# ОПТИМИЗИРАНЕ НА БАЗА ОТ ПРАВИЛА С ГЕНЕТИЧНА РАЗМИТА СИСТЕМА

**Пенка В. Георгиева, Иван Попчев**
*Бургаски свободен университет*
penka.georgieva@bfu.bg, ipopchev@iit.bas.bg

# RULE BASE OPTIMIZATION WITH A GENETIC FUZZY SYSTEM

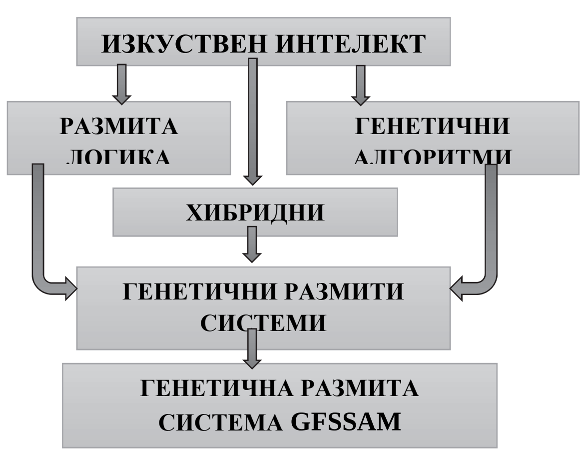
**Penka V. Georgieva, Ivan Popchev**
*Burgas Free University*

***Abstract:*** *This paper examines the concepts for designing and building a system that integrates the potential of fuzzy modeling with the capabilities of genetic algorithms in the process of finding optimal solutions. The rule-based genetic fuzzy system is a hybrid system that aims at optimizing the weight of the rules in the knowledge base of FSSAM. Some results from the optimization process are shown as an illustration of the system's capabilities.*

***Key words:*** *hybrid systems, genetic fuzzy system, artificial intelligence*

## I. ВЪВЕДЕНИЕ

**1.** Хибридните системи на изкуствения интелект се проектират и създават така, че да се използват предимствата на една или повече от изчислителните му парадигми за компенсиране недостатъците на друга. Хибридизацията на размити системи и генетични алгоритми се реализира в две посоки: 1) размитата система настройва параметрите на генетичния алгоритъм и избора на генетични оператори и 2) генетичните алгоритми се използват за настройване параметрите на размита система(фиг. 1). [1], [2]




***Фигура 1. Генетичните размити системи като подобласт на изкуствения интелект***

**2.** Основните идеи на теорията на размитите множества са предложени от Лотфи Заде през 1965. От строго теоретичните разработки се развиват различни направления в областите размити множества и размита логика: принцип на продължението, размити релации, размити графи, размито диференциране и интегриране, теория на възможностите, размита логика, приблизителни разсъждения, размити бази от данни, размит анализ на данни, размито линейно програмиране, размито динамично програмиране и др. [2]

В класическата математическа теория на множествата, понятието множество е съвкупност от обекти или елементи (числа, имена, цветове и т.н.), като даден обект $x$ или принадлежи или не принадлежи на множеството $A \subseteq U$, където $U$ е универсум. В първия случай твърдението „$x$ принадлежи на $A$“ е вярно, а във втория – невярно. Класическо множество може да бъде зададено по различни начини: чрез изброяване на елементите, аналитично или чрез характеристична функция $\chi_A(x)$ по следния начин:

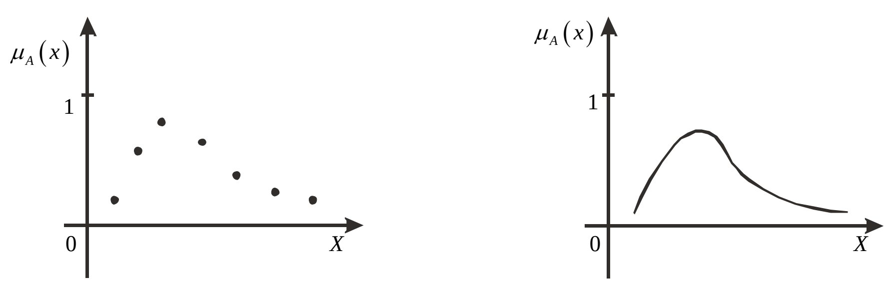
$$\chi_A(x) = \begin{cases} 1 & | \quad x \in A \\ 0 & | \quad x \notin A \end{cases}.$$

За разлика от класическото множество, при дефиниране на понятието размито множество се използва идеята, че обектите могат да принадлежат към дадено множество с различна степен на принадлежност, която се измерва с реални числа в интервала $[0;1]$.

Така, ако $U$ е универсум и $A \subseteq U$ е негово фиксирано подмножество, то множеството от наредени двойки

$$\Phi_A = \{< x\,;\, \mu_A(x) > \ | \ x \in U\}$$

се нарича размито множество върху $U$, ако функцията $\mu_A(x)$ съпоставя на всяко $x \in U$ точно едно реално число, принадлежащо на интервала $[0;1]$ (фиг. 2). Стойностите на $\mu_A(x)$ се наричат степени на принадлежност. [3]




***Фигура 2. Функция на принадлежност $\mu_A(x)$ - дискретна и непрекъсната***

Размитата логика използва лингвистични променливи, лингвистични модификатори, съждителна размита логика и правила за извод.

Лингвистична променлива се задава като наредена петорка ( $x, T(x), U, G, M$), където $x$ е име на променливата, $\boldsymbol{T(x)}$ е терм-множество, $U$ е универсум, $G$ е синтактично правило и $M$ е семантично правило.

Лингвистичните модификатори имат вида:

- $_{not}T$ , където $\mu_{_{not}T}(x) = 1 - \mu_T(x)$
- $T_1 \; and \; T_2$ , *където* $\mu_{T_1 \, and \, T_2}(x) = min \ \{\mu_{T_1}(x)\mu_{T_2}(x)\};$
- $T_1 \; or \; T_2$ , *където* $\mu_{T_1 \, or T_2}(x) = max \ \{\mu_{T_1}(x)\mu_{T_2}(x)\};$

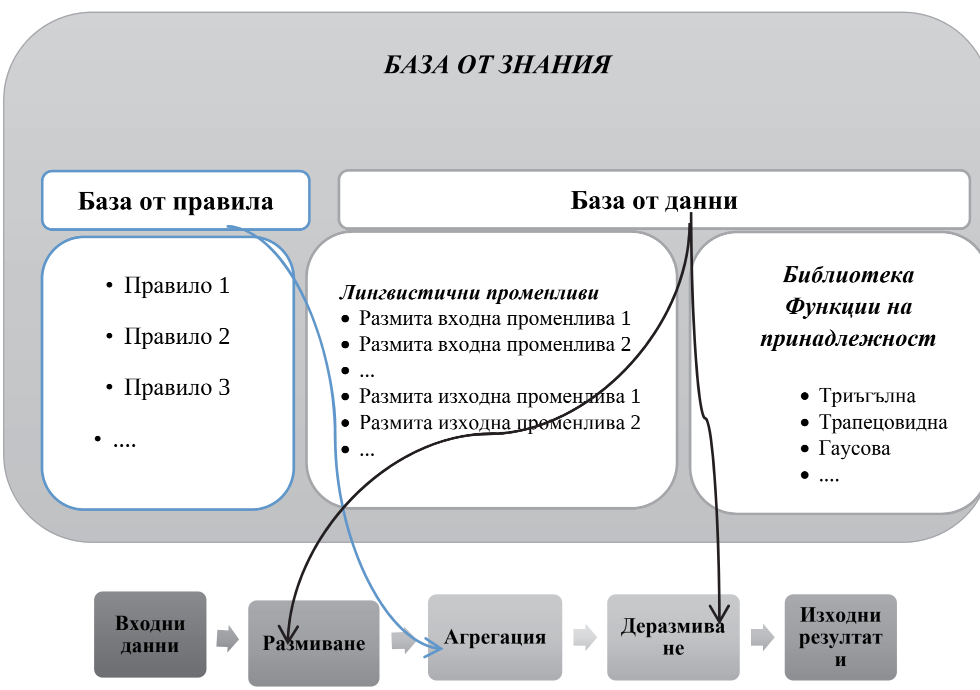
- ${}_{very}T$ , където $\mu_{{}_{very}T}(x)[\mu_T(x)]^2$;
- ${}_{nearly}T$ , където $\mu_{{}_{nearly}T}(x)[\mu_T(x)]^{1/2}$.

Размита система за изводи (*fuzzy inference system=FIS*) е изчислителна структура, използваща теорията на размитите множества, правила от вида *IF-THEN* и размитата логика. [4], [5], [6], [7]

Основната структура на размита система за изводи съдържа три концептуални компонента:

1) база от правила, включваща всички размити правила за вземане на решения;
2) база от данни (речник), съдържаща всички функции на принадлежност;
3) машина за изводи, която изпълнява процедурата за вземане на решения от правилата и дадените факти, за да се получи коректен изход или заключение.

Базата от правила и базата от данни формират базата от знания (фиг. 3). [8]




***Фигура 3. Размита система, базирана на правила***

**3.** Генетичният алгоритъм е адаптивен алгоритъм, зададен със следната наредена седморка от оператори и параметри:

$$GA = (PP, \mathcal{P}, \mathcal{F}, \mathcal{S}, \Omega, \Psi, \varsigma),$$

където $PP$ е *популация* с обем $\mathcal{P}$ с елементи *хромозомите* $c^j = \left(c_1^j, c_2^j, \dots, c_l^j\right) \in PP$, $j = 1,2, \dots, \mathcal{P}$, които са $l$ –мерни бинарни вектори; $\mathcal{F}$ е функция на $l$ променливи и с област от функционални стойности $\mathbb{R}^+$, наречена *функция на пригодност (целева функция):*

$$\mathcal{F}: c^j \to \mathbb{R}^+, \qquad j = 1,2, \dots, \mathcal{P};$$

$\mathcal{S}$ е *оператор за селекция*, след чието прилагането от популацията $PP$ се избират $u$ на брой *родители* $p^k$:

$$\mathcal{S}: PP \to \{p^1, p^2, \dots, p^u\} \ ;$$

$\Omega$ е множество от генетични оператори:

$$\Omega = \{\Omega_{Cross};\ \Omega_{Mut}; \dots\},$$

където $\Omega_{Cross}$ е *оператор за кръстосване, $\Omega_{Mut}$ е оператор за мутация,* които генерират $v$ на брой *деца* $q^m$ от родителите:

$$\Omega: \{p^1, p^2, \dots, p^u\} \to \{q^1, q^2, \dots, q^v\} \, ;$$

$\Psi$ е *оператор за заличаване* на $v$ на брой хромозоми от текущата популация, като следващата $i+1$-ва популация се получава по формулата:

$$PP(i+1) = PP(i) - \Psi\big(PP(i)\big) + \{q^1, q^2, \dots, q^v\};$$

$\varsigma$ е *критерий за край* на алгоритъма.

Операторите $\mathcal{S}$ и $\Omega$ са винаги вероятностни, а $\Psi$ може да е както вероятностен, така и детеминистичен.

Всеки генетичен алгоритъм се състои от последователно конструиране на популации

$$PP(0)\, , PP(1), \dots, PP(i), PP(i+1), \dots$$

като първоначалната популация $PP(0)$ се избира случайно, преходът от популацията $PP(i)$ към популацията $PP(i+1)$ се осъществява чрез генетичните оператори $\mathcal{S}$, $\Omega$ и $\Psi$. Този преход се нарича *генерация* и е аналог на итерация при числените методи.

Генетичният алгоритъм може да бъде описан и като процедура за решаване на оптимизационна задача от вида:

$$max\{\ F(\boldsymbol{c}) \quad | \quad \boldsymbol{c} \in \{0,1\}^l\ \} \tag{1}$$

или

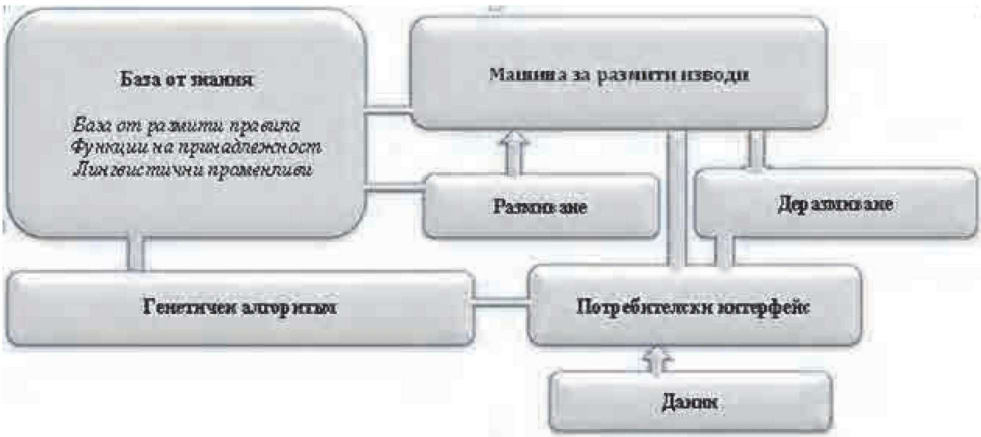
$$min\{\ F(\boldsymbol{c}) \quad | \quad \boldsymbol{c} \in \{0,1\}^l\ \}, \tag{2}$$

където $F$ е целева функция и $\boldsymbol{c}$ е индивид, който е възможно решение на (1) (или (2)).

## II. АРХИТЕКТУРА НА GFSSAM

Целта, която се постига при проектирането и реализирането на хибридната генетичната размита система *GFSSAM=Genetic Fuzzy Software System for Asset Management*, е да се използва генетичен алгоритъм, за да се настроят параметрите на базата от знания на вече създадената размита система FSSAM (фиг. 4).

В системата FSSAM базата от знания се състои от базата с функциите на принадлежност на терм-множествата на входните и изходни размити променливи, както и от базата от правила [9]. Приложенията, за които хибридната система е проектирана за настройване параметрите на функциите на принадлежност чрез търсене на оптимални стойности, без да променя техния вид и без да променя базата от правила, са описани в [10].




***Фигура 4. Генетична размита система за настройване на базата от знания***

## III. ОПТИМИЗИРАНЕ НА БАЗАТА ОТ ПРАВИЛА

В това изследване генетичният алгоритъм е използван за оптимизиране на базата от правила, като параметрите на функциите на принадлежност на входните и изходните променливи се запазват.

Всяко правило в размитата система има вида:

*IF*

$$\left\{K_{m_1}\ is\ \ X_{m_1 j_{m_1}}\right\}\ AND\ \left\{K_{m_2}\ is\ \ X_{m_2 j_{m_2}}\right\} AND \dots AND\ \left\{K_{m_k}\ is\ \ X_{m_k j_{m_k}}\right\}$$

*THEN*

$$\left\{Q_{m_1}\ is\ \ Y_{m_1 j_{m_1}}\right\} AND\ \left\{Q_{m_2}\ is\ \ Y_{m_2 j_{m_2}}\right\} AND \dots AND \left\{Q_{m_l}\ \ is\, Y_{m_l j_{m_l}}\right\},$$

където $m_i = 1, 2, \dots, M$ и $M$ е броят на правилата.

След избора и изпълнението $m$-тото правило системата FSSAM използва $min$ оператор за пресмятане на величината $\Theta_m$ по формулата:

$$\Theta_m = \min\left\{\mu_{m_1 j_{m_1}}(x_1^*), \qquad \mu_{m_2 j_{m_2}}(x_2^*), \dots, \mu_{m_k j_{m_k}}(x_k^*)\right\}.$$

Всяко правило има тегло $w_m$ за $m = 1, 2, \dots, M$ и след умножение на получената стойност на $\Theta_m$ със съответното тегло се получава претеглената стойност:

$${\Theta_m}^o = \Theta_m . w_m .$$

След изпълнението на всички правила от базата от правила, за всяко терм-множество $Y_{sp}$ на изходните променливи са получени съответните степени на принадлежност ${\mu_{sp}}^m = {\Theta_m}^o$.

Агрегацията се получава след пресмятането на

$$P_{sp} = max \ \{ \mu_{s\,p}{}^{1},\ \mu_{s\,p}{}^{2}, \dots, \mu_{s\,p}{}^{M} \}$$

за всяко $Y_{sp}$, $s = 1, 2, 3, \dots, S$ и $p = 1, 2, 3, \dots, p_s$ .

Последна процедура е деразмиване на $Y_{sp}$за получаване на изходната промен-лива $Q$.

В системата FSSAM са реализирани 24 правила със съответните 24 тегла $w_m$ за $m = 1, 2, \dots, 24$. Например шестото правило с тегло $w_m = 0{,}8$ е следното:

**<u>Правило 6</u>: Ако *Възвращаемост*та е много висока и *Риск*ът е неутрален и *q* е голямо, то *Q* е добро.**

Изборът на теглата е направен като е използвано експертно мнение и така в системата са заложени началните стойности, показани в таблица 1.

В тази разработка генетичният алгоритъм търси стойности на индивидите ***c*, *при*** които се достига минимум на целевата функция:

$$F = (Q_1 - Q_2)^2 + \ (Q_1 - Q_3)^2 + (Q_2 - Q_3)^2,$$

където $Q_1, Q_2, Q_3$ са три стойности на изходната променлива $Q$, пресметнати през равни интервали от време. Мотивацията за избор на целевата функция е публикувана в [10] I [11].

Индивидите ***c*** на популацията са 24-мерни вектори:
***c*** $=(w_1;\quad w_2;\quad w_3;\ \cdots\quad w_{23};\quad w_{24})$.

## IV. РЕЗУЛТАТИ

За целите на изследването са проведени редица тестове, като са променяни стойностите на параметрите *nGen (брой генерации)*, *nPop обем на популацията)*, *Pc (вероятност за кръстосване)*, *Pm (вероятност за мутация)* на генетичния алгоритъм.

В таблица 1 са показани резултатите от четири от тестовете.

***Таблица 1. Резултати от генетична размита система при оптимизация на теглата на правилата***

| ***nGen*** | ***nPop*** | ***Pc*** | ***Pm*** | ***w 1*** | ***w 2*** | ***w 3*** | ***w 4*** | ***w 5*** | ***w 6*** | ***w 7*** |
|---|---|---|---|---|---|---|---|---|---|---|
| ***начални стойности*** | | | | ***1*** | ***1*** | ***1*** | ***1*** | ***0,8*** | ***0,8*** | ***0,8*** |
| **10** | **20** | **0,75** | **0,2** | 0,976 654 | 0,820 508 | 0,766 591 | 0,868 349 | 0,717 322 | 0,656 674 | 0,130 368 |
| **20** | **20** | **0,5** | **0,05** | 0,745 752 | 0,907 513 | 0,672 978 | 0,872 377 | 0,359 415 | 0,772 108 | 0,074 125 |
| **100** | **8** | **0,2** | **0,02** | 0,817 967 | 0,713 620 | 0,802 406 | 0,746 703 | 0,554 740 | 0,537 281 | 0,324 936 |
| **500** | **8** | **0,05** | **0,5** | 0,9614 73 | 0,8275 03 | 0,8546 06 | 0,8861 66 | 0,7373 83 | 0,4600 31 | 0,2446 58 |

| *w 8* | *w 9* | *w 10* | *w 11* | *w 12* | *w 13* | *w 14* | *w 15* | *w 16* |
|---|---|---|---|---|---|---|---|---|
| *0,8* | *0,8* | *0,8* | *0,7* | *0,7* | *0,7* | *0,7* | *1* | *1* |
| 0,315154 | 0,114078 | 0,872515 | 0,963561 | 0,820828 | 0,769732 | 0,86348 | 0,773251 | 0,750632 |
| 0,766436 | 0,064852 | 0,843793 | 0,842725 | 0,901302 | 0,377857 | 0,87272 | 0,975418 | 0,410403 |
| 0,938790 | 0,284316 | 0,708947 | 0,896706 | 0,716036 | 0,8006659 | 0,76704 | 0,854374 | 0,278267 |
| 0,817248 | 0,564073 | 0,990088 | 0,673013 | 0,947549 | 0,846327 | 0,86167 | 0,833842 | 0,360323 |

| *w 17* | *w 18* | *w 19* | *w 20* | *w 21* | *w 22* | *w 23* | *w 24* | | *min F* |
|---|---|---|---|---|---|---|---|---|---|
| *1* | *1* | *0,8* | *0,8* | *0,8* | *0,8* | *0,8* | *0,8* | | |
| 0,826624 | 0,605528 | 0,114072 | 0,272518 | 0,630360 | 0,711492 | 0,862963 | 0,889126 | | **0,00074098** |
| 0,553412 | 0,981641 | 0,064855 | 0,143789 | 0,774119 | 0,864329 | 0,692651 | 0,905124 | | **0,00063489** |
| 0,692999 | 0,829301 | 0,284319 | 0,208943 | 0,924936 | 0,838792 | 0,906175 | 0,794910 | | **0,00003211** |
| 0,828457 | 0,939215 | 0,564076 | 0,190088 | 0,844658 | 0,717243 | 0,905238 | 0,767541 | | **0,00000069** |

Получените резултати показват, че при такава архитектура на размитата система няма оптимално тегло със стойност 1, т.е. нито едно правило не е абсолютно важно. Значителна част от теглата $w_m$, за $m = 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 21, 22, 23, 24$, имат стойности след оптимизацията близки до началните. Но има и правила, за които получените тегла са от различен порядък. Такива са:

**Правило 6:** Ако *Възвращаемост*та е много висока и *Риск*ът е неутрален и $q$ е голямо, то $Q$ е добро;

**Правило 7:** Ако *Възвращаемост*та е висока и *Риск*ът е малък и $q$ е голямо, то $Q$ е добро;

**Правило 16:** Ако *Възвращаемост*та е много ниска и *Риск*ът е голям и $q$ е малко, то $Q$ е лошо;

**Правило 20:** Ако *Възвращаемост*та е много ниска и *Риск*ът е неутрален и $q$ е малко, то $Q$ е донякъде лошо.

При условие, че следващите тестове покажат подобни резултати, то тези правила или ще бъдат изключени от базата с правила и така в нея ще има по-малък брой правила или те ще бъдат запазени, но теглата им ще бъдат променени.

Освен тези четири правила има и други две, за които резултатите са противоречиви:

**Правило 9:** Ако *Възвращаемост*та е много висока и *Риск*ът е много малък и $q$ е неутрално, то $Q$ е добро;

**Правило 19:** Ако *Възвращаемост*та е много ниска и *Риск*ът е много голям и $q$ е малко, то $Q$ е донякъде лошо.

За тези резултати са възможни две причини – или правилата противоречат на други правила от базата или се припокриват с други правила. За изясняване на първоизточника на тези резултати също са необходими допълнителни тестове и резултати.

## V. ЗАКЛЮЧЕНИЕ

GFSSAM е генетична размита система, базирана на правила и реализирана в средата MatLab. Основната цел при създаването на системата е намирането на оптимални стойности на параметрите на размитата система.

Хибридизацията на генетичния алгоритъм и размитата система е успешна, защото независимо че оптималните тегла на няколко от правилата са от различен порядък с началните стойности, стойностите на целевата функция са достатъчно близки до абсолютния й минимум 0.

Важен резултат от това изследване е използването на генетична размита система за оптимизиране на архитектурата на размитата система и по-специално броя и вида на размитите правила.

**Литература:**


[1] Herrera, F., M. Lozano, E. Herrera-Viedma, J. Verdegay. Fuzzy tools to improve genetic algorithms. Proc. of the European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, 1994, pp. 1532-1539.

[2] Георгиева П., Генетични размити системи, Полиграф, 2016.

[3] Георгиева П., Изследване на модели на софт компютинг за управление в реално време, Академик Дринов, София, 2013.

[4] Georgieva P. V., FSSAM: A Fuzzy Rule-Based System for Financial Decision Making in Real Time. - Handbook of Fuzzy Sets Comparison - Theory, Algorithms and Applications, Science Gate Publishing, 2016, pp. 121-148.

[5] Jang R., Fuzzy inference systems. NJ: Prentice-Hall, 1997.

[6] Georgieva P. V., Fuzzy Rule-based Systems for Decision-making. Engineering Sciences, BAS, Vol. LIII, 2016, No 1, pp. 5-16.

[7] Zafari A., Developing a fuzzy inference system by using genetic algorithm and expert knowledge. Netherlands: Enschede, 2014.

[8] Popchev, I., P. Georgieva, A Fuzzy Approach for Solving Multicriteria Investment Problems. –In: Iskander M. (eds) Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education. Springer Science+Business Media B.V., 2008, pp. 427-431.

[9] Melin P., O. Castillo, E. Ramírez, „Analysis and Design of Intelligent Systems Using Soft Computing Techniques,“ Series: Advances in Soft Computing, Vol. 41, 2007

[10] Georgieva P. V., I. Popchev, St. Stoyanov. A Multi-Step Procedure for Asset Allocation in Case of Limited Resources. – CIT BAS, Vol. 15, no. 3, 2015, pp. 41–51

[11] Georgieva Penka V., Genetic Fuzzy System for Financial Management. CIT, BAS, Sofia 2018.

[12] Georgieva P. V., Applying FSSAM for Currency Rates Forecasting. –In: Transactions on Machine Learning and Artificial Intelligence, Manchester, SSE UK, Vol. 4, 2016, no. 3, pp. 30-40