Simulating Chemical Evolution

In Soo Oh Computer Science and Engineering
Seoul National University, Korea
Email: erdos.cs@gmail.com Yun-Geun Lee Computer Science and Engineering
Seoul National University, Korea
Email: ey9ey9@gmail.com RI (Bob) McKay Computer Science and Engineering
Seoul National University, Korea
Email: rimsnucse@gmail.com

***Abstract*—Chemical methods such as directed evolution and some forms of the SELEX procedure implement evolutionary algorithms directly in vitro. They have a wide range of applications in detecting and targeting diseases and potential applications in other areas as well [1].**

However it is relatively difficult and expensive to carry out these processes (by comparison with evolutionary computation), so that the underlying theory has seen limited development. For more complex problems, where multiple and dynamic objectives are involved, there is potential for substantial improvement in the search protocols. Simulation through the methods of evolutionary computation is one potential way to gain the necessary insights.

The complex fitness functions and huge populations involved in combinatorial chemistry render detailed simulation infeasible. However detailed simulation is not needed, so long as simulations are sufficiently similar to yield qualitative insights. In this paper, we investigate whether one class of problems – those involving short-chain evolution, where stereochemical effects do not dominate – are likely to have sufficiently similar fitness landscapes to a simple problem, string matching, for useful inferences to be made. In the outcome, it appears that the differences between more detailed simulations and string matching are not sufficient to significantly alter the behaviour of evolutionary algorithms, so that string matching could be used as a realistic surrogate. This is valuable, because string matching can be implemented in GPUs, offering speed-ups to the level where populations of 10^7, or even 10^8, might be feasible, thus reducing the population gap between chemical and computer evolution.

***Index Terms*—SELEX, Directed Evolution, *NK* Model, Genetic Algorithm.**

I. Introduction

DNA-based combinatorial chemistry is rapidly developing as an important source of active organic molecules, especially in medicine, with drugs already on the market [2], and others on the way. The two major variants, Systematic Evolution of Ligands by Exponential Enrichment (SELEX [3]) and directed evolution [4], use large populations of randomised DNA molecules as genotype, with automated selection of more useful molecules (usually, based on binding energy to a target protein). In the former, the phenotype, also known as an aptamer, is the DNA itself, or more commonly, the transcribed RNA; in the latter, the phenotype is usually the resulting translated peptide or protein. In the early days of SELEX, aptamers were generally short – generally less than 25 bases – so that typical experimental sample sizes of up to 10^{15} molecules could be expected to contain most of the $4^{25} \approx 10^{15}$ possible combinations. In this case, chemical processes only needed to amplify and purify the desired molecules.

In protein synthesis, 25 bases can only generate peptide chains up to length 8, too short to be useful. So right from the start, variation operators (generally mutation, but often combined with some form of recombination) were used. Today, longer aptamers are often sought, so that some form of mutation is often combined into SELEX protocols as well. In these applications, chemists are in effect performing evolutionary computation (albeit with vastly larger populations than usual) in test tubes.

It has taken a long time, and many millions of experiments, to reach a reasonable understanding of the performance of evolutionary algorithms. Combinatorial chemists don't have this luxury. Each trial is expensive, both in human and equipment time, and in materials. In the early days, the work focused on simple optimisation, so this may not have mattered unduly. Today, there may be multiple conflicting or time varying objectives in a co-evolutionary context. Moreover combinatorial chemists suffer additional algorithmic complications – rank-based methods are unavailable, mutation and crossover operators may be biased in unknown ways, and so on.

Our aim in this program of work is to simulate this chemical evolution to the extent that we feasibly can – qualitatively, certainly not quantitatively – in the hope that this can help to elucidate the constraints on the performance of chemical evolution, and provide some guidance as to how to generate effective evolutionary protocols. This paper is a first step in determining suitable simulations.

In section II, we provide more detail of the motivation of this work, following it in section III with background on the relationship between chemical evolution and evolutionary computation. Section IV details the energy models we used, section V details the experiments, and section VI provides the results. Analysis follows in section VII, and we conclude the paper with a summary, discussion of assumptions and limitations, and proposals for future work.

II. Motivation

Why is simulation of chemical evolution needed? After all, if evolutionary optimisation is by now relatively well-

This is a self-archived copy of the accepted paper, self-archived under IEEE policy. The authoritative, published version can be found at http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5949958&tag=1

understood, can we not just directly apply its principles to chemical evolution (modulo some adaptation for the different scale of chemical evolution)?

There are two main reasons why it is not so simple:

- Chemical evolution is subject to restrictions that don't apply in evolutionary computation, and new understanding may be necessary for these.
- Chemical evolution is moving beyond static optimisation of a single objective into areas that are still under active investigation in evolutionary computation.

For the former, we note that rank-based selection (e.g. tournament selection) is infeasible to implement; the selection is inherently probabilistic, but has a sigmoid profile differing from both the linear profile of roulette selection, and the step profile of truncation selection. Even more important, mutation is inherently biased. Even getting a reasonably even probability of generating each of the four bases through mutation requires the combination of multiple polymerases under carefully controlled conditions. But this does not remove probabilistic dependency on the base which is being substituted, much less on its neighbours. It is simply not known, at present, what effect these biases may have on evolution.

One of the major targets of SELEX and directed evolution is pharmaceuticals. One approach targets proteins of disease vectors such as HIV. Thus the aim is to transport toxins into cells, in inactivated form, bound to an evolved aptamer or peptide which is able to recognise an HIV protein; when the HIV protein is bound, the toxin is released and the cell killed. However this requires not only high affinity for the HIV target, but also high selectivity for it. Even in the severely ill, HIV-infected cells are sufficiently rare that positive:negative binding ratios as high as 10^6 would probably kill the patient before the infection. It is well-known that these two objectives – affinity and selectivity – are likely to be in conflict [5], raising all the complex issues of multi-objective optimisation.

Equally important, in personalised medicine, molecules are evolved to match diseases of specific patient. In some cases (HIV, tumours'[6]), the target is itself evolving rapidly under the influence of the selective pressure of the treatment, leading to issues of dynamic optimisation and co-evolution.

All these considerations imply that there is a need for better understanding of the behaviour of algorithms implicitly used in chemical evolution. In some cases (multi-objective or dynamic optimisation, for example) better protocols might result. In others (biased mutation), it would be useful to know whether there would be major benefit from research efforts to reduce the bias: perhaps the bias has little effect on search effectiveness. The cost of each experiment makes it unlikely that sufficient studies to gain this understanding could ever be performed in chemistry. Thus qualitative simulation of these algorithms is the best solution we are likely to get.

III. Background

A. SELEX and Directed Evolution as Evolutionary Algorithms

In the simplest form of SELEX, where the aim is to find a DNA molecule with high binding affinity for a specific protein, the process itself is conceptually simple. In one form, the target protein is bonded to a surface, then washed with the initial population at a suitable temperature, so that only the more strongly-binding DNA molecules are bound. The remainder of the solution is washed away, then the bound DNA is amplified through polymerase chain reaction (PCR). The cycle is repeated a number of times – generally in the low tens with the reaction temperature being gradually increased (increasing the binding energy required for selection).

More typical forms of SELEX search for RNA, so that steps of transcription and reverse transcription have to be interleaved into the process. Directed evolution creates even greater complications, requiring RNA-protein translation in the forward direction. Since reverse translation is impractical, complex protocols are needed to ensure that the protein product's binding energy can be used to select the corresponding DNA for the next round of amplification. Further detail on how this is accomplished may be found in [7].

Evolutionary search, rather than simple concentration of an already-present molecule, uses error-prone PCR, in which the normally high-fidelity PCR (error rates as low as $4 * 10^{-7}$ mutations per base per cycle) are deliberately disrupted by chemical and other manipulation, as high as 10^{-2} [8]. In directed evolution, this may be combined with gene shuffling to induce recombinations between DNA strands.

Noisy SELEX, and directed evolution, thus really do perform evolutionary search in a search space of possible genetic combinations, using binding energy to define the fitness landscape. They differ most obviously from typical evolutionary algorithms in their relatively short runs, and in the huge populations used (typically 10^9 up to 10^{15}).

B. Simulating Chemical Evolution

If we are to simulate these processes, there are two obvious difficulties:

- We cannot hope to simulate specific RNA-protein or protein-protein binding energies accurately (even simulating the 3-D conformation of the protein binding target is known to be NP-Hard [9],).
- Even if we could solve this, we have no realistic possibility of simulating populations of 10^{15}, because of both space and time limits. Even populations of 10^9 are probably beyond the limits of current technology.

Fortunately, we don't need to solve these. Our aim is not to simulate specific runs of chemical evolution, but rather to gain a qualitative understanding. For the first difficulty, what we need is to find fitness landscapes reasonably analogous to binding energy, which we can use for our studies. For the latter, we have little alternative but to undertake scaling studies, and hope to extrapolate our results into the region of interest.

But what are reasonable analogues for RNA-protein and protein-protein binding? Since the target is fixed there is, in fact, a highest-binding-energy phenotype, we just have to find it. At one extreme, we may treat the problem as one of string matching: matching the evolved phenotype location-by-location with this fittest phenotype. String matching has many advantages as a test problem, the most notable being that it is readily implemented in a GPU, and thus we might realistically

be able to simulate quite large populations, perhaps close to the lower bound of real chemical evolution.

But string matching is also a gross over-simplification. In fact, there is epistasis between genes. Naturally, there are local interactions – substituting a single base is likely to disturb not only its own binding, but that of its neighbours. But there are also longer-distance effects, because a substitution in one location of a molecule may change the molecule's overall shape, and thus may change which bases are available for binding to the target. This led Kauffman [10] to argue that protein binding was an example of his famous $N-k$ fitness landscape, in which each of N bases interacts with a randomly selected subset of k other bases. Unfortunately $N-k$ evaluation is quite expensive, and not readily implemented on GPUs, so that it is unlikely that we would be able to scale simulations up to anything approaching the same size as for string matching. It is also likely that the random nature of the epistasis is unrealistically complex for some protein-protein binding, and especially so for RNA-protein binding, where there is chemical evidence that in at least some systems, most interactions are local [11].

The true fitness landscape is likely to lie between these extremes, in some cases closer to the string-matching end; in others, closer to $N-k$ fitness landscapes. At the simpler end, though, string-matching will always be an over-simplification – there will always be at least local energy interaction effects. Our question, in this paper, is do local interactions matter? Do they affect the fitness landscape enough to make a substantial difference to the conclusions we reach? This is important, because while pure string-matching will be relatively easy to implement in a fast GPU algorithm, even local interactions would greatly complicate the coding, so that our simulation scales would be reduced by up to an order of magnitude.

C. Previous Simulation of Chemical Evolution

Computer and mathematical simulation of non-mutational SELEX has a long tradition, and indeed non-mutational SELEX may now be considered well-understood. Corne et al. [12] appear to be the only researchers to previously simulate directed evolution; their focus was on the effects of mutation and selection pressure, and led to new insights into appropriate parameter setting in directed evolution.

IV. Binding Energy Model

In chemical evolution, the binding energy between phenotype and target generally determines the fitness (to be precise, binding affinity depends both on binding energy and on entropy, but with the relatively simple phenotypes used in directed evolution, energy generally dominates). For a length L sequence S, the binding energy in an equilibrium reaction: $S + TP \rightleftharpoons S\text{-}TP$ idepends on the sequence $S = s_1 s_2 \cdots s_L$ where $s_i \in \{A,C,G,T\}$ (DNA), $s_i \in \{A,C,G,U\}$ (RNA), $s_i \in \{A,R,N,D,C,E,Q,G,H,I,L,K,M,F,P,S,T,W,Y,V,U,O\}$ (peptide/protein) and TP is target protein. The binding energy $E(S)$ consists of two parts, [13] the specific binding energy ϵ_s and the nonspecific binding energy ϵ_0:

$$E(S) = \epsilon_s + \epsilon_0 \simeq \epsilon_s. \tag{1}$$

ϵ_s is determined by the binding details of the sequence S to TP, and thus depends on the specific sequence of nucleotides s_i in S. ϵ_0 is independent of the sequence, and represents the contribution of the Coulomb interaction to the RNA-protein-binding affinity. At least for RNA- and DNA-protein interactions, in equation (1), the non-specific interactions energy, ϵ_0 is typically several orders of magnitude smaller than ϵ_s [13], and in any case is a constant for any sequence of L bases of the same length. Thus for simplicity, we ignore it and assume that approximation (1) holds.

It is known that at least for some systems where it has been measured (the Mnt-repressor system [11]), it is a good approximation to assume that each nucleotide in the sequence contributes to the specific binding energy ϵ_s independently:

$$\epsilon_s = \sum_{i=1}^{L} \epsilon_{s_i}, \tag{2}$$

where ϵ_{s_i} is the energy contribution of the nucleotide s_i in the S sequence. Practically, the binding energy ϵ_s cannot be determined directly from experiments. However, if we arbitrarily choose a sequence $S^* = s_1^* s_2^* \cdots s_L^*$ as reference, then the discrepancy in binding energy of S^* from any sequence S, which we can treat as F_s, the fitness of the sequence S is:

$$\begin{aligned} F_s &\equiv \epsilon_{s^*} - \epsilon_s & (3)\\ &= \sum_{i=1}^{L} \epsilon_{s_i^*} - \sum_{i=1}^{L} \epsilon_{s_i} & (4)\\ &= \sum_{i=1}^{L} (\epsilon_{s_i^*} - \epsilon_{s_i}) = \sum_{i=1}^{L} f_i & (5) \end{aligned}$$

In this case, $f_i \equiv \epsilon_{s_i^*} - \epsilon_{s_i}$ can be measured experimentally by using point mutations [14].

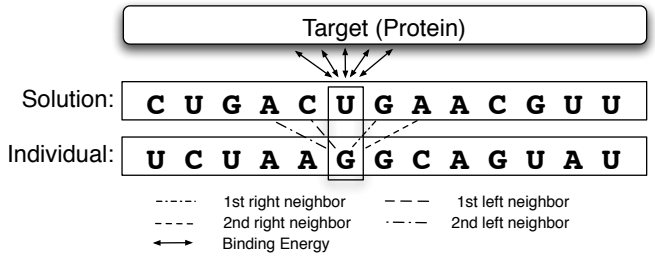

Fig. 1. Energy Model Example (sequence length $L = 13$ with optimal binding sequence (*Aptamer*) $S^* =$ `CUGACUGAACGUU` and current sample sequence $S =$ `UCUAAGGCAGUAU`).

We can further decompose the the binding energy, ϵ_{s_i} between a base s_i in a sequence S and its binding site in the target protein through equation 6:

$$\begin{aligned} \epsilon_{s_i} &= c_0(i) + r_1(i) + l_1(i) + r_2(i) + l_2(i) + \cdots & (6)\\ &= c_0(i) + \sum_{j=1}^{L} (r_j(i) + l_j(i)) \\ &\simeq c_0(i) + \sum_{j=1}^{2} (r_j(i) + l_j(i)) & (7) \end{aligned}$$

where $c_0(i)$ is the binding energy between base s_i and most adjacent amino acid residue in the target and $r_j(i)$ (or $l_j(i)$) is the binding energy between base s_i and j th adjacent amino acid residue in the target to the right (or left) direction (see the arrow $\leftrightarrow$ in the Figure 1). If we assume that we can ignore effects on binding energy of neighbours further than two sequence locations away, then we can use the approximated equation 7. Combining equations 5 and‘7, we get equation 8:

$$\begin{aligned} F_s &= \sum_{i=1}^{L} \left\{ c_0^*(i) + \sum_{j=1}^{2} (r_j^*(i) + l_j^*(i)) \right. \\ &\quad \left. - c_0(i) - \sum_{j=1}^{2} (r_j(i) + l_j(i)) \right\} \\ &= \sum_{i=1}^{L} (c_0^*(i) - c_0(i)) + \sum_{i=1}^{L} \left\{ \sum_{j=1}^{2} (r_j^*(i) - r_j(i)) \right\} \\ &\quad + \sum_{i=1}^{L} \left\{ \sum_{j=1}^{2} (l_j^*(i) - l_j(i)) \right\} \end{aligned} \tag{8}$$

If we adapt von Hippel and Berg's *two-state model* [15], which assigns an energy difference, α to each nucleotide s_i which does not match the s_i^* in the optimal binding sequence, and norrmalise α as 1, then the directed interaction effect term in equation 8 can be rewritten as equation 9:

$$c_0^*(i) - c_0(i) = 1 - \delta_{s_i s_i^*} \tag{9}$$

where δ is the Kronecker delta function.

Combining equations 8 and 9, we finally obtain the fitness function F_s of equation 10:

$$\begin{aligned} F_s &= \sum_{i=1}^{L} \underbrace{(1 - \delta_{s_i s_i^*})}_{\text{direct interaction effect}} + \\ &\quad \underbrace{\sum_{i=1}^{L} \left\{ \sum_{j=1}^{2} (r_j^*(i) - r_j(i)) \right\}}_{\text{right neighbour effect}} + \\ &\quad \underbrace{\sum_{i=1}^{L} \left\{ \sum_{j=1}^{2} (l_j^*(i) - l_j(i)) \right\}}_{\text{left neighbhour effect}}. \end{aligned} \tag{10}$$

With these simplifications, all that is now needed is to find suitable values for the neighbour interaction effects in equation 10. In principle, we could take these directly from measured values. In practice, there are few such measurements available even for the direct interaction effect, and (to the best of our knowledge) none for neighbour interaction effects. Instead, we followed Levitan and Kauffman [16] in treating them as Gaussian noise. For example, for the first neighbour to the right, we generated a probability table indexed by the three values s_i^*, s_i and s_{i+1}^*, by sampling from a Gaussian with mean 0 and standard deviation σ_1. For the second neighbour, we instead indexed by s_i^*, s_i and s_{i+2}^*, and sampled from a Gaussian with standard deviation σ_2. We made two important simplifications. We used the same table for all positions i, and we also assumed that the table for l_j was the same as for r_j. While these two simplifications will impose symmetries on the solution space, our mutation-only algorithm would not be able to make use of them, so these symmetries are unlikely to affect results (this may need to be revisited for algorithms incorporating recombination, since such algorithms might be able to take advantage of the symmetries). There is a further complication. If we follow the same procedure for the case $s_i^* = s_i$, then the fitness of the intended optimal solution will not, in general, be zero (in principle, it might not even be the optimum). To forestall this, in the case where $s_i^* = s_i$, we set the neighbour effects to zero.

TABLE I
A TYPICAL FIRST NEIGHBOUR ENERGY MATRIX (GENERATED BY SAMPLING FROM $\mathcal{N}(0, 0.1^2)$).

$s_i^* = \mathtt{A}$	$s_{i\pm1}^* = \mathtt{A}$	$s_{i\pm1}^* = \mathtt{G}$	$s_{i\pm1}^* = \mathtt{C}$	$s_{i\pm1}^* = \mathtt{U}$
$s_i = \mathtt{G}$	-0.104048	0.211441	0.231371	0.136227
$s_i = \mathtt{C}$	-0.181555	0.458834	0.308115	-0.248519
$s_i = \mathtt{U}$	-0.268548	-0.008879	-0.191869	0.161160
$s_i^* = \mathtt{G}$	$s_{i\pm1}^* = \mathtt{A}$	$s_{i\pm1}^* = \mathtt{G}$	$s_{i\pm1}^* = \mathtt{C}$	$s_{i\pm1}^* = \mathtt{U}$
$s_i = \mathtt{A}$	-0.354902	-0.061527	0.311967	0.187853
$s_i = \mathtt{C}$	0.392159	-0.114302	0.117499	-0.132544
$s_i = \mathtt{U}$	-0.082219	0.299967	-0.177403	0.252034
$s_i^* = \mathtt{C}$	$s_{i\pm1}^* = \mathtt{A}$	$s_{i\pm1}^* = \mathtt{G}$	$s_{i\pm1}^* = \mathtt{C}$	$s_{i\pm1}^* = \mathtt{U}$
$s_i = \mathtt{A}$	-0.418617	0.418001	0.008278	-0.198901
$s_i = \mathtt{G}$	0.055708	-0.102984	-0.304718	0.224832
$s_i = \mathtt{U}$	0.214901	-0.210827	0.200560	-0.269409
$s_i^* = \mathtt{U}$	$s_{i\pm1}^* = \mathtt{A}$	$s_{i\pm1}^* = \mathtt{G}$	$s_{i\pm1}^* = \mathtt{C}$	$s_{i\pm1}^* = \mathtt{U}$
$s_i = \mathtt{A}$	-0.120875	0.011116	0.236024	-0.078499
$s_i = \mathtt{G}$	-0.210954	-0.042641	0.033997	-0.220864
$s_i = \mathtt{C}$	0.043810	0.315074	-0.076174	-0.107441

Of course, we needed to choose suitable values for the standard deviations. We can generally assume that interaction effects will be smaller than direct effects, so we used $\sigma_1 = 0.1$, $\sigma_2 = 0.01$. Typical energy tables may be seen in tables I and II.

As an example of their application, consider the boxed nucleotides in figure 1. Since the nucleotides do not match, for the direct interaction effect we have $1 - \delta s_i s_i^* = 1$. For the first neighbour effect, we have $s_i = \mathtt{G}$, $s_i^* = \mathtt{U}$ and $s_{i-1}^* = \mathtt{C}$, so the value is -0.042641 according to table I. In calculating the second-left-neighbour effect, we have $s_{i-2}^* = \mathtt{A}$, so looking up table II, we find that the relevant value is 0.010910.

V. EXPERIMENTS

In this section, we detail the experiments we carried out. The main aim of our experiments was to determine whether string matching can be a useful analogue for chemical binding, when interaction effects are both largely spatial, and not too large relative to the primary bonding. As discussed, the latter is probably a reasonable model for binding by small RNA aptamers, but a quite poor one for binding of large proteins, where stereoscopic effects are far more important.

TABLE II
A TYPICAL SECOND NEIGHBOUR ENERGY MATRIX (GENERATED BY SAMPLING FROM $\mathcal{N}(0, 0.01^2)$).

$s_i^* = \mathtt{A}$	$s_{i\pm2}^* = \mathtt{A}$	$s_{i\pm2}^* = \mathtt{G}$	$s_{i\pm2}^* = \mathtt{C}$	$s_{i\pm2}^* = \mathtt{U}$
$s_i = \mathtt{G}$	-0.013801	0.041062	0.002391	0.018501
$s_i = \mathtt{C}$	0.025344	-0.019649	0.034144	0.021813
$s_i = \mathtt{U}$	-0.000799	0.010910	0.007023	0.042567
$s_i^* = \mathtt{G}$	$s_{i\pm2}^* = \mathtt{A}$	$s_{i\pm2}^* = \mathtt{G}$	$s_{i\pm2}^* = \mathtt{C}$	$s_{i\pm2}^* = \mathtt{U}$
$s_i = \mathtt{A}$	-0.016213	0.024762	-0.001358	0.042338
$s_i = \mathtt{C}$	0.014774	-0.031148	0.016136	-0.019808
$s_i = \mathtt{U}$	0.007768	0.008814	-0.027459	0.045023
$s_i^* = \mathtt{C}$	$s_{i\pm2}^* = \mathtt{A}$	$s_{i\pm2}^* = \mathtt{G}$	$s_{i\pm2}^* = \mathtt{C}$	$s_{i\pm2}^* = \mathtt{U}$
$s_i = \mathtt{A}$	0.011656	0.001185	-0.011229	0.006667
$s_i = \mathtt{G}$	-0.017614	-0.028490	-0.003105	-0.038451
$s_i = \mathtt{U}$	0.021532	0.039736	0.021032	-0.000874
$s_i^* = \mathtt{U}$	$s_{i\pm2}^* = \mathtt{A}$	$s_{i\pm2}^* = \mathtt{G}$	$s_{i\pm2}^* = \mathtt{C}$	$s_{i\pm2}^* = \mathtt{U}$
$s_i = \mathtt{A}$	-0.005360	-0.006246	-0.022683	-0.0047054
$s_i = \mathtt{G}$	0.000751	-0.016758	0.020551	-0.000128
$s_i = \mathtt{C}$	0.001281	0.034197	-0.024489	-0.015193

TABLE III
GENETIC ENVIRONMENT AND PARAMETERS

Alphabet :	A(0), G(1), C(2), U(3)
Genotype expression :	1 dimensional array S
Genotype Length(L) :	20 bp
Population size :	10^6
Maximum rounds :	60
Selection Method :	deterministic tournament with size 2
Fitness function :	Equation 10
Recombination rate :	0 (not used)
Mutation rate (per base) :	0.0001
Number of Runs:	100
Energy Models:	three randomly sampled models
Neighbour energy effect :	None/1st/1st & 2nd
Target Type :	Uniform/PeriodicRandom

A. Evolutionary Computation Environments

The evolutionary environments used in these experiments are summarised in table III. The parameter settings are a compromise between computational feasibility and similarity to settings in SELEX and directed evolution. The shortest mutable regions used in chemistry are 15 bases, so we used 15 bases as our smallest genome length. To get some idea of scaling, we also used 20 bases – realistic, but still on the low end of what might be used in chemistry. For space reasons, we only present the 20-base results here. We used the largest size population (10^6)we realistically could handle in cpu-based computing. To partially compensate for the smaller populations, we used 60 generations, substantially more than would normally be used in directed evolution, but necessary for our far smaller populations to converge to an optimum. We used a mutation rate of 10^{-4} per cycle per base, comparable to that used in SELEX [17].

Selection mechanisms based on RNA- or protein-protein binding have a complex relationship with the fitness (i.e. binding energy) function. Under the assumption that a binding site can never be occupied by more than one sequence element at a time, the binding probability of a sequence S by the target molecules has the form of the Fermi function,

$$P(S, \mu) = \frac{1}{1 + \exp\left(\frac{-E(S)-\mu}{k_B T}\right)}, \tag{11}$$

where k_B is the Boltzmann constant, μ is the chemical potential, and $E(S)$ is the binding energy of S to the target [18]. This probability function is a little too complex to implement efficiently in a 10^6 population evolutionary algorithm, but it has the form of a fairly soft selection pressure; we substituted it with the computationally far more efficient mechanism of a tournament of size 2. Recombination method is not generally used in SELEX or the simpler forms of directed evolution, so we omitted this operator from our simulation.

We used three different forms of binding energy model: neighbour-independent (i.e. effectively, string matching); dependence on only the first neighbour, and dependence on the first and second neighbours. To eliminate the possibility that results might depend on the specific randomly-sampled energy models, we repeated the experiments three times with independently Gaussian-sampled models.

We found in preliminary studies that the results depended to some degree on the target molecule, so we studied three kinds of targets:

1) Uniform targets, in which the same symbol is repeated for the length of the string
2) Periodic targets, in which a substring is repeated for the length of the string
3) Random targets, in which the whole string is randomly initialised

They are shown in detail in table IV (the code shown there is used in the legends of the figures showing our results).

TABLE IV
BINDING TARGETS FOR EVOLUTIONARY EXPERIMENTS

Type	Code	Target
Uniform	0	AAA$\cdots$A
	0a	GGG$\cdots$G
	0b	CCC$\cdots$C
	0c	UUU$\cdots$U
Periodic	1	AGCUAGCU$\cdots$AGCU
Random	3	UGCUAGAAAGCAUGCGGGGA
	3a	CGUGCGGGGAUCGCAUGCUA

B. Software and Hardware Environments

TABLE V
SOFTWARE AND HARDWARE SUMMARY

Base Library :	EO C++ Library 1.0.2 [19]
Operating System :	Linux (Kernel 2.6.26)
C++ compiler :	g++ 4.1.2 with -O3 optimization
Programming language :	C++
CPU :	Intel(R) Xeon(R) E5310 1.60GHz
Main memory :	2 Gigabyte

The software and hardware environments used in these experiments are summarised in table V. As the basic framework, we used the used ANSI-C++ compliant evolutionary computation library Evolving Objects (EO) [19].

VI. Results

Figures 2–8 show results (best-of-generation and mean fitness) averaged over 100 runs for each of the seven experimental settings (no neighbour influence, three randomly-set first-neighbour influence models, and three randomly-set first- and second-neighbour models). Each curve represents one of the seven target strings.

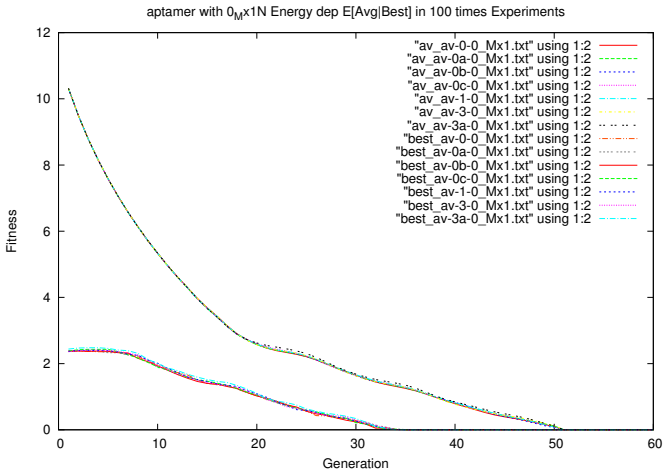

Fig. 2. Best and Average Fitnesses, no Neighbour Interaction

Figure 2 shows the fitness curves obtained with no interaction between neighbours (i.e. a pure string-matching problem). As we might anticipate, all curves are essentially coincident, because the fitness landscapes for each target are the same.

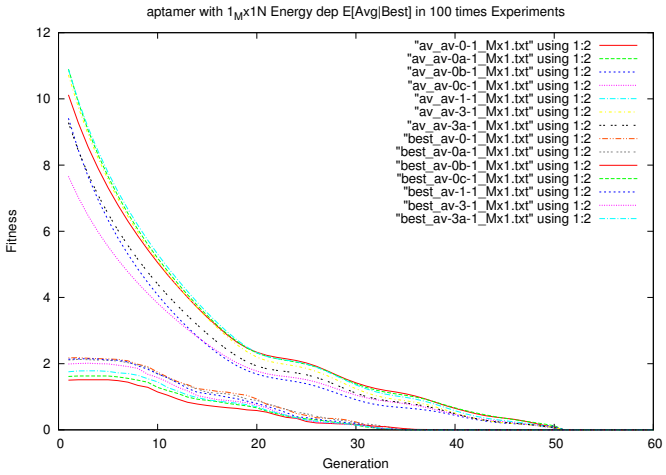

Fig. 3. Best and Average Fitnesses, Nearest Neighbour Interaction (Case 1)

Figures 3–5 show the corresponding behaviour for three different randomly-sampled energy dependence matrices, when only adjacent neighbours are taken into account. In these cases, the initial energy distributions differ depending on the targets (because with different targets, random differences in the energy matrices will slightly change the energy distributions), but these differences rapidly disappear through evolution. Convergence of the best and average fitness still occur at almost exactly the same times (33 and 50 generations respectively).

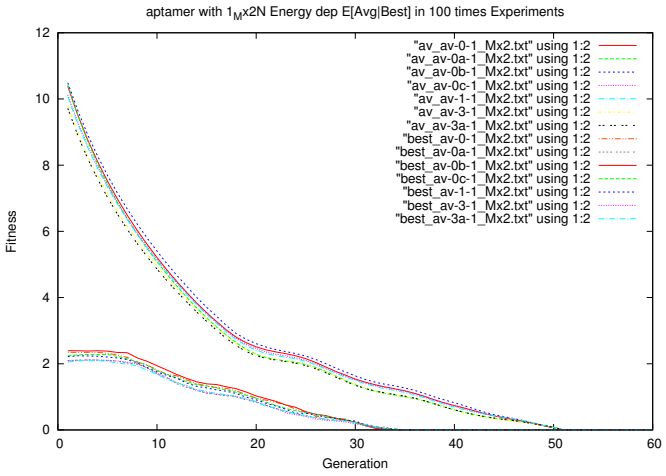

Fig. 4. Best and Average Fitnesses, Nearest Neighbour Interaction (Case 2)

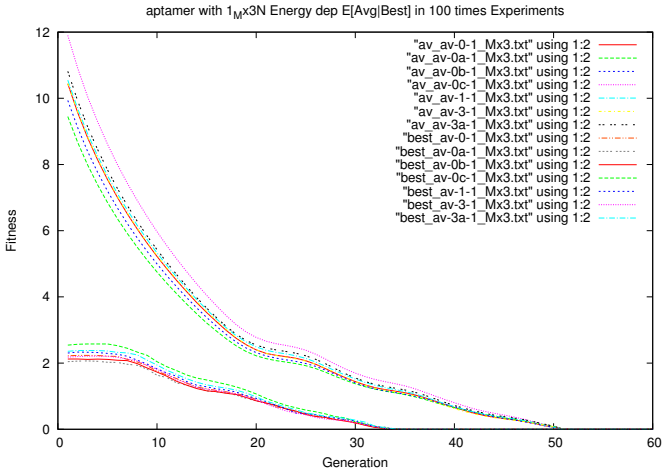

Fig. 5. Best and Average Fitnesses, Nearest Neighbour Interaction (Case 3)

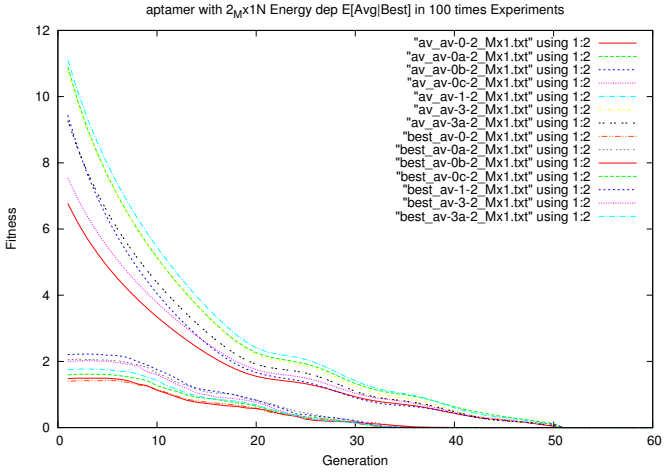

Fig. 6. Best and Average Fitnesses, Second Neighbour Interaction (Case 1)

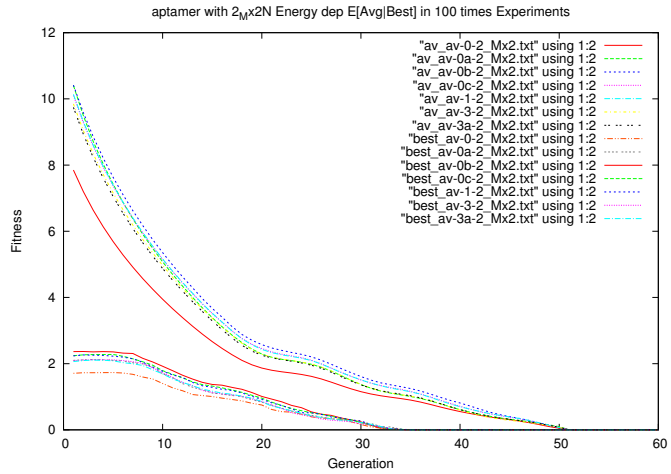

Fig. 7. Best and Average Fitnesses, Second Neighbour Interaction (Case 2)

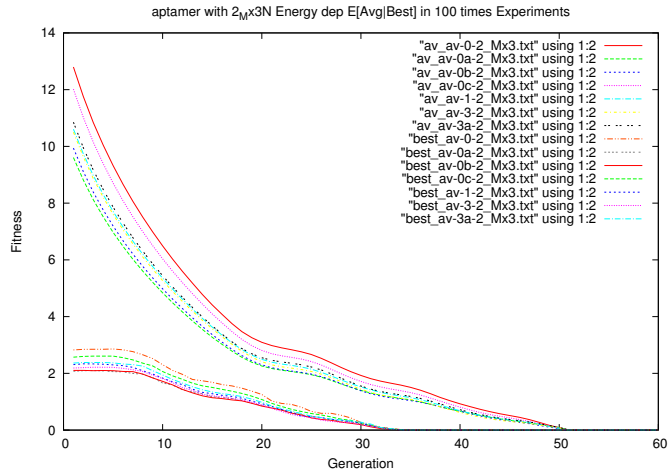

Fig. 8. Best and Average Fitnesses, Second Neighbour Interaction (Case 3

Figures 6–8 illustrate the same properties for energy models incorporating both first and second neighbour interactions. Interestingly, despite the substantially reduced scale of second neighbour interactions compared with first neighbour, the spread in initial energy levels between different targets is substantially increased. Nevertheless, there is again absolutely no effect on the time to convergence, with both best and average fitness converging at exactly the same times.

VII. Analysis

The key result of this work lies in a contrast.

When there was no dependence of binding energy on neighbours, the (normalised) distribution of binding energies was essentially independent of the target. Adding first-neighbour dependence to the binding energy created a significant dependence on the target, while incorporating second-neighbour dependences as well increased it still further. In the latter case, the mean binding energy of the initial (random) population could vary from 7 to 11 (figure 6), depending on the target (note that in the context of populations of 10^6 individuals, this could not be an artefact of sampling error).

Yet adding first and second neighbour dependences made no discernible difference to the convergence time (for either best or mean fitness). That is, changes to the shape and scale of the fitness landscapes induced by these local dependences did not change the difficulty of finding the optimum.

VIII. Conclusions

A. Summary

The primary conclusion we can draw is that for binding energy problems where there are no long-distance epistases, string matching is a reasonable surrogate: conclusions drawn from it will likely be applicable to the original chemical problem. Any local dependences have little effect on the problem difficulty. This is important, because string matching can be readily coded in libraries such as CUDA, and may be expected to yield very substantial speed-ups. Realistically, we may be able to run carefully-tuned evolutionary algorithms with populations of 10^7 or even 10^8, i.e. approaching the lower-end populations of real directed evolution.

B. Assumptions and Limitations

The key limitation of this problem is its restriction to local influences (and in particular, its assumption that the genotype components are the primary interacting components). This is a reasonable assumption for RNA/DNA aptamers. It is also a reasonable assumption for short peptides, with the one proviso, that in those cases, we also need to take into account the RNA-amino acid triplet mapping. Although it needs to be validated, we do not expect that the triplet mapping would greatly change our conclusions.

On the other hand, it is clear that for longer proteins, local interactions are not a reasonable approximation to the real fitness function. While they will be present, they will not be the dominant component of the energy function, especially far from the optimum. In these cases, $N-k$ landscapes may well be a better model. Of course the downside of N-k landscapes is the difficulty of implementing them in restricted computational models. Thus it is likely that, for N-k landscapes, we would be limited to populations of the order of the 10^6 used here.

C. Further Work

The simplest extensions of this work are further validation: testing with $N-k$ landscapes, testing whether any likely biases in mutation operators are likely to affect results, investigating the effects of the sigmoid-probability selection operators and so on.

But the most interesting directions are extensions to the kinds of problems really faced by chemists,. We noted previously that the two objectives – binding affinity and selectivity – are generally in conflict. This is especially the case in medical applications, where disease agents may well have been selected to be antigenically similar to human tissues (indeed, there is strong evidence that this mimicry is closely associated with autoimmune diseases [20]). It is simply unknown whether this antigenic similarity would elicit similar binding responses from the kinds of molecules derived through SELEX or

directed evolution. In any case, it seems desirable to find good evolutionary methods for these conflicting objectives. Current methods typically consist of evolving (or selecting) first for binding affinity, then further evolving (or selecting) for selectivity. This is unlikely to be even a halfway-good algorithm.

On the other hand, classic multi-objective algorithms are unlikely to be easy to implement in chemistry. Elite management is difficult enough, while dominance ranking is clearly infeasible. It might be possible to implement some form of noisy, quantised dominance ranking through microfluidics technology, but this would be expensive, and unlikely to be developed unless clear benefits could be demonstrated through simulation. Alternatively, thinking within the restrictions of directed evolution, it may be possible to find acceptable multi-objective algorithms that don't require these expensive operations.

Similarly, what we know about dynamic optimisation and co-evolution could be useful in personalised medicine, in re-optimising treatments as diseases such as cancers and HIV re-adapt themselves. Here, simple algorithms such as the hybrid immigrants GA [21] could be relatively easily adapted to chemistry, and might well make a significant contribution.

ACKNOWLEDGMENT

Seoul National University Institute for Computer Technology provided research facilities for this study, which was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Project No. 2010-0012546), Research Expenses for Foreign Professors funded by Seoul National University(Project No. 400-20100189), and the BK21-IT program of MEST.

We would like to thank Dr Sung Chun Kim of Genoprot Co. and Prof. Sunjoo Jeong of DanKook University for insightful discussions on the chemistry background to this paper.

REFERENCES

[1] R. Stoltenburg, C. Reinemann, and B. Strehlitz, "Selex–a (r)evolutionary method to generate high-affinity nucleic acid ligands." *Biomol Eng*, vol. 24, no. 4, pp. 381–403, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.bioeng.2007.06.001

[2] D. Vavvas and D. D'Amico, "Pegaptanib (Macugen): treating neovascular age-related macular degeneration and current role in clinical practice." *Ophthalmology clinics of North America*, vol. 19, no. 3, pp. 353–360, 2006.

[3] C. Tuerk and L. Gold, "Systematic evolution of ligands by exponential enrichment: Rna ligands to bacteriophage t4 dna polymerase," *Science*, vol. 249, no. 4968, pp. 505–510, 1990. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/249/4968/505

[4] W. Stemmer *et al.*, "Rapid evolution of a protein in vitro by DNA shuffling," *Nature*, vol. 370, no. 6488, pp. 389–391, 1994.

[5] J. Carothers, S. Oestreich, and J. Szostak, "Aptamers selected for higher-affinity binding are not more specific for the target ligand," *J. Am. Chem. Soc*, vol. 128, no. 24, pp. 7929–7937, 2006.

[6] C. Ferreira, C. Matthews, and S. Missailidis, "DNA aptamers that bind to MUC1 tumour marker: design and characterization of MUC1-binding single-stranded DNA aptamers," *Tumor Biology*, vol. 27, no. 6, pp. 289–301, 2006.

[7] Y. G. Lee, R. I. McKay, K. I. Kim, D. K. Kim, and X. H. Nguyen, "Investigating vesicular selection: A selection operator from in-vitro evolution," Seoul National University Structural Complexity Laboratory, Tech. Rep. TRSNUSC:2009:001, 2009.

[8] T. Rasila, M. Pajunen, and H. Savilahti, "Critical evaluation of random mutagenesis by error-prone polymerase chain reaction protocols, escherichia coli mutator strain, and hydroxylamine treatment," *Analytical Biochemistry*, vol. 388, no. 1, pp. 71–80, 2009.

[9] N. A. Pierce and E. Winfree, "Protein design is np-hard," *Protein Eng*, vol. 15, no. 10, pp. 779–82, Oct 2002. [Online]. Available: http://peds.oxfordjournals.org/cgi/content/full/15/10/779

[10] S. A. Kauffman and E. D. Weinberger, "The nk model of rugged fitness landscapes and its application to maturation of the immune response," *Journal of Theoretical Biology*, vol. 141, no. 2, pp. 211 – 245, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/B6WMD-4KCPS8W-6/2/6907cba4ac11f94b7f90ffec66b1ae3e

[11] G. D. Stormo, S. Strobl, M. Yoshioka, and J. S. Lee, "Specificity of the mnt protein : Independent effects of mutations at different positions in the operator," *Journal of Molecular Biology*, vol. 229, no. 4, pp. 821–826, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/B6WK7-45PV5P8-80/2/0ad1853960c6537fe3f42ecc930a9686

[12] D. Corne, M. Oates, and D. Kell, "On fitness distributions and expected fitness gain of mutation rates in parallel evolutionary algorithms," *Parallel Problem Solving from Nature —PPSN VII*, pp. 132–141, 2002.

[13] G. D. Stormo and D. S. Fields, "Specificity, free energy and information content in protein-dna interactions." *Trends Biochem Sci*, vol. 23, no. 3, pp. 109–113, 1998.

[14] D. S. Fields, Y. yuan He, A. Y. Al-Uzri, and G. D. Stormo, "Quantitative specificity of the mnt repressor," *J. Mol. Biol.*, vol. 271, no. 2, pp. 178–194, 1997.

[15] P. H. von Hippel and O. G. Berg, "On the specificity of dna-protein interactions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 83, no. 6, pp. 1608–1612, Mar 1986.

[16] B. Levitan and S. Kauffman, "Adaptive walks with noisy fitness measurements," *Molecular Diversity*, vol. 1, pp. 53–68, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF01715809

[17] K. A. Eckert and T. A. Kunkel, "High fidelity dna synthesis by the termus aquaticus dna polymerase," *Nucleic Acids Research*, vol. 18, no. 13, pp. 3729–3744, Jul 1990.

[18] Y. Yang, H. Wang, and Q. Ouyang, "Dynamics of dna in vitro evolution with mnt-repressor: Simulations and analysis," *Phys. Rev. E*, vol. 68, no. 3, p. 031903, Sep 2003. [Online]. Available: http://prola.aps.org/abstract/PRE/v68/i3/e031903

[19] M. Keijzer, J. Merelo, G. Romero, and M. Schoenauer, "Evolving objects: A general purpose evolutionary computation library," in *Artificial Evolution*, ser. Lecture Notes in Computer Science, P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton, and M. Schoenauer, Eds. Springer Berlin / Heidelberg, 2002, vol. 2310, pp. 829–888. [Online]. Available: http://www.springerlink.com/index/QPC0FDXGT3523M4R.pdf

[20] A. Ebringer and C. Wilson, "HLA molecules, bacteria and autoimmunity," *Journal of Medical Microbiology*, vol. 49, no. 4, pp. 305–311, 2000.

[21] S. Yang and R. Tinós, "A hybrid immigrants scheme for genetic algorithms in dynamic environments," *International Journal of Automation and Computing*, vol. 4, no. 3, pp. 243–254, 2007.