Biol. Cybern. 69, 275–281 (1993)

Biological Cybernetics

Self-replicating sequences of binary numbers. Foundations II: Strings of length $N = 4$

Wolfgang Banzhaf *

Central Research Laboratory, Mitsubishi Electric Corporation, 1-1, Tsukaguchi Honmachi 8-chome, Amagasaki 661, Japan

Received: 19 May 1992/Accepted in revised form: 19 March 1993

Abstract. We study an algorithm which allows sequences of binary numbers (strings) to interact with each other. The simplest system of this kind with a population of 4-bit sequences is considered here. Previously proposed folding methods are used to generate alternative two-dimensional forms of the binary sequences. The interaction of two-dimensional and one-dimensional forms of strings is simulated in a serial computer. The reaction network for the $N = 4$ system is established. Development of string populations initially generated randomly is observed. Nonlinear rate equations are proposed which provide a model for this simplest system.

1 Introduction

We want to study the application of a previously proposed algorithm inspired by the interaction of strands of RNA (Banzhaf 1993a). The algorithm is based on the fact that an RNA sequence can assume two forms, depending on the conditions of its environment. The one-dimensional form of the sequence ("genotype"), which only contains and conserves the information of the ordered sequence of nucleotides, may be termed a structural form. Under certain specific conditions, however (proper solvent and temperature range), an RNA sequence assumes a two-dimensional and even a three-dimensional form ("phenotype"), resulting from weaker hydrogen bondings and van der Waals interactions between the nucleotides.

Like proteins, which may be called the machines of life (Fraunfelder 1988), RNA molecules can perform certain functions such as cutting and splicing (Kruger et al. 1982; Guerrier-Takada et al. 1983). It is the alternative phenotypic form of the RNA sequence which determines its ultimate function and whether it has a function at all (Cech and Bass 1986).

Therefore, if we take sequences of binary numbers in a computer and add, in close analogy to the RNA system, an alternative form of the sequences, we can expect some interesting phenomena of self-organization to take place. For this purpose, we have set up in a previous contribution (Banzhaf 1993a) the algorithmic framework of a competitive system consisting of binary strings. We provided a mapping from one-dimensional sequences of binary numbers to two-dimensional forms, and rules for the interaction of these two forms with each other. In the present paper we want to start filling this framework by considering the simplest nontrivial system of this kind with strings of length $N = 4$. We shall refer to equations, figures, and tables of the previous report (Banzhaf 1993a) by adding I to the respective numbering.

2 Static features

Let us start by naming the string types. We shall use decimal numbers that correspond to the binary numbers carried by a string as compact descriptions. Thus, e.g.,

$$s = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

will be called $s^{(5)}$.

Folding strings into 2×2 matrices can take place in various ways. One of these ways will allow us to consider the operations involving scalar products [according to (I-4)] with the string acting on itself as ordinary matrix multiplications, so we shall consider it first. The arrangement is

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \rightarrow \mathscr{P}_s = \begin{pmatrix} s_1 & s_2 \\ s_3 & s_4 \end{pmatrix} \tag{1}$$

which can be easily generalized to the arbitrary size $\sqrt{N}$ (see Fig. I-1a).

Dedicated to Professor Hermann Haken on the occasion of his 65th birthday

* *Present address:* Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA

Another example of folding is a topological folding method where string neighbors also assume neighboring positions in the matrix (see also Fig. I-1b, c):

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \rightarrow \mathscr{P}'_s = \begin{pmatrix} s_1 & s_2 \\ s_4 & s_3 \end{pmatrix}. \tag{2}$$

It goes without saying that each folding method comes also in a transposed form $\mathscr{P}_s^T, \mathscr{P}_s'^T$. Table 1 gives the resulting operators for both folding methods.

Next we shall give an example of each of the five types of reactions listed as (I-7 to 11). Using the squashed scalar product and the first folding method, we find, for example,

$$\mathscr{P}_{s^{(1)}} \oplus s^{(6)} \Rightarrow s^{(4)} \tag{3}$$

$$\mathscr{P}_{s^{(3)}} \oplus s^{(1)} \Rightarrow s^{(1)} \tag{4}$$

$$\mathscr{P}_{s^{(1)}} \oplus s^{(11)} \Rightarrow s^{(1)} \tag{5}$$

$$\mathscr{P}_{s^{(4)}} \oplus s^{(4)} \Rightarrow s^{(8)} \tag{6}$$

$$\mathscr{P}_{s^{(8)}} \oplus s^{(8)} \Rightarrow s^{(8)} \tag{7}$$

where the sign $\Rightarrow$ indicates only the string that was newly produced by the interaction (suppressing the conserved reactants). A list of all reactions for the present case is given in Table 2. Similar reaction tables can be derived for other folding methods.

Note again the fact that we are dealing here with a system of binary strings, each of which has only four components. This poor material is able to "react" in quite a complicated manner, as a glance at Table 2 tells us. Hence, already for $N = 4$, we expect rather complicated dynamic behavior. In general, these systems exploit the phenomenon of combinatorial explosion (see Table I-1). Therefore, by studying $N = 4$, we shall have gained only a slight impression of what might be possible in larger systems of this sort. An entire reaction universe is opened

Table 1. Results of folding methods 1 [Eq. (1)], and 2 [Eq. (2)], applied to strings 0, . . . , 15, where only a rearrangement of matrices is observable

Folding method	String number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$
2	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

Table 2. Reactions using computations according to (I-4) with folding (1). Four reactions are self-replications (1⊕1, 8⊕8, 9⊕9, 15⊕15), 76 are replications

Operator	String														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	4	5	4	5	0	1	0	1	4	5	4	5
2	0	1	1	0	0	1	1	4	4	5	5	4	4	5	5
3	1	1	1	4	5	5	5	4	5	5	5	4	5	5	5
4	2	0	2	8	10	8	10	0	2	0	2	8	10	8	10
5	3	0	3	12	15	12	15	0	3	0	3	12	15	12	15
6	2	1	3	8	10	9	11	4	6	5	7	12	14	13	15
7	3	1	3	12	15	13	15	4	7	5	7	12	15	13	15
8	0	2	2	0	0	2	2	8	8	10	10	8	8	10	10
9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	0	3	3	0	0	3	3	12	12	15	15	12	12	15	15
11	1	3	3	4	5	7	7	12	13	15	15	12	13	15	15
12	2	2	2	8	10	10	10	8	10	10	10	8	10	10	10
13	3	2	3	12	15	14	15	8	11	10	11	12	15	14	15
14	2	3	3	8	10	11	11	12	14	15	15	12	14	15	15
15	3	3	3	12	15	15	15	12	15	15	15	12	15	15	15

once we consider larger strings with length, say, of order $N \approx O(100)$.

In the real world, the smallest virus contains about 3 000 base pairs. Since its elements are certainly able to interact in much more complicated ways than the binary strings we are considering here, we may have a small hint about how intricate the fundamental mechanisms of life really are.

3 Dynamic behavior

For a discussion of the system's dynamic behavior we first have to identify observables. Global quantities which characterize the time development of this system (Haken 1983) are the concentrations $x_i(t)$ of all the different string types $s^{(i)}$:

$$x_i(t) = m_i(t)/M \tag{8}$$

where $m_i(t)$ is the number of actual appearances of string type $s^{(i)}$ in the string soup at time t, and $M \equiv M_1 = M_2$ is the total number of strings. We have

$$\sum_{i=1}^{n_s} x_i(t) = 1 \tag{9}$$

Figure 1 shows the first 2×10^6 iterations through the algorithm with a population size of $M = 10^5$. A very special initial distribution of strings was used here, concentrating the entire population in one type, $x_8(0) = 1$. The system relaxes to a macroscopic attractor state given by a fixed distribution of type concentrations at some later time. Despite the fact that the initial distribution favored one self-replicating string type with additional selective advantage, other types emerge until a stable distribution is reached.

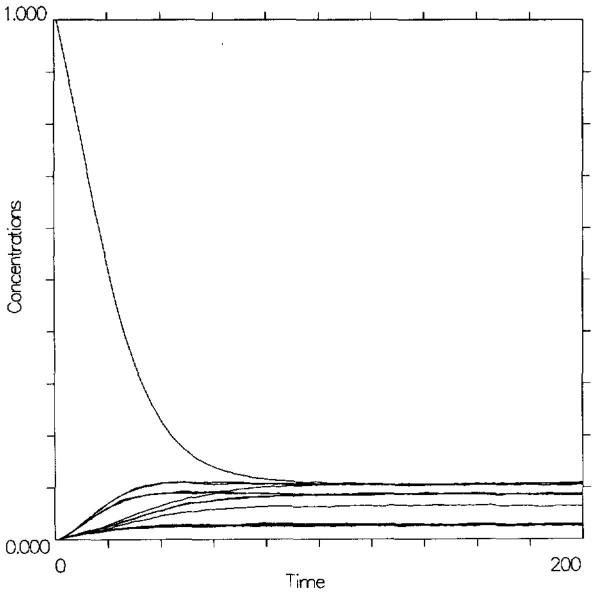

Fig. 1. Time development of $n_s = 15$ string-type concentrations x_i. Folding according to (1). Number of strings, $M = 100\,000$; initial condition, $x_8(0) = 1$, $x_i(0) = 0\,\forall i$. Shown are the first 200 time steps, each consisting of 10 000 sweeps through the algorithm

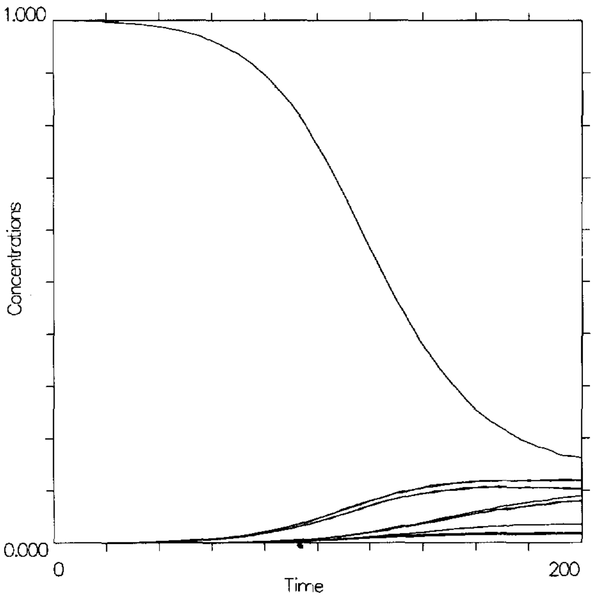

Fig. 2. Selection parameter $n = 5$. Same folding and initial conditions as in Fig. 1

Running the algorithm under different initial conditions, for instance, nearly equal initial distribution of strings (Banzhaf 1993c), reveals that the macroscopic behavior changes only slightly, well within the range of fluctuations present in the system. We observe, however, more pronounced oscillations between concentrations of the various string types if we use smaller population sizes.

If we change selection probabilities by choosing another parameter n in (I-13), we end up with a different distribution of types $s^{(i)}$ (see Fig. 2). Note that this change of global behavior was brought about by applying a higher selection pressure on the system. Thus, $s^{(8)}$ is more reluctant to give room to other string types. The global behavior also changes if we use another folding rule. Figure 3 shows the development of concentrations for the second folding method. The difference shows up clearly as another combination of concentrations dominates the long-term behavior.

As far as the overall stability of the system is concerned, we observe in Fig. 4 a case where the balance between string sorts is seriously disturbed. It was generated by altering the basic algorithm of part I (Banzhaf 1993a) to include destructors (by skipping step 6 of the algorithm). This example clearly demonstrates the need for the countermeasure taken in the algorithm.

Since the overall system behavior is approximately the same for different initial conditions and qualitatively similar for different folding rules, we shall consider our first simulation example in more detail. Table 3 shows the average concentration levels of the different sorts of strings for the simulation reported in Fig. 1. Interestingly, the most frequent string types are not necessarily self-replicating strings [$s^{(1)}, s^{(8)}, s^{(9)}, s^{(15)}$], but other sorts able to muster support from cross-reactions and resistant to the selection pressure.

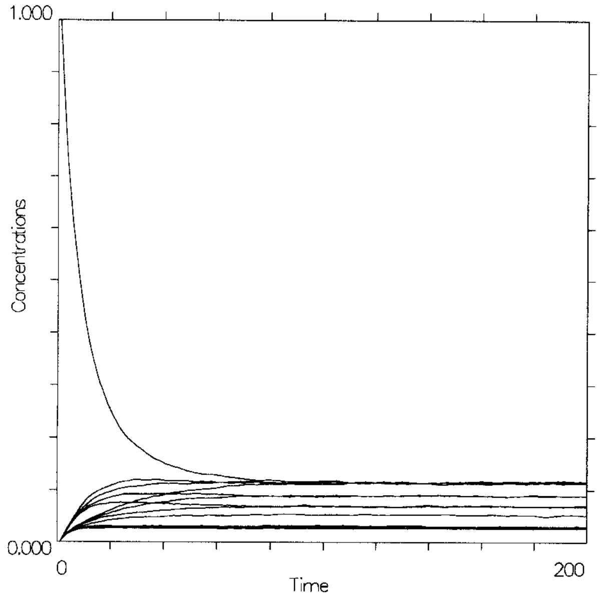

Fig. 3. Folding according to (2), system otherwise same as in Fig. 1. System relaxes to a different attractor characterized by another combination of concentration levels

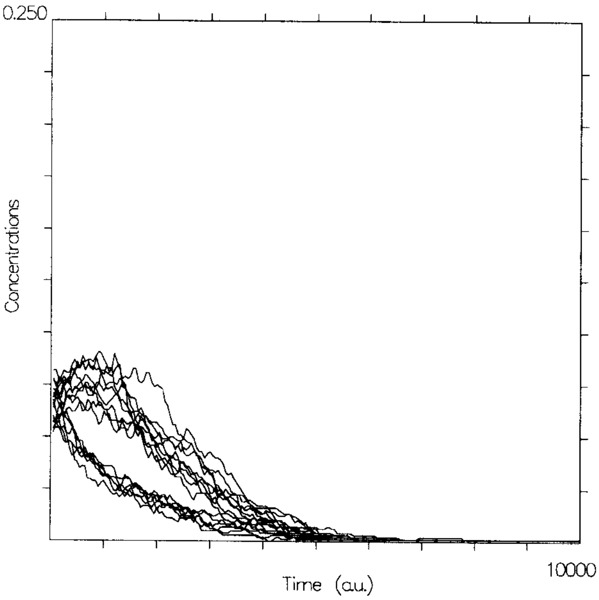

Fig. 4. The destructor is not removed regularly. $M = 1000$. Folding and initial conditions as in Fig. 1. All concentrations, except $x_0(t)$, are shown for 10000 elementary sweeps through the algorithm. The destructor is able to suppress all activity quickly. *a.u.*, arbitrary units

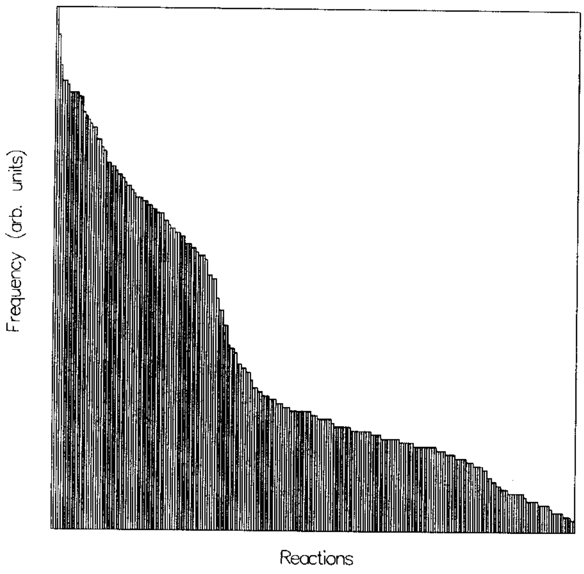

Fig. 5. Reaction frequency histogram for all 225 reactions accumulated over elementary sweeps 90000, . . . , 100000. Same folding as in Fig. 1. $M = 1000$

If we recall that the algorithm creates each string type by an approximately constant rate (at the expense of removed destructors), we can imagine that nearly every possible reaction is actually taking place once in a while. Figure 5 demonstrates this fact clearly, as it shows a frequency histogram for the reactions accumulated over 10000 iterations. We can see that all reactions have happened at least sometimes.

This concludes the section on $N = 4$. As we consider longer strings in a forthcoming contribution (Banzhaf 1993b), more complicated systems will emerge, and soon the combinatorial explosion will prohibit us from completely investigating the sequence space of strings. This was the reason why we studied the $N = 4$ system in so much detail.

4 Model equations

A system of coupled differential equations similar to those studied by Eigen and Schuster for the hypercycle (Eigen and Schuster 1977, 1978a, b) can model the proposed string reactions. As was shown by Gillespie and Mangel (1981), most stochastic reaction systems are sufficiently well behaved to allow for a deterministic description with continuous nonrandom concentration functions $y_i(t)$ of the different string types $1 \leqslant i \leqslant n_s$. The

Table 3. Concentrations x_i for simulation of Fig. 1 at $t = 200$

Sort	2	4	8	1	12	3	5	10
Frequency	10.6%	10.5%	10.5%	10.4%	8.9%	8.8%	8.8%	8.7%
Sort	15	9	6	7	11	13	14	0
Frequency	6.5%	2.9%	2.8%	2.6%	2.6%	2.6%	2.6%	0%

functions $y_i(t)$ are considered to approximate the time-averaged concentrations $\langle x_i \rangle_t$:

$$y_i(t) \cong \langle x_i \rangle_t, \quad 0 \leqslant y_i(t) \leqslant 1 . \tag{10}$$

For the simplest algorithm used in $N = 4$, deterministic rate equations for $y_i(t)$ can be easily formulated:

$$\begin{aligned} \dot{y}_i(t) = {} & A(t) + [B_i y_i(t) + \sum_{k \neq i} C_{ik} y_k(t) - D_i] y_i(t) \\ & + \sum_{j,k \neq i} W_{ijk} y_j(t) y_k(t) - \frac{y_i(t)}{\sum_k y_k(t)} \Phi(t) \end{aligned} \tag{11}$$

where B_i, C_{ik}, D_i, and W_{ijk} are (coupling) constants; $A(t)$ is an unspecific growth term; and $\Phi(t)$ is a flow term used to enact competition between the various string types $s^{(i)}$.

Let us discuss in more detail the different contributions to (11). The first term, $A(t) > 0$ is a growth term due to step 6 in our algorithm. For this term, we may assume that the probability to generate the destructor does not change over time and is, hence, approximately equal for all types of strings. If this is too rough an approximation, we may compute A by

$$A(t) = \sum_{ij} a_{ij} y_i(t) y_j(t) \tag{12}$$

where

$$a_{ij} = \begin{cases} 1 & \text{if } s^{(i)} \oplus s^{(j)} \Rightarrow s^{(0)} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

reflects reactions producing the destructor.

The second term describes self-replications of type $s^{(i)}$ [see (I-11)] in steps 3 and 4 with $B_i = 1$, if this reaction exists, and $B_i = 0$ otherwise. It is quadratic in concentration y_i, since operator and string are required to be of type i. The third term describes all other replication reactions between strings [see (I-8) and (I-9)] in steps 3 and 4 of the algorithm with $C_{ik} = 1$, if replication occurs between i and k, and 0 otherwise. It depends on two concentrations, y_i and y_k. The fourth term is linear in y_i as it models the spontaneous decomposition of strings according to step 7.

The first of the two remaining terms describes a reaction between operator $\mathscr{P}_{s^{(j)}}$ and string $s^{(k)}$ leading to string $s^{(i)}$, in which case $W_{ijk} = 1$, and 0 otherwise. Such reactions take place as a consequence of steps 5 and 8 of the algorithm. The flow term, finally, assures that the sum of all concentrations is constant over the relaxation process. $\Phi(t)$ is defined as

$$\Phi(t) = \sum_i \dot{y}_i(t) . \tag{14}$$

By keeping $\sum_i y_i$ constant, a strong competition between string types is caused. The flow term corresponds to step 4 in the algorithm.

We are now in a position to examine the behavior of these equations for 15 string types with concentrations $y_i(t)$ in two special cases: (a) with random couplings and (b) with couplings derived from the $N = 4$ system.

Table 4 shows the choice for the various parameters in these two cases. Qualitatively, the behavior for both cases is quite similar, as can be seen from a comparison of Figs. 6 and 7. Figure 6 depicts a typical random parameter run. Most of the concentrations are relaxing to individual levels, and these are independent of the starting concentrations. The system is competitive, having point attractors in state space.

Figure 7 demonstrates a run with couplings from an $N = 4$ binary string system. In this case, we have used the reaction tables to compute the outcome which is only feasible for $N = 4$ or 9. Figure 7 then shows the simulation of the differential equations (11) under the same initial conditions as that for the stochastic system simulation of Fig. 1. We can clearly observe that some concentrations merge into the same levels, due to the particular interactions present in the dynamics of this binary string system. The comparison between the statistical data and the numerical integration of (11) shows very good agreement.

Figure 8 gives a more systematic account of what happens if we change the selection pressure. Resulting

Table 4. Couplings between 15 string types of the $N = 4$ system for a simulation of (11) and $D = \sum_k D_k y_k$

Coupling	Short description	Random value	System value
A	Spontaneous generation	$\frac{1}{240}$	$\sum a_{ij} y_i y_j + \frac{D}{15}$
B_i	Self-replication	$\begin{cases} 1 & \text{with } p = \frac{1}{4} \\ 0 & \text{with } p = \frac{3}{4} \end{cases}$	$\begin{cases} 1 & \text{if reaction exists} \\ 0 & \text{otherwise} \end{cases}$
C_{ik}	Replication	$\begin{cases} 0 & \text{if } i = k \\ 1 & \text{with } p = \frac{1}{3} \\ 0 & \text{with } p = \frac{2}{3} \end{cases}$	$\begin{cases} 0 & \text{if } i = k \\ 1 & \text{if reaction exists} \\ 0 & \text{otherwise} \end{cases}$
D_i	Spontaneous decomposition	$0 \leqslant D_i \leqslant 1$	$\sim \left[\frac{I^{(i)}}{N}\right]^n$
W_{ijk}	Reaction	$\begin{cases} 0 & \text{if } i = j \text{ or } i = k \\ 1 & \text{with } p = \frac{1}{10} \\ 0 & \text{with } p = \frac{9}{10} \end{cases}$	$\begin{cases} 0 & \text{if } i = j \text{ or } i = k \\ 1 & \text{if reaction exists} \\ 0 & \text{otherwise} \end{cases}$

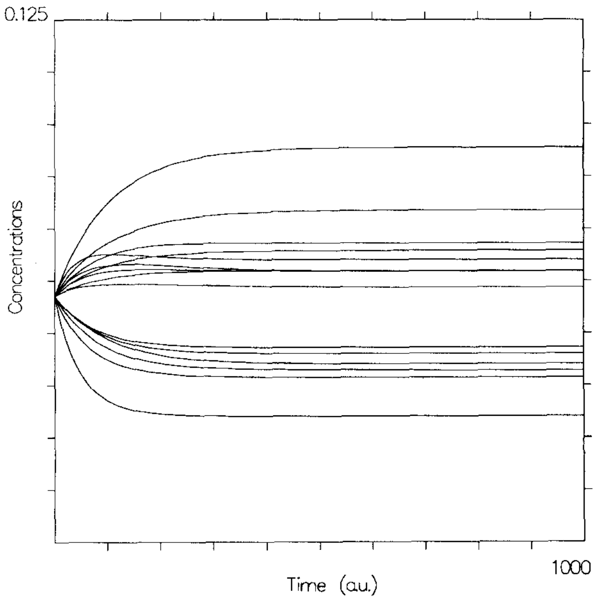

Fig. 6. Simulation of the differential equations (11). Random choice of coupling constants according to Table 4.

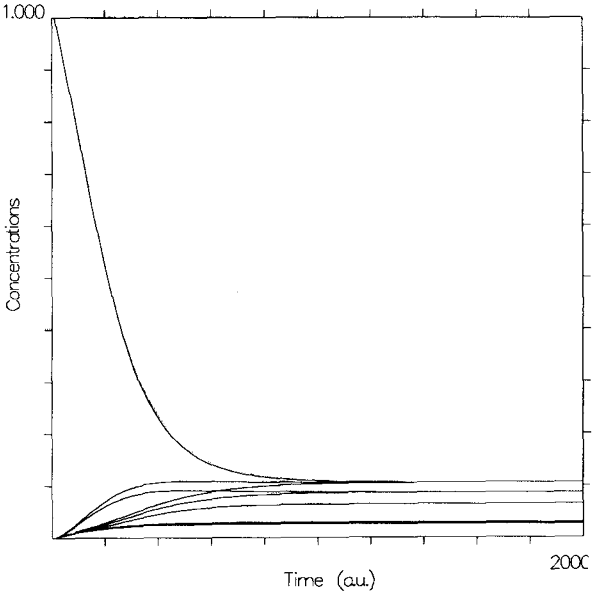

Fig. 7. Simulation of the differential equations (11). Constants derived from binary string system $N = 4$.

asymptotic concentrations $x_i(\infty)$ stabilize when sufficient selection pressure is applied.

If we want to apply the equations to cases involving larger strings, however, we quickly run into problems. For $N = 9$, W_{ijk} has 1.3×10^8 components.

5 Conclusion

In this contribution we have discussed a particular example for a new self-organizing system based on sequences of binary numbers. The example we studied was the simplest nontrivial system possible within the proposed framework.

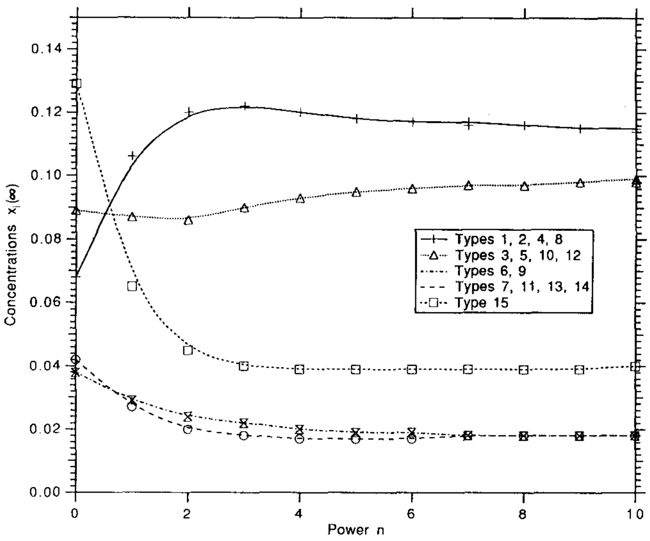

Fig. 8. Varying selection pressure in model equations for $N = 4$. Parameter n of (I-13) is increased until asymptotic concentrations $x_i(\infty)$ show no further change. The selective pressure favors types with low numbers of 1's: $s^{(1)}$, $s^{(2)}$, $s^{(4)}$, $s^{(8)}$.

Certainly one lesson that can be drawn from these simulations is that, provided populations are large enough, the characteristic feature of self-replication is *not* the most important feature of a string type. What is more important is the network or web of interactions that a type can exploit to maintain or increase its level of concentration, i.e., the catalytic cycle or hypercycle in which it is involved (see also Kauffman 1986). As a consequence, we expect that mutations possibly occurring during a string interaction are *not* as important as the semilocal character of the interaction itself. In other words, the role of mutations to explore neighborhoods of already existing strings in sequence space may be taken over by regular string interactions which usually generate (at least slightly) modified strings (except for replication and self-replication reactions). For much larger strings, however, we expect mutation to play an important role.

In this report we have only discussed the basic algorithm applied to a very simple system, $N = 4$. More complex systems will be treated elsewhere (Banzhaf 1993b, d), at least preliminarily. Together with applications of these systems centering around combinatorial optimization (W. Banzhaf, in preparation) they will constitute a major part of future work. Yet the algorithm outlined here is but one example of an entire class of algorithms employing a new interpretation of the manipulation of logical and mathematical symbols.

Acknowledgements. I wish to thank my colleagues T. Iwamoto, T. Ishii, and E. Lange. I am grateful to Drs. T. Nakayama and K. Kyuma for establishing and maintaining the creative atmosphere in our research group. I am indebted to Mr. S. Winquist for carefully reading an earlier version of this manuscript.

References

Banzhaf W (1993a) Self-replicating sequences of binary numbers. Foundations I: General. Biol Cybern 69:269–274

Banzhaf W (1993b) Self-replicating sequences of binary numbers. III. Larger systems

Banzhaf W (1993c) Self-replicating sequences of binary numbers. Comp Math (in press)

Banzhaf W (1993d) Self-replicating binary strings: the buildup of complexity. Complex Syst (submitted)

Cech TR, Bass BL (1986) Biological catalysis by RNA. Annu Rev Biochem 55:599–629

Eigen M, Schuster P (1977) The hypercycle: a principle of natural self-organization, part A. Naturwissenschaften 64:541–565

Eigen M, Schuster P (1978a) The hypercycle: a principle of natural self-organization, part B. Naturwissenschaften 65:7–41

Eigen M, Schuster P (1978b) The hypercycle: a principle of natural self-organization, part C. Naturwissenschaften 65:341–369

Fraunfelder H (1988) Biomolecules. In: Pines D (ed) Emerging synthesis in science. (SFI studies in the sciences of complexity, vol 1). Addison-Wesley, Reading, Mass

Gillespie DT, Mangel M (1981) Conditioned averages in chemical kinetics. J Chem Phys 75:704–709

Guerrier-Takada C, Gardiner K, Marsh T, Pace N, Altman S (1983) The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme. Cell 35:849–857

Haken H (1983) Synergetics. An Introduction. Springer, Berlin

Kauffman S (1986) Autocatalytic sets of proteins. J Theor Biol 119:1–24

Kruger K, Grabowski PJ, Zaug AJ, Sands J, Gottschling DE, Cech TR (1982) Self-splicing RNA: autoexcision and autocyclization of the ribosomal RNA intervening sequence of *Tetrahymena*. Cell 31:147–157