

RULE BASE OPTIMIZATION WITH A GENETIC FUZZY SYSTEM

*Penka V. Georgieva, Burgas Free University, pgeorg@bfu.bg
Ivan Popchev, Burgas Free University, ipopchev@iit.bas.bg*

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.

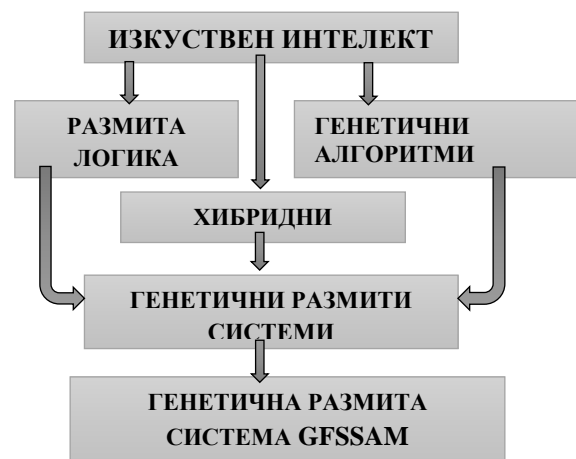
Keywords: hybrid systems, genetic fuzzy system, artificial intelligence

ОПТИМИЗИРАНЕ НА БАЗА ОТ ПРАВИЛА С ГЕНЕТИЧНА РАЗМИТА СИСТЕМА

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

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 размитата система (фиг. 1). [1], [2]



Фиг. 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:

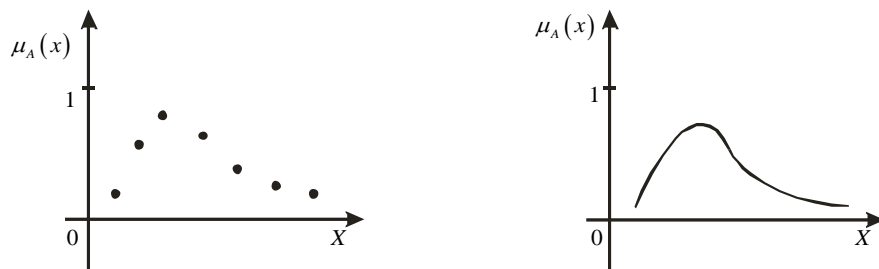
$$\chi_A(x) = \begin{cases} 1 & | \ x \in A \\ 0 & | \ x \notin A \end{cases}$$

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

$$\Phi_A = \{ \langle x; \mu_A(x) \rangle \mid x \in U \}$$

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]



Фиг. 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, $T(x)$ е терм-множество, U е универсум, G е синтактично правило и M е семантично правило.

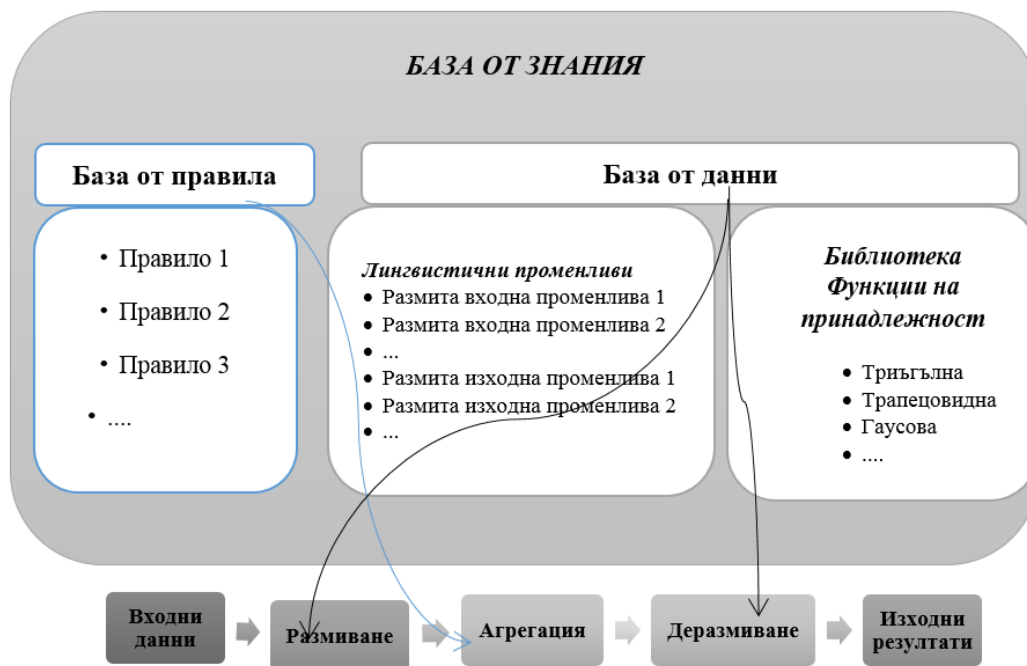
Лингвистичните модификатори имат вида:

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



Фиг. 3. Размита система, базирана на правила

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

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

където PP е популация с обем \mathcal{P} с елементи хромозомите $c^j = (c_1^j, c_2^j, \dots, c_l^j) \in PP$, $j = 1, 2, \dots, \mathcal{P}$, 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):

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

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

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

Ω е множество от генетични оператори:

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

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

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

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

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

ζ е критерий за край на алгоритъма.

Операторите \mathcal{S} и Ω са винаги вероятностни, а Ψ 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} , Ω и Ψ . 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:

$$\max\{ F(\mathbf{c}) \mid \mathbf{c} \in \{0,1\}^l \} \quad (1)$$

или

$$\min\{ F(\mathbf{c}) \mid \mathbf{c} \in \{0,1\}^l \}, \quad (2)$$

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

II. АРХИТЕКТУРА НА GFSSAM

Целта, която се постига при проектирането и реализирането на хибридна генетичната размита система *GFSSAM=Genetic Fuzzy Software System for Asset Management*, е да се използва генетичен алгоритъм, за да се настройят 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 хибридна 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].



Фиг. 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:

$$\begin{array}{l} \text{IF} \\ \{K_{m_1} \text{ is } X_{m_1j_{m_1}}\} \text{ AND } \{K_{m_2} \text{ is } X_{m_2j_{m_2}}\} \text{ AND } \dots \text{ AND } \{K_{m_k} \text{ is } X_{m_kj_{m_k}}\} \\ \text{THEN} \\ \{Q_{m_1} \text{ is } Y_{m_1j_{m_1}}\} \text{ AND } \{Q_{m_2} \text{ is } Y_{m_2j_{m_2}}\} \text{ AND } \dots \text{ AND } \{Q_{m_l} \text{ is } Y_{m_lj_{m_l}}\}, \end{array}$$

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 θ_m по формулата:

$$\theta_m = \min \{ \mu_{m_1j_{m_1}}(x_1^*), \mu_{m_2j_{m_2}}(x_2^*), \dots, \mu_{m_kj_{m_k}}(x_k^*) \}.$$

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

$$\theta_m^o = \theta_m \cdot w_m.$$

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

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

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 променлива 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:

Правило 6: Ако *Възвращаемостта* е много висока и *Рискът* 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 заложили 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:

$$c = (w_1; w_2; w_3; \dots; w_{23}; w_{24}).$$

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.

Gen	Pop	c	t	1	2	3	4	5	6	7
начални стойности				1	1	1	1	,8 0	,8 0	,8 0
0	0	,75	,2	,976654	,820508	,766591	,868349	,717322	,656674	,130368
0	0	,5	,05	,745752	,907513	,672978	,872377	,359415	,772108	,074125
00		,2	,02	,817967	,713620	,802406	,746703	,554740	,537281	,324936
00		,05	,5	,961473	,827503	,854606	,886166	,737383	,460031	,244658

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

17	18	19	20	21	22	23	24	n F	mi
1	1	,8 0	,8 0	,8 0	,8 0	,8 0	,8 0		
,826624	,605528	,114072	,272518	,630360	,711492	,862963	,889126		0,0 0074098
,553412	,981641	,064855	,143789	,774119	,864329	,692651	,905124		0,0 0063489
,692999	,829301	,284319	,208943	,924936	,838792	,906175	,794910		0,0 0003211
,828457	,939215	,564076	,190088	,844658	,717243	,905238	,767541		0,0 0000069

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

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 *Възвращаемостта* е много висока и *Рискът* е неутрален и q е голямо, то Q е добро;

Правило 7 : Ако *Възвращаемостта* е висока и *Рискът* е малък и q е голямо, то Q е добро;

Правило 16 : Ако *Възвращаемостта* е много ниска и *Рискът* е голям и q е малко, то Q е лошо;

Правило 20 : Ако *Възвращаемостта* е много ниска и *Рискът* 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 *Възвращаемостта* е много висока и *Рискът* е много малък и q е неутрално, то Q е добро;

Правило 19 : Ако *Възвращаемостта* е много ниска и *Рискът* 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 ѝ 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.

References

[1] HERRERA, F., M. LOZANO, HERRERA-VIEDMA, E., VERDEGAY, J.. 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., Georgieva, P.. 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., CASTILLO, O., RAMÍREZ, E.. "Analysis and Design of Intelligent Systems Using Soft Computing Techniques," Series: Advances in Soft Computing, Vol. 41, 2007.
- [10] GEORGIEVA, P. V., POPCHEV, I., STOYANOV., ST.. A Multi-Step Procedure for Asset Allocation in Case of Limited Resources. – CIT BAS, Vol. 15, no. 3, 2015, pp. 41–51
- [11] GEORGIEVA, P. 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.