IIDLE: An Immunological Inspired Distributed Learning Environment for Multiple Objective and Hybrid Optimisation

Jason Brownlee

Abstract— **The acquired immune system is a robust and powerful information processing system that demonstrates features such as decentralised control, parallel processing, adaptation, and learning. The Immunological Inspired Distributed Learning Environment (IIDLE) is a clonal selection inspired Artificial Immune System (AIS) that exploits the inherent parallelism, decentralised control, spatially distributed nature, and learning behaviours of the immune system. The distributed architecture and modular process of the IIDLE framework are shown to be useful features on complex search and optimisation tasks in addition to facilitating some of the desired robustness of the inspiration.**

I. Introduction

THE mammalian immune system is an intriguing natural defence system that has been shown to be capable of learning, memory, and adaptation [7,16,21]. Lymphocyte cells recirculate, migrating and homing throughout the host organism via the vascular and lymphatic systems [5,22] providing a mobile yet consistent defence (immuno-surveillance). Conceptually the acquired immune system can be viewed as a spatially distributed, circulating, and heterogeneous population of specialised discrete units that provide a homogeneous defence against external pathogenic material. In this abstraction of the biological system, the population of discrete units themselves are atomic (operation applied entirely or not at all) and immutable (specificity is fixed for the units lifetime). Once activated or triggered by an antigenic stimuli, the immune system is capable of manipulating the local physiology to facilitate the allocation of additional resources.

This manner of local site-specific learning is decentralised (no global controlling process), given that an external pathogen can be encountered at any spatially distributed location in the host system. Further, the system is robust both in its ability to learn to defend itself and the host organism over time, and in that it is not dependent upon any single or for that matter local group of units for the effectiveness of the system. This robustness facilitated through inbuilt redundancy permits quantities of discrete units and perhaps entire localised sections of the host system to be removed, whilst the immunity learned by the system is persisted, although at some cost.

Manuscript received January 31, 2006, revised April 15 2006.
Jason Brownlee is with the Faculty of Information and Communication Technologies, Swinburne University of Technology P.O. Box 218, Hawthorn, Australia, 3122, email: jbrownlee@ict.swin.edu.au

This work will elicit design goals from these canonical observed actions of the biological immune system. An Artificial Immune System (AIS) is proposed that addresses these design goals that is a unique interpretation of the clonal selection principle in its adoption of a spatially distributed population that permits antibody movement in an abstracted but similar manner to the circulatory system of the biological immune system. The discretised and spatially distributed learning framework is shown to be inherently applicable to multiple objective optimisation problems domains and to facilitate the adoption of learning paradigms other than the classical clonal selection algorithm.

II. Design Goals

The clonal selection principle proposed by Burnet [17] is core to understanding the learning and memory behaviours of the acquired immune system. The theory describes the selection of useful antibodies by antigen, and the subsequent proliferation and differentiation (learning) processes triggered by the selection event. This Darwinian-like learning theory has been a popular basis for learning algorithms in the field of AIS for search and optimisation tasks, the principle contribution being de Castro and Von Zuben's work on CLONALG [14,15] applied to optimisation problem domains. The inherent parallelism of this immunological metaphor has been acknowledged in previous work through the parallelisation of CLONALG [2] and the AIRS algorithm [1], although to the authors knowledge there as been no clonal selection-based AIS that has been designed to be decentralised and distributed from the ground up to investigate and exploit the spatially distributed nature of the biological metaphor. To this end, the following provides a listing of the design goals for a distributed clonal selection inspired artificial immune system.

1) **Context Specific Learning** – The clonal selection principle can be configured and applied to a specific engineering application.
2) **Self-Regulated Resource Maintenance** – Processes exist to govern allocation, and de-allocation of resources as required in the context of the clonal selection-based learning ensuring bounded space complexity.
3) **Decentralised Control** – Intrinsic processes (such as learning and resource maintenance) of the system are localised and do not require a master controlling process, thus permitting a distributed implementation.

4) **Triggered Adaptation** – As implied by the introduction and recognition of a pathogen in the clonal selection principle, the learning process is triggered by an external stimulus.
5) **Robustness and Fault Tolerance** – Inbuilt redundancy of discrete units demonstrated in the immune system permitting the removal (killing), and potentially the addition of components dynamically to the system.

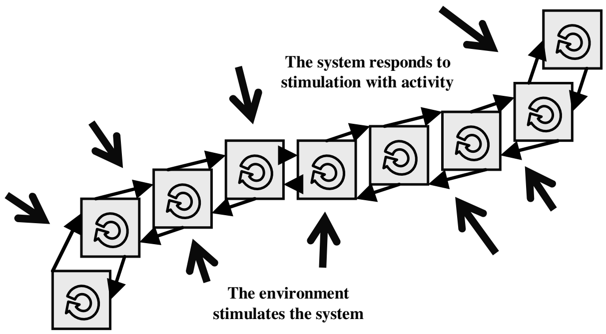

Fig. 1. Conceptualisation of a distributed immunological architecture showing locality connectivity, external stimulation, and internal activation.

Such a distributed clonal selection algorithm is envisioned to be composed of a population of discrete information packets (units) that provide the substrate for the triggered adaptation and knowledge representation in the system. A multiple discrete population scheme similar to island population genetic algorithms [6] could be adopted to allow the population to be spatially distributed in a parallel and distributed implementation. Such a configuration would also permit the discrete units to be in continuous movement similar to the behaviour of antibody in the circulatory system in an attempt to maximise physical coverage of the triggered adaptation across the distributed system. Positive selection, proliferation, and differentiation of units could be addressed through an implementation of CLONALG or other adaptation schemes, although the algorithm would require modification to make it modular (decentralised) and be amended to support the aging (decay and ultimate death) of mobile discrete units.

III. An Immunological Inspired Distributed Learning Environment

The algorithm described in this paper is called the Immunological Inspired Distributed Learning Environment or IIDLE for short. It has been labelled a learning environment because, as will be shown, it is composed of system *architecture* and associated *processes* including a learning process that is not limited to the clonal selection algorithm.

A. Architecture

The continuous biological circulatory system is discretised to a number of localities that are chained together in a loop configuration. A loop network topology is selected for simplicity of both the abstract model and the implementation, although in practice other topologies could be used (such as bus, star, and small world). A locality abstraction provides both an interface and scope at which to apply local (decentralised) processes, as well as a transient storage location for discrete antibody units.

A locality supports a population structure called a tail that stores a number of discrete units for a given time. The structure is called a tail given the manner in which it is depicted diagrammatically (see figure 2). The locality and the tail are differentiated for conceptual modularity where the physical storage medium and conceptual data structure for units can be changed and reconfigured independently of the locality and its associated governing and maintenance processes.

A tail contains a number (zero or more) of units at any given time during a simulation. Each unit is a discrete information packet meaningful in the context of the specific problem domain to which IIDLE is being applied. For example, a unit may represent a single candidate solution in a search or optimisation problem, or an exemplar in the case of a classification or collaborative filtering application. In short, a unit contains a piece of knowledge relevant to the problem being addressed. A unit also has an associated energy or age used by resource maintenance processes. A unit may have additional metadata such a quality scoring or a time of last evaluation to facilitate context specific learning.

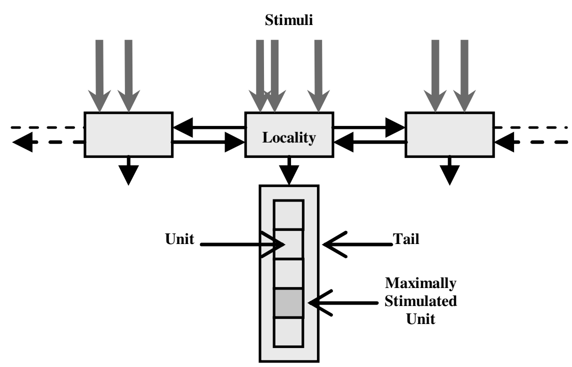

Fig. 2. Overview of the IIDLE architecture showing localities, tails, and units.

B. Processes

The system is simulated in discrete time intervals and processes represent the algorithmic component that operates upon the discrete unit substrate in the context of a specific problem application. Processes are confined to a locality and can be implemented as independent threads of execution or executed in aggregate sequentially by a single thread of execution at a desired scoping of localities. This provides an interesting level of implementation and runtime flexibility. IIDLE is designed with three core processes; *movement*, *decay*, and *expansion*, to meet the proposed design goals. The inclusion of additional processes (such as random unit insertions or statistics collection) would be a relatively trivial exercise.

1) *Movement Process*

Movement is the process of removing units from the tail of one locality, and adding them in the tail of a neighbouring locality. This unit migration facilitates the mobility of the knowledge learned by the system thus facilitating the redundancy of units and general robustness of the architecture. The cut-and-paste implementation of this migratory process enforces the design that the only process that can create and add a unit to the system is the expansion process providing a single point of control for new resource allocation.

The movement process cannot be executed on a locality that either has no units in its tail at the time of execution, or has no neighbouring localities at the time of execution. A random selection method is used to choose a user-defined number of units in the locality to be moved – typically a small number (one or two). A random locality neighbour selection strategy is used as well as an insertion strategy (units are appended to the population) for adding the selected units into the selected neighbouring locality.

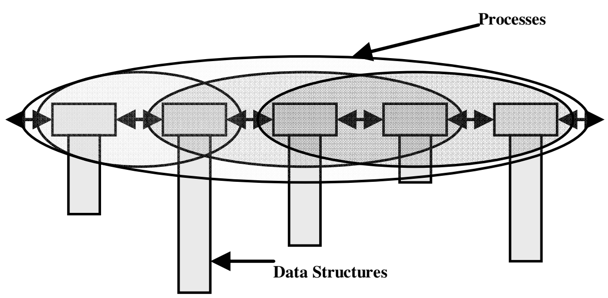

Fig. 3. An illustration of the weak coupling of data structure and processes as well as the variable scoping of processes on data structures.

2) *Decay Process*

The decay process is designed to manage the size of the unit population, specifically in response to the creation of new units by the proliferation strategy in the expansion process. Each unit has a finite energy, which is initialised upon unit creation, and it is the task of the decay process to decrement unit energy and discard those units that deplete all assigned energy. There are two primary approaches to manipulating unit energy, they are the conformer homeostasis approach (blind uniform unit decay) and the regulator homeostasis approach (seek global energy equilibrium), which is a simplification of density dependent regulation of T-cell populations [20]. The conformer homeostatic approach is the simpler and more efficient of the two approaches requiring a constat decrement of unit energy, although it elicits greater oscillations in system-wide population size (less control). The regulator homeostatic approach to population maintenance maintains the system population size more consistently although with the overhead of determining the total energy of all units within the scoped localities before decrementing unit energy. The consistency and control over unit population size that was demonstrated in early experimentation resulted in the regulator homeostatic approach being selected as the default approach, although both decay approaches are provided in the software implementation.

The regulator homeostasis approach to unit decay seeks to maintain the scoped total energy level at a user defined value. This process can be scoped per locality or across all localities in the system, and it is the latter configuration, which is recommended for a basic implementation of the system. The process requires the specification of an idealised total energy summed across all units at the configured scope. When executed, the process calculates the total unit energy, calculates the difference between the actual and the ideal energy values, and if positive, subtracts a fraction of the energy delta from each unit – thus returning energy to a point of transient equilibrium.

3) *Expansion Process*

Core to the design of the expansion process is the conceptualisation of *external* stimulation (input) that triggers the *internal* adaptation (learning) processes of the system. This conceptualisation is remarkably flexible in the context of IIDLE's modular framework. The stimulation of a locality may be implemented as the provision of computation time (such as execution of an objective function), provision of memory resources (such as extension and exploration of a state space), or access to a limited resource (such as database records in a mainframe). Stimulation can be sporadic or uniform in terms of the spatial location of its execution across localities and in terms of its temporal occurrences. Unlike the inspiring metaphor, the "immune response" is immediate leaving no pathogen to roam the distributed system.

The external stimulation of a locality results in an internal proliferation and differentiation process within the data structure of the locality called clonal expansion that was modelled on CLONALG. The triggered process starts with the application of a selection strategy that determines the unit(s) to participate in the expansion. These are likely the units in the locality at the time of expansion that are most efficacious or useful in the context of the problem domain. The proliferation and differentiation strategy uses the selected units as a basis to construct a clone of progeny units in an attempt to improve the response to stimulation.

This modularised variation on the clonal selection algorithm requires the selection of a problem specific *stimulation strategy* to trigger the adaptation, a problem specific *selection strategy* to choose useful units, and a problem specific *proliferation strategy* to create new units from selected units. The stimulation strategy further may require the design of a system stimulation policy in the event that the user has control over the stimulation-related resources (such as in the event of an objective function in an optimisation task). This policy involves a magnitude and frequency of stimulation events as well as a locality selection procedure (such as random).

IV. EXPERIMENTATION

To demonstrate the effectiveness of the IIDLE system with regard to the design goals outlined in Section II, the platform was preliminary tested on three experimental setups. The following three tests were selected to demonstrate potential rather than optimal usage of the framework and highlight both the versatility and flexibility of IIDLE. As such, the framework and algorithm configurations have not been tuned for optimal performance.

1) **Multiple-objective combinatorial optimisation** – Optimisation of an instance of the classical Travelling Salesman Problem (TSP) with three desirable and complementary objectives.
2) **Embedding multiple optimisation strategies** – Optimisation of a multimodal function using three standard optimisation approaches in parallel.
3) **Dynamic structure changes** – Optimisation performance and behaviour of IIDLE when localities are added and removed dynamically.

In the interest of field maturity and reproducibility all software, source code, data, and configuration files used in the experimentation has been made available on the internet[1].

A. *Multiple-Objective Combinatorial Optimisation*

A standard symmetric TSP [8,9] was selected (Berlin52) to demonstrate the flexibility of IIDLE to support multiple constraints concurrently. The objective for a TSP is to minimise overall tour length between the cities. Two additional potentially useful complementary objectives for this problem are to minimise the number of path intersections (tour crossing itself on the 2D plain) and to maximise the number of nearest-neighbour city connections[2].

A well-known algorithm used on this optimisation problem is the Ant Systems (AS) algorithm, an implementation of Ant Colony Optimisation (ACO) [18,19], which uses a historic pheromone map and a nearest-neighbour based heuristic to complete the search. This algorithm was modified to support all three of the selected heuristics and to use a discrete-history (population-based) representation called Discrete History Ant Systems (DHAS) and was embedded into the IIDLE framework[3].

The objective functions were used to evaluate units (city tour permutations), the scoring of which was used in the contribution of solutions to the temporary pheromone map used in the DHAS-based proliferation strategy of the expansion process. Table I lists the configurations for the five experiments. Experiments 1-3 used a single objective consistently across all localities. In experiment 4 the localities were partitioned into three groups, each exposed to a different objective function, and in experiment 5, all three objective functions stimulated the system in an un-partitioned manner.

TABLE I
MULTIPLE OBJECTIVE CONFIGURATION SETUP

No	Experimental Setup
1	Tour length (minimised)
2	Tour intersections (minimised)
3	Nearest neighbour connections (maximised)
4	All three objective functions partitioned
5	All three objective functions mixed

Table II provides a summary of tour lengths achieved with each configuration, averaged over 100 runs. Results obtained using the minimise-intersections objective (experiment 2) was quite similar to those of the conventional tour length objective (experiment 1). When the three complementary objective functions were used in parallel (experiments 4 and 5), the system achieved better results than using any single objective independently – likely because of the additional problem-specific information the system was exposed to. This is an expected result given that each of the three heuristics ultimately seeks a similar aim of minimising tour length, although from differing perspectives. Intriguingly the partitioned configuration (experiment 4) rather than the mixed configuration achieved the better result. It is assumed that the partitions of objective functions resulted in some form of implicit niching effect, allowing specialisation of the search within each partition, although increasing performance through information sharing between the partitions as facilitated by the movement process.

TABLE II
MULTIPLE OBJECTIVE EXPERIMENTAL RESULTS

No	Mean Tour Length	Stdev Tour Length
1	11,502.75	429.34
2	11,507.25	420.18
3	11,565.72	386.60
4	**11,342.81**	391.36
5	11,438.45	395.06

The optimal tour for Berlin52 is 7544.36.

B. *Hybrid-Technique Function Optimisation*

Function optimisation is a commonly used problem domain in population-based search such as the already mentioned CLONALG. A classical test function-optimisation problem is Schwefel's function [10]. This function was tested in 5-dimensions and was selected because of the extreme modality of its fitness surface. As in Section IV.A, five experiments were executed with IIDLE, although in this case three different optimisation algorithms were embedded as proliferation strategies of the expansion process. The three

[1] Software URL: http://www.it.swin.edu.au/personal/jbrownlee/iidle

[2] It is conventional to evaluate orthogonal rather than parallel (complementary) constraints on multiple objective domains. This remains an exercise for future research given time constraints.

[3] The Discrete History Ant Systems (DHAS) algorithm was designed and developed in equal collaboration by this author and Daniel Angus, and is described in an unpublished work titled "Discrete History Ant Systems" to be submitted to ANTS 2006.

optimisation algorithms that were embedded in IIDLE for the optimisation task were a Genetic Algorithm (GA) [3], Particle Swarm Optimisation (PSO) [12], and DHAS. Table III provides a summary of the experimental setup, and Table IV provides a summary of global minima located by each IIDLE configuration, averaged over 100 runs.

TABLE III
HYBRID TECHNIQUE CONFIGURATION SETUP

No	Experimental Setup
1	GA proliferation strategy
2	PSO proliferation strategy
3	DHAS proliferation strategy
4	All three strategies partitioned
5	All three strategies mixed

The results show that the GA strategy out performed PSO and DHAS, although the overall efficacy leader was the configuration that used all three strategies across the all localities (experiment 5). It was expected that combining the three strategies together would provide improved results and this was the primary outcome of this round of experimentation. The results are promising for the argument for IIDLE as a hybrid parallel search platform. The final interesting point to stress is the application of an ACO algorithm, which is conventionally used for combinatorial optimisation problems, was applied to a continuous function optimisation problem. Although the results of the ACO algorithm used in this way were merely satisfactory rather than optimal, it further demonstrates the flexibility of the DHAS technique embedded in IIDLE.

TABLE IV
HYBRID TECHNIQUE EXPERIMENTAL RESULTS

No	Mean Minima	Stdev Minima
1	-1,817.19	123.68
2	-1,494.65	185.67
3	-1,566.87	99.36
4	-1,797.70	143.35
5	**-1,904.59**	116.43

The global minima for Schwefel's function in 5-dimensions is -2094.31

C. *Dynamic Structure Changes*

The localities of IIDLE are loosely coupled requiring only connectivity to other localities to facilitate the movement of units. This modular architecture in conjunction with the inbuilt redundancy of units permits dynamic structural changes not only in the addition of localities, but also in the deletion of randomly selected localities and all the units contained in the locality at the time of removal.

Three different configurations were tested for comparison with a DHAS proliferation strategy on the Berlin52 TSP. The first experimental setup (Static) involved five tests that used a static number of localities in each test, each with fewer localities than the previous run. The second experimental setup (Added) involved the adding of localities to the IIDLE architecture at a random interval during a run. This setup involved four tests, each starting with a different number of localities that were increased from the initial number to 50. The third experimental setup (Removed) involved the removal of localities in the opposite manner to the Added experiment. Here each test started with 50 localities, and a different number were removed at a random interval over the course of the run. Table V provides a summary of the three experimental setups and the tests complete for each.

TABLE V
STRUCTURE CHANGES CONFIGURATION SETUP

Static	Added (x increased to y)	Removed (x decreased to y)
50→50	10→50	50→40
40→40	20→50	50→30
30→30	30→50	50→20
20→20	40→50	50→10
10→10	-	-

When the number of localities of the system is adjusted, it has a direct impact on the probabilities and thus the amount each locality is stimulated. The amplitude (number of stimulations) per iteration was kept constant, thus fewer localities (as in the removal case) increases the probability of locality stimulation, whereas the addition of localities has the opposite effect. It is also interesting to note that the decay process used was configured to maintain total system energy at equilibrium. Thus, the removal of localities and their unit's results in a short dip in overall unit population size, which is quickly returned to a stable point. Whereas the addition of localities did not affect the population size, instead, causing the unit population to be more sparsely spatially distributed. Table VI provides a summary of mean tour length results for all three configurations and tests averaged over 100 runs.

TABLE VI
STRUCTURE CHANGES EXPERIMENTAL RESULTS

Static mean (stdev)	Added mean (stdev)	Removed mean (stdev)
11,201.52 (411.35)	11,062.86 (379.38)	10,993.38 (392.82)
10,911.16 (337.06)	10,926.64 (367.37)	10,678.00 (396.93)
10,495.15 (360.71)	10,490.72 (402.16)	10,163.28 (370.34)
9,948.29 (279.32)	**9,765.75** (338.38)	**9,505.22** (301.09)
9,240.89 (252.11)	-	-

The results for the static configuration showed that the less localities (higher the probability) of being selected improved the final optimised result significantly. Similar results were observed when localities were added, an improved final result was obtained from fewer initial localities. It is speculated that this may be caused by the focusing of attention of the search process both in the frequency and the amplitude of stimulation (unit evaluated) and in the concentration of units in a confined population configuration. Resource removal demonstrated the overall better performance, indicating that the more resources removed, the better the final scoring. This provided further support to the idea that increased performance can be achieved with a small number of localities with large unit population sizes as opposed having many localities with smaller unit population sizes.

V. Discussion and Possible Extensions

The preliminary experiments in this work demonstrated IIDLE's potential as an adaptive and flexible artificial immune system that, although inspired by the acquired immune system and the clonal expansion principle in architecture and process, is not limited to clonal selection learning mechanisms. IIDLE's implementation of triggered adaptation was demonstrated to facilitate multiple objective functions in parallel, and the decentralised control implemented as modular self-regulated processes facilitated hybrid optimisation techniques. The discretised and distributed population structure promoted the robustness of the inspired metaphor, demonstrating that the localities and units of IIDLE's architecture can be manipulated in real-time with some interesting and potentially beneficial behaviours on an optimisation problem domain.

The flexibility and versatility of the IIDLE frameworks parallel implementation does come at cost of computational efficiency. Whether implemented on multiple processor hardware or a computer network, there are additional inter-process communication costs that must be taken into account. This concern of measuring costs and quantifying the parallelism trade-off against in IIDLE has been acknowledged, and remains an area for future research.

It is interesting to note the similarity of the final IIDLE system to the generic population-based framework proposed by Newborough and Stepney [13]. In their work, Newborough and Stepney proposed a distributed population (island) based system that employed the abstracted principles from common population-based optimisation algorithms, and was implemented on a number of connected FPGA's. Although the immunological inspired software approach of IIDLE and the hardware framework approach of Newborough and Stepney differ, the resultant learning systems have many commonalities, the principles of which undoubled can be capitalised upon by both streams of research.

This was an introductory work on the IIDLE framework and extensive sensitivity analysis is required for the selection and configuration of the processes and strategies fundamental to the movement, decay, and expansion processes of the framework. In particular, the features and limitations of various different decay, and movement processes, locality network topologies and expansion schemes needs to be modelled to provide a choice and indication of effective configuration during implementation.

The exciting contribution of this immunological inspired learning system is that from its inception it facilitates the decentralised, distributed, and adaptive power of the biological inspiration, rather than having to be reengineered after it's inception as in the case of CLONALG and AIRS. Moreover, the learning system "naturally" facilitates multiple objective and hybrid technique optimisation within a framework that is easily distributed on a computer network (such as the internet). Some addition potentially rewarding work may be in applying the framework to problem domains other than optimisation such as classification with an exemplar-based proliferation strategy like the successful negative selection algorithm [4], or in using human-feedback as stimulation to IIDLE in a human interactive learning system [11].

Acknowledgment

The author would like to thank Professor Tim Hendtlass for the mentoring and support in preparation of this paper. The author also acknowledges and thanks Daniel Angus for his collaboration work in the design and development of the Discrete History Ant Systems (DHAS) algorithm. Finally, the author would like to thank the reviewers of this work who provided insightful comment and highlighted some very interesting relevant research.

References

[1] Andrew Watkins and Jon Timmis, "Exploiting Parallelism Inherent in AIRS, an Artificial Immune Classifier," *Proceedings of the 3rd International Conference on Artificial Immune Systems (ICARIS2004),* Catania, Sicily, Italy, pp. 427-438, 2004.

[2] Andrew Watkins, Xintong Bi, and Amit Phadke, "Parallelizing an Immune-Inspired Algorithm for Efficient Pattern Recognition," *Intelligent Engineering Systems through Artificial Neural Networks: Smart Engineering System Design: Neural Networks,* pp. 225-230, 2003.

[3] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, USA, Canada: Addison Wesley Publishing Company, Inc., 1989.

[4] Dennis L. Chao and Stephanie Forrest, Information Immune Systems *Genetic Programming and Evolvable Machines*, vol. 4, pp. 311-331, December 2003.

[5] Dov J. Stekel, Claire E. Parker, and Martin A. Nowak , A model of lymphocyte recirculation *Immunology Today*, vol. 18, pp. 216-221, May, 1997.

[6] E. Alba and M. Tomassini, Parallelism and evolutionary algorithms *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 443-462, 2002.

[7] Eli Benhamini, Richard Coico, and Geoffrey Sunshine. *Immunology - A Short Course* , USA: Wiley-Liss, Inc., 2000.

[8] G. Reinelt, TSPLIB - A Traveling Salesman Problem Library *ORSA Journal on Computing*, vol. 3, pp. 376--3841991.

[9] G. Reinelt. *The traveling salesman : computational solutions for TSP applications*, Berlin ; New York: Springer-Verlag, 1994.

[10] H. P. Schwefel. *Numerical optimization of computer models*, New York, NY, USA: John Wiley & Sons, Inc., 1981.

[11] H. Takagi, Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation *Proceedings of the IEEE* , vol. 9, pp. 1275-1296, Sep, 2001.

[12] James Kennedy and Russell Eberhart. *Swarm Intelligence*, USA: Morgan Kaufmann Publishers, 2001.

[13] John Newborough and Susan Stepney, "A Generic Framework for Population-Based Algorithms, Implemented on Multiple FPGAs," *Artificial Immune Systems: 4th International Conference, ICARIS 2005,* Banff, Canada, pp. 43-55, 2005.

[14] Leandro N. de Castro and Fernando J. Von Zuben, "The Clonal Selection Algorithm with Engineering Applications," *GECCO 2000,*

Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, pp. 36-37, 2000.

[15] Leandro N. de Castro and Fernando J. Von Zuben, Learning and Optimization Using the Clonal Selection Principle *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, vol. 6, pp. 239-251, 2002.

[16] Leandro N. de Castro and Jon Timmis. *Artificial Immune Systems: A new computational intelligence approach* , Springer-Verlag, 2002.

[17] Macfarlane Burnet. *The clonal selection theory of acquired immunity*, Nashville, Tennessee, U.S.A.: Vanderbilt University Press, 1959.

[18] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*, USA: The MIT Press, 2004.

[19] Marco Dorigo, Vittoria Maniezzo, and Alberto Colorni, The Ant System: Optimization by a colony of cooperating agents *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, pp. 29-41, 1996.

[20] R. E. Callard, J. Stark, and A. J. Yates, Fratricide: a mechanism for T memory-cell homeostasis *Trends in Immunology*, vol. 24, pp. 370-375, Jul, 2003.

[21] William E. Paul. *Immunology - Recognition and Response : Readings from Scientific America magazine / edited by William E. Paul.*, USA: Scientific American Inc., W. H. Freeman and Company, 1991.

[22] Z. H. Farooqi and R. R. Mohler, Distribution models of recirculating lymphocytes *IEEE Transactions on Biomedical Engineering*, vol. 36, pp. 355-362, Mar, 1989.