An Idiotypic Solution Sieve for Selecting the Best Performing Solutions in Real-World Distributed Intelligence

Shashi Shekhar Jha* and Shivashankar B. Nair †
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati-781039, INDIA
Email: {j.shashi*, sbnair†}@iitg.ernet.in

***Abstract*—Jerne's Idiotypic Network theory features autonomous network formation, adaptation, learning and self-stabilization, all of which find extensive applications in computational realm. Researchers have used this model in a myriad of applications, however, the use of this model in real networked environments has hardly been addressed. This paper describes an *Idiotypic Sieve* to filter out the optimal solutions from a set of available solutions for a set of heterogeneous problems that could occur asynchronously or concurrently across a real network. The *Idiotypic Sieve* described herein, is conceived by emulating an Idiotypic network wherein antibodies (solutions) within a real physical network asynchronously interact with one another and also with the antigens (problems) in a distributed and decentralized manner and stimulate and suppress one another consequently changing their respective global populations across the network. The antibodies (solutions) are provided the much required mobility across the network by a set of mobile agents that autonomously patrol and migrate to nodes that are invaded by the antigens (problems). Emulation results carried out on a real network portrayed in this paper, show the effectiveness of the *Idiotypic Sieve* in generating and controlling the populations of both optimal and generic solutions to the heterogeneous set of problems.**

Keywords–Idiotypic networks; Emulation; Distributed Intelligence; Mobile agents; Typhon;

I. Introduction

In the last decade, concepts derived from the biological immune system have drawn significant attention and have proved to be a potential source of inspiration for novel approaches to solve complex computational problems [1]. The immune inspired models have been employed in various research areas including computer security, pattern recognition, fault detection, autonomous navigation in robots [2], natural language processing [3] and a variety of other applications [4]. One such model is the Idiotypic network model proposed by Jerne [5] which postulates that the antibodies are in continuous interaction with each other within the body. This makes the Idiotypic model inherently autonomous along with the ability to tune itself. The interactions among the antibodies through the coupling of their paratopes and idiotopes modulate their responses against any antigenic attack.

Farmer $et\ al.$ [6] proposed a computational model for the change in concentrations of antibodies based on this Idiotypic network model. This computational model calculates the rate of change of concentrations of an antibody as a cumulative sum of the amount of stimulations and suppressions accumulated by it from other antibodies along with the feedback (stimulation) received from neutralizing the antigen. Death of antibodies due to ageing also affects this concentration by decrementing its value. The continuous interactions among the antibodies make the Idiotypic network model a dynamic one. However, in most AIS literature [7], these concentrations are always presumed to be a mere numeric single valued parameter. In addition, most of the implementations available for the Idiotypic network model are in the form of *simulations* [8] thus providing less room for their practical viability. This calls for an architecture or framework which can emulate the inherent distributed and decentralized characteristics of the Idiotypic network model for use in real systems. The interactions among the antibodies, which can act as solutions serving nodes in a network within a pervasive environment [9], can be exploited to embed and evolve distributed intelligence. The same can also be used in the embedding intelligence into the applications in the domain of the Internet-of-Things (IoT) [10] and Cyber-Physical Systems (CPS)[11] for selecting a set of better solutions based on the needs and locally available parameters of the system.

In this paper, we present an emulated Idiotypic network acting as a sieve to arrive at a set of optimally performing solutions in a real system of asynchronously operating networked nodes. In this context, the solutions mean a program to service a requirement/request at a node. These requirements could be of different type and can occur concurrently across a network. The novelty of our approach is that the environment comprising a network of nodes acts as an active entity (instead of being a passive facilitator) wherein the solutions (antibodies) are scattered in the form of payloads carried by mobile agents [12]. These agents selectively mitigate the problems (service requirement) arising at different nodes in a decentralized and distributed manner. While the best performing antibodies eventually dominate the network, the least performing ones are automatically purged from the system by the Idiotypic Sieve. The main contribution of our work is the manner in which a real-world

978-1-4673-8675-3/15 $31.00 © 2015 IEEE
DOI 10.1109/AIMS.2015.22

implementation of the Idiotypic network can be conceived within a completely distributed setting. In the succeeding sections, we provide a background on the use of mobile agents in realizing AIS based applications followed by a description of the proposed Idiotypic Sieve. Subsequent sections cover the experimental results, discussions and conclusions.

II. Mobile agents As Antibodies

Mobile agents are autonomous programs that have the capability to migrate within a network of nodes, carry out executions on behalf of a user, process data at remote servers, carry payloads, clone and also terminate themselves when required [13]. Dasgupta *et al.* [14] describe a system for intrusion/anomaly detection and subsequent responses in networked computers using mobile agents. In their approach, the mobile agents roam around the computer nodes and routers and monitor the situation of the network. Godfrey and Nair [15] describe an architecture of a multi-robot system that uses the AIS and mobile agents to service robots. Based on pain, nodes that control the robot are triggered to indicate an antigenic attack. Mobile agents moving in a round-robin manner within the network carry the programs (antibodies) to decrease the pain levels of the robot. Bakhouya *et al.* [16] use the Idiotypic network model to regulate the population of mobile agents in a network. They assign three behaviours to each mobile agent viz. clone, kill or move. The mobile agents choose the behaviour having highest concentration while the inter-arrival times at the nodes constitute the antigen.

Although researchers have used mobile agents to realize AIS models, the *mutual interactions* among these agents to evolve better solutions have not been addressed. The mobile agents can be used to collect feedback on the effectiveness of the solutions they carry, from a node. This feedback can be translated into interactions viz. stimulations and suppressions among agents to evolve a better set of solutions manifested in the form of real-world distributed intelligence.

III. Conceptualizing an Idiotypic Network

In the computational world, an Idiotypic network can be emulated just as in [17] wherein a number of nodes form a network. An attack on a node is synonymous to a problem faced at a node. Possible solutions to such problems are carried by a set of heterogeneous mobile agents as their individual payloads; each agent carrying one solution each. These agents act as antibody carriers. In a real world scenario, the Idiotypic network evolves not by forming physical links between each and every agent or by manipulating global shared variables signifying the concentrations of the individual antibodies, but by increasing (stimulating) or decreasing (suppressing) the actual populations (concentrations) of the various types of antibodies in the network in a distributed and decentralized manner.

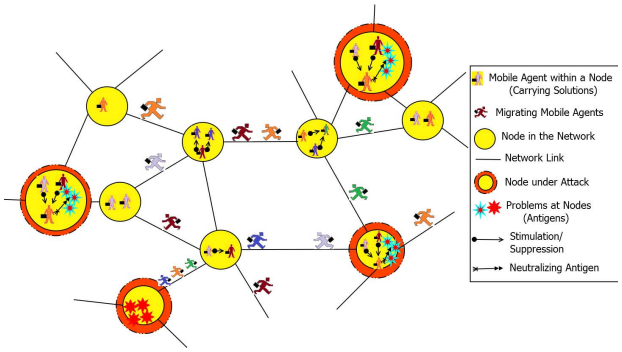


Figure 1. A portion of a network of nodes (depicting the components within) on which Idiotypic Sieve was implemented.

The interactions (stimulations and suppressions) among the antibodies cause dynamic changes in their respective populations thus contributing to a dynamic network. Antibodies (mobile agents) by themselves remain oblivious of the concentration of their individual populations (number of agents) within the network.

Figure 1 depicts a portion of the real network on which the Idiotypic Sieve is embedded. As can be seen the portion contains a number of networked nodes which constitute computers, devices and/or robots. Since the nodes are heterogeneous in nature the network could be either static (stationary) or dynamic (mobile). Mobile agents that carry the antibodies (solutions to problems) for a range of antigens (problems or service requirements) as their payload migrate constantly within the network. A node-under-attack is shown with a red periphery indicating that the node is in need of a solution carried by an agent. For instance, a node-under-attack can be a mobile robot requiring an obstacle avoidance program while there can be multiple mobile agents carrying different solutions for obstacle avoidance. In the portion shown in Figure 1, two nodes are attacked by a blue antigen while another is attacked by a red one indicating concurrency and heterogeneity in the nature of attacks. This node flashes danger signals, described later, by virtue of which it attracts the right set of agents that contain solutions to the problem at hand. The solution carried by one of the agents that arrive at the node-under-attack is selected and used to solve the issue. Based on the feedback received after this, the set of agents stimulate and/or suppress one another before migrating to other nodes. Stimulations accumulated over time at various nodes cause the agent to clone thus increasing its population, while suppressions tend to decrease its life-time which eventually results in its death and consequent removal from the network. The mechanism is thus akin to that of the Idiotypic network [5] and works on a real, asynchronous, distributed and decentralized environment where concurrent and heterogeneous service requirements could occur at many nodes in the network.

In the next section we describe the proposed Sieve to evolve the set of optimally performing solutions using local stimulations and suppressions within the nodes.

IV. The Idiotypic Sieve

The Idiotypic Sieve is the underlying mechanism that is embedded within a real network of nodes such as the one shown in Figure 1. The entire system including the Idiotypic sieve consists of a physical network of nodes (such as networked mobile robots). In addition, there exists a set of mobile agents, each of which carries a solution to a problem (service requirement) that may occur at various nodes in the network. These problems constitute the antigenic attack at a node. Each node in the network is equipped with an agent platform to facilitate all the functionalities of mobile agents. The problems and the solutions have their own descriptors. Descriptors need to fashioned *a priori* based on the application under consideration. The problem descriptors form the epitopes of an antigen while the solution descriptors act as the paratopes of the antibodies. There is an affinity function, ψ_{p_i} for each problem p_i, which defines the degree of interaction between the epitopes of an antigen and the paratopes of the antibody i.e. the affinity of a solution carried by a mobile agent to the problem p_i. A feedback function, ξ_{p_i} for each problem p_i, provides a performance measure for the solution applied to circumvent the problem p_i. This feedback ξ_{p_i} can be attributed to antigenic stimulation in the proposed Idiotypic Sieve.

The occurrence of a problem or service requirement at a node initiates an antigenic attack. These attacks can be sparse, for instance, a robot requesting an obstacle avoidance routine or dense such as the detection of fire by a set of fire sensors (spread across the network) raising an alarm at all the respective nodes. Different types of antigens (problems) may also attack several nodes concurrently in which case the respective services (solutions) may have to be provided accordingly.

A. Circumventing an Attack

In the work presented herein, we have used a mobile agent based network service model similar to the one proposed in [18]. The mobile agents that carry the solutions continuously migrate within the network of nodes using the PherCon migration strategy, as described in [19]. As soon as a node detects a problem or requirement of a service, it starts emanating danger signals to its immediate neighbours which in turn diffuse the same onto their neighbours at a lesser intensity than that received. As shown in Figure 1, these danger signals thus penetrate the immediate neighbourhood of the node-under-attack similar to the pheromone diffusion model proposed by Godfrey and Nair [19].

Whenever, an agent detects a danger signal at a node, it ascertains whether it can cater to the attack by calculating the affinity ψ between the solution it is carrying (antibody) and the problem descriptor of the antigen within the danger signal. This process is akin to the paratope-epitope matching. If the affinity ψ is less than a threshold Γ, the affinity threshold, the mobile agents ignore the danger signals and continue to migrate to other nodes. However, if this affinity ψ is found to be greater than Γ then the mobile agent assumes that it can cater to the problem at the node-under-attack. It then moves on to the node-under-attack by following danger signal strength gradient. This gradient aids the mobile agent which has detected the danger signal to reach the node-under-attack via the shortest path [19]. This mechanism of attracting the relevant mobile agents could lead to many of such reaching the node-under-attack. However, those mobile agents which are redundant (carry the same solutions) at the node-under-attack continue to their migration on the network. Hence, the node-under-attack gets populated with several mobile agents having distinct solutions to the problem.

One of the solutions carried by the mobile agents within the node-under-attack is selected to neutralize the antigen using the roulette wheel approach over the respective affinities of the solutions carried by these agents. After neutralizing the attack, the mobile agent carrying this solution receives the antigenic descriptor and the antigenic stimulation using the feedback function ξ_{p_i}, both of which are stored within its payload.

B. Concieving the Idiotypic Sieve

Once the selected mobile agent receives the antigenic feedback, all agents within the node-under-attack interact with each other after which they continue their migration to other nodes in the network. Such interactions could take place at all nodes in the network. The mobile agents interact with only those agents which carry solutions to similar problems. This means that their respective antigenic descriptors are similar. This similarity can be attributed to the paratope-idiotope matching within the Idiotypic network model. A simple method to find the related agents could be based on the affinity function ψ as discussed in the previous section. However, more complex matching functions can be derived depending upon the requirements and nature of the system or application under consideration.

Once an mobile agent has identified the set of related agents with whom it can interact within a node, it exchanges the stimulation and suppression signals. The exchange of the stimulations or suppressions is based on the past performance ξ_{p_i} of mobile agents for servicing the problem p_i. Mobile agents having higher ξ_{p_i} values are stimulated by the rest while the higher ones suppress those with lower values. These stimulations or suppressions are accumulated within a parameter called an *endurance potential* ρ_{m_i} of a mobile agent m_i which is defined as the difference between the accumulated value of the stimulations and suppressions stored within each agent. The dynamics governing the value

of ρ_{m_i} are discussed later. If the value of ρ_{m_i} crosses a maximum threshold (ρ_{max}), the mobile agent m_i clones itself whereas if the same goes below a minimum threshold (ρ_{min}) the agent m_i terminates itself (*apoptosis*). Thus cloning changes the populations of those agents whose solutions are more effective which in turn dominate the network. Those agents whose solutions are ineffective are purged out of the system. The mechanism described herein thus acts as an Idiotypic Sieve to filter out ineffective solutions while retaining the more effective ones in the network. It may be noted that the Idotypic interactions are local and asynchronous in nature, across the network. The mechanism also does not use any globally shared parameters and is distributed and decentralized. As can be observed, the Idiotypic Sieve is not dependent on the number of nodes in the network thus making the whole mechanism scalable.

In the proposed Idiotypic Sieve, the change in the concentration of antibodies emerges as the cumulative effect of the stimulations or suppressions received by the individual antibodies. The equations that govern the dynamics of the formation of the Idiotypic Sieve and the consequent changes in the populations of various mobile agents carrying their respective solutions within the network are given below.

The endurance potential ρ_{m_i} for a mobile agent m_i accumulates all the stimulations and suppressions including the antigenic ones. The value of ρ_{m_i} when a solution is selected to neutralize a problem p_j at the node-under-attack is given by:

$$\rho_{m_i}(n+1) = \rho_{m_i}(n) + \xi'_{p_j}, \quad \xi'_{p_j} = \frac{1}{1+e^{(\alpha-\beta\xi_{p_j})}} \tag{1}$$

where,

$\rho_{m_i}(n)$ denotes the value of ρ_{m_i} at the n^{th} instant. ξ'_{p_j} is used to squash the value of ξ_{p_j} between 0 and 1.

The change in the value of ρ_{m_i} due to interactions within a node is given by Equation 2.

$$\rho_{m_i}(n+1) = \rho_{m_i}(n) + \sum_{j \in S_T} S^t_{m_j} - \sum_{k \in S_U} S^u_{m_k} \tag{2}$$

where, $S^t_{m_j}$ denotes the stimulations received from the mobile agent m_j and $S^u_{m_k}$ denotes the suppressions received from the mobile agent m_k within the node. S_T and S_U are the set of related mobile agents within the node that impart the stimulations and suppressions respectively to the antibody carried by the mobile agent m_i. The values of S^t and S^u is calculated as:

$$S^t_{m_j} = \frac{k_1 \rho_{m_j}(n)}{N_{m_j}}, \quad S^u_{m_k} = \frac{k_2 \rho_{m_k}(n)}{N_{m_k}} \tag{3}$$

where, N_{m_i} for a mobile agent m_i denotes the difference between the total number of unique antigenic descriptors that the mobile agent m_i is carrying and the number of problem descriptors matching with the related agents. k_1 and k_2 are the non-zero positive constants.

The ageing due to the lifetime of the antibodies as mentioned in the Farmer's model [6] is implemented by conferring a fixed number of hops (Δ) to every *cloned mobile agent* (antibody) in the network. The value of Δ is reduced by unity whenever a mobile agent lands on a node in the network. Once the value of Δ becomes zero, the mobile agents are terminated and hence removed from the system.

V. Experimental Setup

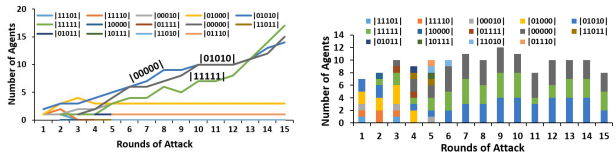
The proposed model was implemented over a real networked environment using *Typhon* mobile agent framework [20]. Though the testing of the sieve was performed using static networks, it was felt that the study of its performance on dynamic networks would throw more light on its efficacy in mobile computing environments. The dynamic networks were generated by continuously altering the neighbours of the nodes in the *Typhon* based network with a given probability.

In the implementation, we considered binary (5-bit) sequences to form antigens which were presented at several nodes (nodes-under-attack) to emulate multiple heterogeneous and concurrent antigenic attacks within the network. This binary sequence forms a generalized representation of a multi-dimensional vector manifested as a problem at a node. The sequence could be synonymous to a set of sensor vectors or the state information at a node in the network (problem descriptor) for which an action (solution) is to be taken. We consider the corresponding best antibody to be a 5-bit complemented sequence capable of neutralizing the antigen. In the proposed Idiotypic Sieve, every mobile agent carries with it one 5-bit neutralizing sequence (solution). The Hamming distance [21] (η) is used for deriving the affinity function (ψ) between the binary sequences constituting the antigens and the antibodies. The decimal equivalent of the binary operation ***XoR*** between the binary sequences of antigens and antibodies was used as the reward function (ξ). The value of ψ in the implementation is given by:

$$\psi(E,P) = \eta_{max} - \eta(E,P) \tag{4}$$

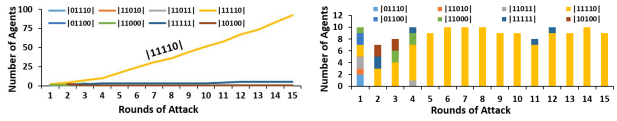
where, E and P denote the bit sequences of the epitopes of the antigen and the paratopes of the antibody respectively. η_{max} is the maximum possible Hamming distance between E and P.

Initially since there are no antibodies in the network, we allowed the nodes to generate random antibodies (5-bit binary sequences) pro-actively after the danger signals had died down. The nodes generate a random 5-bit pattern and ascertain its ψ value. If the same is greater than Γ (Section IV-A) then the node treats this pattern as the antibody or solution to neutralize the antigen. This new antibody or solution is then encapsulated within a mobile agent and released



(a) (b)

Figure 2. Antigens attacking in each round: [00000], [11111], [10101] (a) The change in the population of different Antibodies (b) The number of Antibodies generated



(a) (b)

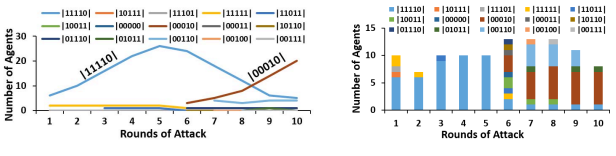
Figure 3. Antigens attacking in each round: [00000], [00001], [00011] (a) The increase in the population of generic Antibody along with other Antibody populations (b) The number of Antibodies generated

into the network. Each experiment consisted of multiple rounds of concurrent and heterogeneous antigenic attacks. The antibodies generated in each round were retained for use in the subsequent ones.

VI. Results and Discussions

Experiments were performed on dynamic networks comprising 100 nodes by forcing distinct antigens viz. 5-bit sequences to attack separate nodes chosen at random from the network. The number and types of antibodies (mobile agents equipped with 5-bit binary sequences) generated/terminated in each round were recorded to plot the graphs. The following values of the parameters were used in the experiments: $\Gamma = 4, \alpha = 10, \beta = 0.5, k_1 = 0.5, k_2 = 0.5, \rho_{max} = 1.5, \rho_{min} = 0.5, \Delta = 1000$

The graph in Figure 2(a) shows the change in the population of antibodies in each round of antigenic attack within the dynamic network of 100 nodes. The antigens used for the attack were: [00000], [11111], [10101]. As can be observed, a total of 14 unique antibodies were generated by the nodes in the 15 rounds of attacks. However, only three antibodies viz. [11111], [00000], [01010] which are the more optimal solutions, can be observed to be actively growing in their respective populations. The population of the rest, which constitute the non-optimal solutions, remain limited and eventually dwindle within the network indicating clearly the effectiveness of the Idiotypic Sieve to retard the generation and proliferation of the less performing antibodies while allowing the dominant ones to grow in number. It can be noted that the three dominating antibodies are the complementary bit sequences of the antigens used for the attacks. Also, the antibodies [11110] and [01000] have maintained a small population 3 and 1 respectively in the network. The reason behind this is that the left over population of both these antibodies [11110] and [01000] have not encountered



(a) (b)

Figure 4. (a) The change in the population of different Antibodies (b) The number of Antibodies generated, when Antigens [00000], [00001], [00011] attacked the nodes in the initial five rounds while [11100],[11101],[11001] attacked in later five.

their related dominating antibodies within the nodes of the network so that they can be suppressed and completely removed from the system.

The graph in Figure 2(b) depicts the number of antibodies of each type generated in each round of attack. As mentioned earlier (Section V), the antibodies can be generated either by the nodes-under-attack (randomly, if they do not receive any antibody) or by cloning. It can be observed in Figure 2(b) that different types of antibodies are generated till round 5. Beyond this only the three distinct antibodies dominate within the network. This shows the capability of the Idiotypic Sieve to selectively maintain the population of best performing solutions within the network while weaning of the remaining ones.

To verify the ability of the Idiotypic Sieve to retain not only the optimal solutions but also generic ones, another set of three antigens viz. [00000], [00001], [00011] having very small Hamming distances amongst each other were used to attack the network. The graphs in Figures 3(a) and 3(b) show the results gathered through 15 rounds of their attack. It can be clearly seen from the graph in Figure 3(a) that the antibody [11110] solely dominates the network. The computed ξ values for the dominant antibody against the three antigens [00000], [00001] and [00011] is found to be 30, 31 and 29 respectively using the method of calculation mentioned in Section V. These constitute the three highest ξ values achievable within the network which makes this antibody receive maximum stimulations. It also means that it suppresses all other antibodies. It may also be noted that the bit sequence [11111] can cater to the antigens [00000] and [00001] but not to [00011] since the ψ value in case of [00011] is less than 4 i.e. $\psi < \Gamma$ (Section IV-A). Hence the population of [11111] can be observed to be thriving within the network. Similar observations can be drawn from the graph shown in Figure 3(b) wherein the antibody [11110] generates clones from round 5 to 15. Traces of antibody [11111] can also be observed in round 11 and 12.

The graph in Figure 4(a) shows the variations in the populations of different antibodies when the set of antigens - [00000], [00001] and [00011] were used to attack in the first five rounds while the set of antigens - [11100], [11101] and [11001] were used in the rest of the rounds. As can observed, the population of the antibody [11110] grows

initially till round 5 and then starts to wane due to death of its clones which have a fixed value of Δ. On the contrary the new antibody [00010] is allowed to grow within the network since its requirement at the nodes is higher. This emphasizes the adaptive nature of the Idiotypic Sieve which endeavours to retain and proliferate only those antibodies that are required or are in demand in the network. The graph shown in Figure 4(b) also depicts the waxing and waning of the population of the antibodies [11110] and [00010] respectively in the network. From the results it is clear that the Idiotypic Sieve can efficaciously empower a network of nodes to churn and retain the populations of the optimal solution(s) required, on demand while suppressing the proliferation of the others. Further, since all experiments were performed using dynamic networks, one may note that this sieve can cope up with the rigours encountered in mobile computing environments.

VII. Conclusions

In this paper, we portrayed an Idiotypic Sieve that can be used in conjunction with a real distributed network of nodes. The Idiotypic Sieve is conceived by emulating an Idiotypic network model within a computational networked environment. The performance of the Idiotypic Sieve was tested in both static and dynamic networks. It was found that the results in both the cases were similar indicating its robustness and usefulness in mobile computing scenarios. Embedded within a distributed network of nodes, the Idiotypic Sieve selectively evolves the best performing solutions based on the current demand (problems) in the network and purges off the others. The results further show that the Idiotypic sieve also generates generic solutions that can cater to a range of problems with high degrees of effectiveness. It can also cope up with multiple, concurrent, asynchronous and heterogeneous attacks at the various nodes in the network. Scalability of the system is also inherent as the sieve functions oblivious of the number of nodes in the network. In future, we plan to hybridize this Sieve with the clonal selection mechanism [22] so as to generate memory cells that will be retained within the network and be triggered as a secondary response on demand thus reducing the time taken to contain the respective attacks. Learning and sharing *on-the-fly* by mobile agents [23] can further ameliorate the quality and time taken to generate a solution especially when the dimensionality of the epitopes/paratopes are high. This Idiotypic Sieve can also be used in a plethora of applications ranging from networked robotic systems, Internet-of-Things and Cyber-Physical Systems to pervasive and ambient computing.

Acknowledgements

The first author wish to acknowledge Tata Consultancy Services (TCS) for their support during the course of this research.

References

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

[2] A. Raza and B. R. Fernandez, "Immuno-inspired robotic applications: A review," *Applied Soft Computing*, vol. 37, pp. 490 – 505, 2015.

[3] S. K. Borgohain and S. B. Nair, "An immuno-inspired approach towards sentence generation," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, ser. GECCO '15. ACM, 2015, pp. 97–104.

[4] D. Dasgupta, S. Yu, and F. Nino, "Recent advances in artificial immune systems: Models and applications," *Applied Soft Computing*, vol. 11, no. 2, pp. 1574 – 1587, 2011.

[5] N. K. Jerne, "Towards the network theory of the immune system," *Ann. Immunol.(Inst. Pasteur)*, vol. 125, pp. 373–389, 1974.

[6] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1, pp. 187–204, 1986.

[7] D. Dasgupta, *Artificial Immune Systems and Their Applications*. Springer Publishing Company, Incorporated, 2014.

[8] J. Greensmith, A. Whitbrook, and U. Aickelin, "Artificial immune systems," in *Handbook of Metaheuristics*. Springer, 2010, pp. 421–448.

[9] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, 2001.

[10] H. Kopetz, "Internet of things," in *Real-Time Systems*. Springer US, 2011, pp. 307–323.

[11] W. Wolf, "Cyber-physical systems," *Computer*, vol. 42, no. 3, pp. 88–89, 2009.

[12] J. E. White, "Mobile agents," in *Software agents*. MIT press, 1997, pp. 437–472.

[13] A. Outtagarts, "Mobile Agent-based Applications : a Survey," *International Journal of Computer Science and Network Security*, vol. 9, pp. 331–339, 2009.

[14] D. Dasgupta, "Immunity-based intrusion detection system: a general framework," in *Proc. of the 22nd NISSC*, vol. 1, 1999, pp. 147–160.

[15] W. W. Godfrey and S. B. Nair, "An Immune System Based Multi-robot Mobile Agent Network," in *Artificial Immune Systems*, ser. LNCS, 2008, vol. 5132, pp. 424–433.

[16] M. Bakhouya, M. Nemiche, and J. Gaber, "An adaptive regulation approach of mobile agent population size in distributed systems," *International Journal of Intelligent Systems*, vol. 00, pp. 1–16, 2015.

[17] S. S. Jha, K. Shrivastava, and S. B. Nair, "On emulating real-world distributed intelligence using mobile agent based localized idiotypic networks," in *Mining Intelligence and Knowledge Exploration*, ser. LNCS. Springer International Publishing, 2013, vol. 8284, pp. 487–498.

[18] W. W. Godfrey, S. S. Jha, and S. B. Nair, "On a mobile agent framework for an internet of things," in *International Conference on Communication Systems and Network Technologies*, 2013, pp. 345–350.

[19] W. W. Godfrey and S. B. Nair, "A Pheromone Based Mobile Agent Migration Strategy for Servicing Networked Robots," in *Bio-Inspired Models of Network, Information, and Computing Systems*. Springer, 2012, pp. 533–541.

[20] J. Matani and S. B. Nair, "Typhon - A Mobile Agents Framework for Real World Emulation in Prolog," in *Multi-disciplinary Trends in Artificial Intelligence*. Springer, 2011, pp. 261–273.

[21] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[22] K. Rajewsky, "Clonal selection and learning in the antibody system," *Nature*, vol. 381, pp. 751–758, 1996.

[23] S. S. Jha and S. B. Nair, "On a multi-agent distributed asynchronous intelligence-sharing and learning framework," in *Transactions on Computational Collective Intelligence XVIII*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2015, vol. 9240, pp. 166–200.