*Journal of Biological Physics* **28:** 493–498, 2002.



# A DNA-based *in vitro* Genetic Program

J.A. ROSE[1], M. HAGIYA[2], R.J. DEATON[3] and A. SUYAMA[1,4]
[1]*Department of Computer Science, The Universityof Tokyo, Japan,*
[2]*Department of Computer Science, The University of Tokyo, Japan, e-mail: hagiya@is.s.u.-tokyo.ac.jp*
[3]*Department of Computer Science and Computer Engineering, The University of Arkansas, USA,*
[4]*Institute of Physics, The University of Tokyo, Japan*
(**Author for correspondence, e-mail: johnrose@is.s.u-tokyo.ac.jp*)

**Abstract.** In PNA-mediated Whiplash PCR (PWPCR), autonomous molecular computation is implemented by the recursive polymerase extension of a mixture of DNA hairpins. Like other methods based on exhaustive search, however, application to problem instances of realistic size is prevented by the exponential scaling of the solution space. The tendency of evolving populations to minimize the sampling of large, low fitness basins suggests that a DNA-based evolutionary approach might be an effective alternative to exhaustive search. In this work, PWPCR is modified to support the evolution of a population of finite state machines. A practical, *in vitro* algorithm for applying this architecture to evolve approximate solutions to instances of the NP-complete problem, *Hamiltonian Path* is described in detail.

**Key words:** DNA computing, genetic program, Hamiltonian path, *in vitro* evolution, PNA, Whiplash PCR

## 1. Introduction

In PNA-mediated Whiplash PCR (PWPCR) [1], a variation of the Whiplash PCR method of DNA computing [2], the autonomous, parallel execution of a set of finite state machines is implemented *in vitro* by the recursive polymerase extension of a mixture of single-stranded (ss) DNAs. Combined with the ability to generate an arbitrary pool of initially encoded ssDNAs (e.g., by the annealing biostep of Adleman's algorithm [3]), this process is capable of solving instances of a variety of problems in NP-complete [2]. Like other DNA computing methods which rely on a *generate and search* strategy, however the practical application of PWPCR to problem instances of realistic size is prevented by the exponential scaling of the size of the solution space [4, 5].

The combination of the massive parallelism inherent in DNA computing with the directed search capability of evolutionary computing [6] has been suggested by a number of researchers as an alternative to current, generate and search based methodologies [4, 5, 7, 8]. Conceptually, this approach is motivated by the tendency for evolving populations to minimize the sampling of large, low fitness basins,

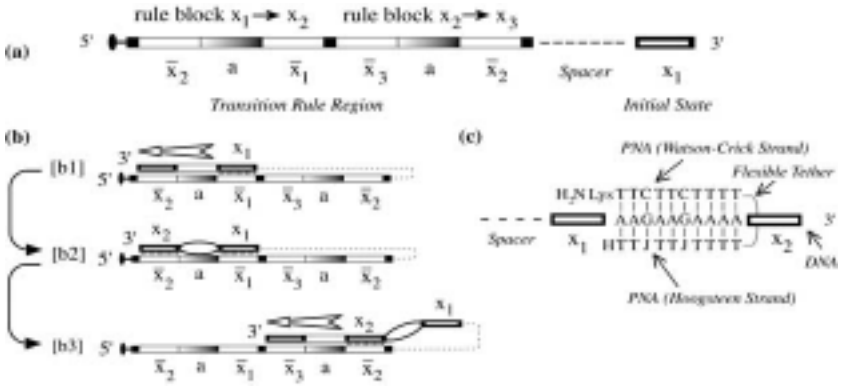


*Figure 1.* The PWPCR Architecture. (a) ssDNA encoding for computing the path $x_1 \rightarrow x_2 \rightarrow x_3$. Darkened oval denotes $5'$ biotinylation. (b) Computation by recursive extension. Vertical dashes denote hydrogen bonding prior to polymerase extension (horizontal arrows) or $PNA_2$/DNA triplex formation (light ovals). Each extension halted by stop sequence (black squares). (c) $PNA_2$/DNA triplex structure.

effectively reducing the size of the search space. Recently, a brief sketch for a DNA-based evolutionary algorithm, which uses a hypothetical mutation operator to generate population diversity (i.e., no genetic recombination is implemented) was proposed as a means of evolving approximate solutions to instances of the NP-complete problem, *Maximum Clique* [4]. In this work, PWPCR is combined with standard protocols from biotechnology to facilitate the realistic, recombination-based *in vitro* evolution of a population of simple programs, each of which implements a deterministic finite state machine (FSM). Because recombination is between the ssDNA FSM representations, rather than between computation results themselves, this implementation constitutes a Genetic Program [6], combining a limited form of programmability with the directed search capability of a standard genetic algorithm. The focus application, which is described in detail, is a realistic algorithm for applying this architecture, Evolutionary PWPCR (EWPCR) to evolve approximate solutions to instances of the NP-complete problem, *Hamiltonian Path* (HPP).

## 2. PNA-Mediated Whiplash PCR

The PWPCR protocol, shown in Figure 1 for the 2-step path, $x_1 \rightarrow x_2 \rightarrow x_3$, begins with the encoding of a ssDNA for each path in an instance graph of interest. Each strand contains a transition rule region, which encodes a rule block of the form, $5'\bar{x}_t a \bar{x}_s 3'$ for each transition $x_s \rightarrow x_t$ in the path (overscore denotes Watson-Crick reverse complementation), a $3'$ word which encodes the initial state, and a DNA spacer sequence. State transitions then proceed by hybridization of the $3'$ codeword with a complementary sequence in the transition rule region, followed by polymerase extension. Termination of polymerization at the block's $5'$ terminus

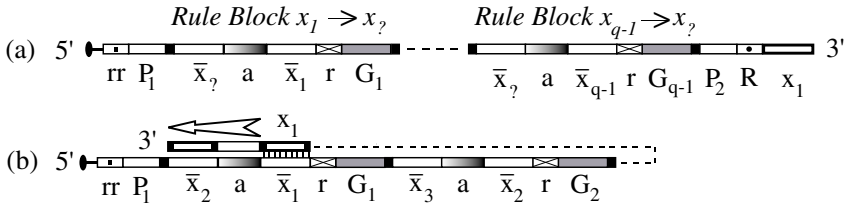


*Figure 2.* EWPCR. (a) Initial ssDNA structure. (b) Example state transition.

is implemented by a short poly-Adenine stop sequence combined with the absence of free dTTP in the buffer. In Figure 1, transitions $x_1 \rightarrow x_2$ and $x_2 \rightarrow x_3$ are shown in panels (b1) and (b3), respectively. The advance made by PWPCR is the addition of a protocol for reducing the occupancy of ssDNAs in the set of previously extended hairpin structures (i.e., backhybridization), an effect which is believed to hobble the efficiency of WPCR [1]. This protocol consists of the generation of codeword $\bar{a}$ during each extension, combined with the formation of a $PNA_2$/DNA triplex (Figure 1, panel (c)) at $\bar{a}$ by a subsequent bisPNA treatment (panel (b2)). Triplex formation is predicted to increase the efficiency of the next extension process by roughly three orders of magnitude, an effect which increases overall efficiency sufficiently for the practical implementation of massively parallel computation [1].

## 3. A PWPCR-based *in vitro* Genetic Program for HPP

### 3.1. REPRESENTATION

HPP asks the question, 'given a directed graph I, with distinguished vertices $V_{in}$ and $V_{out}$, is there a path in I which begins at $V_{in}$, ends at $V_{out}$, and visits each vertex exactly once?'. The ssDNA representation of an EWPCR-based program for computing a candidate Hamiltonian path from a $q$-vertex HPP instance of interest, $I$ is shown in Figure 2(a). Each vertex, $V_i \in I$ is represented by a unique computational state, $x_i, i \in \{1, \ldots, q\}$ where $x_1$ and $x_q$ represent vertices $V_{in}$ and $V_{out}$, respectively. Each of the strand's $q - 1$ encoded transition rules, $x_i \rightarrow x_?$, implements a randomly selected edge from $I$, subject to the constraint that the $i^{th}$ block (from the 5′ end of the strand) encode a transition from state $x_i$. Here, the state symbol, $x_?$ is used to indicate strand-by-strand variation of the target state encoded by each rule block, $i$. This encoding guarantees deterministic computation (i.e., no forked paths), and ensures that any encoded path which visits all vertices is also Hamiltonian. State transitions proceed as in PWPCR, as shown in Figure 2(b) for the transition, $x_1 \rightarrow x_2$. Each rule block, $i$ also contains codewords, $r$ and $G_i$, which enable targeted restriction site formation within the block. $r$ encodes a restriction site for *BseDI* (C↓CTTGG) [9], while $G_i$ encodes both the block's initial state and relative order from the 5′ end of the strand. The transition rule region is flanked by words $P_1$ and $P_2$, which facilitate PCR, and words $rr$ and $R$, which

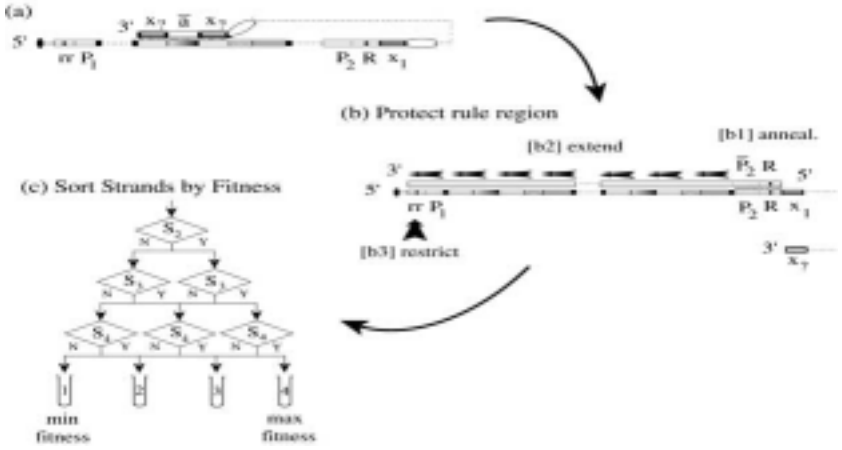


*Figure 3.* Protocol for the parallel evaluation of strand fitness.

encode restriction sites for *SnaBI* (TAC↓GTA) and *EcoRV* (GAT↓ATC) [10], respectively, to facilitate 5′ de-biotinylation and 3′ tail detachment. Each strand is 3′ terminated by initial codeword, $x_1$.

### 3.2. INITIALIZATION, FITNESS EVALUATION AND SELECTION

Initialization for HPP instance $I$ consists of the preparation of a combinatorial, incomplete mixture of biotinylated, streptavidin-conjugated ssDNAs, each of which encodes a PWPCR program for the production of a 3′ ssDNA tail, which encodes a path in $I$. Parallel strand assembly may be accomplished using the annealing biostep of Adleman's algorithm [3]. The EWPCR protocol consists of $g_{max}$ iterations of (1) fitness evaluation, (2) selection, (3) recombination, and (4) re-initialization.

Fitness evaluation begins with the production of a 3′ tail for each ssDNA, by $q - 1$ rounds of recursive extension, after which strands have the form shown in Figure 3(a). Here, $x_?$ denotes state codewords which vary strand-by-strand. Rule regions are then converted to dsDNA by polymerase extension of the primer $R\bar{P}_2$, in a buffer containing all 4 dNTPs (Figure 3(b)). This eliminates hairpin structure, prevents unwanted annealing, and generates a restriction site at $rr$. After strand release by restriction at $rr$, fitness evaluation, shown in Figure 3(c) for a 4-vertex HPP instance, is accomplished by a cascaded set of $q - 1$ affinity separations [3], $\{S_i; i = 2, \ldots, q\}$, where $S_i$ parses an input strand set into two groups, based on affinity for $\bar{x}_i$, the reverse complement of the DNA word for state $i$. A microflow architecture for performing a cascade of affinity separations has recently been reported [11]. The net result is the sorting of strands by fitness into $q$ tubes, $\{T_f; f = 1, \ldots, q\}$ where strand fitness, $f$ is defined as the number of distinct codewords in the 3′ tail, as determined by the number of successful separations.

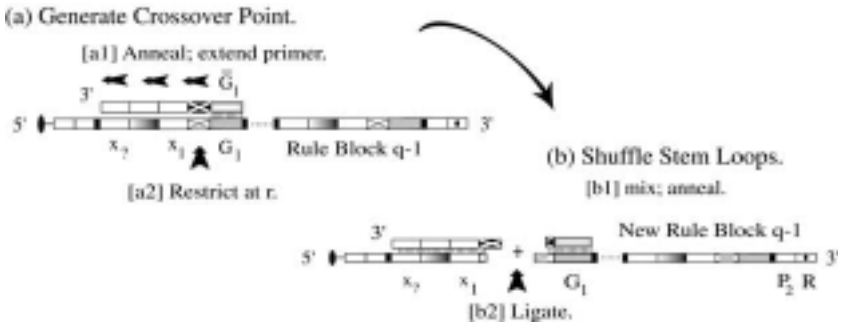


*Figure 4.* Protocol for strand recombination, based on the coordinated formation of a restriction site at rule block $i$. Implementation shown targets block $i = 1$.

The strand concentration, $C_f$ of each tube $T_f$ is then assessed. If $T_q$ is nonempty, or if $T_q$ is empty and $g_{max}$ has been reached, computation halts, and $I$ is assigned an answer of yes or no, respectively.

The fitness-squared-proportional selection process begins with restriction of all strands at $R$, followed by repeated affinity separation of each tube on word, $x_1$ to remove the $3'$ tails. Each tube $T_f$ is then PCR amplified, using primers $\bullet$-rrP$_1$ and $R\bar{P}_2$ ($\bullet$ denotes biotinylation), to ensure that the strand number in $T_f$ satisfies the condition, $n_f \geq n_o f^2 b$, where $n_o$ is the total population size, and $b \equiv \Sigma_f f^2$. Concentrations are then measured, and an extract of volume $V_f = n_o f^2 / C_f b$ is removed from each tube $T_f$, and merged to form new tube $T_s$. Strands are then reanchored, denatured, and washed.

### 3.3. GENETIC RECOMBINATION AND REINITIATION

Recombination begins by parsing $T_s$ into tubes, $T_c$ and $T_{c'}$, with volumes $V_c = p_c V_s$ and $V_{c'} = V_s - V_c$, respectively, where $p_c$ denotes the crossover probability and $V_s$ is the volume of $T_s$. Strands in $T_{c'}$ bypass recombination. $T_c$ is parsed into $q - 1$ tubes of equal volume, $\{T_j;\ j = 1, \ldots, q-1\}$. Each tube, $T_j$ is then subjected to an *in vitro* recombination procedure, which implements a uniform-length, singlepoint, two-parent crossover operation at rule block $i = j$ (shown in Figure 4 for tube $T_1$). First, the targeted formation of a crossover point in rule block $i = j$ (Panel (a)) is implemented by the addition of excess primer $\bar{G}_j$, followed by primer extension (horizontal arrows) with polymerization stop, and restriction at $r$ (vertical arrow). Restricted fragments (Panel (b)) are then mixed, allowed to re-anneal, and ligated [10]. Strands are then denatured, washed, and merged with tube $T_{c'}$. Following recombination, each strand is prepared for the next round of EWPCR by appending initial state codeword, $x_1$ to the $3'$ terminus, by the annealing of primer $\bar{X}$, $R\bar{P}_2$, followed by extension with polymerization stop.

### 3.4. GENERALIZATION

Although the goal of this work is the development of a DNA-based algorithm for evolving approximate solutions to instances of HPP, EWPCR is clearly generalizable, particularly to other problems in NP-complete.

## Acknowledgements

Financial support provided by a JSPS Postdoctoral Fellowship for Foreign Researchers, JST-Crest, and NSF grant EIA-0130385.

## References

1. Rose, J. et al.: Equilibrium analysis of an autonomous molecular computer, *Physical Review E* **65** (2002), Article 021910, pp. 1–13.
2. Sakamoto, K. et al.: State transitions by molecules, *Biosystems* **52** (1999), 81–91.
3. Adleman, L.: Molecular Computation of Solutions to Combinatorial Problems, *Science* **266** (1994), 1021–1024.
4. Bäck, T., Kok, J. and Rozenberg, G.: Cross-Fertilization between Evolutionary Computation and DNA-based Computing, In: *Proc. Cong. Evol. Comp.*, IEEE Press, Washington, DC, 1999, 980–987.
5. Chen, J. and Wood, D.H.: Computation with Biomolecules, *PNAS* **97** (2000), 1328–1330.
6. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs,* 3rd ed., Springer-Verlag, Berlin, 1996.
7. Deaton, R. et al.: A DNA based Implementation of an Evolutionary Search for Good Encodings for DNA Computation, In: *Proc. 1997 IEEE ICEC,* IEEE Press, Indianapolis, 1997, pp. 267–271.
8. Wood, D. et al.: DNA Starts to Learn Poker, in [12], pp. 23–32.
9. REBase restriction enzyme database, http://rebase.neb.com/.
10. Sambrook, J., Fritsch, E. and Maniatis, T: *Molecular Cloning: A Laboratory Manual,* 2nd ed., Cold Spring Harbor Press, 1992.
11. van Noort, D., Gast, F.-U. and McCaskill, J.: DNA Computing in Microreactors, in [12], pp. 128–137.
12. Jonaska, N. and Seeman, N. (eds.), *Prelim. Proc. of the 7th Int'l Meeting on DNA-based Computers,* Univ. of S. Florida, Tampa, 2001.