

Emergent organisation in colonies of simple automata

I. W. Marshall and C. M. Roadknight

B54/124, Adastral Park, Martlesham Heath, Ipswich, Suffolk, UK. IP5 7RE

Christopher.roadknight@bt.com, ian.w.marshall@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 food types simultaneously. Group organisations that process short lived food types 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 chemicals 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:

The network simulation here differs to the previous versions in that the colony is partially subdivided. Food items are forwarded with some randomised lateral movement as before, but when forwarding would take the food beyond the bottom of a sub-colony of microbes the food is forwarded to the top of another sub-colony. This enables 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.

A further difference is that plasmids are no longer obtained from a globally accessible gene pool by stressed individuals. Instead plasmids attach to foodstuffs that are being forwarded, in line with the following rules.

1. If an individual FAILS to process a food item (and so must forward it) AND a random number between 0-100 is less than it's busyness (a percentage) THEN a 'plasmid' from it's genome is attached to the request.

2. If an individual 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.

3. If an individual SUCCEEDS in processing 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.

The use of Location Penalties is no longer necessary and has been removed. Though this penalty term was useful for managing the activity of nodes within the colony, in a spatial way, it's overheads (penalty based, rigid) proved unnecessary once the gene pool was decentralised. The lack of attached 'plasmids' on food items arriving at the colony is enough of a penalty to deter the blind migration of 'organisms' to the food source at the edge of the colony.

Routing down the colony has been changed slightly to speed up the diffusion process. Food items can now be forwarded to the next OR the next but one layer.

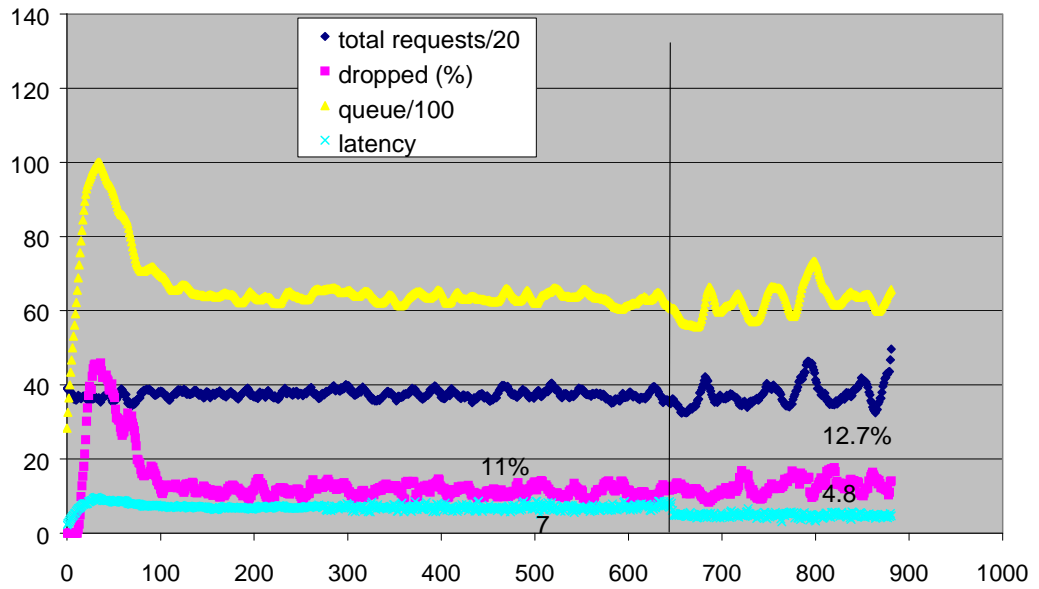
The QoS and service discrimination ability of an adaptive approach have been shown previously, so the performance results will focus on:

A: The ability of the algorithm to effectively load balance demands on a network of clusters.

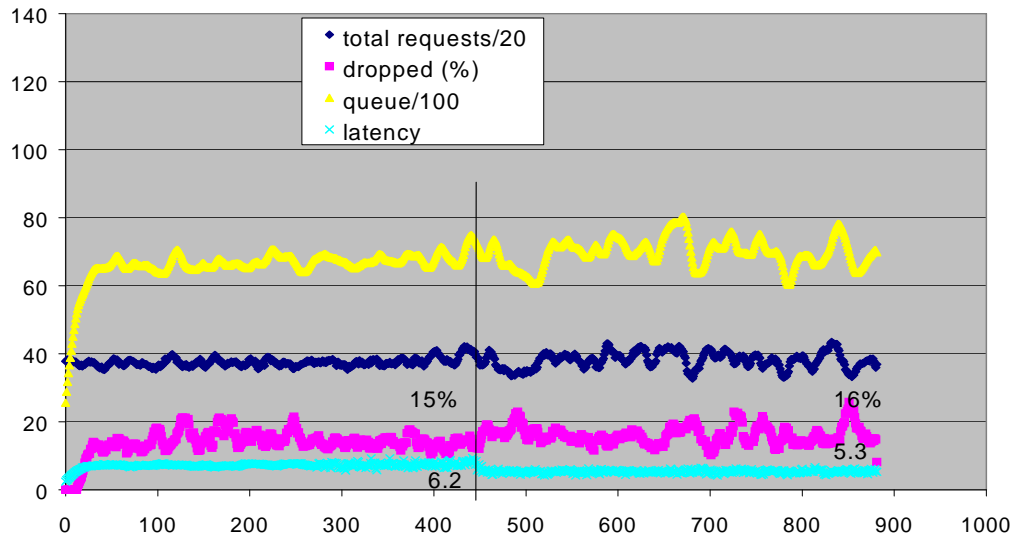
B: The ability to handle increasingly variant load.

Results

Bact

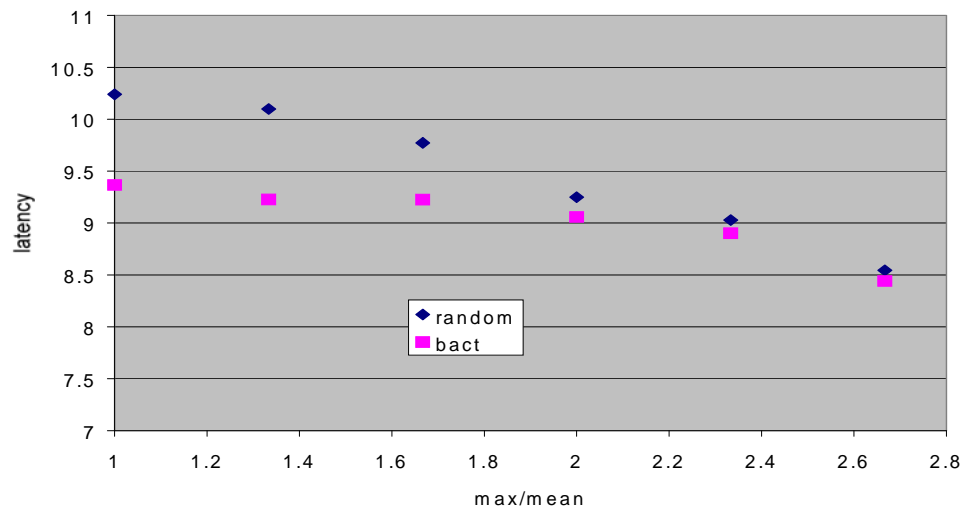
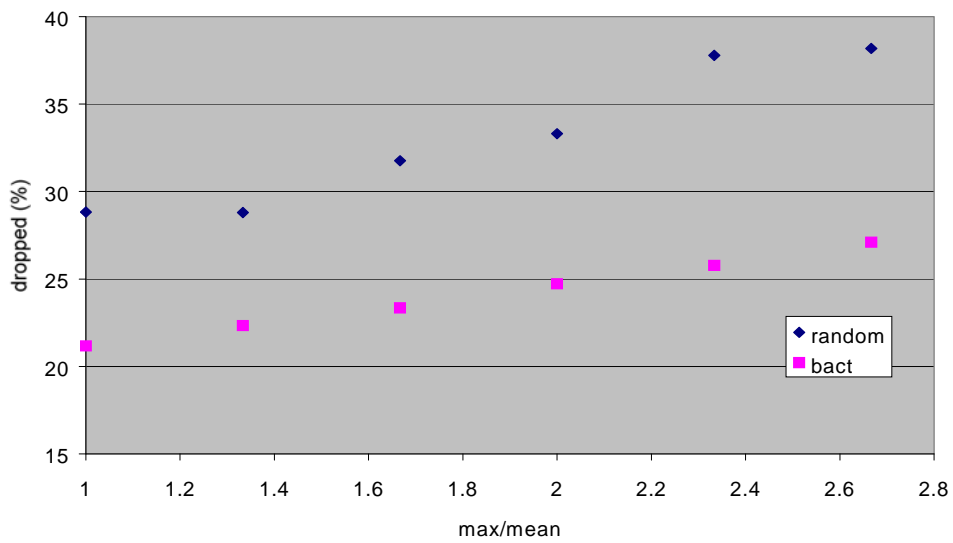


Rand



Load balancing.

One well defined task of a distributed network of any kind is to balance a load placed on the edge nodes equally around the network. This is not a primary task of the bacterial algorithm and there may be more efficient, dedicated load balancing tactics available. But the proposed bacteria inspired algorithm must be able to load balance as one of its assets. Figs xx shows how well a network of randomly configured nodes and bacteria algorithm configured nodes distribute load as it gets increasingly skew in favour of one of the nodes, both in terms of percentage of dropped requests (fig) and latency (figX)



It is apparent that there is little difference in this experiment between the random algorithm and the bacterial one. It must be remembered that in these base line tests a very simple request topology is used, no differences in time to lives, values or cpu requirements for different requests; a relatively similar number of requests at each node; no temporal dependence; even distribution of all 100 different service requests. This simplicity was essential to show that even in conditions suitable to stochastic or rule based algorithms the bacterial based adaptive algorithm could perform sufficiently well. As more complexity, temporal behaviour and burstyness was added to the request stream an adaptive system should outperform other static rivals.

Conclusions

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.