Experiments with a Turing gas: Mimicking chemical evolution*

RENUKA RAI and HARJINDER SINGH*

Department of Chemistry, Panjab University, Chandigarh 160 014, India

**Corresponding author (Fax, 91-172-541409; Email, raman%phys@puniv.chd.nic.in).*

Three different experiments with colliding artificial molecules, made of binary strings, are reported. Rules of evolution are designed such that the innovative collisions, i.e., those producing new species at the end of a reaction, become a negligible part of total number of collisions after a sufficiently large number of collisions have taken place. The distribution of species in the final mixture according to their bit size as well as the decay profile of the reactive and the innovative collisions are obtained. It is seen that the prescribed rules are sufficient to characterize a self-regulatory feature of the evolving system.

1. Introduction

The study of evolving systems is a major part of complex systems research. In its widest sense, evolution is just another word for a change in time. We will, however, consider evolving systems as those that move towards a greater extent of self-organization and complexity as time flows. Such systems are abundant in nature, from the world of molecules to that of organisms. Man-made evolving systems share numerous features with natural systems. Therefore, similar theoretical methods have been used to study organismal evolution, evolution of proteins from small molecules, economic systems, urbanization and many other problems. Experimental studies of dynamical behaviour at diverse scales of biological organization are difficult because of limited techniques and objects available for experimentation and severe constraints of time and spacé. Moreover, realistic description of such systems is often next to impossible. An alternative approach to studying evolving systems is to create and study artificial entities that mimic natural evolution (Artificial Life 1989, 1992, 1994).

In physical and chemical systems a wide spectrum of interactions between atoms and molecules producing new atoms and molecules exists. Molecules are arrangements of atoms in a specific way according to their chemical properties. They have the ability to interact with other atoms and molecules to create new molecules. This is analogous to the manner in which the individual components in any other evolving system, biological or otherwise, show a tendency toward self-organization.

We present here a pedagogical exercise in exploring adaptive behaviour in a simple computer model involving strings of bits. The model utilizes an important characteristic of an adaptive system; namely, the objects of which it is made interact with the objects of the system themselves and a self-referential loop of object-function properties is formed. We try to isolate simple features of this model, such as self-regulatory growth, and limit ourselves to elucidating the rules leading to these properties.

2. The Turing gas

Evolving systems frequently show a tendency to organize themselves. The order that emerges in a system may be a consequence of different factors. Most generally, a Darwinian model is utilized, where populations of organisms are randomly varying systems shaped to adaptation by external forces by the process of natural selection. In the process of molecular evolution, all this information is contained in the expressions for the rate processes involving reactivity as well as other terms contributing to spatio-temporal changes. For instance, consider the building of large molecules, P (say, a polymer) from smaller units, M (say, a monomer). The polymerization process, leading potentially to the formation of various types of polymers, itself is an evolution

*Presented at the National Symposium on Evolution of Life.

Keywords. Artificial molecules; binary strings; Turing gas; chemical evolution

in time. Simultaneously, random mutations may be occurring. Thus, there is a competition between species that is expressed by means of non-linear kinetics. Solution of the resulting non-linear differential equations provides varying concentration (population) profiles of the different species. In nature, however, species are typically numerous. Thus, what one has is a larger set of coupled non-linear differential equations, the solution of which is enormously computer-intensive.

Instead of the differential equation approach, networks of artificial 'logical' reactions can be used as alternative models of interactive processes in nature (Kauffman 1993). A specific collection of polymers, which are in fact interacting strings of characters in a language, autocatalytic in their reactive behaviour, describe the evolving set at a point of time. Such a set, called a Turing gas, itself evolves out of a simple, undifferentiated initial state, generating a sequence of highly differentiated final states. In such abstract artificial chemistries, the goal is not to achieve a precise reproduction of the path from the pre-biotic soup to the modern day *Homo sapiens*; rather it is to seek a better theoretical insight into how complexity and, perhaps, life emerges within a wide class of pre-biotic soups.

Artificial chemistry models of evolving systems have been used by a number of people. Fontana (1992, 1994), for instance, has used functions expressed in a symbol manipulation system called λ-calculus, represented as strings of characters and reacting with similar functional strings via function composition, producing new strings in the process. The ability to utilize functional programming provides in turn an enhanced ability to search for combinatorial loops in evolution. We describe below three experiments done with a Turing gas using a common procedural programming language. As mentioned earlier, this is a pedagogical exercise and hence, some of the intricate features mentioned above are not investigated. Rather, using a procedural language we wish to make the point that even with poor computational facility, such experiments can be carried out and meaningful conclusions can be reached.

3. Three experiments with evolving molecules

We have used a procedural programming language, C, to build a model for representation and isolation of a few features of adaptive systems. In our model, the combinatorial objects are represented by strings of characters ('0' and '1'). Three different sets of rules are designed that transform the strings into other strings.

In all the three experiments, strings consisting of '0' and '1' are generated randomly. Once generated, the strings are allowed to interact and the expression for their interaction is evaluated. The rules of interaction for the three experiments are given in table 1. In some of the interactions, the least significant bit(s) [LSB, the bit(s) with the least place value(s)] of the first string and/or the most significant bit (MSB, the bit with the highest place value) of the second string are involved, such that a product different from the original reactants is produced. These are the reactive collisions. In another kind of reactive collisions, a string that is not new but a copy of one of the reactants is produced or a reactant simply disappears. Every other collision, when nothing else happens other than the original reactants remaining unchanged, is termed non-reactive or elastic. After each reactive collision, the system is scanned to check if the product string is already present in the set. A collision that leads to the formation of a string that was not present earlier in the set is termed innovative.

The rules are to a large extent arbitrary. However, what we present here are three select set of rules amongst many we used, with an objective of restricting innovative collisions between strings so that limit to growth may become possible. We impose the general restriction that strings with more than 5 bits cannot react. This is a reflection of the steric or other inhibition of functional activity. In natural systems, this could correspond, for instance, to the trapping of the functionally active parts of the system in inert areas created by a size above a threshold or to a catastrophe that has a terminal impact only on the larger strings. Such a restriction has been used in earlier studies also (Kauffman 1993).

Experiment 1: A set of 1000 strings is generated randomly. Out of this set two strings are selected at random and allowed to react according to rules prescribed in table 1. A new string is generated if the collision is reactive. To keep the number of strings in the set constant, one of the original 1000 strings is removed randomly from the set after each reactive collision. Since each string may occur more than once, the probability of selection of strings for collision as well as for removal is proportional to its frequency in the set.

Experiment 2: From a set of 15 randomly generated strings, two are selected to collide and the interaction expression is evaluated as prescribed in table 1. In this experiment the colliding strings as well as the string which is produced are preserved. No string is removed from the set. As a result, the number of strings after every reactive collisions increases by one.

Experiment 3: A set of 15 strings is generated and from it 2 are picked at random for collision. In this experiment, the rules of interaction are such (table 1) that a reactive collision may lead to generation, modification or deletion of a string. There is no imposed mechanism here to keep the total number of strings constant.

Since the bit structure of each string determines the outcome of its interactions with other strings, there is a built-in structure-function correlation. In this sense, we can use the words 'string' and 'function' interchangeably.

4. Results

Experiment 1: The initial set of 1000 randomly generated strings are shown in table 2. There are 12 different kind of strings. To investigate how the rules of interaction operate, we allowed for all possible, namely 144, collisions between the pairs of strings. Among these 132 were reactive, of which 48 are innovative.

A pair collision analysis reveals that 3 out of 12 functions are general copiers. A general copier is a function that copies every function in the system with which it collides. It is also a self-replicator, i.e., it copies itself. There are 3 selective copiers that copy only particular functions. The other collisions may be either elastic (non-reactive) collisions or a modified string may be generated. Initially the strings consist of two or three bits and the collisions result in strings that may contain higher number of bits. Thus, initially most of the collisions are innovative. The rate of innovation is found to decrease as the number of collisions increases (figure 1).

Table 1. Rules of interaction.

Experiment 1: (The reactants are preserved)

If the 2 LSB of 1st string are	And additionally	Result
00	—	Copy the second string
01	If the number of bits in second string is:	
	(i) 2	Copy the second string
	(ii) else	Append 2nd string to 1st
10	If the 3rd LSB of the 1st string is:	
	(i) 0	Copy the 1st string
	(ii) 1	No reaction
	(iii) Does not exist	No reaction
11	—	Delete the LSB of 1st string and append 2nd string to this

Experiment 2: (The reactants are preserved)

LSB of 1st string	MSB of 2nd string	Result
0	0	Delete LSB of 1st and MSB of 2nd; then append 2nd to 1st
0	1	Append 2nd to 1st
1	0	Complement LSB of 1st and MSB of 2nd and append 2nd to 1st
1	1	Delete MSB of 2nd and append it to 1st

Experiment 3:

LSB of 1st string	MSB of 2nd string	Result
0	0	Delete LSB of 1st and MSB of 2nd string; append 2nd to 1st; replace the 1st string by the new string; store 2nd as such
0	1	Append 2nd to 1st and replace 2nd string by this; store 1st as such
1	0	New string generated; complement the LSB of 1st and MSB of 2nd; then append 2nd to 1st
1	1	Delete MSB of the 2nd and append it to 1st remove the colliding strings from the set

Strings with more than 5 bits cannot react (for all experiments).
LSB, Least significant bit, i.e., the bit with the lowest place value; MSB, most significant bit, i.e., the bit with the highest place value.

Experiment 2: The initial set of 15 2-bit strings generated randomly is shown in table 2. There are 4 kinds of strings with 16 possible pair collisions. All of the 16 collisions are reactive and 12 of these are innovative. Similar to the first experiment most of the collisions were found to be innovative initially. But as the number of collisions increases, the elastic collisions are overwhelming in number. This is shown in figure 2a. In figure 2b, we show the distribution of the strings according to the number of bits after 5000 collisions.

Experiment 3: The initial distribution of randomly generated strings is the same as in experiment 2. Now, out of the 16 possible pair-collisions, 8 result in modification of one of the colliding strings. Four of the reactive collisions lead to deletion of a string and a new string is generated (innovative) from the rest of the four collisions. In figure 3a, is seen a decreasing ratio of innovative collisions as the total number of collisions increase, similar to what was observed in the case of the first two experiments. In figure 3b, the distribution of the strings according to the bit size after 10000 collisions is shown.

Table 2. The initial distribution of strings.

String	Frequency
Experiment 1:	
00	98
01	124
10	121
11	130
000	67
001	62
010	49
011	67
100	64
101	72
110	74
111	72
Experiments 2 and 3:	
00	4
01	5
10	3
11	3

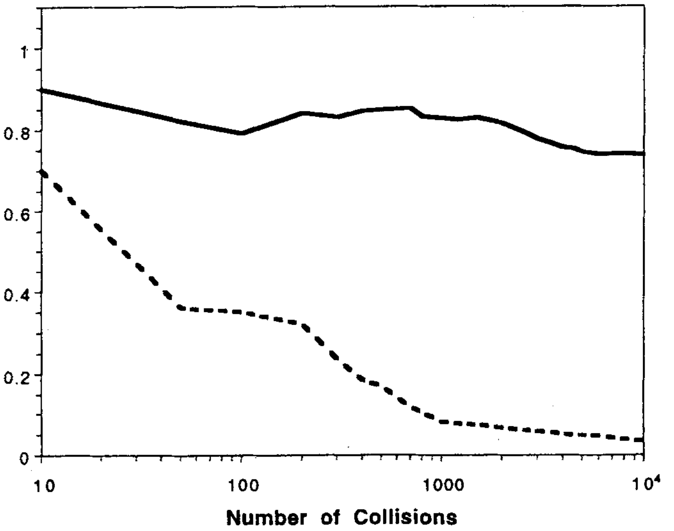

Figure 1. The number of reactive (—) and innovative (- - -) collisions compared to the total number of collisions in experiment 1.

Thus the most common observation is that the kind of rules chosen in these experiments lead to a regulation of the growth (in terms of ratio of innovative interactions to the total number of collisions) in the evolving systems.

An interesting feature emerging from figures 2a and 3a is that a distinct power-law like behaviour over a range of number of collisions. This is specially true for the third experiment. This needs to be explored further.

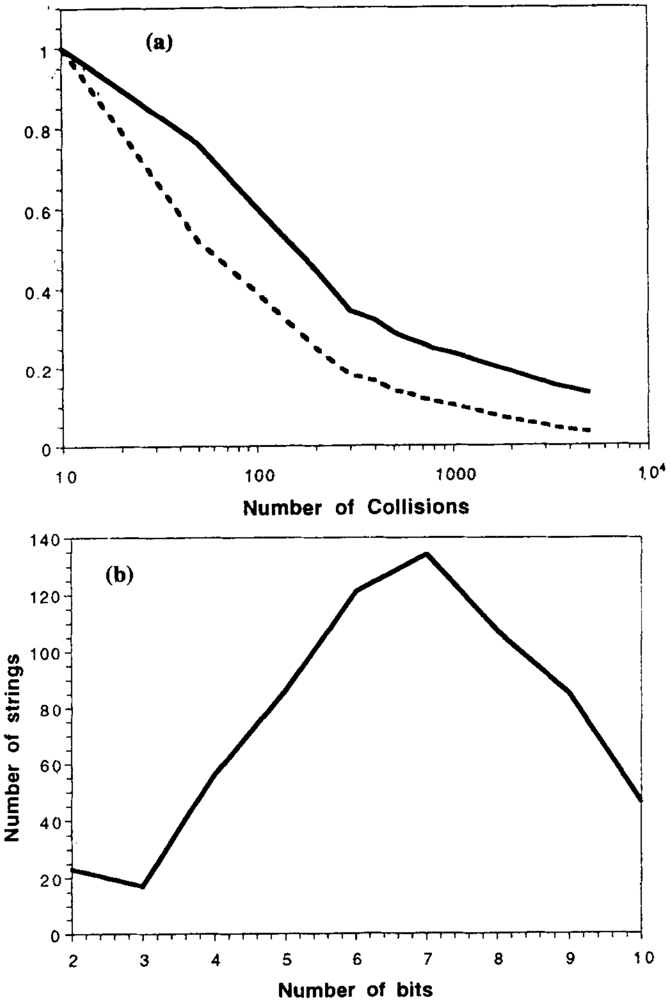

Figure 2. (**a**) The number of reactive (—) and innovative (- - -) collisions compared to the total number of collisions in experiment 2. (**b**) The bit size distribution of strings after 5000 collisions in experiment 2.

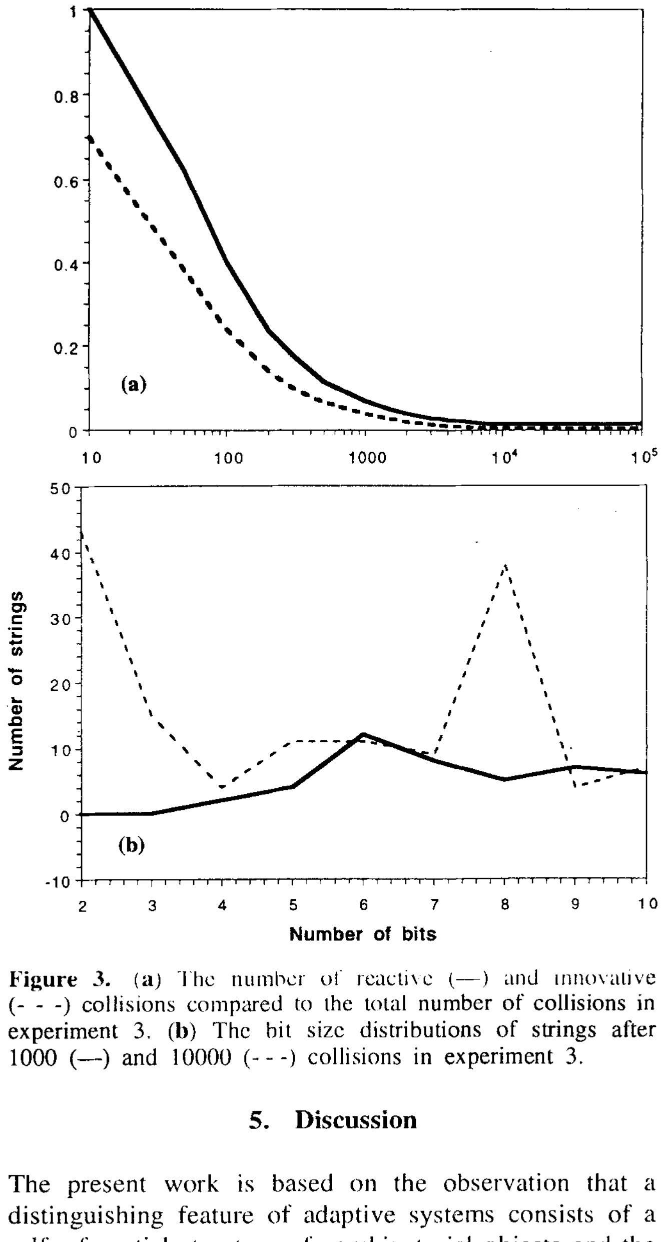

Figure 3. (**a**) The number of reactive (—) and innovative (- - -) collisions compared to the total number of collisions in experiment 3. (**b**) The bit size distributions of strings after 1000 (—) and 10000 (- - -) collisions in experiment 3.

5. Discussion

The present work is based on the observation that a distinguishing feature of adaptive systems consists of a self-referential structure of combinatorial objects and the functions they themselves encode. Our effort has been to develop a pedagogic model with an aim towards modelling self-regulatory growth.

The basic principle is to represent such a system using strings of characters. The behaviour of an ensemble of functions has been studied in terms of a set of character strings. These strings are randomly chosen for pairwise collisions. A collision scheme is a set of rules that determines what happens with a newly created string and with the collision partners.

We have conducted three experiments with different interaction rules and collisions schemes. In all the experiments, it was observed that the rate of innovative collisions decreases rapidly. Thus, effectively, a system with a large number of nonreactive strings emerges.

With the increase in number of collisions, the ratio of innovative collisions to the total number of collisions encountered is also found to decrease in all the experiments. Thus, after some time, the system displays mostly autocatalytic processes. The new strings produced already have their copies in the set, indicating a steady closure with respect to interactions. It should be noted that after every reactive collision, a string is removed randomly from the original set, thus there always remains a finite probability for innovative collisions.

What is the most interesting outcome of such experiments? The central question here is to see if an organization emerges naturally in an evolving system. What we have addressed is a small part of this question, namely the curtailment of growth beyond a certain size. Many of the interacting species or functions utilized here clearly form an auto-catalytic set. In contrast to realistic experiments in a laboratory, considerations like the size of the entire system of species (volume confining the reactants) do not arise here. Otherwise, similar dynamical characteristics and connectivity features may be explored. The size distribution of species obtained by us (figure 2b) is effectively unimodal at one stage of growth (after 5000 collisions) for a particular set of rules of evolution (experiment 2) and bimodal at a later stage (after 10000 collisions) and a different set of rules (experiment 3; figure 3b). Kauffman (1993) has shown in more extensive studies that the distribution of the number of species generated after a set of interactions depends on the sub-critical or supra-critical values of a parameter related to the number of kinds of species and the maximal size chosen. Further extensive studies requiring much larger system sizes and the number of collisions as well as a rule-space of much greater expanse need to be made to explore the issue of organization in its entirety.

References

Artificial Life 1989, *Artificial Life II* 1992 and *Artificial Life III* 1994 *Series of proceedings of interdisciplinary workshops on the synthesis and simulation of living systems* (eds) C G Langton, C Taylor, J D Farmer and S Rasmussen (Sante Fe Institute, Santa Fe, since 1987, Addison Wesley, New York)

Fontana W 1992 Algorithmic chemistry; in *Artificial Life II* p 159

Fontana W and Buss L W 1994 *Bull. Math. Biol.* **56** 1

Kauffman S A 1993 *The origins of order: Self-organization and selection in evolution* (New York: Oxford University Press)

MS received 28 April 1997; accepted 27 February 1998

Corresponding editors: AMITABH JOSHI and RAGHAVENDRA GADAGKAR