Accelerating Improvement of Fuzzy Rules Induction with Artificial Immune Systems

EDWARD MĘŻYK, OLGIERD UNOLD
Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
Wyb. Wyspianskiego 27, 50-370 Wroclaw
POLAND
{edward.mezyk, olgierd.unold }@pwr.wroc.pl

Abstract: - The paper introduces an algorithmic improvement to IFRAIS, an existing Artificial Immune System method for fuzzy rule mining. The improvement presented consists of using rule buffering during the computation of fitness of rules. This is achieved using a hash table. The improved method has been tested against two different fitness functions and various data sets. Experimental results show improvements in computing times in the order of 3 to 10 times maintaining same levels of accuracy.

Key-Words: - Artificial Immune System, Data Mining, Fuzzy Rules Induction

1 Introduction

Data mining tasks are often categorized into types of tasks they are applied to. One of them is a classification task, whose aim is to find general features of objects in order to predict classes they are associated with. Quite novel approaches, among others, integrate Artificial Immune System (AIS) [3] and Fuzzy Systems (FS) [6] to find not only accurate, but also comprehensible fuzzy rules that predict the class of an example. This kind of algorithms discovers a set of rules of the form "IF (fuzzy conditions) THEN (class)", whose interpretation is as follows: IF an example's attribute values satisfy the fuzzy conditions THEN the example belongs to the class predicted by the rule. One of the AIS-based algorithms for mining IF-THEN rules is based on extending the negative selection algorithm with a genetic algorithm [4]. Another one is mainly focused on the clonal selection and so-called a boosting mechanism to adapt the distribution of training instances in iterations [1]. A fuzzy AIS was proposed also in [5], however that work addresses not the task of classification, but the task of clustering.

The paper introduces a speed boosting extension to IFRAIS, the first AIS-based method for fuzzy rules mining [2].

2 IFRAIS

Data preparation for learning in IFRAIS consists of the following steps: (1) create for each attribute in data set fuzzy variable; (2) create class list for actual data set; (3) and compute information gain for each attribute in data set.

```
Input: full training set
Output: fuzzy rules set

rules set=0
FOR EACH class c value in class values list DO
  values count = number of c in full training set
  training set = full training set
  WHILE values count > number of maximal uncovered
  examples AND values count >percent of maximal
  uncovered examples
     rule = CLONAL SELECTION ALGORITHM(training
     set, c)
     covered = COVER SET(training set, rule)
     training set=training set / covered with rule
     set
     values count = values count - size of covered
     set
     ADD(rules set, rule)
  END WHILE
END FOR EACH
FOR EACH rule R in rules set DO
  COMPUTE FITNESS(R, full training set)
END FOR EACH
RETURN rules set
```

Fig. 1. Sequential covering algorithm in IFRAIS (based on [2])

IFRAIS uses a sequential covering as a main learning algorithm (see Fig. 1). In the first step a set of rules is initialized as an empty set. Next, for each class to be predicted the algorithm initializes the training set with all training examples and iteratively calls clonal selection procedure with the parameters: the current training set and the class to be predicted. The clonal selection procedure returns a discovered rule and next the learning algorithm adds the rule to the rule set and removes from the current training set the examples that have been correctly covered by the evolved rule.

Clonal selection algorithm is used to induct rule with best fitness from training set (see Fig. 2). Basic elements of this method are antigens and antibodies which refers directly to biological immune systems. Antigen is an example from data set and antibody is a fuzzy rule. Similarly to fuzzy rule structure, which consists of fuzzy conditions and class value, antibody comprises genes and informational gene. Number of genes in antibody is equal to number of attributes in data set. Each gene consists of a fuzzy rule and an activation flag that indicates whether fuzzy condition is active or inactive

```
Input: training set, class value c
Output: fuzzy rule

Create randomly antibodies population with size s
and class value c
FOR EACH antibody A in antibodies population
   PRUNE(A)
   COMPUTE FITNESS(A, training set)
END FOR EACH
FOR i=1 do generation size n
   WHILE clones population size < s-1
        antibody to clone = TOURNAMENT
        SELECTION(antibodies population)
        clones = CREATE x CLONES(antibody to clone)
        clones population = clones population +
        clones
   END WHILE
   FOR EACH clone K in clones population
        muteRatio = MUTATION PROBABILITY(K)
        MUTATE(K, muteRatio)
        PRUNE(K)
        COMPUTE FITNESS(K, training set)
   END FOR EACH
   antibodies population = SUCCESSION(antibodies
   population, clones population)
END FOR
result = BEST ANTIBODY(antibodies population)
RETURN result
```

Fig. 2. Clonal selection algorithm in IFRAIS [2]

In the first step the algorithm generates randomly antibodies population with informational gene equal to class value *c* passed in algorithm parameter. Next each antibody from generated population is pruned. Rule pruning has a twofold motivation: reducing the overfitting of the rules to the data and improving the simplicity (comprehensibility) of the rules [7]. Fitness of the rule is computed according to the formula (called Alves fitness function)

$$fitness = \frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP} \tag{1}$$

where *TP* is number of examples satisfying the rule and having the same class as predicted by the rule; *FN* is the number of examples that do not satisfy the rule but have the class predicted by the rule; *TN* is the number of examples that do not satisfy the rule and do not have the class predicted by the rule; and *FP* is the number of examples that satisfy the rule but do not have the class predicted by the rule.

Hence, the rules are fuzzy, the computation of the *TP*, *FN*, *TN* and *FP* involves measuring the degree of affinity between the example and the rule. This is computed by applying the standard aggregation fuzzy operator *min.* An example satisfies a rule if *AFFINITY(rule, example)* > *L*, where *L* is an activation threshold.

For each antibody to be cloned the algorithm produces *x* clones. The value of *x* is proportional to the fitness of the antibody. Next, each of the clones undergoes a process of hypermutation, where the mutation rate is inversely proportional to the clone's fitness. Once a clone has undergone hypermutation, its corresponding rule antecedent is pruned by using the previously explained rule pruning procedure. Finally, the fitness of the clone is recomputed, using the current training set. In the last step the *T*-worst fitness antibodies in the current population are replaced by the *T* best-fitness clones out of all clones produced by the clonal selection procedure. Finally, the clonal selection procedure returns the best evolved rule, which will then be added to the set of discovered rules by the sequential covering. More details of the IFRAIS is to be found in [2].

In order to extend analysis of proposed improvement, IFRAIS with another fitness function was tested. A new, more complex function is based on fitness computation formula proposed in [1], and it is given with following equation (called Alatas fitness function)

$$fitness = w_1 \cdot Q_1 + w_2 \cdot Q_2, \tag{2}$$

where Q_1 is given with equation

$$Q_1 = \frac{sensitivity \times specificity + aw \times accuracy'}{1 + aw}, \tag{3}$$

where

$$sensitivity = \frac{TP}{TP + FN}, \tag{4}$$

$$specificity = \frac{TN}{TN + FP}, \tag{5}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

$$accuracy' = \begin{cases} accuracy & when\ accuracy > 0{,}7 \\ 0 & when\ accuracy \leq 0{,}7 \end{cases}, \tag{7}$$

and Q_2 is given with formula

$$Q_2 = 1 - \frac{ActiveConditionsCount}{20} \tag{8}$$

where *aw* is the weight of the *accuracy* and is set to 0.01. Coefficients w_1 and w_2 are weights in turn for Q_1 and Q_2 ,and were set to 1 and 0,0005 respectively. *ActiveConditionsCount* is a number of active conditions in a computed rule.

3 Improved IFRAIS

Applying the IFRAIS to real data mining problems is time-consuming process. To decrease training time and at the same time not to influence the quality of learning rules some improvements were introduced to the algorithm. The main idea is simply and relies on the rules buffering while their fitness is computed in clonal selection algorithm (Fig. 3). In a hash table the pairs: rule and its fitness to actual training set are saved. If the hash table contains given rule, then fitness associated with this rule is returned as a result and an exhaustive computation of fitness is omitted. In other case rule fitness computation is executed and next rule with its fitness is saved in the hash table and computed value is returned as a function output.

```
Input: training set, class value c
Output: fuzzy rule

Empty hash table BUFFER initiation
Randomly create antibodies population of size s and
class value c
FOR EACH antibody A in antibodies population
```

```
    PRUNE(A)
    IF CONTAINS(BUFFER, A)
       RETURN ASSOCIATED ELEMENT(BUFFER, A)
    ELSE
       fitness = COMPUTE FITNESS(A, training set)
       ADD(BUFFER, A, fitness)
END FOR EACH
FOR i=1 do generation count n
    WHILE clones populations size < s-1
       antibody to clone = TOURNAMENT
       SELECTION(antibody population)
       clones = CREATE x CLONES(antibody to clone)
       clones population = clones population +
       clones
    END WHILE
    FOR EACH clone K in clones population
       muteRatio = MUTATION PROBABILITY(K)
       MUTATE(K, muteRatio)
       PRUNE(K)
       IF CONTAINS(BUFFER, K)
          RETURN ASSOCIATED ELEMENT(BUFFER, K)
       ELSE
          fitness = COMPUTE FITNESS(K, training
          set)
          ADD(BUFFER, A, fitness)
    END FOR EACH
    Antibody population = SUCCESSION(antibody
    population, clones population)
END FOR
result = BEST ANTIBODY(antibodies population)
RETURN result
```

Fig. 3. Clonal selection algorithm in an improved IFRAIS

4 Experimental results

In order to evaluate the performance of the speed boosting extensions, IFRAIS with Alves function, improved IFRAIS with Alves function, IFRAIS with Alatas function, and improved IFRAIS with Alatas function were applied to 6 public domain data sets available from the UCI repository (http://archive.ics.uci.edu/ml/datasets.html). UCI data sets are widely used in other works, which aimed at classification problems (i.e. [11]) .The experiments were conducted using a Distribution-Balanced Stratified Cross-Validation [8], which is a one of the version of well-known *k*-fold cross-validation, and improves the estimation quality by providing balanced intraclass distributions when partitioning a data set into multiple folds.

Table 1. Data sets and number of rows, attributes, continuous attributes, classes and full UCI repository data set name.

Data set	# Rows	# Attrib.	# Cont.	# Class.	Full UCI name
Bupa	345	6	6	2	Liver Disorders
Crx	653	15	6	2	Credit Approval
Hepatitis	80	19	6	2	Hepatitis
Ljubljana	277	9	9	2	Breast Cancer
Wisconsin	683	9	9	2	Congressional Voting Records
Votes	232	16	0	2	Breast Cancer Wisconsin (Original)

Table 1 shows the number of rows, attributes, continuous attributes, and classes for each data set. Note that only continuous attributes are fuzzified. The *Votes* data set does not have any continuous attribute to be fuzzified, whereas the other data sets have 6 or 9 continuous attributes that are fuzzified by IFRAIS.

All experiments were repeated 50-times using 5-fold cross-validation. Table 2 and Table 3 show for each data set the average time of working and the average accuracy rate, both with standard deviations, for IFRAIS and improved IFRAIS. Table 2 shows results for Alves fitness function, whereas table 3 results for Alatas fitness function. As shown in both Table 2 and Table 3 the improved IFRAIS obtained comparable results as the standard version, but in considerably better time. For example, the time to achieve the same accuracy rate for *Bupa* set (ca 58.3 %) is almost tenfold (!) less for the improved IFRAIS (0.71±0.02 s) than the standard one (6.83±0.17 s). For the other data sets the time needed for learning for speed boosting IFRAIS is several times less.

Table 2. Time and accuracy rate on the test set for Alves fitness function.

Data set	IFRAIS		Improved IFRAIS	
	Accuracy	Time [s]	Accuracy	Time [s]
Bupa	58,23±0,81	6,83±0,17	58,38±0,61	0,71±0,02
Crx	86,09±0,15	10,46±0,31	86,13±0,17	1,82±0,07
Hepatitis	77,35±1,69	1,04±0,03	77,55±1,58	0,43±0,02
Ljubljana	69,47±1,26	5,47±0,05	69,80±1,14	0,73±0,02
Votes	96,98±0,00	1,45±0,01	96,98±0,00	0,34±0,01
Wisconsin	95,06±0,33	10,13±0,28	94,87±0,37	1,71±0,06

Table 3. Time and accuracy rate on the test set for Alatas fitness function.

Data set	IFRAIS		Improved IFRAIS	
	Accuracy	Time [s]	Accuracy	Time [s]
Bupa	58,56±0,79	6,89±0,12	58,43±0,68	0,72±0,02
Crx	86,06±0,17	10,37±0,25	86,11±0,14	1,81±0,05
Hepatitis	78,20±1,84	1,02±0,03	77,95±2,07	0,42±0,02
Ljubljana	69,54±1,12	5,29±0,06	69,27±1,12	0,72±0,01
Votes	96,98±0,00	1,47±0,01	96,98±0,00	0,34±0,00
Wisconsin	94,89±0,35	9,81±0,25	94,90±0,33	1,67±0,06

For each experimental data set eight charts are provided, four pairs for each version of IFRAIS (Fig. 4-51). First chart from pair presents mean accuracy, maximal accuracy and minimal accuracy of 5-fold cross-validation in each of 50 iterations of experiment. Second chart in pair shows mean time, maximal time and minimal time spent on inducting full rules sets in 5-fold cross-validation.

As shown on accuracy charts, classification effectiveness is very unstable. 5-fold Distribution-Balanced Stratified Cross-Validation gives each time the same folds, so this diversity is not a result of different training sets in each test iterations. Probably accuracy instability is an effect of too short rule induction process, weak resistance to local extremes or both. This issue is a subject for further research.

Presented charts also shows that accuracy mean is on the same level for each of tested IFRAIS modifications, what is a proof that, both buffering and new fitness function do not affect an accuracy. It is a strong proof for a correctness of presented improvement.

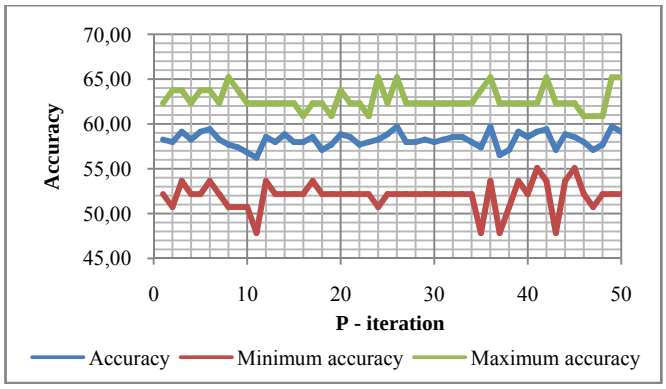

Fig. 4. Accuracy chart based on IFRAIS with Alves fitness function and *Bupa* data set.

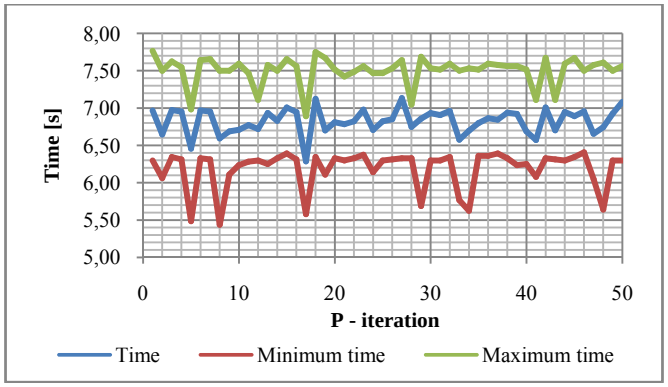

Fig. 5. Time chart based on IFRAIS with Alves fitness function and *Bupa* data set.

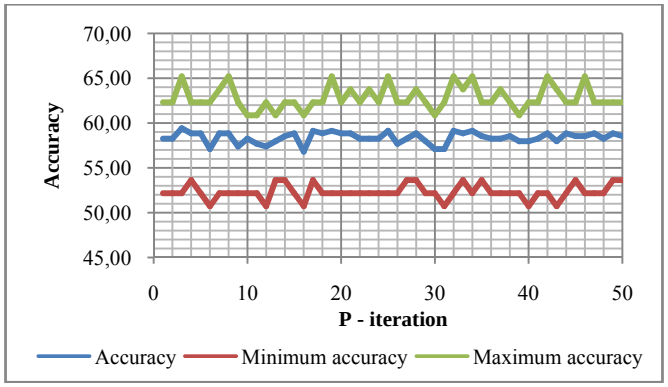

Fig. 6. Accuracy chart based on improved IFRAIS with Alves fitness function and *Bupa* data set.

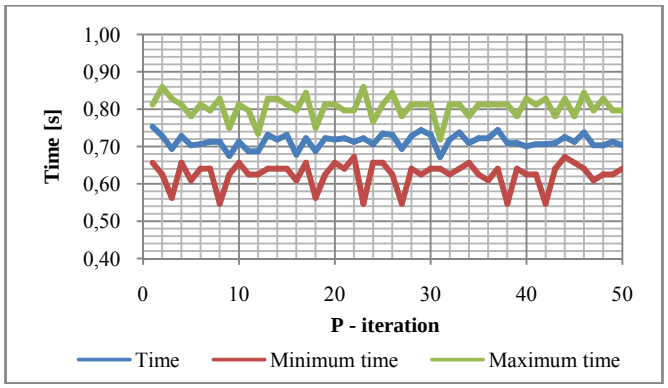

Fig. 7. Time chart based on improved IFRAIS with Alves fitness function and *Bupa* data set.

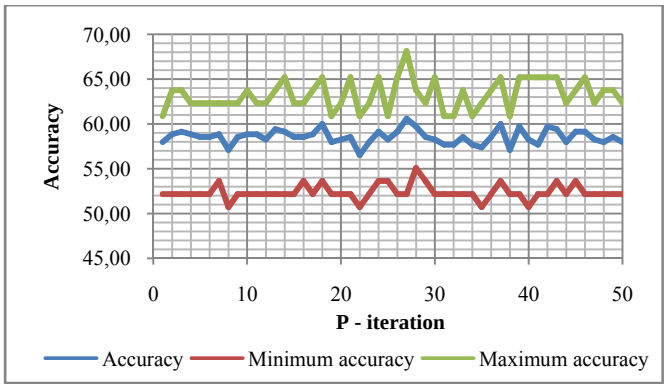

Fig. 8. Accuracy chart based on IFRAIS with Alatas fitness function and *Bupa* data set.

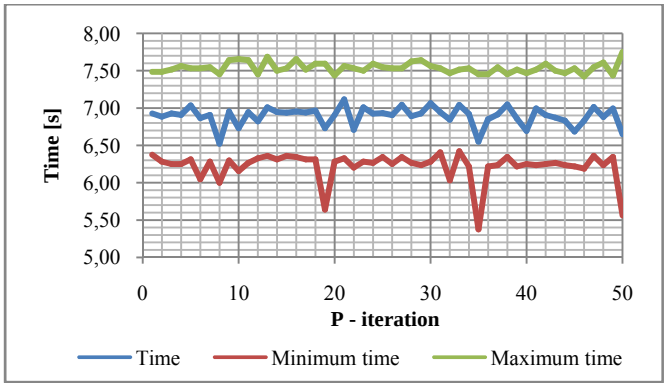

Fig. 9. Time chart based on IFRAIS with Alatas fitness function and *Bupa* data set.

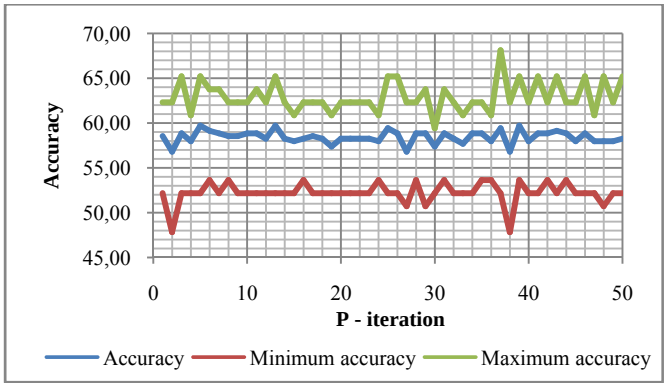

Fig. 10. Accuracy chart based on improved IFRAIS with Alatas fitness function and *Bupa* data set.

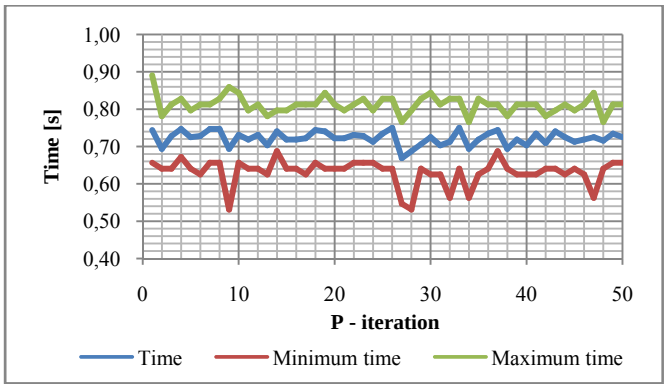

Fig. 11. Time chart based on improved IFRAIS with Alatas fitness function and *Bupa* data set.

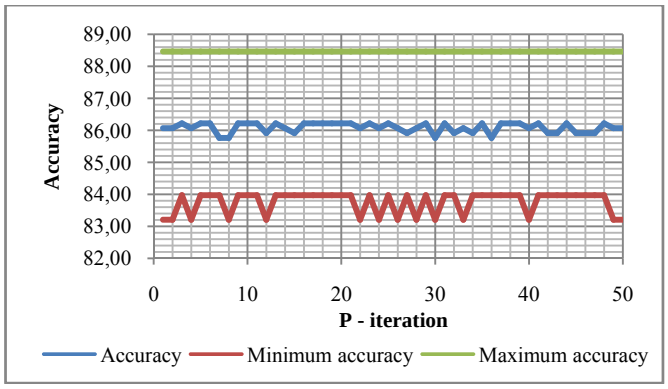

Fig. 12. Accuracy chart based on IFRAIS with Alves fitness function and *CRX* data set.

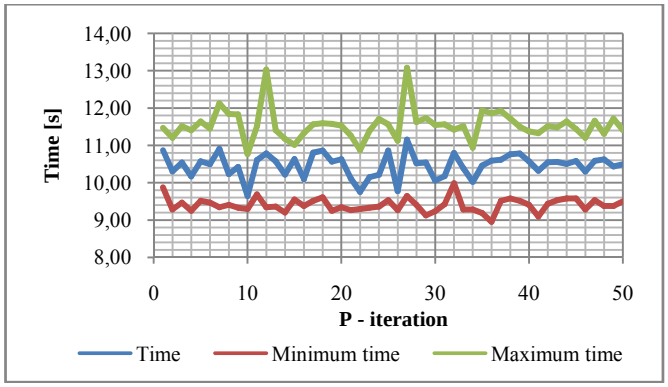

Fig. 13. Time chart based on IFRAIS with Alves fitness function and *CRX* data set.

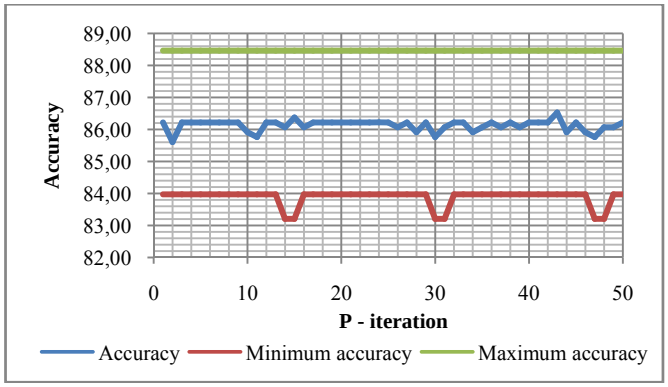

Fig. 14. Accuracy chart based on improved IFRAIS with Alves fitness function and *CRX* data set.

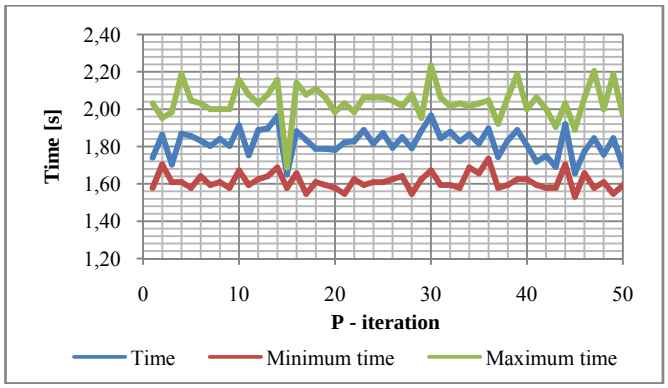

Fig. 15. Time chart based on improved IFRAIS with Alves fitness function and *CRX* data set.

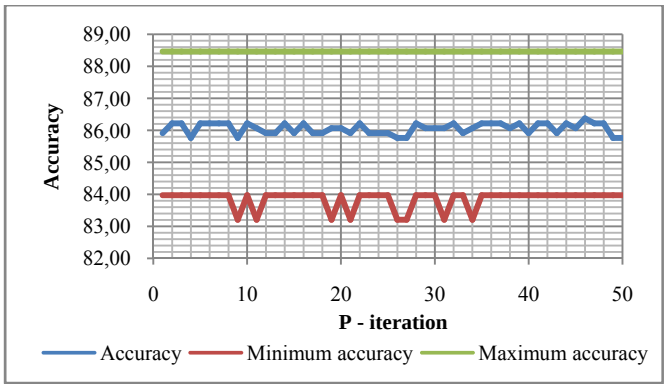

Fig. 16. Accuracy chart based on IFRAIS with Alatas fitness function and *CRX* data set.

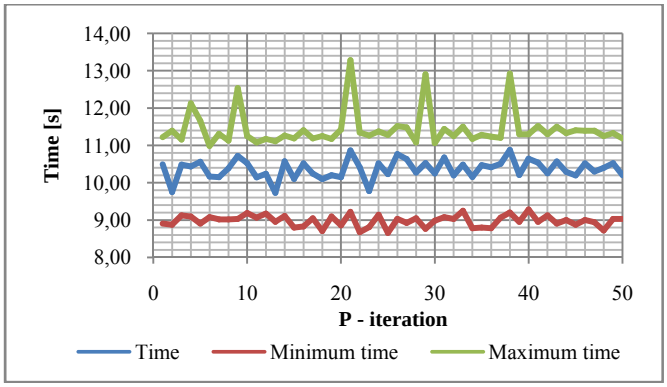

Fig. 17. Time chart based on IFRAIS with Alatas fitness function and *CRX* data set.

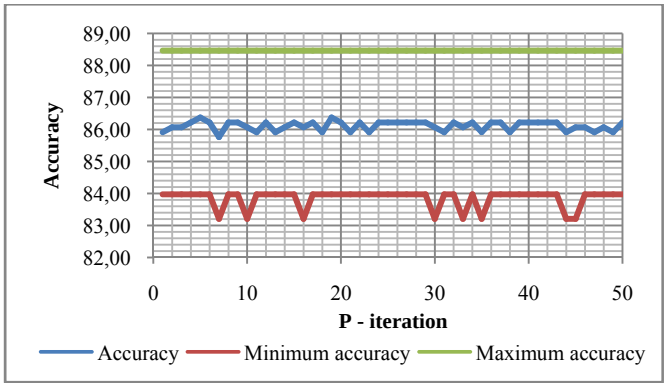

Fig. 18. Accuracy chart based on improved IFRAIS with Alatas fitness function and *CRX* data set.

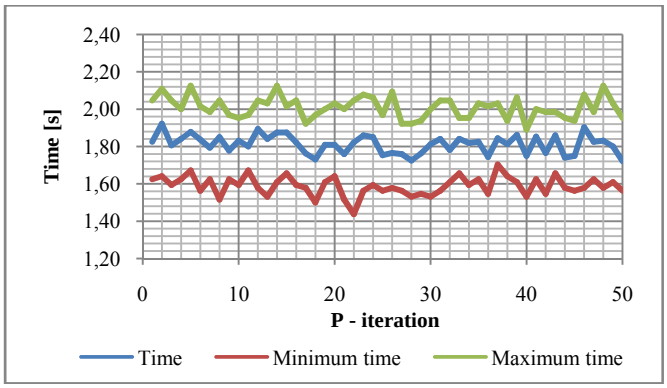

Fig. 19. Time chart based on improved IFRAIS with Alatas fitness function and *CRX* data set.

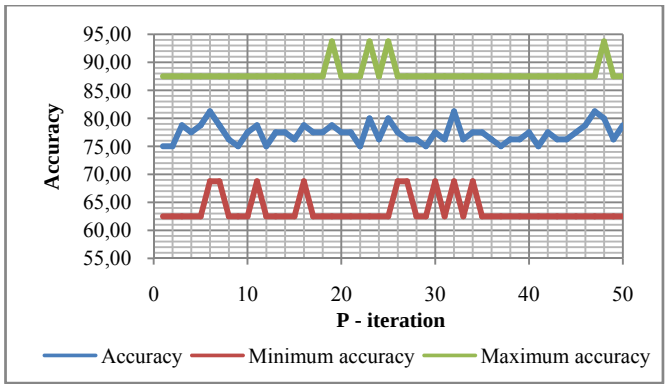

Fig. 20. Accuracy chart based on IFRAIS with Alves fitness function and *Hepatitis* data set.

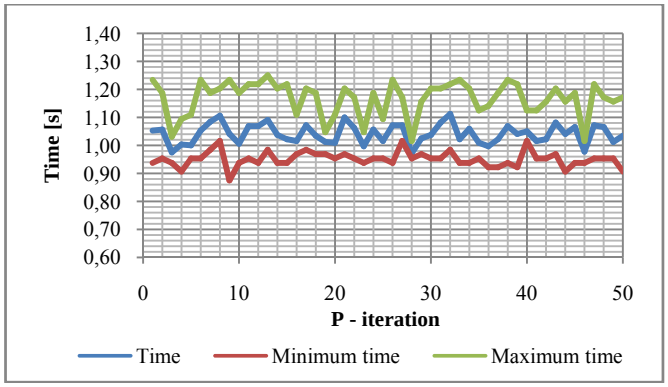

Fig. 21. Time chart based on IFRAIS with Alves fitness function and *Hepatitis* data set.

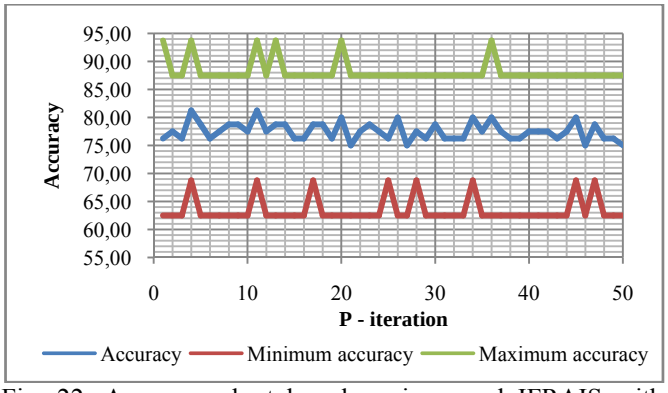

Fig. 22. Accuracy chart based on improved IFRAIS with Alves fitness function and *Hepatitis* data set.

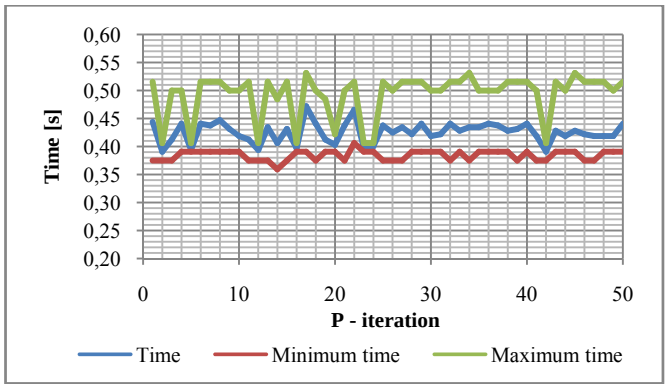

Fig. 23. Time chart based on improved IFRAIS with Alves fitness function and *Hepatitis* data set.

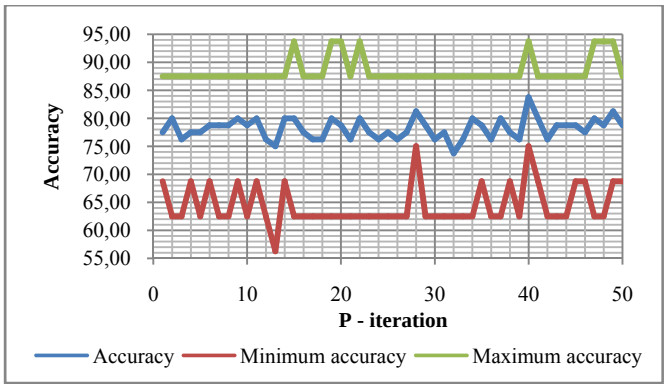

Fig. 24. Accuracy chart based on IFRAIS with Alatas fitness function and *Hepatitis* data set.

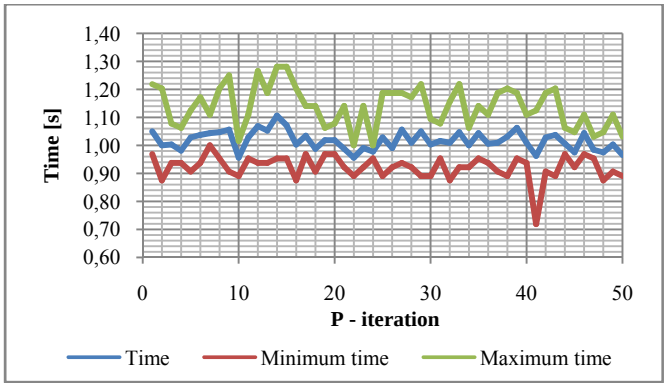

Fig. 25. Time chart based on IFRAIS with Alatas fitness function and *Hepatitis* data set.

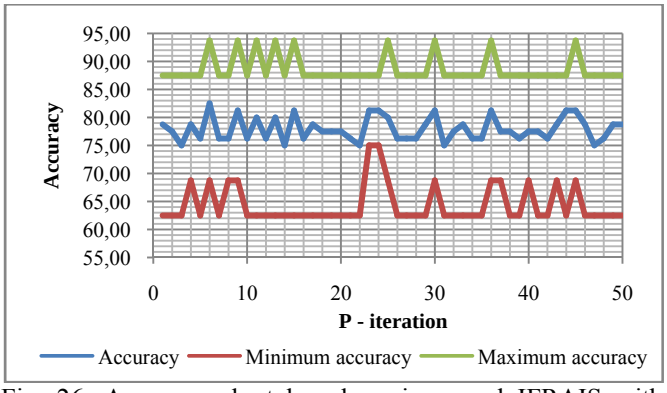

Fig. 26. Accuracy chart based on improved IFRAIS with Alatas fitness function and *Hepatitis* data set.

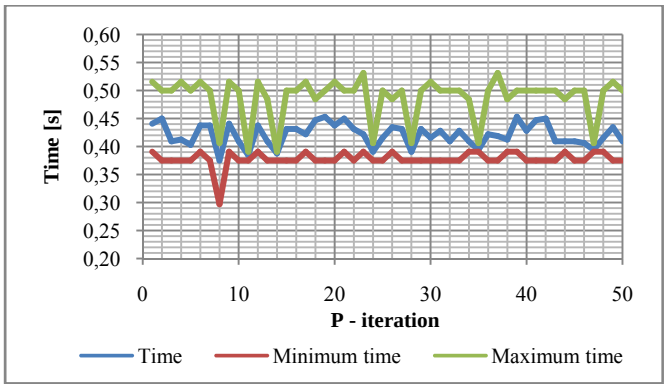

Fig. 27. Time chart based on improved IFRAIS with Alatas fitness function and *Hepatitis* data set.

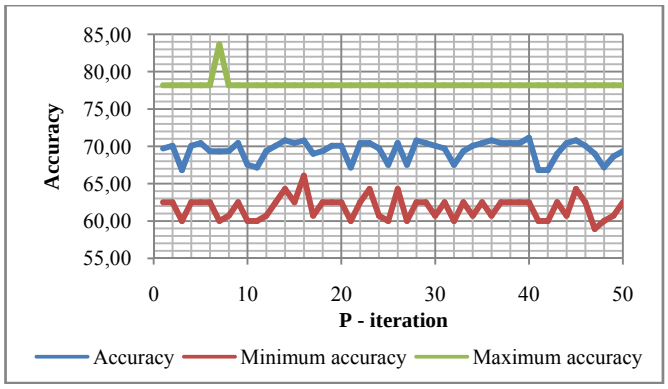

Fig. 28. Accuracy chart based on IFRAIS with Alves fitness function and *Ljubljana* data set.

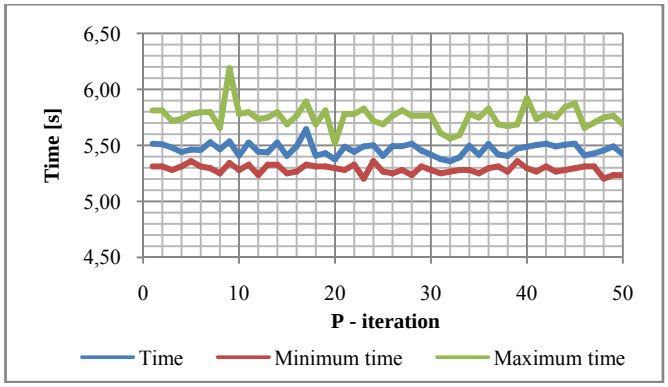

Fig. 29. Time chart based on IFRAIS with Alves fitness function and *Ljubljana* data set.

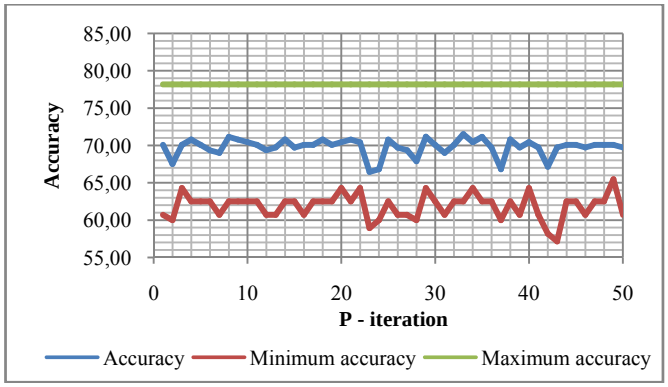

Fig. 30. Accuracy chart based on improved IFRAIS with Alves fitness function and *Ljubljana* data set.

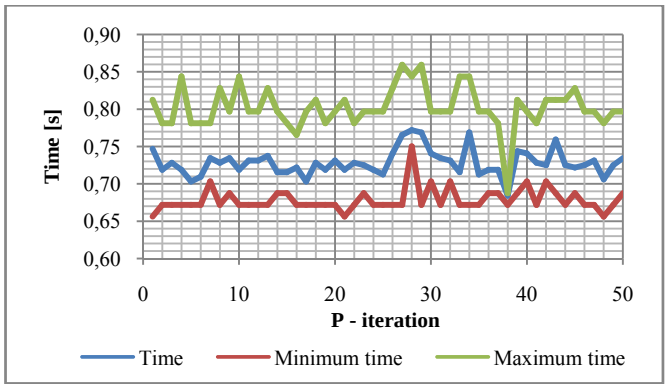

Fig. 31. Time chart based on improved IFRAIS with Alves fitness function and *Ljubljana* data set.

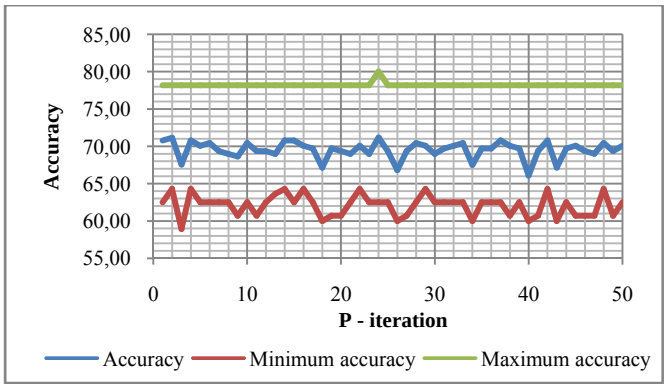

Fig. 32. Accuracy chart based on IFRAIS with Alatas fitness function and *Ljubljana* data set.

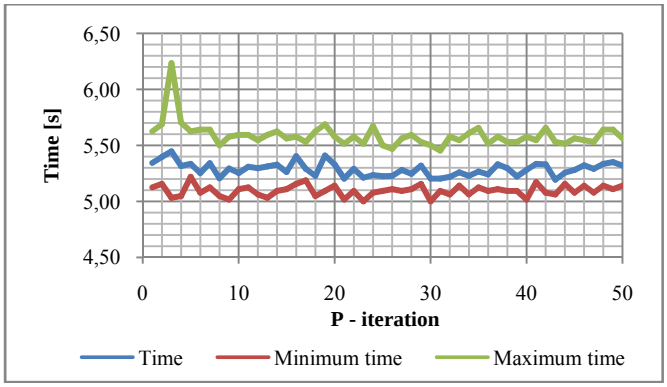

Fig. 33. Time chart based on IFRAIS with Alatas fitness function and *Ljubljana* data set.

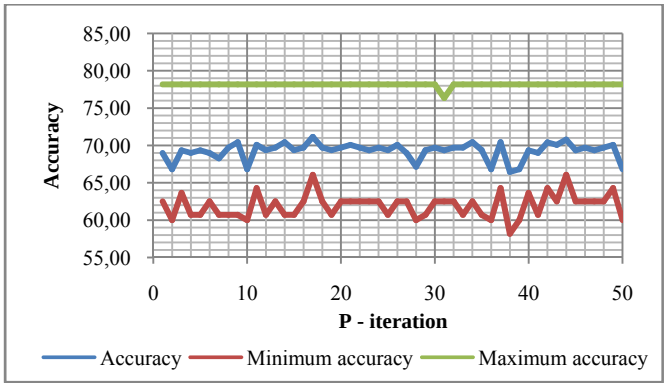

Fig. 34. Accuracy chart based on improved IFRAIS with Alatas fitness function and *Ljubljana* data set.

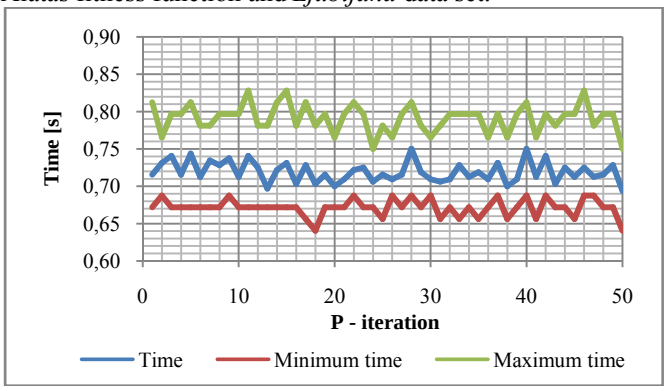

Fig. 35. Time chart based on improved IFRAIS with Alatas fitness function and *Ljubljana* data set.

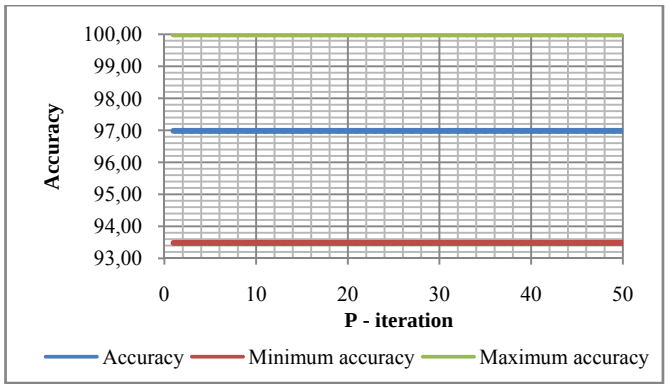

Fig. 36. Accuracy chart based on IFRAIS with Alves fitness function and *Votes* data set.

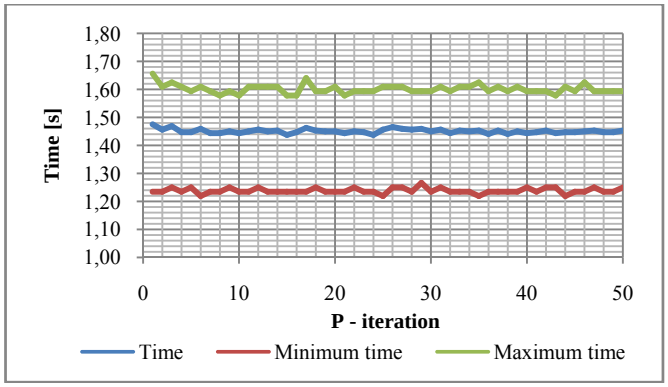

Fig. 37. Time chart based on IFRAIS with Alves fitness function and *Votes* data set.

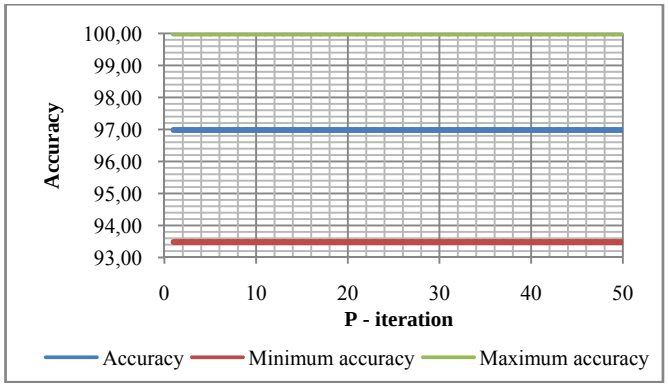

Fig. 38. Accuracy chart based on improved IFRAIS with Alves fitness function and *Votes* data set.

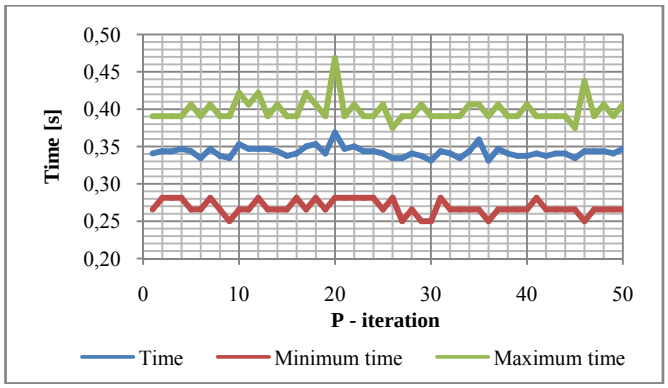

Fig. 39. Time chart based on improved IFRAIS with Alves fitness function and *Votes* data set.

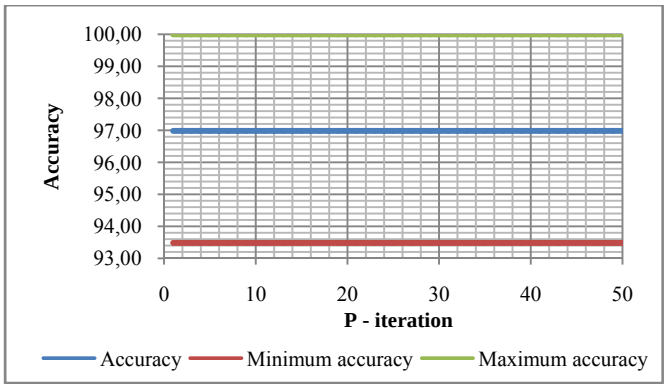

Fig. 40. Accuracy chart based on IFRAIS with Alatas fitness function and *Votes* data set.

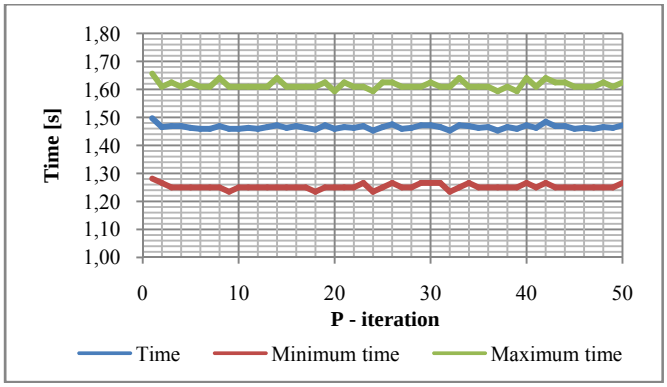

Fig. 41. Time chart based on IFRAIS with Alatas fitness function and *Votes* data set.

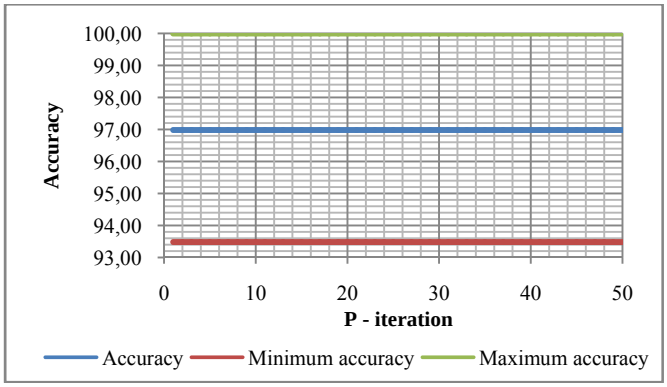

Fig. 42. Accuracy chart based on improved IFRAIS with Alatas fitness function and *Votes* data set.

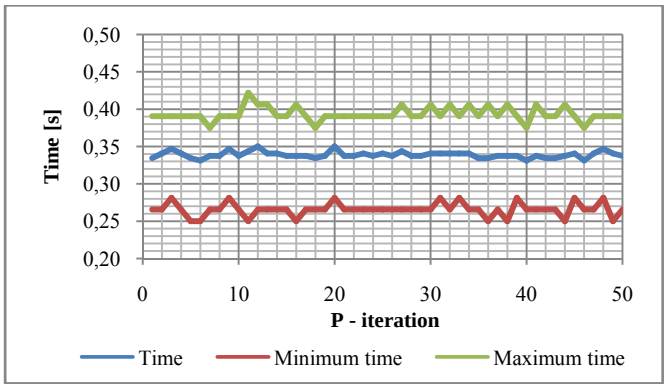

Fig. 43. Time chart based on improved IFRAIS with Alatas fitness function and *Votes* data set.

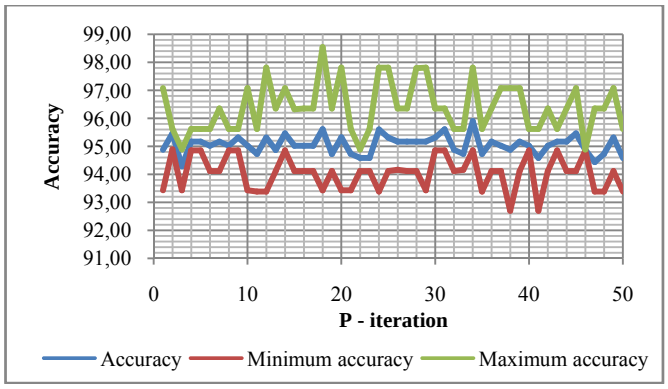

Fig. 44. Accuracy chart based on IFRAIS with Alves fitness function and *Wisconsin* data set.

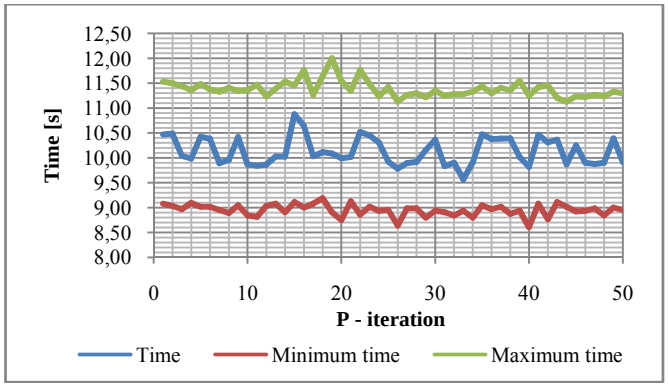

Fig. 45. Time chart based on IFRAIS with Alves fitness function and *Wisconsin* data set.

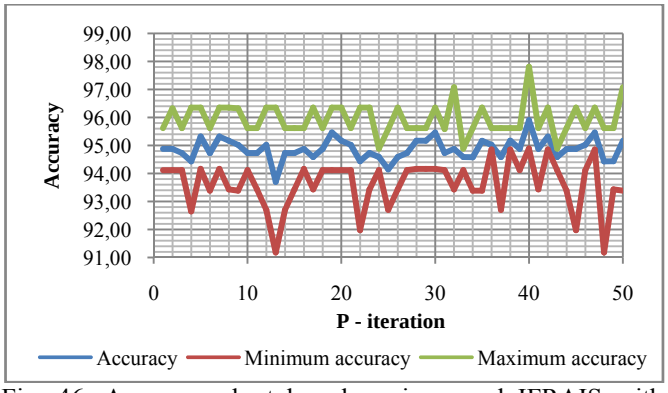

Fig. 46. Accuracy chart based on improved IFRAIS with Alves fitness function and *Wisconsin* data set.

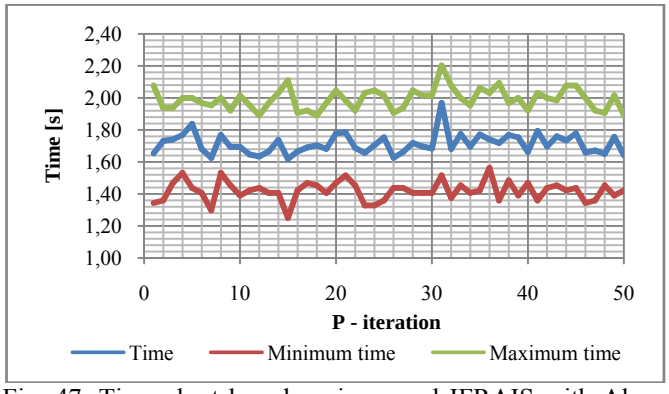

Fig. 47. Time chart based on improved IFRAIS with Alves fitness function and *Wisconsin* data set.

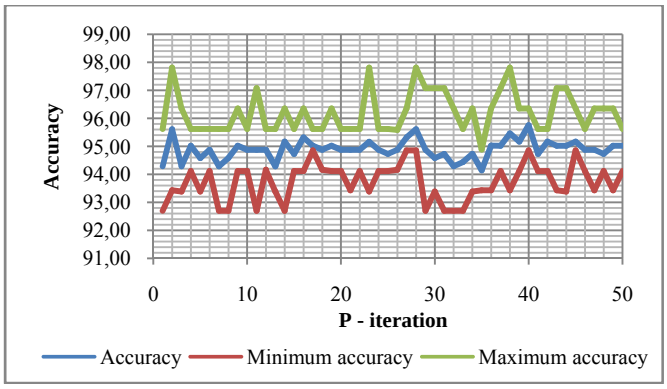

Fig. 48. Accuracy chart based on IFRAIS with Alatas fitness function and *Wisconsin* data set.

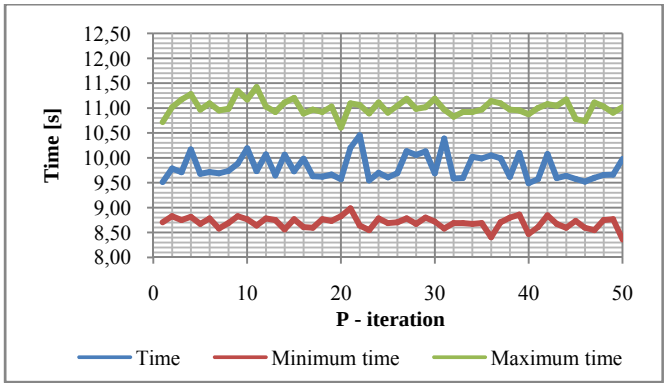

Fig. 49. Time chart based on IFRAIS with Alatas fitness function and *Wisconsin* data set.

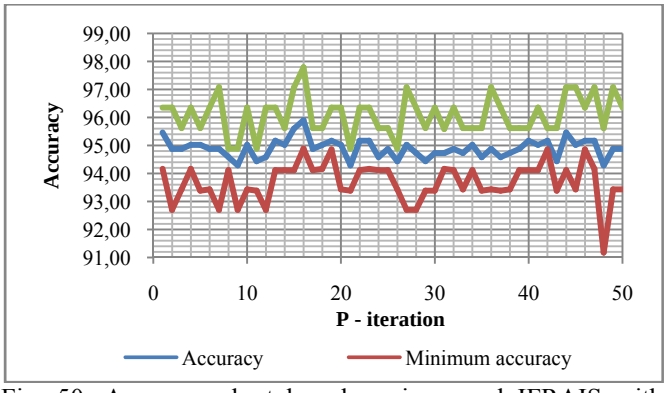

Fig. 50. Accuracy chart based on improved IFRAIS with Alatas fitness function and *Wisconsin* data set.

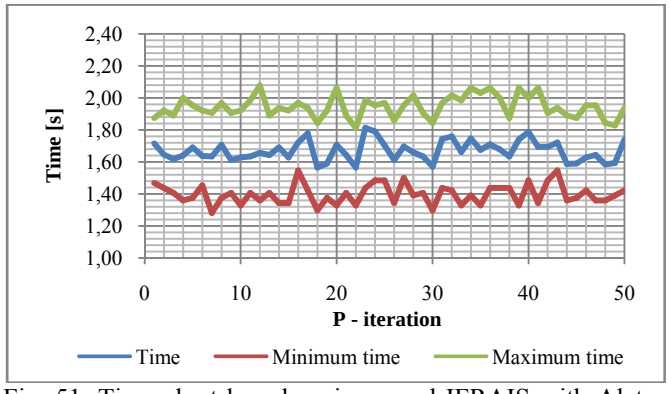

Fig. 51. Time chart based on improved IFRAIS with Alatas fitness function and *Wisconsin* data set.

5 Conclusions and future research

The accelerating extension was introduced to the IFRAIS algorithm - an AIS-based method for fuzzy rules mining. The acceleration uses the hash table, which contains the saved pairs: a rule and fitness of a rule. The hash table speeds up rapidly the computation of rule fitness in clonal selection algorithm. The improved IFRAIS was compared with standard IFRAIS algorithm in six real-world data sets. The modified algorithm works several times faster.

Performed experiments proved that introducing more complex fitness function to clonal selection doesn't affect a whole system. Both accuracy and time parameters haven't changed, stability of results remains on the same level, as well.

It seems to be still possible to improve the Induction of Fuzzy Rules with Artificial Immune Systems, and not only considering the time of working, but also the effectiveness of the induced fuzzy rules and results stability. That could be achieved mostly by introducing the learning of fuzzy partitions [10], introducing genetic operators [12] into clonal selection and also tuning clonal selection algorithm parameters. At present three and only three linguistic terms (low, medium, high) are associated with each continuous attribute. Each linguistic term is represented by the triangular membership functions. We have performed experiments, in which speed boosting IFRAIS [9] learns fuzzy partition for each attribute separately. The results are very promising. We also consider modifying the pruning function to reinforce the fitness of high-accuracy rules, as in [1]. Introduced accuracy stability is a next parameter which might be very influential on the whole system and we are in progress to investigate this indicator.

References:

[1] Alatas, B., Akin, E.: Mining Fuzzy Classification Rules Using an Artificial Immune System with Boosting , In: Eder, J. et al. (eds.) ADBIS 2005. LNCS, vol. 3631, pp. 283 – 293. Springer-Verlag Berlin Heidelberg (2005)

[2] Alves, R.T., et al.: An artificial immune system for fuzzy-rule induction in data mining. In: Yao, X., et al (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 1011–1020. Springer, Heidelberg (2004)

[3] Dasgupta, D. (ed.): Artificial Immune Systems and Their Applications. Springer-Verlag (1999)

[4] Gonzales, F.A., Dasgupta, D.: An Immunogenetic Technique to Detect Anomalies in Network Traffic. In: Proceedings of Genetic and Evolutionary Computation. pp. 1081-1088. Morgan Kaufmann, San Mateo (2002)

[5] Nasaroui, O., Gonzales, F., Dasgupta, D.: The Fuzzy Artificial Immune System: motivations, Basic Concepts, and Application to Clustering and Web Profiling. In: Proceedings of IEEE International Conference on Fuzzy Systems. pp. 711-716 (2002)

[6] Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. Analysis and Design. MIT Press, Cambridge (1998)

[7] Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann, San Mateo (2005)

[8] Zeng X., Martinez T. R.: Distribution-Balanced Stratified Cross-Validation for Accuracy Estimations. Journal of Experimental and Theoretical Artificial Intelligence. Vol. 12, number 1, pp. 1-12. Taylor and Francis Ltd (2000)

[9] Mężyk E., Unold O.: Speed Boosting Induction of Fuzzy Rules with Artficial Immune System, In: Proc. of the 12th WSEAS International Conference on SYSTEMS, Heraklion, Greece, July 22-24, 2008, 704-706 (2008)

[10] Kalina A., Mężyk E., Unold O.: Accuracy Boosting Induction of Fuzzy Rules with Artificial Immune Systems, In: Proc. of International Multiconference on Computer Science and Information Technology, Wisla, Poland, October 20-22, 2008, (2008) (in print)

[11] Sawarkar S., Ghatol A., Pande A.: Neural Network Aided Breast Cancer Detection and Diagnosis Using Support Vector Machine, In: Proc. of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp158-163)

[12] Frausto-Solis J., Rivera-Lopez R.: A Simplex-Genetic method for solving the Klee-Minty cube, In: Transactions on Systems, WSEAS, 2(1) pp 232-237, ISSN 1109-2777, April 2002