Active Transport in Biological Computing (Preliminary Version)

Stuart A. Kurtz Stephen R. Mahaney James S. Royer
Janos Simon

May 24, 1996

Abstract

Early papers on biological computing focussed on combinatorial and algorithmic issues, and worked with intentionally oversimplified chemical models. In this paper, we reintroduce complexity to the chemical model by considering the effect problem size has on the initial concentrations of reactants, and the effect this has in turn on the rate of production and quantity of final reaction products.

We give a sobering preliminary analyses of Adleman's technique for solving Hamiltonian path. Even on the simplest problems, the annealling phase of Adleman's technique requires time $\Omega(n^2)$ rather than the $O(\log n)$ complexity given by a computationally inspired but chemically naive analysis. On more difficult problems, not only does the rate of production of witnessing molecules drop exponentially in problem size, the final yield also drops exponentially.

These issues are not objections to biological computing *per se*, but rather difficulties to be overcome in its development as a viable technology.

1 Reaction Rates

We assume familiarity with Adleman's DNA-based technique [Adl94] for solving the hamiltonian path problem. For the present analysis, we will consider the hamiltonian path problem on directed graphs, rather than graphs. We recommend *Molecular Cell Biology* [LBB$^+$95] as a useful general reference for all chemical and biological concepts not otherwise referenced.

1.1 An Easy Case

We begin by considering a very simple family of directed graphs, the *line digraphs*.

Definition 1 *The* line digraph L_n *consists of* n*-vertices* $\{v_0, v_1, \ldots, v_{n-1}\}$ *and* $n-1$ *edges* $\{e_{0,1}, e_{1,2}, \ldots, e_{n-2,n-1}\}$*, where* $e_{i,i+1} = \langle v_i, v_{i+1}\rangle$.

The line digraphs all have hamiltonian paths, indeed, the line digraphs *are* hamiltonian paths.

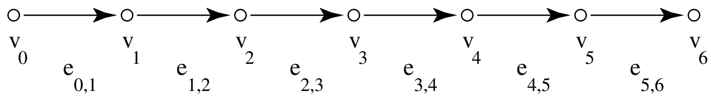

Figure 1: The line digraph L_7.

Adleman's technique begins with constructing molecules that correspond to the edges of the graph. These edge molecules undergo annealling (ligation) reactions to form molecules that represent paths through the graph. Finally, these path molecules are filtered for length and content, so that only path molecules that represent hamiltonian paths remain.

Definition 2 *A path is* useful *if it is a subpath of a hamiltonian path, otherwise it is* useless*.*

The molecules that represent useless paths are a potential problem for Adleman's technique, as demonstrated in Section 2 below. Useless molecules are a problem because they reduce the concentration of useful molecules, and therefore reduce the rate of production of witnessing molecules.

We begin our analysis with line digraphs simply because they are precisely the positive instances of hamiltonian path for which there are no useless paths.

There is a nontrivial path in L_n from v_i to v_j if and only if $i < j$, in which case the path is unique. By abuse of notation, the unique path from v_i to v_j will be denoted by $e_{i,j}$. We will also use $e_{i,j}$ to denote the molecule corresponding to this path.

1.2 Analysis

In this simple case, the ligation reactions take the form

$$e_{i,k} + e_{k,j} \to e_{i,j} \qquad \text{for } i < k < j.$$

The desired end product of these reactions is the molecule $e_{0,n-1}$.

If we ignore chemical considerations, and utilize only our computational intuitions, we would be led more or less directly to consideration of a balanced binary tree of chemical reactions, with path molecules at the leaves and the unique molecule representing a hamiltonian path as its root. Because the depth of this tree is $\lceil \log n \rceil$, we might naively expect that rate of production of the hamiltonian path molecule is $O(\log n)$.

In this section, we give a less naive analysis. Our goal will be to give lower bounds on how long it will take to produce the hamiltonian path molecule $e_{0,n-1}$ as a function of the number n. The analysis that follows proves that ligation phase requires time $\Omega(n^2)$, rather than $O(\log n)$. The main purpose behind this section is not to establish the $\Omega(n^2)$ lower bound, even though it was unexpectedly slow, but rather to introduce chemical principles and analytical techniques that demonstrate a far more dramatic loss of performance on denser digraphs.

Definition 3 *The concentration of a (family of) chemical compound(s) K is denoted by $[K]$.*

The reaction rate law (cf. [LBB+95, p. 28ff]) states that the rate at which a chemical reaction occurs is proportional to the products of the concentrations of the reactants. Thus, e.g., the reaction

$$e_{i,k} + e_{k,j} \rightarrow e_{i,j}$$

proceeds at the rate

$$\alpha_{i,j,k} \, [e_{i,k}] \, [e_{k,j}]$$

for some constant $\alpha_{i,j,k}$. For the line digraph L_n, we have $\binom{n}{2}$ different types of path molecules, and $\binom{n}{3}$ reactions to account for. For the sake of simplicity, we assume that the reaction rate constants are all equal, and that the units of time time are chosen so that $\alpha = 1$.

This gives rise to the following system of $\binom{n}{2}$ nonlinear ordinary differential equations:

$$[e_{i,j}]' = \left(\sum_{\{k:i<k<j\}} [e_{i,k}] \, [e_{k,j}] \right) - [e_{i,j}] \left(\sum_{\{l:j<l\}} [e_{j,l}] \right) - \left(\sum_{\{h:h<i\}} [e_{h,i}] \right) [e_{i,j}] \, .$$

We will assume, for the sake of simplicity, that units of volume are chosen so that the magnitude of the molar concentration of the solution is one, independent of problem size.[1] This concentration must be partitioned among the $n-1$ edge molecules $e_{i,i+1}$, and so our initial conditions are:

$$e_{i,j}(0) = \begin{cases} \frac{1}{n-1}, & \text{if } j = i+1, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

Our analysis focusses on the creation of path molecules that include a given interior vertex v.

Definition 4 *Let $\langle v|$ denote the set of molecules that represent paths that end at vertex v, likewise let $|v\rangle$ denote the set of molecules that represent paths that begin at v, and $\langle v \rangle$ denote the set of molecules that represent paths containing v as an interior vertex.*

[1] The *molar* concentration of a compound in solution is expressed in units of molecules per volume. This is somewhat unrealistic, as it does not take into account that for larger instance sizes, the edge molecules will necessarily be longer, and therefore fewer can be accommodated within a given volume.

All of the reactions of the form

$$e_{i,v} + e_{v,j} \to e_{i,j}$$

can be bundled together into a single reaction of the form

$$\langle v| + |v\rangle \to \langle v\rangle\,.$$

All other reactions can be neglected, because the conserve the concentrations of $\langle v|$ and $|v\rangle$, and so we can explicitly solve the differential equations that arise from the rate law:

$$[\langle v\rangle]\,(t) = \frac{t}{(t+n-1)(n-1)}$$

Because the hamiltonian path molecule is just one of the path molecules that together make up $\langle v\rangle$, it immediate follows that $[e_{0,n-1}] \le [\langle v\rangle]$, and from this follows the $\Omega(n^2)$ lower bound.

At this point, we do not have a rigorous analysis that gives a corresponding upper bound, but numerical solution of the system of differential equations for $n \in [10, 50]$ suggests that the time that it takes for the $[e_{0,n-1}]$ to reach a given threshold is essentially quadratic in n, cf. Figure 2.[2]

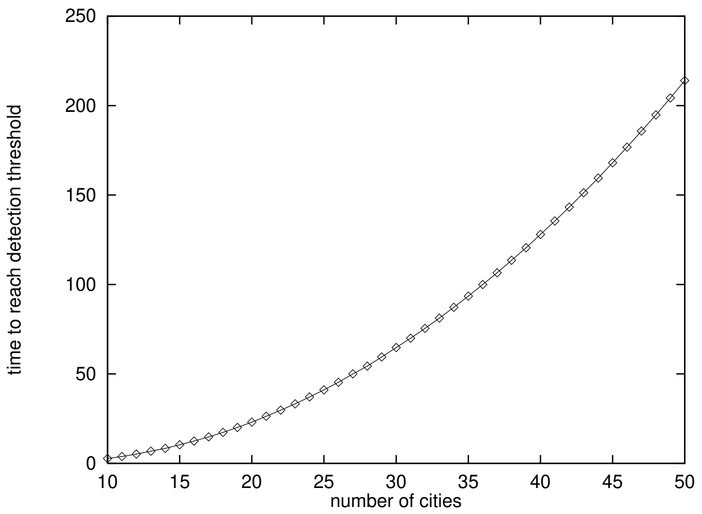

Figure 2: Time for $[e_{0,n-1}]$ to reach 1.0×10^{-6}.

[2] This plot is based on a threshold concentration of 1.0×10^{-6} in the normalized units in which the initial molar concentration was 1.0.

2 Concentration

2.1 Line Digraphs Revisited

A somewhat more complete story is told by considering the concentrations of chemicals of a given path length, as a function of time. Let $[k]$ denote the sums of the concentrations of path molecules of length k, i.e.,

$$[k] = \sum_{i=0}^{n-k-1} [e_{i,i+k}] \,.$$

Figure 3 shows how these concentrations vary as a function of time for the reactions based on the line digraph L_{30}.

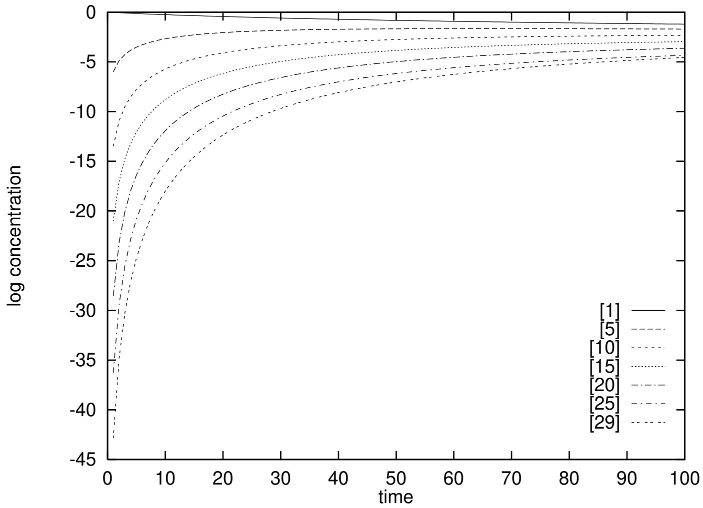

Figure 3: $\log_{10}[k]$ as a function of time for L_{30}.

Although the plot gives a good "feel" for the data, there are a few points that may not be apparent. The concentration of simple edge molecules, [1], starts at 1.0 and falls steadily during the reaction. Excepting $k = n-1$, $[k]$ reaches a maximum value in finite time, and falls slowly to zero thereafter. The successive maximums fall quadratically far apart, and because $[k+1]$ remains below $[k]$ throughout the reaction, it reachs a lower maximum. For the hamiltonian path molecule, $[n-1]$ tends in the limit to $\frac{1}{n-1}$.

2.2 A Hard Case

Having considered line digraphs, we now move on to consider a more difficult family of digraphs: the *feedback digraphs*[3]. The loss of concentration due to the presence of useless paths makes this a far more realistic example. Useless paths accumulate mass which is not recoverable, and so cannot be used to produce witnessing molecules.

Definition 5 *The* simple feedback digraph F_n *has* n *vertices and* $2n-4$ *edges. Like the line graph, it has* $n-1$ forward edges $e_{i,i+1} = \langle v_i, v_{i+1}\rangle$ *for* $i = 0 \ldots n-1$, *but it also has* $n-3$ back edges, $b_k = \langle v_k, v_1\rangle$ *for* $k = 2 \ldots n-2$. *(See Figure 4 below.)*

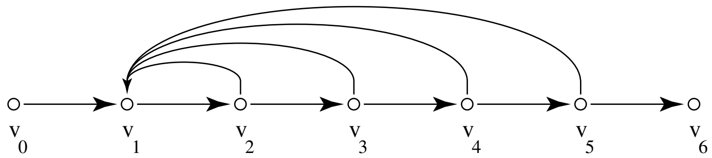

Figure 4: The simple feedback digraph F_7.

It should be clear that F_n has a unique hamiltonian path consisting of the edges $e_{i,i+1}$, for $i = 0 \ldots n-2$. Thus, any path that involves a back edge is useless, i.e., it cannot be extended to a hamiltonian path.

Definition 6 *We begin by defining* $\langle v|$, $|v\rangle$, *and* $\langle v\rangle$ *for* F_n *as in Definition 4. By extension, if* p *denotes a path through* F_n, *let* $\langle p|$ *denote the set of molecules that represent paths that end with* p, *and define* $|p\rangle$ *and* $\langle p\rangle$ *analogously.*

If we assume that all edge molecules are present in equal concentrations at the beginning of the ligation phase, then for $i = 2, 3, \ldots, n-2$, $[|e_{i,i+1}\rangle]$ equals $[|b_i\rangle]$ because they have the same initial concentration, and both satisfy the first order differential equation

$$y' = -\,[\langle v_i|]\, y$$

from the rate law.

Let $e_{i,j}$ denote the simple path molecule from v_i to v_j involving only forward edges. We have

$$[\langle e_{0,i+1}|] = [\langle e_{0,i} b_i|]\,,$$

for $i = 2, 3, \ldots n-2$, because both have the same initial value (0), and both satisfy the differential equation:

$$y' = y\,[|e_{i,i+1}\rangle] = y\,[|b_i\rangle]$$

[3]This particular family of digraphs is a standard example as to why random walk strategies fail to explore directed graphs efficiently, but is not the worst case.

By a more complicated version of essentially the same argument $[\langle e_{0,i+1}|] \leq \frac{1}{2}[\langle e_{0,i}|]$. Thus,

$$[e_{0,n-1}] \leq \frac{1}{(2n-4)2^{n-3}}$$

holds throughout the computation (recall that we are using normalized units in which all physical constants have the value one). The ligation phase will never produce the hamiltonian path molecule $e_{0,n-1}$ at or above our detection threshold for $n \geq 18$.

The *complete feedback digraph*, which contains L_n, but has a back edge from v_i to v_j whenever $0 < j < i < n-1$ represents what is presumably the worst case. An analysis similar to that of the simple feedback graph shows that for complete feedback digraphs,

$$[e_{0,n-1}] \leq \frac{1}{(\binom{n-2}{2}+2)(n-2)!}$$

holds throughout the computation. Thus, for the complete feedback digraph, the ligation phase will never produce $e_{0,n-1}$ at or above our detection threshold for $n \geq 10$.

The point behind these two examples is that an exponential loss of concentration occurs even on sparse digraphs (simple feedback graphs are sparse), and that a loss of concentration can become a significant consideration for problems not much larger than that solved by Adleman.

3 Other Algorithms and Issues

While reaction rate and concentration issues suggest that Adleman's technique will not scale as easily as might be hoped, they are devastating for various improvements and/or extensions of his technique.

For example, one version of Adleman's technique considers a "serialized" version of the original graph. For every vertex v_i in the original graph, we have new vertices $v_{\langle i,t \rangle}$ for $t = 0 \ldots n-1$; and for every edge $e_{i,j}$ of the original graph we have new edges $e_{\langle i,t \rangle, \langle j,t+1 \rangle}$. Hamiltonian paths in the original graph correspond to paths in the derived graph that never visit the same city at two different times.

The advantage of the serialized version of Adleman's technique is that the molecules that correspond to hamiltonian paths of the original graph are non-reactive, and so one never needs to worry about molecules that are "too long," nor about when it is most advantageous to terminate the annealing phase. Unfortunately, we now have quadratically many different reaction families of the form

$$\langle v_{\langle i,t \rangle}| + |v_{\langle i,t \rangle}\rangle \to \langle v_{\langle i,t \rangle}\rangle$$

Because the concentration of reactants in each of these families falls quadratically in the number of vertices, the reaction rates of these families fall quartically.

Reaction rate issues are even more damaging for techniques that attempt to use exponentially many reaction families, such as the algorithms for solving arbitrary PSPACE problems proposed by Beaver [Bea95] and Rooß-Wagner [RW95]. Although the vision of exponentially many reactions taking place in polynomially many parallel steps is very attractive, it is now all too clear that each of these parallel reaction steps requires exponential time.

An exponential loss of concentration seems to affect all DNA based algorithms for solving NP complete problems. For example, it is not difficult to contrive bad cases for Lipton's satisfiability algorithm in which the relative concentration of molecules representing satisfying assignments falls exponentially with respect to the concentration of nonsatisfying molecules as successive clauses are handled. Problems involving a few dozen clauses on a dozen variables can result in pushing the concentration of satisfying molecules to homeopathic levels.

4 Active Transport

In a sense, the two problems we've describe are different facets of the same issue. Computational systems based on chemical reactions will need to account for, and perhaps actively monitor and adjust, the concentration of reactants involved. Failure to do this leads to unexpectedly bad performance in mild cases of concentration loss, and to pushing the concentration witnessing compounds below detection thresholds in more severe cases.

These issues are addressed by biological systems in nature. Cells need to control the concentrations of various compounds, they need to arrange for rare molecules to react, and they need to deal with undesirable byproducts of their own activity.

While many different mechanisms have evolved for managing the concentrations within the cell, the most successful involve membranes to separate a volume (a vesicle) and active systems that transport selected chemicals across that membrane. Since this transportation typically involves moving a chemical from a region of low concentration to a region of higher concentration, it decreases entropy and therefore requires energy. Nevertheless, the benefits to the cell of higher concentration (in terms of higher reaction rates, or under certain circumstances, of sequestering harmful compounds) justifies the expenditure of energy.

4.1 The Microstructure of Biological Computers

Moving from the cell to biological computers, we envision a microstructure of computational organelles that directs nanostructure of molecular computation.

This microstructure is based on a few basic types of vesicles: *collection vesicles* which contain the initial compounds, *reaction vesicles* which contain relatively high concentrations of a relatively few reaction families; *separation vesicles* in which molecules are separated according to what reaction families they participate in; and *waste vesicles* in which useless molecules are concentrated. These vesicles have impermeable outer membranes, and are connected to one another by microtubules. Reaction products that concentrate in waste vesicles can be digested into shorter path molecules, which then re-enter the collection vesicles.

The design of such a microstructure presents some interesting problems.

The components have to fit into a three-dimensional universe. This should not be a significant issue for Adleman's technique, because there are only linearly many reaction families; but it may be significant for other algorithms for which the number of reaction families involved grows super-cubically. Algorithms that are based on exponentially many reaction families remain impractical.

It is not at all clear how these structures will be fabricated. Undoubtedly there are difficult problems in the domain of materials science and self-organizing systems that will come out of the attempt to build such a structure.

Transporting molecules along microtubes requires time proportional to the length of the tube. This introduces a routing problem reminscent of VLSI design, but in three dimensions rather than two. The mathematical structure of the resulting system is a family of differential equations with time-delay terms—an impediment to both analysis and numerical simulation.

Finally, the energy requirements of such a system will need to be accounted for—especially at the separation vesicles where a decrease in entropy occurs.

Admittedly, what we are offering here is a crayon sketch of an architecture rather than a detailed proposal. We believe that it is a useful sketch nevertheless. Molecular computers, like Adleman's, that represent their state through the collection of chemicals found in a single well-mixed reaction vessel quickly loose performance as the number of reaction families increase. The key to avoiding this loss of performance is to concentrate the chemicals that make up a single reaction family into a much smaller volume. Because concentrations are bounded above, this implies that many different small volumes are involved—that is, that efficient biological computers must actively transport molecules to separate volumes based on the reaction families they are associated with.

4.2 CNA

The microstructure we discussed in Section 4.1 was first developed for an alternative model of biological computing based on protein synthesis [KMRS96]. Protein synthesis takes place at a ribosome, where molecules of mRNA are transcribed into proteins. If we view individual mRNA and protein molecules as strings over a finite alphabet, the process of protein synthesis is simply a finite-state transcription.

Traditional models of computation can be viewed as iterated finite-state transcription, and so if we ignore the inconvenient fact that the input and output "alphabets" of protein synthesis are different, it seems reasonable to base a computational model on iterated transcription. Our model of CNA-based computing is a biologically inspired, biochemically plausible model based on the notion that a ribosome-like structure can be constructed that performs a computationally useful transcription from CNA (a hypothetical class of chemicals) back to CNA. Let this seem completely without biological foundation, we would point to the literature on the "RNA-world" that preceeded the present protein world (e.g. [ESB95]). The equation CNA = RNA motivates and partially justifies our basic biocomputational model.

CNA-based computing has some significant advantages and disadvantages with respect to DNA-based computing.

Because the key reactions of CNA-based computing depend on only a single low-concentration reactant, our intuition is that the computational performance of CNA-based computers will degrade linearly as concentration decreases, rather than quadratically as in the case of ligation-based computation.

The translation of CNA occurs at membranes and is energetically favorable. Therefore, there do not seem to be any major difficulties associated with the design of separation vesicles; whereas in DNA-based computing, it will be necessary to design molecular "pumps" to move DNA against a concentration gradient.

On the other hand, annealling-based computing is really a triumph of message passing. We do not know of a satisfactory message passing mechanism for CNA, and this remains an area of research.

Finally recombinant DNA-technology is widely available, and Adleman's successful solution of a seven-city instance of hamiltonian paths is a most convincing demonstration of the feasibility of his technique. We cannot yet give an equally convincing demonstration; nevertheless, we believe that CNA-based computation is possible. Indeed, we believe that it will be a natural outgrowth of existing research on the RNA-world.

4.3 Conclusions

Early papers in biological computing did not attempt to analyze concentrations or reaction rates. We have shown that the best known biological algorithm (Adleman's) requires at least quadratic time, even on the simplest instances, which is exponentially worse than our naive expectations would suggest. On more complicated instances, a simple complexity-based analysis shows that even on sparse graphs, the yield of Adleman's algorithm can fall exponentially in problem size. We conclude that these simple chemical notions should be accounted for in the analysis of biochemical algorithms.

We have also sketched a biologically inspired microarchitecture based on membrane limited vesicles and the active transport of selected molecules across membranes. Our hope and expectation is that such an architecture will circumvent the reaction rate problems inherent in prior biological computing schemes and may lead to more viable biological computing technologies.

References

[Adl94] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[Bea95] D. Beaver. A universal molecular computer. Technical report, Pennsylvania State University, 1995.

[ESB95] E. H. Ekland, J. W. Szostak, and D. P. Bartel. Structurally complex and highly active RNA ligases derived from random RNA sequences. *Science*, 269:364–370, July 21 1995.

[KMRS96] S. Kurtz, S. Mahaney, J. Royer, and J. Simon. Biological computing. In L. Hemaspaandra and A. Selman, editors, *Complexity Retrospective, II.* Springer Verlag, 1996. to appear.

[LBB+95] H. Lodish, D. Baltimore, A. Berk, S. L. Zipursky, P. Matsudaira, and J. Darnell. *Molecular Cell Biology.* Scientific American Books, third edition, 1995.

[RW95] D. Rooß and K. Wagner. On the power of bio-computers. Technical report, Universität Würzburg, 1995.