

Emergent Organisation in Colonies of Simple Automata

I.W. Marshall and C.M. Roadknight

B54/124, Aadastral Park, Martlesham Heath, Ipswich, Suffolk, UK. IP5 7RE
ian.w.marshall@bt.com, christopher.roadknight@bt.com

Abstract. We have simulated a colony of approx 3000 simple automata, with interaction behaviours that attempt to emulate those observed in stromatolite building microbes. The colony is able to maintain a diverse genome that can metabolise large quantities of up to 100 different nutrient molecules simultaneously. Group organisations that process short lived molecules more rapidly than others readily emerge under the correct conditions. We have applied the lessons to the control and management of a distributed communications system.

Introduction

Stromatolites [1] are rock mounds built by heterogeneous colonies of bacteria [2]. Colonies of this type can be found today in Shark Bay, on the coast of Western Australia. Fossil colonies can also be found throughout earth history – currently the oldest known was found in rocks dated at 3.5Ga, also in Western Australia [1]. The colonies exhibit sophisticated internal organization that emerges from interactions between individual colony members. In comparison with higher organisms such as ants and termites, the interactions are relatively simple. It is thus possible that they could be completely understood and modeled. In the work we report here, we have made no attempt to model a realistic colony. Instead we have taken the known interaction types from microbial colonies, and modeled a colony of autonomous automata that exhibit similar interaction styles. The aim was to assess whether the known interactions are sufficient to produce stable colonies with useful emergent properties. Our ultimate objective is to discover simple and effective ways of enabling self-configuration of large-scale complex structures, such as next generation communication networks. Previously [3] we have described how the colony can evolve and how this could be applied. As an illustration of the benefits of emergent colony behaviours, we have also demonstrated how the colony can be used to generate differentiated quality of service in an autonomous and emergent manner [4]. In this paper we focus on the relationship between our simulated colony and the interactions in real colonies, and show for the first time that our colony exhibits emergent structure similar to that found in real stromatolite building communities. The structure enables the colony to metabolise available food sources extremely efficiently. We go on to demonstrate how this can be used to balance system load in a computational environment

Interaction Model

Bacteria have no differentiated sensory organs, so interaction is necessarily limited. Our simulation allows for limited motility (as exhibited by *Spirochaetes*) and the resultant physical interaction between individuals attempting to occupy the same space, plasmid interchange (used in previous bacteria inspired genetic algorithms [5,6]), and chemical exchanges via the environment. The chemical exchanges contain no explicit messages from one individual to another. The interaction is simply that if an individual metabolises a chemical it reduces the local concentration of that chemical, and increases the local concentration of the metabolite that it excretes. These changes in turn affect the behaviour of near neighbours, and the colony wide diffusion gradients can lead to rich behavioural patterning [7]. Chemical diffusion is modeled by treating molecules as packets, and routing the packets across a dense grid of 4 port switches, always away from their source (where concentration is highest). This allows the diffusion model to be discrete, and easily combined with the individual automata we use to model the microbes. It also makes application of the results into a communication network very straightforward. We believe that our approach to modeling chemical interaction is unique. Motility is constrained to slow movement along chemical gradients, where movement is not blocked by other microbes, and is thus much less significant than in models based on tools such as SWARM [8]. The microbes evolve using a modified version of the distributed GA we described in earlier work [3] that uses plasmid interchange to play the role of crossover. The modifications make the GA completely decentralized, and are presented here for the first time.

Experimental Details

The bacterial evolution algorithm has already been discussed in several papers [3,4]. To summarise, our model randomly places automata (with random initial genomes) at some of the vertices of a 2-dimensional grid that forms their environment. The genome of each automaton (crudely simulated bacterium) contains plasmids that enable the host to metabolise different kinds of available nutrients, and earn a reward (energy) for each molecule metabolised. For the purposes of this research a plasmid is the specific software that requires a certain nutrient plus a set of subjective rules as to whether the nutrient should be accepted (eg. Software A will be run if the unit is less than X% busy and has a queue smaller than Y). There are up to 100 different nutrient molecules. A random subset of these molecules is injected at each vertex along one side of the grid, at the beginning of every epoch (an epoch is 100 timesteps or 1 hop). The size of the injected subset is a variable of the model, and can frequently be close to zero. Once injected, nutrient molecules are forwarded (away from or parallel to the side they entered) automatically from vertex to vertex until a vertex is reached with a resident 'bacterium'. If a suitable gene is present, and the bacterium has not already got enough nutrients, the molecule is metabolized. Otherwise it is forwarded after a small delay. Each movement from one vertex to another is referred to as a hop. The bacteria can maintain a 'queue' of up to 100 nutrient molecules before they are forced to forward nutrients. If a lot of molecules are decaying in the queue the bacterium can move one place up the grid if there is a

vacant vertex. If no molecules are decaying the bacterium can move one place down the grid if there is a vacant vertex. They communicate with neighbours immediately above and below, so swaps are also possible. The time it takes to process 4 molecules is referred to as an epoch (about 1 hop). The colony evolves through the exchange and mutation of plasmids. Bacteria that accumulate sufficient energy replicate. Bacteria that do not accumulate energy die. It is possible for some individuals to survive without ever replicating, but they must do some metabolic processing, since energy is deducted each cycle (to simulate respiration).

The network simulation here differs to the previous versions in that the colony is partially subdivided. Nutrient molecules are forwarded with some randomised lateral movement as before, but when forwarding would take the nutrient beyond the bottom of a sub-colony of microbes the nutrient is forwarded to the top of another sub-colony. Routing down the colony has also been changed slightly to speed up the diffusion process. Molecules can now be forwarded to the next OR the next but one layer. These changes enable us to spread the model across several physical computers and increase the size of colony that can be modeled real time, since links between the concentrations are less rich and slightly slower than within concentrations. Our test topology had 6 semi-independent sub-colonies. The 6 sub-colonies are connected in a star formation (1 forwards to 2 and 4, 2 forwards to 1 and 5, 3 forwards to 4 and 6, 4 forwards to 3 and 1, 5 forwards to 6 and 2 and 6 forwards to 5 and 3). Each sub-colony runs on identical hardware and has the same available resources at its disposal.

To allow the automata to be truly autonomous, as they would be in a real colony, plasmids are no longer obtained from a globally accessible gene pool by stressed individuals. Instead randomly chosen plasmids attach to molecules that are being forwarded, in line with the following rules.

1. If an individual FAILS to process a molecule (and so must forward it) AND a random number between 0-100 is less than the no. of nutrient molecules it has already stored for later processing (max 100) THEN a 'plasmid' from its genome is attached to the request. In simpler terms; if do not process this item but I AM busy then attach a plasmid from my genome to the item.
2. If an individual succeeds OR fails to process a food item AND its busyness (plus a constant, currently 5%) is less than a random number between 0-100 THEN take a 'plasmid' from the 'plasmid attachment' space, if there is one. In simpler terms; regardless of if I process the item or not, if I am NOT BUSY then take a 'plasmid' from the request if there is one.

For the experiments, groups of 25% of the different nutrient molecules were given lifetimes of 2, 5, 10 and 20 hops respectively, and the injection rate was around 25% of the maximum allowed by the model. This is enough to allow around 50% of the vertices to be inhabited. The processing efficiency is compared to a simulation in which evolution was blocked by stopping plasmid interchange, so the initial random gene distribution persists

Results

The evolution of one of the six sub-colonies is illustrated in Fig 1 below. The unlabelled microbes are generalists, the marked microbes are specialists adapted to

handle foods with particular lifetimes (in this sub-colony) as follows; 1 =2 hops or less, 2 = 2-5 hops , 3 = 5-10 hops, 4 = 10-20 hops. The snapshots are taken after 15, 50, 200, 400, 800 and 1600 epochs. Food is being injected at the top of the colony. A substantial degree of layering is clear after 200 epochs, with individuals adapted to metabolise short lived nutrients concentrating closest to the source. This slowly improves as the colony continues to evolve.

To illustrate the benefit of the limited motility Figure 2 shows the sub-colony after 1600 epochs of an equivalent simulation when no movement was allowed. The structure is similar but somewhat less organized, and it took longer for the structure to emerge. This is because without motility it is hard for genomes adapted to process the short lived nutrients to supplant those adapted to more persistent nutrients residing at the top of the colony.

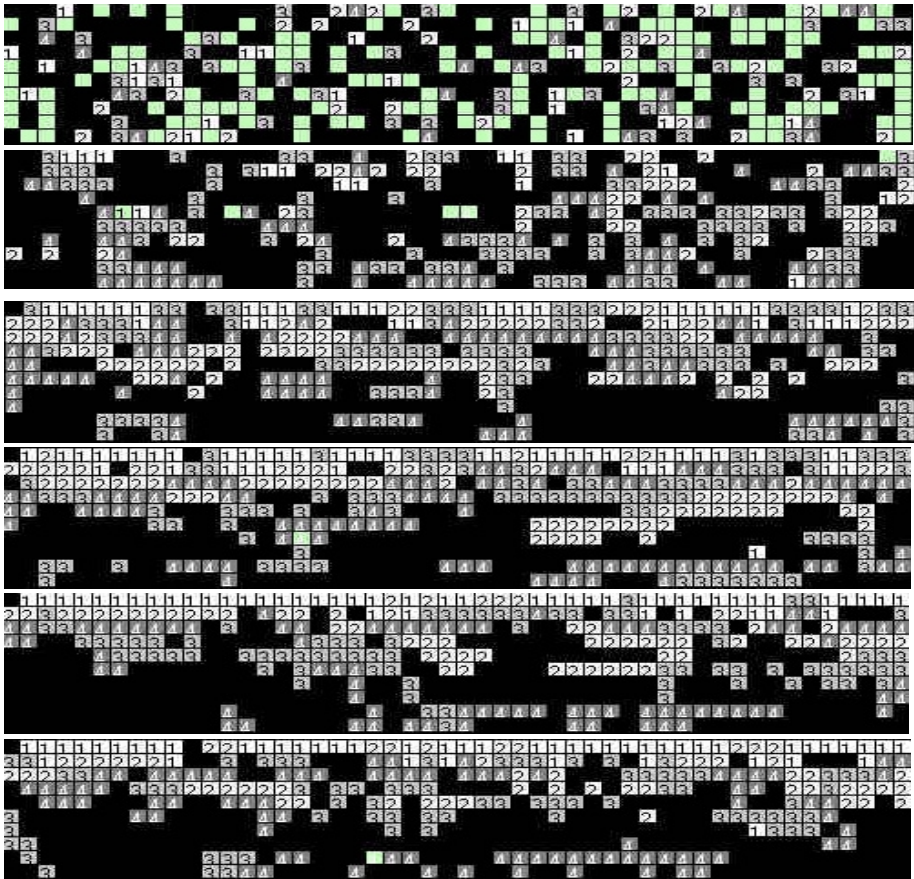


Fig. 1. Emergence of layering in a 500 member sub-colony. Initial state is at the top, images down the page are at successively later stages in the evolution of the colony (see text for details).

To illustrate the temporal evolution of the colony characteristics we show the consumption efficiency of the colony against simulated time in figure 4 (for the same conditions as in figure 3). The evolved colony is compared with a colony that cannot evolve. This comparison is valid as current networks are non-adaptive, a comparison with other adaptive techniques would be useful but is beyond the scope of this paper. As might be expected the properties of the random colony do not change with time. In the evolved case however, 2 timescales of evolution are apparent. A fast initial transient due to the sharing of initial plasmids (plasmid learning) and a slow improvement due to the continuing evolution and sharing of novel plasmids via random mutations. Reassuringly the evolved colony performs rather better (80%) than the random colony (70%) with far less living individuals (1000 evolved, 2000 random). Perhaps more significantly when a new nutrient is introduced to the evolved colony it adapts, whereas the random colony does not. The graph does not show this since the new nutrient was only 1% of the total input and the small performance changes cannot be resolved at this scale.

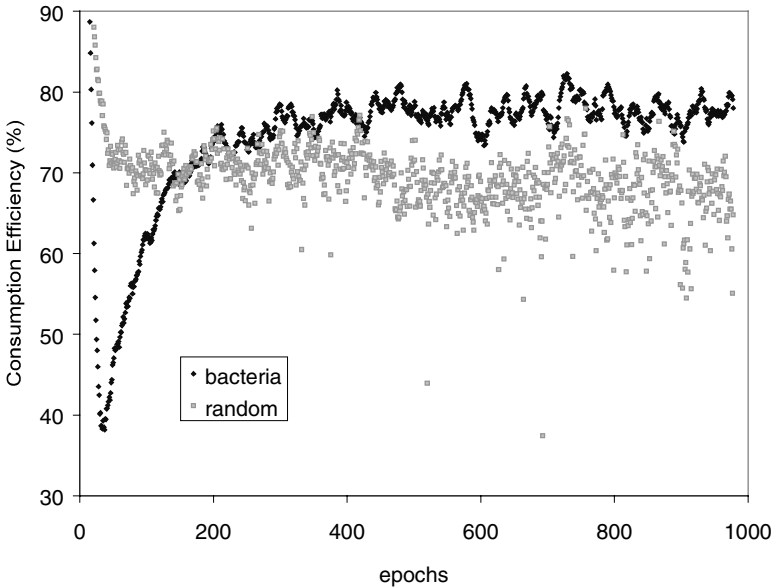


Fig. 4. Temporal evolution of the colony, comparing evolved efficiency with efficiency of a random distribution (averaged over 6 colonies)

The evolved colony also copes well with uneven distributions of the injected nutrients. In figure 5 we show the effect of progressively concentrating nutrient input into fewer sub-colonies. The x-axis has an even distribution of input (at 80% max as in figures 3 and 4) into the top edge of all six sub-colonies at the left. The figures shown are the average of all 6 members of the network. The input is concentrated until the supply to one colony is the max possible, and the other colonies are fed so that the total nutrient supply remains constant. It is clear that at all times the evolved colony is more effective than the random colony. This final graph is particularly interesting since the colony is behaving in exactly the way we would like a good distributed load balancing algorithm to. It is spreading load across available

resources, ensuring a high probability of success (comparable to real computing systems at high load), whilst minimizing the resources required. We are confident that by appropriate adjustment of the simulation parameters the success rate could be increased to any desired value, and required resources would always be less than for a static distribution of resources.

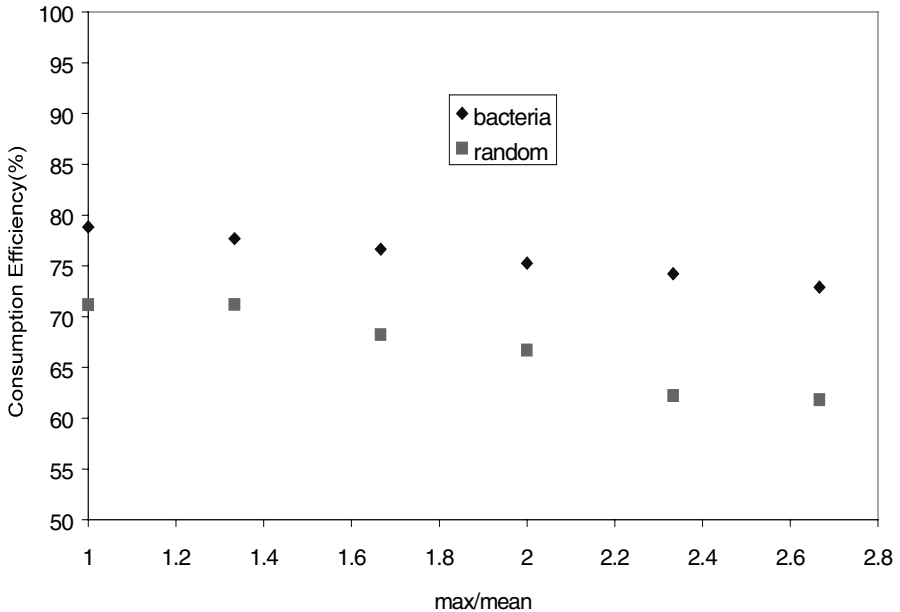


Fig. 5. Consumption efficiency of evolved colony and random colony for a range of load distributions

Discussion

The evolution of colony structure, and colonial learning that we have observed is consistent with the thinking and observations of Shapiro [9,10]. Colonies of symbol processing devices have also been investigated in the grammar system area [11]. It is our hope that this work will inspire radical new approaches to decentralized control in distributed systems, and also shed light on the fundamental issues in microbial genetics raised by Shapiro and others. Our objectives are thus similar to those of the vast majority of a-life researchers. Our methods however are a little different. In particular, the choices in our model are driven by the needs of distributed systems applications rather than a desire to emulate biology. The packet network model for distribution is one example of this, that builds on our background in networking. We believe that this approach is a productive way of avoiding the complexities of true biological versimilitude, whilst at the same time enabling us to build simulations that do more than simply replay sequences of events based on current (often highly simplified) biological knowledge. In other words our model has emergent properties because we have built into it some detailed knowledge of a real system (the Internet)

with emergent properties, that is nevertheless probably considerably simpler than a real biological system.

Conclusions

We have demonstrated that enabling individuals in a distributed individual based evolutionary colony simulation to interact via a simple discrete distribution model is sufficient to produce emergent colony properties. Adding limited motility speeds up the self-organisation and stabilizes it somewhat, but does not alter the qualitative properties of the organized colony. The organized colony is a highly efficient consumer of a wide range of resources with different lifetimes. Control algorithms based on our observations can be applied to load balancing and QoS in distributed computing systems.

References

- [1] S.Golubic "Stromatolites of Shark Bay" pp103-149 in Environmental evolution ed. Margulis and Olendzenski, MIT press 1999
- [2] S.Sonea and M.Panisset, "A new bacteriology" Jones and Bartlett, 1983.
- [3] I.W.Marshall and C.Roadknight, "Adaptive management of an active services network", *British Telecom. Technol. J.*, 18, 4, pp78-84 Oct 2000
- [4] I.W.Marshall and C.Roadknight "Provision of quality of service for active services" *Computer Networks*, April 2001
- [5] I.Harvey, "The Microbial Genetic Algorithm", unpublished work, 1996, available at <ftp://ftp.cogs.susx.ac.uk/pub/users/inmanh/Microbe.ps.gz>
- [6] N.E.Nawa, T.Furuhashi, T.Hashiyama and Y.Uchikawa, "A Study on the Relevant Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm" *Proc IEEE Int Conf. on evolutionary computation* 1997
- [7] E.Ben-Jacob, I.Cohen and H.Levine "Cooperative self-organisation of microorganisms" *Advances in Physics*, 49, 4, 395-554, 2000
- [8] G.Booth, "Gecko: A Continuous 2D World for Ecological Modeling", *Artificial Life*, 3, pp. 147-163, Summer 1997.
- [9] J.A.Shapiro, "Thinking about bacterial populations as multicellular organisms" *Ann. Rev. Microbiol.* 52, 81-104. 1998
- [10] J.A.Shapiro, "Genome system architecture and natural genetic engineering in evolution", In *Molecular Strategies in Biological Evolution*, L. Caporale, ed., *Annal. NY Acad. Sci.* 870, 23-35 1999
- [11] E. Cshahj-Varju, J. Dassow, J. Kelemen, G. Paun, *Grammar Systems*, Gordon and Breach, London, 1994