finding good strategies, no definite procedures are known which can consistently outperform simple enumeration. This is analogous to the complexity of seeking solutions or approximations for NP-complete problems, plus extra difficulties arising from dynamically changing situations.

Even in the case of finding Nash equilibrium, which we regard as regrettably static, no polynomial algorithm is known. In fact, Papadimitriou says [?],

> "...the complexity of finding a Nash equilibrium is in my opinion the most important concrete open question on the boundary of P today."

3 Simplified Poker via DNA

In this paper, we use an very simplified version of poker taken from a game theory textbook [?]. Even so, it incorporates bluffing, calling, and folding—all of which must be done with varying probabilities, if good payoffs are to be achieved.

1. There is a Dealer and a Player. Each contributes $1 into the pot to start one hand of play. The Dealer deals a single card, an Ace or a 2, so that only the Player can see it.
2. If the card it is an Ace, the Player must add $1 and say "Ace." If the card is a 2, the Player may say "2," losing the hand, or may add $1 and bluff by saying "Ace."
3. If the Player has said "Ace," it becomes the game continues an it is the Dealer's turn. The dealer may choose to fold, losing the hand and ending the game, or the Dealer may add $1 and call.
4. If called, the Player must show the card. Player wins the hand if the card is an Ace, and loses if it is a 2.

3.1 Strategies Are Learned by Playing Trillions of Simultaneous Hands Using DNA

Our DNA based implementation of playing poker is organized as shown in Fig. 1. This figure gives a broad overview of three independent but linked processes. The overall approach of selection by fitness, adding variation by crossover, is an extension of *in-vitro* evolution [?].

At the top, differing strategies compete, and the resulting histories of play are separated by outcomes.

In the middle, the many dealer-strategies are evaluated and selected by using a procedure based on payoffs achieved. This must be done carefully. Many selection criteria are possible. Considerable care is needed. For example, it is foolish to insist on consistently high payoffs because such strategies become predictable, and therefore exploitable by one's opponent. Population size is restored by making many copies of the selected strategies. Then, crossover can be used to induce

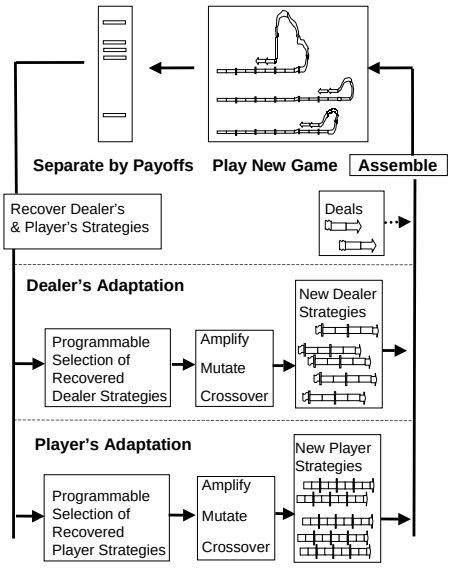

Fig. 1. Evolving poker playing strategies.

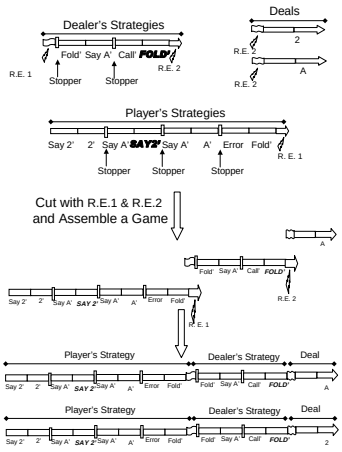

Fig. 2. A Player's strategy and a Dealer's strategy join for one hand of poker.

variation within the population of strategies. Finally, these new dealer strategies are entered into another round of competition.

At the bottom of Fig. 1 the other player uses a similar process, but with an independent method of selection.

3.2 Encoding Strategies in DNA Strands

The Player's strategies are encoded in single-stranded DNA as shown at the top of Fig. 2. This DNA strand consists of four pairs of labeled regions, with pairs separated by "stoppers." The roles of the various regions will be explained shortly.

It is important to note that all of the Player's strategy strands are identical except in one variable region, labeled *SAY 2'*. This region will vary throughout the Player's population of strategies. Its purpose, as we will see, is to implement diverse probabilities of bluffing. The Dealer's strategies are similarly encoded, having one variable region labeled *FOLD'* to implement various probabilities of calling. Of course, the variable regions are not predictable in advance. These variable regions will probabilistically determine the course of the play.

The cards to be dealt are Ace and 2, shown with unlabeled spacers at their left (labeled Deals in Fig. 2.)

Restriction enzymes, R.E. 1 and R.E. 2, cut DNA strands at specific locations, where indicated. This facilitates first joining each Dealer strategy strand to one Player strategy strand and to a Deal strand. The restriction enzymes are later used to sever these strands so they can be individually recovered (separated by length using gel electrophoresis).

Two situations can result, as shown at the bottom of Fig. 2. They differ in that their right hand ends have an Ace or a 2.

3.3 The DNA Sequences Used

The specific DNA sequences used are shown in Table 1. These sequences are based on [?] where they were used in DNA hairpin extension. The hairpin extension of [?] is similar to the play of one hand of simplified poker, except using two fixed non-probabilistic strategies.

Table 1. DNA sequences encoding simplified poker.

Names	Size	Sequences
A (Ace)	15-mer:	5' CCGTCTTCTTCTGCT 3'
A'	15-mer:	5' CCGTCTTCTTCTGCT 3'
2	15-mer:	5' TTCCCTCCCTCTCTT 3'
2'	15-mer:	5' AAGAGAGGGAGGGAA 3'
Say A'	15-mer:	5' CGTCCTCCTCTTGTT 3'
Say 2	15-mer:	5' CCCCTTCTTGTCCTT 3'
SAY 2'	15-mer:	Random with T,G, & C
Fold	15-mer:	5' TGCCCCTCTTGTTCT 3'
FOLD'	15-mer:	Random with T,G, & C
Call'	20-mer:	5' CTCCTCTTCCTTGCTCTTCTCCCTT 3'

3.4 DNA Hairpin Extension Plays a Hand of Poker

At the top of Fig. 3 below, two strategies are combined with a dealt Ace. The rest of the figure shows how the play of this one hand can result in two possible outcomes, depending on whether the Dealer decides to fold or call. The play of a hand when a 2 is dealt is similar.

Having been dealt an Ace, the Player must say "Ace." This is accomplished using DNA in the following way. The sequence encoding A at the end of the DNA strand strongly pairs with its Watson-Crick complementary sequence A'. This enables DNA polymerase to extend the strand by appending the sequence Say A'. Extension halts at a "stopper." To continue extension into a stopper region (encoded with 4 A-bases) would require dTTP, which is withheld from the reaction. Raising the temperature disrupts interstrand pairing. Recooling begins the Dealer's turn.

The Dealer must decide to call or fold. This is an IF-THEN-ELSE type decision, but we implement it in the form, "By default, fold, but if the probability is large enough, change your mind and call." That is, in the part of Fig. 3 labeled Dealer Folds, the Dealer's strategy encodes this situation and extends the DNA strand with the Fold sequence.

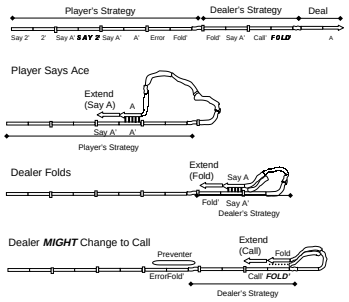

Fig. 3. Play of one hand of simplified poker in the case when an Ace is dealt.

After heating followed by cooling, the FOLD sequence may or may not pair with the *FOLD'* sequence, as at the bottom of Fig. 3. If and only if pairing occurs, the DNA strand is extended by the Call sequence, essentially changing the Dealer's decision from fold to call.

Success of pairing depends on the *FOLD'* sequence, which is generally different for different Dealer strategy strands. The *FOLD'* sequences in the initial population of strategies are initially randomized during the synthesis of Dealer's strategy strands. Therefore, the population of Dealer's strategies will generally produce some fold outcomes and some call outcomes. These outcomes are later used to select strategies by payoffs. Thus, it is the *FOLD'* sequences within Dealer's strategy strands that adapt by learning from outcomes.

3.5 Experimental Results

In this experiment, about a million distinct Dealer strategies are used. (Each one is likely to occur with many duplicates of itself.) This large variety of strategies is present because the *FOLD'* labeled region is randomized during the DNA synthesis. Similarly, about a million independent Player strategies are present.

The experiment consists of about a million million distinct hands of poker all being played at one time, with and Ace being dealt in all cases. Each hand to be played pairs up one Dealer strategy with one Player strategy.

Each hand played is some one strand of DNA, which is extended as shown in Fig. 3. Because of the randomization in the *FOLD'* region, some Dealer strategies will fold and some will call, as seen in Fig. 3.

Figure 4 shows the result of laboratory reactions that enable extensions of the poker hands encoded in DNA. The gel band at 282 shows that in some cases

a dealer strategy has called. The band at 262 are the cases where the dealer has folded. The band at 247 contains games yet to be completed, and the band at 232 consists of hands that have failed to extend.

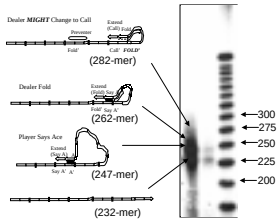

Fig. 4. The extensions illustrated in Fig. 3 are carried out in this experiment.

3.6 The Dealer and Player Independently Evolve Their Strategies

So far, we have explained how DNA laboratory techniques are able to pair off Dealer strategies and Player strategies along with a deal of an Ace or a 2. The result is: each such DNA strand is extended so that it records the entire history of the play of one hand of simplified poker. So far, we have elaborated on the top part of Fig. 1. We now go on to explain how the Dealer and the Player can independently evolve their populations of strategies.

Fig. 5A (the extensive form of the game of simplified poker, plus an error output) contains all five possible game histories, along with their payoffs, positive or negative, for the Dealer. The left side of the game tree corresponds to Fig. 3. Figure 5A also indicates how different final lengths of the DNA strands encode each of the possible histories.

Differing lengths make it convenient to physically separate histories using denaturing gel electrophoresis. Readout is provided by quantifying the amounts of DNA in each band of the gel. Other techniques could also be used, for example the 2d-DGGE techniques that we have used in evolutionary computations [?,?].

Physical separation by length via gel electrophoresis is indicated on the left of Fig. 5B. Each band of the gel corresponds to a different payoff.

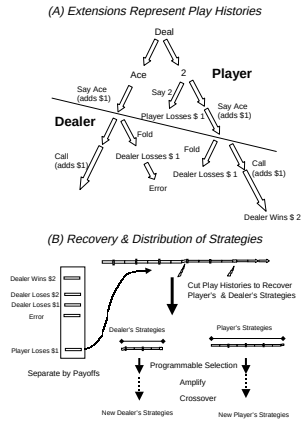

Fig. 5. Each branch of the game tree produces different length DNA strands.

What follows is that for each possible payoff the Dealer receives a quantified sample of the strategies that led to the given payoff. These samples are obtained by literally cutting the bands from the gel and extracting the DNA from them. The Dealer is then able to recombine strategies in various dilutions of her own choosing. Using this freedom of choice, and a chosen amount of crossover to explore further variations, the Dealer produces a new generation for strategies that will hopefully improve her net payoff. Improvement cannot be guaranteed, of course, because the Player is independently striving for the opposite outcome.

3.7 Regular Poker Would Use Similar Techniques

In regular poker decisions are similar but somewhat more complex. For example, the Dealer's decision in Fig. 3 would become the following. "By default, I fold, but append a copy of the hand I have been dealt. If my hand is good enough to match an evolved criterion, I can change my mind and call. But I append another copy of my hand and if it is good enough, I make a small raise." An additional comparison can result in a larger raise, etc.

A "wait your turn" feature is also shown in the last step in Fig. 3. Strictly speaking, this feature is not needed in simplified poker, but we wish to test it because it is needed in other games where players may take several turns. In essence, the Player's strategy is prepared to react to folding, but must not react before there is a chance for the Dealer to change from fold to call. Thus, as we cool the DNA we include a Preventer stand that preferentially (at higher temperature) pairs as shown in the Player's strategy. Should prevention fail, we would detect the presence of the Error sequence in some outcomes.

4 Anticipated Directions

We wish to address some game theoretic questions on the evolution of strategies for simplified poker. For simplified poker the questions can also addressed by analytic means, and by computer simulation. However, computer simulation would be difficult for populations as large as when using DNA.

The main outcome sought is to gain confidence in the DNA encodings and techniques that could be applied to more challenging games, especially poker. We will address questions such as the following. Is equilibrium maintained once it is induced? If one party uses an equilibrium strategy, will the other party evolve to equilibrium? If one party does a poor job of learning strategies, does the other party exploit this? What are good choices for programmable selection in evolving strategies? Will they result in obtaining equilibrium? If so, how fast? How much does crossover help? What crossover rates are best?

Thus, we have cited many more questions than answers. However, we hope to provide a technique for answering such questions—namely, taking advantage of the massive parallelism of DNA computing to test huge numbers of strategies in competition and to improve them based on their outcomes.

References

1. Nash, Jr., J.F., Shapley, L.S.: A simple three-person poker game. In Kuhn, H.W., Tucker, A.W., eds.: Contributions to the Theory of Games. Annals of Mathematics Studies. Princeton University Press, Princeton, NJ (1950) 105–116
2. Thomas, L.C.: Games, Theory and Applications. Ellis Horwood, Ltd., West Sussex, England (1984)
3. Chen, J., Wood, D.H.: Computation with biomolecules. Proceedings of the National Academy of Sciences, USA **97** (2000) 1328–1330 Commentary.
4. Wood, D.H., Chen, J., Antipov, E., Lemieux, B., Cedeño, W.: A design for DNA computation of the OneMax problem. Soft Computing **5** (2001) 19–24
5. Wood, D.H., Chen, J., Antipov, E., Lemieux, B., Cedeño, W.: *In vitro* selection for a OneMax DNA evolutionary computation. In Gifford, D., Winfree, E., eds.: DNA Based Computers V: DIMACS Workshop, June 14-15, 1999. Volume 54 of DIMACS series in discrete mathematics and theoretical computer science., Providence, American Mathematical Society (2000) 23–37
6. Deaton, R., Murphy, R.C., Rose, J.A., Garzon, M.H., Franceschetti, D.R., Jr., S.E.S.: A DNA based implementation of an evolutionary search for good encodings for DNA computation. In: Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, Indianapolis, Indiana, IEEE Press (1997) 267–271
7. Bäck, T., Kok, J.N., Rozenberg, G.: Evolutionary computation as a paradigm for DNA-based computing. In Landweber, L., Winfree, E., Lipton, R., Freeland, S., eds.: Preliminary Proceedings DIMACS Workshop on Evolution as Computation, DIMACS, Piscataway NJ (1999) 67–88 Available on request from DIMACS. Paper found at URL: http://www.wi.LeidenUniv.nl/~joost.
8. Sakamoto, K., Gouzu, H., Komiya, K., Kiga, D., Yokoyama, S., Yokomori, T., Hagiya, M.: Molecular computation by DNA hairpin formation. Science **288** (2000) 1223–1226

9. Komiya, K., Sakamoto, K., Gouzu, H., Yokoyama, S., Arita, M., Nishikawa, A., Hagiya, M.: Successive state transitions with I/O interface by molecules. In Condon, A., Rozenberg, G., eds.: DNA Computing: Revised papers / 6th International Workshop on DNA-Based Computers, DNA 2000, Leiden, The Netherlands, June 13–17, 2000. Volume 2054 of Lecture Notes in Computer Science., Berlin, Springer-Verlag (2000) 17–26
10. Winfree, E.: Whiplash PCR for $O(1)$ computing. Unpublished manuscript available from http://dope.caltech.edu/winfree/Papers/pcr.ps (1998)
11. Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: Handbook of Evolutionary Computation. Institute of Physics Publishing, Philadelphia (1997)
12. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley (1989)
13. Vose, M.D.: The Simple Genetic Algorithm. MIT Press, Cambridge (1999)
14. Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. Complex Systems **6** (1992) 333–362
15. Goldberg, D.E., Miller, B.L.: Genetic algorithms, selection schemes, and the varying effects of noise. Evolutionary Computation **4** (1996) 113–131
16. : High end architectures. Web page at http://www.ccic.gov/pubs/blue00/hecc.html#architectures (2000) National Coordination Office for Computing, Information, and Communications.
17. Lawrence, S., Giles, C.: Accessibility of information on the web. Nature **400** (1999) 107—109
18. Von Neumann, J., Morgenstern, O.: Theory of Games & Economic Behavior. Princeton University Press, Princeton, NJ (1953) ISBN:0691003629.
19. Maynard Smith, J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge, Great Britain (1982) ISBN: 0-521-28884-3.
20. Ekeland, I.: Agreeing on strategies. Nature **400** (1999) 623–624
21. Kimbrough, S.O., Dworman, G.O., Laing, J.D.: On automated discovery of models using genetic programming: Bargaining in a three-agent coalitions game. Journal of Management Information Systems **12** (Winter 1995-96) 97–125
22. Kimbrough, S.O., Dworman, G.O., Laing, J.D.: Bargaining by artificial agents in two coalition games: A study in genetic programming for electronic commerce. In Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L., eds.: Genetic Programming 1996: Proceedings of the First Annual Genetic Programming Conference, July 28-31, 1996, Stanford University. The MIT Press (1996) 54–62
23. Wu, D.J.: Discovering near-optimal pricing strategies for the deregulated power marketplace using genetic algorithms. Decision Support Systems **27** (1999) 25–45
24. Wu, D.J.: Artificial agents for discovering business strategies for network industries. International Journal of Electronic Commerce **5** (Fall 2000)
25. Papadimitriou, C.H.: Algorithms, games, and the internet. In: Proceedings of the 33rd annual ACM Symposium on Theory of Computing: STOC'01, July 6–8, 2001, Hersonissos, Crete, Greece, ACM Press, New York (2001) 749–753
26. Forst, C.V.: Molecular evolution. Journal of Biotechnology **276** (1997) 546–547