# Controlling Complex Dynamics with Artificial Biochemical Networks

Michael A. Lones[1], Andy M. Tyrrell[1], Susan Stepney[2] and Leo S. Caves[3]

[1] Department of Electronics, {mal503,amt}@ohm.york.ac.uk
[2] Department of Computer Science, susan.stepney@cs.york.ac.uk
[3] Department of Biology, lsdc1@york.ac.uk
University of York, Heslington, York YO10 5DD, UK

**Abstract.** Artificial biochemical networks (ABNs) are computational models inspired by the biochemical networks which underlie the cellular activities of biological organisms. This paper shows how evolved ABNs may be used to control chaotic dynamics in both discrete and continuous dynamical systems, illustrating that ABNs can be used to represent complex computational behaviours within evolutionary algorithms. Our results also show that performance is sensitive to model choice, and suggest that conservation laws play an important role in guiding search.

## 1 Introduction

Biochemical networks are the complex dynamical systems which underlie the functional and structural complexity seen within biological organisms. From an evolutionary computation perspective, biochemical networks are interesting because they describe complex behaviours in a way that is both concise and evolvable. This has led to growing interest in the use of computational models of biochemical networks, particularly within genetic programming. These include artificial genetic regulatory networks [2, 12, 17, 18], computational models of cellular metabolism [1, 8, 10, 19], and those derived from signalling networks [6]. We refer to them collectively as artificial biochemical networks (ABNs).

Traditionally, robotic control has been a popular application area for ABNs [7, 17, 19]. In principle, it reflects one of the main biological roles of biochemical networks: maintaining correct behaviour when exposed to a complex, dynamic, environment. However, high overheads mean that it is generally not feasible to use large populations or to carry out statistically significant numbers of trials, limiting the use of robotic control as a testbed for studying ABNs. In this paper, we take a different approach: we use ABNs to control numerical dynamical systems. As a testing environment, this has a number of advantages: numerical simulation is relatively fast, the dynamical properties are highly configurable, and test conditions can be replicated between experiments. Control of dynamical systems is also an important problem in its own right, having many applications in science and engineering [15], including those in robotics [3].

The paper is organised as follows: Section 2 introduces the dynamical systems addressed in this paper, Section 3 introduces ABNs, Section 4 describes our models and methodology, Section 5 presents results, and Section 6 concludes.

## 2 Dynamical Systems

A dynamical system [16] is any system whose subsequent state is determined by a function, or *evolution rule*, of the system's current state. Dynamical systems can be *discrete* or *continuous* time: the evolution rule can be described by a difference equation in the former, and a differential equation in the latter. Starting at a particular point within the system's state space, its *initial conditions*, the path that the system follows through its state space, its *trajectory*, is determined by iterating the evolution rule over a period of time. The set of possible trajectories within the state space are known as the *orbits* of the system. Following initial periods of wandering, *transients*, orbits may converge to limited parts of the state space known as *attractors*. Dynamical systems in which all orbits converge to one or more attractors are *dissipative*. Those which do not converge in this fashion are *conservative*.

Perhaps the most interesting class of dynamical systems are those which display the complex, unpredictable dynamics known as *chaos*. Two of the main hallmarks of a chaotic system are exponential sensitivity to initial conditions and strange attractors. The former entails that small changes in initial conditions can lead to wildly different trajectories through state space (a phenomenon popularly known as the *butterfly effect*), whereas the latter are the complex, typically fractal, regions of state space to which these chaotic orbits converge.

In the following sections, we introduce two dynamical systems, the Lorenz system and Chirikov's standard map. Both of these display chaotic behaviour, but otherwise lie at different ends of the classification spectrum: the former is a continuous dissipative system, the latter is a discrete conservative system.

### 2.1 The Lorenz System

The Lorenz system [13] is a continuous-time dynamical system whose behaviour is defined by the following set of differential equations:

$$\dot{x} = \sigma\,(y - x) \qquad \dot{y} = x(\rho - z) - y \qquad \dot{z} = xy - \beta z \tag{1}$$

For $\rho \gtrsim 24.74$, the Lorenz system displays chaotic behaviour, with all initial points attracted to a single two-lobed strange attractor (see Fig. 1) which orbits two unstable equilibrium points, which we term $\epsilon_+$ and $\epsilon_-$, located at:

$$\epsilon_+ = (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1) \qquad \epsilon_- = (-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho - 1) \tag{2}$$

The attractor consists of an infinite number of *unstable periodic orbits*. These are periodic in the sense that they orbit one or both of the fixed points a certain number of times before returning to roughly the same location. The orbits are unstable in the sense that trajectories will follow them for only a limited period of time before moving to another orbit. The dynamics of the system lead to trajectories which appear to flip unpredictably between the two lobes of the attractor.

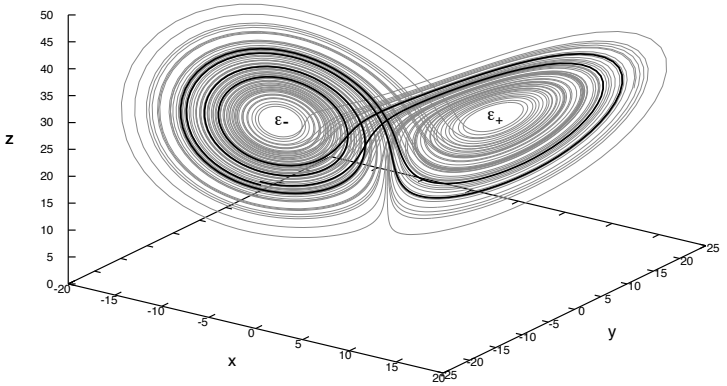



Fig. 1: A trajectory within the Lorenz attractor ($\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$), showing the location of the unstable equilibrium points $\epsilon_-$ and $\epsilon_+$. The heavy line shows one of the unstable periodic orbits followed by the trajectory.

### 2.2 Chirikov's Standard Map

Chirikov's standard map [5] describes a conservative discrete-time dynamical system which iteratively maps points within the unit square[4]:

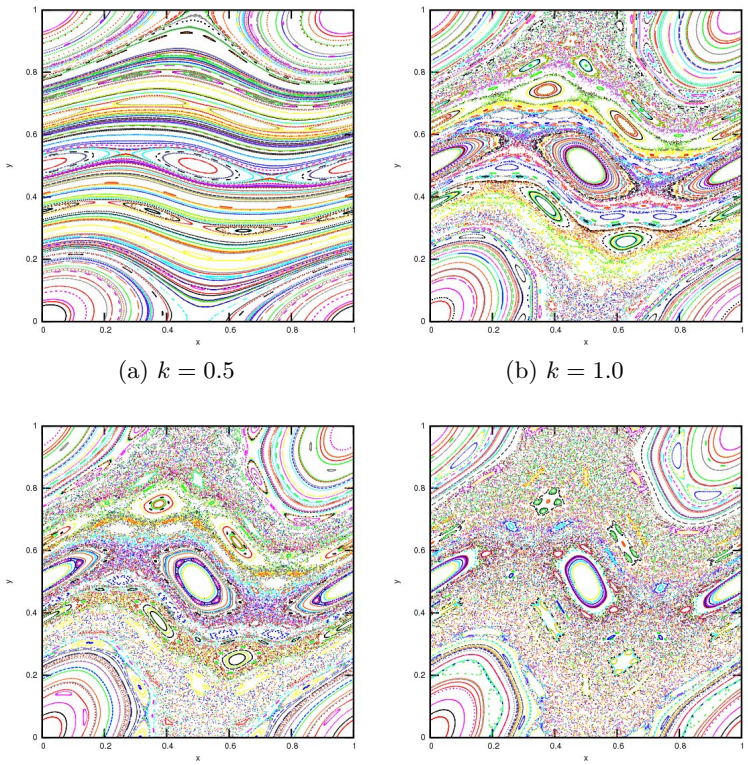
$$x_{n+1} = (x_n + y_{n+1}) \bmod 1 \qquad y_{n+1} = y_n - \frac{k}{2\pi} \sin(2\pi x_n) \tag{3}$$

The map's name follows from its ability to locally capture the behaviour of all systems with co-existing chaotic and ordered dynamics. For low values of $k$, the dynamics of the system are ordered, with initial points converging to cyclic orbits which remain bounded on the $y$ axis (see Fig. 2a). As $k$ increases, islands of chaotic dynamics begin to appear (see Fig. 2a–d). The map has a critical point at $k_c \approx 0.972$. For $k > k_c$, the chaotic islands are fully connected along the $y$ axis; meaning that, in principle, it is possible to follow a chaotic orbit from $y = 0$ to $y = 1$. However, the permeability of the central region increases only slowly as $k$ moves past $k_c$ [4] (see Fig. 2b–c). As an example of this, when $k = 1.1$, using 1000 randomly chosen initial points and an upper limit of $10^6$ iterations, we measured a median transit time of 64000 iterations of equation 3 to move from the bottom to the top of the map, with 27% of trajectories not reaching the target within the upper limit.

### 2.3 State Space Targeting

Chaos is found in many physical systems. The Lorenz system, for example, is a model of dynamical phenomenon seen in atmospheric, laser, and electronic systems [13]. Chaos leads to complex, unpredictable, behaviour; yet because of

[4] Following [15], we do not take the modulus of the $y$ co-ordinate, so $y$=0 and $y$=1 are not close, and the unit square as drawn recurs periodically along the $y$-axis.



(a) $k = 0.5$ (b) $k = 1.0$

(c) $k = 1.1$ (d) $k = 1.25$

Fig. 2: Sampled orbits of the standard map for various values of $k$, showing the transition from ordered to chaotic behaviour as $k$ increases. Each plot shows 200 trajectories of length 500, with the same set of initial conditions used for each plot.

its sensitivity to small perturbations, it is also inherently controllable. This has led to considerable interest in methods for controlling trajectories within chaotic dynamical systems, a process which is termed *chaos control*, or *chaos targeting*. This is achieved by adjusting one or more of the system's accessible parameters, with the goal of perturbing the trajectory towards a desired orbit or location within the state space. For dissipative systems such as the Lorenz system, chaos control can be achieved using methods such as OGY [14], which use small perturbations to prevent the trajectory from leaving a particular unstable periodic orbit. For conservative systems, such as Chirikov's standard map, control strategies are somewhat more complicated due to the heterogeneity of the state space. However, optimal targeting can be achieved by mapping the orbit structure and using perturbations to cross between different regions of the state space [4, 15]. For example, in [15], the authors calculate orbits which traverse the standard map in $\sim 125$ iterations when $k = 1.25$ and $\sim 600$ iterations when $k = 1.01$.

## 3 Artificial Biochemical Networks

The structure and function of biological organisms emerges from the orchestrated activities of the biochemical networks operating within individual cells. Broadly speaking, there are three types of biochemical network within a cell: (i) the *metabolic network*, which comprises the protein-mediated chemical reactions that take place within the cell; (ii) the *signalling network*, which represents the protein-mediated responses to chemical messengers received by the cell; and (iii) the *genetic regulatory network*, which determines the proteins that are present in a cell at any given time, and hence the structure of the metabolic and signalling networks. These three types of biochemical network are reflected by three classes of computational model:

**Artificial Genetic Networks (AGNs):** These model the regulatory interactions which occur between genes in biological cells. The canonical AGN model is the Boolean network (often referred to as a *random* Boolean network, or RBN). In Kauffman's [11] original model, an RBN consists of a set of genes, each of which is either fully on or fully off and whose state is determined by a Boolean function of other genes' states. In effect, they are a generalisation of binary cellular automata in which update rules can reference non-neighbouring cells and functions are heterogenous. In practice, computation can be achieved in the same way as cellular automata: by providing input via the initial activity state of the genes, running an appropriate network for a certain number of time steps, and then reading the output from the final activity states of the genes [7]. AGN models can also be constructed using continuous values for gene expression and continuous-valued regulatory functions [2, 12, 17].

**Artificial Metabolic Networks (AMNs):** The best known examples of metabolic-level models are artificial chemistries [1]. An artificial chemistry consists of a set of *chemicals*, a set of rules — which model the transformative agents (such as enzymes) found in natural systems — and an algorithm that determines when these rules are applied. Chemicals may be symbols to which some computational meaning can be associated [19], they may directly encode data structures, they may be overtly computational in nature, e.g. lambda-expressions [10], or they may even be other ABNs [8]. Likewise, rules vary from simple symbolic transformations to functional composition and complex structural modifications. By encoding inputs and outputs in the concentration, internal structure or positioning of chemicals, these artificial chemistries have been applied to a number of computational tasks, including robotics [19].

**Artificial Signalling Networks (ASNs):** In biology, signalling networks have the role of transducing environmental information to the metabolic and genetic networks. In addition to delivering the information to the correct spatial location(s), they are also responsible for integrating and pre-processing diverse incoming signals; a process which requires a host of cognitive activities [9]. Whilst

signalling networks have not yet received the same level of interest as the other classes of biochemical network, artificial signalling networks do show potential as a computational model [6].

## 4 State Space Targeting with ABNs

In this section, we report our work on using artificial biochemical networks to carry out state space targeting in the Lorenz system and Chirikov's standard map. We use two ABN models: an artificial genetic regulatory network and an artificial metabolic network. The AGN is a continuous-valued version of the Boolean network model. The AMN is an artificial chemistry with continuous-valued chemicals and transition rules. To allow meaningful comparison, both are deterministic and use synchronous updates.

**The artificial genetic network** (AGN) consists of an indexed set of genes, each of which has an expression level, regulatory inputs, and a regulatory function which maps the expression levels of its regulatory inputs to its own expression level. Formally: AGN $=< G, L_G, I_G, O_G >$, where:

$\boldsymbol{G}$ is the indexed set of genes $\{g_0, ..., g_n : g_i =< \lambda_i, R_i, f_i >\}$, where:
  $\boldsymbol{\lambda_i} : \mathbb{R}$ is the expression level of a gene.
  $\boldsymbol{R_i} \subseteq G$ is the set of regulatory inputs used by a gene.
  $\boldsymbol{f_i} : R_i \rightarrow \lambda_i$ is a gene's regulatory function.

$\boldsymbol{L_G}$ is an indexed set of initial expression levels, where $|L_G| = |G|$.
$\boldsymbol{I_G} \subset G$ is the set of genes used as external inputs.
$\boldsymbol{O_G} \subset G$ is the set of genes used as external outputs.

The AGN is executed as follows:

G1. $\lambda_0 ... \lambda_n$ are initialised from $L_G$ (if AGN not previously executed).
G2. Expression levels of enzymes in $I_G$ are set by the external inputs.
G3. At each time step, each gene $g_i$ applies its regulatory function $f_i$ to the current expression levels of its regulating genes $R_i$ in order to calculate its expression at the next time step, $\lambda_i'$.
G4. After a certain number of time steps, execution is halted and the expression levels of enzymes in $O_G$ are copied to the external outputs.

**The artificial metabolic network** (AMN) comprises an indexed set of enzyme-analogous elements which transform the concentrations of an indexed set of real-valued chemicals. Each enzyme has a set of substrates, a set of products, and a mapping which calculates the concentrations of its products based upon the concentrations of its substrates. Formally: AMN $=< C, E, L_C, I_C, O_C >$, where:

$\boldsymbol{C}$ is the indexed set of chemical concentrations $\{c_0, ..., c_n : \mathbb{R}\}$.
$\boldsymbol{E}$ is the indexed set of enzymes $\{e_0, ..., e_n : e_i =< S_i, P_i, m_i >\}$, where:

$\boldsymbol{S_i} \subseteq C$ is the set of chemicals used by the enzyme (*substrates*).
$\boldsymbol{P_i} \subseteq C$ is the set of chemicals produced by the enzyme (*products*).
$\boldsymbol{m_i} : \mathbb{R}^n \to \mathbb{R}^n$ is the enzyme's substrate-product mapping.

$\boldsymbol{L_C}$ is an indexed set of initial chemical concentrations, where $|L_C| = |C|$.
$\boldsymbol{I_C} \subset C$ is the set of chemicals used as external inputs.
$\boldsymbol{O_C} \subset C$ is the set of chemicals used as external outputs.

Execution of the AMN is similar to that of the AGN:

M1. $C$ is initialised from $L_C$ (if AMN not previously executed).
M2. The concentrations of chemicals in $I_C$ are set by the external inputs.
M3. At each time step, each enzyme $e_i$ applies its mapping $m_i$ to the current concentrations of its substrates $S_i$ in order to determine the new concentrations of its products $P_i$. Where the same chemical is produced by multiple enzymes, i.e. when $\exists j, k : j \neq k \wedge c_i \in P_j \cap P_k$, the new concentration is the mean output value of all contributing enzymes.
M4. After a certain number of time steps, execution is halted and the concentrations of chemicals in $O_C$ are copied to the external outputs.

We also look at the effect of applying a *mass conservation law*, such that the sum of chemical concentrations remains constant over time. This more closely reflects biological systems, where mass balance results in indirect regulatory interactions between chemical reactions. It is implemented by uniformly scaling concentrations at the beginning of step M3 so that $\sum_{c_i \in C} c_i = 0.5|C|$.

### 4.1 Mappings

Regulatory ($f_i$) and enzyme ($m_i$) mappings are chosen from three parameterisable functions: a Sigmoid, the Michaelis-Menten equation, and the logistic map.

**Sigmoids** are often used to model the switching behaviour of non-linear biological systems, and are therefore a natural choice for ABNs. For this, we use the logistic function $f(x) = (1 + e^{-sx-b})^{-1}$, where $s \in [0, 20]$ determines the slope and $b \in [-1, 1]$ the slope offset (or *bias*). For multiple inputs, $x = \sum_{j=0}^{n} i_j w_j$, where $i_0 ... i_n$ are inputs and $w_0 ... w_n \in [-1, 1]$ are corresponding input weights (negative values indicating repression).

**The Michaelis-Menten equation** defines the kinetics of enzyme-mediated reactions, making it particularly interesting from the perspective of AMNs. It is a hyperbolic function, $f(x) = vx(k + x)^{-1}$, where $v \in [0, 1]$ is the asymptotic value of the output and $k \in [0, 1]$ determines the slope. For multiple inputs, $x = \sum_{j=0}^{n} \frac{i_j w_j}{n}$, truncating negative values.

**The logistic map** is a discrete dynamical system with both ordered and chaotic regimes, defined $f(x) = rx(1-x)$, where $r \in [0, 4]$ determines whether the system exhibits a periodic or chaotic orbit. In effect, we are interested in whether evolved ABNs can make use of 'pre-packaged' dynamics.

### 4.2 Methods

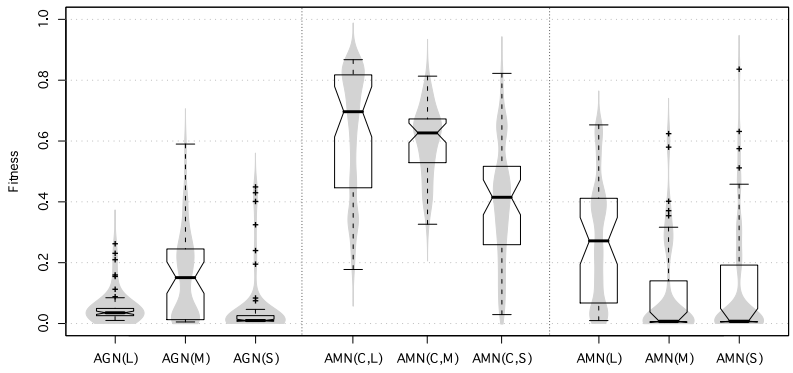
ABNs are evolved using a standard generational evolutionary algorithm with tournament selection (size 4), uniform crossover (p=0.15), and point mutation (p=0.06). An ABN is represented as an array of genes ($G$) or enzymes ($E$), an array of initial values ($L_G$ or $L_C$), and an integer in the range $[1, 100]$ specifying the number of time steps for execution. To simplify analysis, the number of genes, enzymes and chemicals are fixed at 10. Crossover points always fall between gene or enzyme boundaries. Inputs and outputs ($R_i, S_i$ and $P_i$) are represented by absolute references to array positions. Function parameters (e.g. slopes, input weights) and initial values are represented as floating-point values and are mutated using a Gaussian distribution centred around the current value.

For both problems, the ABN is provided with the current state space location at the start of execution and outputs the new value for a specified control parameter at the end of execution. Inputs are copied into the lowest-numbered genes or chemicals and the output is taken from the highest numbered. All runs are terminated after 50 generations.

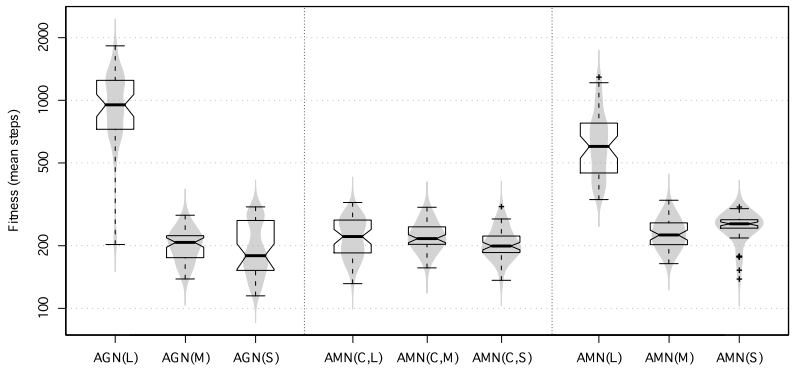
**Lorenz system:** The goal is to find an ABN-based controller which can (i) stabilise the system at its equilibrium points and (ii) move between the two points as required. This is tested by requiring it to move from $\epsilon_-$ to $\epsilon_+$ and remain there for 5000 time steps, then return to $\epsilon_-$, remaining for another 5000 time steps. In order to do this, the ABN is allowed to modulate the Rayleigh parameter, $\rho$ (see Equ. 1), within the range [0,100]. For inputs, the ABN is given the current location, $< x, y, z >$ (values scaled from $[-50, 50]$ to $[0, 1]$), and the distance to the target, $d$. To make the problem more challenging, exact distance is given only when the Euclidean distance to the target $E < 2.0$. Above this, it is set to the maximum input value, i.e. $d = \min\{1, \frac{E}{2}\}$. The ABN generates a single output, the new value of $\rho$. The Lorenz equations are numerically integrated using the fourth-order Runge-Kutta method with a step size of $\Delta_t = 0.01$. The ABN is executed every 10 steps to get a new value of $\rho$. For $s$ time steps, fitness is $\frac{\sum_s 1-d}{s}$, rewarding stability at equilibrium points and short transients. A population size of 500 is used (found to be suitable through trial-and-error).

**Standard map:** Following the examples of [4] and [15], the goal is to find a controller which can navigate from the bottom to the top of the standard map in the shortest number of steps. In order to do this, the ABN is allowed to modulate parameter $k$ in Equ. 3 within the range $[1.0, 1.1]$. We use the same initial and target regions (shown in Fig. 5) as used in [15]. Inputs to the RBN are the current position $< x, y >$ and the Euclidean distance from the top-centre of the map, and the single output is the new value of $k$. The evolved ABNs are evaluated on 20 random points within the initial region. Fitness is the mean number of steps required for these trajectories to reach the target region. Trajectories which do not reach the target region within 1000 steps are assigned an arbitrary figure of 2000 steps, biasing search towards controllers effective over all initial conditions. A population size of 200 is used.

## 5 Results



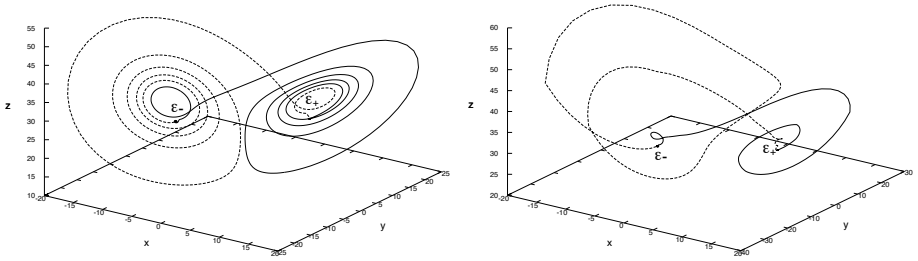

(a) Lorenz system. High numbers are better.




(b) Standard map. Low numbers are better.

Fig. 3: State space targeting using evolved ABNs with (C)onserved mass, (S)igmoids, (M)ichaelis-Menten equations and (L)ogistic maps. Summary statistics of 50 runs are shown as notched box plots. Overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the 95% confidence level. Kernel density estimates of underlying distributions are also given (in grey), showing that some of the fitness distributions are multimodal.

Results for the Lorenz system and the standard map are shown in Fig. 3. Effective controllers were found for both problems (see Figs. 4 and 5 for examples). However, the two problems appear to need quite different ABN models: for the Lorenz system, best performance comes from AMNs with conserved mass and logistic maps; whereas best performance on the standard map comes from AGNs with Sigmoid functions. Notably, the best solution for one problem is the worst solution for the other; although this is perhaps unsurprising, given that the two problems lie at opposite ends of the dynamical systems spectrum.




(a) Michaelis-Menten. (b) Logistic map.

Fig. 4: Example behaviours of evolved controllers moving between unstable points in the Lorenz system via control of the Rayleigh parameter. Both controllers use AMNs with conserved mass. (a) The Michaelis-Menten AMN (fitness 0.821) moves from $\epsilon_-$ to a distance within 1.0 of $\epsilon_+$ in 447 steps (broken line), returning to $\epsilon_-$ in 454 steps (unbroken line). (b) The logistic map AMN (fitness 0.872) does it in 182 and 170 steps, respectively.

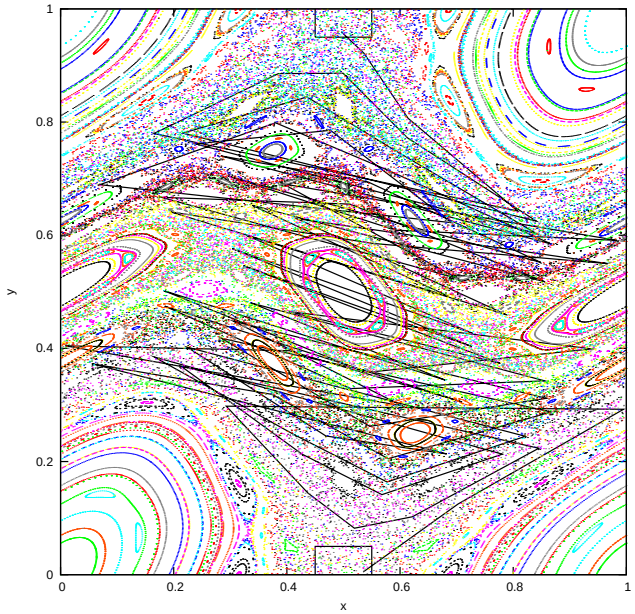



Fig. 5: Example behaviour of an evolved controller guiding a trajectory from a region at the bottom of the standard map, (0.45,0)→(0.55,0.05), to a region at the top, (0.45,0.95)→(0.55,1), in 83 steps. The standard map is plotted for $k = 1.1$.

Perhaps the most interesting result is the positive effect of conserving mass in the AMN model. In the Lorenz system, this effect is particularly significant and appears critical for good performance. The effect of mass conservation is to force covariance between the chemical concentrations. For example, if the concentration of one chemical increases, the concentrations of all other chemicals are (stoichometrically) decreased. We can hypothesise that this causes a reduction in the number of effective variables of the system, reducing the search effort required to find viable solutions. It also suggests the potentially important role that conservation laws may play in reducing the effective complexity of more biologically-realistic models.

Function choice also has interesting consequences: Michaelis-Menten equations work relatively well on both problems, Sigmoids work well on the standard map, but not so well on the Lorenz system, and logistic maps work well on the Lorenz system but generally perform poorly on the standard map. Whilst the continuous functions offer more consistent performance across the problems, the logistic map did lead to the best overall solutions for the Lorenz system. As demonstrated by Fig. 4, these achieve short paths using eccentric orbits, whereas Michaelis-Menten solutions tend to follow smoother paths. It seems likely that the multimodal, wide distributions of the AMNs using logistic maps reflect a trade-off between expressiveness and evolvability.

## 6 Conclusions

In this paper, we have demonstrated how evolved artificial biochemical networks (ABNs) can be used to control complex dynamical systems. Our results are promising, showing that relatively simple ABNs are capable of state space targeting within both the Lorenz system and Chirikov's standard map — numerical systems located at opposite ends of the dynamical systems spectrum. Notably, our results on Chirikov's standard map are broadly similar to the analytical methods described in [4] and [15], but without requiring prior knowledge of the system's orbital structure. This supports the notion that ABNs can be used to represent complex computational behaviours in evolutionary algorithms, such as genetic programming, which evolve executable structures.

More generally, we have introduced the notion that dynamical systems are a useful domain for studying ABNs. Our results support their use in comparing the properties of different ABN models, illustrating that different models are suited to different problems. In particular, we have shown the benefit of using a conservation rule in the perturbation of ABN variables to introduce covariance (e.g. masses in the AMN), highlighting the important role constraints may play in guiding search towards viable solutions. Our results also show the sensitivity of ABNs to function choice, and that iterative maps can provide a useful source of pre-packaged dynamics in certain situations.

In future work, we plan to investigate a broader collection of ABN models and dynamical systems, to analyse how the ABNs solve these problems, and to look at how issues of evolvability and representation affect their ability to do so.

## References


1. Banzhaf, W.: Artificial chemistries—towards constructive dynamical systems. Solid State Phenomena 97/98, 43–50 (2004)
2. Banzhaf, W.: Artificial regulatory networks and genetic programming. In: Riolo, R.L., Worzel, B. (eds.) Genetic Programming Theory and Practice, chap. 4, pp. 43–62. Kluwer (2003)
3. Beer, R.D.: Beyond control: The dynamics of brain-body-environment interaction in motor systems. In: Sternad, D. (ed.) Progress in Motor Control V: A Multidisciplinary Perspective, pp. 7–24. Springer (2009)
4. Bollt, E.M., Meiss, J.D.: Controlling chaotic transport through recurrence. Physica D: Nonlinear Phenomena 81(3), 280–294 (1995)
5. Chirikov, B.V.: Research concerning the theory of nonlinear resonance and stochasticity. Tech. rep., Institute of Nuclear Physics, Novosibirsk (1969)
6. Decraene, J., Mitchell, G.G., McMullin, B.: Evolving artificial cell signaling networks: Perspectives and methods. In: Dressler, F., Carreras, I. (eds.) Advances in Biologically Inspired Information Systems, pp. 167–186. Springer (2007)
7. Dellaert, F., Beer, R.D.: A developmental model for the evolution of complete autonomous agents. In: Maes, P., et al. (eds.) From Animals to Animats 4: Proc. 4th Int. Conf. Simulation of Adaptive Behavior. MIT Press, Cambridge, MA (1996)
8. Faulconbridge, A., Stepney, S., Miller, J.F., Caves, L.S.D.: RBN-World: A subsymbolic artificial chemistry. In: Proc. ECAL 2009. LNCS, Springer (2009)
9. Fisher, M.J., Paton, R.C., Matsuno, K.: Intracellular signalling proteins as 'smart' agents in parallel distributed processes. BioSystems 50, 159–171 (1999)
10. Fontana, W.: Algorithmic chemistry. In: Langton, C.G., Taylor, C., Farmer, J.D., Rasmussen, S. (eds.) Artificial Life II, pp. 159–210. Addison-Wesley (1992)
11. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22(3), 437–467 (Mar 1969)
12. Kumar, S.: The evolution of genetic regulatory networks for single and multicellular development. In: Keijzer, M. (ed.) GECCO 2004 Late Breaking Papers (2004)
13. Lorenz, E.N.: Deterministic nonperiodic flow. Journal of the Atmospheric Sciences 20(2), 130–141 (March 1963)
14. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaos. Phys. Rev. Lett. 64(11), 1196–1199 (Mar 1990)
15. Schroer, C.G., Ott, E.: Targeting in Hamiltonian systems that have mixed regular/chaotic phase spaces. Chaos 7, 512–519 (December 1997)
16. Stepney, S.: Nonclassical computation: a dynamical systems perspective. In: Rozenberg, G., Bäck, T., Kok, J.N. (eds.) Handbook of Natural Computing, vol. 2, chap. 52. Springer (2009)
17. Taylor, T.: A genetic regulatory network-inspired real-time controller for a group of underwater robots. In: Groen, F., et al. (eds.) Intelligent Autonomous Systems 8 (Proceedings of IAS8). pp. 403–412. IOS Press, Amsterdam (2004)
18. Zhan, S., Miller, J.F., Tyrrell, A.M.: An evolutionary system using development and artificial genetic regulatory networks. In: Wang, J. (ed.) 2008 IEEE CEC. IEEE Press (2008)
19. Ziegler, J., Banzhaf, W.: Evolving control metabolisms for a robot. Artificial Life 7, 171–190 (2001)