J Intell Manuf
DOI 10.1007/s10845-017-1370-y

CrossMark

Artificial immune systems based multi-agent architecture to perform distributed diagnosis

Andre Dionisio Rocha[1] · **Pedro Lima-Monteiro**[1] · **Mafalda Parreira-Rocha**[1] · **Jose Barata**[1]

Received: 6 January 2017 / Accepted: 22 October 2017

Abstract Cyber-physical systems (CPS) emerge as a new idea to implement new manufacturing paradigms. There paradigms aim at answering the socio-economic factors that characterise modern enterprises, such as mass customisation and new markets. The authors propose an architecture that performs distributed diagnosis. The proposed solution uses artificial immune systems (AIS) to perform evolutionary diagnose. Industrial approaches to machine diagnosis are centralised. The authors pretend to make a CPS capable of distributed diagnosis with learning capabilities. An architecture capable of machine diagnosis and learning is also presented. This is done by bio-inspired algorithms. These were rated by a fuzzy inference system. The algorithms were tested for situations a system may endure and for their learning capability. The results of the obtained research, study and development are hereby presented. These results constitute proof of the sustainability of the AIS paradigm as a solution to distributed diagnosis.

Keywords Artificial immune systems · Cyber-physical systems · Multi-agent systems · Bio-inspired algorithms

✉ Andre Dionisio Rocha
andre.rocha@uninova.pt

Pedro Lima-Monteiro
pedro.monteiro@uninova.pt

Mafalda Parreira-Rocha
mafalda.rocha@uninova.pt

Jose Barata
jab@uninova.pt

1 Uninova - CTS, Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2825-149 Caparica, Portugal

Introduction

New trends of market are focused on a high level of customisation by demands of the consumers. Along with the current global financial crisis this obligates the companies, mostly small and medium enterprises (SME), to change their strategies and approaches. Although this need is real, and a huge effort has been done to give the SMEs an effective solution, this is yet to be achieved. According to the majority of the enterprises, a self-diagnosis/learning system would provide the high level of customisation needed. This would be the result of a more efficient error recovery, thus making the production more lucrative.

In order to offer highly dynamic and flexible manufacturing environments, new production paradigms recently appeared to fulfil this new SMEs requirements, such as Evolvable Production Systems (EPS) (Frei et al. 2007; Lohse et al. 2006; Barata and Camarinha-Matos 2003), Holonic Manufacturing Systems (HMS) (Leitão and Restivo 2006, 2008) and flexible manufacturing systems (FMS) (Tetzlaff 1990; Storey 1994).

These paradigms' goal was to give manufacturers the ability to face new challenges, providing flexibility and adaptability. These systems are mostly supported by distributed systems, constituted by logical entities responsible for abstracting different components of the production system.

This distributed approach opens the possibility to perform a different type of diagnosis. With the different modules presented on the production environment it is possible for each module to diagnose itself. Self-diagnosis is an important and efficient solution to face disturbances and failures during execution whilst not decreasing the performance of the other components, since the problems are solved locally.

To do so, the authors did some research on artificial immune systems (AIS), which is believed to be an effec-

Published online: 01 November 2017

tive solution. This type of systems is biased on bio-inspired algorithms which aim at replicating the functioning of the human immune system, used to fight diseases. This paradigm is mainly supported by four different algorithms which aim at mimicking the functioning of the human immune system. In this study, these algorithms were implemented using the Java Agent Development Framework (JADE). Once this was achieved, the algorithms were ranked according to their performance in a simulated environment. This environment was designed to be like those one can find in real industrial scenarios. The algorithms were then tested under scenarios they would face in an industrial environment and the algorithm that was most suitable for the simulated environment was identified. The main purpose of this study is, therefore, to understand whether this type of systems could perform well in an industrial-like scenario. Well is hereby acknowledge as the ability to cope with abnormalities in the scenario and to do it in a decent amount of time given the environment where it is inserted.

Therefore, this work will start by presenting some literature on the aforementioned subjects in “Related work” section. In “AIS algorithm’s selection” section, a view on the reasons to choose one algorithm over the others will be given. In “A different diagnostic approach” section, the developed architecture will be presented and thoroughly explained. In “Achieved framework” section, the developed work will be presented, with an insight on the accomplished framework. In “Results” section, the obtained results will be presented, explained and further analysed. Lastly, in “Conclusion” section, some conclusions on the accomplished work and obtained results will be drawn.

Related work

In order to put the reader on the current mind-set for this paper, some related work on its main subjects will be given in this Section.

Multi-agent systems

Usually, the emergent manufacturing paradigms are implemented based on a multi-agent system (MAS). This kind of technology makes it possible to completely decentralise the production system entities whilst giving them the capacity to evolve and change, accordingly to its society. These kind of systems (Peixoto et al. 2015) are now beginning to be used in a wide variety of applications, ranging from air traffic control, to manufacturing and even games. These systems allow these applications (Shen and Norrie 2013) to execute faster and more efficiently. It so happens because, in MASs, each component of the system (agent) works in relative autonomy from the main system.

Distributed diagnosis

Given the current evolution direction of the industrial systems, it made no sense that diagnosis kept being a centralised, non-communicative, unintelligent system on top of an intelligent, decentralised one. To enable these characteristics in diagnosis, several theories were elaborated but only some actually came to life.

In Barata et al. (2007a), the authors proposed a service oriented architecture (SOA) for diagnosis in which an extreme importance is attributed to intelligent devices, thus offering a distributed intelligence. This concept opens new doors as to what diagnosis concerns, since collaborative diagnosis becomes a possibility supported by distributed devices.

In Barata et al. (2007b), the authors propose an architecture for diagnosis in evolvable production systems (EPS), giving particular emphasis towards self-capabilities. It is also stated that there are still challenges and open issues that should be addressed. Some architectural principles are also elaborated taking into account the premises in which the current system is limited to.

In Ribeiro and Barata (2012), a validation of a co-evolving diagnosis algorithm for EPS is given. To do so, an architecture which implements distributed diagnosis was also proposed in this paper. The architecture encompassed two main phases: the initial configuration phase, where “the system designer has to establish the initial interactions between the existing modules” and the runtime phase, in which “the system may undergo structural changes” by a variety of reasons.

In this context, the main topic of this document emerges: artificial immune system (AIS), which is a diagnosis paradigm based on bio-inspired algorithms. It was developed as an image of the actual human immune system. As such, it includes concepts such as B Cells and self and non-self-cells which will be approached later on. It is a learning and adaptive mechanism that grows with the system.

Artificial immune systems

Artificial immune systems is a diverse area of research that attempts to bridge the divide between immunology and engineering. It is developed through the application of mathematical and computational modelling to immunology. These immunologic models are abstracted using algorithm (and system) design and implemented in the context of engineering. For this work, four algorithms were studied: negative selection, clonal selection, immune network theory and danger theory.

Negative selection

Its purpose is that of allowing some degree of tolerance for self-cells (those normally present in the organism). Dealing with the Immune System's ability of detecting unknown antigens (harmful cells) without prejudicing its own cells. The immune system's generated B Cells, which fight these antigens, are formed of a pseudorandom genetic rearrange. Those of which that react against the antigens are used to destroy it and replicated in the organism as matured cells (Zuccolotto et al. 2015).

Clonal selection

The main idea behind this algorithm is that only the B Cells that recognize the antigen will thrive and replicate. This principle describes the basic characteristic of an immunologic response to an antigen caused stimulus. This algorithm's main characteristics may be enumerated as follows:

(a) New cells are copies of those they derive from, and are then subjected to a high rate mutation mechanism (somatic hyper mutation).
(b) Procedural elimination of the new cells that, after mutation, endanger the survival of the non-prejudicial cells for the organism (self).
(c) Further cloning and mutation of the cloned cells that respond positively to the antigen.

This mechanism allows for a faster response to the antigen (El-Sharkh 2014).

Immune network theory and danger theory

The immune network theory algorithm is based on the assumption that the immune system maintains a regulated network of interconnected B Cells with the purpose of easing antigen detection. These cells stimulate and suppress each other with the ultimate purpose of providing stability. The connection between two B Cells is proportional to their affinity to each other (Zhong and Zhang 2012).

The danger theory algorithm is based on the idea that the immune system is more preoccupied with the entities that damage it than with those that are strange to it. Hence, this algorithm assumes that the immune system is activated by dangers signals emitted by the damaged cells (affected by the antigens or with mechanical damage). Therefore, the main challenge of this algorithm is to distinguish between the real and the fake signals. Whenever a signal is emitted, the antigen-present cells are activated. This would then stimulate the B Cells (Yin et al. 2012).

Even though there are four algorithms under the context of AIS, only the first three were further developed and implemented, because the authors considered that the last one, danger theory, was not suitable for the pretended goal.

Hybrid approaches

There are also plenty of hybrid solutions on the literature. Such solutions use AIS for a variety of reasons. In Yıldız (2009a), an hybrid solution composed of both AIS and the hill-climbing algorithm is used. In the solution, the immune algorithms are used for its speed exploration. The results obtained by the immune algorithm are used as a starting point for the tuning of the parameters by the hill-climbing algorithm. The results presented indicate that the hybrid approach tops the optimization achieved by other approaches. In Öztürk et al. (2006), an hybrid approach is also proposed to optimize design applications. The main purpose is to obtain an optimised shape based on the structure features. The proposed approach utilises both genetic algorithms and the Taguchi's method. The latter is used to obtain interval levels for the design parameters. Once achieved, the genetic algorithm bases its initial population on such intervals and computes the intermediate optimal design parameters. It concludes that the hybrid approach tops the other approaches as far as optimisation concerns. The exact same approach is used in Yıldız et al. (2007) to optimise a different component design. This component's design is also optimised in Yildiz (2013a). Again using the Taguchi method to obtain interval levels, the remainder of the proposed hybrid approach relies on the bee colony algorithm. Yet another hybrid approach is proposed in Yildiz (2009). This work proposes an approach based on AIS and the simulated annealing algorithm. The AIS algorithm used is described in this document in "Clonal selection" section. Similar to the first hybrid approach presented in this document, the AIS algorithm results are used as a starting point for the tuning of the parameters by the simulated annealing algorithm. The obtained results once more prove that a hybrid approach tops the remaining ones. Vehicle components are also used in Yildiz et al. (2016) to demonstrate the benefits of yet another hybrid approach for structural design. This turn, a combination of the gravitational search and charged system search algorithms is used. In Yıldız and Lekesiz (2017) the very same vehicle components are used to prove the viability of yet another hybrid approach. Using a combination if the charged system search and the Nelder–Mead algorithm, the proposed approach managed to reduce the weight of the component by 5.8%. The pressure vessel optimisation problem is used in Yıldız (2009b) to evaluate the performance of yet another hybrid approach. In the proposed approach, AIS are used to prevent the premature convergence of the particle swarm algorithm. The exact same approach is used in Yildiz and Solanki (2012). However, in this case, the objective is to optimize the design of a car so that its ability to withstand a crash can be largely improved.

In Yildiz (2013b), different optimisation algorithms' performance is compared when optimising the design of a vehicle component. Yet again, hybrid approaches seem to output the best results for this type of use cases. The aforementioned pressure vessel optimisation problem is also used in Yıldız (2017) to compare the following eight algorithms: gravitational search, ant lion optimiser, firefly, bat, league championship, symbiotic organisms search, imperialist competitive and charged system algorithms; with an hybrid approach composed of the genetic and hill-climbing algorithms. Other problems are also used to test the performance of the aforementioned algorithms. The results show that the hybrid approach is one of the most suitable ones for solving all the problems. Both Kiani and Yildiz (2016) and Karagöz and Yıldız (2017) provide a comparison based on the crashworthiness of a car. Whereas the former focus on five different, non-traditional methods (artificial bee colony, differential evolution, genetic algorithm, particle swarm optimisation and simulate annealing) and concluded that the most suitable one was differential evolution, the latter focused on metaheuristic algorithms such as gravitational search, cuckoo search, among others. In this last comparison, it was conclude yet again that an hybrid approach was the most suitable for the crashworthiness test case.

AIS algorithm's selection

In order to evaluate the different AIS algorithms, two models, capable of analysing the collected data, were developed. For a better understanding of the following models and studies, an explanation of what a "word" is becomes necessary. A word is a set of bits, of various byte sizes. An example of a word of half a byte is "0001", whereas a word of 1 byte can be something like "00001001". In this sense, each word represents a possible combination of inputs and outputs in each moment, corresponding to the current state of the system.

Models' development

The models were developed based on fuzzy logic. The first model defines a fuzzy inference system (see Fig. 1).

The first model (Fig. 1) uses two performance indicators (inputs of the FIS), the B-Cell Number and the time lapse, to define the system's Responsiveness Level (output of the FIS). The Responsiveness Level, $F(x(i,j))$, is defined by the FIS of each combined sample, i.e., for each sample x_{ij}, where i represents the B Cells Number input and j represents the Time Lapse input. This FIS is applied to all samples of all word sizes of the three algorithms. With the first model's outputs, the Responsiveness Level (per sample), the data is aggregated through the use of Eq. 1.

$$\bar{X}_k = \frac{\sum_{k=1}^{n} F(x(i,j))_k}{n} \quad \text{and}$$
$$\sqrt{S_k^2} = \frac{\sum_{k=1}^{n} \left(F(x(i,j))_k - \bar{X}_k\right)^2}{n-1} \tag{1}$$

Where k is the sample number for the Responsiveness Level result for each sample x_{ij}. Eq. 1's results were used on the second model (Fig. 2).

The model's (Fig. 2) result is the Responsiveness Level per algorithm. The higher the value, the better will the algorithm result be, i.e., the algorithm with the highest Responsiveness Level is the best algorithm.

Models' implementation

In order to apply the model, each indicator needs a scale. Table 1 presents the scale for each performance indicator.

The present scales were defined by experts in automotive industry. The sample average and the sample standard deviation scales are similar to the previous ones. The sample average, is also defined as very low, low, moderate, high, and

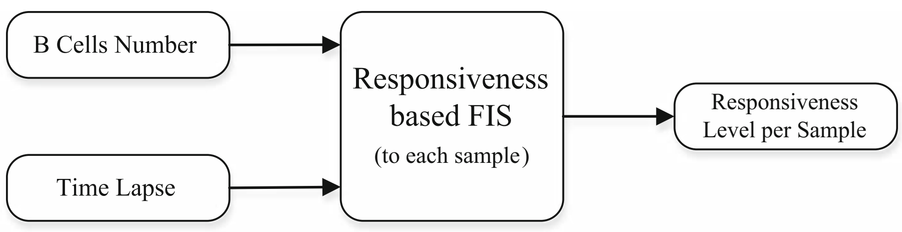

Fig. 1 Framework of the responsiveness based FIS (per sample)

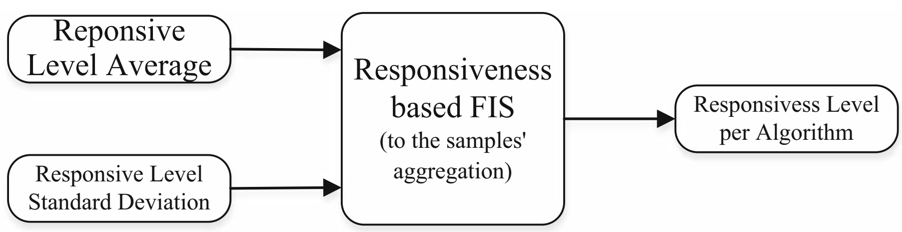

Fig. 2 Framework of the responsiveness based FIS (to the samples' aggregation per each algorithm)

Table 1 B Cells number and time lapse scales used to define the FIS first model

Scale	Description	
	B Cells number	Time lapse
Very high	> 60 B Cells	≤ 5 ms
High	Between 45 and 60 B Cells	Between 5 and 10 ms
Moderate	Between 30 and 45 B Cells	Between 10 and 50 ms
Low	Between 15 and 30 B Cells	Between 50 and 100 ms
Very low	≤ 15 B Cells	≥ 100 ms

Table 2 Extreme conditions' test

Time	B Cells	Responsiveness Level
0	0	0.917
1	1	0.069
Average	Standard deviation	Responsiveness Level
1	0	0.931
0	0.3	0.069

very high which, in B-Cells number relates to ≤ 15, between 15 and 30, between 30 and 45, between 45 and 60 and > 60 respectively (in time lapse, it relates to ≥ 100, between 50 and 100, between 10 and 50, between 5 and 10 and ≤ 5, respectively). The sample standard deviation is defined by the scale low, moderate and high. All membership functions are defined with a Gaussian shape. The FIS rules where defined based on experts' knowhow.

Models' validation

To validate the models two methods were used: the extreme conditions test, and the face validity test. Table 2 presents the results of the extreme conditions test, for both models.

For both models, the results were the expected ones according to the experts' opinion. Figure 3 presents the face validity test.

Results' analysis

In order to have a better perspective over the models results for the three algorithms, a graphical representation was defined. Figure 4 is the graphical representation of all algorithms for the six defined word sizes. The results were, according to experts, the expected ones:

- *Cure finding speed* All three algorithms revealed themselves capable of finding a cure for the error being treated. They did so in an irrelevant time lapse when compared to those of the PLC's cycle time, which is between seven and one hundred milliseconds (7 ∼ 100ms)—this scale was based on the automotive industry case).
- *Algorithms capability* An operating station was considered a station that needs, at least, one byte to read and write all inputs and outputs. It is safe to say that, from the three algorithms that were developed, two of them, clonal selection and network model, present fair, even good, results when tested on words larger than one byte, offering above 50% performance for words with a three-byte size.
- *Size coverage* This study also offers a view on which algorithm should be used, according to the word size being used. For words smaller or equal to four bits, the Negative Selection Algorithm should be used. On the other hand, for words larger than four bits, both Clonal Selection and Network Model could be used, despite the latter presenting far better results than the first.
- *Expected decaying* Even though the test results were rather satisfactory, an expected decaying on the Responsiveness Level occurred, as the word size grew bigger.
- *One to rule them all* Despite not being, as the results suggest, the best algorithm for words smaller than four bits, Network Model algorithm presents better results for all the remaining case scenarios. Moreover, the first case scenario, for words four bits sized, should be deemed irrelevant for the present analysis since all the algorithms present extremely good results for that specific case scenario.

Now that the algorithms have been compared and ranked between them, it is possible to proceed with the development of the most suitable architecture. This architecture should reflect the results of the previous ranking and consider the advantages and disadvantages of each of the algorithms.

A different diagnostic approach

The proposed approach varies from the ones currently being used in industrial scenarios. It so is due to the fact it relies in a decentralised diagnosis implementation, in order to provide a both efficient and cost-effective Error Recovery Tool.

Proposed architecture

A conceptual overview of the proposed architecture can be observed in Fig. 5. The architecture is divided in two separate layers. One global and accessible by all, to learn the different generated solutions (evolution layer). And another one constituted by the entities that run at the device and module level, responsible for detecting malfunctions and generate new cures (diagnosis and recovery layer).

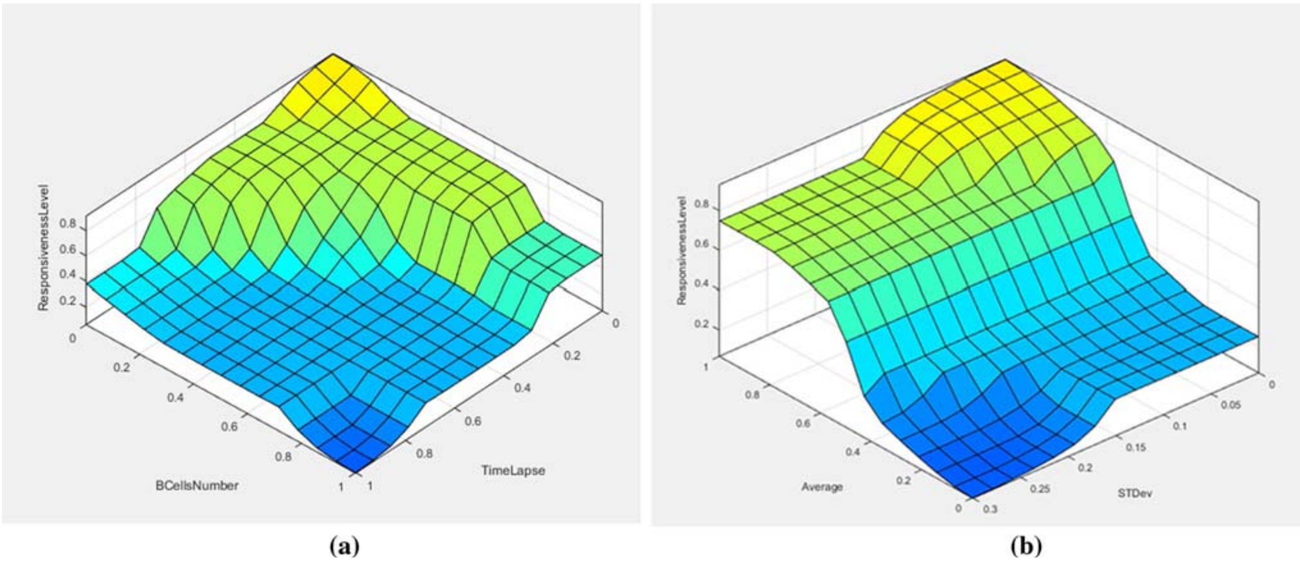

Fig. 3 Face validity test **a** Responsiveness level FIS (per sample), **b** responsiveness level FIS (per algorithm)

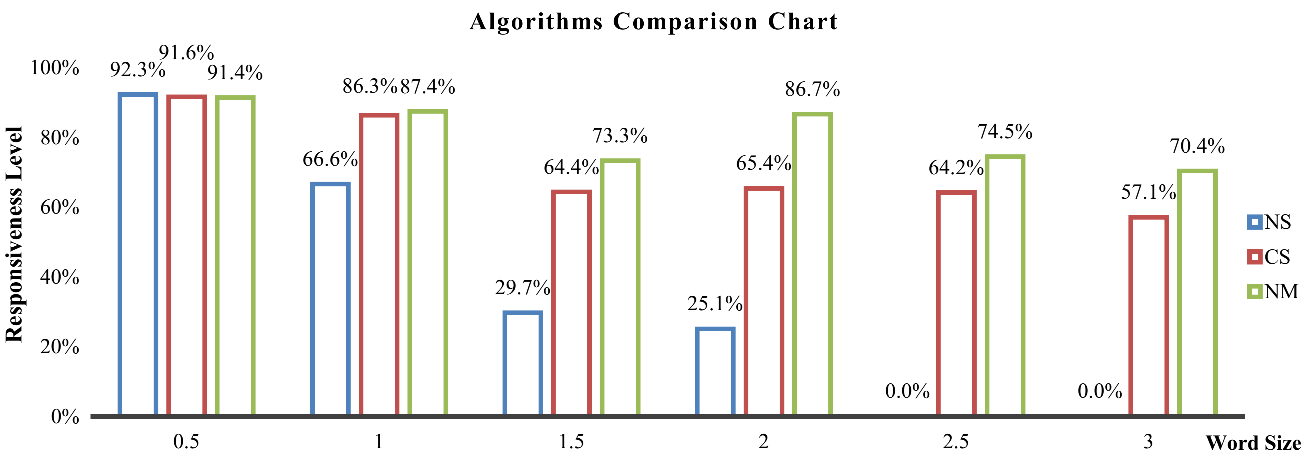

Fig. 4 Algorithms comparison chart

The architecture is constituted by four generic entities that abstract different capabilities of the system:

Cure provider agent

The Cure Provider Agents (CPA) are part of a cloud that covers the totality of the system. This cloud is available to provide cures to all the lower level entities. Each entity has an associated database where all the known possible cures are stored. Hence, when a lower level agent asks for a possible cure to solve a given problem, the CPA queries the database for possible cures to the problem.

Grouped diagnosis agent

The Grouped Diagnosis Agent (GDA) represents a group of physical devices that work together and consequently influence the execution of each other. These entities can collect the malfunctions of the resources and understanding if the errors are correlated or not. When the DAs are grouped, the resolution of the errors is responsibility of this higher-level entity. Therefore, when the GDA agent does not have a cure for a specific problem, a B Cell Agent is launched to find new possible cures.

Diagnosis agent

The diagnosis agent (DA) is the lowest level entity of the entire system. It is responsible for the diagnosing of a physical resource, such as a robot, conveyor, etc. During the execution, the DA constantly verifies the methods of a specific resource trying to find errors and failures. Once a malfunction is detected, the DA verifies if there are available cures for the detected error. If so, the DA performs the

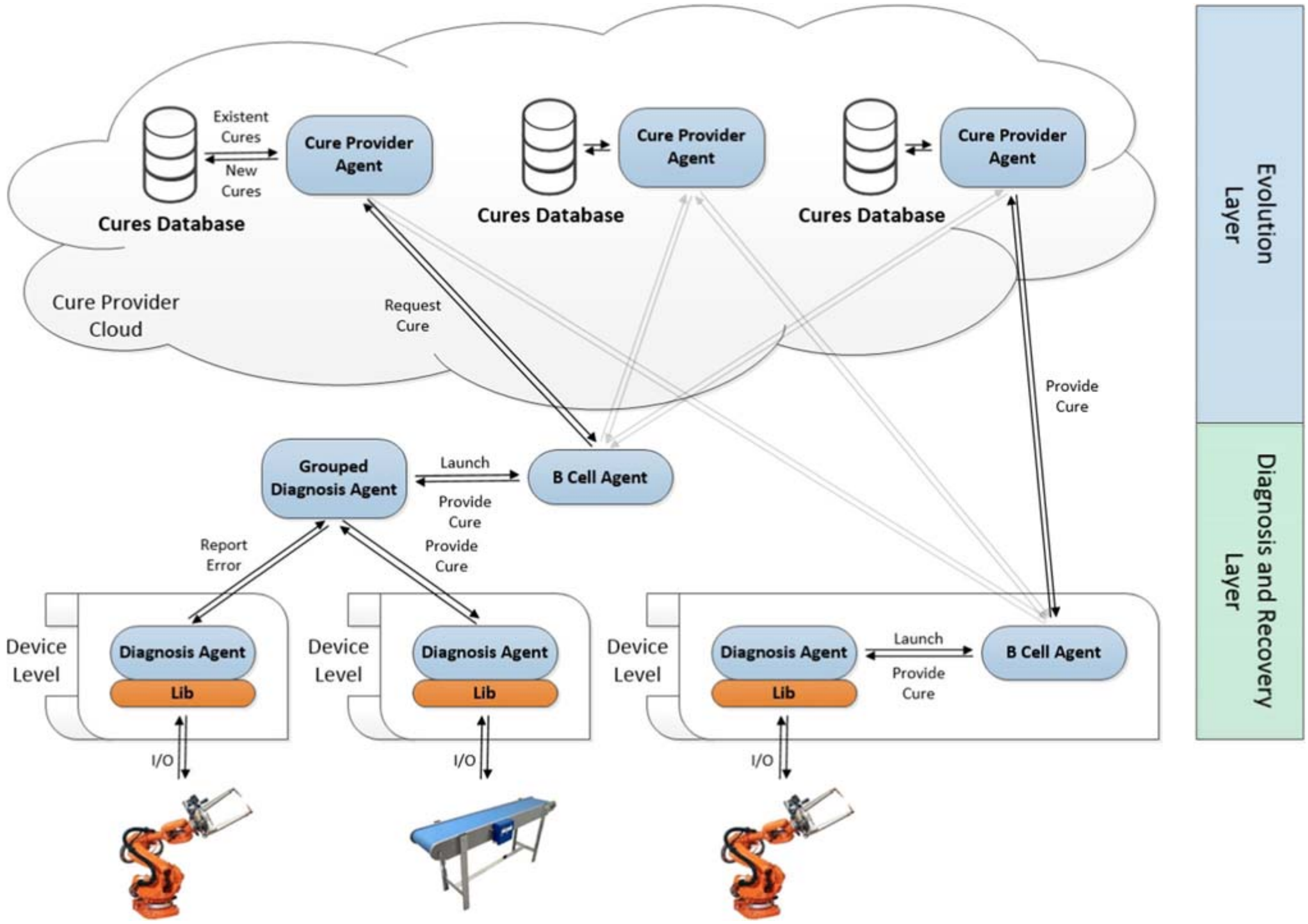

Fig. 5 Proposed architecture

cure immediately; otherwise, the DA runs the AIS algorithm to define what B Cell Agent should be launched.

B Cell Agent

The B Cell Agent (BCA) emulates a real B Cell. Hence, when a malfunction is detected and neither the GDA nor the DA have cures for the error, a BCA is created in order to find new possible cures. The creation of these entities is defined by the AIS algorithm running inside the GDA and DA, accordingly to the error that needs to be solved.

Multi-agent data model

The proposed architecture has been developed on the JADE framework. JADE is a Java based middleware that eases multi-agent systems development. One of the main system characteristics is the FIPA-compliant agent behaviours.

In order to comply with the used framework requirements, JADE agents fulfil the FIPA protocols when performing communications. JADE offers two main communication protocols: FIPA Request and FIPA Contract NET. Whilst the first one is a simple request with a possible accept/refuse answer, the second one is a negotiation between all the parties involved to find the cheapest solution. Table 3 shows the different protocols used by the different agents in the system to interact with each other.

Achieved framework

The developed framework is biased on the theoretical modelling explained in the previous chapter. Therefore, it will be centred in the four different entities that relate amongst themselves according to its needs.

Execution

During execution, the framework has different stages and functionalities. Figure 6 represents the steps performed by each entity during a malfunction detection by a DA. As each DA is periodically checking for possible errors/malfunctions, whenever an error is detected, the current state (word of bits) is retrieved from the lower level library. This word repre-

Table 3 Communication amongst agents

Initiator	Responder			
	DA	GDA	BCA	CPA
DA		FIPA Request		FIPA Contract Net
GDA	FIPA Request			
BCA	FIPA Request	FIPA Request		FIPA Contract Net
CPA				

sents the current state of the system, and the DA triggers the recovery routine.

Whenever a malfunction is detected, the DA verifies if it has a possible cure for this malfunction; if so, it performs a cure based on the locally stored method for the cure. Another possibility is when a DA has no local cures to solve the malfunction and has no group (DA is not part of a GDA). In this case, the DA launches new BCAs. The BCAs are launched according to the AIS algorithm running inside DA. The algorithm decided what BCAs and possible cures are generated, and the DA locally launched the new BCAs, each BCA works as a possible cure for the detected malfunction.

When a DA is grouped to a GDA, the DA sends the diagnosed error to the GDA making it the one responsible for managing the error. Basically, the first step is to understand if the GDA has cures for the error; if so, it sends it to the DA and immediately solves the problem. On the other hand, if a local cure is not available, the GDA launches a set of new BCA (based on the AIS algorithm result) which are responsible for providing a new cure.

When a new BCA is initiated, its only responsibility is to provide a new cure that solves the malfunction. In order to do so, it analyses the received error and the genome which represents the BCA itself, if the genome is a cure, the BCA automatically replies the cure, otherwise starts a negotiation to the available CPAs (stored in the Cure Provider Cloud) to find new possible solutions for this case. At this stage, the CPAs will check the availability of cures to solve the requested error. If a satisfactory cure is found, the CPA will respond with it.

For a better insight on how the agents interact among themselves and their classes' relationships, please refer to Fig. 6. In this figure, it is possible to understand the relationship between all the entities in the system and the classes responsible for the communication amongst them. The communication in this framework respects, as said above, the FIPA protocols making it necessary that the communication responsible classes extend any of the JADE FIPA compliant classes.

As one can observe in the above figure, both the DA and the GDA contain a library of cures, errors and skills, representing the known cures, errors and skills of each DA and GDA. The latter is composed of the cures, errors and skills of every DA associated to it. Once an unforeseen error comes across a DA and it is grouped, then the GDA in which it is grouped launches the BCAs that will attempt to fix it, according to the AIS algorithm with which the system is working; if the DA is not grouped, it will itself launch the BCAs. If the BCA finds a cure, it returns it to either the GDA or the DA that launched it; on the other hand, if the BCA doesn't find any cure, it will contact the cloud where the CPAs rest in an attempt to find the cure. Once a cure request arrives to the CPA it will look in its database for the requested error, sending the cure information back to the BCA if it ever finds a cure or sending an error message on the contrary. The above depicted behaviour is supported by the class diagram presented in Fig. 7. This class diagram illustrates he Java classes, and respective relationships, used to develop the presented work. It also depicts the most important methods and variables of each of the entities.

Results

The optimal way of testing the work which motivated this paper would be in a real industrial system. Given the impossibility of such, the authors adopted a simulation environment approach that would allow to testify whether the proposed framework behaves as expected, as well as to infer whether it works better in a centralised or decentralised manner.

Simulation environments

Two simulation environments were developed in order to assess whether the architecture performed better in a decentralised or centralised manner. Both environments were equal in the following points:

Six low level entities

After some tests, it was concluded that, for both simulation environments, either de or centralised, a total of six low level entities was appropriate to populate the system with.

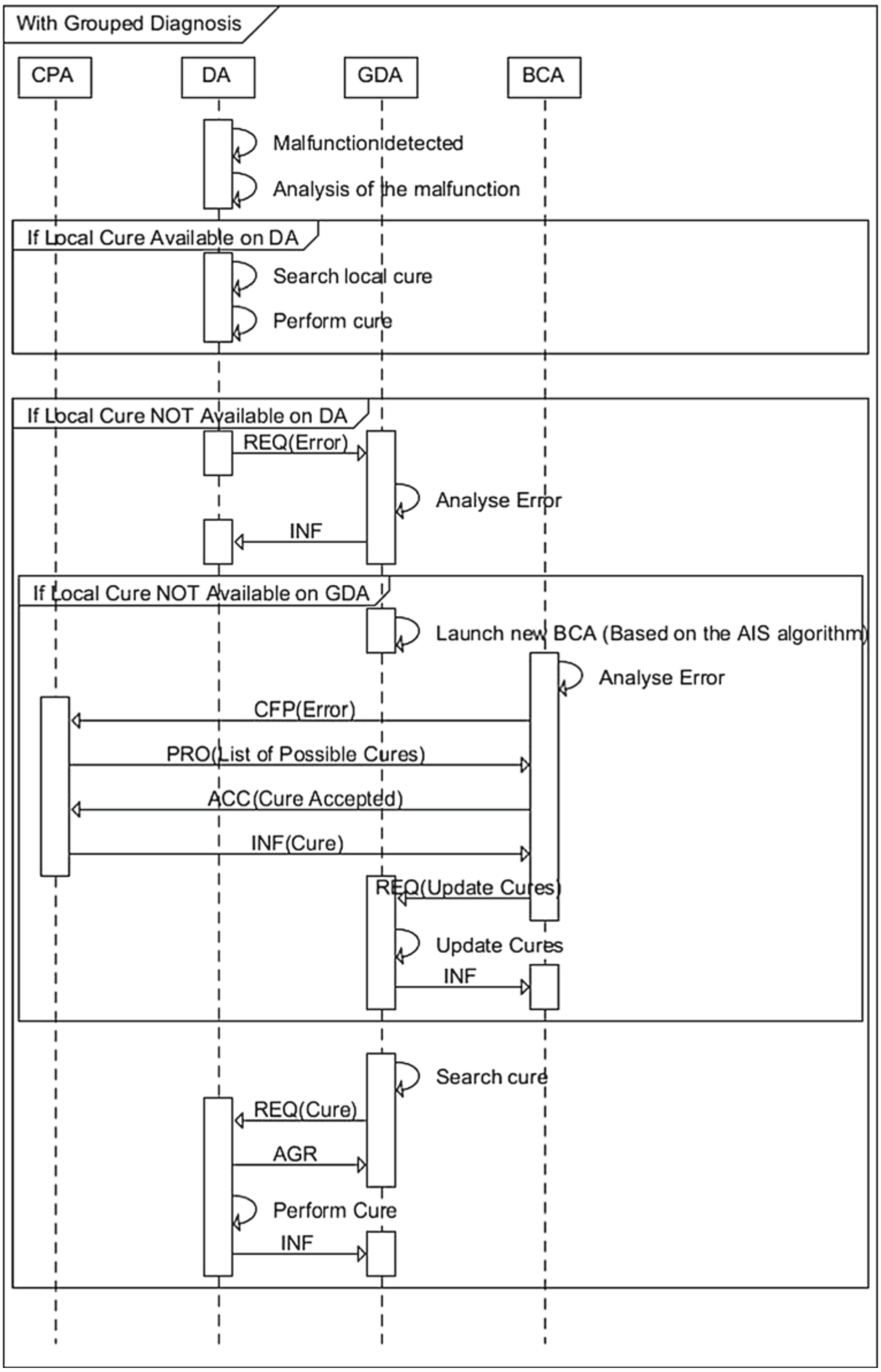

Fig. 6 Sequence diagram after a malfunction detected

One high level entity for three low level entities

In order to simulate a scenario as real as possible, it was determined that one Grouped Diagnosis Agent (GDA) for three diagnosis agents (DAs) would represent the ideal entity-level organization of this system in an industrial environment.

Despite its similarities, the environments were different in the sense that, in the decentralised one, the low-level entities

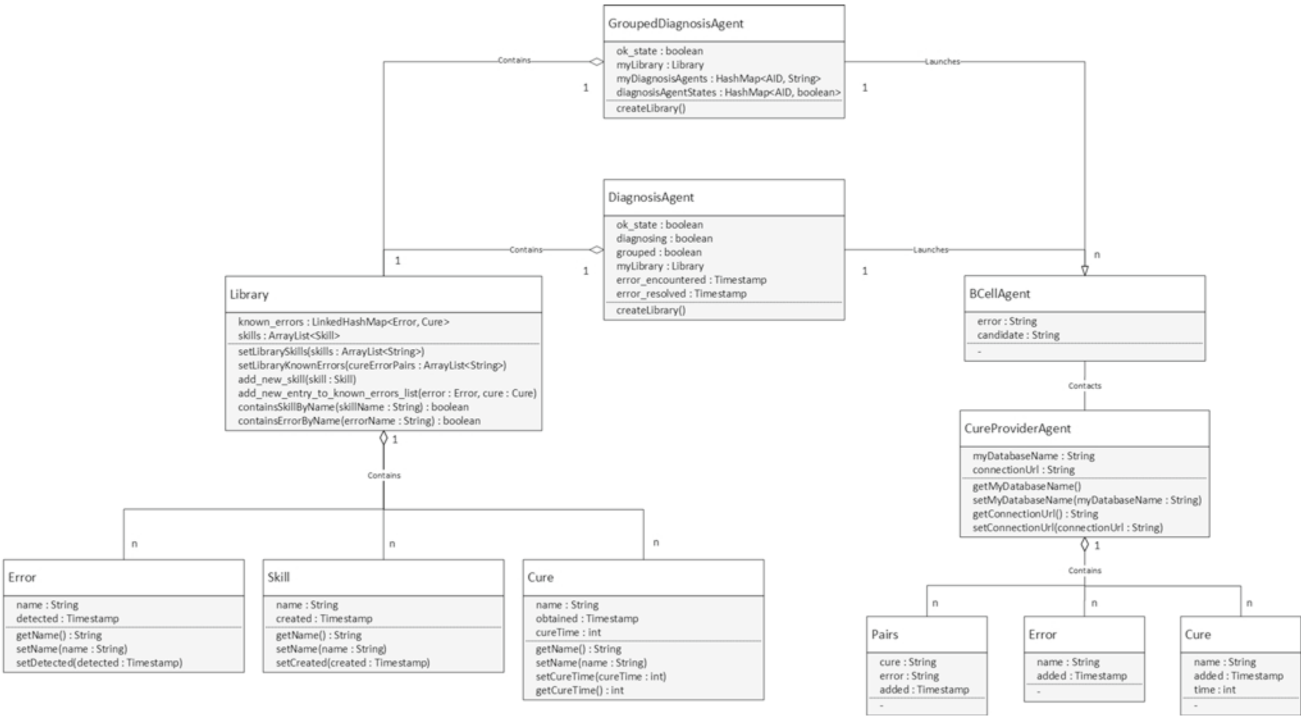

Fig. 7 System's class diagram

were distributed three ways by two independent computers whereas in the centralised, all the entities were hosted in the same computer, along with the high level entities that supervised them.

Payload tests

In this type of tests, the authors tested for the overall responsiveness of the system when its entirety was under an error scenario. To do so, the system was launched and waited until an order from a remote system was given in order to launch errors in the entirety of the system, as much of the same time, as it is possible, with the current hardware. The purpose of these tests was to infer if the system could recover from a full inactive situation and restore its previous state, in the lesser time possible. As it was said before in this document, these tests were used to determine whether the architecture behaved best in a decentralised or centralised environment. The results of said tests are presented in Fig. 8.

The above picture represents the collected data for all the tests performed in order to determine which of the case scenarios would be best fit for the developed architecture. As it was expected, the decentralised environment presents far better results than the centralised one, representing a decrease of, as the word size grows bigger, more than half of the cure finding time.

Another conclusion that can be withdrawn from Fig. 7 is that, as the word size grows bigger, so does the gap between both environments, as to what time lapse concerns. This was expected and is logical, since the bigger the word size, the harder it will be, theoretically, to find a cure for it (this is sustained by the increasing time lapses in both environments).

That being said, and considering that, in a centralised system, our processing capability is rather diminished, it is only normal that the rate at which the time lapses grow in a centralised system are immensely superior to those of a decentralised one.

Last, but not least, it is possible to conclude that this architecture is far more scalable in a decentralised environment than in a centralised one. The time lapses are smaller in the first scenario, which leads to the conclusion that, theoretically, it is possible to sustain bigger systems in a decentralised fashion rather than in a centralised one.

Hence, it is hereby proven that a decentralised environment is, indeed, and for all it matters, better for this architecture. Therefore, for the following tests to be presented it will be assumed that the environment being tested is no other but the decentralised one.

Overtime tests

These tests were used with the sole purpose of evaluating the learning capability of the system. By evaluating the cure finding times it was outputting it was possible to evaluate if

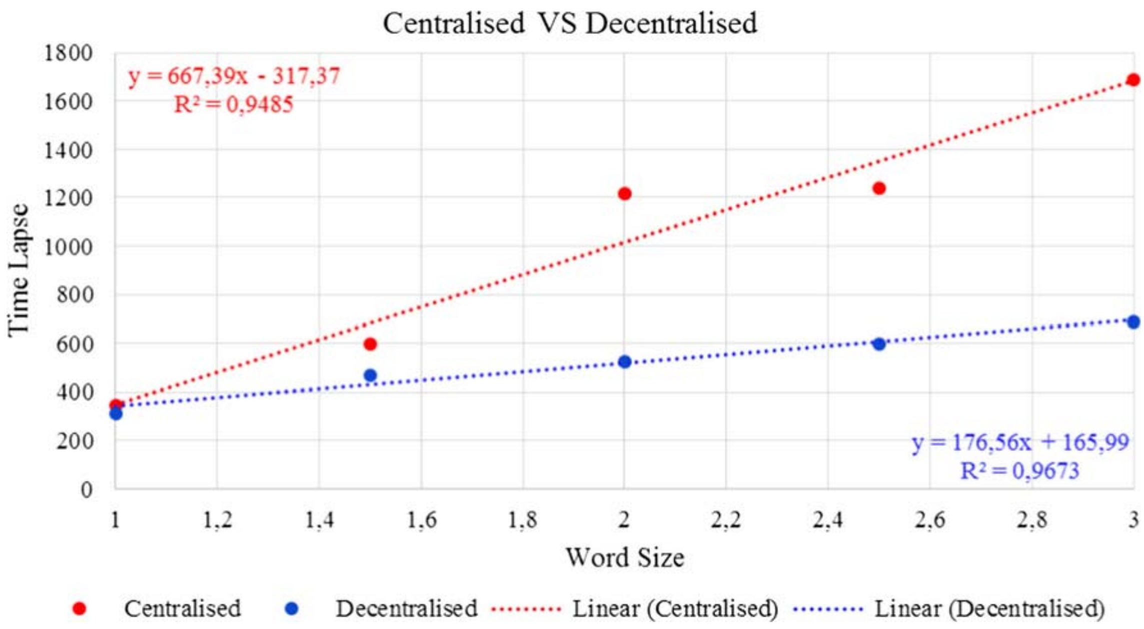

Fig. 8 Centralized versus decentralized data comparison

Fig. 9 Error average overtime

they were getting smaller, as they should, from an evolutionary perspective. This would mean the system was learning. In order to achieve this, the system behaves very similarly to the aforementioned one in the beginning, as in it waits for a communication from a remote machine so it can start its procedures, and this is where it all changes.

Instead of launching errors throughout the entirety of the system, this test launched a hardware class in each of the low-level entities in the system. This hardware class is meant to simulate an actual hardware output situation, such as those from the programmable logic controllers (PLCs), to give an example. This would, occasionally, send an error to the entity responsible for it.

That being said, these tests were performed to analyse whether the cure finding time actually diminishes with the amount of errors that have previously been found and cured or not. This is expected to occur due to the fact that, as a new cure is learnt, it will constitute, along with the previous ones, the initial population for the cure finding algorithm in a new error event. Given that this algorithm is based in the affinity between the known cure's genomes and the error's, it is only natural that, the more cure's that constitute the initial population, the bigger the chances will be of finding a cure in a shorter time.

For a graphical view on this subject, an analysis on the average time each error took to be resolved is shown in Fig. 9.

The above figure depicts a graphic that represents the average time lapse obtained from all the DAs for each consecutive error. If one is to take a closer look it becomes easily observable that the time lapses do slightly diminish as new cures enter the system once the error is resolved.

The sole exception to this trend is the last error, where the average time goes up. It so happens because, once again, the last DA means the return of the system to its prior state, where no BCAs are present in the system. This means that a lot of queued messages need to be processed which ends up delaying the cure finding time of the last DA.

Despite the last value does not follow the overall tendency, this faulty behaviour, if you may, may be caused due to the accumulated messages in the JADE scheduler or due to the JAVA garbage collector.

That being said, and with all the presented data, it is fair to conclude that this system possesses a learning mechanism, corroborated by the above data, and which results in a faster cure finding by the system, as long as there have been previous errors detected.

Conclusion

The main purpose of this document was to offer an alternative to the current diagnosis paradigms through the means of the artificial immune systems (AIS). This was quite a challenge since this is still a much undeveloped topic. That being said, there are three principal conclusions that can be drawn from the work presented in this document.

Firstly, the work developed around the AIS algorithms allowed for some sort of ranking between them to be established which, as much as this document's author concerns, was yet to be drafted. With this ranking, it was possible to conclude on which of the algorithms, negative selection (NS), clonal selection (CS) or network model (NM), would fit better in an industrial environment, with all the processing (dis)abilities to it associated. The NM proved to be the most efficient of the algorithms for the tested word sizes, accounting for an improvement of around forty pp in the biggest word sizes. However, the AIS is based on the launching of several B Cell like entities which are, if you may, a performance killer.

Secondly, this document also shows that AIS is a viable, promising alternative to the current diagnosis paradigms. The developed architecture does not aim to be the best, but it presents itself as a valid alternative to the, if you may, classic diagnosis architectures.

It proves that, with no more than four distinct entities, it is possible to somewhat mimic the human immune system. Once an error, which is the representative of an infection, is detected, the system triggers a cure finding mechanism that launches countless B Cells alike entities (much like our own immune system does). These cells' genome will be compared to the error's (both represented by a binary string) in as much of a resemblance as possible with the Human Immune System.

Not only that, it has proven to be a somewhat rather efficient methodology when it comes to error recovery since it does not take that much to actually find a cure, as the tests have shown, since the average is around the five hundred milliseconds, which represents five times the average PLC cycle.

Moreover, it may serve as an argument in favour of those who sustain that the human system is possible to be mimicked and represented by mathematical equations, making it possible to adapt natural behaviours to unnatural proceedings, as it is the industrial scenario.

Lastly, but not least, the work hereby developed also constitutes proof that, in yet another paradigm, a distributed, decentralised system presents better results than a centralised one. This comes to show that, in a cloud like system, it is, theoretically, possible, to withstand a ridiculous amount of processing in order to achieve better and/or faster results for the case of study, which is, in this case, diagnosis.

With only two separate machines, which means the tested network was divided into two separate, independent machines, it was possible to halve the time it took for the system to find a cure. These comes to show that some further work should be made in this area, in order to further sustain these claims and, perhaps, even enhance them.

From a further work point of view, future research on other fields of diagnosis can lead to a comparison of results between this work and others that aim at the same goal. An example of such research is presented in Ghasemi et al. (2016) where the Colonial Competitive Differential Evolution Algorithm is used to solve the problem of economic load dispatch.

To wrap things up, the AIS paradigm shows great potential, given that the basis in which it was built upon have proven its usefulness – The Human Immune System; and it should definitely go under a more thorough investigation by the manufactory paradigms investigators.

Acknowledgements Funding was provided by H2020 Industrial Leadership (BE).

References

Barata, J., Ribeiro, L., & Colombo, A. (2007a). Diagnosis using service oriented architectures (SOA). In *2007 5th IEEE international conference on industrial informatics. Presented at the 2007 5th IEEE international conference on industrial informatics* (pp. 1203–1208). https://doi.org/10.1109/INDIN.2007.4384902

Barata, J., Ribeiro, L., & Onori, M. (2007b). Diagnosis on evolvable production systems. In *IEEE international symposium on industrial electronics, 2007. ISIE 2007. Presented at the IEEE international symposium on industrial electronics*, 2007. ISIE 2007 (pp. 3221–3226). https://doi.org/10.1109/ISIE.2007.4375131

Barata, J., & Camarinha-Matos, L. M. (2003). Coalitions of manufacturing components for shop floor agility-the CoBASA architecture. *International Journal of Networking and Virtual Organisations*, *2*, 50–77. https://doi.org/10.1504/IJNVO.2003.003518.

El-Sharkh, M. Y. (2014). Clonal selection algorithm for power generators maintenance scheduling. *International Journal of Electrical Power & Energy Systems*, *57*, 73–78. https://doi.org/10.1016/j.ijepes.2013.11.051.

Frei, R., Barata, J., & Onori, M. (2007). Evolvable production systems context and implications. In *IEEE international symposium on industrial electronics, 2007. ISIE 2007. Presented at the IEEE international symposium on industrial electronics*, 2007. ISIE 2007 (pp. 3233–3238). https://doi.org/10.1109/ISIE.2007.4375132

Ghasemi, M., Taghizadeh, M., Ghavidel, S., & Abbasian, A. (2016). Colonial competitive differential evolution: An experimental study for optimal economic load dispatch. *Applied Soft Computing*, *40*, 342–363.

Karagöz, S., & Yıldız, A. R. (2017). A comparison of recent metaheuristic algorithms for crashworthiness optimisation of vehicle thin-walled tubes considering sheet metal forming effects. *International Journal of Vehicle Design*, *73*, 179–188.

Kiani, M., & Yildiz, A. R. (2016). A comparative study of non-traditional methods for vehicle crashworthiness and NVH optimization. *Archives of Computational Methods in Engineering*, *23*, 723–734.

Leitão, P., & Restivo, F. (2008). A holonic approach to dynamic manufacturing scheduling. Robotics and Computer-Integrated Manufacturing. In *BASYS'06: Balanced automation systems 7th IFIP international conference on information technology for balanced automation systems in manufacturing and services* (Vol. 24, pp. 625–634). https://doi.org/10.1016/j.rcim.2007.09.005.

Leitão, P., & Restivo, F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, *57*, 121–130. https://doi.org/10.1016/j.compind.2005.05.005.

Lohse, N., Ratchev, S., & Barata, J. (2006). Evolvable assembly systems: On the role of design frameworks and supporting ontologies. In *2006 IEEE international symposium on industrial electronics. Presented at the 2006 IEEE international symposium on industrial electronics* (pp. 3375–3380). https://doi.org/10.1109/ISIE.2006.296008

Öztürk, N., Yıldız, A. R., Kaya, N., & Öztürk, F. (2006). Neuro-genetic design optimization framework to support the integrated robust design optimization process in CE. *Concurrent Engineering*, *14*, 5–16.

Peixoto, J. A., Oliveira, J. A. B., Rocha, A. D., & Pereira, C. E. (2015). The migration from conventional manufacturing systems for multi-agent paradigm: The first step. In L. M. Camarinha-Matos, T. A. Baldissera, G. D. Orio, & F. Marques (Eds.), *Technological innovation for cloud-based engineering systems, IFIP advances in information and communication technology* (pp. 111–118). Berlin: Springer. https://doi.org/10.1007/978-3-319-16766-4_12.

Ribeiro, L., & Barata, J. (2012). IMS 10–Validation of a co-evolving diagnostic algorithm for evolvable production systems. *Engineering Applications of Artificial Intelligence*, *25*, 1142–1160. https://doi.org/10.1016/j.engappai.2012.02.008.

Shen, W., & Norrie, D. H. (2013). Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems*, *1*, 129–156. https://doi.org/10.1007/BF03325096.

Storey, J. (1994). *New wave manufacturing strategies: Organizational and human resource management dimensions*. Thousand Oaks: SAGE.

Tetzlaff, D. U. A. W. (1990). Flexible manufacturing systems. In *Optimal design of flexible manufacturing systems, contributions to management science. Physica-Verlag HD* (pp. 5–11). https://doi.org/10.1007/978-3-642-50317-7_2

Yildiz, A. R. (2009). Hybrid immune-simulated annealing algorithm for optimal design and manufacturing. *International Journal of Materials and Product Technology*, *34*, 217–226.

Yıldız, A. R. (2009a). An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry. *Journal of Materials Processing Technology*, *209*, 2773–2780.

Yıldız, A. R. (2009b). A novel particle swarm optimization approach for product design and manufacturing. *The International Journal of Advanced Manufacturing Technology*, *40*, 617–628.

Yildiz, A. R. (2013a). A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing. *Applied Soft Computing*, *13*, 2906–2912.

Yildiz, A. R. (2013b). Comparison of evolutionary-based optimization algorithms for structural design optimization. *Engineering Applications of Artificial Intelligence*, *26*, 327–333.

Yıldız, B. S. (2017). A comparative investigation of eight recent population-based optimisation algorithms for mechanical and structural design problems. *International Journal of Vehicle Design*, *73*, 208–218.

Yıldız, B. S., & Lekesiz, H. (2017). Fatigue-based structural optimisation of vehicle components. *International Journal of Vehicle Design*, *73*, 54–62.

Yildiz, B. S., Lekesiz, H., & Yildiz, A. R. (2016). Structural design of vehicle components using gravitational search and charged system search algorithms. *Materials Testing*, *58*, 79–81.

Yıldız, A. R., Öztürk, N., Kaya, N., & Öztürk, F. (2007). Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm. *Structural and Multidisciplinary Optimization*, *34*, 317–332.

Yildiz, A. R., & Solanki, K. N. (2012). Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach. *The International Journal of Advanced Manufacturing Technology*, *59*, 367–376.

Yin, M., Zhang, T., & Shu, Y. (2012). An artificial immune model with danger theory based on changes. In *2012 international conference on computer science service system (CSSS). Presented at the 2012 international conference on computer science service system (CSSS)* (pp. 672–676). https://doi.org/10.1109/CSSS.2012.174

Zhong, Y., & Zhang, L. (2012). An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *50*, 894–909. https://doi.org/10.1109/TGRS.2011.2162589.

Zuccolotto, M., Pereira, C.E., Fasanotti, L., Cavalieri, S., & Lee, J. (2015). Designing an artificial immune systems for intelligent maintenance systems. *IFAC-Pap., 15th IFAC symposium on information control problems in manufacturing INCOM 2015* (Vol. 48, pp. 1451–1456). https://doi.org/10.1016/j.ifacol.2015.06.291