

Antigens, Antibodies, and the World Wide Web

Daniel Stow and Chris Roadknight

B54/124, Aadastral Park, Martlesham Heath, Ipswich, Suffolk, UK. IP5 7RE
dstow@lineone.net, Christopher.roadknight@bt.com

Abstract. In this report the immune system and its adaptive properties are described and a simple artificial immune system (AIS) model based on the clonal selection theory is presented. This immune system model is demonstrated to be capable of learning the structure of novel antigens, of memory for previously encountered antigens, and of being able to use its memory to respond more efficiently to antigens related to ones it has previously seen (cross-reactivity). The learning, memory and cross-reactivity of the AIS are fruitfully applied to the problem of fuzzy resource identification. Interesting antigen/antibody relationships are also identified.

1 Introduction

Biologically inspired approaches to computing have found a natural application to problems in telecommunications (see [Di Caro & Dorigo 1998; Seth 1998; Roadknight & Marshall 2000] for examples). Modern telecommunication systems can be seen as complex structures built out of simple interacting components, much like biological systems. In this light, the co-operation between the two fields is hardly surprising. A current concern is that the introduction of new Internet services is being held back by human-intensive approaches to network management. It is becoming clear that management of the Internet must be less dependent on manual intervention.

Immune systems have several properties that make them of interest to computer scientists, including diversity, distributedness, adaptability, self-organisation and autonomy [Hunt & Cooke 1996; Somayaji *et. al.* 1998; Hofmeyr & Forrest 1999]. These are properties that they share with Roadknight and Marshall's bacteria.

2 The Artificial Immune System

Immune systems have a variety of appealing properties (condensed from similar lists in [Hunt & Cooke 1996] and [Somayaji *et. al.* 1998]). *Distributedness, Diversity, Dynamically changing coverage, Imperfect detection, Autonomy, Unsupervised learning, Content addressable memory, Adaptability.*

As in other machine learning applications of artificial immune systems (e.g. [Hunt & Cooke 1996; Timmis *et. al.* 2000]), the antigens in our model represent the problem that the immune system must learn to solve. In order to show that our model could

successfully learn about real world data, we developed an encoding scheme to convert URLs, taken from a cache access log file, to 64-bit binary strings. This scheme encodes the first three numbers of the IP address (8 bits each), the number of forward slashes in the URL (3 bits), the number of characters between the first and last forward slash (5 bits), the first six letters of the name of the requested file (5 bits each), and the type of file requested, e.g. GIF, HTML, etc. (2 bits).

Antibodies in our model are 64-bit binary strings, just like antigens. In the biological immune system, molecular binding between antibodies and antigens depends on them having complementary shapes. Similarly, matching between antibodies and antigens in our model depends on the number of complementary bits they have. This was also the matching method used by Hightower *et al.* [1995]. After counting the number of complementary bits we apply a sigmoid function to give the probability of the antibody binding the antigen. This simulates the idea that antibodies must match antigens over a certain minimum area before they can bind

The genes used to create antibodies are divided into four libraries, each containing a number of 16-bit segments. In our model there are currently 16 segments per library. To create an antibody a segment is picked at random from each library and these segments are joined together. We experimented with two different types of stopping criteria for the learning. One of these was to stop learning when an antibody with a certain minimum score was found. The other involved limiting the number of antigens available for the antibodies to match. Each time an antibody attempted to match an antigen, an antigen had a chance to copy itself. Each time an antigen was matched it was killed. The learning was stopped when there were no antigens left.

Memory in our model is maintained in the simplest way possible – when the system finishes learning about an antigen it finds the highest scoring antibody in the population and makes it a memory antibody. Unlike other antibodies these can live for an unlimited length of time. The number of memory antibodies is limited and when the memory is full the new antibody replaces the oldest.

3 Experimental Results

In our research on our immune system model, we were interested in answering the following question: in what circumstances does it perform best and why? By performance we mean the speed at which it learns to recognise antigens, which will obviously be crucial if it is to be applied in the management of network resources. The first factor that we noticed affecting the performance of the model was the set of antigens itself. Figure 1 shows the speed of learning for antigens encoded from nine different cache log files and four random number seeds. Speed of learning in this case is measured by the average number of generations of clonal selection required to evolve an antibody with a score of at least 95 (out of 100).

The graph shows that the performance is affected far more by the cache log file than by the random number seed. The likely explanation is that different log files have different patterns of repetition of URLs in them. If there is more repetition of the same or similar URLs in the log file, this should be mirrored in the set of antigens. The more repetition there is in the set of antigens, the more useful memory will be.

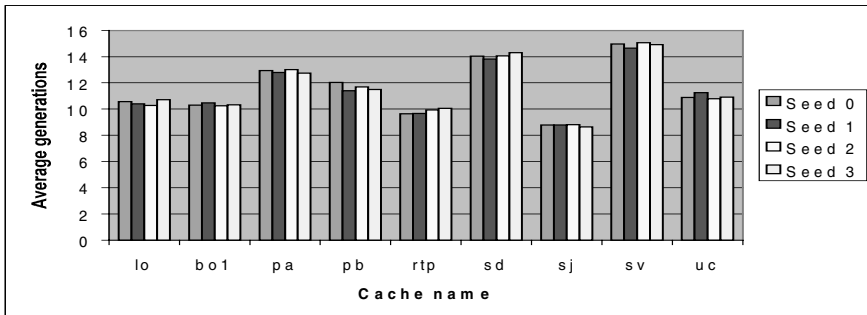


Fig. 1. Comparison of performance on nine different cache log files.

Figure 2 shows the performance on one log file (10) for memory sizes from 0 to 1000 (the maximum possible), and numbers of randomly generated antibodies from 10 to 2000. Here performance is measured by the length of time the system takes to learn 1000 antigens. We discovered, by comparing this graph with the corresponding graph for number of generations, that time is the better measure. With a large number of randomly generated antibodies, the time taken is high but the number of generations is low. This is because the size of the population of antibodies affects the time taken, the more antibodies there are the more time it takes.

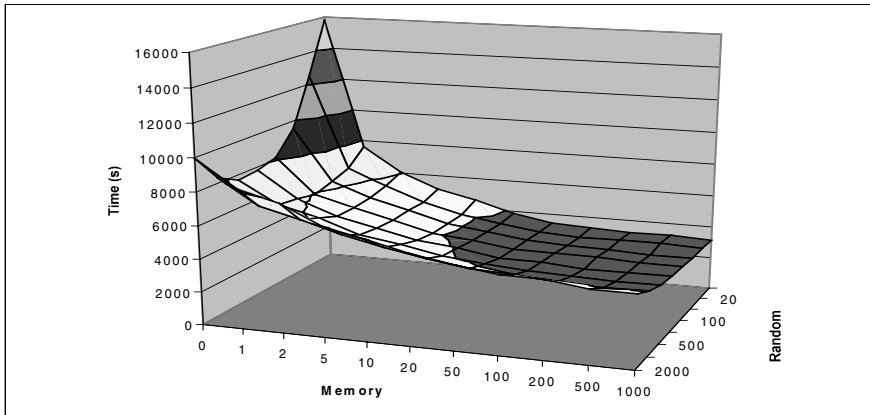


Fig. 2. Performance with different numbers of random and memory antibodies.

The graph shows that the memory size generally has a much larger impact on the speed of learning than the number of random antibodies. Only when there is very little or no memory does the number of random antibodies have much effect. The graph also shows that a memory size of 50 is almost as good as the maximum of 1000, suggesting that repetition in the log files generally happens within a short space of time. If antigens took a long time to be repeated such a short memory would not be very useful. It also shows that the system does not have to remember everything (and thereby use a large amount of space) to be effective. We also experimented with a different stopping criterion for the learning, which involved limiting the amount of antigen available for antibodies to match. The learning was stopped when there was none left. When the initial amount of antigen was 1000 we achieved similar results as

shown in Figure 2 for the time taken to learn. However, since we could no longer set the minimum score that had to be achieved we were interested in what the best antibody scores being achieved were. Figure 3 shows the time taken with different numbers of random and memory antibodies. Figure 4 shows the average best antibody scores achieved for the same runs. Scores greater than 95 were achieved in most runs, and in many cases this was done in a similar time to experiments using the old stopping criterion. The advantage of the new stopping criterion is that it can achieve average best scores approaching 98 in a time of only 7500 seconds for 1000 antigens

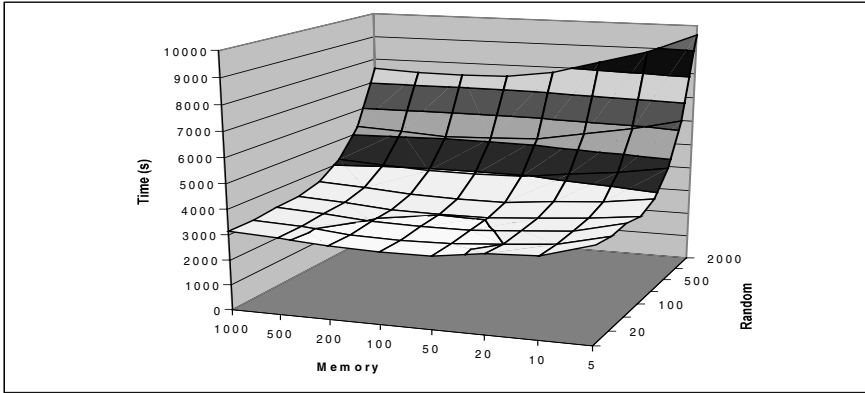


Fig. 3. Performance with different numbers of random and memory antibodies.

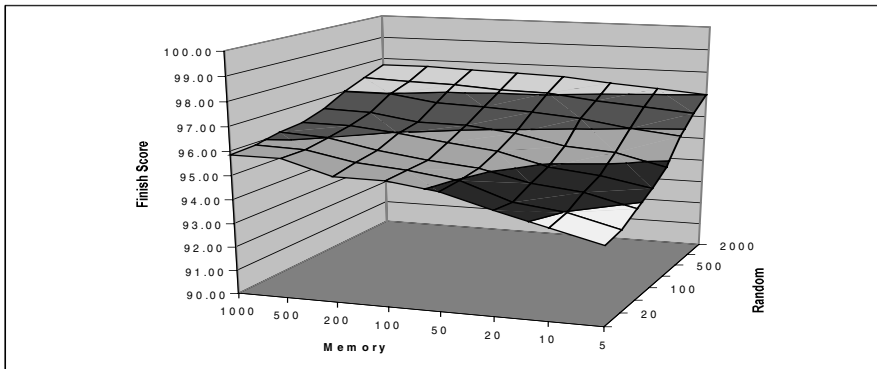


Fig. 4. Average best scores (out of 100) in the same runs shown in Figure 3.

4 Conclusions

We believe that immunological principles can be fruitfully applied to network management, particularly the problem of resource identification. In this work we have presented a simple model of the immune system as a first step towards this. Our immune system model proved to be capable of learning and remembering the structure of novel antigens, encoded from URLs in cache access log files. What is more, the model could achieve this even with limited resources. We also suggest that the success of the system is somewhat dependent on patterns in the antigens it must

learn about. This may limit the domains to which it can be applied. We will need to consider possible real world applications for the system, which means considering very carefully how antigens are to be encoded so that similar real world situations map to similar antigens, and also how the antibodies can encode real world solutions for these situations.

References

- Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, **9**, pp. 317-365.
- Hightower, R., Forrest, S., & Perelson, A.S. (1995). The evolution of emergent organisation in immune system gene libraries. In L.J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, San Francisco, CA: Morgan Kaufmann, pp. 344-350.
- Hofmeyr, S.A., & Forrest, S. (1999). Immunity by design: An artificial immune system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, San Francisco, CA: Morgan Kaufmann, pp. 1289-1296.
- Hunt, J.E., & Cooke, D.E. (1996). Learning using an artificial immune system. *Journal of Network and Computer Applications*, **19**, pp. 189-212..
- Roadknight, C.M., & Marshall, I.W. (2000). Adaptive management of a future service network using a bacteria inspired genetic algorithm. In *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, Las Vegas, Nevada, pp. 331-337.
- Seth, A. (1998). A co-evolutionary approach to the call admission problem in telecommunications. In J. Halloran, & F. Retkowsky (eds.), *The Tenth White House Papers: Graduate Research in the Cognitive and Computing Sciences at Sussex*, Cognitive Science Research Paper 478, School of Cognitive and Computing Sciences, University of Sussex.
- Somayaji, A., Hofmeyr, S.A., & Forrest, S. (1998). Principles of a computer immune system. In *1997 New Security Paradigms Workshop*, ACM, pp. 75-82
- Timmis, J., Neal, M., & Hunt, J. (2000). An artificial immune system for data analysis. *Biosystems*, **55**, pp. 143-150.