Design of a Parallel Immune Algorithm based on the Germinal Center Reaction

Ayush Joshi*
University Of Birmingham
School of Computer Science
Birmingham, B15 2TT
axj006@cs.bham.ac.uk

ABSTRACT

Artificial Immune algorithms are relatively new randomized meta-heuristics and not a lot of work has been done on parallel immune algorithms yet. Most of these implementations use some version of the first generation artificial immune algorithms. In this research a novel parallel artificial immune algorithm for optimization is proposed based on cutting edge research in the study of germinal center reaction. This parallelism of the algorithm is inherent in the system as a whole, which is different than other parallel implementations of nature inspired algorithms, where several instances of the algorithm is run multiple times to exploit parallel architecture of computers. This system is being developed with input from immunologist and incorporates new ideas which have not been explored before. Some preliminary results are presented which hint that it could perform better than the evolutionary algorithm ((1+1)EA), with which it is compared. The algorithm is not limited to optimization and in the future the research will look into other application areas. Also limitations, improvements and applications where it excels, will be explored in the research.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic Methods*; G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization, Unconstrained optimization*

General Terms

Algorithms, Design, Performance

Keywords

Artificial Immune Systems, Germinal center reaction, Nature inspired algorithms

*PhD Researcher

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.

1. INTRODUCTION

Artificial Immune Systems (AIS) are a class of meta-heuristic algorithms which are inspired from the biological immune system. Some prominent features of the immune system are pattern recognition, self-identity, anomaly detection, noise tolerance, diversity, fault tolerance, robustness, immune learning, memory and more [6]. The immune system stands out among the various biological systems, as hardly any other system has all these properties together. The desirable properties of the immune system make it a good inspiration for designing algorithms, as a result AIS have been applied to a large number of problem domains from anomaly detection and security to optimization and machine learning. [7, 20] provide a survey of known applications of AIS.

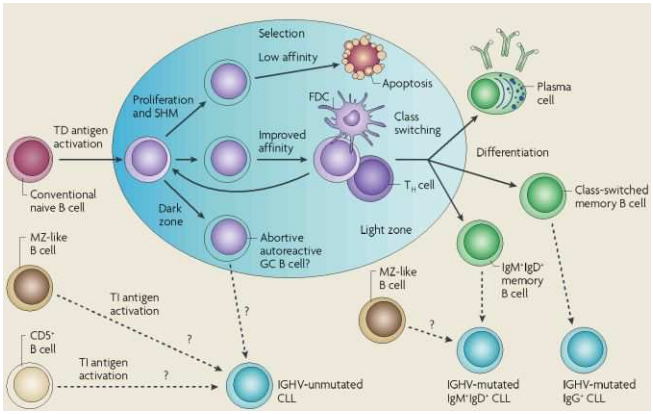
The main purpose of the immune system is to fight infection and protect the organism from diseases caused by pathogens [17]. To do so the system needs to be able to perform the following tasks effectively:

1. Identify the pathogen as well as the self organism
2. On attack from a pathogen, initiate and regulate some form of response
3. Remember the characteristics of the pathogen for better response for the future.

The immune system has a multi-layer structure which performs these tasks efficiently. There are two kinds of immune response to an invading *Antigen* (Ag), the static, innate immune response and the dynamic, adaptive immune response. [6] states that for an algorithm to be classified as an AIS it must include at least one characteristic model from the immune system and it be designed towards problem solving. Different abstractions can be made for different parts and/or the granularity of the immune system. Based on the popular theories namely, Clonal selection, negative selection, danger theory, immune network theory, different versions and kinds of AIS have been proposed and experimented with [4, 16].

Compared with other nature inspired algorithms AIS are still relatively new. The first known ideas were published by [10, 13] on immune networks and [2, 1, 3] on problem solving but it is only in the mid nineties that AIS gained an independent footing. In the book [6] published in 2002, an attempt was made to formalize the definition of an AIS. The development in the field can roughly be divided into two generations. The first generation of algorithms were abstract models which used some model or property of the immune system and often used logical instead of natural extensions in their design. There was little if any interaction between

Figure 1: Diagram of the germinal center [21]



immunologists and computer scientists and these algorithms depicted basic understanding of the immune system being manipulated to solve problems. Examples of the first generation algorithms include negative selection algorithms for classification [11], CLONALG [9], B cell algorithms [14] and AiNET [8] algorithms. These were followed by the second generation algorithms which are a recent development and systems being developed are much more involved and complex. There is a lot more input from immunologists which facilitates better understanding of the immune systems. Also, to an extent attempts are being made to introduce more than one feature of the immune system into the AIS models. This is beneficial as many of the first generation algorithms are not that robust or do not scale well. As different parts of the immune system do not work in solitary it is evident that incorporating more features and detail can lead to better properties in AIS. The Dendridic cell algorithm [12], TLR [18] and MILA [5] are some examples in this category.

2. PROPOSED RESEARCH

Adaptive immunity is responsible for protecting the body against the Ag which pass through the chemical and natural barriers of the body and are not recognized by the natural killer cells. B cells and T cells in the adaptive immune system play different roles to protect the body. T cells are of two types, namely Helper and Killer T cells. The Killer cells can recognize certain cells infected by viruses or damaged cells and eliminate them while, the helper cells do not directly kill any cells but control the immune response. B cells on the other hand interacts with the Ag and mutate to produce antibodies which eliminate the Ag. The B cells must be helped by T cells to activate them after they interact with the Ag. When *Antibodies* (Ab) produced by B cells in the extra cellular spaces prevent Ag growth and protect from the pathogen, this phenomenon is called the *Humoral Immune Response*[17]. Naive B cells are triggered by T cells and Ag, and they develop into memory and plasma cells. The memory cells recognize the Ag to provide a better attack in the future, and the plasma cells are responsible for secreting antibodies which cause destruction of antigens.

2.1 The Germinal Center Reaction

Some of the activated B cells in the immune response move to the secondary lymphoid tissue. It is here that they undergo rapid proliferation, mutation and selection to form the *Germinal Center* (GC). A germinal center is composed of mature B cells, T cells and *Follicular Dendridic Cell* (FDC), loosely separated into two zones, the light zone and the dark zone. The Ag is bound to the FDC and is presented to the B cells in the light region. Figure 1 depicts the germinal center with B cells at different stages, from naive B cells towards the left to maturing B cells in center and apoptotic cells towards the right part. Memory cells and plasma cells can also be seen outside the germinal center towards the extreme right.

The selection proceeds as follows: B cells proliferate and mutate in the dark region and the clones produced move to the light region where their ability to bind with the antigen is tested. This is called the Cyclic re-entry model [15]. B cells which bind to Ag can interact with T cells and are selected for further generations. The unselected B cells die due to apoptosis.

How the selection pressure is maintained in the GCs is an area of active research and a new theory is proposed by Zhang et al [22] which forms the basis of this research. This model suggests that the selection of B cells is via their secreted products i.e. the Ab. Selection pressure is maintained by Ab masking the Ag presented by FDCs, thereby increasing competition to bind with Ag continuously. [22] states "The antibody-dependent selection mechanism demonstrated here makes inter-GC B cell migration dispensable, as soluble antibodies produces a systemic selection threshold". This can be viewed as antibodies from GCs interacting with other GC to regulate selection pressure.

Towards the end of the GC reaction germinal centers begin to disappear as B cells are prone to dying unless they can bind with Ag. When Ab is specific enough, most of the Ag is eliminated and we see recovery from illness. At this stage there is hardly any antigen left and GC disappear.

2.2 Algorithm Design

Here a preliminary design of the AIS inspired by the new advances in understanding the GC reaction [22] is presented. The model is composed of a group of GCs each having a number of B cells which form the individuals. The individuals follow the GC reaction i.e. proliferation, somatic hyper-mutation and selection to generate better individuals. In this case individuals with better fitness value. At some predetermined stage in the algorithm there is a migration of individuals from one GC to others which corresponds to antibody movement. Towards the later part of the algorithm cell death occurs by eliminating individuals from the population. In the domain of optimization problem this would occur at a stage where a sufficiently high fitness is achieved.

2.2.1 Pseudocode

Here a simple descriptive version of the algorithm is presented in pseudo-code, with the objective to minimize fitness. Some variables and strategies still need to be decided for inclusion after appropriate discussions with immunologist, this work is currently under progress.

Algorithm 1 Pseudocode for the generic version of the Algorithm

No. of GC $= \gamma$
Population Size $= \mu$; Offspring per individual $= \lambda$
1. Initialization: Create γ GC $G = \{z_1, z_2, \ldots z_\gamma\}$, each with population $P = \{x_1, x_2, \ldots x_\mu\}$
2. Mutation and Expansion
for all $j \in \{1, 2, \ldots, \gamma\}$ **do**
  for all $i \in \{1, 2, \ldots, \mu\}$ **do**
    Create λ clones of x_i and place them in pool $C_{ij} = \{y_{ij}^1, y_{ij}^2, \ldots y_{ij}^\lambda\}$
    for all $k \in \{1, 2, \ldots, \lambda\}$ **do**
      apply mutation to each clone(any mutation strategy can be used)
    end for
  end for
end for
3. Selection
for all $i \in \{1, 2, \ldots, \mu\}$ **do**
  for all $j \in \{1, 2, \ldots, \gamma\}$ **do**
    if $min\{f(y_{ij}^1), \ldots f(y_{ij}^\lambda)\} \leq f(x_{ij})$, replace x_i by some randomly chosen y_{ij} with minimal f-value
  end for
end for
4. Every m generations use Ab for Selection
5. Apoptosis
Every o generation eliminate $min(x_i)$
until some stopping criterion is met, repeat at step 2.

3. PRELIMINARY RESULTS

Some preliminary results are presented here using a stripped down version of the algorithm. The algorithm is composed of 4 GC limited to population size 1 and is compared with a version of the (1+1) EA with random restarts. The individuals are represented as pseudo-boolean bit strings of length n, set to 100. In each generation only 1 clone is produced per individual. The tests are carried out on a variation of the TwoMax [19] problem and 500 individual runs are performed. The immune algorithm is run for 2000 generations and EA is given 10,000 generations. These choice of generations give 10,000 and 5000 fitness evaluations to the EA and immune algorithm respectively. This is done as since the immune algorithm is population based it can be expected that with a high probability at least one of the individuals will be generated on the right side of the TwoMax function so it has a chance of finding the maxima. The EA on the other hand has an equal probability for the individual being generated on either side. This is the reason we introduce restarts as if the individual is not replaced by a better clone given 500 tries then it is replaces by a randomly generated new individual. The apoptosis of 1 individuals occurs at every 500 generations based on worst fitness value.

For the sake of comparison the mutation rate for both the AIS and EA is set to 1/n. The EA is given $5 \times n$ generations if it gets stuck in a minima after which it starts at a random position. Uniform random mutation is used, i.e, each bit is flipped with a probability 1/n. Fitness evaluations are used for the comparison. The function used for comparison is

Table 1: Results on fitness evaluations

a	Algo	Min	Max	Avg	Std. Dev	success
1	EA	42	8406	1486	1800.45	99.8
	Immune	32	202	95	24.25	93.4
2/3	EA	89	9963	1626	1790.81	99.2
	Immune	39	500	233	82.32	89.6
1/3	EA	341	9995	4572	2647.98	54.8
	Immune	218	1909	1002	374.02	15

(where #1 stands for the number of ones)

$$f = \begin{cases} n/2 - \#1 & \#1, \leq n/2 \\ a * (\#1 - n/2) & n/2 < \#1 \leq 3n/2 \\ n & otherwise \end{cases}$$

The algorithm needs to be slightly changed to ensure that maximization is achieved as objective for our toy function is to find the maximum value. This can be achieved by changing step 2 of the algorithm by replacing individual by clone if it has equal or better fitness than the individual. Tables 1, 2 and 3 show fitness evaluations when the variable a is set to 1, 2/3 and 1/3 respectively. This can be seen as variations in the slope of the right side of the TwoMax function. From table 1 it can be seen that immune algorithm finds the best solution using less fitness evaluations than the EA. The results presented in the table are values for runs in which the best result was found, there were runs when this was not true. Percentage success shows how many times the algorithm found the best solution in the 500 runs provided. It should be noted that percentage success of the EA is higher than the immune algorithm due to the random restart strategy. At this stage it can be said that for the instances when the immune system finds the solution, it does so using less fitness evaluations than the EA. The cell death strategy is a factor which needs to be experimented with as in the last case with $a = 1/3$ for the immune system the chance that individual on the right path gets killed is high as compared to the others, which could be a cause why the success is low.Since in the current cell death strategy we kill the individual with the worst fitness, suppose we have two individuals equal distance apart from the center (the fitness is 0 at center), then the individual with more 1's (the individual to the right of the graph) will have a less fitness value, hence more probable to die. This means that there is a higher chance for the individual on the path to the global optimum to die than the one on the path to the local optimum. Further analysis will be carried out to find better strategies for cell death and conditions when to use a particular strategy given a problem.

4. CONCLUSIONS

In this paper the description of a parallel immune system is presented which incorporates new ideas from immunology, namely the germinal center reaction. This system has been kept as close to real immune principles as possible without loss of detail. It is shown here that for a toy problem this novel approach finds solutions faster than the EA but not as often. These results are empirical in nature and will need theoretical results to support them, which is will be carried out in the future. A novel AIS is demonstrated in its very early days of development with sections which need

work done to implement the final version of the algorithm. Some interesting questions which will be addressed in the future work will include trying to find applications where the algorithm excels as well as looking into immunological implications, as to why the immune system fails for certain illnesses and whether this issue can be investigated from a optimization perspective.

5. ACKNOWLEDGMENTS

Support from Prof. Jonathan E. Rowe, Head of School and Christine Zarges, Fellow at the School of Computer Science,University of Birmingham is gratefully acknowledged.

6. REFERENCES

[1] H. Bersini. The immune recruitment mechanism. *A Selective Evolutionary Strategy, Proc. 4th ICGA, 1991*, 1991.

[2] H. Bersini. Immune network and adaptive control. *Varela and Bourgine*, 2332:217–226, 1992.

[3] H. Bersini and F. Varela. Hints for adaptive problem solving gleaned from immune networks. In H.-P. Schwefel and R. MÃd'nner, editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 343–354. Springer Berlin Heidelberg, 1991.

[4] D. Dasgupta. Advances in artificial immune systems. *Computational Intelligence Magazine, IEEE*, 1(4):40–49, Nov. 2006.

[5] D. Dasgupta, S. Yu, and N. Majumdar. Mila–multilevel immune learning algorithm. In *Genetic and Evolutionary Computation GECCO 2003*, pages 201–201. Springer, 2003.

[6] L. N. De Castro and J. Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer, 2002.

[7] L. N. De Castro and F. J. Von Zuben. Artificial immune systems: Part ii–a survey of applications. *Department of Computer Engineering and Industrial, Tech. Rep*, 2000.

[8] L. N. de Castro and F. J. Von Zuben. ainet: an artificial immune network for data analysis. *Data mining: a heuristic approach*, 12:231–259, 2001.

[9] L. N. De Castro and F. J. Von Zuben. Learning and optimization using the clonal selection principle. *Evolutionary Computation, IEEE Transactions on*, 6(3):239–251, 2002.

[10] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1):187–204, 1986.

[11] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on*, pages 202–212. Ieee, 1994.

[12] J. Greensmith, U. Aickelin, and S. Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *ICARIS*, pages 153–167, 2005.

[13] G. W. Hoffmann. A neural network model based on the analogy with the immune system. *Journal of Theoretical Biology*, 122(1):33–67, 1986.

[14] J. Kelsey and J. Timmis. Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation. In E. C.-P. et al, editor, *Genetic and Evolutionary Computation Conference - GECCO 2003*, volume 2723 of *Lecture Notes in Computer Science*, Chicago. USA., July 2003. Springer-Verlag.

[15] T. B. Kepler and A. S. Perelson. Cyclic re-entry of germinal center b cells and the efficiency of affinity maturation. *Immunology today*, 14(8):412–415, 1993.

[16] F. Liu, Q. Wang, and X. Gao. Survey of artificial immune system. In *Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on*, pages 5–pp. IEEE, 2006.

[17] K. M. Murphy, P. Travers, and M. Walport. Janeways immunology. *Garland science*, 2008.

[18] J. Twycross, U. Aickelin, and A. M. Whitbrook. Detecting anomalous process behaviour using second generation artificial immune systems. *IJUC*, 6(3-4):301–326, 2010.

[19] C. Van Hoyweghen, D. E. Goldberg, and B. Naudts. From twomax to the ising model: Easy and hard symmetrical problems. *generations*, 11(01):10, 2001.

[20] X. Wang, X. Gao, and S. Ovaska. Artificial immune optimization methods and applications-a survey. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3415–3420. IEEE, 2004.

[21] T. Zenz, D. Mertens, R. Küppers, H. Döhner, and S. Stilgenbauer. From pathogenesis to treatment of chronic lymphocytic leukaemia. *Nature Reviews Cancer*, 10(1):37–50, 2009.

[22] Y. Zhang, M. Meyer-Hermann, L. A. George, M. T. Figge, M. Khan, M. Goodall, S. P. Young, A. Reynolds, F. Falciani, A. Waisman, et al. Germinal center b cells govern their own fate via antibody feedback. *The Journal of experimental medicine*, 210(3):457–464, 2013.