

## АДАПТИВНЫЙ МЕТОД ГИБРИДНОГО ОБУЧЕНИЯ ЭВОЛЮЦИОНИРУЮЩЕЙ НЕЙРО-ФАЗЗИ СИСТЕМЫ

**Аннотация.** Предложен метод обучения нейро-фаззи систем, который включает процессы эволюции архитектуры, самообучения функций принадлежности и обучения синаптических весов. Этот подход обеспечивает высокое быстродействие и возможность обрабатывать информацию в on line режиме, адаптируя при этом параметры и структуру системы к условиям задачи.

**Ключевые слова:** нейро-фаззи система, гибридное обучение, эволюционирующая система, вычислительный интеллект, эволюция архитектуры.

### ВВЕДЕНИЕ

В настоящее время нейро-фаззи системы (NFS) вычислительного интеллекта [1–4] вследствие универсальных аппроксимирующих возможностей, способности обучаться на основании данных, характеризующих функционирование исследуемого явления или объекта, а также лингвистической интерпретации получаемых результатов широко применяются для решения различных задач Data Mining, интеллектуального управления, прогнозирования, идентификации, эмуляции и т.п. в условиях неопределенности, нелинейности, стохастичности и хаотичности, разнообразных возмущений и помех.

В случае необходимости обработки информации в адаптивном on line режиме прежде всего возникает проблема скорости сходимости процесса обучения, существенно ограничивающая количество NFS, пригодных для работы в этих условиях. В связи с этим наиболее эффективны ANFIS- и TSK-системы [1, 2, 5], выходной сигнал которых линейно зависит от настраиваемых параметров (синаптических весов), что позволяет использовать для их обучения множество известных линейных алгоритмов обучения [6], методов адаптивной идентификации [7, 8], имеющих высокое быстродействие.

Вместе с тем, чтобы избежать возникновения «дыр» в пространстве входных сигналов в результате рассеянного разбиения (scatter partitioning) [9], используемого в ANFIS- и TSK-системах, необходимо определять и настраивать параметры функций принадлежности в первом скрытом слое NFS, количество которых обычно выбирается из сугубо эмпирических предположений. Для настройки параметров функций принадлежности, как правило, используется алгоритм обратного распространения ошибок, реализуемый на множестве эпох [6], так что рассматривать обучение в on line режиме в данном случае не будем.

В связи с этим целесообразна разработка метода адаптивной настройки параметров функций принадлежности, а также определение их количества непосредственно в процессе обучения NFS по мере последовательного поступления на ее вход данных обучающей выборки. Для решения этой задачи можно воспользоваться методами теории эволюционирующих систем вычислительного интеллекта [10–14], сочетая при этом парадигмы контролируемого обучения (с учителем) и самообучения (без учителя).

### АРХИТЕКТУРА ЭВОЛЮЦИОНИРУЮЩЕЙ НЕЙРО-ФАЗЗИ СИСТЕМЫ

Архитектура рассматриваемой системы, приведенная на рис. 1, состоит из пяти последовательных слоев.

На входной слой NFS подается  $(n \times 1)$ -мерный вектор подлежащих обработке сигналов-образов  $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ , где  $k = 1, 2, \dots$  — текущее дискретное время. Первый скрытый слой содержит  $h$  узлов (см. рис. 1) и  $nh$  функций

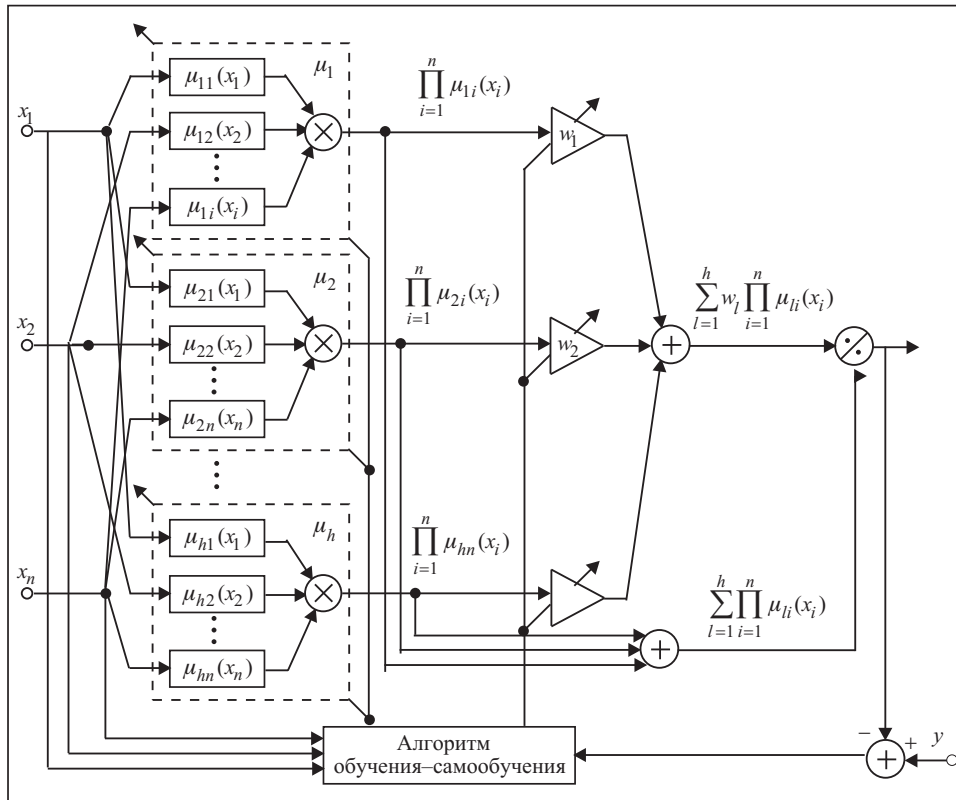


Рис. 1. Архитектура эволюционирующей нейро-фаззи системы

принадлежности  $\mu_{li}(x_i)$ ,  $i=1, 2, \dots, n$ ,  $l=1, 2, \dots, h$  и выполняет фаззификацию входного пространства. При этом подразумевается, что в процессе обучения–эволюции настраиваются параметры этих функций и определяется их число.

Второй скрытый слой обеспечивает агрегирование уровней принадлежности, рассчитанных в первом скрытом слое, и состоит из  $h$  блоков умножения. Третий скрытый слой содержит настраиваемые синаптические веса  $w_1, w_2, \dots, w_h$ , определяемые в процессе контролируемого обучения. Четвертый скрытый слой образован двумя сумматорами и вычисляет суммы выходных сигналов второго и третьего скрытых слоев. И наконец, в пятом (выходном) скрытом слое проводится дефаззификация, в результате которой вычисляется выходной сигнал NFS  $\hat{y}(k)$ .

Таким образом, если на вход NFS подан векторный сигнал  $x(k)$ , элементы первого скрытого слоя проводят его фаззификацию, вычисляя уровни принадлежности  $0 < \mu_{li}(x_i(k)) \leq 1$ . Обычно в качестве функций принадлежности используются традиционные гауссианы

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right), \quad (1)$$

где  $c_{li}$ ,  $\sigma_i$  — параметры центров и ширины соответственно. Заметим также, что предварительное нормирование данных на некоторый интервал, например  $-1 \leq x_i(k) \leq 1$ , позволяет упростить расчеты, поскольку параметры ширины  $\sigma_i$  можно принять одинаковыми для всех входов, т.е.  $\sigma_i = \sigma$ .

Кроме гауссианов (1) можно использовать и иные ядерные функции, например  $B$ -сплайны, соответствующие условию единичного разбиения, а также четные вейвлеты, гибкие функции активации–принадлежности [15, 16] и т.д.

Во втором скрытом слое вычисляются агрегированные значения  $\prod_{i=1}^n \mu_{li}(x_i(k))$ , при этом для гауссианов с одинаковыми параметрами ширины

$$\prod_{i=1}^n \mu_{li}(x_i(k)) = \prod_{i=1}^n \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right),$$

где  $c_l = (c_{l1}, c_{l2}, \dots, c_{ln})^T$ , т.е. сигналы на выходах блоков умножения второго скрытого слоя аналогичны сигналам на выходах нейронов первого скрытого слоя обычных сетей радиально-базисных функций (Radial Basis Function Network, RBFN) [6].

Выходами третьего скрытого слоя есть значения  $w_l \prod_{i=1}^n \mu_{li}(x_i(k))$ , четвертого — значения  $\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))$  и  $\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))$  и, наконец, на выходе системы (пятого выходного слоя) появляется сигнал

$$\begin{aligned} \hat{y}(x(k)) &= \frac{\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \sum_{l=1}^h w_l \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \\ &= \sum_{l=1}^h w_l \varphi_l(x(k)) = w^{hT} \varphi^h(x(k)), \end{aligned} \quad (2)$$

где

$$\begin{aligned} \varphi_l(x(k)) &= \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))}, \quad w^h = (w_1, w_2, \dots, w_h)^T, \\ \varphi^h(x(k)) &= (\varphi_1(x(k)), \varphi_2(x(k)), \dots, \varphi_h(x(k)))^T. \end{aligned}$$

Несложно заметить, что рассматриваемая система реализует нелинейное отображение входного пространства в скалярный выходной сигнал ( $R^n \rightarrow R^1$ ) подобно нормализованной RBFN [8], а ее архитектура (при фиксированном  $h$ ) совпадает с TSK-системой нулевого порядка, т.е. с архитектурой Ванга–Менделя [17].

#### САМООБУЧЕНИЕ–ЭВОЛЮЦИЯ ПЕРВОГО СКРЫТОГО СЛОЯ

Процесс настройки параметров и увеличение числа функций принадлежности происходит следующим образом. Пусть на вход системы, у которой в исходном состоянии в первом скрытом слое не имеется функций принадлежности, поступает первый сигнал обучающей выборки  $x(1) = (x_1(1), x_2(1), \dots, x_n(1))^T$ . Данное наблюдение формирует первый узел первого скрытого слоя  $\mu_1$  так, что  $c_{1i}(1) = x_i(1)$ . Таким образом создается  $n$  функций принадлежности и формируется единственный синаптический вес  $w_1(0)$ , задаваемый случайно в интервале  $-1 \leq w_1(0) \leq 1$ . Далее для этой функции принадлежности  $\mu_1$  задается радиус соседства  $r$ , определяемый максимально возможным числом  $h$  функций принадлежности в NFS. Так, если функции принадлежности по осям распределены равномерно, то  $r = \frac{2}{h-1}$ .

Далее при поступлении второго векторного сигнала  $x(2)$  проверяется условие

$$\max_i |c_{1i} - x_i(2)| \leq r, \quad (2)$$

и, если оно выполняется, проводится коррекция центров функций принадлежности узла  $\mu_1$  согласно правилу  $c_{1i}(2) = c_{1i}(1) + \eta(2)(x_i(2) - c_{1i}(1))$ , где  $\eta(2)$  — параметр шага обучения, т.е. при  $\eta(2) = 0,5$  имеем

$$c_{1i}(2) = \frac{c_{1i}(1) + x_i(2)}{2}.$$

Если условие (2) не выполняется, формируется второй узел  $\mu_2$  функций принадлежности первого скрытого слоя, центры которых  $c_{2i}(2) = x_i(2)$ . Одновременно с узлом  $\mu_2$  в NFS вводится второй синаптический вес  $w_2$ , также задаваемый случайным образом.

Пусть к моменту поступления на вход NFS векторного сигнала  $x(k)$  сформировано  $p$  узлов функций принадлежности  $\mu_1, \mu_2, \dots, \mu_p$ ,  $p < h$ , с центрами  $c_{li}(k-1)$ ,  $l=1, 2, \dots, p$ ,  $i=1, 2, \dots, n$ , при этом проверяется условие

$$\max_i |c_{li} - x_i(k)| \leq r \quad \forall l=1, 2, \dots, p, \quad (3)$$

и, если оно выполняется, проводится коррекция центров функций принадлежности, ближайших к соответствующим компонентам  $x_i(k)$  согласно правилу

$$c_{li}(k) = c_{li}(k-1) + \eta(k)(x_i(k) - c_{li}(k-1)). \quad (4)$$

Несложно заметить, что правило (4) весьма близко к правилу самообучения Т. Кохонена: «Победитель получает все» [18], которое реализуется на гиперсфере  $\|x(k)\|_2 = 1$ , тогда как правило (4) — на гиперкубе  $\|x(k)\|_\infty = 1$ .

Если условие (3) не выполняется, в системе формируется  $(p+1)$ -й,  $p+1 \leq h$ , узел  $\mu_{p+1}$  с центрами функций принадлежности  $c_{p+1,i}(k) = x_i(k)$ . Одновременно с узлом  $\mu_{p+1}$  формируется синаптический вес  $w_{p+1}$ .

Данная процедура есть гибрид эволюционирующего алгоритма Н. Касабова [13] и самообучающейся карты Т. Кохонена [18], при этом процесс эволюции архитектуры и самообучения функций принадлежности может происходить как непрерывно, так и до достижения предельного значения  $nh$  числа функций принадлежности.

#### АДАПТИВНОЕ ОБУЧЕНИЕ СИНАПТИЧЕСКИХ ВЕСОВ

Как уже отмечалось, для настройки синаптических весов NFS можно использовать известные алгоритмы обучения-идентификации, например:

— экспоненциально-взвешенный рекуррентный метод наименьших квадратов

$$\left\{ \begin{aligned} w^h(k) &= w^h(k-1) + \frac{P^h(k-1)(y(k) - w^{hT}(k-1)\varphi^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)) = \\ &= w^h(k-1) + \frac{P^h(k-1)(y(k) - \hat{y}^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)), \\ P^h(k) &= \frac{1}{\beta} \left( P^h(k-1) - \frac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \right) = \\ &= \left( \sum_{\tau=1}^k \varphi^h(x(\tau))\varphi^{hT}(x(\tau)) \right)^{-1}, \quad 0 < \beta \leq 1, \end{aligned} \right. \quad (5)$$

где  $y(k)$  — внешний обучающий сигнал,  $\beta$  — параметр забывания устаревшей информации;

— оптимальный по быстродействию градиентный одношаговый алгоритм Качмажа–Уидроу–Хоффа

$$w^h(k) = w^h(k-1) + \frac{y(k) - w^{hT}(k-1)\varphi^h(x(k))}{\|\varphi^h(x(k))\|^2} \varphi^h(x(k)); \quad (6)$$

— алгоритм обучения, имеющий следящие и сглаживающие свойства [19, 20],

$$\begin{cases} w^h(k) = w^h(k-1) + (\beta^h(k))^{-1} (y(k) - w^{hT}(k-1)\varphi^h(x(k)))\varphi^h(x(k)), \\ \beta^h(k) = \beta\beta^h(k-1) + \|\varphi^h(x(k))\|^2, \quad 0 \leq \beta \leq 1. \end{cases} \quad (7)$$

Отметим, что алгоритмы (5) и (7) взаимосвязаны соотношением  $\beta^h(k) = \text{Tr } P^h(k)$ , а при  $\beta = 0$  алгоритм (7) подобен алгоритму (6).

Процесс настройки синаптических весов может проходить одновременно с самообучением–эволюцией первого скрытого слоя.

Пусть к моменту поступления векторного сигнала  $x(k)$  сформировано  $p$  узлов функций принадлежности  $\mu_1, \mu_2, \dots, \mu_p$  и рассчитан вектор синаптических весов  $w^p(k-1)$ . Пусть условие (3) не выполняется, что приводит к формированию узла  $\mu_{p+1}$  и заданию произвольного начального значения синаптического веса  $w_{p+1}$ . При этом выходной сигнал NFS можно представить в виде

$$\hat{y}^{p+1}(x(k)) = (w^{p+1}(k-1))^T \varphi^{p+1}(x(k)) = w^{pT}(k-1)\varphi^p(x(k)) + w_{p+1}\varphi_{p+1}(x(k)),$$

а алгоритм (7) — следующим образом:

$$\begin{cases} w^{p+1}(k) = \begin{pmatrix} w^p(k-1) \\ \text{-----} \\ w_{p+1} \end{pmatrix} + (\beta^{p+1}(k))^{-1} (y(k) - \hat{y}^{p+1}(x(k))) \begin{pmatrix} \varphi^p(x(k)) \\ \text{-----} \\ \varphi_{p+1}(x(k)) \end{pmatrix}, \\ \beta^{p+1}(k) = \beta\beta^p(k-1) + \|\varphi^p(x(k))\|^2 + \varphi_{p+1}^2(x(k)). \end{cases}$$

Как можно видеть, процесс одновременных эволюции–самообучения–контролируемого обучения не вызывает вычислительных проблем.

#### РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Решение задачи эмуляции динамического объекта [21], описанного уравнением

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)), \quad (8)$$

подтверждает эффективность предложенного подхода.

Для эмуляции объекта использовалась нейро-фаззи система с количеством входов  $n = 5$ . Обучение системы проводилось на основе выборки, полученной с помощью уравнения (8), с управляющими сигналами  $f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}$  и  $u(k) = \sin(2\pi k / 250)$  для  $k < 500$ . В качестве проверочных данных эмулятора использовался динамический объект (8) с аналогичными управляющими сигналами для  $k \leq 500$ , а далее для  $k > 500$

вид управляющих сигналов менялся на  $u(k) = 0,8 \sin(2\pi k / 250) + 0,2 \sin(2\pi k / 25)$ .

Критерием качества прогноза являлась среднеквадратичная ошибка. Результаты эмуляции динамического объекта (8) приведены на рис. 2. Как видно, две кривые, представляющие фактические значения (сплошная линия) и значения эмуляции (штриховая линия), практически совпадают.

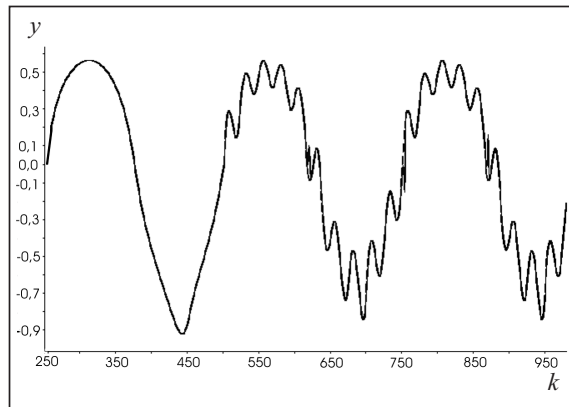


Рис. 2. Фрагмент результатов эмуляции динамического объекта

Данный эксперимент проводился для нескольких значений  $h$ . Так, средне-квадратичная ошибка составила 0,00145 при  $h = 5$ ; 0,00122 при  $h = 10$ ; 0,00121 при  $h = 20$ . Как следует из приведенных экспериментальных результатов, предложенная архитектура нейро-фаззи системы обеспечивает высокую точность эмуляции.

#### ВЫВОДЫ

Описан метод гибридного обучения многослойных нейро-фаззи систем, включающий процессы эволюции архитектуры, самообучения функций принадлежности и контролируемого обучения синаптических весов. Основными преимуществами предлагаемого подхода к обучению являются простота вычисления, высокое быстродействие, использование в адаптивном on line режиме, обеспечение высоких аппроксимирующих возможностей наряду со следящими и фильтрующими свойствами.

#### СПИСОК ЛИТЕРАТУРЫ

1. Jang J.-S., Sun C.-T., Mizutani E. Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. — Upper Saddle River: Prentice Hall, 1997. — 640 p.
2. Rutkowski L. Computational intelligence. Methods and techniques. — Berlin; Heidelberg: Springer-Verlag, 2008. — 514 p.
3. Kruse R., Borgelt C., Klawonn F., Moewes C., Steinbrecher M., Held P. Computational intelligence. — Berlin: Springer, 2013. — 488 p.
4. Du K.-L., Swamy M.N.S. Neural networks and statistical learning. — London: Springer-Verlag, 2014. — 824 p.
5. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control // IEEE Trans. on Systems, Man, and Cybernetics. — 1985. — **15**. — P. 116–132.
6. Haykin S. Neural networks: A comprehensive foundation. — Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. — 842 p.
7. Ljung L. System identification: Theory for the user. — Upper Saddle River: Prentice Hall, Inc., 1987. — 432 p.
8. Nelles O. Nonlinear system identification. — Berlin: Springer, 2001. — 785 p.
9. Friedman J., Hastie T., Tibshirani R. The elements of statistical learning. Data mining, inference and prediction. — Berlin: Springer, 2003. — 552 p.
10. Kasabov N. Evolving fuzzy neural networks — algorithms, applications and biological motivation // Eds. T. Yamakawa, G. Matsumoto «Methodologies for the Conception, Design and Application of Soft Computing». — Singapore: World Scientific, 1998. — P. 271–274.
11. Kasabov N. Evolving fuzzy neural networks: theory and applications for on-line adaptive prediction, decision making and control // Australian J. of Intelligent Information Processing Systems. — 1998. — **5** (3). — P. 154–160.
12. Kasabov N. Evolving fuzzy neural network for on-line supervised/unsupervised, knowledge-based learning // IEEE Trans. on Man, Machine and Cybernetics. — 2001. — **31**, N 6. — P. 902–918.
13. Kasabov N. Evolving connectionist systems. — London: Springer-Verlag, 2003. — 307 p.
14. Lughofer E. Evolving fuzzy systems — methodologies, advanced concepts and applications. — Berlin: Springer, 2011. — 410 p.
15. Bodyanskiy Ye., Pliss I., Vynokurova O. Flexible neo-fuzzy neuron and neuro-fuzzy network for monitoring time series properties // Information Technology and Management Science. — 2013. — **16**. — P. 47–52.
16. Bodyanskiy Ye., Pliss I., Vynokurova O. Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks // Oil and Gas Power Engineering. — 2013. — **2**(20). — P. 158–162.
17. Wang L.-X., Mendel J.M. Fuzzy basis functions, universal approximation and orthogonal least squares learning // IEEE Trans. on Neural Networks. — 1993. — **3**. — P. 807–814.
18. Kohonen T. Self-organizing maps. — Berlin: Springer-Verlag, 1995. — 362 p.
19. Bodyanskiy Ye., Kolodyazhnyi V., Stephan A. An adaptive learning algorithm for a neuro-fuzzy network // Ed. by B. Reusch «Computational Intelligence. Theory and Applications». — Berlin-Heidelberg-New York: Springer, 2001. — P. 68–75.
20. Otto P., Bodyanskiy Ye., Kolodyazhnyi V. A new learning algorithm for a forecasting neuro-fuzzy network // Integrated Computer-Aided Engineering. — 2003. — **10**, N 4. — P. 399–409.
21. Narendra K.S., Parthasarathy K. Identification and control of dynamical systems using neural networks // IEEE Trans. on Neural Networks. — 1990. — **1**. — P. 4–26.

*Поступила 01.12.2014*