DNA Starts to Learn Poker

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

Departments of [1]Chemistry & Biochemistry and [4]Computer & Information Sciences, University of Delaware, Newark, DE 19716
{wood, hongbi, junghuei}@udel.edu
[2]The Wharton School, University of Pennsylvania, Philadelphia, PA 19104 kimbrough@wharton.upenn.edu
[3]Benett S. Lebow College of Business, Drexel University, Philadelphia, PA 19104 wudj@drexel.edu

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 method is known for these problems. The ultimate payoff for our research is a method for finding 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 medium term, this is to be addressed for the game of poker.

This project demonstrates the necessary DNA laboratory techniques for an example from a textbook on game theory (49). This 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, to aggressively override this 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 (3-5, 9,13,20,58-61), whiplash PCR (22, 30, 38,39,55), and the evolutionary computation paradigm from conventional computing (2,12,21,23,25-28,35, 40,51,52,62,63). All three techniques been used before, but have never been combined. One obtains massively parallel nanoscale computers where communication is not dependent on a myriad of fixed physical connections, but rather on pattern matching of information encoded in independent free-floating molecules.

!.1 The Advantages of DNA For Computing

Computations of evolving strategies seem particularly well suited to DNA implementation (4-8,20,41-46,59-61).

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 (59-61).
3. Evolutionary computation, like natural evolution, benefits from tolerance of error (18, 19), 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[1] projected next-generation silicon computers (24).
5. DNA laboratory procedures can multiplex[2] many simultaneously evolving populations at no extra cost. Multiplexing permits large-scale sampling of the distribution of possible population evolutions.
6. 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[3].

[1] Assuming one laboratory operation per minute.

[2] In multiplexing, populations are combined, independently and uniformly processed, and separately recovered.

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[4]. Consider poker. 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 a dynamic process of adapting one's strategies to exploit 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 (e.g., Nash and its refinements). 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
We are led to using evolutionary computation because: (I
(II) Its robustness under change or uncertainty is importa
changes as opponents evolve their own strategies.

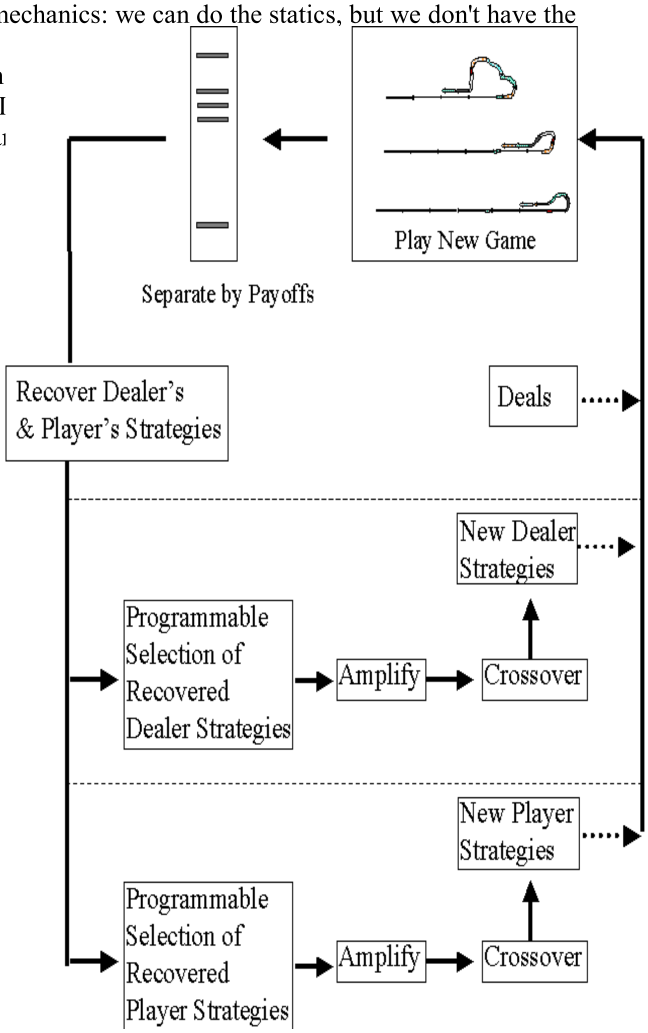

Separate by Payoffs
Play New Game
Recover Dealer's & Player's Strategies
Deals
New Dealer Strategies
Programmable Selection of Recovered Dealer Strategies
Amplify
Crossover
New Player Strategies
Programmable Selection of Recovered Player Strategies
Amplify
Crossover

Fig. 1. Evolving poker playing 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.

3 Simplified Poker via DNA

In this paper, we use an very simplified version of poker taken from a game theory textbook (49). Even so, it incorporates bluffing, calling, and folding --- all of which must be done with varying probabilities, if good payoffs are to be achieved.

> 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. If 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." If the Player has said "Ace," it becomes the Dealer's turn. The dealer may choose to fold, losing the hand, or may add $1 and call. At this point the Player shows the card and wins the hand if it is an Ace, and loses if it is a 2.

[3] The Internet is estimated to contain 9 x 10^{16} bytes (32). One gram of single-stranded DNA is approximately 1.8x 10^{21} nucleotides or about 10^{22} bytes.

[4] A strategy is a set of instructions for all possible situations that can be encountered in a game.

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 Figure 1. This gives a broad overview of three independent but linked processes. The overall approach of selection by fitness, adding variation by crossover, is similar to in-vitro evolution (1,5,11,14,16,36).

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, and it is foolish to insist on consistently high payoffs. Population size is restored by amplifying the selected strategies. Then, crossover induces variation within the population of strategies. Finally, the new dealer strategies are entered into another round of competition.

At the bottom, 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 Figure 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.

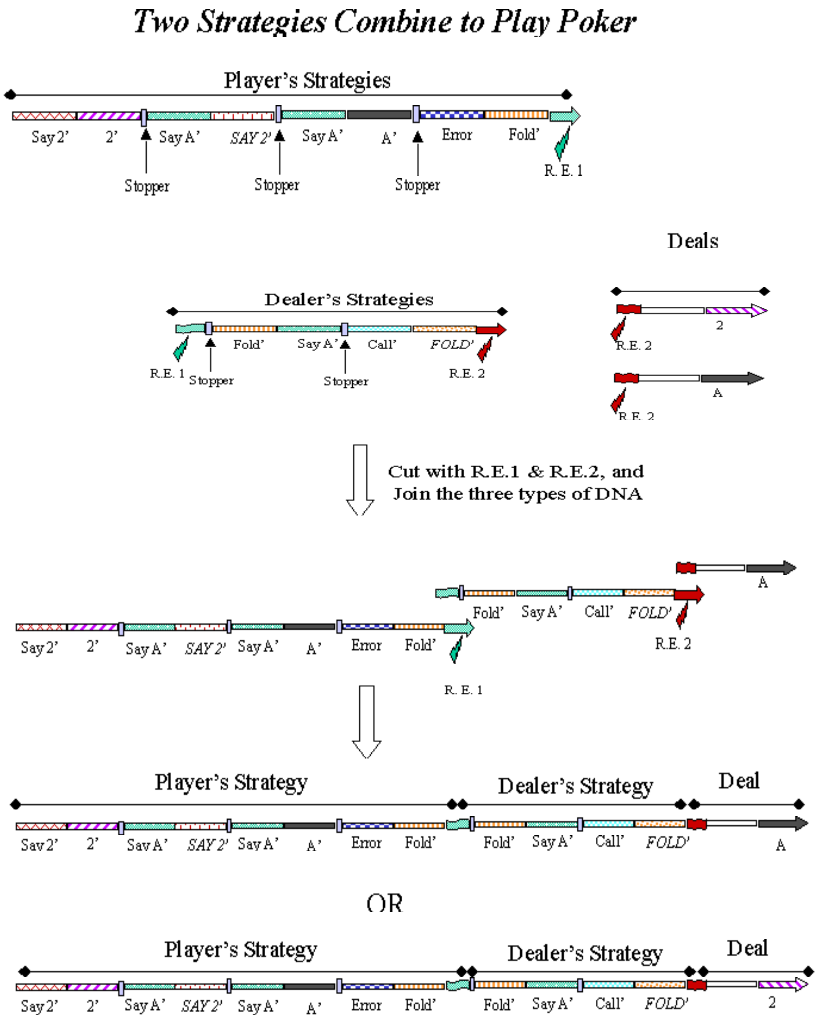
The cards to be dealt are Ace and 2, shown with unlabeled spacers at their left (labeled Deals in Figure 2).

Restriction enzymes, R.E. 1 and R.E. 2, cut DNA strands at specific locations, where indicated. This facilitates first joining Dealer strategy strands to Player strategy strands to Deal strands; and to later 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 Figure 2. They differ by ending with A or with 2. Of course, the variable regions are not predictable in advance. These variable regions will probabilistically determine the course of the play.

3.3 The DNA Sequences Used

The specific DNA sequences used are shown in Table 1. These sequences are based on (32) where they were used in Whiplash[5] PCR, which is similar to the play of one Hand of simplified poker using two fixed strategies.



l
g. 2. A Player's and a Dealer's strategy join a deal for a hand of poker.

[5] Sometimes called hairpin or banjo PCR.

Table 1.

Names	Size	Sequences
A (Ace)	15-mer:	5′CCGTCTTCTTCTGCT3′
A′	15-mer:	5′AGCAGAAGAAGACGG3′
2	15-mer:	5′TTCCCTCCCTCTCTT3′
2′	15-mer:	5′AAGAGAGGGAGGGAA3′
Say A′	15-mer:	5′CGTCCTCCTCTTGTT3′
Say 2	15-mer:	5′CCCCTTCTTGTCCTT3′
SAY 2′	**15-mer:**	**Random with T,G,& C**
Fold′	15-mer:	5′TGCCCCTCTTGTTCT3′
FOLD′	**15-mer:**	**Random with T,G,& C**
Call′	20-mer:	5′CTCCTCTTCCTTGCT CTTCTCCCTT3′
Error	10-mer:	5′TCCCCTTGTG3′

3.4 Whiplash PCR Plays a Hand of Poker

At the top of Figure 3 below, two strategies are combined with a dealt Ace. The rest of the figure shows how the play of the hand can result in two possible outcomes, depending on whether the Dealer decides to fold or call. Figure 4 is almost the same, showing the play of a hand when a 2 is dealt.

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 [6] 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 Dealers 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 Figure 3 labeled Dealer Folds, the Dealer's strategy encodes this situation and extends the DNA strand with the Fold sequence. After heating followed by cooling, the Fold sequence may or may not pair with the FOLD' sequence, as at the bottom of Figure 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.

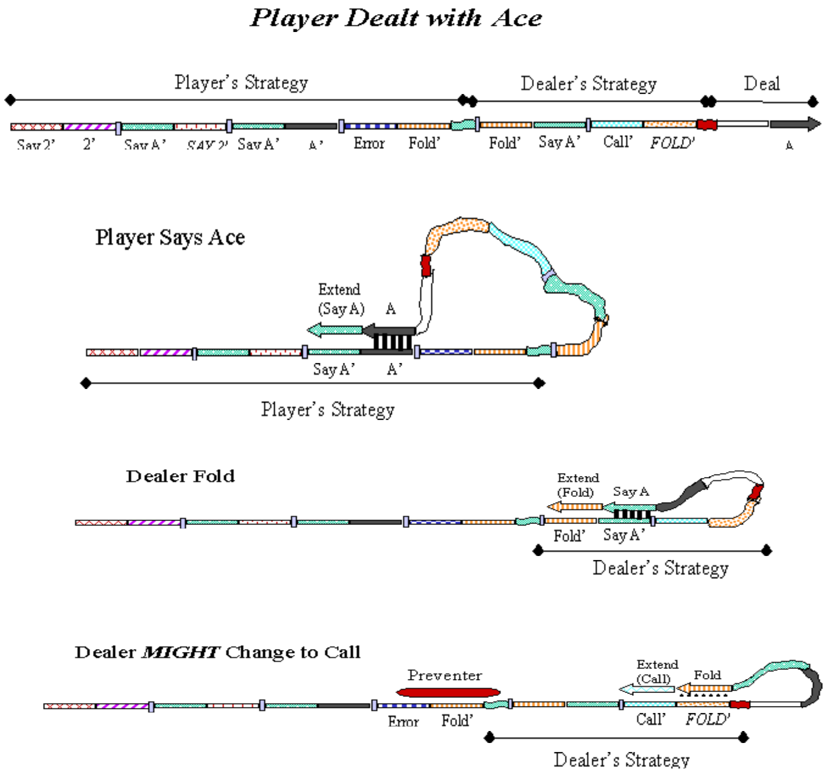


Fig. 3 Player is dealt an Ace.

Success of pairing depends on the FOLD' sequence, which is generally different for different Dealer 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 do the adapting. The FOLD' sequences in the initial population of strategies are randomized during the synthesis of Dealer's strategy strands.

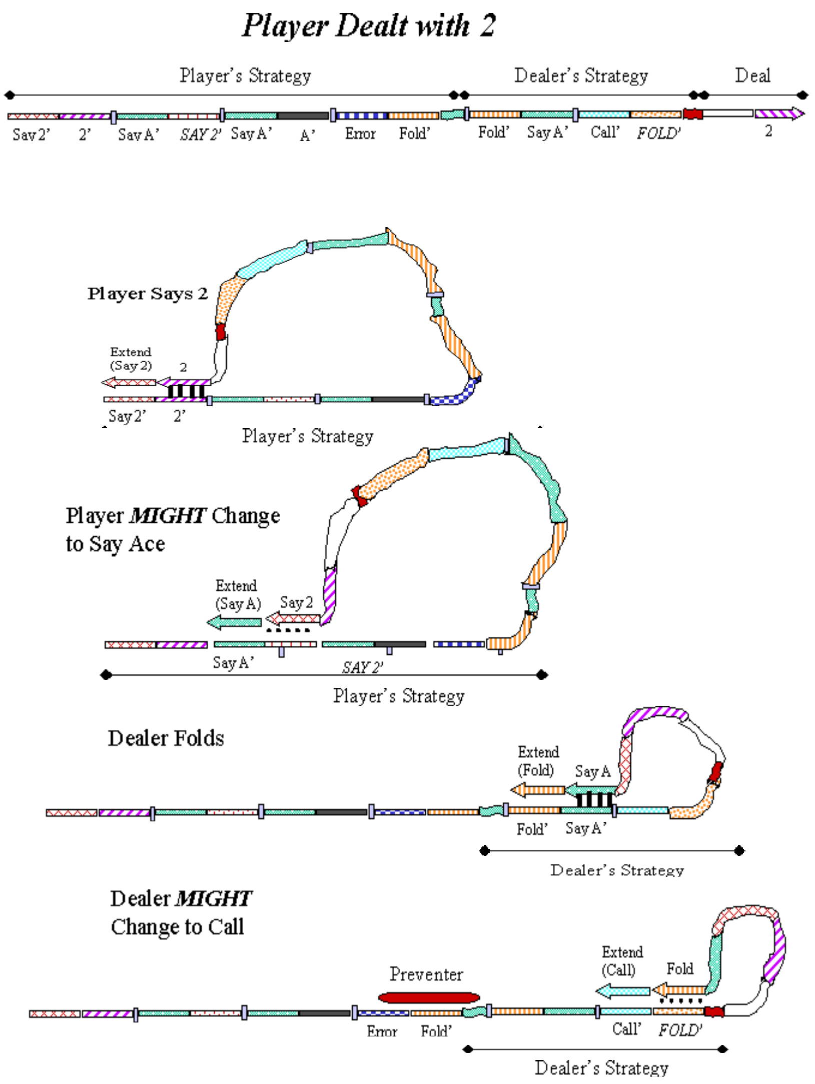
[6] In Watson-Crick complementary pairing within double stranded DNA, each base A in one strand pairs with a base T in the other strand, and similarly, each base G pairs with a base C. Any such pairings tend to hold two strands together.

3.5 Regular Poker Would Use Similar Techniques

A “Wait Your Turn” feature is also shown in the last step in Figure 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.

In regular poker decisions are similar but somewhat more complex. For example, the Dealer’s decision in Figure 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.

Fig. 4. Player is dealt a 2.



g. 4. Player is dealt a 2.

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 a DNA strand recording the entire history of the play of one hand of simplified poker. So far, we have elaborated on the top part of Figure 1. We now go on to explain how the Dealer and the Player can independently evolve their populations of strategies.

Figure 5 A (the extrinsic 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 corresponds to Figure 3 and the right side to Figure 4. Figure 5 A also reflects the different final lengths of the DNA strands encoding each of the histories.

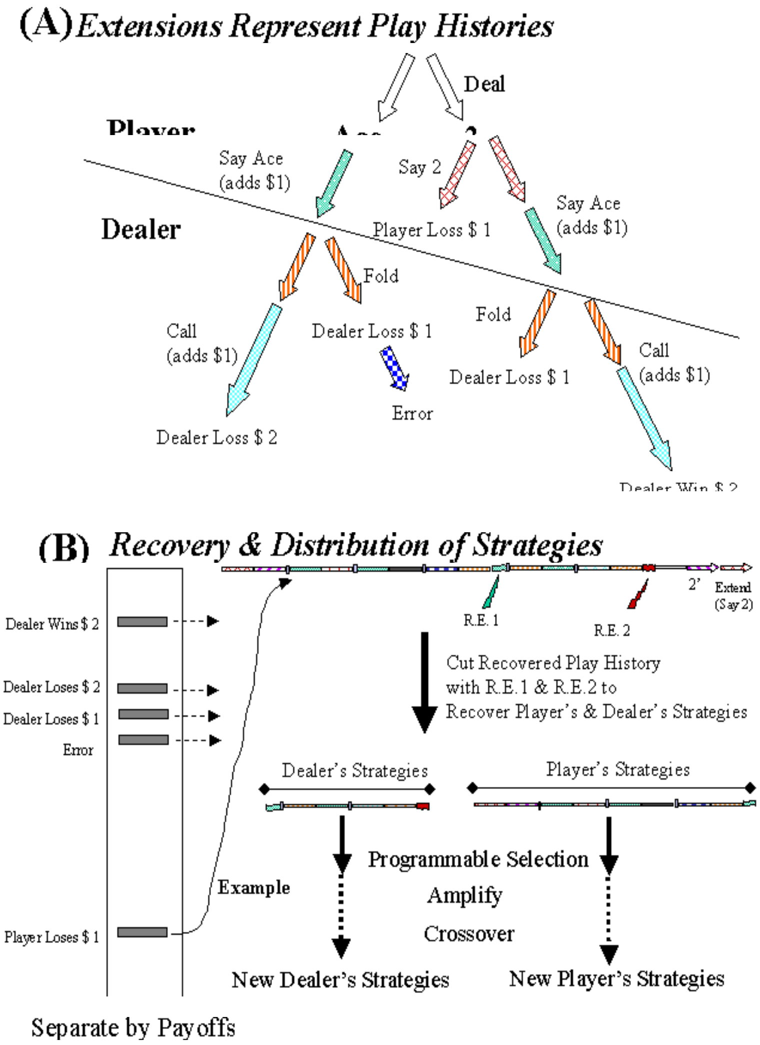
Differing lengths make it convenient to physically separate histories using denaturing gel electrophoresis. Readout is provided by quantifying the amounts of DNA in each band of the gel. Other techniques could also be used, for example the 2d-DGGE techniques that we have used in evolutionary computations (4-5, 20, 59-61).

Physical separation gel is indicated on the left of Figure 5 B. Each lane of the gel corresponds to a different payoff. 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.

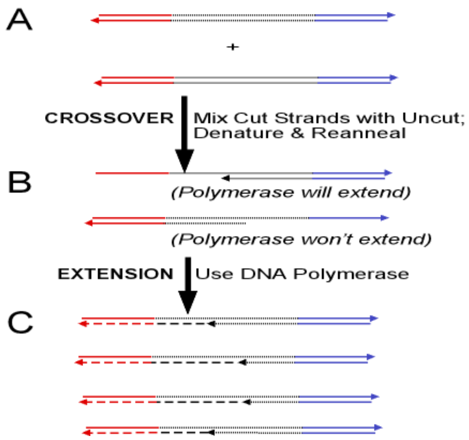
3.7 Implementation of Crossover

The most common recombining of genetic information in evolutionary computation is single point crossover (2). Here, the beginning part of an offspring's genetic material comes from one parent and the rest from a second parent[7].

Laboratory procedures are available for crossover (47, 48). Our method (5, 59-61), shown in Figure 6, is designed to accommodate highly variable regions. It begins with double stranded PCR product obtained restoring population size. Part of the PCR product is reserved. The unreserved DNA strands are cut at random locations about once per strand[8] and then combined with the reserved strands. The mixture is denatured (strands are melted apart) and allowed to reanneal (cool down). This allows single strands to recombine in various configurations. Some of these configurations are suitable for completing crossover.



g. 5. Each branch of the game tree gives different DNA lengths.



Recall that the ends of the DNA strands encode PCR primer sites. This means the rightmost (3') end is designed to Watson-Crick pair to its complement in correct alignment at a relatively high temperature during cooling. Thus, a truncated Players' or Dealers' strategies DNA strand can pair in proper alignment with an intact complement of a different DNA strand. As shown in Figure 6B (top), some of the truncated DNA strands may additionally pair well at their

[7] Multiple-point, multiple-parent crossover could be easily obtained by iterating single-point crossover.

[8] We use the enzyme *DNAase I* to mostly nick (cut only one strand) double stranded PCR product. Strand nicking could also be done by other enzymes or chemicals.

other (3’) ends. When this happens, they can be extended to full length using DNA polymerase. The result is single-point crossover. Many undesired products not shown in Figure 6 are also formed in the crossover reactions. But virtually none of the undesired products achieve the same length as the original DNA strands. Therefore, the desired single-point crossover products can be purified by length using denaturing gel electrophoresis.

3.8 Experimental Results

We have constructed the two ssDNAs to represent the two possible plays shown in Figures 3 and 4. Currently, demonstration of whiplash PCR is in process. Preliminary experimental results will be presented in the meeting.

4 Anticipated Directions

We attempt 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, whenever possible, and by computer simulation. However, computer simulation would be difficult for populations as large as 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.

5 References

[1] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. *Natur*e, 400:107—109, 1999.

[2] K. Komiya, K. Sakamoto, H. Gouzu, S. Yokoyama, M. Arita, A. Nishikawa, and M. Hagiya. Successive state transitions with I/O interface by molecules. In Condon and Rozenberg [62], pages 21–30.

[3] Kensaku Sakamoto, Daisuke Kiga, Ken Komiya, Hidetaka Gouzu, Shigeyuki Yokoyama, Shuji Ikeda, Hiroshi Sugiyama, and Masami Hagiya. State transitions by molecules. In *Proceedings of the Fourth Interantional Meeting on DNA BASED COMPUTERS* [63], pages 81–91.

[4] Masami Hagiya, Masanori Arita, Daisuke Kiga, Kansaku Sakamoto, and Shigeyuki Yokoyama. Towards parallel evaluation and learning of boolean formulas with molecules. In *DNA Based Computers III: DIMACS Workshop, June 23-25, 1997* [11], pages 105–114.

[5] Erik Winfree. Whiplash PCR for O(1) computing. Unpublished manuscript availble from http://dope.caltech.edu/winfree/Papers/pcr.ps, 1998.

[6] High end architectures. Web page at http://www.ccic.gov/pubs/blue00/hecc.html#architectures, 2000. National Coordination Office for Computing, Information, and Communications.

[7] L. C. Thomas. *Games, Theory and Application*s. Ellis Horwood, Ltd., West Sussex, England, 1984.

[8] J¨ orgen W. Weibull. *Evolutionary Game Theor*y. The MIT Press, Cambridge, MA, 1995.

[9] John Maynard Smith. *Evolution and the Theory of Game*s. Cambridge University Press, Cambridge, Great Britain, 1982.

[10] John Von Neumann and Oskar Morgenstern. *Theory of Games & Economic Behavio*r. Princeton University Press, Princeton, NJ, 1953.

[11] Harvey Rubin and David Harlan Wood, editors. *DNA Based Computers III: DIMACS Workshop, June 23-25, 199*7, volume 48 of *DIMACS series in discrete mathematics and theoretical computer scienc*e, Providence, 1999. American Mathematical Society.

[12] David Harlan Wood and Junghuei Chen. Toward large populations for Royal Road problems via DNA. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *1999 Congress on Evolutionary Computatio*n, pages 1011–1016. IEEE Computer Society Press, New York, 1999.

[13] N. C. Seeman and J. Chen. Synthesis from DNA of a molecule with the connectivity of a cube. *Natur*e, 350:631–633, 1991.

[14] N. C. Seeman, Y. Zhang, S. Du, H. Wang, J. E. Mueller, and J. Chen. The control of DNA structure and topology: An overview. *Mat. Res. Soc. Symp. Proc*., 356:57–66, 1994.

[15] N. C. Seeman, Y. Zhang, and J. Chen. DNA nanoconstructions. *J. Vac. Sci. Technol*., A 12(4):1895–

1905, 1994.

[16] N. C. Seeman, Y. Zhang, S. M. Du, and J. Chen. Construction of DNA polyhedra and knots through symmetry minimization. In J. S. Siegel, editor, *Supramolecular Stereochemistr*y, pages 27–32, 1995.

[17] Nadrian C. Seeman, Hui Wang, Bing Liu, Jing Qi, Xiaojun Li, Xiaoping Yang, Furong Liu, Weiqiong Sun, Zhiyong Shen, Ruojie Sha, Chengde Mao, Yinli Wang, Siwei Zhang, Tsu-Ju Fu, Shouming Du, John E. Mueller, Yuwen Zhang, and Junghuei Chen. The perils of polynucleotides: The experimental gap between the design and assembly of unusual DNAstructures. In *DNA Based Computers II: DIMACS Workshop, June 10-12, 199*6. American Mathematical Society, 1996.

[18] N. C Seeman, J. Qi, X. Li, X. Yang, N. B. Leontis, B. Liu, Y. Zhang, S. M. Du, and J. Chen. The control of DNA structure: From topological modules to geometrical modules. In J. Michl, editor, *Modular Chemistr*y. Kluwer, to appear.

[19] Junghuei Chen and David Harlan Wood. A new DNA separation technique with low error rate. In Rubin and Wood [11], pages 47–56.

[20] Junghuei Chen and David Harlan Wood. A new DNA separation technique with low error rate. In Rubin and Wood [11], pages 43–56.

[21] Junghuei Chen and David Harlan Wood. The role of high-fidelity separation in DNA computing. In *Proeedings of the 7th International Conference on Molecular Electronics & and Bio-Computing, November 10–12, 1997, Nanjing, Chin*a, pages 124–127, 1997. Invited plenary address.

[22] David Harlan Wood, Junghuei Chen, Eugene Antipov, Bertrand Lemieux, and Walter Cede˜ no. *In vitro* selection for a OneMax DNA evolutionary computation. In David Gifford and Erik Winfree, editors, *DNA Based Computers V: DIMACS Workshop, June 14-15, 199*9, volume 54 of *DIMACS series in dicrete mathematics and theoretical computer scienc*e, pages 23–37, Providence, 2000. American Mathematical Society.

[23] R. Deaton, R. C. Murphy, J. A. Rose, Max H. Garzon, Donald R. Franceschetti, and S. E. Stevens Jr. A DNA based implementation of an evolutionary search for good encodings for DNA computation. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computatio*n, pages 267–271, Indianapolis, Indiana, April 13–16, 1997. IEEE Press.

[24] J. C. Cox, P. Rudolph, and A. D. Ellington. Automated RNA selection. *Biotechnology Progres*s, 14(6):845–850, December, 1998.

[25] A. D. Ellington, M. P. Robertson, and J. Bull. *In vitro* evolution - Ribozymes in wonderland. *Scienc*e, 276(5312):546–547, April 25, 1997.

[26] David Harlan Wood, Junghuei Chen, Eugene Antipov, Walter Cede˜ no, and Bertrand Lemieux. A DNA implementation of the Max 1s problem. In Wolfgang Banzhaf, A. E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, 1999, Orlando, Florida USA.*, pages 1835–1842, San Francisco, 1999. Morgan Kaufman.

[27] David Harlan Wood. A DNA computing algorithm for directed Hamiltonian paths. In John Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoym Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22-25, 1998, University of Wisconsin, Madison, Wisconsi*n, pages 731–734, San Francisco, 1998. Morgan Kaufman.

[28] David E. Goldberg, Kalyanmoy Deb, and James H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex System*s, 6(4):333–362, August 1992.

[29] David E. Goldberg and B. L. Miller. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computatio*n, 4(2):113–131, 1996.

[30] John H. Holland. *Adaptation in natural and artificial system*s. MIT Press, Boston, 1992.

[31] D. L. Robertson and F. G. Joyce. Selection *in vitro* of an RNA enzyme that specifically cleaves single-stranded DNA. *Natur*e, 344(6265):467–468, March 29, 1990.

[32] Thomas B¨ ack, Joost N. Kok, and Grzegorz Rozenberg. Evolutionary computation as a paradigm for DNA-based computing. In Laura Landweber, Erik Winfree, Richard Lipton, and Stephan Freeland, editors, *Preliminary Proceedings DIMACS Workshop on Evolution as Computatio*n, pages 67–88, DIMACS, Piscataway NJ, January 1999. Available on request from DIMACS. Paper found at URL: http://www.wi.LeidenUniv.nl/˜joost.

[33] C. V. Forst. Molecular evolution. *Journal of Biotechnolog*y, 276(5312):546–547, April 25, 1997.

[34] Dirk Faulhammer, Anthony R. Cukras, Richard J. Lipton, and Laura F. Landweber. Molecular computation: RNA solutions to chess problems. *Proceedings of the National Academy of Sciences, US*A, 97(4):1385–1389, 2000.

[35] Junghuei Chen and David Harlan Wood. Computation with biomolecules. *Proceedings of the National*

*Academy of Sciences, US*A, 97(4):1328–1330, 2000. Commentary.

[36] Elizabeth Goode, David Harlan Wood, and Junghuei Chen. DNA implementation of Royal Road fitness evaluation. In Condon and Rozenberg [62], pages 223–237.

[37] Chengde Mao, Thomas H. LaBean, John H. Reif, and Nadrian C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Natur*e, 407:493–496, 2000.

[38] WillemP. C. Stemmer. DNA shuffling by random fragmentation and reassembly: *In vitro* recombination for molecular evolution. *Proceedings of the National Academy of Science, U.S.A.*, 91:389–391, 1994.

[39] Willem P. C. Stemmer. Searching sequence space. *Bio/Technolog*y, 13:549–553, 1995.

[40] Junghuei Chen, Eugene Antipov, Bertrand Lemieux, Walter Cede˜ no, and David Harlan Wood. DNA computing implementing genetic algorithms. In Laura Landweber, Erik Winfree, Richard Lipton, and Stephan Freeland, editors, *Preliminary Proceedings DIMACS Workshop on Evolution as Computatio*n, pages 39–49, DIMACS, Piscataway NJ, January 1999. Available on request from DIMACS. Paper found at URL: http://www.cis.udel.edu/˜wood/papers/DIMACS 99.ps.

[41] Robert M. Williams and David Harlan Wood. Exascale computer algebra problems interconnect with molecular reactions and complexity theory. In Landweber and Lipton [64].

[42] Erik Winfree, Furong Lin, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Natur*e, 394(6693):539–545, August 1998.

[43] Qinghua Liu, Liman Wang, Anthony G. Frutos, Anne E. Condon, Robert M. Corn, and Lloyd M. Smith. DNA computing of surfaces. *Natur*e, 403:175–179, January 13 2000.

[44] J. H. M. Dassen and P. Frisco. A bibliography of molecular computation and splicing systems. HTML source: http://www.wi.LeidenUniv.nl/ pier/dna.html, BibTeX source: http://www.wi.LeidenUniv.nl/ pier/dna.bib.

[45] Steven O. Kimbrough, Garett O. Dworman, and James D. Laing. On automated discovery of models using genetic programming in game–theoretic contexts. In Jay F. Nunamaker, Jr. and Ralph H. Sprague, Jr., editors, Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences, Volume III: Information Systems: Decision Support and Knowledge-Based Systems, pages 428–438, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[46] Steven O. Kimbrough, Garett O. Dworman, and James D. Laing. On automated discovery of models using genetic programming: Bargaining in a three-agent coalitions game. *Journal of Management Information System*s, 12(3):97–125, Winter 1995-96.

[47] Steven O. Kimbrough, Garett O. Dworman, and James D. Laing. Bargaining by artificial agents in two coalition games: A study in genetic programming for electronic commerce. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Genetic Programming Conference, July 28-31, 1996, Stanford Universit*y, pages 54–62. The MIT Press, 1996.

[48] Philippe Gigure and David E. Goldberg. Population sizing for optimum sampling with genetic algorithms: A case study of the Onemax problem. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conferenc*e, pages 496–503, University of Wisconsin, Madison, Wisconsin, USA, 22-25 July 1998. Morgan Kaufmann.

[49] Kensaku Sakamoto, Hidetaka Gouzu, Ken Komiya, Daisuke Kiga, Shigeyuki Yokoyama, Takashi Yokomori, and Masami Hagiya. Molecular computation by DNA hairpin formation. *Scienc*e, 288:1223–1226, May 19 2000.

[50] Thomas B¨ ack, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computatio*n. Institute of Physics Publishing, Philadelphia, 1997.

[51] J¨ org Heitk¨ otter and David Beasley. The hitch-hiker's guide to evolutionary computation (FAQ for comp.ai.genetic). Web page at http://alife.santafe.edu/ joke/encore/www/, September 1999.

[52] Steven O. Kimbrough and Ronald M. Lee. Formal aspects of electronic commerce: Research issues and challenges. *International Journal of Electronic Commerc*e, 1(4):11–30, Summer 1997.

[53] Tino Gramß, Stephan Bornholdt, Michael Gramß, Melanie Mitchell, and Thomas Pellizzari. *Non-Standard Computatio*n. Wiley-VCH, Weinheim, 1998.

[54] J. Ackermann, B. Wlotzka, and J. S. McCaskill. *In vitro* DNA-based predator-prey system with oscillatory kinetics. *Bulletin of Mathematical Biolog*y, 60(2):329–354, March, 1999.

[55] Larry Samuelson. *Evolutionary Games and Equilibrium Selectio*n. MIT Press Series on Economic Learning and Social Evolution. The MIT Press, Cambridge, MA, 1997.

[56] Junghuei Chen, Eugene Antipov, Bertrand Lemieux, Walter Cede˜ no, and David Harlan Wood. *In vitro* selection for a Max 1s DNA genetic algorithm. In David Gifford and Erik Winfree, editors, *Preliminary Proceedings of the Fifth Annual Workshop on DNA Based Computers, June 14-16, 1999, MI*T, pages 23–37. Massachusetts Institute of Technology, 1999.

[57] Michael D. Vose. *The Simple Genetic Algorith*m. MIT Press, Cambridge, 1999.

[58] David Harlan Wood, Junghuei Chen, Eugene Antipov, Bertrand Lemieux, and Walter Cede˜ no. A design for DNA computation of the OneMax problem. *Soft Computin*g, 5(1):19–24, 2001.

[59] Robert M. Williams and David Harlan Wood. Exascale computer algebra problems interconnect with molecular reactions and complexity theory. In Landweber and Lipton [64], pages 267–275.

[60] D. J. Wu. Artificial agents for discovering business strategies for network industries. *International Journal of Electronic Commerc*e, 5(1), Fall 2000.

[61] D. J. Wu. Discovering near-optimal pricing strategies for the deregulated power marketplace using genetic algorithms. *Decision Support Systms*, 27(1–2):25–45, 1999.

[62] Anne Condon and Grzegorz Rozenberg, editors. *DNA6 Sixth International Meeting on DNA Based Computers Leiden Center for Natural Computing 13-17 June 200*0. LCNC, 2000.

[63] Lila Kari, Harvey Rubin, and David Harlan Wood, editors. *Proceedings of the Fourth Interantional Meeting on DNA BASED COMPUTER*S. BioSystems, **5**2, 1999.

[64] Laura F. Landweber and Richard J. Lipton, editors. DNA Based Computers II: DIMACS Workshop, June 10-12, 1996, volume 44 of DIMACS series in discrete mathematics and theoretical computer science, Providence, 1998. American Mathematical Society.