DEVELOPMENT AND ANALYSIS OF GENETIC ALGORITHM FOR TIME SERIES FORECASTING PROBLEM

Leonid Hulianytskyi, Anna Pavlenko

Abstract: *This paper presents developed genetic-based algorithm for time series forecasting problem and describes approaches to learning procedures design. Different techniques of population representation, recombination, formation of niches, calculation of fitness, conflict resolution methods are proposed. Results of computational experiments with real time series are analyzed.*

Keywords: *forecasting, genetic-based machine learning, rule-based forecasting, genetic algorithm, time series forecasting, evolutionary algorithms.*

ACM Classification Keywords: *I.2.8 Problem Solving, Control Methods, and Search*

Introduction

Applications of evolutionary computation to machine learning are referred to as genetic-based machine-learning (GBML). Evolutionary computation (EC) techniques belong to the class of optimization tools, inspired by biological processes. The main idea of EC lies in the iterative modification of the population of individuals (candidate solutions of the problem – chromosomes) with selection and recombination procedures.

Rule-based genetic algorithms (GA) are successfully applied to the solution of machine learning problems due to natural scalability, parallelization, noise resilience, flexibility of objective function, universality of computational scheme and possibility of using heuristics for data representations [Kovacs, 2010]. On the other hand, rule-based forecasting is able to take into account several time series at once and consider existing causal relationships in complex economic processes, which are significantly affected by various factors.

We consider time-series forecasting as a special type of *classification* – or supervised learning task (learning algorithm knows about class of an example). Classification problem can be defined as follows: Given a set of instances $\Xi = \{i_1,...,i_n\}$, each of them labeled with a finite set of classes $C = \{c_1,...,c_m\}$, one wants to create a certain theory T based on Ξ [Bacardit, 2004].

Learning algorithm produces instances, where each one contains a finite and fixed set of elements – attributes, that represent features of the instances. This research operates with nominal attributes – values can be obtained from given discrete set. Set of classification rules is a result of the learning

algorithm. Classification rules are usually represented as follows: **If** *condition* **Then** *action.* The left hand side of each rule (*Condition*) is a conjunction of one or more tests involving feature values. The right hand side of a rule (*Action*) indicates the class is assigned to examples, which match its left hand side. Classification rules in the set might overlap [De Jong K.A., 1993]. GBML systems use sets of rules as knowledge representation.

This paper proposes a forecasting method based on GA [Mahfoud, 1996] and explores issues of its implementation and application.

Objectives of the research include proposing an approach to GA application for constructing rule base that should be able to recognize instances of the target concept correctly and discriminate them from objects that do not belong to it. This paper also describes approach to fitness-function calculation, data representation and conflict resolving scheme for forecasting problem.

General Scheme of the Genetic Algorithm

The underlying commonality of GBML is the use of an EC as the search mechanism.

Rule-based forecasting process is divided into the following steps:

1. Learning phase – construction of the rule base that describes all dependencies in time series using selected learning algorithm (GA in this research). Current algorithm employs incremental learning scheme – knowledge base could be updated as new examples arrive to the system.

2. In the match phase, all rules are checked for admissibility of use at the particular position, and conflicting rules set is formed.

3. In the conflict resolution phase, we select rules that would be used for forecasting from the set of conflicting rules.

4. During action phase, we apply rules to the forecast for the predefined period.

GA is a common technique and has different implementations for each problem, but it can be generally defined with scheme represented in the figure 1 [Eiben A.E., 2007]. GA works with population of individuals – admissible solutions to the problem. Initial population can be composed randomly, or it can be filled with expert solutions. After initiation, each individual is evaluated via fitness function. When population is fully constructed, reproduction cycle begins: the population is being modified through parents' selection, recombination, probabilistic mutation and evaluation. It is worth to mention that many operations in GAs are stochastic. Current algorithm is steady-state – only a subset of individuals (usually, only two of them) is modified during each iteration [Fernandez A., 2010].

There are two main approaches for representing a rule set in terms of GA. In the Pittsburgh approach, one chromosome encodes one rule, therefore, a problem of complexity of structures appear. In the

Michigan approach, one rule is usually represented by many chromosomes, therefore, credit assignment appears to be more difficult.

In terms of GA, a single rule is a chromosome and rule base is a population. Population is a set of possible solutions. It also could be defined as a single unit of evolution. Quantity of different solutions in the population represents its *diversity*.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Figure 1. Pseudo-code of GA general scheme

Conflict resolution strategy is used in rule-based systems to make decision which rule should be applied if there are several applicable ones. There are many different approaches for conflict resolution [Sasikumar M., 2007]. One of the easiest is rule ordering (first come, first served), or selecting a rule that appeared in the conflict set first. Another way to resolve the conflict is to select the rule that is the most specific, i.e. the rule with the longest conditional part is preferred. According to the recency strategy, the rule, that uses the most recently added data, is applied. Priority strategy selects rules that are important in some ways (usually, priorities are determined by expert). Refractoriness approach ignores rules, which had already been selected (rules are removed from the conflict set after selection). A combination of different strategies is often used to solve forecasting problems. Sensitivity and stability throughout the system depends on effectiveness of the conflict resolution schemes.

Rule Representation

In this paper, population is represented using Pittsburgh approach, i.e. *rule* is an *if-then* construction and consists of arbitrary number of conditions w (atomic subrules) and final value w_c. Atomic subrule is the proposition that growth of time series values in random points belongs to some interval retrieved by time-series ranking [Гуляницкий Л.Ф., 2014]:

$$w = (F_{t_1} - F_{t_2}) \in [y_b, y_{b+1})$$

where F_t – value of the time series at the point t;

t_1, t_2 – arbitrary points,

y_b, y_{b+1} – real numbers, used for increment ranking ,

$b = \overline{-m, m-1}$, $m \in N$.

Here:

$$y_b = \frac{F_{\max}}{m} \times b$$

where $F_{\max}$ – maximal absolute value of the forecast (positive or negative), which is expertly determined for particular model. Ranking procedure results in construction of $2m$ intervals (for negative and positive gains). It is worth noting that the prediction algorithm, based on GAs, is able to work with multiple time series and, therefore, take into account causal factors of complex systems. In this case, parameters $F_{\max}$ and m must be set individually for each series [Гуляницкий Л. Ф., 2014а].

Full rule is defined as:

$$\Omega_j = if\left(\bigwedge_{i=1}^{z} w_i\right) \quad then \quad w_c,$$

where w_i – subrule of the conditional part,

i – index of the subrule,

$i = \overline{1, z}$, z – rule length (number of subrules in the full rule),

w_c – subrule, which is applied in case of triggering condition *if*.

Note that the rules use relative indexing instead of absolute values, i.e. all subrule indices are determined by the distance to the initial zero index.

Thus, all rules represent pattern that is applied to the time series via serial shift:

$$\tilde{w} = \Delta(s, k) \in [y_b, y_{b+1}),$$

where s, k – indices that define the distance relative to the initial index of rules.

Consider a graphical interpretation of a rule with one subrule and its application to a time series (Figure 2). The rule $if(\Delta(s_0, s_7) \in [0,50))$ $then$ $(\Delta(s_7, s_{16}) \in [0,50))$ indicates that if the growth function of two points (the distance between which is 7) is in the range from 0 to 50, then the growth between the 7th and the 16th points is in the range from 250 to 300.

Pattern (rule) is verified for all t. It is obvious that the rule is satisfied for $s_0 = 10$, $s_1 = 17$, $s_2 = 26$.

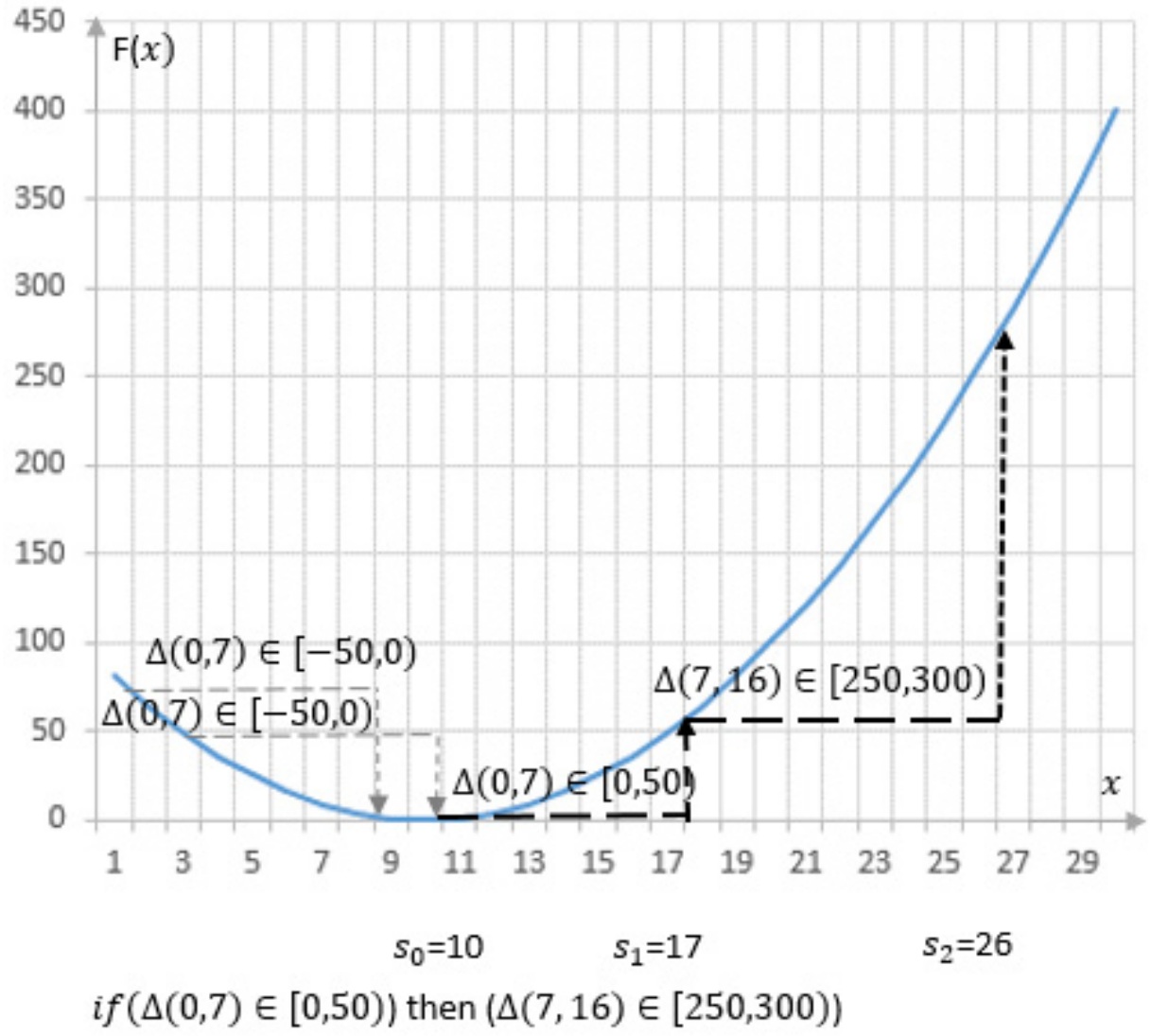

Figure 2. Graphical interpretation of the rule application to the time series

Fitness Function

The quality of each individual of the population is evaluated with fitness function ϕ, that takes into account frequency φ of the rule in the time series F and length of the rule z:

$$\phi = \begin{cases} \alpha z + (1-\alpha)\varphi, & \varphi > \delta \\ 0, & \varphi \le \delta \end{cases},$$

where δ – parameter of the algorithm that controls minimal allowed frequency of the rule,

α – parameter of the algorithm that adjusts weight values for frequency and rule length.

Despite definition of the rule as a conjunction of conditions, we assume that the rule is satisfied at the point t, if the share of satisfied subrules ω_i is higher than a defined parameter ϖ, and ω_c is completely satisfied.

Rule satisfaction frequency $\varphi(\Omega_j)$ is calculated by sequential shifting of rule indices along time series and reviewing rule compliance at each point t:

$$\varphi(\Omega_j) = \sum_{t=1}^{N} \Omega_j^t .$$

where $\Omega_j{}^t$ indicates whether rule is satisfied ($\Omega_j{}^t = 1$, if rule is satisfied at the moment t, $\Omega_j{}^t = 0$ in other cases).

Rule similarity

When GAs are used for optimization, the goal is generally to return a single value – the best solution found. Usually, if the conventional GA performs a sufficient number of iterations, the entire population converges to the neighborhood of one solution [Mahfoud SW, 1995]. Such approach is not suitable for forecasting problem, since it is necessary to obtain a sufficiently broad base of rules to make a final forecast.

GAs, that use niching procedures, are able to find and use different rules from the same population. The basic idea is simultaneous optimization in several areas of the search space, performed by reducing competition between sufficiently dissimilar individuals. In financial forecasting, different rules within a single population of GA are able to make predictions for various market and company conditions [Sam Mahfoud, 1996].

There is a variety of niching strategies: *fitness sharing* [Goldberg DE., 1987] – reducing the value of fitness for similar individuals in a population; *crowding* – new individuals replace older elements of the population; *sequential niching* – forcing restart of traditional GA in different search spaces and so on.

To implement niching procedure for the genetic-based forecasting algorithm we define the concept of the rules similarity as the degree of similarity of chromosomes genotypes. Two rules are similar, if the percentage $\tilde{w}$ of their similar subrules is higher than specified model parameter σ, $\sigma = \overline{1,\ldots,100}$.

Subrules $w_1 = \Delta(s_1, k_1) \in [y_{b_1}, y_{b_1+1})$ and $w_2 = \Delta(s_2, k_2) \in [y_{b_2}, y_{b_2+1})$ are similar, if their subrules have similar shift with respect to the initial rule index, cover segment of similar size on the time series and belong to the similar range of values:

$$|s_1 - s_2| \le \vartheta, \quad \vartheta = \overline{1,n}$$

$$|(s_1 - k_1) - (s_2 - k_2)| \le \varsigma, \quad \varsigma = \overline{1,n},$$

$$|b_1 - b_2| \le \gamma, \quad \gamma = \overline{1,n}$$

$$s_1 > k_1, \quad s_2 > k_2, \quad s_1 = \overline{1,n}, \quad k_1 = \overline{1,n}, \quad s_2 = \overline{1,n}, \quad k_2 = \overline{1,n}$$

where s_1, s_2 – starting subrule indices,

k_1, k_2 – ending subrule indices,

ϑ – model parameter that defines permissible shifting error of the starting index of similar subrules,

ς – model parameter that defines permissible error of segment length,

b_1, b_2 – indices that define intervals (ranks) for subrules;

γ – model parameter that defines permissible rank error of similar subrules.

Two rules are similar, if the percentage of similar subrules is higher than a defined model parameter ρ.

The procedure for similarity search is used to eliminate redundant rules at the recombination phase, at the stage of survival selection, at selection for recombination in inbreeding and outbreeding methods etc.

Initialization

Initial population has a fixed size, specified by the parameter of the algorithm. It is constructed from a given time series, considering the minimal allowed size of rules, specified by the parameter.

Subrules are constructed by selecting random points of the time series and comparing the corresponding growth at the points. Thus, the initial population always contains rules that are satisfied at least once. Every new rule is checked for already existing similar rules in the population. Fitness value is also calculated for each rule at the initialization step.

Selection for Recombination. Crossover, Mutation

Selection for recombination determines parental pairs for further crossover and mutation. In the research, there were implemented such methods of selection for recombination: panmixia, inbreeding, outbreeding and roulette-wheel selection. *Panmixia* – selection type, where any two random individuals have the same probiliity to form a parent couple. *Inbreeding* – selection type, where genetically similar chromosomes are preferred as parent couple. In *outbreeding*, individuals with genetically different encodings are preferred. *Tournament selection* is held by selecting random $1 \le d < N$ individuals from the population (N – population length) and choosing best individuals according to their fitness. *Roulette-wheel selection* is a stochastic selection, where probability of selection is proportional to the fitness function.

Implementation of inbreeding and outbreeding uses search of similar rules. Share of similar subrules is used as comparative characteristic of similarity.

The method of selection is the same throughout the run of the algorithm and is determined by model parameter.

Recombination consists of two stages: crossover and mutation. Crossover operators are implemented differently depending on data presentation, but, usually, offsprings inherit traits of both parents.

The algorithm uses one-point crossover, so crossing point v (subrule index) for rules Ω^1 and Ω^2 is selected randomly, and offsprings Ω^{12} and Ω^{21} are expressed as following:

$$\Omega^1 = if\left(w^1{}_1 \wedge w^1{}_2 \wedge \ldots \wedge w^1{}_v \wedge w^1{}_{v+1} \wedge \ldots \wedge w^1{}_{z-1} \wedge w^1{}_z\right) \quad then \quad w^1{}_c,$$

$$\Omega^2 = if\left(w^2{}_1 \wedge w^2{}_2 \wedge \ldots \wedge w^2{}_v \wedge w^2{}_{v+1} \wedge \ldots \wedge w^2{}_{z-1} \wedge w^2{}_z\right) \quad then \quad w^2{}_c,$$

$$\Omega^{12} = if\left(w^1{}_1 \wedge w^1{}_2 \wedge \ldots \wedge w^1{}_v \wedge w^2{}_{v+1} \wedge \ldots \wedge w^2{}_{z-1} \wedge w^2{}_z\right) \quad then \quad w^2{}_c,$$

$$\Omega^{21} = if\left(w^2{}_1 \wedge w^2{}_2 \wedge \ldots \wedge w^2{}_v \wedge w^1{}_{v+1} \wedge \ldots \wedge w^1{}_{z-1} \wedge w^1{}_z\right) \quad then \quad w^1{}_c.$$

Mutation procedure is held with probability p, defined as a model parameter. Indices s, k or rank b could be changed in a random subrule $\tilde{w} = \Delta(s,k) \in [y_b, y_{b+1})$ according to the allowed space of values.

Procedures of reconstructing rule indices with respect to the initial one, verification of similar subrules and fitness evaluation are performed after each recombination operation.

Survival selection. GA termination condition

After the recombination stage, offsprings are added to the population, if no similar chromosomes already exist, or they replace existing similar individuals, if their fitness value is higher. Note that in this implementation, there exists a threshold of minimal number of rule satisfactions in the time series.

Stopping criterion of the algorithm is defined as a sufficient number of run iterations I. GA produces a set of rules with a given estimation of quality – the value of the fitness function, which is used for forecasting.

Construction of the final forecast

Matching phase. Each rule is checked if it can be applied for each of the prediction position $\tilde{t}$. Rules are checked for those elements of the time series, where the conditional part belongs to known values of the time series, and "*then*" part belongs to the period of advance. If the conditional part of the rule is satisfied for a certain position t, then a possible forecast can be made using "*then*" part.

The result of matching phase is a set of conflicts $\{< \Omega_j, \phi, offset >\}$ of length $h_{\tilde{t}}$ for each prediction position $\tilde{t}$. The set contains information about the rule Ω_j, its fitness value ϕ and rule's *offset* on the time series. Thus, each position $\tilde{t}$ is corresponded to $h_{\tilde{t}}$ alternative values $\hat{G}$ of the forecast.

Conflict resolution phase. In the research, aggregated value across all conflicting values $\hat{G}$ is calculated considering priority of each rule (its fitness function). The above interpretation of the meaning of fitness contains information about the specificity of the rule and its satisfaction frequency in the time series.

Action phase. Input information of the forecasting algorithm contains sets of conflicts, time-series data and period of advance . The predicted value is calculated as:

$$\hat{F}_{\tilde{t}} = \frac{\sum_{i=1}^{i=h_{\tilde{t}}} \hat{G}_i \times \phi_i}{\sum_{i=1}^{i=h_{\tilde{t}}} \phi_i},$$

where $\tilde{t}$ – position in the advancing period for which the forecast is calculated, $\tilde{t} = \overline{1,P}$,

P – advancing period,

$\hat{F}_{\tilde{t}}$ – resulting value of the forecast for the position $\tilde{t}$,

$\hat{G}_i$ – conflicting value of the possible forecast for the position $\tilde{t}$,

$h_{\tilde{t}}$ – number of conflicts for the position $\tilde{t}$,

ϕ_i – fitness value of the rule that caused conflict $\hat{G}_i$,

$i = \overline{1,h_{\tilde{t}}}$.

Analysis of implementation issues of the forecasting algorithm

We developed specialized forecasting software package in language C#, using proposed approach. The calculations were performed on the virtual machine Microsoft Azure A8, designed for intensive computing, which has the following characteristics: processor Intel Xeon 2.5 Gh, 8 cores, 56 GB of RAM. To evaluate the efficiency of established software and algorithmic tools, we conducted computational experiment on the prediction of the real-world time series (Figure 3). We used monthly time series of actual sales of drugs in pharmacies in Ukraine for the period of 2004-2013 years, expressed in UAH. To achieve the goal of computing experiment, time series data were divided into a training sample (2004-2012 years), test sample (2012 year) and control sample (2013 year).

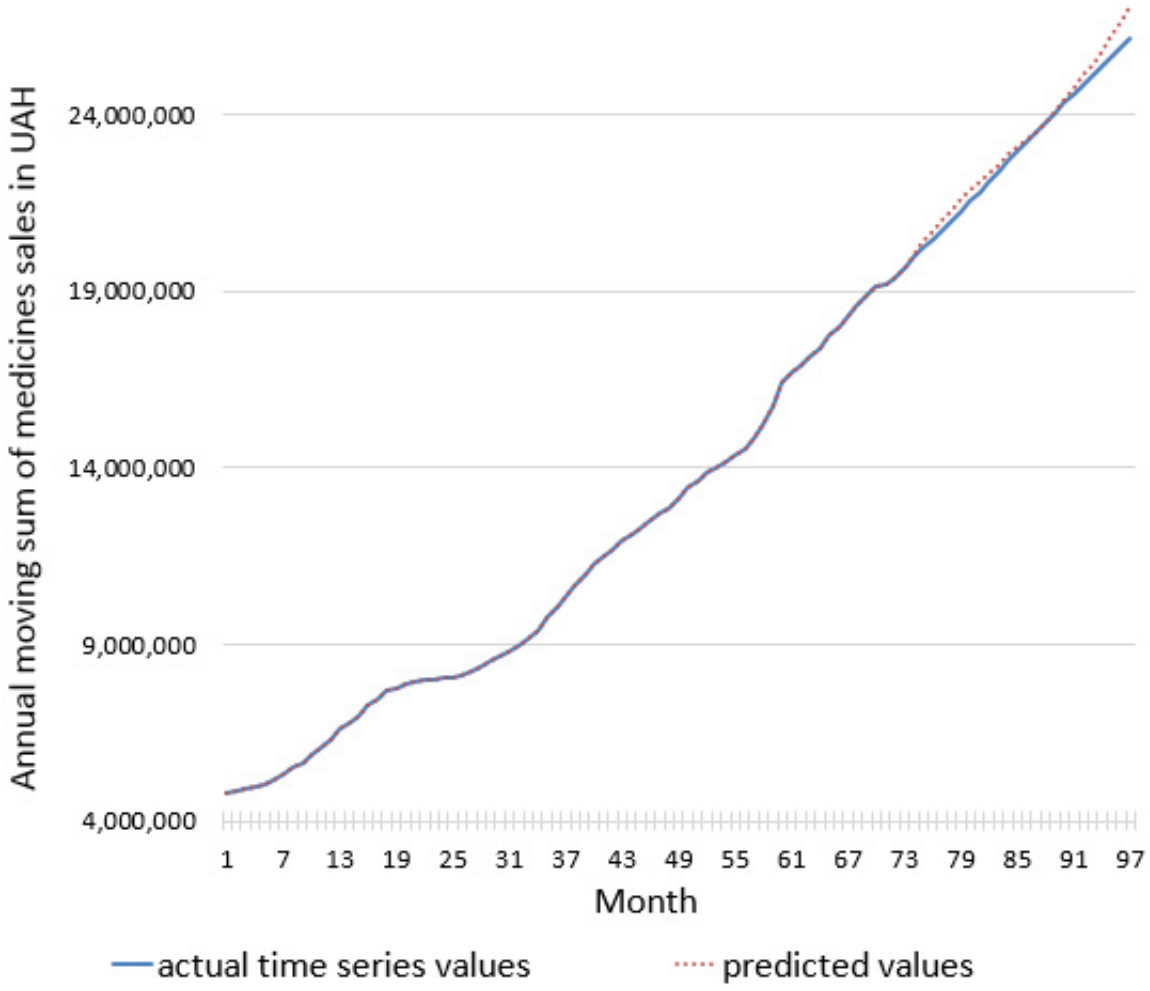

Figure 3. Forecast of annual moving sum of drug sales in UAH

Error evaluation is based on the known criteria MPE, MAPE [Hanke, 2008]:

$$MPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\hat{F}_t - F_t}{F_t} \times 100\%,$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{\hat{F}_t - F_t}{F_t}\right| \times 100\%$$

where $\hat{F}_t$ – predicted value,

F_t – actual value of the time series,

N – number of points for comparison (usually, the available time series),

t – time moment, $t = \overline{1,N}$.

Three sets were obtained from the time series: training set, test set and control set. The test set is used to validate correctness of the generated theory, that is, that the learning system has been able to model the concept, represented by instances in the training set, instead of modelling only instances themselves [Bacardit, 2004].

Table 1 shows the actual data for 2012th year, forecast for 2012th year (test sample), actual data for 2013th year and forecast for 2013th year (control sample), obtained via computational experiment.

Computational experiment shows that average values of forecast errors by the criterion of MPE on the test and the control samples were -1.1% and -1.2% respectively, while for the criterion MAPE – 1,1% and 1.2%, which is considered acceptable. [Гуляницкий Л.Ф., 2014b].

Table 1. Forecasting results

№	Actual values of the test sample	Forecast values of the test sample	Actual values of the control sample	Forecast values of the control sample
1	19949504	20059232	23228344	23295940
2	20206460	20387146	23494700	23487183
3	20463903	20712021	23765372	23762372
4	20721069	20981187	24051883	24069264
5	20993146	21297783	24323290	24412651
6	21261310	21593159	24567628	24754267
7	21530600	21851015	24828381	25108251
8	21817053	22098811	25091582	25418312
9	22105440	22312336	25350755	25794917
10	22393529	22604263	25609683	26224947
11	22679450	22882337	25876747	26606814
12	22961491	23044748	26141320	27063485
MAPE		**1.1%**		**1.2%**
MPE		**-1.1 %**		**-1.2%**

Forecasting algorithm's parameters settings and tuning

In the study, we carried on an experiment to determine optimal parameters of the algorithm. Number of conducted experiments for each combination was $K = 50$, the period of advance of the forecast was 12 months. Experiment used a smoothed time series – annual moving sum of medicines sales in UAH.

Figure 4 shows the dependence of the error MAPE, selection for recombination approach and minimal size z of the chromosome in the initialization phase with other conditions remaining the same ($\vartheta = 3$, $\varsigma = 3$, $\rho = 70\%$, $\gamma = 3$, $m = 100$, $\alpha = 0.8$, $\delta = 1$, $I = 100$, $\upsilon = 20\%$). It is obvious that with the growth of the initial size of the chromosome, error increases. This can be explained by the fact that during the growth of the chromosome size, number of rule satisfactions is reduced, and rules lose generalization feature. Additionally, it is necessary to analyze the possibility of simultaneous increase of the specifity of initial rules (size of chromosomes), the initial population size and the number of learning iterations.

Figure 5 represents the dependence of MAPE criterion, selection for recombination approach and number of iterations with other conditions remaining the same ($\vartheta = 3$, $\varsigma = 3$, $\rho = 70\%$, $\gamma = 3$, $m = 100$, $\alpha = 0.8$, $\delta = 1$, $z = 3$, $\upsilon = 20\%$). Figure 5 shows that the best MAPE value is reached at $I = 600$ for all selection methods. Rule set is not learnt enough within small number of iterations and becomes overlearnt (rule set converges quicklier and niching methods become less efficient) with growth of learning duration.

Figures 6-8 show dependence between initial rule length z (subrules count), initial population size and MAPE error.

It is obvious from figure 6 that aggregated MAPE error across all values of initial population size is the best for initial chromosome length z equal to four. Additional experiments showed that long rules have lower satisfaction frequency, therefore, their fitness is smaller. Figure 7 demonstrates that with growth of initial population size for fixed number of iterations MAPE value becomes higher. Figure 8 displays dependence between above parameters in three dimensions.

Figures 9-11 describe dependence between permissible subrule shifting error ϑ, permissible covered segment error ς and MAPE error. Both parameters ϑ and ς are used in niching procedures, but experiment results show that they are not correlated.

Figures 12-14 show dependence between permissible interval errors γ, total number of intervals $2m$ in time series and MAPE Error.

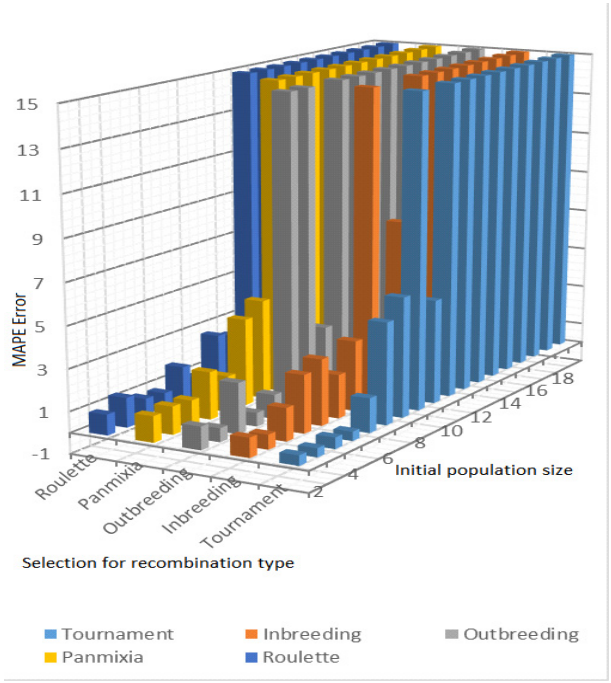

Figure 4. Dependence of the MAPE, selection for recombination approach and minimal size *z*

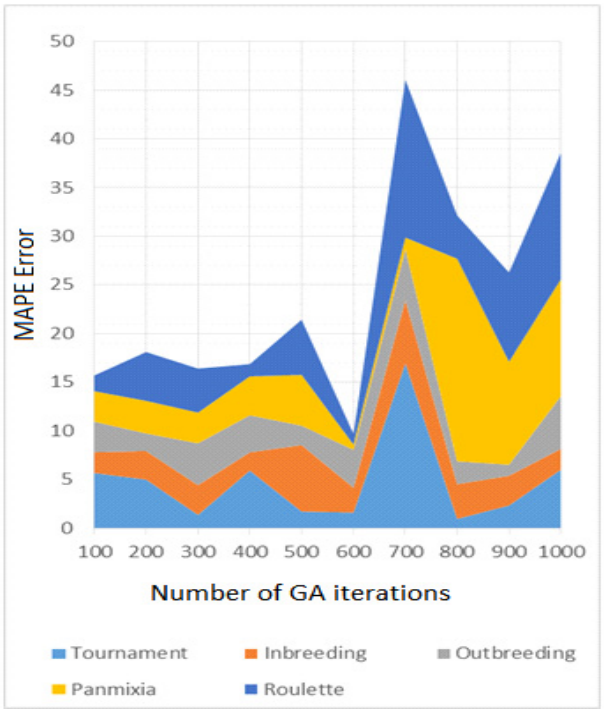

Figure 5. Dependence of the MAPE, selection for recombination approach and minimal size *z*

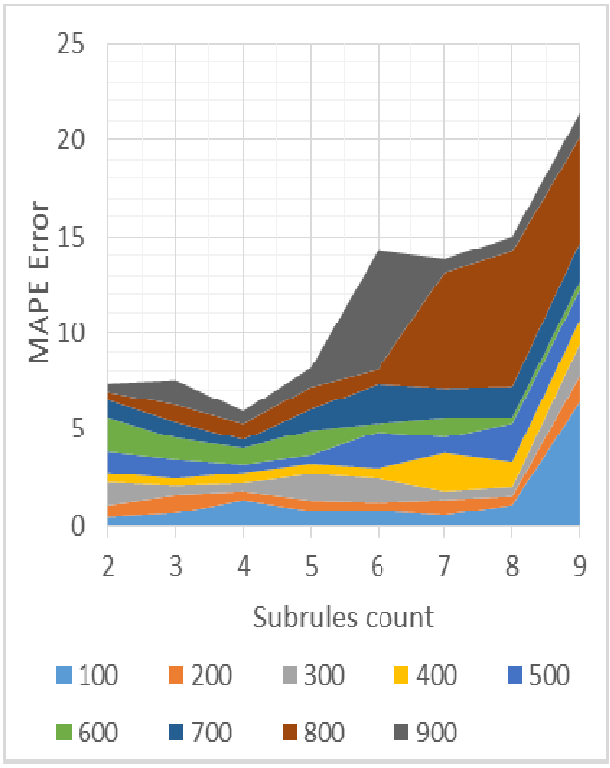

Figure 6. Dependence between subrules count, population size and MAPE

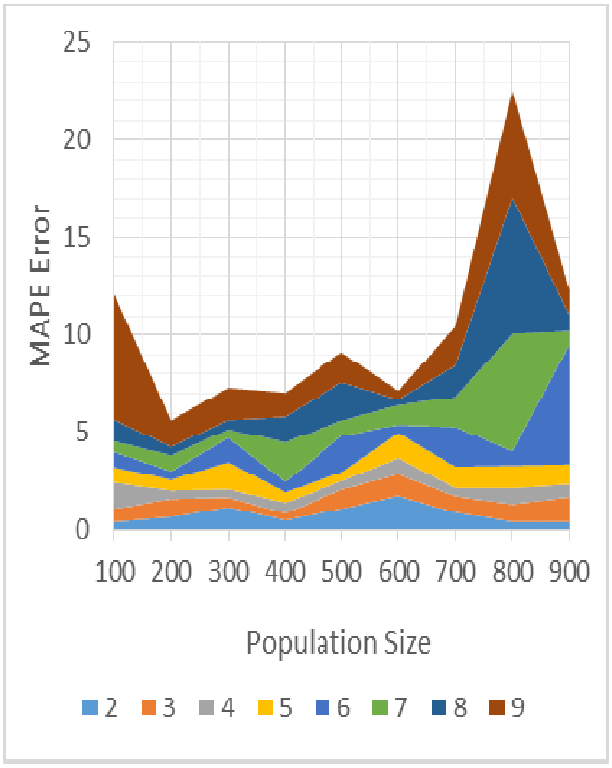

Figure 7. Dependence between subrules count, population size and MAPE

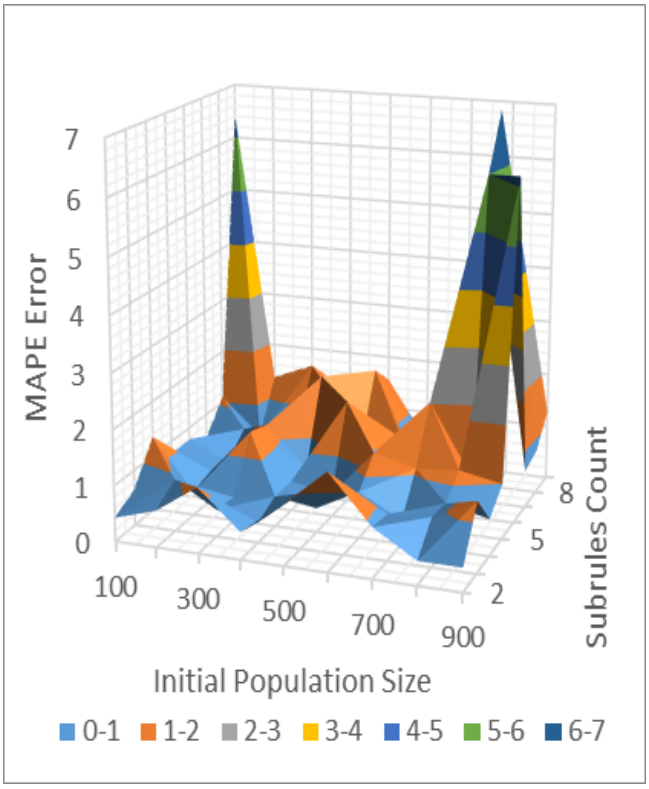

Figure 8. Dependence between subrules count, population size and MAPE

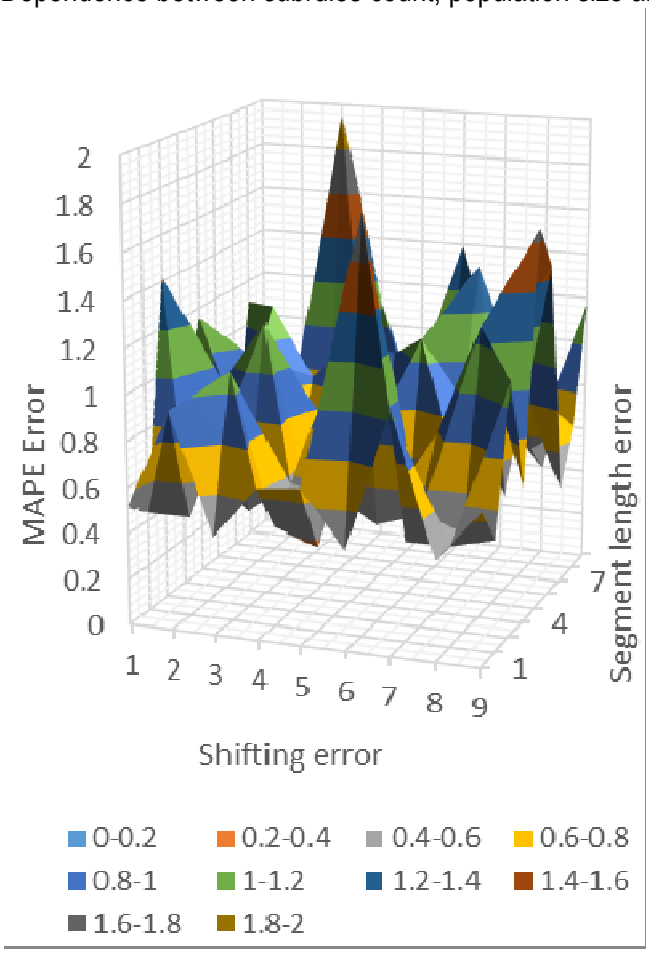

Figure 9. Dependence between permissible subrules' shifting errors, covered segment error and MAPE

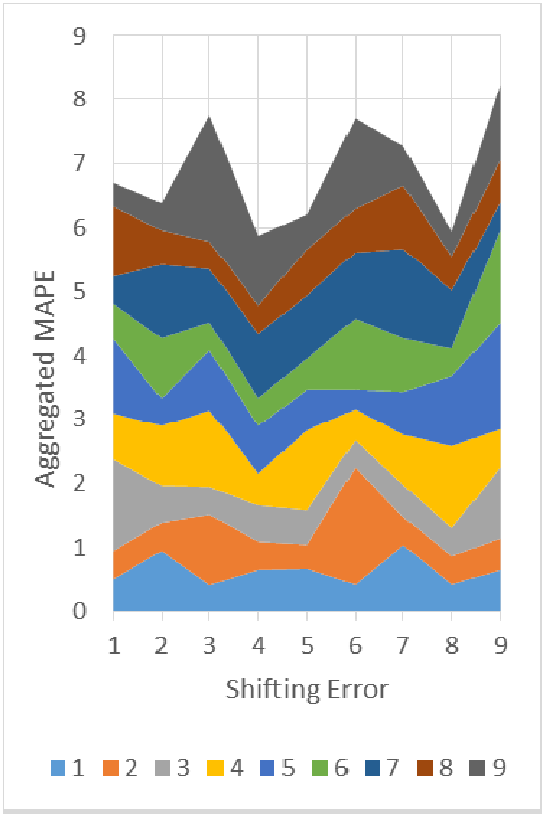

Figure 10. Dependence between permissible subrules' shifting errors, covered segment error and MAPE

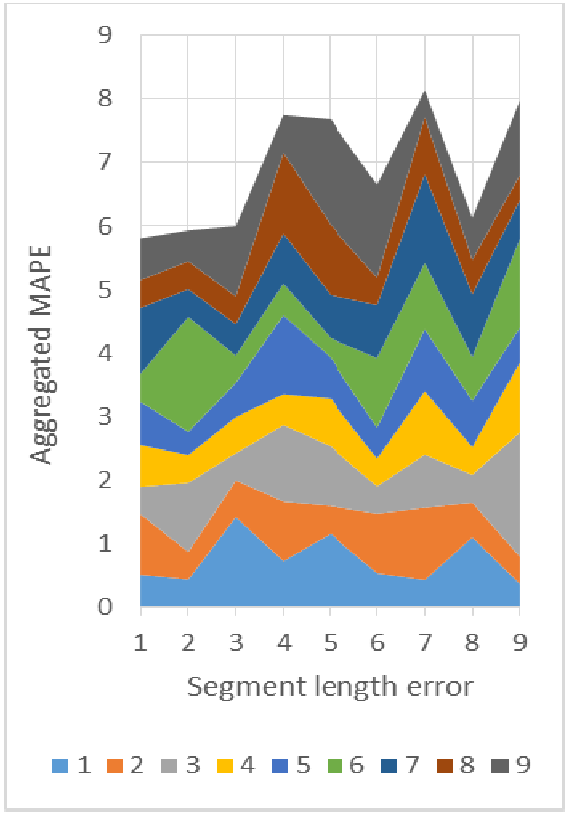

Figure 11.– Dependence between permissible subrules' shifting errors, covered segment error and MAPE

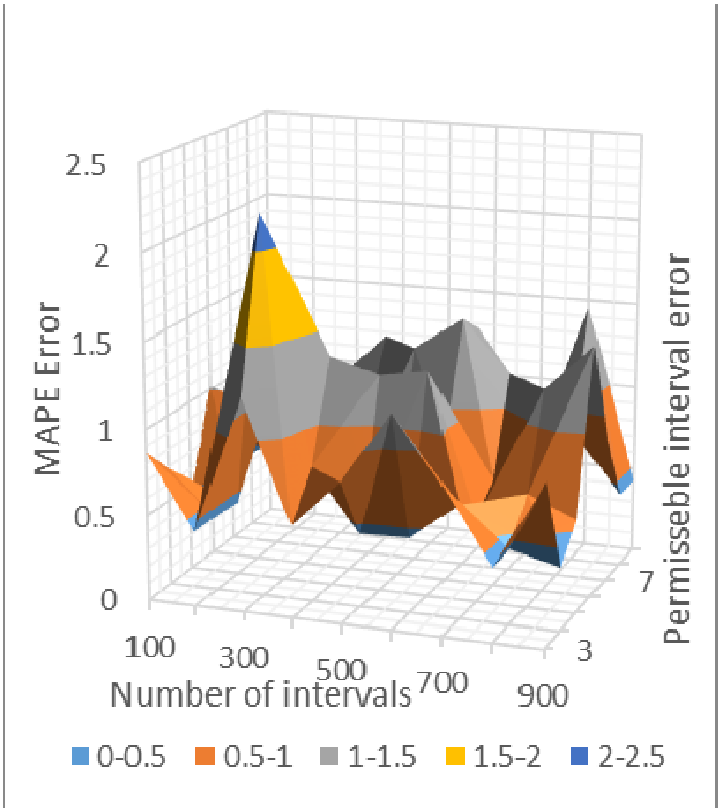

Figure 12. Dependence between permissible interval errors, total number of intervals in time series and MAPE

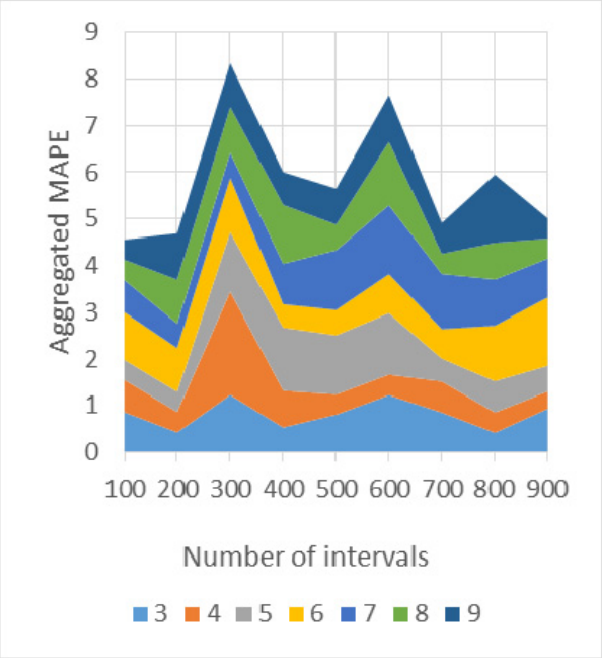

Figure 13. Dependence between permissible interval errors, total number of intervals in time series and MAPE

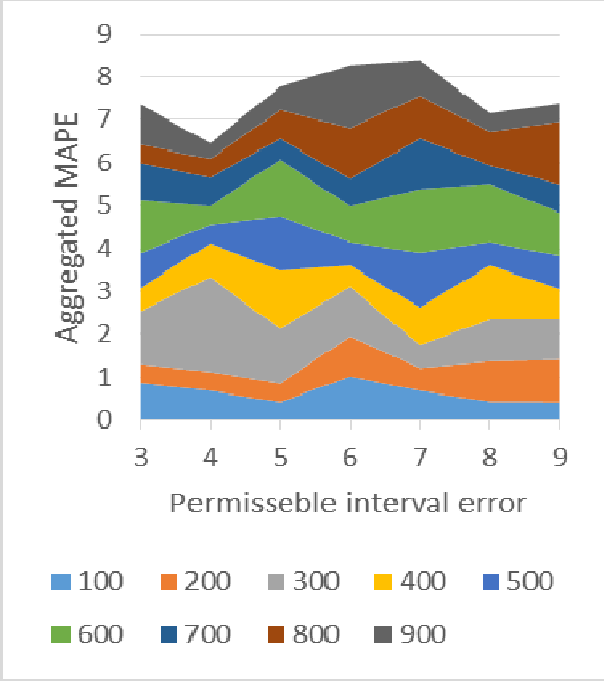

Figure 14/ Dependence between permissible interval errors, total number of intervals in time series and MAPE

Conclusion

This paper describes main stages of rule-based forecasting using GA. The proposed method of data representation allows applying the algorithm for forecasting problems with multiple time series. We also designed an approach to chromosome quality evaluation, which considers its length and satisfaction frequency. These measures of rules quality are used at the stage of conflict resolution and construction of the final forecast. A mechanism for the diversification of the search space was developed. Genetic operators for the proposed method of data presentation are described.

We conducted several computational experiments to select optimal values of algorithm parameters.

Further research requires to analyze issues of selection of optimal parameters of the algorithm and construction of the forecast with the influence of several factors.

Bibliography

[Kovacs, 2010] Tim Kovacs. Genetics-based Machine Learning. Handbook of Natural Computing: Theory, Experiments, and Applications. Springer Verlag, 2010. – 62p.

[Eiben A.E., 2007] Eiben A.E., Smith J. E. Introduction to Evolutionary Computing / Natural Computing Series .– Springer, 2007. – 300p.

[Bacardit, 2004] Jaume Bacardit. Pittsburgh Genetic-Based Machine Learning in the Data Mining era: Representations, PhD thesis, Ramon Llull University, Barcelona, Spain, 2004. – 352p.

[Sam Mahfoud, 1996] Sam Mahfoud, Ganesh Mani. Financial forecasting using genetic algorithms. – Applied Artificial Intelligence, 1996. – P. 543-560.

[Fernandez A., 2010] A Fernández, S García, J Luengo, E Bernadó-Mansilla, F Herrera. Evolutionary Computation, IEEE Transactions, 14 (6), 2010. – P. 913-941.

[Павленко А.І., 2014] Павленко А.І. Застосування генетичного алгоритму для прогнозування на основі бази правил / Компьютерная математика. - 2014. – P. 74-80.

[Sasikumar M., 2007] Sasikumar M., Ramani S., Muthu Raman S., Anjaneyulu KSR. A Practical Introduction to Rule Based Expert Systems. - Narosa Publishing House, New Delhi, 2007. – 294p.

[De Jong K.A., 1993] De Jong K.A., Spears W.M., Gordon D.F. Using genetic algorithms for concept learning.– Machine Learning, 1993. – P. 161-188.

[Гуляницкий Л.Ф., 2014a] Гуляницкий Л.Ф., Павленко А.И. Прогнозирование на основе генетического алгоритма обучения. - Конференция KDS, 2014. – P. 41-42.

[Гуляницький Л. Ф., 2014b] Гуляницький Л. Ф., Павленко А.І. Налаштування параметрів моделі алгоритму прогнозування на основі генетичних алгоритмів. Теорія прийняття рішень. – 2014. – P. 90-93.

[Mahfoud S.W., 1995] Mahfoud S.W. Niching methods for genetic algorithms. – Dissertation Abstracts International. – University Microfilms, 1995. – 251p.

[Goldberg D. E., 1987] Goldberg D. E., Richardson J. Genetic algorithms with sharing for multimodal function optimization. – In Genetic algorithms and their applications: Proceedings of the second international conference on genetic algorithms, 1987. – P. 41-49.

[Hanke, 2008] John E. Hanke, Dean Wichern. Business Forecasting. – Prentice Hall, 9 edition, 2008. – 576p.

Authors' Information

Hulianytskyi Leonid *– Dr.Sc. (Technology), Head of department of Glushkov Institute of Cybernetics of NAS of Ukraine, Professor of NTUU "KPI" (Kyiv); e-mail: leonhul.icyb@gmail.com*

Major fields of scientific research: combinatorial optimization; decision making; mathematical modeling and applications; forecasting.

Pavlenko Anna *– PhD student at Glushkov Institute of Cybernetics of NAS of Ukraine (Kyiv) Glushkov Ave., 40, Kyiv, 03680, Ukraine; e-mail: dmitrieva.anya@gmail.com*

Major fields of scientific research: forecasting, genetic algorithms, artificial intelligence; combinatorial optimization; mathematical economics.