DOI 10.17587/issn.1684-6400

ISSN 1684-6400

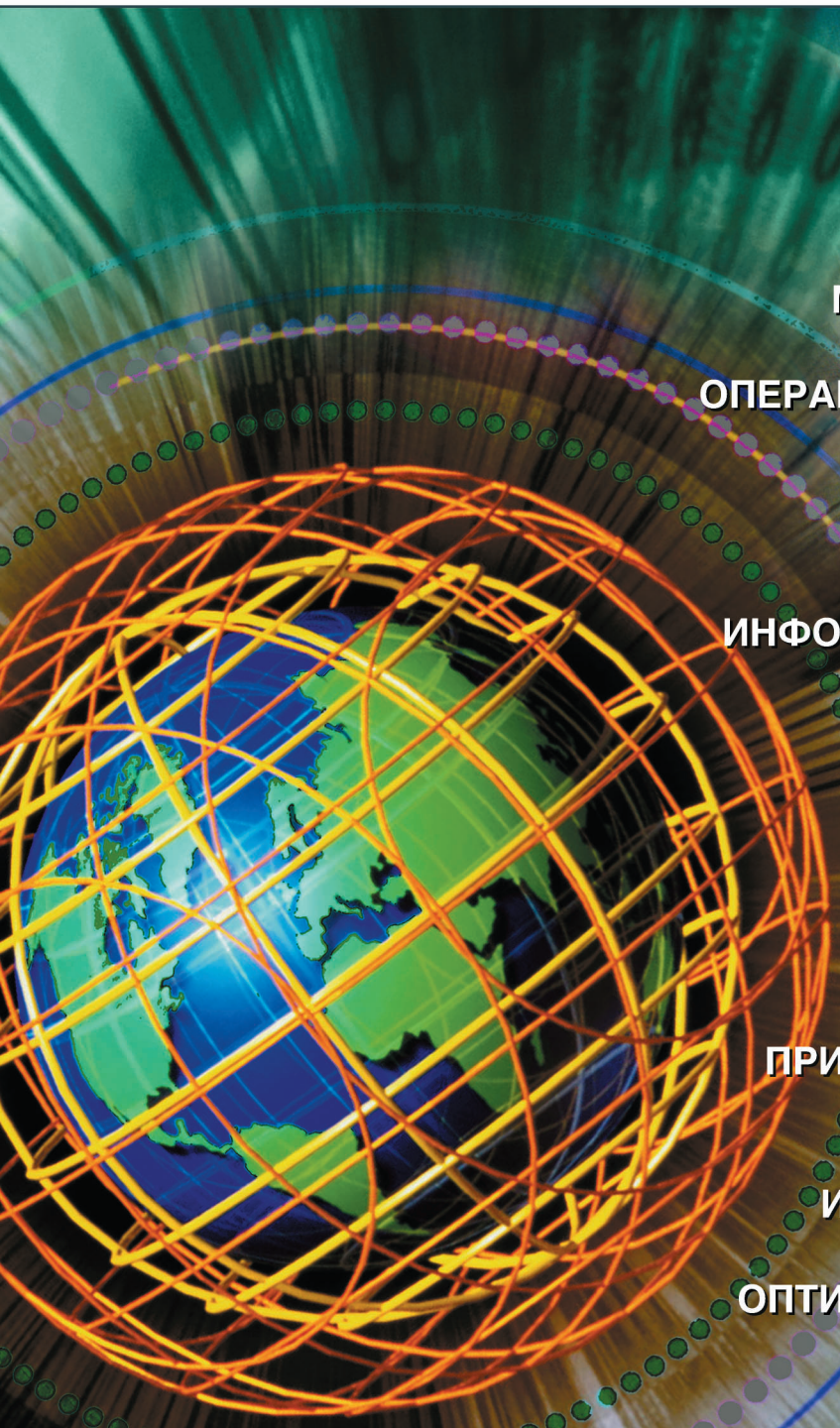
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 27

2021

№ 10

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ



САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

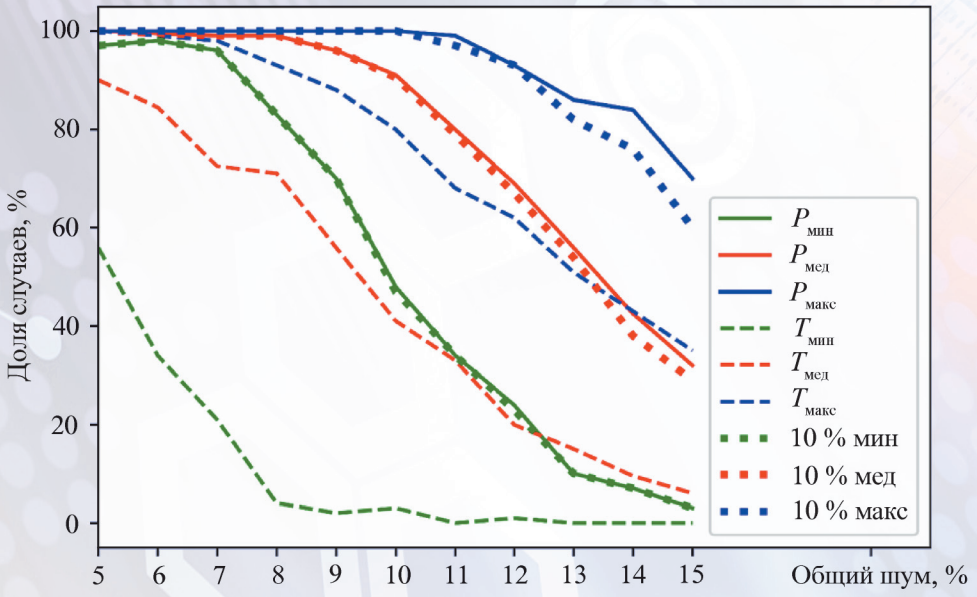
ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

Рисунок к статье Г. Н. Жуковой, М. В. Ульянова

«ВОССТАНОВЛЕНИЕ СИМВОЛЬНОЙ ПЕРИОДИЧЕСКОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ПО ПОСЛЕДОВАТЕЛЬНОСТИ С ШУМОМ»



Результаты экспериментов при возрастании общего уровня шума:
P – доля распознанных случаев в процентах; *T* – доля случаев, когда был точно найден период (длина повторяющегося фрагмента)

Рисунок к статье Г. Ч. Набибековой

«ПРИМЕНЕНИЕ ТЕХНОЛОГИИ OLAP В СРЕДЕ ЭЛЕКТРОННОЙ ДЕМОГРАФИИ»

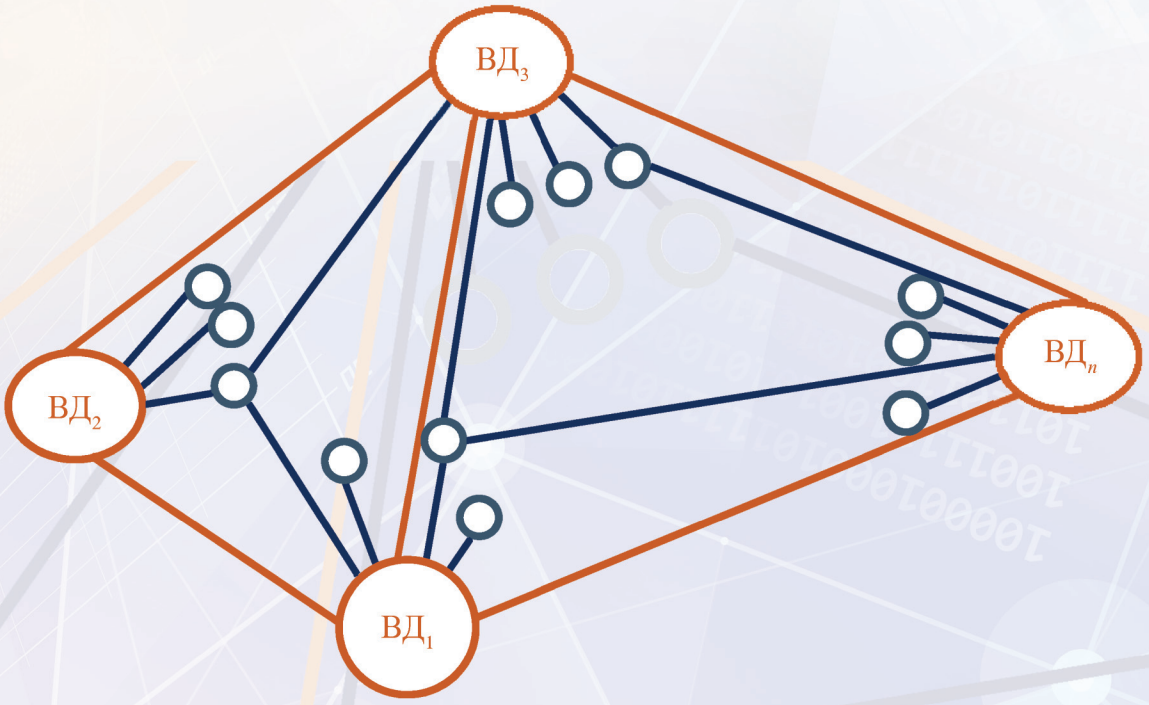


Рис. 4. Обмен информацией между ВД в среде э-демографии

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 27
2021
№ 10

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г. DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу **http://novtex.ru/IT.**
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES
INFORMACIONNYE TEHNOLOGII

Vol. 27
2021
No. 10

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

DOI 10.17587/issn.1684-6400

ISSN 1684-6400

CONTENTS

Complete Internet version of the journal at site: **http://novtex.ru/IT.**
According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

УДК 004.832.23 DOI: 10.17587/it.27.507-520

В. В. Курейчик, д-р техн. наук, проф., e-mail: vkur@sfedu.ru,
С. И. Родзин, канд. техн. наук, проф., e-mail: srodzin@yandex.ru,
Южный федеральный университет, г. Таганрог

Вычислительные модели эволюционных и роевых биоэвристик (обзор)*

Приводятся вычислительные модели эволюционных и роевых алгоритмов, использующих инспирированные природой механизмы самоорганизации и обучения. Представлены экспериментальные результаты для задачи размещения графа на плоскости с минимальной суммарной длиной ребер графа.

Ключевые слова: *биоэвристика, сходимость алгоритма, диверсификация пространства поиска, популяция, фитнесс-функция, оптимизация, граф*

Введение

Оптимизация является актуальной проблемой в таких областях, как распознавание образов, робототехника, компьютерные сети, информационная безопасность, инженерное проектирование, интеллектуальный анализ данных, финансы, цифровая экономика. В результате интенсификации исследований, направленных на развитие мощных и гибких инструментов оптимизации, было предложено много различных подходов к решению широкого спектра реальных задач поисковой оптимизации, но ни один из этих подходов не стал таким популярным, как семейство алгоритмов оптимизации, известных как алгоритмы, инспирированные природой, или биоэвристики. Природа оказалась прекрасным примером адаптивного решения проблем, множество раз показав, как они решаются, применяя оптимальную стратегию поиска, подходящую для конкретного природного явления.

Анализ мировой публикационной активности в областях, связанных с разработкой и применением инспирированных природой алгоритмов, показывает, что с 2010 г. по настоящее время число публикаций ежегодно превышает 10 000 новых статей. В пятерку лидеров по числу публикаций входят генетические, роевые и муравьиные алгоритмы, а также алгоритмы моделирования отжига и гравитационного взаимодействия. Однако вопрос, возникший с момента формулировки первого из таких алгоритмов — какая из биоэвристик работает лучше, — до сих пор остается открытым вопросом в этой области искусственного интеллекта и машинного обучения.

Биоэвристики лучше всего работают над широким спектром оптимизационных задач, когда в их механизме присутствует баланс между скоростью сходимости алгоритма и диверсификацией пространства поиска решений. Следует иметь в виду *NFL*-теорему [1], в которой говорится, что любая биоэвристика в среднем будет работать одинаково хорошо как алгоритм случайного поиска по всем возможным целевым функциям. По этой причине предполагается, что одна биоэвристика может оказаться предпочтительнее другой на конкретной задаче с учетом оценки баланса между скоростью сходимости алгоритмов и диверсификацией пространства поиска решений.

1. Особенности и классификация биоэвристик

Биоэвристики моделируют разнообразные эволюционные, биологические, физические или когнитивные явления. Структура, используемая большинством биоэвристик, остается

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00570.

практически идентичной независимо от моделируемого природного явления или процесса.

Обычно первый шаг биоэвристики включает в себя определение множества N случайно инициализированных решений (популяции) $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ ($\boldsymbol{x}_i = [x_{i1}, ..., x_{id}]$) в пространстве поиска и фитнесс-функции $f_i = f(\boldsymbol{x}_i)$ для их оценки. Биоэвристики используют итеративную схему поиска, в которой новые решения генерируются путем модификации имеющихся решений с помощью конкретных операторов, используемых каждым отдельным алгоритмом. Наконец, большинство биоэвристик включают в себя определенный процесс отбора (селекцию), в которой вновь сгенерированные решения сравниваются с теми, что находятся в текущем состоянии популяции на k-й итерации, с точки зрения их качества, как правило, в целях выбора лучшего решения. В результате этого процесса появляется новый набор решений $\boldsymbol{X}^{k+1} = \{\boldsymbol{x}_1^{k+1}, \ldots, \boldsymbol{x}_N^{k+1}\}$, соответствующий следующей $(k + 1)$-й итерации алгоритма. Процесс повторяется до тех пор, пока не будет удовлетворен определенный критерий остановки. Как только это произойдет, лучшее решение, найденное алгоритмом, определяется как наилучшее приближение для глобального оптимума.

В работах [2, 3] представлена современная классификация биоэвристик. Она включает эволюционные и роевые биоэвристики, а также биоэвристики, основанные на физических процессах, на когнитивных процессах и деятельности человека.

Согласно этой классификации к *эволюционным биоэвристикам* относятся дифференциальная эволюция (*DE*), эволюционные стратегии (*ES*), генетические алгоритмы (*GA*), генетическое программирование (*GP*), алгоритмы эволюции растущих деревьев (*SGA*) и сорняковой оптимизации (*IWO*).

Роевые биоэвристики представлены алгоритмами муравьиных (*ACO*) и пчелиных (*ABC*) колоний, алгоритмами летучих мышей (*BA*), ворон (*CSA*), кукушки (*CS*), светлячков (*FA*), алгоритмами опыления цветов (*FPA*), серых волков (*GWO*), косяка криля (*KHA*), мотыльков, летящих на свет (*MFO*), стаи птиц (*PSO*), колонии пауков (*SSO*), группы китов (*WOA*), хемотаксиса бактерий (*BFO*), косяка рыб (*FFS*), прыгающих лягушек (*SFLA*), обезьяньего поиска (*MSA*).

Биоэвристики, основанные на физических процессах, включают алгоритмы движения заряженных частиц (*EM*), гравитационного взаимодействия (*GSA*), моделирование отжига (*SA*), синусоидальных функций (*SCA*), перехода вещества из одного теплового состояния в другое (*SMS*), капель воды (*ADW*).

К *биоэвристикам, основанным на когнитивных процессах и деятельности человека*, относятся алгоритмы фейерверка (*FWA*), меметики (*MA*), биогеографии (*BBO*), поиска гармонии (*HS*), неоколониализма (*ICA*).

В данном обзоре представлены основные особенности эволюционных и роевых биоэвристик.

2. Эволюционные модели биоэвристик

Они основаны на законах эволюции в природе. В качестве операторов используют скрещивание (кроссинговер), мутацию и отбор (селекцию).

Биоэвристика *дифференциальной эволюции* (*DE*), предложенная Р. Сторном и К. Прайсом в 1996 г. [4], наряду с *GA*, является одной из самых популярных вычислительных моделей. На очередном k-м шаге алгоритма *DE* применяются операторы мутации, кроссинговера и селекции, чтобы позволить популяции решений $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ "эволюционировать" к оптимальному решению.

При выполнении операции мутации новые решения $\boldsymbol{m}_i^k = [m_{i,1}^k, m_{i,2}^k, \ldots, m_{i,d}^k]$ генерируются для каждого отдельного индивидуума $\boldsymbol{x}_i$ по формуле

$$\boldsymbol{m}_i^k = \boldsymbol{x}_{r3}^k + \boldsymbol{F}(\boldsymbol{x}_{r1}^k - \boldsymbol{x}_{r2}^k), \tag{1}$$

где $r_1, r_2, r_3 \in \{1, 2, ..., N\}$ ($r_1 \neq r_2 \neq r_3 \neq i$) — случайным образом выбранный индекс; $F \in \in [0, 2]$ — дифференциальный вес, используемый для управления разностью величин $(\boldsymbol{x}_{r1}^k - \boldsymbol{x}_{r2}^k)$. При выполнении оператора кроссинговера формируется вектор пробного решения $\boldsymbol{u}_i^k = [u_{i,1}^k, u_{i,2}^k, \ldots, u_{i,d}^k]$ для i-го индивида популяции решений. Компоненты $u_{i,j}^k$ в пробном векторе формируются из решения $\boldsymbol{x}_i^k$ и соответствующего ему решения $\boldsymbol{m}_i^k$ после выполнения операции мутации следующим образом:

$$u_{i,j}^k = \begin{cases} m_{i,j}^k, \text{ если } (rand \leqslant CR) \text{ или } (n = n^*), \\ x_{i,j}^k, \text{ если } (rand > CR), \end{cases} \tag{2}$$

где $n^* \in \{1, 2, ..., d\}$ — случайно выбранный индекс; *rand* — случайное число из интервала [0, 1], $CR \in [0, 1]$, которое используется в алгоритме *DE* при выполнении кроссинговера для формирования элементов $u_{i,j}^k$ из элементов $x_{i,j}^k$ либо $m_{i,j}^k$. При выполнении селекции фитнесс-

функция пробного решения $\boldsymbol{u}_i^k$ сравнивается с фитнесс-функцией решения-кандидата $\boldsymbol{x}_i^k$ с использованием жадного критерия.

Эволюционная стратегия (*ES*), впервые предложенная И. Рехенбергом в 60-х годах [5], первоначально представляла наиболее простой подход — (1 + 1)-*ES*. Этот подход предполагал, что из одного решения-родителя $\boldsymbol{x} = [x_1, ..., x_d]$ посредством мутации генерируется новое решение-потомок $\boldsymbol{x}'$ следующим образом: $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{N}(0, \sigma)$, где $\boldsymbol{N}(0, \sigma)$ обозначает d-мерный случайный вектор, значения которого берутся из гауссовского распределения со средним значением 0 и фиксированным стандартным отклонением σ. Далее оператор отбора исключал решение с худшим значением фитнесс-функции. В более поздних работах использовались варианты (μ + 1)-*ES* и (μ + λ)-*ES* [5].

Генетический алгоритм (*GA*) первоначально был разработан Дж. Холландом в 1960 г. [6] и развит в 1975 г. с целью понять феномен естественной адаптации и то, как этот механизм может быть реализован в компьютерных системах для решения сложных оптимизационных задач. В *GA* сначала инициализируется популяция из N решений-хромосом $\boldsymbol{x}_i = [x_{i1}, ..., x_{id}]$, каждое из которых представляет строку битов ($x_{ij} \in \{0, 1\}$). На каждой итерации *GA* популяция хромосом модифицируется путем применения различных операторов кроссинговера, мутации и селекции. Вероятность участия некоторой хромосомы в следующей популяции зависит от значения ее фитнесс-функции.

Генетическое программирование (*GP*), предложенное Дж. Коза в 1992 г. [7], рассматривается как расширение *ES* и *GA*. В *GP* решения представляют собой компьютерные программы в виде древовидных структур переменной длины, включающих набор функций и операндов.

Алгоритм эволюции растущих деревьев (*SGA*), предложенный А. Карчи в 2002 г. [8], инспирирован природным явлением — эволюцией роста деревьев. Этот алгоритм состоит из двух фаз: фазы посадки и фазы роста. На стадии посадки молодые деревья (начальные решения) располагаются в поисковом пространстве равномерно случайным образом, формируется равномерный сад.

На стадии роста применяются операторы скрещивания, ветвления и прививки. Оператор скрещивания основан на скрещивании деревьев-агентов s_i, s_j, $i, j \in [1{:}|S|]$, $i \neq j$. В процессе скрещивания векторы X_i и X_j обмениваются некоторым числом компонент. Вероятность ξ_m скрещивания агентов определяется как евклидово расстояние между агентами

$$\xi_m(s_i, s_j) = 1 - \frac{\left\| X_i - X_j \right\|_E}{r}, \tag{3}$$

где r — длина сада, равная

$$r = \left(\sum_{k=1}^{|X|} (x_k^+ - x_k^-)^2 \right)^{1/2}. \tag{4}$$

Оператор ветвления реализуется следующим образом. Пусть s_i, $i \in [1{:}|S|]$, — агент и X_i — соответствующее агенту текущее решение. Тогда предположим, что в компоненте $x_{i,k}$, $k \in [1{:}|X|]$, в точке k выполнено ветвление. В этом случае вероятность ξ_b изменения компоненты $x_{i,l}$, $l \in [1{:}|X|]$, $l \neq k$ (ветвление будет проведено в точке l), вычисляется по формуле

$$\xi_b = (x_{i,l} | x_{i,k}) = 1 - \frac{1}{(|l - k|)^2}. \tag{5}$$

Если мера близости $\sigma(s_i, s_j)$ агентов s_i, s_j, $i, j \in [1{:}|S|]$, $i \neq j$, превышает пороговое значение $\varepsilon|X|$ (ε — заранее заданная положительная константа), то применяется оператор прививки. Мера близости вычисляется по формуле

$$\sigma(s_i, s_j) = \sum_{k=1}^{|X|} \frac{|x_{i,k} - x_{j,k}|}{x_k^+ - x_k^-}. \tag{6}$$

В результате формируется промежуточная популяция (S, S'), которая объединяет исходные векторы X_i и промежуточные X_i', $i \in [1{:}|S|]$. В следующую популяцию отбираются только лучшие агенты. Процесс продолжается до достижения заданного критерия останова.

Алгоритм сорняковой оптимизации (*IWO*), предложенный А. Мехрабианом и К. Лукасом в 2002 г. [9], инспирирован таким явлением, как колонизация сельскохозяйственных угодий сорняками. В алгоритме *IWO* модель поведения сорняков при колонизации учитывает следующие основные свойства этого процесса. На этапе инициализации популяции конечное число семян распределяется по всей области поиска. На этапе репродукции производство семян выросшими растениями зависит от приспособленности растений, а произведенные семена размещаются в случайном порядке по области поиска до тех пор, пока не будет достигнут заданный максимум числа растений. На этапе отбора сохраняются растения с более высокой приспособленностью. В *IWO* при репродукции число семян n_s^i, которое произво-

дит сорняк s_i, $i \in [1:|S|]$, зависит от его текущей фитнесс-функции $\varphi_i = \varphi(X_i)$ и имеет вид

$$n_s^i = \frac{n_{\max}^s - n_{\min}^s}{\varphi^{best} - \varphi^{worst}} \varphi_i + \frac{\varphi^{best} n_{\min}^s - \varphi^{worst} n_{\min}^s}{\varphi^{best} - \varphi^{worst}}, \quad (7)$$

где $n_{\max}^s, n_{\min}^s$ — константы, определяющее минимальное и максимальное число семян, которые воспроизводятся одним сорняком на данной итерации. Пространственное распределение семян, произведенных сорняком s_i, $i \in [1:|S|]$, происходит в области родительского растения и соответствует нормальному закону распределения:

$$X_{i,j} = X_i + N_{|X|}(0, \sigma), \; i \in [1:|S|], \; j \in [1:n_i^s], \quad (8)$$

где σ — стандартное отклонение, которое зависит от номера текущего поколения t. Таким образом обеспечивается уменьшение степени вероятности попадания семян, находящихся на некотором удалении от родительского растения, с увеличением числа итераций.

В сорняковом алгоритме конкурентное исключение особей реализуется следующим образом. Пусть S_i^s — популяция сорняков, которые являются потомками s_i. Чтобы популяция сорняков достигла максимального размера $|S|$, формируется новая популяция S' путем объединения текущей популяции S со всеми остальными популяциями S_i^s:

$$S' = S \cup \left(\bigcup_{i=1}^{|S|} S_i^s \right).$$

Конкурентное исключение заключается в уничтожении тех сорняков, которые имеют меньшую приспособленность.

3. Роевые модели биоэвристик

Роевые биоэвристики инспирированы коллективным поведением разнообразных живых организмов, взаимодействующих друг с другом и окружающей средой. Несмотря на то, что движение поисковых агентов в роевых алгоритмах основано на упрощенных правилах поведения, популяция в целом способна проявлять сложные поведенческие паттерны, что позволяет им исследовать обширное пространство решений.

Алгоритм муравьиных колоний (*ACO*) является, пожалуй, одной из самых известных биоэвристик, он был впервые предложен М. Дориго в 1992 г. [10]. Муравьи перемещаются случайным образом, добывая пищу, и возвращаются в колонию, оставляя при этом постепенно испаряющийся феромонный след. Они ориентируются в направлении найденного источника пищи по этому следу. У более короткого пути феромонный след будет более интенсивным, нежели у длинных маршрутов, что, в итоге, приводит всех членов колонии к необходимости следовать оптимальным маршрутом. Вычислительная модель *ACO* представляет собой популяцию из N муравьев, которые перемещаются по графу $G(V, W)$, где V — множество узлов, W — множество дуг графа. Муравей выбирает путь в соседний узел, основываясь на плотности феромона и длине дуги. На k-й итерации алгоритма вероятность $p_{i(xv)}^k$, что i-й муравей выберет путь из узла x в узел v, задается следующим выражением:

$$p_{i(xy)}^k = \frac{(\alpha \tau_{(xy)}^k)(\beta \eta_{(xy)}^k)}{\sum\limits_{v \in V_x} (\alpha \tau_{(xv)}^k)(\beta \eta_{(xv)}^k)}, \quad (9)$$

где $\tau_{(xy)}^k$ — плотность феромонов на дуге (xv); $\eta_{(xy)}^k$ — длина дуги (xy); V_x — множество узлов, смежных x; α и β — параметры, управляющие значениями $\tau_{(xy)}^k$ и $\eta_{(xy)}^k$ соответственно.

Применяя этот механизм, муравей перемещается по дугам до узла назначения и возвращается назад по пройденному маршруту, оставляя феромонный след. Количество феромона, выделяемого i-м муравьем на дуге (xy), определяется по формуле

$$\Delta \tau_{i(xy)}^k = \begin{cases} \dfrac{Q}{L_i}, \text{ если муравей использовал} \\ \text{в маршруте дугу } (xy); \\ 0, \text{ в противном случае,} \end{cases} \quad (10)$$

где L_i — длина маршрута, пройденного i-м муравьем; Q — константа.

ACO включает в себя процедуру обновления плотности феромона по маршрутам графа для следующей итерации $(k + 1)$. При этом учитывается как количество феромонов, выделяемых каждым муравьем при обратном прохождении его маршрута, так и естественное рассеивание феромонов с течением времени:

$$\tau_{(xy)}^{k+1} = (1 - \rho)\tau_{(xy)}^k + \sum_{i=1}^{N} \tau_{i(xy)}^k, \quad (11)$$

где ρ — коэффициент испарения феромона; $\tau_{i(xy)}^k$ — количество феромонов, выделяемых i-м муравьем на дуге (xy) согласно (10).

Алгоритм пчелиного роя (*ABC*), предложенный Д. Карабога и Б. Бастурком в 2007 г. [11], пожалуй, является самым популярным из множества пчелиных алгоритмов. В *ABC* поисковыми агентами являются пчелы из искусственной медоносной колонии, которые исследуют *d*-мерное пространство в поисках нектара. Расположение источников нектара представляет собой возможное решение для данной задачи оптимизации, а количество нектара связано с фитнесс-функцией каждого из решений. Пчелиная семья неоднородна, в нее входят рабочие пчелы, пчелы-наблюдатели и пчелы-разведчики. На *k*-й итерации алгоритма каждая из рабочих пчел генерирует новое решение $\boldsymbol{v}_i^k$ следующим образом:

$$\boldsymbol{v}_i^k = \boldsymbol{x}_i^k + \varphi(\boldsymbol{x}_i^k - \boldsymbol{x}_r^k), \tag{12}$$

где $\boldsymbol{x}_i^k$ — местоположение источника, обнаруженного рабочей *i*-й пчелой; $\boldsymbol{x}_r^k$ — местоположение любого другого случайно выбранного источника нектара; φ — случайное число из интервала [−1, 1]. Пчелы-наблюдатели могут случайно посетить любой источник нектара, обнаруженный рабочими пчелами, с вероятностью

$$p_i^k = \frac{f(x_i^k)}{\sum\limits_{j=1}^{N} f(x_j^k)}. \tag{13}$$

Как только пчела-наблюдатель решила посетить определенный источник $\boldsymbol{x}_i^k$, генерируется новое решение $\boldsymbol{v}_i^k$ согласно (12), которое сравнивается с исходным местоположением $\boldsymbol{x}_i^k$. Лучшее из них выбирается в качестве нового местоположения источника нектара для следующей итерации:

$$x_i^{k+1} = \begin{cases} v_i^k, \text{ если } (f(x_i^k) < f(v_i^k)); \\ x_i^k, \text{ в противном случае.} \end{cases} \tag{14}$$

Пчелы-разведчики начинают случайным образом исследовать все пространство поиска в поисках новых источников нектара, если найденное решение не улучшается ни рабочими пчелами, ни пчелами-наблюдателями после заданного числа итераций алгоритма.

Алгоритм летучих мышей (*BA*), предложенный С. Янгом в 2010 г. [12], инспирирован механизмом эхолокации: летучие мыши испускают громкие частотно-модулированные звуковые импульсы и принимают эхо, которое отражается от окружающих объектов. Это позволяет им строить трехмерный сценарий ближайшего окружения: обнаружить добычу, уклониться от препятствий, определить место для ночлега.

Летучая мышь использует эхолокацию для достижения некоторой цели, определяемой как глобальное лучшее решение. В алгоритме *BA* используются следующие параметры, значения которых постоянно корректируются в процессе поиска: частота, громкость и скорость импульсного излучения. Частота моделируется набором *d*-мерных векторов, каждый из которых связан с *i*-й летучей мышью. Значения этих векторов случайным образом корректируются на каждой *k*-й итерации по формуле

$$\mathcal{F}_i^k = \mathcal{F}_{\min} + \beta(\mathcal{F}_{\max} - \mathcal{F}_{\min}), \tag{15}$$

где параметры $\mathcal{F}_{\min}$ и $\mathcal{F}_{\max}$ обозначают минимальную и максимальную частоты соответственно; β — это вектор случайных чисел, каждое из которых находится в интервале [0, 1].

Для параметров громкости A_i и скорости импульсного излучения r_i начальные значения A_i^0 и r_i^0 определяются при инициализации алгоритма. По мере развития процесса поиска оптимального решения значения этих параметров изменяются в соответствии со следующим выражением:

$$A_i^{k+1} = \alpha A_i^k,\ r_i^{k+1} = r_i^0(1 - \exp(-\gamma k)), \tag{16}$$

где $\alpha < 1$ и $\gamma < 1$ — некоторые константы.

Что касается обновления местоположения *i*-й летучей мыши на *k*-й итерации, то в алгоритме *BA* применяется следующий оператор:

$$\boldsymbol{x}_i^k = \boldsymbol{x}_i^{k-1} + \boldsymbol{v}_i^k, \tag{17}$$

где $\boldsymbol{x}_i^{k-1}$ представляет положение *i*-й летучей мыши на предыдущей итерации $(k - 1)$, в то время как $\boldsymbol{v}_i^k$ обозначает скорость *i*-й летучей мыши, которая, в свою очередь, вычисляется следующим образом:

$$\boldsymbol{v}_i^k = \boldsymbol{v}_i^{k-1} + \mathcal{F}_i^k(\boldsymbol{x}_i^{k-1} - \boldsymbol{x}_{best}), \tag{18}$$

где $\boldsymbol{x}_{best}$ — текущее глобально лучшее решение, найденное в процессе поиска, $\mathcal{F}_i^k$ — частотный вектор, определяемый в соответствии с формулой (15).

Кроме того, в *BA* предлагается также включить локальную схему поиска, согласно которой на каждой итерации случайно выбранный индивид среди текущих лучших решений дополнительно уточняется путем выполнения случайного блуждания следующим образом:

$$\boldsymbol{x}_*^k = \begin{cases} \boldsymbol{x}_i^k + \varepsilon A_i^k, \text{ если } (rand > r_i^k); \\ \boldsymbol{x}_i^k, \text{ в противном случае,} \end{cases} \quad (19)$$

где *rand* — случайное число, полученное из равномерно распределенного интервала [0, 1].

Заметим, что новое решение $\boldsymbol{x}_*^k$ принимается в качестве местоположения *i*-й летучей мыши только при соблюдении определенных условий, в частности:

$$\boldsymbol{x}_i^k = \begin{cases} \boldsymbol{x}_*^k, \text{ если } (rand < A_i^k \,\&\, \boldsymbol{x}_i^k < \boldsymbol{x}_*^k); \\ \boldsymbol{x}_i^k, \text{ в противном случае.} \end{cases} \quad (20)$$

Алгоритм вороньего поиска (CSA), предложенный А. Аскарзаде в 2016 г. [13], инспирирован поведением ворон. Вороны известны своей склонностью к воровству и попытками спрятать, например, излишки пищи, запоминая местоположение укрытия. Они также наблюдают и следуют за другими воронами, чтобы найти их укрытия и украсть их ресурсы. При этом вороны способны разрабатывать различные тактики, чтобы предотвратить воровство других ворон.

В *CSA* поисковыми агентами является стая, состоящая из *N* ворон, каждая из которых обладает памятью и занимает определенное местоположение $\boldsymbol{x}_i = [x_{i1}, ..., x_{id}]$ в *d*-размерном пространстве решений. Память *i*-й вороны хранит лучшее решение $\boldsymbol{m}_i = [m_{i1}, ..., m_{id}]$. Для обновления положения каждой вороны используются два оператора движения: (1) *i*-я ворона следует за случайно выбранной *j*-й вороной к ее укрытию; (2) *i*-я ворона обманывается *j*-й вороной и перемещается в другое место. На очередной итерации положение каждой вороны обновляется следующим образом:

$$x_i^{k+1} = \begin{cases} x_i^k + r_i \cdot fl_i^k \cdot (m_j^k - x_i^k), \\ \text{если } (rand \geqslant AP_j^k); \\ r_i, \text{ если } (rand < AP_j^k), \end{cases} \quad (21)$$

где AP_j^k — вероятность появления *j*-й вороны на *k*-й итерации алгоритма; *rand* — случайное число из равномерного распределения в интервале [0, 1]. Если $(rand \geqslant AP_j^k)$, то это означает, что *j*-я ворона "не знает", что *i*-я ворона следует за ней. В результате *i*-я ворона приближается к укрытию *j*-й вороны $\boldsymbol{m}_j^k$. Параметр fl_i^k обозначает максимальную длину полета *i*-й вороны; $r_i \in [0, 1]$ — случайная величина. Если $(rand < AP_j^k)$, то предполагается, что *j*-я ворона осознает, что его преследует *i*-я ворона, и в ответ выбирает ложное направление. При этом позиция *i*-й вороны обновляется до r_i — случайно сгенерированного решения в допустимом пространстве решений.

Алгоритм кукушки (*CS*), разработанный С. Янгом и С. Дебом в 2009 г. [14], инспирирован явлением гнездового паразитизма некоторых видов кукушек, которые подкладывают свои яйца в гнезда других птиц. Если хозяин гнезда обнаружит, что яйца не его, то он или выбросит эти чужие яйца или просто покинет гнездо и создаст новое где-то в другом месте. Согласно алгоритму *CS* инициализируется популяция из *N* хозяйских гнезд и положение кукушки. На очередной итерации алгоритма моделируется ситуация выбора кукушкой гнезда для откладывания собственных яиц путем применения случайного блуждания с помощью полетов Леви следующим образом:

$$x_{new}^k = x_i^k + \alpha \oplus Levy(\lambda), \quad (22)$$

где $\alpha > 0$ — размер шага, который зависит от масштаба задачи (обычно принимают равным 1); $\oplus$ — произведение Адамара; $Levy(\lambda)$ обозначает случайное блуждание, в котором длина полета кукушки имеет распределение вероятности Леви.

После того как решение x_{new}^{k+1} сгенерировано, его фитнесс-значение сравнивается с качеством решения, выбранного случайным образом. Если новое решение лучше, то $x_i^{k+1} = x_{new}^k$. С вероятностью p_a удаляем из популяции некоторое число худших случайно выбранных гнезд и строим новые гнезда в местах, определенных с помощью полетов Леви. Если поколение достигло заданного предела, то заканчиваем алгоритм.

Алгоритм светлячка (*FA*), предложенный С. Янгом в 2010 г. [15], инспирирован светоизлучающим поведением, наблюдаемым у светлячков. В *FA* поисковыми агентами является рой светлячков, взаимодействующих посредством биолюминесцентного свечения. *FA* использует следующую модель поведения светлячков: привлекательность светлячка для других особей пропорциональна его яркости; менее привлекательные светлячки перемещаются в направлении более привлекательного светлячка; яркость излучения данного светлячка, видимая другим светлячком, уменьшается с увеличением расстояния между ними; если светлячок не видит возле себя светлячка более яркого, чем он сам, то он перемещается случайным образом. Яркость излучения светлячка f_i в популяции из *N* светлячков принимается равной значению фитнесс-функции

в его текущем положении. Привлекательность светлячка f_i для светлячка f_j полагаем равной

$$\beta_{ij} = \beta_0 \exp(-\gamma r_{ij}^2), \qquad (23)$$

где r_{ij} — расстояние между светлячками f_i и f_j; β_0 — взаимная привлекательность светлячков при нулевом расстоянии между ними; γ — вещественная величина, имеющая смысл коэффициента поглощения света средой.

Перемещение i-го светлячка к j-му светлячку задается следующим выражением:

$$\Delta \boldsymbol{x}_{ij} = \boldsymbol{x}_i + \beta_{ij}(\boldsymbol{x}_j - \boldsymbol{x}_i) + \alpha \boldsymbol{o}_i, \qquad (24)$$

где $\boldsymbol{o}_i$ — d-мерный вектор, обозначающий случайное движение; α — параметр рандомизации, значения которого находятся в интервале [0, 1]. По мнению авторов алгоритма, перемещение позволяет поисковым агентам более эффективно исследовать пространство решений оптимизационной задачи, позволяет находить как локальные, так и глобальный оптимум.

Алгоритм опыления цветов (*FPA*), предложенный С. Янгом в 2012 г. [16], инспирирован процессом опыления цветов. В *FPA* поисковыми агентами являются отдельных цветы (пыльцевые гаметы), которые используют два метода опыления: перекрестное опыление, при котором пыльца переносится на большие расстояния, и самоопыление. В контексте роевого интеллекта перекрестное опыление рассматривается как процесс поиска глобального оптимума, а самоопыление — как поиск локального оптимума. Поэтому результатами опыления цветка на некоторой итерации алгоритма могут быть следующие положения его гаметы:

$$\boldsymbol{x}_i^{k+1} = \begin{cases} \boldsymbol{x}_i^k + L(\boldsymbol{x}_* - \boldsymbol{x}_i^k), \text{ если } (rand > P); \\ \boldsymbol{x}_i^k + \boldsymbol{o}(\boldsymbol{x}_j^k - \boldsymbol{x}_q^k), \text{ если } (rand \leqslant P), \end{cases} \qquad (25)$$

где *rand* — случайное число из интервала [0, 1]; P — некоторый порог вероятности, при его превышении происходит перекрестное опыление, иначе — самоопыление; $\boldsymbol{x}_*$ — текущий глобальный оптимум; L — масштабирующий параметр, отражающий силу опыления. Значение L задается из распределения Леви:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi(s^{1+\lambda})}, (s \gg s_0 > 0), \qquad (26)$$

где $\Gamma(\lambda)$ — стандартная гамма-функция (относительно постоянного параметра λ); s — заданный пользователем размер шага.

Алгоритм серых волков (*GWO*), предложенный С. Янгом в 2007 г. [17], инспирирован процессом охоты стаи серых волков со строгой социальной иерархией (α-, β-, δ- и ω-волки). Вожак стаи направляет остальных особей на поиск добычи. Волки "исследуют" местность, кто-то из них чует запах жертвы, начинается ее поиск в направлении усиления запаха добычи. Волки разделяются на небольшие группы, и каждая группа осуществляет поиск в определенном направлении, отличном от направлений других групп. Алгоритм *GWO* моделирует процесс их охоты. Поисковые агенты — это волки. Стая (популяция) представляет собой множество потенциальных решений, координаты которых обновляются на каждой итерации, пока не будет найдено оптимальное решение или не будет выполнено максимально заданное число вычислений фитнесс-функции, которая характеризует, насколько сильно ощущается запах добычи волками, а координаты добычи — оптимальное решение. Совместная охотничья тактика волков заключается в преследовании добычи, пока ее не окружат, не приступят к нападению и не убьют.

На очередной итерации каждый волк обновляет свою позицию для следующего шага:

$$\boldsymbol{x}_i^{k+1} = (\boldsymbol{\alpha}_i^k + \boldsymbol{\beta}_i^k + \boldsymbol{\delta}_i^k)/3, \qquad (27)$$

где $\boldsymbol{\alpha}_i^k, \boldsymbol{\beta}_i^k$ и $\boldsymbol{\delta}_i^k$ — позиции α-, β- и δ-волков, равные

$$\boldsymbol{\alpha}_i^k = \boldsymbol{x}_\alpha^k - \boldsymbol{A}^k \left| \boldsymbol{C}^k \boldsymbol{x}_\alpha^k - \boldsymbol{x}_i^k \right|; \qquad (28)$$

$$\boldsymbol{\beta}_i^k = \boldsymbol{x}_\beta^k - \boldsymbol{A}^k \left| \boldsymbol{C}^k \boldsymbol{x}_\beta^k - \boldsymbol{x}_i^k \right|; \qquad (29)$$

$$\boldsymbol{\delta}_i^k = \boldsymbol{x}_\delta^k - \boldsymbol{A}^k \left| \boldsymbol{C}^k \boldsymbol{x}_\delta^k - \boldsymbol{x}_i^k \right|, \qquad (30)$$

где $\boldsymbol{x}_\alpha^k$, $\boldsymbol{x}_\beta^k$ и $\boldsymbol{x}_\delta^k$ — текущие положения α-, β- и δ-волков соответственно, причем

$$\boldsymbol{A}^k = 2\boldsymbol{a}^k \boldsymbol{r}_1^k - \boldsymbol{a}^k, \qquad (31)$$

$$\boldsymbol{C}^k = 2\boldsymbol{r}_2^k, \qquad (32)$$

где $\boldsymbol{a}^k$ — вектор коэффициентов, значения которых линейно убывают от 2 до 0 в ходе итераций; $\boldsymbol{r}_1^k$ и $\boldsymbol{r}_2^k$ — случайные векторы, значения которых равномерно распределены на интервале [0, 1].

Из формул (27)—(32) и описания алгоритма видно, что в *GWO* обновляются лишь координаты волков без учета скорости их перемещения в пространстве. Достаточно подобрать лишь два параметра — размер популяции и

шаг, с которым перемещаются волки в направлении вожака и добычи.

Алгоритм косяка криля (*KHA*), предложенный А. Гандоми и А. Алави в 2012 г. [18], инспирирован поведением косяка криля в процессе поиска пищи. В *KHA* поисковыми агентами являются особи криля, которые двигаются, руководствуясь следующими принципами: движение в направлении к другим особям, чтобы поддержать высокую плотность косяка; движение в сторону источника пищи; случайное диффузионное движение в поисках пищи. При этом положение i-й особи криля в момент времени $(t + 1)$ определяется следующим образом:

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta t \frac{d\boldsymbol{x}_i^t}{dt}, \qquad (33)$$

где $\frac{d\boldsymbol{x}_i^t}{dt}$ — вектор скорости i-й особи криля, представленный моделью Лагранжа:

$$\frac{d\boldsymbol{x}_i^t}{dt} = N_i + F_i + D_i, \qquad (34)$$

где N_i — движение в направлении к другим особям; F_i — движение в сторону источника пищи; D_i — случайное движение в поисках пищи.

Алгоритм мотылька, летящего на свет (*MFO*), предложенный С. Мирджалили в 2015 г. [19], инспирирован механизмом ночной навигации мотыльков, которые буквально гипнотизируются каким-либо источником света, забывая о собственной безопасности. Движение мотылька напоминает логарифмическую спираль, в центре которой расположен источник света и тепла. Они вьются вокруг лампочки не в случайном порядке, а подчиняясь неким неведомым до сих пор законам. В *MFO* поисковые агенты представлены популяцией N_M мотыльков, каждый из которых имеет определенную позицию $\boldsymbol{M}_i = [m_{i,1}, m_{i,2}, ..., m_{i,d}]$ в заданном пространстве, и множеством N_F источников света, случайно распределенных в этом пространстве, так что каждый из них имеет определенную позицию $\boldsymbol{F}_j = [f_{j,1}, f_{j,2}, ..., f_{j,d}]$. На итерации k каждому мотыльку сначала присваивается определенный источник света, а затем применяется оператор его движения к свету по логарифмической спирали:

$$\boldsymbol{M}_i^{k+1} = \boldsymbol{D}_{ij}^k \mathbf{e}^{bl} \cos(2\pi l) + F_j^k, \qquad (35)$$

где $\boldsymbol{D}_{ij}^k = \left| F_j^k - M_i^k \right|$; b — постоянный параметр; l — случайное число из интервала $[r, 1]$ (r линейно уменьшается от -1 до -2 на каждой итерации алгоритма). В *MFO* используется механизм, согласно которому число источников света N_F в пространстве поиска уменьшается по мере продолжения итерационного процесса:

$$N_F^{k+1} = round\left(N_F^0 - k \frac{N_F^0 - 1}{K} \right), \qquad (36)$$

где N_F^0 — начальное (максимальное) число источников света; K — максимальное число итераций алгоритма.

Алгоритм роя частиц (*PSO*), предложенный Дж. Кеннеди и Р. Эберхартом в 1995 г. [20], инспирирован поведением стаи птиц, коллективно добывающих пищу. В *PSO* поисковые агенты (частицы) описываются набором из трех d-мерных векторов: текущего положения частицы, ее предыдущего наилучшего положения и ее скорости. Предполагается, что частица роя знает наилучшую позицию ее ближайших соседей. Текущая позиция i-й частицы вычисляется по формуле

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1}, \qquad (37)$$

где $\boldsymbol{v}_i^{k+1}$ — скорость i-й частицы, которая вычисляется следующим образом:

$$\boldsymbol{v}_i^{k+1} = \boldsymbol{v}_i^k + c_1(\boldsymbol{r}_1^k(\boldsymbol{p}_i^k - \boldsymbol{x}_i^k)) + c_2(\boldsymbol{r}_2^k(\boldsymbol{g}_i^k - \boldsymbol{x}_i^k)), \qquad (38)$$

где $\boldsymbol{p}_i^k$ — предыдущая лучшая позиция i-й частицы; $\boldsymbol{g}_i^k$ — текущая наилучшая позиция ее ближайших соседей; $\boldsymbol{r}_1^k$ и $\boldsymbol{r}_2^k$ — d-мерные векторы из случайных чисел на интервале [0, 1]; c_1 — когнитивный параметр, учитывающий "собственный опыт" (историю) частицы; c_2 — социальный параметр, управляющий воздействием глобального лучшего положения на скорость частицы.

Идея алгоритма заключается в том, что частицы, которые вначале равномерно распределены по поверхности отклика функции, с течением времени (от поколения к поколению) начинают группироваться ("сбиваться в стаи") около локальных оптимумов, причем наибольшая стая собирается около глобального оптимума. При этом почти всегда имеются частицы, находящиеся в стороне от таких стай, а также частицы, "вылетающие" за границы допустимой области.

Алгоритм колонии пауков (*SSO*), предложенный Е. Куэвасом и др. в 2013 г. [21], инспирирован коллективным поведением, наблюдаемым в колонии социальных пауков. Колония включает пауков и сеть из паутины, которая служит средой для взаимодействия. Особенностью колонии является преобладание самок

пауков в пределах популяции, а также различие в видах деятельности (создание и поддержание сети, охота, спаривание). Важной чертой социальных пауков является их способность воспринимать вибрации паутины для получения информации, например, о размере пойманной добычи.

В *SSO* поисковыми агентами является колония из N пауков (каждый с соответствующей позицией $\boldsymbol{s}_i = [s_1, s_2, ..., s_d]$), взаимодействующих в трехмерном пространстве решений (паутина). Стимул, воспринимаемый i-м пауком в результате вибраций от j-го паука, моделируется следующим образом:

$$Vib_{ij} = \omega_{s_j} \mathbf{e}^{-r_{ij}^2}, \tag{39}$$

где $r_{ij} = \boldsymbol{s}_i - \boldsymbol{s}_j$ — расстояние между i-м и j-м пауками; ω_{sj} — "вес" j-го паука, равный

$$\omega_{s_j} = \frac{f(\boldsymbol{s}_j) - f_{worst}}{f_{best} - f_{worst}}, \tag{40}$$

где f_{best} и f_{worst} — соответственно лучшие и худшие текущие значения фитнесс-функции среди всех пауков в общей паутине.

В *SSO* моделируются передвижения самки по следующим правилам:

$$f_i^{k+1} = \begin{cases} f_i^k + \alpha \cdot Vib_{c_i}(\boldsymbol{s}_{c_i}^k - \boldsymbol{f}_i^k) + \\ + \beta \cdot Vib_{b_i}(\boldsymbol{s}_b^k - \boldsymbol{f}_i^k) + \delta \cdot \left(\gamma - \frac{1}{2}\right), \\ \text{если } (P > P_f); \\ f_i^k - \alpha \cdot Vib_{c_i}(\boldsymbol{s}_{c_i}^k - \boldsymbol{f}_i^k) - \\ - \beta \cdot Vib_{b_i}(\boldsymbol{s}_b^k - \boldsymbol{f}_i^k) + \delta \cdot \left(\gamma - \frac{1}{2}\right), \\ \text{если } (P \leqslant P_f), \end{cases} \tag{41}$$

где $\boldsymbol{s}_{c_i}^k$ — положение ближайшего лучшего паука; $\boldsymbol{s}_b^k$ — положение самого лучшего паука в паутине; Vib_{ci} и Vib_{bi} — вибрации, воспринимаемые i-м пауком от $\boldsymbol{s}_{c_i}^k$ и $\boldsymbol{s}_b^k$ согласно (39); α, β, δ, γ и P — случайные числа в интервале [0, 1]; P_f — порог вероятности, используемый для определения вида передвижения самки паука к самцу (притяжение или отталкивание). Доминирующие самцы пауков имеют больший размер и вес, а доминантные самцы, как правило, концентрируются в центре мужской популяции. В *SSO* характерные для самцов модели поведения моделируются следующим образом:

$$\boldsymbol{m}_i^{k+1} = \begin{cases} \boldsymbol{m}_i^k + \alpha \cdot Vib_{f_i}(\boldsymbol{s}_{f_i}^k - \boldsymbol{m}_i^k) + \delta \cdot \left(\gamma - \frac{1}{2}\right), \\ \text{если } (\omega_{m_i}^k > M(\omega_m)), \\ m_i^k + \alpha \cdot \left(\frac{\sum\limits_{h=1}^{N_m} \boldsymbol{m}_h^k \cdot \omega_{m_h}^k}{\sum\limits_{h=1}^{N_m} \omega_{m_h}^k} - \boldsymbol{m}_i^k \right), \\ \text{если } (\omega_{m_i}^k \leqslant M(\omega_m)), \end{cases} \tag{42}$$

где $\omega_{m_i}^k$ — вес i-го самца паука; $M(\omega_m)$ — медианное значение по отношению к весу всех самцов пауков; $\boldsymbol{s}_{f_i}^k$ — положение ближайшей самки паука к $\boldsymbol{m}_i^k$; Vib_{fi} — стимул, воспринимаемый i-м самцом паука в результате вибраций его ближайшей самки; α, β, γ — случайные числа в интервале [0, 1]. Алгоритм *SSO* использует механизм скрещивания для создания новых решений-кандидатов.

Алгоритм охоты группы китов (*WOA*), предложенный С. Мирджалили и Э. Льюисом в 2016 г. [22], инспирирован поведением, наблюдаемым у горбатых китов, которые используют групповой маневр для окружения косяка криля или мелких рыб, плавая по спирали вокруг своей добычи. Во время движения под водой каждый кит на выдохе создает пузырьковый барьер (сеть), который не дает жертве сбежать. Когда добычу загоняют в плотный круг, один кит издает сигнал кормления для других китов, и все особи одновременно плывут к поверхности с открытыми ртами, чтобы питаться пойманной добычей.

В *WOA* поисковыми агентами являются N китов, а процесс поиска добычи включает разведку и коллективную охоту. На очередной итерации алгоритма в ходе разведки положение каждого кита обновляется согласно следующему уравнению:

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_{rand}^k - \boldsymbol{A}^k \boldsymbol{D}^k, \tag{43}$$

где $\boldsymbol{x}_{rand}^k$ — случайная позиция китов в процессе разведки, причем

$$\boldsymbol{A}^k = 2\boldsymbol{a}^k \boldsymbol{r}_1^k - \boldsymbol{a}^k; \tag{44}$$

$$\boldsymbol{D}^k = |(2\boldsymbol{r}_2^k \boldsymbol{x}_{rand}^k) - \boldsymbol{x}_i^k|, \tag{45}$$

где $\boldsymbol{a}^k$ — вектор коэффициентов, значения которых линейно убывают от 2 до 0 в ходе итераций; $\boldsymbol{r}_1^k$ и $\boldsymbol{r}_2^k$ — случайные векторы, значения которых берутся из равномерно распределенного интервала [0, 1]. Фаза коллективной охоты начинается, когда группа китов определила

свою добычу. Алгоритм *WOA* моделирует два правила их движения: окружение добычи и атака сетью из воздушных пузырей. Для окружения добычи киты перемещаются вокруг целевой добычи, для атаки сетью пузырей каждый кит движется по спирали вокруг цели. Эти два правила представляются следующими операторами обновления позиции:

$$\boldsymbol{x}_i^{k+1} = \begin{cases} \boldsymbol{x}_*^k - \boldsymbol{A}^k \boldsymbol{D}^k, \text{ если } p < 0,5; \\ \boldsymbol{D}_*^k (\mathbf{e}^{bl} \cos(2\pi l)) + \boldsymbol{x}_*^k, \text{ если } p \geqslant 0,5, \end{cases} \quad (46)$$

где $\boldsymbol{x}_*^k$ — наилучшая текущая позиция поискового агента; $\boldsymbol{D}_*^k = |\boldsymbol{x}_*^k - \boldsymbol{x}_i^k|$; p — случайное число из интервала [0, 1]; значение 0,5 — порог вероятности. При $p < 0,5$ применяется оператор, соответствующий правилу окружения добычи, при $p \geqslant 0,5$ — оператор, соответствующий правилу атаки пузырьковой сетью по модели логарифмической спирали, форма которой управляется постоянным параметром b и случайной величиной $l \in [-1, 1]$.

Алгоритм бактериальной оптимизации (*BFO*), предложенный К. Пассино в 2002 г. [23], инспирирован перемещением колонии бактерий *E.coli* в зависимости от наличия полезных или вредных вещества в среде их обитания. Они используют групповой маневр, пытаясь найти пищу или избежать вредной среды. В контексте задачи поисковой оптимизации перемещение бактерий можно интерпретировать как механизм оптимизации использования бактерией известных пищевых ресурсов и поиска новых, потенциально более ценных областей. В *BFO* поисковыми агентами являются N бактерий, а процесс поиска пищи включает следующие механизмы: хемотаксис, репродукция и ликвидация/рассеивание.

Пусть $X_{i,r,l}$ ($|X| \times l$) — вектор текущего положения бактерии $s_i \in S$ на итерации t, r-м шаге репродукции и l-м шаге ликвидации/рассеивания. Здесь $i \in [1{:}|S|]$, $t \in [1 : \hat{t}]$, $r \in [1 : \hat{t}^r]$, $l \in [1 : \hat{t}^l]$, где $|S|$ — четное число бактерий в колонии S; $\hat{t}, \hat{t}^r, \hat{t}^l$ — общее число итераций, шагов репродукции, а также шагов ликвидации/рассеивания. Значение фитнесс-функции обозначим $\varphi_{i,\ r,\ l}$. Тогда процедура хемотаксиса описывает следующее положение бактерии $X'_{i,r,l}$ бактерии s_i формулой

$$X'_{i,r,l} = X_{i,r,l} + \lambda_i \frac{V_i}{\|V_i\|_E}, \quad (47)$$

где V_i — текущий направляющий ($|X| \times 1$)-вектор шага хемотаксиса бактерии s_i; λ_i — текущее значение этого шага. При плавании бактерии на следующей итерации вектор V_i остается неизменным: $V'_i = V_i$. При кувырке бактерии вектор V'_i представляет собой случайный вектор, компоненты которого имеют значения в интервале [−1, 1]. При кувырке имеет место равенство $V'_i = U_{|X|}(-1; 1)$. Плавание каждой бактерии продолжается до тех пор, пока значение целевой функции увеличивается. Значение хемотаксиса в формуле (47) может меняться в процессе поиска, уменьшаясь по некоторому закону с ростом числа итераций t.

Процедура репродукции увеличивает скорость сходимости алгоритма, что реализуется за счет сужения поискового пространства. Пусть h_i — "состояние здоровья" бактерии s_i. Тогда суммарное значение фитнесс-функции: $h_i = \sum_{\tau=1}^{t} \varphi_{i,r,l}(\tau)$. Далее бактерии сортируются в порядке убывания состояний здоровья, наиболее слабые агенты исключаются из списка, а каждый из выживших агентов делится на два идентичных агента с одинаковыми координатами. После реализации механизма репродукции общее число бактерий сохраняется неизменным и равно $|S|$.

Процедура ликвидации/рассеивания позволяет алгоритму выходить из найденных локальных экстремумов. Для этого в соответствии с предварительно заданной вероятностью ξ_e отбираются n бактерий $s_{i_1}, s_{i_2}, \ldots, s_{i_n}$, которые исключаются из популяции. Вместо уничтоженных агентов в случайных точках пространства генерируются новые агенты.

Алгоритм поиска пищи косяком рыб (*FFS*), предложенный Б. Фило и Л. Нето в 2008 г. [24], инспирирован коллективным механизмом питания и согласованного движения косяков рыб. Главной особенностью является наличие веса у каждой рыбы-агента. Пусть w_i, $i \in [1{:}|S|]$, — вес агента s_i. Вес прямо пропорционален разности значений фитнесс-функции на последующей и текущей итерациях:

$$w'_i = w_i + \frac{\varphi(X'_i) - \varphi(X_i)}{\max(\varphi(X'_i), \varphi(X_i))}, \ i \in [1 : |S|]. \quad (48)$$

Вес агента ограничивается величиной $w_{\max} > 0$. В процессе индивидуального плавания направление перемещения рыбы-агента задается случайным образом. Пусть шаг перемещения V_i^{ind} представляет собой случайную величину в интервале $[0; v_{\max}^{ind}]$:

$$V_i^{ind} = U_{|X|}(0; 1) v_{\max}^{ind}, \ i \in [1 : |S|]. \quad (49)$$

Величина $v_{\max}^{ind}$ с ростом числа итераций линейно уменьшается. После того, как все $|S|$ агентов завершили индивидуальные плавание, реализуется инстинктивно-коллективное плавание по формуле

$$X_i^{\theta} = X_i^{\tau} + \frac{\sum_j V_j^{ind}(\tau)(\varphi(X_j^{\tau}) - \varphi(X_j^{t}))}{\sum_j (\varphi(X_j^{\tau}) - \varphi(X_j^{t}))}. \quad (50)$$

Второе слагаемое в (50) — это шаг миграции, общий для всех агентов. Затем выполняется коллективно-волевое плавание, в процессе которого все агенты смещаются к центру тяжести косяка рыб, если общий вес косяка в результате выполнения предыдущих этапов увеличился, иначе — в противоположном направлении. Координаты центра тяжести косяка X_c определяются по формуле

$$X_c^{\theta} = \frac{\sum_i w_i^{\theta} X_i^{\theta},\ i \in [1:|S|]}{w_{\Sigma}^{\theta}}, \quad (51)$$

где $w_{\Sigma}^{\theta} = w_{\Sigma}(\theta) = \sum_{i=1}^{|S|} w_i^{\theta}$ — суммарный вес популяции на данной итерации.

Алгоритм прыгающих лягушек (*SFLA*), предложенный М. Юсуфом и К. Лэнси в 2003 г. [25], инспирирован поведением групп прыгающих лягушек в процессе поиска пищи. Основой алгоритма является комбинирование локального поиска в пределах каждой из групп лягушек и глобального поиска путем обмена информацией о положении лучших лягушек с определением глобально лучшей лягушки. Популяция задается множеством из S лягушек, разделенных на группы. Вначале алгоритм *SFLA* генерирует популяцию агентов-лягушек $S = s_i$, $i \in (1{:}|S|)$, оценивается фитнесс-функция каждого агента $\varphi(X_i) = \varphi_i$, $i \in (1{:}|S|)$, и находится глобально лучший агент s^{best}. Затем множество агентов разделяется на S^p групп $S_j^p = (s_{j,k}, k \in [1:n], j \in [1:|S^p|])$. Каждая группа содержит $n = \frac{|S|}{|S^p|}$ агентов, $\bigcap_{j=1}^{|S^p|} S_j^p = S$. Далее в каждой группе определяется наилучший s_j^{best} и наихудший s_j^{worst} агенты и проводится улучшение положения наихудшего агента по формуле

$$X_{j,*}^{worst}(t+1) = X_{j,*}(t) + U_i(-0,5;0,5) \times \times v_{\max} \frac{X_{j,*}^{best} - X_{j,*}^{worst}}{\left\| X_{j,*}^{best} - X_{j,*}^{worst} \right\|_E},\ j \in [1:|S^p|], \quad (52)$$

где $X_{j*}(t)$ — текущее положение агента; U_i — случайное вещественное число, имеющее смысл значения шага перемещения агента, включающего нижнюю и верхнюю границы этой величины; $v_{\max}$ — максимальное значение шага перемещения. Если данная операция не дает желаемого результата, то улучшается положение агента $s_{j,*}^{worst}$ путем перемещения его по направлению лучшего агента s^{best}. Иначе агент $s_{j,*}^{worst}$ заменяется случайно сгенерированным агентом. После повторения этих шагов заданное число раз реализуется тасование агентов между группами с получением новых групп S_j^p, $j \in [1:|S^p|]$, пока не будет достигнут критерий останова алгоритма.

Алгоритм обезьяньего поиска (*MSA*), предложенный Р. Жао и В. Тангом в 2008 г. [26], инспирирован поведением обезьян в процессе их лазания по горам при поиске пищи. Обезьяны исходят из того, что чем выше гора, тем больше пищи на ее вершине. Местность, которую обследуют обезьяны, представляет собой ландшафт фитнесс-функции, так что решению соответствует самая высокая гора. Из своего текущего положения обезьяна движется вверх до тех пор, пока не достигнет вершины горы. Затем обезьяна делает серию локальных прыжков в случайном направлении в надежде найти более высокую гору, и движение вверх повторяется. После выполнения некоторого числа подъемов и локальных прыжков обезьяна полагает, что в достаточной степени исследовала ландшафт в окрестности своего начального положения. Для того чтобы обследовать новую область пространства поиска, обезьяна выполняет длинный глобальный прыжок. Указанные выше действия повторяются заданное число раз. Решением задачи объявляется самая высокая из вершин, найденных данной популяцией обезьян.

На предварительном этапе алгоритма *MSA* выполняется ввод его параметров, инициализация начальной популяции обезьян p_i, $i \in [1{:}|P|]$, а также задается позиция каждой i-й обезьяны. Затем осуществляется процедура набора высоты. При этом указываются значение шага α, направление поиска R, число итераций t. Поиск продолжается до нахождения самой высокой вершины горы. Данное положение объявляется текущим, и происходит запись его в базу данных полученных результатов. На следующем этапе поиск реализуется в окрестности полученного текущего положения за счет выполнения локальных прыжков каждой из обезьян. Новое положение обезьяны p_i вычисляется как

$$X'_{i,j} = U_1((x_{i,j} - b);\ (x_{i,j} + b)),$$
$$i \in [1:|P|],\ j \in [1:|X|]. \quad (53)$$

Это положение обезьяны в координатной плоскости является случайной величиной, равномерно распределенной в интервале $[(x_{i,j} - b); (x_{i,j} + b)]$, где параметр $b > 0$ определяет максимально возможную длину прыжка. Прыжки выполняются заданное число раз. Затем реализуется процедура глобального улучшения за счет выполнения "глобальных прыжков" обезьян p_i с шагом v:

$$v = U_1 (v_{\min}; v_{\max}),$$

где $v_{\min}$, $v_{\max}$ — нижнее и верхнее значения этой величины. Величина v может быть как положительной, так и отрицательной. Если v принимает положительное значение, то совершается прыжок в сторону центра тяжести x^c, иначе — в противоположную сторону. Новое положение обезьяны p_i вычисляется как

$$X'_{i,j} = x_{i,j} + v(x_j^c - x_{i,j}), \quad i \in [1 : |P|], j \in [1 : |X|], \tag{54}$$

где $x_j^c = \frac{1}{|S|}\sum_{i=1}^{S} x_{i,j}$ — текущее положение центра тяжести популяции обезьян по j-му координатному направлению.

4. Результаты экспериментов

Чтобы продемонстрировать эффективность, а также вычислительные характеристики некоторых из представленных выше биоэвристик, были проведены вычислительные эксперименты. В качестве тестовой задачи использовалась задача размещения графа на плоскости с минимальной суммарной длиной ребер графа [27, 28].

Эксперименты проводили на случайных графах, сгенерированных на основе модели неориентированного случайного графа без кратных ребер и петель. Согласно этой модели ребро между парой вершин графа появляется независимо от всех остальных пар вершин с одинаковой вероятностью p. Тестирование проводили на сгенерированных случайных графах с вероятностью $p = 1/2$. Общее число возможных ребер в таком графе C_n^2. Для проведения экспериментальных исследований были выбраны и программно реализованы следующие алгоритмы: эволюционных стратегий (*ES*), генетический (*GA*), муравьиный (*ACO*), пчелиный (*ABC*), роя частиц (*PSO*), светлячковый (*FA*), обезьяний (*MSA*) и бактериальный (*BFO*).

При построении комплекса программ были использованы пакеты *Visual C++*, *Borland C++*, *Builder*. Отладку и тестирование разработанных алгоритмов выполняли на компьютере типа *IBM PC* с процессором *ryzen* 5 3600x с ОЗУ 16 Гбайт. Селекция использовалась на основе колеса рулетки, начальная популяция 50 решений и 100 итераций алгоритмов.

Качество работы биоэвристик рассчитывалось по следующей формуле:

$$\Delta = \frac{F_н - F_к}{F_н} \cdot 100\ \%,$$

где $F_н$ — начальное значение фитнесс-функции; $F_к$ — конечное значение фитнесс-функции; Δ — оценочная функция изменения суммарной длины соединений в графовой модели, выраженная в процентах.

В табл. 1 представлены результаты сравнения биоэвристик по времени решения задачи размещения графов в зависимости от числа вершин графа.

В табл. 2 представлены результаты сравнения биоэвристик по значению оценочной функции (Δ, %) задачи размещения графов в зависимости от числа вершин графа.

Из анализа таблиц можно сделать вывод, что самым быстрым алгоритмом оптимизации из конкурирующих алгоритмов является алгоритм эволюционных стратегий *ES*. Однако он не позволяет получать решения, лучшие по значению оценочной функции Δ. Другие био-

Таблица 1

Сравнение результатов размещения тестовых графов алгоритмами *ES, GA, ACO, ABC, PSO, FA, MSA* и *BFO* по времени решения

№ теста	Число вершин графа	*ES*	*GA*	*ACO*	*ABC*	*PSO*	*FA*	*MSA*	*BFO*
		t, с							
1	500	10	16	11	11	12	13	14	15
2	1000	28	36	32	32	30	34	34	36
3	2500	44	52	48	48	48	50	50	51
4	5000	114	132	121	122	120	125	127	130
5	7500	163	182	174	174	172	176	176	180
6	10000	246	285	267	268	264	270	270	282
7	20000	298	340	326	328	324	330	332	334
8	30000	354	405	377	379	371	385	387	396
9	40000	426	482	455	457	444	464	466	470
10	50000	525	584	549	550	545	561	565	571

Таблица 2

Сравнение результатов размещения тестовых графов алгоритмами *ES, GA, ACO, ABC, PSO, FA, MSA* и *BFO* по значению оценочной функции Δ

№ теста	Число вершин графа	*ES*	*GA*	*ACO*	*ABC*	*PSO*	*FA*	*MSA*	*BFO*
		Δ, %							
1	500	6,9	7,0	7,1	7,0	7,0	7,2	7,2	7,1
2	1000	7,4	7,5	7,6	7,6	7,6	7,6	7,6	7,6
3	2500	7,7	8,1	8,1	8,1	8,1	8,2	8,2	8,2
4	5000	8,2	8,4	8,5	8,6	8.4	8,7	8,7	8,8
5	7500	8,4	8,6	8,7	8,7	8,7	8,9	8,9	9,0
6	10000	8,5	8,8	8,9	8,9	8,8	9,1	9,3	9,2
7	20000	9,1	9,4	9,3	9,3	9.3	9,5	9,5	9,5
8	30000	9,1	9,4	9,5	9,5	9,5	9,7	9,7	9,8
9	40000	9,2	9,7	9,8	9,8	9,9	10,1	10,1	10,1
10	50000	9,4	9,8	9,8	9,9	9,8	10,1	9,9	9,9

эвристики для данной задачи сопоставимы по качеству и времени решения.

Заключение

Эволюционные биоэвристики моделируют базовые положения в теории биологической эволюции, такие как процессы воспроизводства, мутации и селекции. Поведение агентов в популяции определяется окружающей средой. Популяция эволюционирует по правилам естественного отбора в соответствии с фитнесс-функцией, задаваемой окружающей средой. Каждому агенту популяции назначается значение его пригодности в окружающей среде. Большие шансы на размножение получают наиболее пригодные виды. Рекомбинация и мутация позволяют агентам изменяться и адаптироваться к среде. Такие алгоритмы относятся к адаптивным поисковым механизмам и успешно используются, например, для решения разнообразных оптимизационных задач и представляются на математическом языке.

Роевые биоэвристики моделируют коллективное поведение децентрализованной самоорганизующейся системы. Как правило, модель роевого интеллекта включает множество агентов, локально взаимодействующих между собой и с окружающей средой. Каждый агент следует простым правилам и, несмотря на отсутствие централизованного управления поведением агентов, локальные и случайные взаимодействия приводят к возникновению интеллектуального группового поведения, не контролируемого отдельными агентами.

Список литературы

1. **Wolpert D., Macready W.** The no free lunch theorems for optimization // IEEE Trans. Evol. Comp. 1997. N. 1. P. 67—82.
2. **Родзин С. И., Курейчик В. В.** Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации (обзор) // Кибернетика и программирование. 2017. № 2. С. 66—74.
3. **Cuevas E., Díaz Cortés M., Oliva Navarro D.** Advances of Evolutionary Computation: Methods and Operators. Springer International Publishing, 2016. 214 p.
4. **Storn R., Price K.** Differential evolution a simple and efficient heuristic for global optimization over continuous spaces // Glob. Optim. 1997. No. 11(4). P. 341—359.
5. **Rodzin S. I.** Schemes of evolution strategies // Proc. IEEE Int. Conf. on Artificial Intelligence Systems (ICAIS'2002). 2002. P. 375—380.
6. **Holland J.** Iterative circuit computers // Proc. Western Joint Comp. Conf. 1960. P. 259—265.
7. **Koza J.** Genetic Programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA. 1992. 813 p.
8. **Carci A.** Theory of Saplings Growing Up Algorithm // Proc. Int. Conf. ICANNGA. 2007. P. 450—460.
9. **Mehrabian A., Lucas C.** A novel numerical optimization algorithm inspired from weed colonization // Ecological Informatics. 2006. no. 1(4). P. 355—366.
10. **Dorigo M., Blum C.** Ant colony optimization theory: a survey // Theor. Comput. Sci. 2005. Vol. 344(2—3). P. 243—278.
11. **Karaboga D., Basturk B.** On the performance of artificial bee colony (ABC) algorithm // Appl. Soft Comput. 2008. N. 8(1). P. 687—697.
12. **Yang X.** A new metaheuristic bat-inspired algorithm // Stud. Comput. Intell. 2010. Vol. 284. P. 65—74.
13. **Askarzadeh A.** A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm // Comput. Struct. 2016. Vol. 169. P. 1—12.
14. **Yang X., Deb S**. Cuckoo search via Lévy flights // Proc. World Congress on Nature and Biologically Inspired Computing (NABIC2009). 2009. P. 210—214.
15. **Yang X.** A new metaheuristic bat-inspired algorithm // Stud. Comput. Intell. 2010. Vol. 284. P. 65—74.
16. **Yang X.** Flower Pollination Algorithm for Global Optimization // Lecture Notes in Comp. Scien. 2012. vol. 7445 LNCS. P. 240—249.
17. **Yang X., Xuyan T., Chen J.** Algorithm of Marriage in Honeybees Optimization Based on the Wolf Pack Search // Proc. on Intell. Pervasive Comp Conf. 2007. P. 462—467.
18. **Gandomi A., Alavi A.** Krill herd: a new bio-inspired optimization algorithm // Commun. Nonlinear Sci. Numer. Simul. 2012. N. 17(12). P. 4831—4845.
19. **Mirjalili S.** Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm // Knowl. based Syst. 2015. Vol. 89. P. 228—249.
20. **Kennedy J., Eberhart R.** Particle Swarm Optimization // Proc. IEEE Int. Conf. on Neural Networks. 1995. Vol. 4. P. 1942—1948.
21. **Cuevas E. et. al**. A swarm optimization algorithm inspired in the behavior of the social spider // Expert Syst. Appl. 2013. Vol. 40(16). P. 6374—6384.

22. **Mirjalili S., Lewis A.** The whale optimization algorithm // Adv. Eng. Softw. 2016. Vol. 95. P. 51—67.

23. **Passino K.** Biomimicry of bacterial foraging for distributed optimization and control // IEEE Control Syst. Magaz. 2002. Vol. 22 (3). P. 52—67.

24. **Filho B. et al**. A novel search algorithm based on fish school behavior // Proc. IEEE Int. Conf. SMC. 2008. P. 2646—2651.

25. **Eusuff M., Lansey K.** Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm // J. Water Resour. Plan. Manag. 2003. Vol. 129. P. 10—25.

26. **Zhao R., Tang W.** Monkey Algorithm for Global Numerical Optimization // J. of Uncertain Systems. 2008. Vol. 2, no. 3. P. 165—176.

27. **Kureichik Vl., Kuliev E., Kureichik V.** Monkey search algorithm for ECE components partitioning // J. of Physics: Conf. Series. 2018. Vol. 1015 (4), no. UNSP 042026. P. 1—10.

28. **Kureichik Vl., Kuliev E., Kureichik V.** Mechanisms of swarm intelligence and evolutionary adaptation for solving PCB design tasks // Proc. Int. Sem. on Electron Devices Design and Production (SED). 2019. P. 109—113.

V. V. Kureychik, Professor, e-mail: vkur@sfedu.ru, **S. I. Rodzin**, Professor, e-mail: srodzin@yandex.ru,
Southern Federal University, Taganrog, Russian Federation

Computational Models of Evolutionary and Swarm Bio Heuristics (Review)

Computational models of evolutionary and swarm algorithms using nature-inspired mechanisms of self-organization and learning are presented. Experimental results are presented for the problem of placing a graph on a plane with the minimum total length of the graph edges.

Keywords: *bio heuristics, convergence of the algorithm, exploration versus exploitation, population, fitness assignation, optimization, graph*

Acknowledgements: The reported study was tunded by RFBR according to the research project № 19-07-00570.

DOI: 10.17587/it.27.507-520

References

1. **Wolpert D., Macready W.** The no free lunch theorems for optimization, *IEEE Trans. Evol. Comp.,* 1997, no. 1, pp. 67—82.

2. **Rodzin S. I., Kureychik V. V.** Theoretical issues, and modern problems of development of cognitive bioinspired optimization algorithms (review), *Kibernetika i programmirovanie*, 2017, no. 2, pp. 66—74 (in Russian).

3. **Cuevas E., Díaz Cortés M., Oliva Navarro D.** Advances of Evolutionary Computation: Methods and Operators, Springer International Publishing, 2016, 214 p.

4. **Storn R., Price K.** Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces, *Glob. Optim.,* 1997, no. 11(4), pp. 341—359.

5. **Rodzin S.** Schemes of evolution strategies, *Proc. IEEE Int. Conf. on Artificial Intelligence Systems (ICAIS'2002),* 2002, pp. 375—380.

6. **Holland J.** Iterative circuit computers, *Proc. Western Joint Comp. Conf.,* 1960, pp. 259—265.

7. **Koza J.** Genetic Programming: on the programming of computers by means of natural selection, MIT Press, Cambridge, MA, 1992, pp. 813.

8. **Carci A.** Theory of Saplings Growing Up Algorithm, *Proc. Int. Conf. ICANNGA*, 2007, pp. 450—460.

9. **Mehrabian A., Lucas C.** A novel numerical optimization algorithm inspired from weed colonization, *Ecological Informatics*, 2006, no. 1(4), pp. 355—366.

10. **Dorigo M., Blum C.** Ant colony optimization theory: a survey, *Theor. Comput. Sci.,* 2005, vol. 344(2—3), pp. 243—278.

11. **Karaboga D., Basturk B.** On the performance of artificial bee colony (ABC) algorithm, *Appl. Soft Comput.,* 2008, no. 8(1), pp. 687—697.

12. **Yang X.** A new metaheuristic bat-inspired algorithm, *Stud. Comput. Intell.,* 2010, vol. 284, pp. 65—74.

13. **Askarzadeh A.** A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, *Comput. Struct.,* 2016, vol. 169, pp. 1—12.

14. **Yang X., Deb S.** Cuckoo search via Lévy flights, *Proc. World Congress on Nature and Biologically Inspired Computing (NABIC'2009),* 2009, pp. 210—214.

15. **Yang X.** A new metaheuristic bat-inspired algorithm, *Stud. Comput. Intell.,* 2010, vol. 284, pp. 65—74.

16. **Yang X.** Flower Pollination Algorithm for Global Optimization, *Lecture Notes in Comp. Scien.*, 2012, vol. 7445 LNCS, pp. 240—249.

17. **Yang X., Xuyan T., Chen J.** Algorithm of Marriage in Honeybees Optimization Based on the Wolf Pack Search, *Proc. on Intell. Pervasive Comp Conf.,* 2007, pp. 462—467.

18. **Gandomi A., Alavi A.** Krill herd: a new bio-inspired optimization algorithm, *Commun. Nonlinear Sci. Numer. Simul.,* 2012, no. 17(12), pp. 4831—4845.

19. **Mirjalili S.** Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl. based Syst.,* 2015, vol. 89, pp. 228—249.

20. **Kennedy J., Eberhart R.** Particle Swarm Optimization, *Proc. IEEE Int. Conf. on Neural Networks,* 1995, vol. 4, pp. 1942—1948.

21. **Cuevas E. et. al**. A swarm optimization algorithm inspired in the behavior of the social spider, *Expert Syst. Appl.*, 2013, vol. 40(16), pp. 6374—6384.

22. **Mirjalili S., Lewis A.** The whale optimization algorithm, *Adv. Eng. Softw.*, 2016, vol. 95, pp. 51—67.

23. **Passino K.** Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst. Magaz.*, 2002, vol. 22 (3), pp. 52—67.

24. **Filho B.** et al. A novel search algorithm based on fish school behavior, *Proc. IEEE Int. Conf. SMC.*, 2008, pp. 2646—2651.

25. **Eusuff M., Lansey K.** Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm, *J. Water Resour. Plan. Manag.,* 2003, vol. 129, pp. 10—25.

26. **Zhao R., Tang W.** Monkey Algorithm for Global Numerical Optimization, *J. of Uncertain Systems,* 2008, vol. 2, no. 3, pp. 165—176.

27. **Kureichik Vl., Kuliev E., Kureichik V.** Monkey search algorithm for ECE components partitioning, *J. of Physics: Conf. Series,* 2018, vol. 1015 (4), no. UNSP 042026, pp. 1—10.

28. **Kureichik Vl., Kuliev E., Kureichik V.** Mechanisms of swarm intelligence and evolutionary adaptation for solving PCB design tasks, *Proc. Int. Sem. on Electron Devices Design and Production (SED)*, 2019, pp. 109—113.

УДК 621.371.2; 621.391.82: 519.25 DOI: 10.17587/it.27.521-530

О. Н. Маслов, д-р техн. наук, проф., e-mail: maslovpsuti@yandex.ru,
Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Защита стационарного объекта от массированного воздействия мобильных объектов в условиях смешанной игры по фон Нейману

Методом статистического имитационного моделирования (СИМ) выполнен анализ условий работы и эффективности функционирования системы физической защиты стационарного объекта от массированного воздействия беспилотных летательных аппаратов (дронов). Показано, что условия решаемой задачи соответствуют рефлексивной версии двусторонней смешанной игры по фон Нейману. Определены статистические риск-ориентированные характеристики двух вариантов реализации системы защиты объекта: использующих силовые механические и электромагнитные воздействия на "облако дронов". Проиллюстрированы возможности и представлены перспективы использования результатов, полученных с применением метода СИМ.

Ключевые слова: *физическая защита объекта, беспилотные летательные аппараты, модель "облако дронов", механическая и электромагнитная защита, метод статистического имитационного моделирования, характеристики системы защиты*

Введение

Применение теории игр в интересах разработки онтологических моделей для решения проблем общего характера сопровождается сегодня конкретными предложениями, связанными с актуальными научно-технологическими задачами. С одной стороны, это способствует развитию методов системного анализа в рамках теории моделирования сложных систем (СС) [1—5], теории игр [6—8], ожидаемой полезности [9], с другой стороны, помогает достаточно просто и эффективно решать задачи в предметных областях, чрезвычайно важных в настоящее время [10—14]. Одной из таких задач является анализ условий и эффективности функционирования систем защиты объектов различного назначения, выполняемый с помощью новых информационных технологий, в частности, метода статистического имитационного моделирования (СИМ) [2, 3, 15—19]. В связи с широким внедрением в бизнес, производство, военное дело и т. д. беспилотных летательных аппаратов (дронов) [20—25] представляет интерес проблема обеспечения физической безопасности объекта стационарного базирования от воздействия совокупности малогабаритных мобильных объектов в виде роевидного "облака дронов" (далее без кавычек), кратко представленная в работе [26].

С точки зрения теории игр рассматриваемая конфликтная ситуация соответствует игре двух участников с противоположными (антагонистическими) интересами [27—30], которую в соответствии с работами [6—8] будем именовать двусторонней игрой по фон Нейману. Стороны конфликта представляют собой СС нерефлекторного (по определению Н. Н. Моисеева [2]) типа, неотъемлемыми компонентами которых является человеческий фактор (персонал, личный состав, население и др.). Условимся, что первым участником конфликта является СС-инициатор, начинающий агрессивные действия в отношении объекта, принадлежащего второму участнику, — СС-мишени [26].

Цель статьи — анализ методом СИМ условий и эффективности функционирования системы защиты объекта стационарного типа, принадлежащего СС-мишени, от воздействия облака дронов, управляемых СС-инициатором, при смешанной игре по фон Нейману, которая соответствует модели рассматриваемой ситуации.

Характеристика игры

Поскольку целью СС-инициатора является воздействие на объект, а целью СС-мишени — предотвращение этого воздействия, ближайшим известным аналогом данной игры является бесконечная антагонистическая игра на пространстве стратегий $A \times B$ с выигрышами H как типовой вариант игры по фон Нейману. Согласно работе [7] в игре по фон Нейману (A, B, H) речь идет о существовании хотя бы одного выигрыша H в любой $A \times B$ ситуации, в том числе стохастического типа. В последнем случае чистые стратегии A, B трансформируются в смешанные стратегии F, G через вероятностные распределения для A, B, что означает переход к смешанной стратегии игры, где выигрыши H становятся неопределенными (стохастическими).

Владельцем объекта Q (рис. 1) является СС-мишень, располагающая конечным мно-

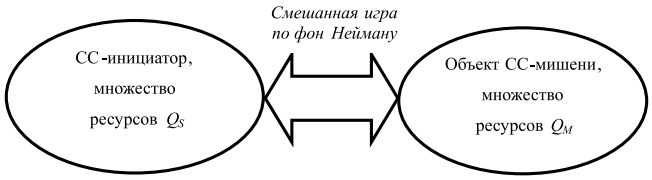


Рис. 1. Ситуация решаемой задачи

жеством ресурсов Q_M для его защиты. Соответственно, у СС-инициатора имеется множество ресурсов Q_S, посредством которых он собирается достичь поставленной цели. Бесконечность игры означает, что жестких исходных ограничений на ее продолжительность нет, хотя в реальности она может быть остановлена как по решению СС-инициатора при достижении или отказе от достижения поставленной цели, так и ввиду истощения ресурсов Q_S и Q_M.

Совокупность выигрышей H будем трактовать как множество полезностей, присущих игре (A, B, H), включая случай (F, G, H). B отличие от рассмотренного в работе [6] случая это множество распадается на два: $V(H_S)$ и $V(H_M)$ — относящихся, соответственно, к СС-инициатору и СС-мишени вследствие принципиального различия их интересов. Это приводит к двум разным взглядам на задачи и цели, текущий ход и ожидаемый результат игры, которая, в то же время, остается их общей игрой. Смешивание стратегий осуществляется игроками самостоятельно, поэтому учет какой-то общей меры перекрытия $V(H_S)$ и $V(H_M)$ теряет смысл, хотя объективно такая мера существует, поскольку конечные результаты игры для обоих участников взаимозависимы и однозначно соответствуют друг другу.

Расщепление множеств на более мелкие части означает увеличение объема информации, которой располагают игроки, и ведет к расширению множества $F\times G$. При полном расщеплении F, G переходят в чистые стратегии A, B — предельному случаю здесь соответствует аксиома определенности, утверждающая, что "для абсолютно осведомленных в правилах и тонкостях игры участников ее исход предопределен, игра становится чисто комбинаторной" [7]. Хотя первая часть аксиомы далека от реальности, вторая часть содержит подсказку, которую можно перенести на любой частный случай: использовать вероятностную комбинаторику в дополнение к аналитическому расчету значений H при неопределенности знаний участников смешанной игры по фон Нейману о состоянии и возможном ее исходе. Отметим также, что при динамическом моделировании СС функция воздействия на противника часто интерпретируется как плотность огня: "пулеметчик" использует последовательность выстрелов (в нашем случае эти серии могут выполняться как друг за другом, так и одновременно); "снайпер" выбирает единственный момент на заданном промежутке времени.

Защита с применением силовых воздействий

Рассмотрим первый типовой вариант, связанный с применением M силовых (механических) воздействий на роевидное облако, состоящее из N дронов, приближающихся к объекту защиты Q (рис. 2). Ситуация соответствует игре между СС-инициатором (владельцем ресурсов Q_S в виде облака дронов, цель которого состоит в преодолении зоны силового воздействия), и СС-мишенью, осуществляющим это воздействие посредством ресурсов Q_M. В распоряжении СС-мишени, согласно рис. 2, имеется многоканальный "пулемет", способный в общем случае M_n раз воздействовать на каждый из N дронов в составе облака.

Начнем с определения субъективного негативного риска со стороны СС-инициатора, связанного с потерей его ресурсов, который одновременно является позитивным риском для СС-мишени (отметим, что далее также имеет смысл периодически уточнять, какой критерий используется, для кого и с какой целью). Элементарным событием здесь является m-е однократное воздействие СС-мишени на одиночный дрон. Риск в виде ущерба по ресурсам воздействия, связанный с этим событием, представляет собой величину

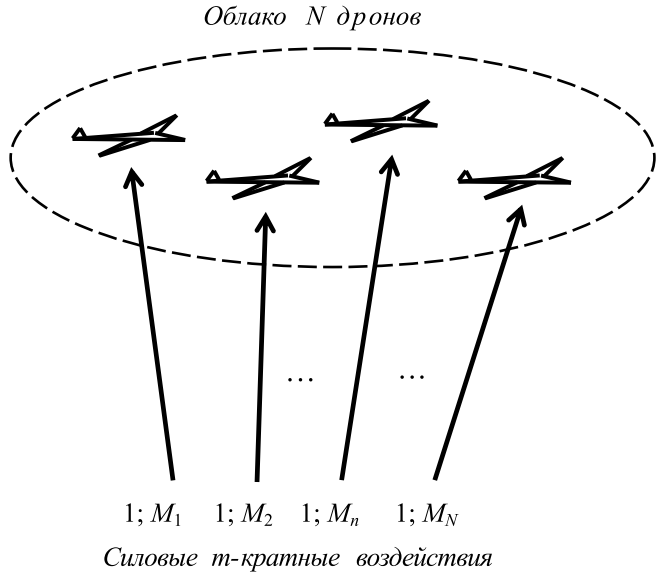


Рис. 2. Первый вариант реализации системы защиты объекта

$$R(m) = P_0^{m-1}(1 - P_0)mA_S, \quad (1)$$

где P_0 — ожидаемая вероятность промаха (неудачного воздействия); m — номер воздействия, m [1; M]; A_S — "стоимость" первого воздействия (далее без кавычек), которая затем возрастает прямо пропорционально m в связи с тем, что шансы дрона преодолеть зону воздействия возрастают.

Суммарный средний риск для M воздействий СС-мишени на один дрон соответствует сумме $R_S(M) = \sum_{m=1}^{M} R(m)$, которую представим как

$$R_S(M) = A_S[(1 - P_0)/P_0] \sum_{m=1}^{M} mP_0^m,$$

чтобы с учетом результатов работы [31] получить

$$R_S(M) = A_S[(1 - P_0)/P_0] \quad + MP_0 - M - \quad P_0^M \quad - P_0 \quad (2)$$

При воздействии на все облако

$$R_S(N; M) = \\ = A_S N [1 + (MP_0 - M - 1)P_0^M]/(1 - P_0). \quad (3)$$

Позитивный риск для СС-инициатора определяется шансами на преодоление зоны воздействия дроном с номером n [1; N] и облаком из N дронов. Риск, связанный с преодолением зоны одним дроном, равен

$$R_D(m) = P_0^m m B_S, \quad (4)$$

где B_S — стоимость первого преодоления, которая затем растет прямо пропорционально m, поскольку шансы дрона преодолеть зону воздействия далее возрастают по аналогии с A_S. По аналогии с (2)—(3) получаем средний позитивный риск для всего облака в наилучшем случае:

$$R_D(N; M) = R_S(N; M)\, B_S P_0 / A_S (1 - P_0). \quad (5)$$

Анализ показывает, что содержание (3) и (5) определяет функция

$$R_S(M) = \quad MP_0 - M - \quad P_0^M \quad - P_0 \quad (6)$$

через которую для заданных A_S, B_S и N однозначно выражаются риски $R_S(N; M)$ и $R_D(N; M)$. Из соотношений (3) и (5) следует также

$$R_D(N; M)/R_S(N; M) = B_S P_0 / A_S (1 - P_0). \quad (7)$$

Субъективные оценки стоимостных показателей A_S и B_S доступны только самим игрокам, да и то порознь. Без ущерба для дальнейшего в выражении (7) можно принять $A_S \approx B_S$, что дает $R_D(N; M)/R_S(N; M) = P_0/(1 - P_0)$. Таким образом, при оценке эффективности защиты объекта от массированного воздействия дронов ключевым параметром является вероятность промаха СС-мишени P_0, поскольку

— если $P_0 \ll 1$, то $R_D(N; M) \ll R_S(N; M)$, что плохо для СС-инициатора и хорошо для СС-мишени;

— если $P_0 \approx 1/2$, то $R_D(N; M) \approx R_S(N; M)$, что означает "баланс" рисков, который не устраивает никого из игроков и переносит их действия в следующую стадию противостояния;

— если $P_0 > 1/2$, то чем больше вероятность промаха СС-мишени, тем позитивный риск для СС-инициатора $R_D(N; M)$ будет больше негативного риска $R_S(N; M)$ что для СС-мишени недопустимо.

Количественную оценку влияния вероятности промаха P_0 и допустимого числа M применения ресурсов СС-мишени на функцию $R_S(M)$, через которую для заданных A_S, B_S и N выражаются риски $R_S(N; M)$ и $R_D(N; M)$, демонстрирует табл. 1. Видно, во-первых, что с увеличением M происходит асимптотическое приближение функции риска $R_S(M)$ к значениям, выделенным жирным шрифтом, поэтому при $P_0 = 0{,}01$ увеличение $M > 2$; при 0,05 — $M > 3$; при 0,10 — $M > 4$ и т. д. не имеют смысла. Во-вторых, при маловероятных промахах, когда $P_0 \ll 1$, "насыщение" функции риска происходит при $M < 3$, т. е. для поражения одного дрона достаточно 1...2 воздействий. По мере увеличения P_0 динамика рисков меняется, и необходимое число воздействий M может оказаться достаточно большим, что соответствует эвристическим представлениям о рассматриваемой ситуации.

В-третьих, переход от $R_S(M)$ к рискам $R_S(N; M)$ и $R_D(N; M)$ показывает, что эффективная защита объекта возможна только при $Q_M \gg N$, что представляется важным ограничением для СС-мишени. В реальных услови-

Таблица 1

Нормированная функция риска $R_S(M)$ для первого варианта реализации системы защиты объекта

P_0	M									
	1	2	3	4	5	6	7	8	9	10
0,01	0,99	**1,01**								
0,05	0,95	1,045	**1,05**							
0,1	0,90	1,08	1,11	**1,11**						
0,15	0,85	1,105	1,16	1,17	**1,18**					
0,20	0,80	1,12	1,22	1,24	1,245	**1,25**				
0,25	0,75	1,125	1,27	1,31	1,32	1,33	**1,33**			
0,5	0,50	1,00	1,37	1,62	1,78	1,87	1,93	1,96	1,98	1,99

ях при сопоставимых Q_M и N позитивный для СС-инициатора риск $R_D(N; M)$ всегда отличается от нуля, что говорит о несовершенстве системы защиты, и увеличивается с ростом N. В-четвертых, очевидно, что полученные данные соответствуют чистым стратегиям игры A, B, когда все параметры функций риска считаются известными, хотя более реалистичны смешанные стратегии F, G, приводящие к случайным параметрам. На реализации процедуры смешивания путем применения метода СИМ совместно с компьютерной версией метода Монте-Карло остановимся более подробно.

Моделирование риска при силовых воздействиях

В нашем случае переход от чистых стратегий игры A, B к смешанным стратегиям F, G означает учет стохастичности числовых параметров модели (1)—(7), которая обусловлена неопределенностью знаний игроков об их свойствах. В рамках метода СИМ для каждого из таких параметров необходимо определить (задать объективным или субъективным способом) финитный закон распределения в области от минимально-возможного до максимально-возможного значения. Кроме того, следует обобщить условия определения параметров модели (1)—(7): например, если ожидаемая вероятность промаха (неудачного воздействия) для N дронов будет равна

$$S_0 = \prod_{n=1} P_{0n},$$

где P_{0n} $[P_1; P_2]$ — вероятность неудачного воздействия на один дрон; n [1; N] — "номер" дрона в составе облака, то аналог (1) представляет собой выражение

$$R(m) = \left(1 - \prod_{n=1}^{N} P_{0n}\right)\left(\prod_{n=1}^{N} P_{0n}\right)^{m-1} mA_S \text{ и т. д. (8)}$$

Здесь m [1; M] — "номер" удачного воздействия.

Детерминированными исходными данными при проведении СИМ случайных значений $R(m)$ являются: $N_R \geqslant 10^3$ — число разыгрываний по методу Монте-Карло случайных величин, соответствующее размеру одномерного массива, необходимого для построения одной гистограммы значений $R(m)$; S_R — число интервалов на горизонтальной оси $R(m)$, необходимое для построения гистограмм; M — максимально-возможное число воздействий; N = 1, 2, 3, ... — число дронов в облаке, равное числу гистограмм $R(m)$, где каждому N соответствует своя гистограмма; n [1; N] — "номера" вероятностей P_{0n}, которые определяются значениями N для каждого варианта; вероятности $P_1 < P_2$ и стоимостные параметры масштаба $A_1 < A_2$ (см. далее).

При формировании гистограмм $R(m)$ используются следующие случайные данные: значения m, разыгрываемые по методу Монте-Карло в пределах [1; M]; упомянутые n [1; N]; вероятности P_{0n}, разыгрываемые в пределах $[P_1; P_2]$; значения A_S, которые устанавливают масштаб гистограммы риска, разыгрываемые в пределах $[A_1; A_2]$. Законы распределения для n, m, P_{0n} и A_S на основании принципа безразличия Лапласа примем равномерными, что соответствует максимальной неопределенности знаний игроков о вероятностных свойствах указанных параметров. Результаты СИМ в виде гистограмм нормированного риска $R(m)$ как ущерба по ресурсам СС-инициатора, соответствующего $R_S(N; M)$, при тестовых значениях N_R = 2000; M = 2; N = 5; P_1 = 0,01; P_2 = 1,00; A_1 = 0 и A_2 = 15 в качестве примера приведены на рис. 3.

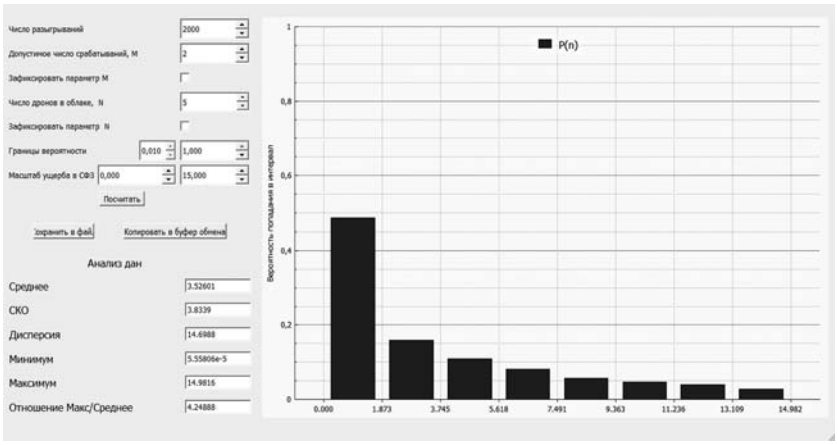


Рис. 3. Результаты СИМ нормированной функции риска $R(m)$ для первого варианта реализации системы защиты объекта

При многофакторном анализе обычно принято фиксировать одни параметры и анализировать влияние других (табл. 1). Возможности метода СИМ позволяют не делать этого и получать статистические оценки результатов моделирования в условиях, когда все задействованные параметры случайны. Табл. 2 отражает влияние параметра M, с негативной стороны оценивающего "меткость" силовых ресурсов СС-мишени, на статистические оценки параметра $R_S(M)$, определяющего согласно (3) и (6) риск $R_S(N; M)$. В данном случае n [1; 20]; m [1; M]; P_{0n} [0,01; 0,25]; A_S [50; 100].

Таблица 2

Влияние на статистические оценки $R_S(M)$ параметра M

Оценки	M							
	1	2	3	4	5	6	8	10
$R_S(M)_{\text{ср}}$	75,4	37,7	23,3	18,7	14,4	12,3	8,8	6,8
СКО	14,5	37,9	34,9	33,1	29,6	27,7	24,2	21,9

Таблица 3

Влияние на средний риск $R_D(N; M)$ числа дронов в облаке N

Показатели	N							
	100	75	50	25	20	15	10	5
$R_S(M)_{\text{ср}}$	23,8	24,9	25,7	24,4	24,8	25,1	26,8	23,7
СКО	35,6	35,6	36,75	35,6	36,0	35,8	36,2	33,4
$R_D(N; M)_{\text{ср}}$	240	180	120	60	48	36	24	12

Видно, что с ростом M среднее значение $R_S(N; M)$ резко снижается, что, с одно стороны, безусловно хорошо для СС-инициатора и плохо для СС-мишени, а с другой стороны, соответствует данным табл. 1.

Табл. 3 демонстрирует влияние на позитивный для СС-инициатора риск $R_D(N; M)$ числа дронов N, убывающего вследствие защитных действий СС-мишени. Здесь n [1; N]; m [1; 3]; P_{0n} [0,01; 0,25]; A_S [50; 100] и $B_S = 10$. Видно, что двадцатикратному уменьшению N соответствует аналогичное снижение среднего риска $R_D(N; M)$, что хорошо для СС-мишени и плохо для СС-инициатора.

В заключение подсчитаем ресурсы, обладателями которыми остаются СС-инициатор и СС-мишень по ходу и после завершения игры. Первый сбитый дрон СС-инициатора согласно (2) "стоит" M_1 ресурсов СС-мишени, второй дрон — M_2, n-й дрон — M_n, последний дрон с номером $N_T - M_N$. Таким образом, в распоряжении СС-мишени остается $M_O = M - M_T \geqslant M_{\text{доп}}$ ресурсов, где $M_{\text{доп}}$ — минимально-допустимое для продолжения игры их число; $M_T = \sum_{n=1}^{N_T} M_n$. Аналогичным образом остаток ресурсов для СС-инициатора есть $N_O = N - N_T \geqslant N_{\text{доп}}$.

Далее ограничимся выводом, что продолжение игры, а также ее исход существенно зависят от того, насколько успешно игроки спрогнозируют с применением всех доступных методов и средств следующие недостающие сведения:

— число потерянных (сбитых) дронов N_T, в принципе известное обоим игрокам;

— остаток дронов $N - N_T$, известный СС-инициатору и неизвестный СС-мишени;

— объем затраченных ресурсов M_T, также в принципе известный обоим игрокам;

— остаток ресурсов $Q_M - M_T$, известный СС-мишени и неизвестный СС-инициатору и т. д.

Электромагнитная защита объекта

Вторым типовым вариантом защиты является формирование в полусферическом пространстве над объектом электромагнитного поля (ЭМП), промодулированного информационными сигналами, способными выводить дроны СС-инициатора из строя. По классификации преднамеренных активных помех такими сигналами могут быть как заградительные шумовые (шумоподобные) помехи, так и импульсные прицельные (в том числе имитирующие) помехи специального вида [27—30].

Ситуацию иллюстрирует рис. 4, где показаны объект защиты Q и расстояние r_M, соответствующее условному "нулевому" (минимально-допустимому для СС-инициатора) риску. Идеальным является выполнение на всей полусфере площадью $S_0 = 2\pi r_M^2$ (телесный угол $\Omega_0 = 2\pi$) условия $\Pi_S \geqslant \Pi_0$, где Π_S — плотность потока мощности (энергии в единицу времени, далее *ППЭ*) преднамеренных активных помех; Π_0 — уровень *ППЭ*, необходимый для вывода из строя элементов облака. В реальности это условие выполняется на части площади полусферы S_M, при этом отношение $\chi_S = S_M/S_0$ может быть расчетным критерием эффективности ЭМП-защиты объекта.

Если принять во внимание, что согласно работе [20] управление дронами СС-инициатор осуществляет на частотах выше 2,4 ГГц (длины волн короче 1,25 м), то излучающую систему СС-мишени можно реализовать в виде $M = 6$ или 8 мобильных радиокомплексов (МРК) с характеристиками направленности (ХН) по уровню половинной мощности в горизонтальной плоскости шириной порядка 60° или 45° (рис. 5) при их "веерной" структуре в вертикальной плоскости (см. рис. 4), где ши-

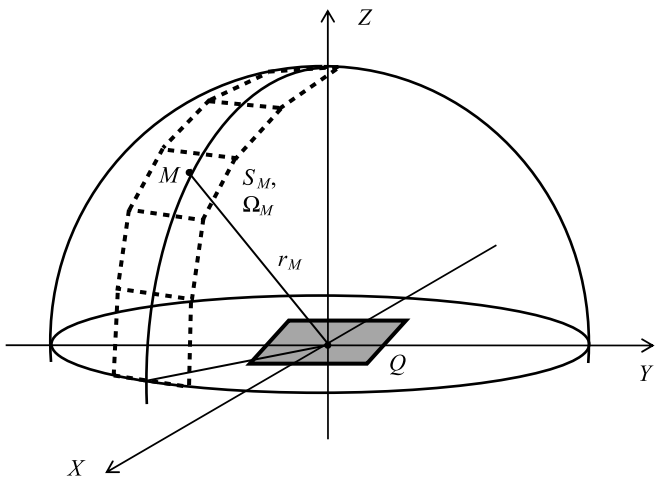


Рис. 4. Реализация второго варианта системы защиты объекта

рина парциальных ХН определяется требуемым значением коэффициента усиления антенной системы. Фрагменты "веера" ХН на рис. 4 условно обозначены точечными линиями, фрагмент с центром точке M удален на расстояние R_M и "закрывает" на полусфере площадку S_M, которая соответствует телесному углу Ω_M. Форма площадок S_M и возможность их взаимного пересечения не играют роли, если энергетическое условие $\Pi_S \geqslant \Pi_0$ при этом выполняется.

Рис. 5 иллюстрирует использование МРК для постановки помех в пределах телесного угла $\Omega_S = \Omega_1 + \Omega_2 + ... + \Omega_k + ... + \Omega_K$ с учетом возможности их приведения к общему центру (см. рис. 4). Поскольку в условиях задачи фигурируют уровни *ППЭ*, представление излучающей системы СС-мишени в виде активной фазированной антенной решетки и определение ее энергетического потенциала [15] не имеют смысла. Ввиду неопределенной поляризации сигналов, управляющих дронами, излучатели СС-мишени должны обеспечивать их "закрытие" на S_M по всем ортогональным составляющим векторов ЭМП.

Применение МРК предпочтительно по сравнению с постоянным обеспечением условия $\Pi_S \geqslant \Pi_0$ на поверхности полусферы, как это делается, например, при защите компьютерных систем [17—18, 32] или демонстрируется в фильмах о космических войнах, во-первых, по технологическим и экономическим причинам, во-вторых, с учетом возможности маневрирования ресурсами СС-мишени. По ходу игры СС-инициатор может использовать разные пространственно-временные схемы воздействия, включая стохастические и псевдохаотические модели построения облака, пытаясь найти изъяны в системе ЭМП-защиты и направляя в эти места сконцентрированное движение дронов.

Система защиты объекта призвана должным образом, оперативно и эффективно реагировать на них. В связи с этим представляет интерес рефлексивный вариант игры по фон Нейману [7, 33], когда противоборствующие стороны как бы имитируют друг друга с целью понять и предугадать действия противника, а затем наилучшим образом воздействовать на него для предотвращения возможного ущерба.

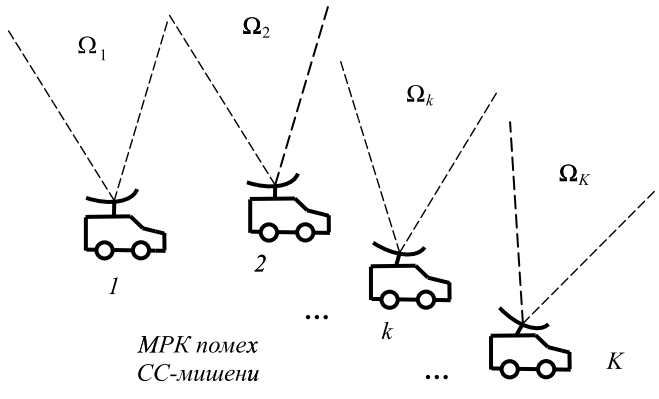


Рис. 5. Перекрытие заданного телесного угла Ω_0 посредством совокупности МРК

Моделирование риска при электромагнитной защите

Будем исходить из того, что в исходном состоянии эффективность системы ЭМП-защиты объекта соответствует установленным нормам, и вероятность ее преодоления элементами облака, с точки зрения СС-мишени, допустимо мала. Поэтому СС-инициатору необходимо оказать предварительное силовое воздействие на МРК для вывода хотя бы части их из строя и образования в структуре ЭМП K "коридоров" (далее без кавычек) от точки M на рис. 4 до объекта Q. Моделировать функцию риска далее будем по аналогии с соотношениями (1)—(8).

Прежде всего примем, что вероятность формирования k-го коридора равна $P_k\,(S_k/S_0) = (1 - S_k/S_0)^{\alpha}$, где S_k — оговоренная площадь части полусферы S_0, где выполняется условие $\Pi_S \geqslant \Pi_0$; α — коэффициент, учитывающий неопределенность принятой записи $P_k(S_k/S_0)$; k [1; K]. Анализ показывает, что в данном случае риск определяется динамикой движения дронов по коридорам и зависит от случайного аргумента r_k/r_M (расстояния r_k и r_M удобнее отсчитывать от точки M на рис. 4), который определяет две ключевые вероятности:

— вероятность успешного продвижения элементов облака по коридору длиной r_M на расстояние r_k, которую согласно (1)—(8) можно представить экспоненциальной моделью вида

$$P_k(U_k/U_0) = 1 - \exp(-U_k r_k / U_0 r_M),$$

где U_k и r_k — соответственно, скорость движения и текущее расстояние, пройденное по k-му коридору; U_0 — скорость для варианта, условно принятого за исходный "ноль" при проведении СИМ [34—36];

— вероятность $P_k\,(G_k/G_0) = 1 - \exp(G_k r_k/G_0 r_m)$ успешного ударного воздействия на объект после прохождения расстояния r_k, где G_k и G_0 — параметры, оценивающие эффективность оборудования, используемого СС-инициатором для нанесения ущерба объекту, соответственно.

Кроме того, учтем вероятности несущественного и существенного подавления СС-инициатором защитных ресурсов СС-мишени

Q_M. В первом случае в формулу риска входит вероятность $P_k(\Pi_k/\Pi_0) = 1 - (r_k/r_M)^2$, которая отражает изменение риска по мере приближения элементов облака к объекту по k-му коридору за счет роста *ППЭ* и облегчения выполнения условия $\Pi_S \geqslant \Pi_0$ (эта вероятность определяется дистанционной зависимостью *ППЭ*, обратно пропорциональной квадрату расстояния от МРК). После перехода к телесным углам Ω_S и $\Omega_0 = 2\pi$ (рис. 5) позитивный для СС-инициатора и негативный для СС-мишени суммарный риск в итоге представляет собой

$$\begin{aligned} R_D(N) &= A_S n_k K P_k(S_k/S_0) \times \\ &\times P_k(U_k/U_0) P_k(G_k/G_0) P_k(\Pi_k/\Pi_0) = \\ &= A_S n_k K (1 - \Omega_S/\Omega_0)^{\alpha} [1 - (r_k/r_M)^2] \times \\ &\times [1 - \exp(-U_k r_k/U_0 r_M)][1 - \exp(-G_k r_k/G_0 r_M)], \end{aligned} \tag{9}$$

где A_S — множитель масштаба, определяемый категорией ценности объекта Q; n_k — число дронов, движущихся по k-му коридору.

Во втором случае при существенном подавлении защитных ресурсов СС-мишени будем считать, что вместо $P_k(\Pi_k/\Pi_0)$ в формуле риска (9) фигурирует вероятность $P_k(r_k/r_M) = (r_k/r_M)^{\beta}$, где β — параметр неопределенности, аналогичный параметру α в $P_k(S_k/S_0)$. Динамика $P_k(r_k/r_M)$ отражает неопределенный, но непрерывный рост вероятности успешного поражения дронами объекта по мере приближения к нему, когда

$$\begin{aligned} R_D(N) &= A_S n_k K P_k(S_k/S_0) \times \\ &\times P_k(U_k/U_0) P_k(G_k/G_0) P_k(r_k/r_M) = \\ &= A_S n_k K (1 - \Omega_S/\Omega_0)^{\alpha} (r_k/r_M)^{\beta} \times \\ &\times [1 - \exp(-U_k r_k/U_0 r_M)][1 - \exp(-G_k r_k/G_0 r_M)]. \end{aligned} \tag{10}$$

Отметим, что модель (9), (10) не является ни единственно возможной, ни оптимальной по эффективности моделью в рассматриваемой ситуации, однако это не играет особой роли, поскольку всем правилам СИМ [3] она удовлетворяет. Из (9), (10) видно, например, что при $r_k \ll r_M$, когда облако находится вблизи точки M (см. рис. 4), риск отсутствует при любых других параметрах модели. При $\Omega_S = \Omega_0 = 2\pi$ риск также равен нулю независимо от коэффициента неопределенности α. Анализ соотношений (9), (10) показывает также, что при существенном подавлении ресурсов СС-мишени, когда объект становится фактически беззащитным, и при несущественном подавлении, когда СС-мишень продолжает его оборонять, в равных условиях риск должен изменяться по-разному, и это может быть проверено посредством СИМ.

Рост U_k/U_0 и G_k/G_0 ведет к увеличению риска, что отражает влияние эффективности оборудования, используемого СС-инициатором для вывода объекта из строя. Аналогичный эффект дают дополнительные коридоры и число дронов n_k, которые СС-инициатору удастся запустить в эти коридоры. Противоположность интересов в игре по фон Нейману проявляется в стремлении СС-инициатора по возможности увеличить риск (9), (10), а СС-мишени — всеми способами воспрепятствовать этому. Отметим также, что у СС-мишени имеется возможность снизить негативный для себя риск, помимо очевидного $\Omega_S \to 2\pi$, косвенным путем: уменьшая U_k/U_0 и G_k/G_0 за счет укрепления, маскировки и других инженерных мероприятий, проводимых на объекте. Помимо общих параметров, указанных при проведении СИМ согласно (8), детерминированные величины здесь A_S, K, $U = U_k/U_0$ и $G = G_k/G_0$. Случайными исходными данными являются $X = r_k/r_M$, где X [X_1; X_2], включая X_1 = const; n_k [1; N/K], включая n_k = const; $F = S_M/S_0 = \Omega_S/\Omega_0$, где F [F_1; F_2], а также α [α_1; α_2] и β [β_1; β_2], включая α_1 = const и β_1 = const.

Результаты СИМ в виде гистограмм риска $R_D(N)$ представлены на рис. 6 для модели (9) при тестовых значениях параметров $A_S = 50$; $U = 2$; $G = 2$; X [0,5; 1]; F [0,3; 0,999], α [0,1; 1,1]; $K = 1$; $n_k = 5$; число разыгрываний $N_p = 2 \cdot 10^3$. Как и в предыдущем случае силового воздействия (см. табл. 2 и 3), модель позволяет давать количественную оценку эффективности системы защиты объекта по "риск-ориентированным" критериям [34—36]: непосредственно на рис. 6 видно, например, что средний риск равняется 50,3; СКО 35,8 и т. д.

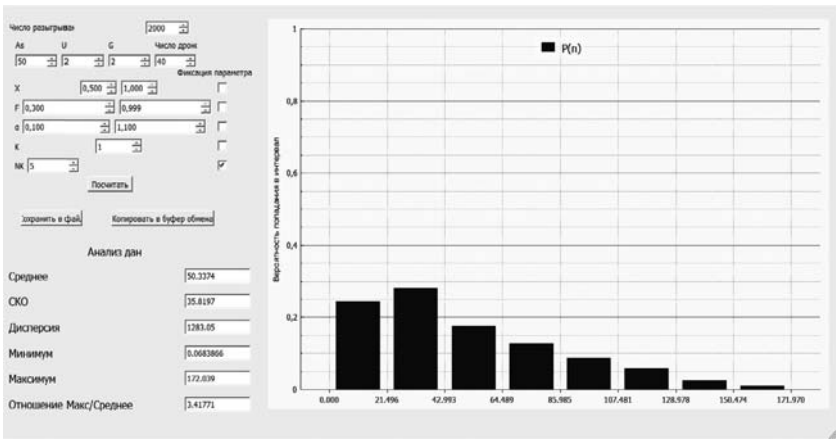


Рис. 6. Результаты СИМ риска $R_D(N)$ для второго варианта реализации системы защиты объекта

Воспользуемся соотношениями (9), (10) для анализа зависимости параметров риска $R_D(N)$ от глубины $X = r_k/r_M$ проникновения элементов облака в область ЭМП-защиты при параметрах F, K и n_k, соответствующих разным ситуациям ее функционирования.

В табл. 4, 5 представлены шесть достаточно подробно исследованных вариантов: № 1; № 2 и № 3 для случая несущественного подавления защитных ресурсов, при значениях параметра F [0,9; 0,999], которые характеризуют систему ЭМП-защиты более высокого качества, поскольку $F = \Omega_S / \Omega_0$; № 4; № 5 и № 6 — для случая существенного подавления ресурсов защиты, когда в системе более низкого качества F [0,5; 0,999]. Для вариантов № 1; № 3; № 4 и № 6 имеет место один коридор: $K = 1$; для вариантов № 2 и № 5 — два коридора: $K = 2$; причем если для вариантов № 1, 2 и № 4, 5 число дронов в коридоре фиксировано, то для вариантов № 3 и № 6 оно является случайным и разыгрывается по методу Монте-Карло (соответствующие значения n_k указаны в боковых столбцах табл. 4, 5). Моделирование, таким образом, охватывает достаточно широкий круг возможных ситуаций, соответствующих смешанной игре по фон Нейману при разных исходных условиях.

Таблица 4

Дистанционная зависимость статистических характеристик риска $R_D(N)$ при числе коридоров K и числе дронов в коридоре n_k для случая несущественного подавления ресурсов ЭМП-защиты: Ω_S / Ω_0 [0,9; 0,999]

Вариант	X_1									
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
№ 1 $K = 1$; $n_k = 10$	4,7 4,1	5,25 4,1	5,6 4,1	5,95 4,25	5,9 4,3	5,6 4,3	5,2 4,2	4,4 3,6	3,2 2,7	1,8 1,55
№ 2 $K = 2$; $n_k = 5$	4,7 4,15	5,3 4,15	5,7 4,2	6,15 4,2	5,95 4,3	5,7 4,4	5,15 4,1	4,4 3,6	3,2 2,7	1,8 1,5
№ 3 $K = 1$; n_k [1; 20]	4,9 5,65	5,4 5,7	6,0 6,0	6,2 6,15	6,2 6,2	6,2 6,3	5,4 5,7	4,6 5,1	3,4 3,8	1,8 2,0

Таблица 5

Дистанционная зависимость статистических характеристик риска $R_D(N)$ при числе коридоров K и числе дронов в коридоре n_k для случая существенного подавлении ресурсов ЭМП-защиты: Ω_S/Ω_0 [0,5; 0,999]

Вариант	X_1									
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
№ 4 $K = 1$; $n_k = 5$	11,6 16,6	14,0 18,3	15,3 18,5	17,3 18,2	20,3 19,0	23,1 18,7	26,3 19,3	31,95 21,2	35,9 21,8	41,1 24,3
№ 5 $K = 2$; $n_k = 2$	7,4 11,6	8,2 11,45	9,5 11,7	10,4 12,1	12,0 12,8	14,0 13,3	15,9 13,3	18,3 14,2	20,9 14,9	24,7 17,2
№ 6 $K = 1$; n_k [1; 10]	14,8 24,1	15,1 23,5	16,7 24,3	19,1 26,0	22,5 26,1	25,5 28,2	28,8 27,6	33,2 30,2	39,0 33,95	45,3 38,8

Фиксированной переменной является X_1 — нижняя граница диапазона случайных значений $X = r_k/r_M$, также разыгрываемых по методу Монте-Карло, при этом верхняя граница зафиксирована: $X_2 = 1$. Параметр масштаба $A_S = 50$; число разыгрываний $N_P = 5 \cdot 10^3$ и $2 \cdot 10^3$; параметр $\beta = 2$. В числителях "дробей" указаны значения среднего риска $R_D(N)_{\text{ср}}$, в знаменателях — значения СКО.

Сравнительный анализ результатов СИМ, приведенных в табл. 4, 5, позволяет сделать следующие выводы.

1. При несущественном подавлении защитных ресурсов СС-мишени (см. табл. 5) максимальный для СС-инициатора риск $R_D(N)_{\text{ср}}$ имеет место при $r_k/r_M = 0{,}3...0{,}5$, и дальнейшее приближение дронов к объекту не имеет смысла, поскольку эффективность ЭМП-защиты на малых расстояниях от объекта (ввиду выполнения условия $\Pi_S \geqslant \Pi_0$) резко возрастает. При существенном подавлении защитных ресурсов (табл. 5) этого не происходит, и чем ближе дроны будут к объекту Q, тем больше будет позитивный для СС-инициатора риск, связанный с нанесением объекту невосполнимого ущерба.

2. Риск $R_D(N)_{\text{ср}}$ системы ЭМП-защиты более высокого качества (см. табл. 4 для Ω_S/Ω_0 [0,9; 0,999]) во всех ситуациях, вне зависимости от числа сформированных коридоров и числа дронов, попавших в коридор, заметно меньше риска системы более низкого качества (см. табл. 5 для Ω_S/Ω_0 [0,5; 0,999]). Поэтому для СС-мишени жизненно важно по ходу игры следовать правилу $\Omega_S \to \Omega_0 = \text{const}$.

3. Число коридоров и число дронов (детерминированное, случайное), попавших в коридор, несущественно влияет на риск $R_D(N)_{\text{ср}}$, поэтому СС-инициатору в рассматриваемой ситуации выгодно как можно больше дронов с максимальной скоростью направить по первому же сформированному коридору. Отметим также, что общая статистическая ситуация является относительно стабильной, о чем говорит динамика СКО практически во всех рассмотренных вариантах.

Заключение

В представленной статье решение задачи о защите стационарного объекта от воздействия облака дронов доведено до конкретных результатов, полученных с применением СИМ. Помимо иллюстрации развития методов теории игр и системного анализа в рамках общей теории СС это имеет достаточно важное практическое значение, поскольку техника и технологии использования беспилотных летательных аппаратов в настоящее время развиваются весьма интенсивно. Ожидается, что удешевление производства и эффективность массированного применения дронов реализуют исходные условия таких задач настолько быстро, что их решения приобретут возрастающие актуальность и значимость. Поэтому можно надеяться, что предлагаемый подход и приведенные результаты будут востребованы в ближайшем будущем специалистами-практиками в области защиты объектов различного назначения.

Список литературы

1. **Основы** теории управления в системах специального назначения. М.: Изд. УДП РФ, 2008. 400 с.
2. **Моисеев Н. Н.** Элементы теории оптимальных систем. М.: Наука, 1975. 528 с.
3. **Бусленко Н. П.** Моделирование сложных систем. М.: Наука, 1968. 400 с.
4. **Советов Б. Я., Яковлев С. А.** Моделирование систем. М.: Высшая школа, 2001. 275 с.
5. **Сирота А. А.** Компьютерное моделирование и оценка эффективности сложных систем. М.: Техносфера, 2006. 280 с.
6. **Нейман Дж. фон., Моргенштерн О.** Теория игр и экономическое поведение / Пер. с англ. М.: Наука, 1970. 708 с.
7. **Воробьев Н. Н.** Основы теории игр. Бескоалиционные игры. М.: Физматлит, 1984. 496 с.
8. **Колесник Г. В.** Теория игр. М.: Книжный дом "Либроком", 2014. 152 с.
9. **Шумейкер П.** Модель ожидаемой полезности: разновидности, подходы, результаты и пределы возможностей // THESIS. 1994. Вып. 5. С. 29—80.
10. **Павловский Ю. Н.** Имитационные модели и системы. М.: Фазис, ВЦ РАН, 2000. 134 с.
11. **Фомин Г. П.** Математические методы и модели в коммерческой деятельности. М.: Финансы и статистика, 2005. 616 с.
12. **Невежин В. П.** Теория игр. Примеры и задачи. М.: ФОРУМ; ИНФРА-М, 2014. 128 с.
13. **Имитационное** моделирование и управление бизнес-процессами в социальных и экономических системах / Под ред. Э. М. Димова. Самара: Изд-во ПГУТИ, 2020. 172 с.
14. **Димов Э. М., Маслов О. Н., Пчеляков С. Н., Скворцов А. Б.** Новые информационные технологии: подготовка кадров и обучение персонала. Ч. 2. Имитационное моделирование и управление бизнес-процессами в инфокоммуникациях. Самара: СНЦ РАН, 2008. 350 с.
15. **Маслов О. Н.** Случайные антенны: теория и практика. Самара: Изд-во ПГУТИ-Офорт, 2013. 480 с. // URL: http://eisn.psuti.ru/ipublishing/ (дата обращения 01.02.2021).
16. **Маслов О. Н.** Теория случайных антенн: первые 10 лет развития и применения // Антенны. 2017. № 9 (241). С. 37—59.
17. Маслов О. Н. Применение метода статистического имитационного моделирования для исследования случайных антенн и проектирования систем активной защиты информации // Успехи современной радиоэлектроники. 2011. № 6. С. 42—55.
18. **Маслов О. Н.** Принципы моделирования систем защиты информации от утечки через случайные антенны // Специальная техника. 2016. № 6. С. 45—55.
19. **Димов Э. М., Маслов О. Н., Раков А. С.** Управление информационной безопасностью корпорации с применением критериев риска и ожидаемой полезности // Информационные технологии. 2016. Т. 22, № 8. С. 620—627.
20. **Беспроводные** сенсорные сети / Под ред. Б. Я. Лихтциндера. М.: Горячая линия — Телеком, 2020. 236 с.
21. **Бондарев А. Н., Киричек Р. В.** Обзор беспилотных летательных аппаратов общего пользования и регулирование воздушного движения БПЛА в разных странах // Информационные технологии и телекоммуникации. 2016. Т. 4. № 4. С. 19—20.
22. **Кучерявый А. Е., Владыко А. Г., Парамонов А. И.** и др. Летающие сенсорные сети // Электросвязь. 2014. № 9. С. 2—5.
23. **Кучерявый А. Е., Прокопьев А. В., Кучерявый Е. А.** Самоорганизующиеся сети. СПб.: Любавич, 2011. 312 с.

24 **Corson S., Macker J.** IETF REC 2501. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, 1999. 12 p.

25. **Koucheryavy A., Vladyko A., Kirichek R.** State of the art and research challenges for public fluting ubiquities sensor networks // Lecture Notes in Computer Science (LNCS). 2015. Vol. 9247. P. 299—308.
26. **Маслов О. Н.** Инфокоммуникационные технологии в мягких конфликтах XXI века. М.: Горячая линия — Телеком, 2020. 100 с.
27. **Козирацкий Ю. Л., Шляхин В. М., Губарев В. А.** Особенности моделирования сложного коалиционного конфликта в условиях противодействия // Радиотехника. 1997. № 6. С. 18—29.
28. **Сухоруков Ю. С., Шляхин В. М.** Особенности моделирования сложного коалиционного конфликта в условиях противодействия // Радиотехника. 1992. № 1—2. С. 4—11.
29. **Владимиров В. И., Шляхин В. М., Лихачев В. П.** Антагонистический конфликт радиоэлектронных систем. Методы и математические модели. М.: Радиотехника, 2004. 384 с.
30. **Радзиевский В. Г., Сирота А. А.** Информационное обеспечение радиоэлектронных систем в условиях конфликта. М.: Радиотехника, 2001. 456 с.
31. **Прудников А. П., Брычков Ю. А., Маричев О. И.** Интегралы и ряды. М.: Наука, 1981. 800 с.
32. **Маслов О. Н.** Электромагнитная безопасность автоматизированных рабочих мест, оснащенных средствами активной защиты информации // Радиотехника и электроника. 2018. Т.63, № 2. С. 182—192.
33. **Маслов О. Н.** NBIC-конвергенция и угрозы информационной безопасности // Защита информации. Инсайд. 2020. № 5. С. 46—51.
34. **Маслов О. Н., Фролова М. А.** Анализ проекта системы технической защиты информации с применением функционала ожидаемой полезности // Защита информации. Инсайд. 2017. № 2. С. 68—72.
35. **Маслов О. Н., Фролова М. А.** Повышение эффективности функционирования системы радиоконтроля на основе риск-ориентированного подхода // Электросвязь. 2017. № 2. С. 36—42.
36. **Маслов О. Н., Фролова М. А.** Интернет вещей: электромагнитная безопасность пикосотовых технологий // Биомедицинская радиоэлектроника. 2017. № 11. С. 18—29.

O. N. Maslov, Dr. Tech. Sc., Professor, e-mail: maslovpsuti@yandex.ru,
Povolzhskiy State University of Telecommunications and Informatics, Samara, 443010, Russian Federation

Protection of a Stationary Object from the Massive Impact of Mobile Objects under Conditions of von Neumann's Mixed Game

The method of statistical simulation modeling (SSM) has been used to analyze the operating conditions and the efficiency of the physical protection system of a stationary object from the massive impact of unmanned aerial vehicles (drones). It is shown that the conditions of the problem correspond to the reflexive version of a two-sided von Neumann's mixed game. Statistical risk-oriented characteristics for two variants of the object protection system implementation using force mechanical and electromagnetic effects on the "drones cloud" are determined. The possibilities and the prospects for using the results obtained using the SSM method are presented.

Keywords: *physical security, unmanned aerial vehicle, model "drones cloud", mechanical and electromagnetic protection, statistical simulation modeling, protection system characteristics*

DOI: 10.17587/it.27.521-530

References

1. **Foundations** of control theory in special-purpose systems, Moscow, Publishinghouse of UDP RF, 2008, 400 p. (in Russian).
2. **Moiseev N. N.** Elements of the theory of optimal systems, Moscow, Nauka, 1975, 528 p. (in Russian).
3. **Buslenko N. P.** Modeling of complex systems, Moscow, Nauka, 1968, 400 p. (in Russian).
4. **Councils B. Ya., Yakovlev S. A.** System modeling, Moscow, Vysshaya shkola, 2001, 275 p. (in Russian).
5. **Sirota A. A.** Computer modeling and evaluation of the effectiveness of complex systems, Moscow, Tekhnosphera, 2006, 280 p. (in Russian).
6. **Neumann J. von., Morgenstern O.** Game theory and economic behavior, Moscow: Nauka, 1970, 708 p. (in Russian).
7. **Vorobiev N. N.** Fundamentals of the theory of games. Coalition-free games, Moscow, Fizmatlit, 1984, 496 p. (in Russian).
8. **Kolesnik G. V.** Theory of games, Moscow, Book House "Librokom", 2014, 152 p. (in Russian).
9. **Shoemaker P.** Model of expected utility: varieties, approaches, results and limits of possibilities, *THESIS,* 1994, iss. 5, pp. 29—80 (in Russian).
10. **Pavlovsky Yu. N.** Simulation models and systems, Moscow, Fazis, Computing Center of the Russian Academy of Sciences, 2000, 134 p. (in Russian).
11. **Fomin G. P.** Mathematical methods and models in commercial activities, Moscow, Finance and Statistics, 2005, 616 p. (in Russian).
12. **Nevezhin V. P.** Theory of games. Examples and tasks, Moscow, FORUM; INFRA-M, 2014, 128 p. (in Russian).
13. **Dimov E. M.** ed. Simulation modeling and management of business processes in social and economic systems, Samara, PSUTI Publishing House, 2020, 172 p. (in Russian).
14. **Dimov E. M., Maslov O. N., Pchelyakov S. N., Skvortsov A. B.** New information technologies: personnel training and personnel training. Part 2. Simulation and management of business processes in info-communications, Samara, SNTs RAN, 2008, 350 p. (in Russian).
15. **Maslov O. N.** Random antennas: theory and practice, Samara, Publishing house PGUTI-Etching, 2013, 480 p., available at: http://eisn.psuti.ru/ipublishing/ (date of access 02/01/2021) (in Russian).
16. **Maslov O. N.** The theory of random antennas: the first 10 years of development and application, Antennas, 2017, no. 9 (241), pp. 37—59 (in Russian).
17. **Maslov O. N.** Application of the method of statistical simulation modeling for the study of random antennas and the design of systems of active protection of information, *Advances in modern radioelectronics*, 2011, no. 6, pp. 42—55 (in Russian).
18. **Maslov O. N.** Principles of modeling information protection systems against leakage through random antennas, *Special equipment*, 2016, no. 6, pp. 45—55 (in Russian).
19. **Dimov E. M., Maslov O. N., Rakov A. S.** Management of information security of a corporation using risk criteria and expected utility, *Informazionnye Tekhnologii,* 2016, vol.22, no. 8, pp. 620—627 (in Russian).
20. **Lichtzinder B. Ya.** ed. Wireless sensor networks, Moscow, Hot line—Telecom, 2020., 236 p. (in Russian).
21. **Bondarev A. N., Kirichek R. V.** Review of unmanned aerial vehicles for general use and control of UAV air traffic in different countries, *Information technologies and telecommunications*, 2016, vol. 4, no. 4, pp. 19—20 (in Russian).
22. **Kucheryavy A. E., Vladyko A. G., Paramonov A. I.** et al. Flying sensor networks, *Electrosvyaz*, 2014, no. 9, pp. 2—5 (in Russian).
23. **Kucheryavy A. E., Prokopyev A. V., Kucheryavy E. A.** Self-organizing networks, SPb.,Lyubavich, 2011, 312 p. (in Russian).
24. **Corson S., Macker J.** IETF REC 2501. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, 1999, 12 p.
25. **Koucheryavy A., Vladyko A., Kirichek R.** State of the art and research chal-lenges for public fluting ubiquities sensor networks, *Lecture Notes in Computer Science (LNCS),* 2015, vol. 9247, pp. 299—308.
26. **Maslov O. N.** Infocommunication technologies in soft conflicts of the XXI century, Moscow, Hot line—Telecom, 2020, 100 p. (in Russian).
27. **Koziratsky Yu. L., Shlyakhin V. M., Gubarev V. A.** Features of modeling a complex coalition conflict in conditions of opposition, *Radiotechnics.* 1997, no. 6, pp. 18—29 (in Russian).
28. **Sukhorukov Yu. S., Shlyakhin V. M.** Features of modeling a complex coalition conflict in conditions of counteraction, *Radiotekhnika*, 1992, no. 1—2, pp. 4—11 (in Russian).
29. **Vladimirov V. I., Shlyakhin V. M., Likhachev V. P.** Antagonistic conflict of electronic systems. Methods and mathematical models, Moscow, Radiotekhnika, 2004, 384 p. (in Russian).
30. **Radzievsky V. G., Sirota A. A.** Information support of radioelectronic systems in a conflict, Moscow, Radiotekhnika, 2001, 456 p. (in Russian).
31. **Prudnikov AP, Brychkov Yu. A., Marichev O. I.** Integrals and series. Moscow, Nauka, 1981, 800 p. (in Russian).
32. **Maslov O. N.** Electromagnetic safety of automated workstations equipped with means of active protection of information, *Radiotechnology and electronics*, 2018, vol. 63, no. 2, pp. 182—192 (in Russian).
33. **Maslov O. N.** NBIC-convergence and threats to information security, *Information security, Inside*, 2020, no. 5, pp. 46—51 (in Russian).
34. **Maslov O. N., Frolova M. A.** Analysis of the project of a system of technical protection of information using the expected utility functional, *Protection of information, Inside,* 2017, no. 2, pp. 68—72 (in Russian).
35. **Maslov O. N, Frolova M. A.** Improving the efficiency of the radio monitoring system based on the risk-based approach, *Electrosvyaz*, 2017, no. 2, pp. 36—42 (in Russian).
36. **Maslov O. N., Frolova M. A**. Internet of things: electromagnetic safety of picocell technologies, *Biomedical radioelectronica,* 2017, no. 11, pp. 18—29 (in Russian).

УДК: 519.24 DOI: 10.17587/it.27.531-541

Г. Н. Жукова, канд. физ.-мат. наук, доц., e-mail: gzhukova@hse.ru,
Национальный исследовательский университет "Высшая школа экономики",
М. В. Ульянов, д-р техн. наук, проф., вед. науч. сотр., проф., e-mail: muljanov@mail.ru,
Институт проблем управления им. В. А. Трапезникова РАН,
Московский государственный университет им. М. В. Ломоносова

Восстановление символьной периодической последовательности по последовательности с шумом*

Рассматривается задача построения периодической последовательности, состоящей из не менее чем восьми периодов, на основе заданной последовательности, полученной путем внесения шума удаления, замены и вставки символов из неизвестной периодической последовательности, также содержащей не менее восьми периодов. Для построения периодической последовательности, аппроксимирующей заданную, искаженную шумами, вначале требуется оценить длину повторяющегося фрагмента (период). Далее искаженная исходная последовательность разбивается на последовательные участки равной длины, длина пробегает все целые значения от 80 до 120 % от оценки периода. Каждый полученный участок сравнивается с каждым из оставшихся. Для построения периодической последовательности выбирается участок с минимальным редакционным расстоянием. Минимизация проводится по всем участками фиксированной длины, а затем — по всем длинам от 80 до 120 % от оценки периода. Для корректности сравнения фрагментов разной длины редакционное расстояние делится на длину фрагмента. Длина фрагмента, доставляющего минимум отношения редакционного расстояния, считается периодом аппроксимирующей периодической последовательности, а сам фрагмент, повторяясь необходимое число раз, образует аппроксимирующую последовательность. Построенная последовательность может в конце содержать неполный повторяющийся фрагмент. Качество аппроксимации оценивается отношением редакционного расстояния от исходной искаженной последовательности до построенной периодической последовательности такой же длины к этой длине.

Ключевые слова: *символьная последовательность, периодическая последовательность, зашумленная последовательность, шум вставки, шум удаления, шум замены*

Введение

В обширном классе задач, связанных с анализом данных, можно выделить группу задач качественного анализа, целью которого является изучение особенностей качественного поведения исследуемых объектов или процессов. Для решения этих задач рационально использовать предобработку числовых данных о наблюдаемых состояниях исследуемого процесса с использованием символьного кодирования в конечном алфавите. Применение символьного кодирования позволяет отбросить несущественные детали, которые не несут полезной информации в аспекте задач качественного анализа. Более того, такое кодирование актуально при качественном анализе больших данных, поскольку высокая точность числовых представлений признаков приводит к неоправданно большим объемам информации и трудоемким вычислениям без улучшения качества получаемых результатов [1, 2].

В научных исследованиях одним из способов представления информации об исследуемых процессах является представление в виде временных рядов. Наблюдаемые значения исследуемого процесса, являющиеся элементами временного ряда, содержат ошибки измерения, и при этом достаточно часто исходные данные подвержены воздействию случайных искажений, вызванных внешними факторами. Такого рода ошибки в общем случае трактуются как шум. При исследовании временных рядов и их прогнозировании работа с зашумленными дан-

*Работа выполнена при поддержке грантов РФФИ 19-07-00150 и 19-07-00151.

ными вызывает значительные трудности, что приводит к формулировке задачи устранения шума. Для шумоподавления в числовых данных используются различные методы сглаживания, такие как методы скользящего среднего, экспоненциального сглаживания и др. [3], которые однако не применимы при работе с зашумленными символьными последовательностями.

Отметим, что определенная часть исследуемых процессов обладает циклическим характером, символьное кодирование таких процессов приводит к периодическим символьным последовательностям (возможно, содержащим предпериод), а случайные искажения приводят к необходимости применения моделей периодической символьной последовательности с зашумленными циклами [4]. При этом под циклом в символьной последовательности понимают устойчиво повторяющееся слово конечного алфавита, а под периодом — длину этого слова. При внесении шума в цикл мы можем говорить уже не об устойчивой повторяемости, а лишь о наличии некоторого шаблона, описывающего цикл с внесенным в него шумом. При этом при качественном анализе и прогнозировании исследуемых процессов возникают задачи оценки периода и возможных реконструкций или аппроксимаций неискаженного цикла.

Целесообразность применения моделей циклов с шумами обусловлена возможностью решения задач по вероятностному прогнозированию символьных зашумленных последовательностей. Их решение позволит разработать эффективные методы прогнозирования, реконструкции и аппроксимации данных в форме символьных кодов по фрагментарной, неполной и искаженной информации. Эти методы будут полезны в различных предметных областях, связанных с анализом динамических процессов и временных рядов, представленных символьными описаниями.

В качестве примеров применения этих методов укажем такие предметные области, как метеорология и прогнозирование погоды, финансовый анализ и прогнозирование котировок на фондовых биржах, медицинские исследования и анализ медицинских данных, в том числе анализ кардиограмм, исследования в биоинформатике и геномике, например, поиск скрытых периодов в ДНК, а также задачи анализа и прогнозирования экономических временных рядов, в которых возникает необходимость исследования периодичности и аппроксимации циклов [5—25].

Приведем краткий обзор современного состояния исследований в проблематике анализа зашумленных символьных последовательностей. Решение задач оценки периода и собственно аппроксимации цикла осложняются при добавлении в периодическую последовательность шумов различных типов, таких как шумы вставки, замены и удаления символов. Отметим, что шумы вставки и удаления искажают также и периоды, что вносит дополнительные сложности в анализ. Поэтому корректно говорить о задаче оценки периода и о задаче аппроксимации цикла. При шуме замены мы имеем некоторый шаблон с длиной, равной периоду, каждая реализация которого может содержать искажения некоторых символов с некоторыми вероятностями.

Символьное представление и наличие шумов не позволяет использовать классические алгоритмы поиска периода в не зашумленных числовых последовательностях [26]. В связи с этим для решения задачи поиска периодичности в зашумленных символьных последовательностях разрабатываются специальные алгоритмы, некоторые из которых указаны ниже:

- алгоритм WARP [27], основанный на использовании редакционного расстояния между двумя символьными последовательностями, основная идея которого состоит в исследовании модифицированной матрицы сравнений (матрицы трансформаций) последовательности с ней же самой;
- алгоритм CONV [28], ключевой идеей которого является вычисление и анализ свертки исходной последовательности с ее инвертированной копией, с предварительным преобразованием этих последовательностей в двоичный код;
- алгоритм STNR [29] — алгоритм, использующий дерево суффиксов, идея заключается в построении древовидной структуры суффиксов и заполнении этой структуры разностями между позициями элементов их дочерних вершин;
- метод, предложенный коллективом авторов под руководством Е. В. Короткова [30, 31] для выявления скрытой периодичности в последовательностях аминокислот, образующих белки, метод был также апробирован авторами на символьных последовательностях, полученных на основе экономических данных.

Оценка качества этих алгоритмов приводит к необходимости их сравнительного анализа по чувствительности к типу и уровню внесенного шума. Проведенный нами ранее эксперимен-

тальный анализ данных алгоритмов показал, что они чувствительны не только к уровню шума, но и к его типу [32]. В связи с этим интерес представляет разработка таких методов оценки периода и аппроксимации не зашумленного цикла, мощность которых в основном определяется только уровнем внесенного шума. Одно из возможных решений оценки периода предложено нами и нашими соавторами в работе [33]. В качестве основы метода мы использовали данные о частотной встречаемости подслов фиксированной длины в исследуемой символьной последовательности, которые являются основой для вычисления функции энтропии конечных слов [34]. Интерес представляет применение этого подхода и к решению задачи аппроксимации цикла, что и составляет предмет настоящей статьи. Предлагаемый нами метод аппроксимации периодической символьной последовательности по исходной последовательности с шумом основан на исследовании частотной встречаемости и расстояний между совпадающими подсловами фиксированной длины.

Постановка задачи

Мы рассматриваем задачу построения периодической последовательности, содержащей не менее восьми периодически повторяющихся фрагментов (последний может быть неполным), на основе заданной последовательности, полученной внесением шумов вставки, удаления и замены в неизвестную периодическую последовательность. Иными словами, мы рассматриваем задачу аппроксимации строго периодической последовательности по ее зашумленному экземпляру. Задача рассматривается в предположении, что неизвестная строго периодическая последовательность содержала не менее восьми периодов, и уровень шума не превышал некоторого порогового значения. Отметим, что для произвольной зашумленной последовательности решение задачи аппроксимации может быть не единственным.

Терминология: символьную последовательность $q = s_1, s_2, \ldots, s_n$ над некоторым конечным алфавитом Σ, $|\Sigma| \geqslant 2$, где s — произвольный символ из Σ, будем называть словом длины n. Полагаем, что конечный алфавит Σ изначально известен. Последовательность символов $s_k, s_{k+1}, \ldots, s_{m-1}, s_m$, $1 \leqslant k \leqslant m \leqslant n$, назовем подсловом (фрагментом) слова q. Любое подслово может быть получено удалением из исходного слова только некоторого суффикса и/или префикса.

Формальная постановка: пусть в строго периодическое слово q над конечным алфавитом Σ, содержащее не менее восьми периодов, были внесены шумы вставки, удаления или замены символов из того же самого алфавита Σ, в результате чего получено слово q_0 длины n. Тем самым мы принимаем гипотезу о том, что слово q_0 является зашумленной версией строго периодического слова q. *Необходимо* построить, если это возможно, слово q_1 длины n, аппроксимирующее q и представляющее себой подслово строго периодического слова $(p)_{r+2}$, где цикл p имеет длину l_0, а слово q_1 состоит из не менее $r \geqslant 8$ циклов, при этом последний цикл может быть неполным.

Замечания: за счет внесения шумов вставки и удаления символов длина слова q_0 может отличаться от длины порождающего чисто периодического слова q, так что оригинальная длина слова неизвестна. Аппроксимирующее слово q_1 имеет такую же длину, как и q_0, за счет чего может содержать в конце неполный период.

Оценка периода на основе анализа частотной встречаемости подслов

Решение поставленной выше задачи определения подслова p длины l_0, которое является циклом в слове q_1, существенно опирается на полученные нами ранее результаты по оценке периода [33]. В связи с этим мы приводим их краткое изложение.

Оценка периода основана на анализе всех подслов длины k исходного зашумленного слова q_0 и их положения в q_0. Поставим каждому подслову w_i длины k из q_0 в соответствие последовательность $n_i(w_i) = m_1, \ldots, m_t$ номеров первого символа подслова w_i в слове q_0, таких что $m_1 < m_2 < \ldots < m_t$. Таким образом, для фиксированного значения m_j подслово $s_{m_j}, s_{m_j+1}, \ldots, s_{m_j+k-1}$ слова q_0 есть w_i. Оценка периода l_0 проводится далее на основе следующих рассуждений [33]. Обозначим $|n_i(w_i)|$ длину последовательности $n_i(w_i)$. Далее проводится отбор таких w_i, которые встретились более C_{thresh} раз в слове q_0 ($C_{thresh} \geqslant 2$), т. е. строится множество $W = \{w_i : |n_i(w_i)| > C_{thresh}\}$. Значения k и C_{thresh} являются параметрами алгоритма [33]. Для каждого отобранного подслова w_i по его последовательности $n_i(w_i)$ строится мультимножество

$$R_i(w_i) = r_j^c : r_j^c = m_j - m_{j-1}$$

разностей последовательных элементов из $n_i(w_i)$, где c — кратность разности. Анализ полученных разностей и позволяет получить оценку периода для неизвестной строго периодической последовательности q.

Далее все полученные мультимножества $R_i(w_i)$ объединяются в единое мультимножество $R = \cup R_i(w_i)$, при этом у одинаковых разностей кратности суммируются. Каждому элементу r_j^c мультимножества R ставится в соответствие число $h(r_j^c)$, равное сумме кратностей всех тех разностей, которые попадают в сегмент $[(1-v)r_j^c, (1+v)r_j^c]$, при этом значение v также является параметром алгоритма, рекомендуемое значение $v = 0{,}2$:

$$h(r_j^c) = \sum_{c_i : |r_j^c - r_j^{c_i}| \leqslant v r_j^c} c_i. \qquad (1)$$

Заметим, что кратность c самой разности r_j^c тоже входит в эту сумму. В мультимножестве R выбираются разности r_j^c с наибольшим значением $h_{\max}$. Если $h_{\max}$ меньше некоторого порогового значения h_{thresh}, то алгоритм прекращает работу с результатом: "решение не найдено". Пороговое значение C_{thresh} также является параметром алгоритма. Может случиться, что у нескольких разных разностей одно и то же значение $h(r_j^c)$ равно максимальному $h_{\max}$, и $h_{\max} \geqslant h_{thresh}$, тогда выбирается наименьшая разность r_j^c. Полученную разность обозначим r_1, а сегмент $[(1-v)r_1, (1+v)r_1]$ обозначим I_1.

После того как определена первая самая часто встречающаяся разность r_1, строится множество $R^1 = r_j^c \in R : r_j^c \notin I_1$. Заметим, что кратности c у разностей, входящих в R^1, не пересчитываются, они остаются такими, какими были в R. Однако величины $h(r_j^c)$ в множестве R^1 будут другими, поскольку не все слагаемые $c_i : |r_j^c - r_j^{c_i}| \leqslant v r_j^c$ из формулы (1) для R войдут в соответствующие суммы для R^1. В множестве R^1 также находится самая часто встречающаяся разность r_2. Если при этом $h_{\max} \geqslant h_{thresh}$, то аналогично R^1 строится $R^2 = r_j^c \in R^1 : r_j^c \notin I_2$, где $I_2 = [(1-v)r_2, (1+v)r_2]$. В R^2 находится самая часто встречающаяся разность r_3.

Рассмотрим возможные случаи относительно h_{thresh}:

— не найдено ни одной разности с $h_{\max} \geqslant h_{thresh}$, в этом случае алгоритм завершает работу с результатом "решение не найдено";

— найдена только одна подходящая разность r_1, у которой $h_{\max} \geqslant h_{thresh}$, тогда для оценки периода используем минимальное и максимальное значения из сегмента I_1, а именно $\min\limits_{r_j^c \in I_1} r_j^c$ и $\max\limits_{r_j^c \in I_1} r_j^c$;

— найдено не менее двух разностей, в этом случае проводим дополнительный анализ, случаи двух и трех разностей рассматриваются отдельно.

Дополнительный анализ для случая двух и более разностей

В случае, когда найдено ровно две разности r_1 и r_2, таких что

$$h(r_1) \geqslant h_{thresh},\ h(r_2) \geqslant h_{thresh},$$

будем без ограничения общности считать, что $l_1 < l_2$, l_1 и l_2 вычисляются по формуле

$$l_m = \frac{1}{2}\left(\min_{r_j^c \in I_1} r_j^c + \max_{r_j^c \in I_1} r_j^c\right),\ m = 1, 2.$$

Будем говорить, что два натуральных числа $a < b$ отличаются в m раз с точностью ω, если $(1-\omega)ma \leqslant b \leqslant (1+\omega)ma$. Далее, если l_1 отличается от r_2 в 2, 3 или 4 раза с точностью ω, то используем l_1 и l_2 для оценки периода. Если $\Delta l = l_2 - l_1$ примерно в два, три или четыре раза отличается от l_1, то l_1 и Δl используем для оценки периода. Если Δl примерно в два, три или четыре раза меньше l_2, то Δl и l_2 используем для оценки периода. Если и это не выполняется, то для оценки периода используем l_1 и l_2.

Для случая трех разностей l_1, l_2 и l_3 проводится процедура попарного сравнения разностей, как в случае ровно двух разностей. Если первая и вторая (по частоте встречаемости) разности отличаются в 2, 3 или 4 раза с точностью ω, то используем их. Если нет, но l_1 и l_3 отличаются в 2, 3 или 4 раза с точностью ω, то используем l_1 и l_3. Если в 2, 3 или 4 раза с точностью ω отличаются l_2 и l_3, используем l_2 и l_3. Если $\Delta l = l_2 - l_1$ в 2, 3 или 4 раза с точностью ω отличается от l_1 (или l_2), то берем Δl и l_1 (или l_2). Далее рассматривается аналогично $\Delta l = l_3 - l_1$ и $\Delta l = l_3 - l_2$. Если не выполнено ничего из вышеперечисленного, то берем $\min\limits_{r_j^c \in I_1} r_j^c$ и $\max\limits_{r_j^c \in I_1} r_j^c$ для первой разности (самой частой), как в случае ровно одной разности.

Определение повторяющегося фрагмента

На основании полученных оценок периода l_1 и l_2 предлагается следующий метод построения периодически повторяющегося фрагмента (подслова). Сначала для каждого целого числа l_k из диапазона от 80 до 120 % l_1 и из диапазона от 80 до 120 % l_2 зашумленная последовательность разбивается на последовательные непересекающиеся фрагменты длины l_k.

Для каждого фиксированного значения l_k находим минимальное редакционное расстояние от каждого фрагмента длины l_k до остальных фрагментов такой же длины, выбираем фрагмент, для которого расстояние минимально. Следует уточнить, что каждый фрагмент w_k сравнивается с каждым из оставшихся w_j по следующей схеме: к w_j приписываются слева и справа m символов зашумленной последовательности, т. е. рассматривается не сам w_j, а фрагмент w_j^* зашумленной последовательности, начинающийся за m символов до w_j и заканчивающийся через m символов после w_j. В w_j^* рассматриваются все фрагменты, состоящие из l_k последовательных символов, находится редакционное расстояние от w_k до каждого из таких фрагментов. Минимальное из этих расстояний и считается расстоянием от w_k до w_j. Такая особенность сравнения фрагментов обусловлена тем, что за счет шума вставки и удаления длина повторяющихся фрагментов искаженной последовательности, полученных из исходного периодического фрагмента, может несколько отличаться от оригинального периода, так что фрагмент, наименее отличающийся от w_k, может быть на несколько символов сдвинут относительно w_j.

После того как для каждого l_k найдено минимальное редакционное расстояние d_k^* между двумя самими похожими фрагментами, выбирается такая длина l^*, что отношение $\frac{d_k^*}{l_k}$ минимально, соответствующее значение $l_k = l^*$ считается оценкой периода, соответствующий фрагмент w_k используется для построения аппроксимирующей периодической последовательности. В случае, если таких фрагментов несколько, выбираем фрагмент произвольно.

Полученный периодически повторяющийся фрагмент циклически сдвигается на столько символов, чтобы редакционное расстояние от него до первых l^* символов зашумленной последовательности было минимальным. После сдвига фрагмент повторяется p раз, где $p : l^*(p-1) < n \leqslant l^*$. Последовательность первых n символов построенной последовательности считается периодической последовательностью, аппроксимирующей исходную зашумленную последовательность.

Для оценки близости найденной периодической последовательности к зашумленной исходной воспользуемся расстоянием Левенштейна. Будем считать точностью приближения ε отношение расстояния Левенштейна между исходной и полученной периодической последовательностями к длине исходной последовательности:

$$\varepsilon = \frac{d(q_0, q_1)}{n}.$$

При проведении вычислительного эксперимента находится не только редакционное расстояние от аппроксимирующей последовательности q_1 до заданной зашумленной последовательности q_0, но и до исходной чистой периодической последовательности q.

Пример аппроксимации периодического слова предложенным методом

Пусть исходная последовательность q состоит из 8 периодов 110010011110 длины 12. Внесем шум удаления, замены и вставки с уровнем 2, 3 и 0 % соответственно, получим искаженную зашумленную последовательность q_0:

11001011110110010011110110010011010110010011110110010011110110010010110110010011110110010111 10.

Проводим анализ со следующими параметрами: ширина окна $k = 10$; параметр *ignore* = 3 означает, что нужно учитывать фрагменты, встретившиеся более 3 раз; параметр *tightness* = = 0,2 означает, что для каждой разности *difference* из массива самых часто встречающихся разностей находится число других разностей, попадающих в сегмент

[(1 − *tightness*)·*difference*;
(1 + *tightness*)·*difference*];

параметр *min_total* = 8 задает минимальное значение числа наиболее часто встречающихся разностей, необходимое для оценки периода.

Результат: позиции, в которых фрагменты длины 10 встретились более трех раз:

первый фрагмент: 6, 18, 42, 54, 78,
второй: 7, 19, 43, 55, 79,
третий: 8, 20, 44, 56, 80,
и т.п. 9, 21, 45, 57, 69,
10, 22, 34, 46, 58, 70,
11, 23, 35, 47, 71,
12, 36, 48, 72,
13, 37, 49, 73,
14, 38, 50, 74,
15, 39, 51, 75,
16, 40, 52, 76,
17, 41, 53, 77

Разности соседних позиций по каждому фрагменту:

12, 24, 12, 24, 12, 24, 12, 24, 12, 24, 12, 24, 12, 24, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 24, 24, 12, 24, 24, 12, 24, 24, 12, 24, 24, 12, 24, 24, 12, 24.

Сегменты значений разностей, таких что в сегмент попадает не менее 6 разностей: [12, 12] — 23 разности; [24, 24] — 20 разностей. Поскольку сегменты соответствуют кратным числам, причем получилось ровно два таких сегмента, то числа 12 и 24 используются как предварительные оценки периода.

Получаем цикл 100100111101, что полностью совпадает с исходным циклом 110010011110 при циклическом сдвиге на один символ. Точность приближения $\varepsilon = \frac{d(q_0, q_1)}{n} = 5,32\,\%$, и при этом $\frac{d(q, q_0)}{n} = 5,32\,\%$, что позволяет сделать вывод, о том, что полученная периодическая последовательность является хорошим приближением зашумленной в смысле редакционного расстояния.

Вычислительный эксперимент

Для оценки эффективности предлагаемого метода был проведен вычислительный эксперимент по следующей схеме.

1. Создаем случайную последовательность длины p, состоящую из символов 0 и 1. Повторяя полученное слово r раз, получаем чистую периодическую последовательность q.

2. Вносим шумы удаления (*del*), замены (*ch*) и вставки (*ins*) (именно в таком порядке), получаем последовательность q_0.

3. Делаем оценку периода по зашумленной последовательности, используем разности положения повторяющихся фрагментов.

4. Полученную оценку используем для определения повторяющегося фрагмента в аппроксимирующей периодической последовательности q_1.

5. Сравниваем зашумленную периодическую последовательность с аппроксимирующей последовательностью равной длины.

▲ *Генерация периодически повторяющегося фрагмента.*

Используется генератор псевдослучайных чисел Numpy.random.randint. из пакета Numpy языка Python. Получаем последовательность из 0 и 1, состоящую из p псевдослучайных чисел (цикл), где p — период, и, повторяя его, получаем последовательность q.

▲ *Внесение шумов удаления, замены и вставки.*

Вначале по заданным уровням шума удаления, замены и вставки вычисляется длина зашумленной последовательности

$$l_{new} = l_0(1 + \omega_{ins} - \omega_{del}).$$

Затем вычисляется число позиций в периодической последовательности, в которые нужно внести шум каждого типа:

$$n_k = l_{new}\omega_k,\ k = del, ch, ins.$$

С помощью numpy.random.shuffle получаем случайную перестановку l_{new} целых чисел, из которых берем первые n_k. Для каждого типа шума отдельно получаем перестановку и выбираем номера позиций для внесения шума. Вначале вносим шум удаления, затем замены и в конце — вставки, чтобы шумы не могли случайно скомпенсировать друг друга.

▲ *Оценка периода по зашумленной последовательности*

Для получения оценки периода используются следующие параметры:

длина фрагментов $k = 10$; ignore = 3 (учитывать только фрагменты, встретившиеся более трех раз); min_total = 8 (если в мультимножестве R меньше 8 элементов, алгоритм завершается с отрицательным результатом); tightness = = 0,2 (для каждой разности r_j^c подсчитывается число разностей, попадающих в интервал ЛЛ∃ЛЛЛr_j^c; + r_j^c]); min_freq = 6 (при оценке периода учитываются разности, для которых в интервал ЛЛ∃ЛЛЛr_j^c; + r_j^c] попадает не менее шести разностей).

Оценка периода не должна превышать 1/6 длины зашумленной последовательности, иначе невозможно получить периодическую последовательность, содержащую восемь периодов, один из которых может быть неполным. Этот параметр можно также выбирать равным 7.

▲ *Определение повторяющегося фрагмента в аппроксимирующей периодической последовательности*

При построении периодически повторяющегося фрагмента рассматривались варианты значений периода в объединении сегментов

$$[(1 - 0,2)l_k; (1 + 0,2)l_k],\ k = 1, 2,$$

где l_1 и l_2 являются оценками периода. Для каждого варианта периода строится фрагмент на основе зашумленной последовательности, из всех таких фрагментов выбирается такой, что построенная по нему периодическая последовательность имеет минимальное редакционное расстояние до зашумленной, длина такого фрагмента является окончательной оценкой периода.

▲ *Сравнение аппроксимирующей последовательности с зашумленной и исходной периодической*

Сравнение делается по редакционному расстоянию, при сравнении последовательностей разной длины из чисто периодической последовательности q берется подслово, равное длине зашумленной последовательности. При необходимости периодическая последовательность удлиняется на один периодический фрагмент или несколько таких фрагментов, если сами фрагменты короткие, а разность шума вставки и удаления относительно велика.

Результаты и обсуждение

Ниже приведены некоторые из полученных экспериментальных результатов. В табл. 1, 2 и 3 показано, как предложенный алгоритм реагирует на особенности порождающего цикла. Исходная чисто периодическая последовательность была сгенерирована на основе трех циклов определенной структуры с длиной в 16 символов. Цикл повторялся восемь раз. Общий уровень шума был фиксирован на уровне 10 %, при этом шумы удаления (*del*) и вставки (*ins*) изменялись на 1 %, уровень шума замены определялся дополнением до 10 %.

В связи с наличием шума предлагаемым методом не всегда удается определить период. Долю экспериментов, в которых удалось определить период и, следовательно, аппроксимировать цикл, будем далее называть долей распознанных случаев.

Таблица 1

Цикл "0000000001111111111"

del, %	*Ins*, %										
	0	1	2	3	4	5	6	7	8	9	10
0	84	84	94	91	99	100	98	99	97	96	100
1	92	93	98	97	100	98	100	99	100	96	
2	97	98	97	100	97	100	99	97	98		
3	96	99	99	100	100	100	98	97			
4	100	100	99	99	100	100	98				
5	99	100	99	100	100	100					
6	100	100	100	100	100						
7	100	99	100	100							
8	100	100	100								
9	100	100									
10	100										

Таблица 2

Цикл "01001000100001000000"

del, %	*Ins*, %										
	0	1	2	3	4	5	6	7	8	9	10
0	62	56	76	75	78	91	87	95	99	93	99
1	70	69	86	80	83	89	92	96	100	98	
2	68	71	78	91	85	96	98	95	97		
3	75	87	90	95	95	98	97	99			
4	93	88	94	93	99	96	97				
5	82	85	92	89	97	100					
6	92	88	98	92	98						
7	95	95	99	95							
8	94	95	93								
9	91	95									
10	92										

Таблица 3

Цикл "01001100011100001111"

del, %	*Ins*, %										
	0	1	2	3	4	5	6	7	8	9	10
0	54	50	67	74	65	88	82	92	92	91	93
1	59	53	85	71	82	82	91	96	90	95	
2	63	73	75	88	88	92	95	92	96		
3	73	76	84	92	92	95	91	95			
4	81	76	95	85	98	90	97				
5	76	72	90	94	94	97					
6	85	88	97	91	98						
7	98	91	96	97							
8	93	96	99								
9	98	99									
10	100										

В табл. 1, 2 и 3 представлена доля распознанных случаев для трех различных циклов, порождающих исходную чисто периодическую последовательность.

Верхние левые позиции таблиц соответствуют б*о*льшим значениям уровня шума замены ($ch = 10 - del - ins$), и мы видим, что качество аппроксимации в этом случае падает, при этом поведение относительно шумов удаления и вставки практически симметрично, и при малых шумах замены исходный цикл аппроксимируется точно более чем в 90 % экспериментов. Отметим, что качество аппроксимации зависит также и от структуры порождающего цикла.

На рисунке (см. вторую сторону обложки) приведены результаты экспериментов при возрастании общего уровня шума. В легенде рисунка: P — доля распознанных случаев в процентах; T — доля случаев, когда был точно найден период (длина повторяющегося фрагмента); 10 % — доля случаев, когда найденный период не более чем на 10 % отличается от исходного. Для каждой величины (P, T и 10 %) приведены максимальные, минимальные и медианные значения при каждом фиксированном значении общего уровня шума от 5 до 15 % с шагом 1 %. Для каждого значения суммарного шума компоненты шума (вставка, удаление, замена) варьировались с шагом в 1 %.

Анализ медианного значения доли распознанных случаев (P) показывает, что качество аппроксимации выше 90 % мы получаем при суммарном уровне шума, не превосходящем 10 %. Заметим, что при этом медианное значение P достаточно близко к медиане доли случаев, когда найденный период не более чем на 10 % отличается от исходного. Падение максимума распознанных случаев ниже 80 % наблюдается при суммарном уровне шума, превышающем 14 %. Эксперименты с уровнем шума в 20 % показали, что максимум P не превышает в среднем 65 %.

Приведем также результаты экспериментов при уровне суммарного шума в 9 %. В ситуации равных долей шумов всех трех типов (по 3 %) доля распознанных случаев составила около 94 %. Покомпонентное варьирование шума от 0 до 9 % при общей сумме в 9 % показало следующие результаты по доле распознанных случаев: по шуму вставки — от 90 % (при 0) до 97 % (при 9 %), по шуму замены — от 96 % (при 0) до 79 % (при 9 %) и по шуму удаления — от 92 % (при 0) до 97 % (при 4 %) с падением до 94 % при 9 % шума.

Данные о доле распознанных случаев по отдельным (не смешанным) шумам при варьировании от 5 до 9 % следующие: для шума вставки — от 100 до 97 % (при шуме в 9 %), для шума замены — изменение от 100 до 79 %, и по шуму удаления — от 98 до 94 %.

Совокупно полученные нами экспериментальные результаты позволяют сделать вывод о том, что предложенный метод удовлетворительно аппроксимирует последовательности, искаженные шумами вставки и удаления символов бинарного алфавита. Внесение шума замены сильнее сказывается на доле распознанных случаев, которая уже при уровне шума этого типа в 9 % падает до 79 %. Это объясняется использованием бинарного алфавита, для которого замена символа приводит к большей неопределенности в задаче поиска периодичности, чем для алфавитов с большим числом символов.

Заключение

В статье предложен метод решения задачи построения периодической последовательности на основе исходной последовательности, полученной путем внесения шумов вставки, удаления и замены в неизвестную периодическую последовательность. Метод основан на исследовании частотной встречаемости и расстояний между совпадающими подсловами фиксированной длины.

Вычислительный эксперимент показал, что доля зашумленных последовательностей, для которых удалось построить периодическую последовательность, зависит от не только от общего уровня шума, но и от соотношения уровня шумов. При этом наибольшее влияние на качество распознавания оказывает уровень шума замены. Как правило, лучше распознаются зашумленные последовательности с примерно одинаковым уровнем шума каждого типа и последовательности с малым уровнем шума замены в суммарном уровне шума. Удовлетворительные результаты по качеству распознавания (более 80 % доли распознанных случаев по медиане) наблюдаются при уровне суммарного шума, не превышающем 11 %. Полученные результаты согласуются с оценками известных методов решения данной задачи [32], но предложенный метод прост в реализации и получаемые результаты в целом зависят от общего уровня шума, особенно при смеси шумов удаления и вставки, однако шум замены оказывает большее влияние на качество аппроксимации.

Авторы видят продолжение данной работы в исследовании характеристик предложенного метода восстановления символьной периодической последовательности по последовательности с шумом для алфавитов с мощностью большей, чем бинарный. Наше предположение состоит в том, что для таких алфавитов влияние шума замены на качество распознавания будет более слабым.

Список литературы

1. **Zhukova G., Smetanin Y., Uljanov M.** Informative Symbolic Representations as a Way to Qualitatively Analyses Time Series // PROCEEDINGS 2019 International Conference on Engineering Technologies and Computer Science: Innovation & Application. 2019. P. 43—47.

2. **Lin J.** et al. A symbolic representation of time series, with implications for streaming algorithms // Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2003. P. 2—11.

3. **Скляр А. Я.** Анализ и устранение шумовой компоненты во временных рядах с переменным шагом // Кибернетика и программирование. 2019. № 1. С. 51—59. DOI: 10.25136/2306-4196.2019.1.27031

4. **Жукова Г. Н., Сметанин Ю. Г., Ульянов М. В.** Вероятностная модель шумов для периодических символьных последовательностей // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 2. С. 431—440. DOI: 10.25559/SITITO.15.201902.431-440.

5. **Hou W.** et al. A new method to analyse protein sequence similarity using Dynamic Time Warping // Genomics. 2017. Т. 109. № 2. P. 123—130.

6. **Parthasarathy S., Mehta S., Srinivasan S.** Robust periodicity detection algorithms // Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 2006. P. 874—875.

7. **Salvador S., Chan P.** Toward accurate dynamic time warping in linear time and space //Intelligent Data Analysis. 2007. Vol. 11, N. 5. P. 561—580.

8. **Otunba R., Lin J.** APT: Approximate Period Detection in Time Series // SEKE. 2014. P. 490—494.

9. **Pommerening K.** Finding the Period of a Periodic Sequence, 2009. P. 1—6. URL: https://www.staff.uni-mainz.de/pommeren/MathMisc/Periods.pdf (accessed 28.07.2019).

10. **Vlachos M., Yu P., Castelli V.** On periodicity detection and structural periodic similarity // Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics. 2005. P. 449—460.

11. **Otunba R., Lin J., Senin P.** Mbpd: Motif-based period detection // Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer International Publishing, 2014. P. 793—804.

12. **Rao K. P., Gayathri M.** Noise Resilient Periodicity Mining in Time Series Data Bases // International Journal of Computer Science and Network Security (IJCSNS). 2014. Vol. 14, N. 7. P. 41.

13. **Chitharanjan K. et al.** Periodicity detection algorithms in time series databases-a survey // International Journal of Computer Science & Engineering Technology. 2013. Vol. 1, N. 4. P. 22—28.

14. **Sujatha B., Pandian S. C.** Noise Removal in Distributed Time Series Database Using Predominant Pattern Distribution Model // IOSR Journal of Engineering. 2013. Vol. 3, N. 2. P. 06—13.

15. **Karthik G. M., Pujeri R. V.** Constraint based periodic pattern mining in multiple longest common sub sequences // Indian Journal of Science and Technology. 2013. Vol. 6, N. 8. P. 5046—5057.

16. **Sujatha B., Pandian S. C.** Multiplex Tree Pruning for Periodic Pattern Mining // International Journal of Soft Computing. 2014. Vol. 9, N. 1. P. 37—43.

17. **Grossi R.** et al. Suffix trees and their applications in string algorithms // Proceedings of the 1st South American Workshop on String Processing. 1993. P. 57—76.

18. **Chanda A. K.** et al. A new framework for mining weighted periodic patterns in time series databases // Expert Systems with Applications. 2017. Vol. 79. P. 207—224.

19. **Chanda A. K.** et al. An efficient approach to mine flexible periodic patterns in time series databases // Engineering Applications of Artificial Intelligence. 2015. Vol. 44. P. 46—63.

20. **Yuan Q.** et al. PRED: Periodic region detection for mobility modelling of social media users // Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017. P. 263—272.

21. **Yuan Q.** et al. Detecting Multiple Periods and Periodic Patterns in Event Time Sequences // Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017. P. 617—626.

22. **Han J., Dong G., Yin Y.** Efficient mining of partial periodic patterns in time series database // Data Engineering, 1999. Proceedings 15-th International Conference on IEEE, 1999. P. 106—115.

23. **Berndt D. J., Clifford J.** Using dynamic time warping to find patterns in time series // KDD workshop. 1994. Vol. 10, N. 16. P. 359—370.

24. **Grassly N. C., Fraser C.** Seasonal infectious disease epidemiology // Proceedings of the Royal Society of London B: Biological Sciences. 2006. Vol. 273, N. 1600. P. 2541—2550.

25. **Xylogiannopoulos K. F., Karampelas P., Alhajj R.** Analysing very large time series using suffix arrays //Applied Intelligence. 2014. Vol. 41, N. 3. P. 941—955.

26. **Нестеренко А. Ю.** Алгоритмы поиска длин циклов в последовательностях и их приложения // Фундаментальная и прикладная математика. 2010. Т. 16, № . 6. С. 109—122.

27. **Elfeky M. G., Aref W. G., Elmagarmid A. K.** Periodicity detection in time series databases // IEEE Transactions on Knowledge and Data Engineering. 2005. Vol. 17, N. 7. P. 875—887.

28. **Elfeky M. G., Aref W. G., Elmagarmid A. K.** WARP: time warping for periodicity detection // Data Mining, Fifth IEEE International Conference on. IEEE. 2005. P. 8.

29. **Rasheed F., Alhajj R.** STNR: A suffix tree based noise resilient algorithm for periodicity detection in time series databases // Applied Intelligence. 2010. Vol. 32, N. 3. P. 267—278.

30. **Korotkov E. V., Korotkova M. A.** Developing new mathematical method for search of the time series periodicity with deletions and insertions // Journal of Physics: Conference Series. IOP Publishing, 2017. Vol. 788, N. 1. P. 12—19.

31. **Frenkel F. E., Korotkova M. A., Korotkov E. V.** Database of Periodic DNA Regions in Major Genomes // BioMed Research International. 2017. Vol. 2017.

32. **Жукова Г. Н., Сметанин Ю. Г., Ульянов М. В.** Сравнение основных алгоритмов поиска циклов в символьных последовательностях при наличии искажений // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 1107—1116. DOI: 10.25559/SITITO.2019.4.1107-1116.

33. **Жукова Г. Н., Жуков А. В., Сметанин Ю. Г., Ульянов М. В.** Метод определения периода зашумленной периодической символьной последовательности, основанный на позициях подслов в последовательности // Современные информационные технологии и ИТ-образование. 2020. Т. 16, № 1. С. 23—32. DOI 10.25559/SITITO.16.202001.23-32.

34. **Сметанин Ю. Г., Ульянов М. В., Пестова А. С.** Энтропийный подход к построению меры символьного разнообразия слов и его применение к кластеризации геномов растений // Математическая биология и биоинформатика. 2016. Т. 11, № 1. С. 114—126. DOI: 10.17537/2016.11.114.

35. **Левенштейн В. И.** Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады АН СССР. 1965. Т. 163. С. 707—710.

G. N. Zhukova, Ph.D., Associate Professor, e-mail: gzhukova@hse.ru,
National Research University Higher School of Economics, Moscow, 101000, Moscow, Russian Federation,
M. V. Ulyanov, Dr. of Tech. Sc., Professor, e-mail: muljanov@mail.ru,
V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences,
Moscow, 117997, Russian Federation,
Lomonosov Moscow State University, Moscow, 119991, Russian Federation

Reconstruction of a Symbolic Periodic Sequence from a Sequence with Noise

The problem of constructing a periodic sequence consisting of at least eight periods is considered, based on a given sequence obtained from an unknown periodic sequence, also containing at least eight periods, by introducing noise of deletion, replacement, and insertion of symbols. To construct a periodic sequence that approximates a given one, distorted by noise, it is first required to estimate the length of the repeating fragment (period). Further, the distorted original sequence is divided into successive sections of equal length; the length takes on integer values from 80 to 120 % of the period estimate. Each obtained section is compared with each of the remaining sections, a section is selected to build a periodic sequence that has the minimum edit distance (Levenshtein distance) to any of the remaining sections, minimization is carried out over all sections of a fixed length, and then along all lengths from 80 to 120 % of period estimates. For correct comparison of fragments of different lengths, we consider the ration between the edit distance and the length of the fragment. The length of a fragment that minimizes the ratio of the edit distance to another fragment of the same length to the fragment length is considered the period of the approximating periodic sequence, and the fragment itself, repeating the required number of times, forms an approximating sequence. The constructed sequence may contain an incomplete repeating fragment at the end. The quality of the approximation is estimated by the ratio of the edit distance from the original distorted sequence to the constructed periodic sequence of the same length and this length.

Keywords: *symbolic sequence, periodic sequence, sequence with noise, noise of insertion, noise of deletion, noise of change*

Acknowledgements: This work was supported by the Russian Foundation for Basic Research, projects no. 19-07-00150 and 19-07-00151.

DOI: 10.17587/it.27.531-541

References

1. **Zhukova G., Smetanin Y., Uljanov M.** Informative Symbolic Representations as a Way to Qualitatively Analyses Time Series, *PROCEEDINGS 2019 International Conference on Engineering Technologies and Computer Science: Innovation & Application*, 2019, pp. 43—47.

2. **Lin J.** et al. A symbolic representation of time series, with implications for streaming algorithms, *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, ACM, 2003, pp. 2—11.

3. **Sklyar A. Ya.** Analysis and elimination of noise component in time series with variable step, *Kibernetika i Programmirovanie*, 2019, no. 1, pp. 51— 59 (in Russian).

4. **Zhukova G. N., Smetanin Yu. G., Ulyanov M. V.** Probabilistic Noise Model for Periodic Symbol Sequences, *Sovremennye Informacionnye Tekhnologii i IT-Obrazovanie,* 2019, vol. 15, no. 2, pp. 431—440 (in Russian).

5. **Hou W.** et al. A new method to analyse protein sequence similarity using Dynamic Time Warping, *Genomics*, 2017, vol. 109, no. 2, pp. 123—130.

6. **Parthasarathy S., Mehta S., Srinivasan S.** Robust periodicity detection algorithms, *Proceedings of the 15th ACM international conference on Information and knowledge management*, ACM, 2006, pp. 874—875.

7. **Salvador S., Chan P.** Toward accurate dynamic time warping in linear time and space, Intelligent Data Analysis, 2007, vol. 11, no. 5, pp. 561—580.

8. **Otunba R., Lin J.** APT: Approximate Period Detection in Time Series, *SEKE*, 2014, pp. 490—494.

9. **Pommerening K.** Finding the Period of a Periodic Sequence, 2009, pp. 1—6, available at: https://www.staff.uni-mainz.de/pommeren/MathMisc/Periods.pdf (accessed 28.07.2019).

10. **Vlachos M., Yu P., Castelli V.** On periodicity detection and structural periodic similarity, *Proceedings of the 2005 SIAM International Conference on Data Mining,* Society for Industrial and Applied Mathematics, 2005, pp. 449—460.

11. **Otunba R., Lin J., Senin P.** Mbpd: Motif-based period detection, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer International Publishing, 2014, pp. 793—804.

12. **Rao K. P., Gayathri M.** Noise Resilient Periodicity Mining in Time Series Data Bases, *International Journal of Computer Science and Network Security (IJCSNS),* 2014, vol. 14, no. 7, pp. 41.

13. **Chitharanjan K.** et al. Periodicity detection algorithms in time series databases-a survey, *International Journal of Computer Science & Engineering Technology,* 2013, vol. 1, no. 4, pp. 22—28.

14. **Sujatha B., Pandian S. C.** Noise Removal in Distributed Time Series Database Using Predominant Pattern Distribution Model, *IOSR Journal of Engineering*, 2013, vol. 3, no. 2, pp. 06—13.

15. **Karthik G. M., Pujeri R. V.** Constraint based periodic pattern mining in multiple longest common sub sequences, *Indian Journal of Science and Technology*, 2013, vol. 6, no. 8, pp. 5046—5057.

16. **Sujatha B., Pandian S. C.** Multiplex Tree Pruning for Periodic Pattern Mining, *International Journal of Soft Computing*, 2014, vol. 9, no. 1, pp. 37—43.

17. **Grossi R.** et al. Suffix trees and their applications in string algorithms, *Proceedings of the Ist South American Workshop on String Processing*, 1993, pp. 57—76.

18. **Chanda A. K.** et al. A new framework for mining weighted periodic patterns in time series databases, *Expert Systems with Applications*, 2017, vol. 79, pp. 207—224.

19. **Chanda A. K.** et al. An efficient approach to mine flexible periodic patterns in time series databases, *Engineering Applications of Artificial Intelligence*, 2015, vol. 44, pp. 46—63.

20. **Yuan Q.** et al. PRED: Periodic region detection for mobility modelling of social media users, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ACM, 2017, pp. 263—272.

21. **Yuan Q.** et al. Detecting Multiple Periods and Periodic Patterns in Event Time Sequences, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ACM, 2017, pp. 617—626.

22. **Han J., Dong G., Yin Y.** Efficient mining of partial periodic patterns in time series database, *Data Engineering, 1999. Proceedings 15-th International Conference on IEEE*, 1999, pp. 106—115.

23. **Berndt D. J., Clifford J.** Using dynamic time warping to find patterns in time series, *KDD workshop*, 1994, vol. 10, no. 16, pp. 359—370.

24. **Grassly N. C., Fraser C.** Seasonal infectious disease epidemiology, *Proceedings of the Royal Society of London B: Biological Sciences*, 2006, vol. 273, no. 1600, pp. 2541—2550.

25. **Xylogiannopoulos K. F., Karampelas P., Alhajj R.** Analysing very large time series using suffix arrays, *Applied Intelligence*, 2014, vol. 41, no. 3, pp. 941—955.

26. **Nesterenko A. Yu.** Algorithms for finding the lengths of cycles in sequences and their applications, *Fundamentalnaya i Prikladnaya Matematika*, 2010, vol. 16, no. 6, pp. 109—122 (in Russian).

27. **Elfeky M. G., Aref W. G., Elmagarmid A. K.** Periodicity detection in time series databases, *IEEE Transactions on Knowledge and Data Engineering*, 2005, vol. 17, no. 7, pp. 875—887.

28. **Elfeky M. G., Aref W. G., Elmagarmid A. K.** WARP: time warping for periodicity detection, *Data Mining, Fifth IEEE International Conference on. — IEEE,* 2005, pp. 8.

29. **Rasheed F., Alhajj R.** STNR: A suffix tree based noise resilient algorithm for periodicity detection in time series databases, *Applied Intelligence*, 2010, vol. 32, no. 3, pp. 267—278.

30. **Korotkov E. V., Korotkova M. A.** Developing new mathematical method for search of the time series periodicity with deletions and insertions, *Journal of Physics: Conference Series*, IOP Publishing, 2017, vol. 788, no. 1, pp. 12—19.

31. **Frenkel F. E., Korotkova M. A., Korotkov E. V.** Database of Periodic DNA Regions in Major Genomes, *BioMed Research International*, 2017, vol. 2017.

32. **Zhukova G. N., Smetanin Yu. G., Ulyanov M. V.** Probabilistic Noise Model for Periodic Symbol Sequences**,** Comparison of the main algorithms for searching for periods in symbolic sequences in the presence of distortions, *Sovremennye Informacionnye Tekhnologii i IT-Obrazovanie,* 2019, vol. 15, no. 4, pp. 1107—1116 (in Russian).

33. **Zhukova G. N., Zhukov A. V., Smetanin Yu. G., Ulyanov M. V.** Method for determining the period of a periodic symbolic sequence with noise based on the positions of subwords in the sequence, *Sovremennye Informacionnye Tekhnologii i IT-Obrazovanie,* 2020, vol. 16, no. 1, pp. 23—32 (in Russian).

34. **Smetanin Yu. G., Ulyanov M. V., Pestova A. S.** Entropic approach to constructing a measure of symbolic diversity of words and its application to clustering plant genomes, *Matematicheskaya Biologiya i Bioinformatika*, 2016, vol. 11, no. 1, pp. 114—126 (in Russian).

35. **Levenshtejn V. I.** Binary codes with corrected dropouts, insertions and character replacements, *Doklady AN SSSR*, 1965, vol. 163, pp. 707—710 (in Russian).

УДК 004.942 DOI: 10.17587/it.27.542-549

Г. Ч. Набибекова, канд. техн. наук, зав. отделом, e-mail: gulnarara58@mail.ru,
Институт информационных технологий НАН Азербайджана, Баку, Азербайджан

Применение технологии OLAP в среде электронной демографии

Для проведения демографических исследований предложен подход к разработке электронной системы поддержки принятия решений в области демографии с использованием технологий хранилища данных и OLAP. Поскольку демография охватывает множество отраслей, каждая из которых имеет множество индикаторов, предлагается использовать множество взаимосвязанных витрин данных в качестве архитектуры хранилища данных, а также теорию нечетких множеств, основанную на технике вычислений со словами. В статье показано практическое применение взаимосвязанных витрин данных.

Ключевые слова: *демография, демографическая политика, демографическое поведение, электронная демография, хранилище данных, OLAP, витрина данных, шина взаимосвязанных витрин данных, теория нечетких множеств, Computing with Words, MapReduce, Hadoop*

Введение

Проведение продуманной демографической политики в целях улучшения демографической ситуации в стране является важной задачей любого государства. Демографическая политика включает решение таких важных проблем, как увеличение продолжительности жизни населения, рост рождаемости, снижение заболеваемости и смертности, регулирование миграции, как внутренней, так и внешней, забота о здоровье населения, оказание государственной помощи семьям с детьми, планирование трудовых ресурсов и т. д. [1]. В решении этих задач важную роль играет проведение демографических исследований, результаты которых составят базу для демографической политики.

Сегодня информационные технологии проникли во все сферы жизни и деятельности человека. В результате этого влияния появилось новое направление демографии — электронная демография (э-демография), которая занимает особое место в ряду электронных институтов, таких как электронное государство, электронная наука, электронное образование, электронная медицина и т. д. Исследования, проведенные в сфере э-демографии, выявили такую ее важную проблему, как необходимость постоянного предоставления для исследований новых источников данных и использование их для более тщательного изучения демографических процессов.

В статье [2] представлена Концептуальная модель системы э-демографии. В данной модели в качестве одного из ключевых блоков указана "оперативная аналитическая обработка", иначе говоря OLAP-технология (англ. Online Analytical Processing, OLAP), которая способна справиться с решением существующих проблем, предоставляя аналитику общую картину течения процесса, что позволяет усовершенствовать аналитическую составляющую процесса.

Технология OLAP является элементом хранилища данных (ХД) и использует его информацию. С помощью OLAP пользователю предоставляется возможность анализировать данные в реальном времени, осуществлять запросы, получать отчеты. OLAP позволяет проводить многомерный анализ данных путем построения кубов и предоставляет возможности для сложных вычислений, анализа тенденций и моделирования сложных данных, планирования, бюджетирования, прогнозирования и т. д.

ХД представляет собой специальным образом разработанную базу данных с большим объемом информации, использующейся для анализа и принятия управленческих решений. Применение предлагаемых технологий обеспечит подготовку качественно нового информационного слоя в целях оказания помощи лицу, принимающему решение в сфере демографии.

В данной статье отмечен мультиотраслевой характер демографии, для каждой отрасли демографии предложен список индикато-

ров. Кроме того, предложена модель ХД в виде множества взаимосвязанных витрин данных (ВД), измерениями которых будут служить предложенные индикаторы. Также показано применение на базе этого ХД технологии OLAP в рамках э-демографической системы поддержки принятия решений (СППР). Кроме того, для эффективного использования больших данных предлагается применение теории нечетких множеств.

Э-демографическая СППР с включенными в нее технологиями ХД и OLAP обладает широкими возможностями для решения поставленных задач: обеспечивает оперативную обработку информации, выдачу готовых отчетов в ответ на запросы, визуализацию отчетов с помощью таблиц и диаграмм и др.

Отметим, что ранее технологии ХД и OLAP были успешно применены автором при разработке терминологической информационной системы [3] в целях совершенствования аналитической деятельности и оказания поддержки лицам, принимающим решение в области терминологии, и системы поддержки принятия решений в сфере внешней политики [4] для исследования интеграционных процессов между странами. При этом были учтены особенности данных сфер.

1. Основные отрасли демографии и их индикаторы

Человек как демографическая единица обладает следующими признаками: пол, возраст, семейное положение, образование, род занятий, национальность, место проживания и т. д. Очевидно, что многие из этих признаков меняются в течение жизни. Изменения в жизни каждого человека приводят к изменениям населения в целом, которые в совокупности составляют движение населения.

Помимо движения населения демография также изучает зависимость возрастной раскладки, национального и этнического составов, географического положения населения, его численности, интенсивности миграций, числа рождений и смертей от различных факторов, таких как социально-экономические, исторические, политические, этнические, экологические и т. д. Другими словами, демография исследует закономерности событий и процессов на основе социальных, экономических, исторических, политических, этнических, экологических и других проблем, которые возникают в структуре, местоположении, миграции и динамике населения. Таким образом, важной характеристикой демографии является тот факт, что она представляет собой междисциплинарную область исследований и определяется как комплексная наука. У демографии установились тесные связи с такими науками, как экономика, политология, этнография, статистика, история, социология и т. д. (рис. 1). Вследствие этого она делится на целый ряд специализированных отраслей, каждая из которых изучает специфические демографические процессы.

Отметим, что для построения э-демографической СППР с использованием ХД и OLAP необходимо определить индикаторы отраслей, которые будут служить измерениями OLAP-кубов. Каждая из отраслей демографии имеет свой набор индикаторов. Рассмотрим эти отрасли с относящимися к ним индикаторами.

Дескриптивная или описательная демография рассматривает общую характеристику численности, территориального распределения населения, уровня и тенденций демографических процессов конкретной страны или региона. Этот термин применяется также для обозначения общих сведений о численности, составе, размещении и движении населения конкретной страны или территории [5]. Индикаторами описательной демографии являются имя, фамилия, отчество личности в комплексе или его ID, пол, год рождения, семейное положение, вид деятельности, место работы, должность и т. д.

Демографическая экология, или экодемография, изучает влияние демографических процессов на перспективы развития общества и окружающей среды. Отметим, что демографические процессы тесно связаны не только с общественными, но и с экологическими процессами. Влияние окружающей среды на процессы воспроизводства невозможно без обращения к данным экологии [6]. В связи с этим к индикаторам экодемографии можно отнести различные показатели качества жизни насе-

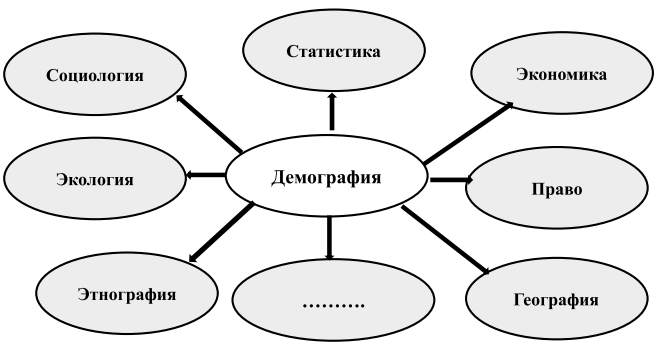


Рис 1. Связь демографии с другими науками

ления, такие как уровни загрязнения воды и воздуха, индекс человеческого развития и т. д.

Экономическая демография определяет взаимосвязь экономических и демографических процессов, изучает особенности влияния возрастно-полового состава населения и составляющих естественного воспроизводства населения на процесс производства, распределения произведенных обществом благ [7]. К индикаторам экономической демографии относятся индикаторы описательной демографии, а также экономические показатели, такие как валовой внутренний продукт (ВВП), валовой национальный доход (ВНД), ВВП на душу населения, ВНД на душу населения, средний размер оплаты труда и др. [8].

Этнодемография изучает этнический состав населения различных территорий, процессы изменения численности народов этих территорий, включая анализ факторов, влияющих на нее, особенности естественного воспроизводства разных этносов [9]. К индикаторам этнодемографии относятся название территории, ее этнический состав и все индикаторы описательной демографии для каждого этноса.

Политическая демография изучает демографические аспекты межнациональных и социальных конфликтов. К индикаторам политической демографии можно отнести военно-мобилизационный потенциал государства, расстановку политических сил вследствие эмиграции, "утечки мозгов", численность различных этносов в высших государственных постах и т. д. [10].

Географическая демография объединяет ряд общественных наук, прежде всего демографию, экономику и социально-экономическую географию. Она изучает региональные особенности демографических процессов, влияние на них как внутренних (демографических), так и внешних (экономических, расселенческих, социальных, этнических, экологических, политических) факторов [11]. Поскольку географическая демография анализирует воздействие среды обитания на демографические процессы, изучает их территориальные различия в динамике, а также анализирует миграционные потоки, к ее индикаторам можно отнести комплекс индикаторов, а именно: индикаторы дискриптивной, экономической, политической демографий, этнодемографии.

Медицинская демография изучает взаимосвязь воспроизводства населения с медико-социальными факторами [12]. К медико-демографическим индикаторам относятся характеристики здоровья населения, процентное соотношение мужчин и женщин в регионе, возрастные показатели рождаемости и смертности, показатели смертности детей в различных возрастных периодах, показатели заболеваемости с указанием названий болезней, показатель естественного прироста или убыли населения, показатель младенческой смертности и т. д.

Историческая демография изучает те же процессы и явления, что и демографическая наука в целом, но в их исторической ретроспективе [13]. К ее индикаторам можно отнести все демографические показатели в их динамике. Выявление на основе полученных данных зависимости демографических характеристик от уровня исторического развития, от особенностей каждого исторического этапа дает возможность обобщить исторические закономерности.

Военная демография изучает мобилизационные возможности государства, людские ресурсы потенциальных военных противников и союзников, потери среди населения и миграционные процессы, вызванные военными действиями, влияние войн на здоровье и воспроизводство населения [14]. Вследствие этого к ее индикаторам можно отнести показатели военной мощи государства, потери среди военного и гражданского населения, напрямую связанные с военными действиями, число заболеваний и смертей, косвенным образом связанных с военными действиями, рождаемость, число мигрантов во время военных действий и т. д.

Социальная демография изучает взаимодействие демографических и социальных процессов [15]. Сюда включено изучение демографических установок личностей, социальных норм, демографическое поведение и влияющие на него факторы, к которым относятся уровень образования, доход, этнические характеристики и т. п. Эти факторы можно отнести к индикаторам социальной демографии.

Как видно из выше изложенного, каждый из этих типов демографии имеет свой набор индикаторов, которые рассматриваются при анализе демографических процессов.

2. Формирование э-демографической системы поддержки принятия решений

Для проведения эффективной демографической политики необходимо проводить исследования в сфере демографии с применением современных информационных технологий, которые вместе с имеющимися в этой сфере большими данными формируют среду э-демографии. В статье [2] помимо представления концептуальной модели э-демографической системы показана необходимость ее разработки, дан пере-

чень важных вопросов, которые можно решить с ее помощью.

Для эффективного решения отмеченных задач необходимо разработать единую национальную э-демографическую систему поддержки принятия решений (СППР) с использованием технологий OLAP и ХД, обладающую средствами ввода, хранения и анализа данных, относящихся к области демографии, в целях принятия правильных решений. Все это имеет огромное значение для обеспечения репродуктивного здоровья населения, улучшения условий жизни населения, укрепления института семьи, решения миграционных процессов, а также для развития кадрового и научного потенциалов в демографической сфере.

Необходимо отметить, что вследствие того, что в сфере демографии наблюдается непрерывный рост объема данных, СППР помимо их ввода должна обеспечить их надежное хранение и эффективное использование. Ввод данных обеспечивается государственными реестрами данных. В некоторых случаях это могут быть OLTP-системы (Online Transaction Processing). Для хранения данных используются СУБД и ХД.

Типичные операции OLAP включают срезы — нарезку и вырезку (slice&dice), свертку (roll-up) и детализацию (drill-down). С помощью этих операций OLAP создаются мультипараметрические модели, цель которых состоит в более адекватном представлении реальных процессов [16].

Поскольку OLAP является элементом ХД, первоочередной задачей является разработка архитектуры ХД. Архитектура ХД зависит от особенностей и свойств выбранной сферы. При построении архитектуры ХД э-демографической СППР следует учесть его следующие характеристики:

- включение в ХД всех отраслей демографии;
- каждая отрасль демографии имеет большое число показателей;
- наличие высокой скорости прироста данных в ХД;
- необходимость высокоскоростной обработки данных;
- многообразие различных типов данных в связи с наличием множества отраслей демографии;
- данные должны быть достоверными;
- должна быть обеспечена визуализация отчетов;
- результаты анализа больших данных должны принести максимум пользы и т. д.

Кроме того, при разработке архитектуры ХД надо учитывать, что некоторые из показателей (индикаторов) принадлежат различным отраслям демографии.

Таким образом, в исследуемом случае можно обнаружить основные определяющие характеристики больших данных (Big Data) — объем (англ. volume), скорость (англ. velocity), многообразие (англ. variety), достоверность (англ. veracity), изменчивость (англ. variability), визуализация (англ. visualization), ценность (англ. value)

Для более рационального и эффективного использования большого объема информации, учитывая при этом его постоянное увеличение, для обеспечения скорости выполнения запросов для данной системы в качестве архитектуры ХД предлагается шина взаимосвязанных ВД [17].

Под ВД понимают специализированные хранилища, обслуживающие одно из направлений деятельности. ВД способствуют смягчению требований к ХД. Каждая ВД включает данные, направленные на решение отдельной задачи. Размеры и сложность структуры ВД, которые также служат для поддержки принятия решений, не имеют ограничений. Тем не менее, ВД, как правило, имеют меньшие размеры и менее сложную структуру, чем ХД, и их легче создавать и поддерживать. ВД являются простой формой ХД. Как и в случае с ХД, с помощью такого аналитического средства, как OLAP, эти данные можно анализировать, выявлять тенденции, прогнозировать будущие результаты и т. д.

Архитектура предлагаемой модели ХД для э-демографической СППР состоит из трех уровней (рис. 2).

Первый уровень — источники данных. Они представляют собой информацию государственных реестров или OLTP-систем.

Второй уровень — подсистема ETL (Extract, Transform, Load). ETL, включенная в ХД в общем случае, а в нашем случае в комплекс ВД, включает следующие процессы:

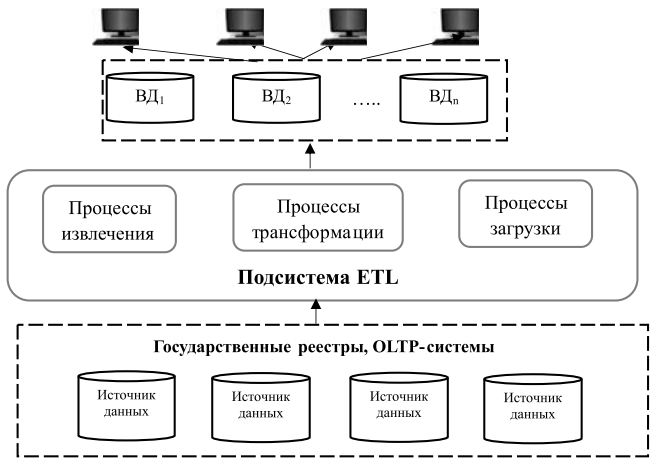


Рис. 2. Архитектура ХД э-демографической СППР

- извлечение данных из внешних источников (Extract);
- преобразование (трансформация) и очистка данных — приведение данных к структурам модели данных и к заданному качеству данных (Transform);
- загрузка данных в область оперативного и постоянного хранения данных ХД (Load) [18].

Необходимость использования ETL объясняется тем, что данные многих источников рассредоточены по разным платформам, использующим самые разнообразные инструменты отчетности и анализа. Данные могут храниться в приложениях, базах данных, электронных таблицах устаревших систем и т. д. Это разнообразие обычно вызвано независимой работой организационных единиц в течение определенного периода времени. Какой бы ни была причина разнообразия, оно обычно приводит к некачественным данным, которые дают неполное и непоследовательное представление о ситуации.

Для преобразования данных низкого качества в информацию высокого качества необходим доступ к широкому спектру источников данных, возможность профилировать, преобразовывать и очищать данные, используя обширную библиотеку преобразований данных для таких типов данных, как текст, числа, дата и др., а также для проверки достоверности выходной информации, что важно для лиц, принимающих решения.

Третий уровень — непосредственно сами ВД, в которые осуществляется процесс загрузки. В текущей задаче ВД ориентированы на проведение анализа по отдельным отраслям демографии.

Одним из преимуществ применения ВД является тот факт, что их использование предполагает распределенную параллельную обработку данных. Такая архитектура означает распределение данных, что обеспечивает гораздо более быструю генерацию результатов. В ее основе лежит модель распределенных вычислений MapReduce и проект Hadoop [19].

Отражение каждой отрасли демографии в электронной демографической системе требует включения в ВД данной отрасли в качестве измерений соответствующих показателей. Например, отрасли "Дескриптивная демография" соответствует ВД с измерениями, относящимися к структуре населения: фамилия, имя, отчество личности или его ID, пол, дата рождения, место проживания, семейное положение и т. д.; в отрасль "Экономическая демография" включены различные экономические показатели для того, чтобы анализировать демографические процессы на их фоне; в отрасль "Военная демография" — показатели военной мощи государства, военные потери населения в половозрастном аспекте, число мигрантов во время военных действий и т. п. В рамках системы на основании каждой ВД строится OLAP-куб. Совокупность OLAP-кубов образует поликубическую модель. Поликубическая модель, включенная в э-демографическую СППР изображена на рис. 3.

Создание ВД начинается с анализа требований для отраслей демографии. Первая ВД строится для отрасли демографии, использующей измерения, а следовательно, и показатели, которые в дальнейшем будут применяться в других отраслях демографии. Такой отраслью демографии является дескриптивная (описательная) демография. Последующие ВД разрабатываются с использованием этих измерений, а также измерений, специфических для конкретной отрасли. Такая архитектура системы в результате приводит к созданию логически интегрированных ВД.

На рис. 4 (см. вторую сторону обложки) в виде графа иллюстрирован обмен информацией между ВД. Коричневым цветом изображены ВД, а синим цветом — измерения, входящие в эти ВД. Некоторые измерения входят в несколько ВД. Это обеспечивает связь между ВД в архитектуре ХД.

Для более эффективного использования большого объема информации также применяют такие операции OLAP, как свертка и детализация, соответствующие агрегации и дезагрегации данных. Эти операции осуществляются над теми измерениями, которые имеют иерархическую структуру [16, 20].

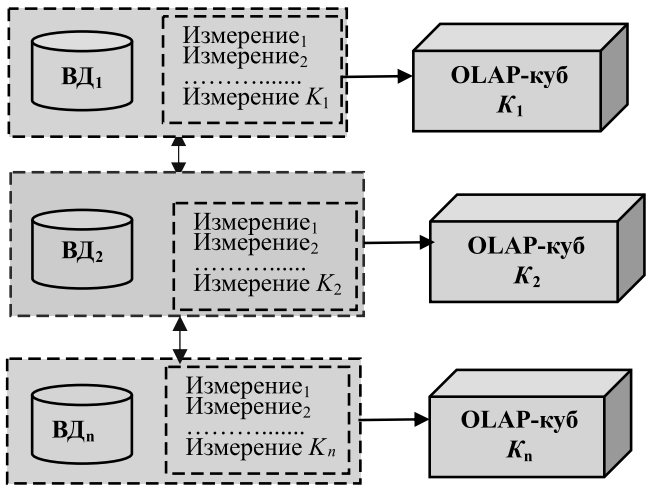


Рис 3. Поликубическая OLAP-модель, входящая в э-демографическую СППР

Еще одним способом работы с большими данными в рамках OLAP является использование теории нечетких множеств, в основе которой лежит техника вычислений со словами (Computing with Words, CW) [21]. В данном случае объектами вычислений являются слова и словосочетания, взятые из естественного языка, например, маленький, большой, дальний, тяжелый, маловероятно и т. д. Для использования этого метода необходимо преобразование числовых измерений ВД в лингвистические переменные. Для определения чисел, с помощью которых будет задана функция принадлежности, на измерениях, которые следует фаззифицировать, применяется кластеризация. Этими числами служат центры кластеров — медоиды [22].

3. Практическое применение

Чтобы реализовать предложенный в статье подход, разработана э-демографическая СППР, в которую включены ХД и OLAP. Архитектура ХД представляет собой шину взаимосвязанных ВД. Как было отмечено выше, на этих ВД строятся OLAP-кубы, измерениями которых являются индикаторы соответствующих типов демографии.

Система реализована для персональных компьютеров, работающих в Windows 7, Windows 8 и т. д. Среда реализации OLAP включает Pivot Table Excel, которая представляет собой визуализацию OLAP. Число персон в разных измерениях определяется агрегатной функцией COUNT().

В качестве примера приведена ВД, в которую включены данные об ученых, которые эмигрировали из Азербайджана за рубеж. Измерениями OLAP-куба, построенного на этой ВД, являются ФИО, год рождения, пол, место рождения, страна миграции, причина миграции, экономическая и политическая ситуация в стране, куда мигрировал сотрудник, место работы, сфера деятельности, должность, число лет пребывания в данной стране, семейное положение и т. д., т. е. в измерения включены данные о сотрудниках, носящие описательный характер, а также данные о стране миграции. Следовательно, эту ВД можно отнести как к дескриптивной, так и к географической демографии, которая демонстрирует миграционные потоки.

На рис. 5 в качестве примера представлен срез куба, полученный в результате запроса: какое число научных работников и в какую страну эмигрировало из Азербайджана за рубеж (на примере пяти стран — США, Германии, России, Турции, Украины), с указанием срока пребывания в данной стране.

Названия строк	Срок проживания (в годах)	4	5	7	8	9	10	12	13	16	17	18	19	20	Общий итог
США								2		1	1				4
Россия			1	1											2
Германия			1	1	1	1	1					1	2	1	9
Турция			1	1	1	1	1					1	2	1	9
Украина			1	1					1						3
Общий итог		1	3	3	3	2	2	2	1	1	1	2	4	2	27

Рис. 5. Число научных работников, эмигрировавших за рубеж с указанием срока пребывания в стране (на примере пяти стран — США, Германии, России, Турции, Украины)

Названия строк	Срок проживания (в годах)	4	5	7	8	9	10	12	13	16	17	18	19	20	Общий итог
США								2		1	1				4
Механика								1							1
Математика								1			1				2
Медицинские науки										1					1
Россия			1	1											2
Аграрные науки			1	1											2
Германия			1	1	1	1	1					1	2	1	9
Физика						1	1					1	1		4
Механика														1	1
Математика					1								1		2
Турция			1	1	1	1	1					1	2	1	9
Биологические науки			1												1
Физика					1	1						1	1		4
Механика														1	1
Математика				1									1		2
Технические науки			1												1
Украина			1	1					1						3
Биологические науки			1	1											2
Математика									1						1
Общий итог		1	3	3	3	2	2	2	2	1	1	1	2	4	1	27

Рис.6. Число научных работников, эмигрировавших за рубеж с указанием специальностей (на примере 5-и стран США, Германия, Россия, Турция Украина)

На рис. 6 представлен срез куба, полученный в результате запроса: какое число научных работников, в какую страну и по какой специальности эмигрировало из Азербайджана за рубеж (на примере пяти стран — США, Германия, Россия, Турция Украина), с указанием срока пребывания в данной стране.

Рассматривая первую страну — США, делаем заключение: в США из отобранных организаций Азербайджана эмигрировало 4 человека. Один — по специальности "Механика", он проживает там 12 лет; двое по специальности "Математика", один из которых проживает там 12 лет, а другой — 17; один по специальности "Медицинские науки", он проживает там 16 лет.

В обоих примерах показано также общее число ученых, эмигрировавших из Азербайджана за рубеж.

В этих примерах выбрано небольшое число измерений. Но возможности данной системы достаточно широки. Зависят они от наполненности ХД. OLAP-кубы позволяют исследовать демографические процессы в различных срезах в зависимости от запроса, а также предоставляют агрегатные данные.

Заключение

Проведение взвешенной, эффективной демографической политики является важной задачей э-государства. Для ее решения необходимо разрабатывать соответствующие системы для оценки, анализа и принятия правильных решений в существующей демографической ситуации, используя различные государственные реестры. На демографические процессы помимо традиционных факторов, к которым относятся миграция, рождение, смерть, также влияют различные социально-экономические ситуации, характеристики здоровья населения, различные показатели качества жизни населения, военные действия, этнический состав, стихийные бедствия и т. д. Наличие в связи с этим различных типов демографии осложняет проведение анализа демографических процессов традиционным способом.

В статье показаны целесообразность и эффективность использования для осуществления демографической политики э-демографической СППР с включенными в нее ХД и OLAP. Предложенная э-демографическая СППР дает возможность в дальнейшем проводить онлайн-мониторинги, анализировать демографические процессы на основании имеющихся данных, выявлять существующие в сфере демографии проблемы. Кроме того, э-демографическая СППР поможет при планировании прогнозировании человеческих ресурсов, при определении стратегических направлений будущего экономического и социального развития, что способствует увеличению численности и повышению уровня квалификации трудоспособного населения. Учитывая актуальность темы, вопросы, связанные с анализом и прогнозированием, будут рассмотрены в дальнейших исследованиях.

Список литературы

1. **Рудницкая А. П., Новиков Е. А.** Основные направления формирования, проблемы и задачи демографической политики в современной России // PolitBook. 2015. № 1. С. 43—56.

2. **Алгулиев Р. М., Алыгулиев Р. М., Юсифов Ф. Ф., Алекперова И. Я.** Формирование электронной демографии как эффективного инструмента социальных исследований и мониторинга данных о населении // Вопросы государственного и муниципального управления. "Высшая школа экономики" (НИУ ВШЭ). 2019. № 4. С. 61—86.

3. **Alguliyev R. M., Nabibayova G. Ch., Gurbanova A. M.** Development of a Decision Support System with the use of OLAP-technologies in the National Terminological Information Environment // International Journal of Modern Education and Computer Science (IJMECS). 2019. Vol. 11, N. 6. P. 43—52.

4. **Набибекова Г. Ч.** Применение OLAP-технологий в системах поддержки принятия решений в сфере внешней политики // Информационные технологии. М.: Новые технологии. 2012. № 2. С. 73—76.

5. **Валентей Д. И.** Демографический энциклопедический словарь. М.: Советская энциклопедия, 1985. 608 с.

6. **Демография.** Фонд Знаний "Ломоносов". URL: http://www.lomonosov-fund.ru/enc/ru/encyclopedia:0147:article?vnum=28696 (дата обращения: 31.01.2020),

7. **Бойко А. И. Карманов М. В.** Экономическая демография: учебно-практическое пособие. М.: МЭСИ, 1999. 68 с.

8. **Мохнаткина Е. В., Голубев А. А.** Показатели, тенденции и факторы экономического развития хозяйствующих субъектов в Российской Федерации // Научный журнал НИУ ИТМО. Серия "Экономика и экологический менеджмент". 2015. № 3. С. 106—116.

9. **Казьмина О. Е., Пучков П. И.** Основы этнодемографии: Учеб. пособ. М.: Наука, 1994. 253 с.

10. **Сакаев В. Т.** Политическая демография: предметное поле и исследовательские возможности. // Научно политический журнал "Власть" Института социологии РАН. 2011. № 7. С. 86—88.

11. **Федоров Г. М.** Об актуальных направлениях геодемографических исследований в России // Балтийский регион. 2014. № 2 (20). С. 7—28.

12. **Шаршакова Т. М., Дорофеев В. М.** Статистика населения и медицинская демография: учебно-методическое пособие. Гомель: ГГМУ. 2009. 57 с.

13. **Мотревич В. П.** Историческая демография России: Учеб. пособ. Екатеринбург: Издательство Уральского федерального университета, 2000. 168 с.

14. **Воронцов А. В.** Демография. 2016. URL: https://studme.org/44089/sotsiologiya/demografiya (дата обращения: 10.01.2020),

15. **Беляевский И. К.** Социально-демографический маркетинг: проблемы, цели, анализ. М: Институт эффективных технологий, 2014. Вып. 3. С. 92—111. URL: http://library.asue.am/open/art27.pdf (дата обращения: 14.01.2020),

16. **Каширин И. Ю., Семченков С. Ю.** Интерактивная аналитическая обработка данных в современных OLAP-системах // Бизнес-информатика. 2009. № 2. С. 12—19.

17. **Ariyachandra T., Watson H. J.** Key Factors in Selecting a Data Warehouse Architecture // Business Intelligence Journal. 2005. Vol. 10, N. 2. P. 19—26.

18. **Реализация** подсистемы ETL (Extract, Transform, Load) корпоративного хранилища данных. URL: https://hadoop.apache.org/ (дата обращения: 27.09.2020).

19. URL: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (дата обращения: 27.09.2020).

20. **Заботнев М. С.** Методы представления информации в разреженных гиперкубах данных. 2006. URL: http://www.olap.ru/basic/theory.asp (дата обращения: 22.09.2020).

21. **Zadeh L.** From Computing with Numbers to Computing with Words — From Manipulation of Measurements to manipulation of Perceptions // IEEE Trans. On Circuits and Systems. 1999. Vol. 45, N. 1. P. 105—119.

22. **Набибекова Г. Ч.** Разработка методов и алгоритмов измерения интеграционных процессов между странами в среде электронного государства: дисс. ... канд. тех. наук. Баку, 2018. С. 69—73.

G. Ch. Nabibayova, PhD in Technical Sciences, Head of Departement, e-mail: gulnarara58@mail.ru,
Institute of Information Technology of Azerbaijan National Academy of Sciences, Baku, Azerbaijan

Application of OLAP Technology in the Environment of Electronic Demography

The article proposes an approach to the development of an electronic demographic decision support system using technologies of Data Warehouse (DW) and Interactive Analytical Processing OLAP. This makes it possible to conduct high-level demographic research and provide support to decision-makers in demographic sphere. The article notes that demography is an interdisciplinary field of research and is defined as a complex science. Each industry of demography has many indicators. A sample list of these indicators is presented. The main characteristics of the DW, which should be taken into account when developing its architecture, are stated. Among these characteristics, one can find the main defining characteristics of Big Data — volume, velocity, variety, veracity, variability, visualization, value etc. For a more rational and efficient use of a large amount of information, taking into account its constant increase, to ensure the speed of execution of requests for a given system, it is proposed to use a Bus of Interconnected Data Marts (DM) as an architecture of DW. One of the advantages of using DM is that their use assumes distributed parallel data processing. This architecture allows for much faster results generation. It is based on the MapReduce distributed computing model and the Hadoop project. In addition, to effectively use large amounts of data, it is also proposed to use OLAP operations such as roll-up and drill-down, as well as fuzzy set theory, based on the technique of computing with words. The article also shows the practical application of interconnected DM. An OLAP cube is built on the basis of these DM. OLAP operations provide the ability to view cubes in different slices and provide aggregate data.

***Keywords**: demography, demographic policy, demographic behavior, e-demography, data warehouse, OLAP, data mart, bus of interconnected data marts, theory of fuzzy sets, Computing with Words, MapReduce, Hadoop*

DOI: 10.17587/it.27.542-549

References

1. **Rudnitskaya A. P, Novikov E. A.** The main directions of formation, problems and tasks of demographic policy in modern Russia, *PolitBook*, 2015, no. 1, pp. 43—56 (in Russian).

2. **Alguliyev R., Aliguliyev R., Yusifov F., Alekperova I.** Developing electronic demography as an effective tool for social research and monitoring population data, *Voprosy gosudarstvennogo i munitsipalnogo upravleniya. "Vysshaya shkola ekonomiki" (NIU VSHE)*, 2020, no. 4, pp. 61—86 (in Russian).

3. **Alguliyev R., Nabibayova G., Gurbanova A.** Development of a Decision Support System with the use of OLAP-Technologies in the National Terminological Information Environment, *International Journal of Modern Education and Computer Science (IJMECS)*, 2019, vol. 11, no. 6, pp. 43—52.

4. **Nabibayova G.** Application of OLAP-technologies in decision support systems in the field of foreign policy, *Informacionnye Technologii,* 2012, no. 2, pp. 73—76 (in Russian).

5. **Valentey D. I.** Demographic Encyclopedic Dictionary, Moscow, Sovetskaya entsiklopediya, 1985 (in Russian).

6. **Demography,** *Fond Znanij "Lomonosov",* 2010, available at: http://www.lomonosov-fund.ru/enc/ru/encyclopedia:0147:article?vnum=28696 (date of access: 31.01.2020). (in Russian).

7. **Boyko A., Karmanov M.** *Economic demography: training manual,* Moscow, Publishing house of MESI, 1999, 68 p. (in Russian).

8. **Mohnatkina E., Golubev A.** Indicators, trends and factors of economic development of economic entities in the Russian Federation, *Nauchnyj zhurnal NIU ITMO. Seriya "Ekonomika i ekologicheskij menedzhment"*, 2015, no. 3, pp. 106—116 (in Russian).

9. **Kasmina O., Puchkov P.** Basics of Ethnodemograph, Moscow, Nauka, 1994, 253 p. (in Russian).

10. **Sakayev V.** Political demography: subject field and research opportunities. *Nauchno-politicheskij zhurnal "Vlast", Institut Sosiologii RAN,* 2011, no. 7, pp. 86—88 (in Russian).

11. **Fedorov G.** On current trends in geodemographic research in Russia, *Baltijskij region,* 2014, no. 2 (20), pp. 7—28 (in Russian).

12. **Sharshakova T., Dorofeev M.** Population statistics and medical demography, *Uchebno-metodicheskoe posobie,* Gomel, GSMU, 2009, 57 p. (in Russian).

13. **Motrevich V.** Historical demography of Russia, *Uchebnoe posobie,* Ekaterinburg, Publishing house of UFU, 2000, 168 p. (in Russian).

14. **Vorontsov A.** Demography. 2016 (in Russian). available at: https://studme.org/44089/sotsiologiya/demografiya (date of access: 10.01.2020),

15. **Belyaevsky I.** Social and demographic marketing: problems, goals, analysis, 2014, available at: http://library.asue.am/open/art27.pdf (date of access: 14.01.2020) (in Russian).

16. **Kashirin I., Semchenkov S.** Interactive analytical data processing in modern OLAP systems, *Busines-informatika*, 2009, no. 2, pp. 12—19 (in Russian).

17. **Ariyachandra T., Watson H.** Key Factors in Selecting a Data Warehouse Architecture, *Business Intelligence Journal*, 2005, vol. 10, no. 2, pp. 19—26.

18. **Implementation** of ETL (Extract, Transform, Load) subsystem of corporate data warehouse, available at: https://hadoop.apache.org/ (date of access: 27.09.2019). (in Russian).

19. **Available at:** https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (date of access: 27.09.2019).

20. **Zabotnev M. S.** Methods for presenting information in sparse data hypercubes, 2006, available at: http://www.olap.ru/basic/theory.asp (date of access: 22.09.2019) (in Russian).

21. **Zadeh L.** From Computing with Numbers to Computing with Words — From Manipulation of Measurements to manipulation of Perceptions, *IEEE Trans. On Circuits and Systems*, 1999, vol. 45, no. 1, pp. 105—119.

22. **Nabibekova G. Ch.** Development of methods and algorithms for measuring of integration processes between countries in the electronic government environment: dis. ... kand.tekh. nauk, Baku, 2018, pp. 66—78 (in Russian).

УДК 004.8 DOI: 10.17587/it.27.550-560

К. В. Данилов[1,2], руководитель направления, danilovkostya@yandex.ru,
С. В. Мальцева[1], д-р техн. наук, проф., smaltseva@hse.ru,
[1] Национальный исследовательский университет "Высшая школа экономики",
[2] ПАО "НЛМК"

Метод автоматической генерации признакового пространства в задаче прогнозирования потребления электроэнергии*

Рассмотрен метод автоматической генерации признакового пространства. Изложен алгоритм работы метода и схема построения модели прогноза. Предлагаемый подход был апробирован на данных о потреблении электроэнергии в регионах России. Результаты проведенных вычислительных экспериментов с применением изложенного метода демонстрируют повышение эффективности разработанной модели и улучшение точности прогнозирования.

Ключевые слова: *автоматическое машинное обучение, автоматическая генерация признаков, прогноз потребления электроэнергии, байесовская оптимизация*

Введение

Интерес к временным рядам как объекту исследования в последние годы резко возрос во многих областях, что связано с широким применением технологий сбора и анализа данных и методов машинного обучения.

Временной ряд представляет собой дискретную упорядоченную последовательность чисел, характеризующую состояние объекта наблюдения в отдельные моменты времени. Временные ряды возникают в результате измерения показателей технических, природных, социально-экономических явлений и процессов, отражая их динамику.

Обзору методов анализа и прогнозирования временных рядов посвящено значительное число работ, рассматривающих как общие подходы [1, 2], так и реализацию этих подходов в различных применениях: энергетике [3, 4], транспорте [5], здравоохранении [6, 7], экономике [8] и других.

Во многих работах рассматриваются особенности анализа и прогнозирования временных рядов в инфраструктуре систем больших данных в сравнении с детерминистскими подходами. В статье [3] выделены преимущества применения методов прогнозирования временных рядов на основе технологий анализа данных: высокая скорость вычислений на данных, поступающих в реальном времени; соответствие нелинейным моделям, часто более высокая точность, чем у детерминистских методов.

В то же время отмечены сложности и недостатки, среди которых указываются необходимость хранения данных предыдущих периодов и проблема обобщения таких методов в контексте других задач.

Также отмечается, что хорошие результаты прогнозирования обеспечиваются при использовании комбинаций методов и учете особенностей прикладных областей, в которых рассматриваются временные ряды.

Задача прогнозирования временного ряда решается на основе модели прогнозирования, адекватно описывающей исследуемый процесс. Модель прогнозирования отображает функциональную зависимость, аппроксимирующую с некоторой погрешностью рассматриваемый временной ряд.

*Работа поддержана грантом Российского фонда фундаментальных исследований 20-07-00651 А "Исследование устойчивости распределенных децентрализованных цифровых систем на основе моделей динамики социальных сетей".

Для построения модели могут быть использованы различные типовые модели и методы в зависимости от особенностей исследуемых данных и требований к использованию результатов анализа. Наиболее известны следующие модели и методы: регрессионные и авторегрессионные модели, нейросетевые модели, модели экспоненциального сглаживания, модели на базе цепей Маркова, классификационно-регрессионные деревья, метод опорных векторов, генетический алгоритм, модель на основе передаточных функций, формализованная нечеткая логика и другие.

Кроме того, существуют задачи, где комбинация (ансамбль) различных моделей предоставляет возможность добиться более точного прогноза целевого временного ряда [9, 10].

В работах [11—13] приведены результаты анализа базовых методов прогнозирования временных рядов. Среди наиболее важных характеристик методов отмечается сложность интерпретации результатов прогнозирования, чувствительность к выбросам и пропущенным значениям, размер доверительного интервала прогноза, объем данных для обучения модели прогнозирования.

Так, высокая интерпретируемость результатов прогнозирования отмечается для моделей, основанных на линейной регрессии, интегрированной модели авторегрессии и скользящего среднего ARIMA, динамических линейных моделей, моделей, основанных на градиентном бустинге на деревьях решений. Низкую интерпретируемость показывают нейросетевые модели.

Модели на основе линейной регрессии и модели экспоненциального сглаживания показывают чувствительность к выбросам и пропущенным значениям.

Для моделей, основанных на экспоненциальном сглаживании, отмечена возможность учитывать сезонность и тренд целевого временного ряда, однако в качестве недостатков отмечается короткий доверительный интервал прогноза.

Для авторегрессионных моделей, таких как ARIMA, в качестве преимуществ отмечаются высокий уровень интерпретируемости результатов прогнозирования и возможность учета сезонного компонента и тренда целевого временного ряда, а в качестве недостатков — необходимость использовать большой объем данных для обучения модели.

Нейросетевые модели позволяют учесть нелинейные зависимости в данных. Эти модели позволяют обрабатывать сложные нелинейные шаблоны, обладают хорошими возможностями для построения прогноза, позволяют автоматизировать процессы их создания и обучения. В то же время для них требуется большой объем данных для обучения модели, для них сложно определить доверительный интервал.

Процесс построения модели прогнозирования временного ряда включает следующие основные этапы:

1. Подготовка временных рядов для повышения качества данных.
2. Проектирование признаков (предикторов).
3. Выбор (проектирование) признаков.
4. Обучение и тестирование модели прогнозирования.
5. Настройка гиперпараметров.
6. Проверка (валидация) модели прогнозирования.

Этапы, связанные с формированием признаков, являются ключевыми, так как пространство признаков (предикторов) является основой, на которой выстраивается модель прогнозирования [12].

В данной работе рассмотрен метод автоматической генерации признакового пространства на основе байесовской оптимизации и особенности применения этого метода при решении задачи прогнозирования объемов потребления электроэнергии на сутки вперед. Методика и алгоритм прогноза основываются на использовании данных прошедших периодов, представленных в форме временных рядов.

Прогнозирование временных рядов в энергетике применяется для решения различных задач. Компании, производящие электроэнергию, заинтересованы в прогнозе, чтобы избежать рисков недостаточности объема производимой энергии при запуске электростанций в ускоренном режиме в условиях рисков возникновения поломок, а также для снижения рисков перепроизводства энергии, что может приводить к проблемам с хранением.

Промышленные предприятия, потребляющие энергию, могут сталкиваться с превышением заявленного потребления, нехваткой мощностей, а также с потреблением, меньшим заявленного. Соответственно, они заинтересованы в получении прогнозных значений, чтобы избежать дополнительных закупок по завышенному тарифу или, наоборот, излишнего расходования средств на неиспользованные ресурсы. Таким образом, решение задачи прогнозирования объемов потребления электроэнергии напрямую влияет на затраты компании, что особенно актуально для орга-

низаций, основные процессы которых связаны с высоким энергопотреблением.

Не менее важно прогнозирование потребления энергии населением, особенно в условиях увеличения используемого оборудования. Например, в работах [14, 15] рассматривается задача автоматизации управления энергопотреблением зарядки электромобилей в определенном районе, чтобы снизить пиковое потребление энергии и наилучшим образом использовать энергию в ночное время.

Еще одной причиной актуальности прогнозирования энергопотребления является предотвращение рисков выхода из строя сетей в случае их перегрузки или поломок. Несмотря на то, что потребление электроэнергии регулируется с использованием различных механизмов управления потреблением, система энергопотребления в значительной мере представляет саморегулирующуюся грид-систему [16, 17], в которой возможно возникновение различных критических явлений, связанных с неравномерностью потребления электроэнергии в различных узлах сети. Предотвращение критических явлений для обеспечения устойчивости системы энергоснабжения требует создания системы мониторинга и сбора информации, а также использования этой информации для прогнозирования энергопотребления [18, 19].

Особенности моделей, применяемых в задачах прогнозирования для приложений в энергетике, определяются, прежде всего, особенностями временных рядов и требованиями к прогнозам.

Одним из важных моментов является тот факт, что на потребление электроэнергии существенно влияют изменения окружающей среды: температурный режим, время суток, погодные условия. В частном секторе существенное влияние могут оказывать важные общественные события, телевизионные трансляции, особенности выходных и праздничных дней.

Современные подходы к прогнозированию, как правило, основываются на использовании нескольких моделей. На выбранном периоде сравнения оценивается их точность, проводится выбор лучшей модели, ее параметры затем перенастраиваются на всей истории. Прогноз далее строится с учетом будущих событий и значений факторов. Именно такой подход использован в работе.

Большинство моделей прогнозирования оперирует набором признаков, которые описывают целевой временной ряд. Используются внешние и внутренние признаки. Важной целью формирования признакового пространства является повышение точности прогноза.

Постановка задачи

Пусть для исходного временного ряда $Y = \{y_1, ..., y_l, ..., y_L\}$, где y_l — l-е значение временного ряда в дискретные моменты времени $t_1, ..., t_L$, $l = 1, 2, ..., L$, $y_l \in R$, L — момент прогноза, определена модель временного ряда:

$$Y_T(X) = F(y_{L+1},\ldots,y_{L+T};X), \qquad (1)$$

где $\{y_{L+1}, ..., y_{L+k}, ..., y_{L+T}\}$ — целевой временной ряд; y_{L+k} — k-е значение целевого временного ряда, $y_{L+k} \in R$, в дискретные моменты времени $t_{L+1},\ldots,t_{L+T}$, $k = 1,2,\ldots,T$, T — глубина прогнозирования, X — вектор параметров модели.

Обозначим:

$Y'_T = \{y'_1,\ldots,y'_k,\ldots,y'_T\}$ — фактический (истинный) целевой временной ряд, y'_k – k -е значение фактического целевого временного ряда в дискретные моменты времени $t_{L+1},\ldots,t_{L+T}$, следующие за L-м моментом времени, $k = 1,2,\ldots,T$, $y_k \in R$;

$\tilde{Y}_T = \{\tilde{y}_1,\ldots,\tilde{y}_k,\ldots,\tilde{y}_T\}$ — целевой временной ряд, который получен с использованием модели прогнозирования, где $\tilde{y}_k$ — k-е значение прогнозного целевого временного ряда в дискретные моменты времени, следующие за L-м моментом времени.

Пусть для каждого x_i, входящего в набор признаков, описываемый вектором X, $i = 1, ..., m$, известны значения в дискретные моменты времени $t_{L+1},\ldots,t_{L+T}$, т. е. для каждого признака x_i существует временной ряд $\{x_{i1},\ldots,x_{iT}\}$.

Пусть имеются N моделей прогнозирования и P вариантов наборов признаков.

Пусть $\tilde{Y}_{jT}(X_p)$ — прогноз j-й модели на $(L + T)$-й момент времени на основе p-го набора признаков, $j = 1, ..., N$; $p = 1, ..., P$.

Тогда $\varepsilon_{jL} = Y'_T - \tilde{Y}_{jT}(X_p)$ — ошибка прогноза с использованием j-й модели прогнозирования на T-й момент времени на основе p-го набора признаков.

На ошибку прогноза влияют как особенности временных рядов, так и состав признаков, а также используемая модель прогнозирования.

Задача формирования признакового пространства состоит в выборе вектора признаков X, при котором ошибка прогноза является минимальной.

Для вычисления оценки ε_{jL} используются различные показатели. Одной из причин этого

является разная чувствительность отдельной оценки к особенностям временных рядов.

Наиболее популярны следующие показатели [20]:

- MAPE (mean absolute percentage error) — средние абсолютные отклонения в процентах. Данная метрика отображает процентное расхождение факта и прогноза, что легко интерпретируется. Недостаток заключается в том, что при малых значениях возникают выбросы в процентном соотношении, и это искажает финальные результаты:

$$MAPE = \frac{1}{T}\sum_{k=1}^{T}\frac{|y'_k - \tilde{y}_k|}{y'_k}; \quad (2)$$

- RMSE (root mean squared error) — среднеквадратичное отклонение:

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T}\left(y'_k - \tilde{y}_k\right)^2}; \quad (3)$$

- MAE (mean absolute error) — среднее абсолютное отклонение:

$$MAE = \frac{1}{T}\sum_{k=1}^{T}|y'_k - \tilde{y}_k|. \quad (4)$$

Метрики $RMSE$ и MAE не масштабируемы и показывают результаты прогнозирования в единицах измерения целевого временного ряда, что позволяет оценить качество модели в терминах поставленной задачи.

Таким образом, для j-й модели прогнозирования требуется найти такой набор признаков X_p, при котором средняя ошибка прогнозируемого значения целевого временного ряда по отношению к истинному стремится к минимальному значению [21]:

$$\frac{1}{T}\sum_{k=1}^{T}\left|y'_k - \tilde{y}_{jk}(X_p)\right| \to \min. \quad (5)$$

Построение модели прогнозирования

Типовая схема процесса построения модели прогнозирования в соответствии с перечисленными выше этапами приведена на рис. 1.

Исходные данные для построения модели прогнозирования представляют собой историческую информацию. Данные обязательно содержат первичный набор признаков X_0 и целевой временной ряд $\tilde{Y}_T$.

На этапе подготовки данных применяются различные подходы к повышению качества данных, проводится обработка выбросов и пропущенных значений во временных рядах. Улучшение качества поступающей информации для обучения модели позволяет нивелировать ее негативное влияние на точность прогноза [12, 21].

Этапы, связанные с формированием признакового пространства X, рассматриваются далее в аспекте их автоматизации, разработки метода и алгоритма автоматической генерации признакового пространства.

Валидация обученной модели прогноза проводилась на отложенной (тестовой) выборке данных [22], что позволило идентифицировать и устранять ошибки переобучения.

В качестве финального прогноза использовалось среднее значение двух прогнозов от моделей на основе градиентного бустинга и нейронных сетей. Именно такая агрегация данных показала минимальное среднеквадратическое отклонение (3).

Метод автоматической генерации признакового пространства

Цель метода автоматической генерации признаков заключается в поиске такого набора признаков, при котором среднеквадратичная ошибка прогнозируемого значения целевого временного ряда по отношению к истинному является минимальной. В результате формируются данные для обучения и тестирования модели прогноза.

В начале работы алгоритма необходимо провести инициализацию возможных трансформаций признаков, целевой функции (4) и параметров оптимизации (время работы алгоритма, число итераций). Метод автоматической генерации признаков включает в себя оптимизацию целевой функции, аналитическое выражение которой не известно и производную которой взять невозможно. Такая задача носит характер оптимизации "черного ящика" [23, 24]. Схема реализации метода представлена на рис. 2

Входными данными для оптимизатора является набор возможных трансформаций **V**

Рис. 1. Построение модели прогноза

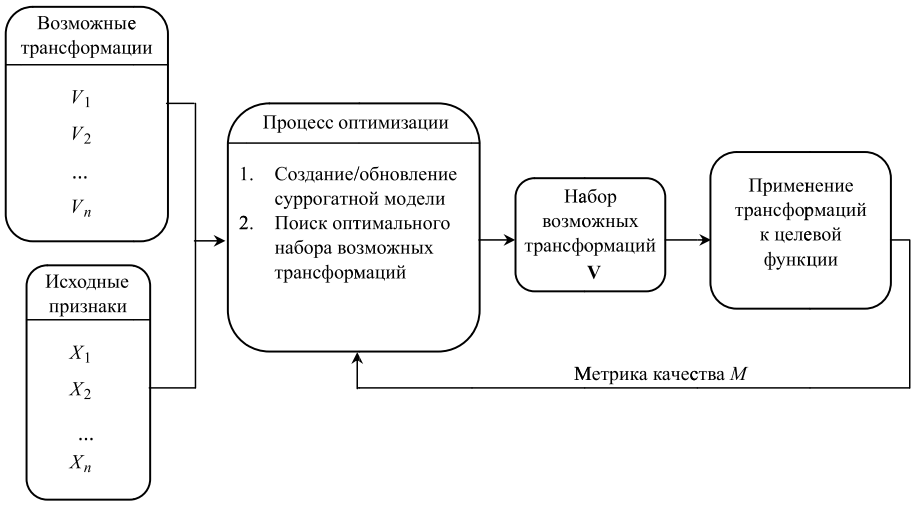


Рис. 2. Схема реализации метода автоматической генерации признаков

для всех первичных признаков X. Выходными данными оптимизатора является такой набор трансформаций **V**, при котором достигается минимум метрики качества M. Метрика качества M (например, (2)—(4)) учитывает отклонения, получающиеся при прогнозировании целевого временного ряда.

Методика оптимизации на основе байесовского подхода [15, 24] состоит из следующих этапов:

1. Построение суррогатной модели целевой функции [25].

2. Поиск оптимального набора возможных трансформаций для первичных признаков.

3. Применение набора возможных трансформаций к целевой функции.

4. Обновление суррогатной модели, включающей новые результаты, полученные на этапе 3.

5. Повторение шагов 2—4 в соответствии с критерием останова, который может определяться заданным числом итераций или временем работы.

Построение суррогатной модели целевой функции. Суррогатная модель, также называемая поверхностью отклика, является вероятностным представлением целевой функции, построенной с использованием предыдущих оценок. В качестве основы для построения суррогатной модели используются гауссовские процессы [25].

Последовательность $\{s_w\}_{w \in W}$ называется гауссовским процессом $Q(s)$, если для любого конечного множества $\{w_1, w_2, \ldots, w_n\}$ случайные величины $\{Q(s_1), Q(s_2), \ldots, Q(s_n)\}$ имеют совместное многомерное нормальное распределение:

$$p(Q|S) = N(Q|\mu, B), \qquad (6)$$

где $Q = \{Q(s_1), Q(s_2), \ldots, Q(s_n)\}$, $\mu = \{m(s_1), m(s_2), \ldots, m(s_n)\}$ и $B_{ij} = B(s_i, s_j)$; m — среднее значение гауссовского процесса; B — положительно определенная функция ядра или ковариационная функция. Гауссовскй процесс представляет собой распределение по функциям, гладкость которых определяется на основе B. Среднее значение и ковариационная функция полностью определяют гауссовский процесс $Q(x)$.

Априорное распределение $p(Q|S)$ может быть конвертировано в апостериорное $p(Q|S, Y)$ после добавления дополнительных данных, которые являются значениями целевой функции. Апостериорное распределение может быть использовано для того, чтобы сделать прогноз Q^*, основанный на признаках S^*:

$$p(Q^*|S^*, S, Y) = \int p(Q^*, Q|S^*, S, Y)\, dQ = N(Q^*|\mu^*, \Sigma^*) \qquad (7)$$

где $p(Q^*|S^*, S, Y)$ — апостериорное распределение, которое получено при рассмотрении новых данных Q^* и также является гауссовским процессом со средним μ^* и Σ^*.

Поиск оптимального набора возможных трансформаций для первичных признаков. После построения суррогатной модели на основе гауссовских процессов требуется найти минимум целевой функции (5) на заданном ограниченном интервале значений (domain) возможных трансформаций первичных признаков [17].

Функция выбора — это критерий, на основе которого выбирается следующий набор трансформаций первичных признаков. Наиболее распространенной функцией выбора является ожидаемое улучшение (expected improvment) [12]

$$EI(s) = \mathbb{E}\max(Q(s) - Q(s^+), 0), \qquad (8)$$

где $Q(s^+)$ — найденное минимальное значение целевой функции; s^+ — соответствующая трансформация первичных признаков, при которых найдено минимальное значение целевой функции. Ожидаемое улучшение может быть выражено через суррогатную модель:

$$EI(s) = \begin{cases} (\mu(s) - Q(s^+) - \xi)\Phi(Z) + \\ + \sigma(s)\varphi(Z) \quad \sigma(s) > \\ 0, \text{ если } \sigma(s) = 0, \end{cases} \qquad (9)$$

где

$$Z = \begin{cases} \dfrac{\mu(s) - Q(s^+) - \xi}{\sigma(s)}, \text{ если } \sigma(s) > 0; \\ 0, \text{ если } \sigma(s) = 0, \end{cases} \qquad (10)$$

$\mu(s)$ и $\sigma(s)$ —это среднее значение и стандартное отклонение аспостериорного распределения в точке s соответственно; Ф и φ — кумулятивная функция распределения (CDF) и функция плотности распределения (PDF) соответственно.

Первое и второе слагаемые в уравнении (9) представляют собой член эксплуатации (exploitation) и разведки (exploration) соответственно. Таким образом, изменяя параметр ξ, можно варьировать важность среднего значения и интервала неопределенности апостериорного распределения при построении функции выбора [19].

Суррогатная модель предназначена для более быстрой вероятностной оценки значений целевой функции. Функция выбора позволяет определить следующие тестовые значения s, которые требуется проверить на целевой функции. После вычисления значений целевой функции на тестовых данных s происходит обновление суррогатной модели, которая включает полученные результаты.

Таким образом, итерационно происходит поиск глобального минимума целевой функции (5). Как правило, критерием останова для описанного алгоритма служит число пройденных итераций.

Далее приведены результаты применения данного алгоритма для поиска трансформаций первичных признаков, которые используются при обучении модели прогноза потребления электроэнергии.

Экспериментальные исследования

Эксперименты проводились на основе данных о потреблении электроэнергии в городах Орел, Томск, Курск, Мурманск, Смоленск и Тверь. Исследуемая историческая информация об энергопотреблении перечисленными регионами РФ была предоставлена компанией ООО "АналитиксХаб" и обладала исторической глубиной 2...3 года. Задача в период апробации описанного в данной работе подхода заключалась в построении ежедневного прогноза потребления электроэнергии на сутки вперед по каждому региону. Для ее реализации были созданы, протестированы и впоследствии переобучались на еженедельной основе модели машинного обучения для прогнозирования временных рядов.

В дополнение к данным об энергопотреблении использовалась информация о погоде в каждом рассматриваемом регионе РФ, полученная из открытых источников (rp5.ru). Анализировались такие факторы, как температура окружающего воздуха, влажность, облачность, атмосферное давление. Также брались в расчет данные о национальных праздниках РФ и важных датах исследуемых регионов. Исходное признаковое пространство для построения моделей прогноза потребления электроэнергии формировалось на ежедневной основе.

Использовался язык программирования Python 3.6 для проведения всех манипуляций с данными.

Подготовка данных. Главная цель этапа подготовки данных заключалась в проведении верификации исторической информации на предмет наличия выбросов и пропущенных значений в данных (рис. 3). Выполнять подобного рода проверку требовалось при каждом поступлении новых данных по всем имеющимся признакам.

Были протестированы следующие подходы к обнаружению выбросов в данных: z-оценка, кластеризация методом DBScan, изоляционный лес (isolation forest) [22].

Лучший результат продемонстрировал метод z-оценки, который довольно просто реализуется на практике и полностью выполняет требуемый поиск выбросов в данных.

Также пропущенные значения были замечены в данных, которые негативно влияют на

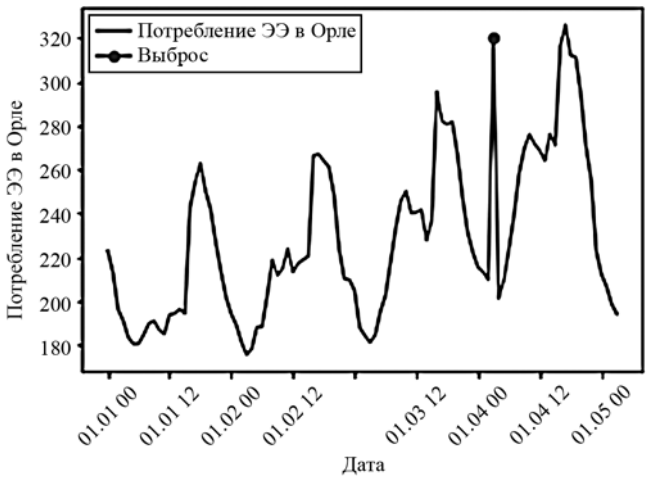


Рис. 3. Выброс в данных

точность прогнозирования при обучении модели машинного обучения. Проводилась их замена на медиану скользящего окна. [23].

Применение метода автоматической генерации признакового пространства. Данные по рассматриваемым регионам для прогнозирования потребления электроэнергии обладают определенной спецификой, которая ставит ограничения на генерацию дополнительных признаков. Например, фактические значения потребления электроэнергии в Мурманске становятся доступными только в конце месяца (по остальным регионам отставание до 3...4 дней), что ограничивает использование лаговых недельных переменных. Влияние подобных ограничений негативно отражалось на точности прогнозирования.

Были рассмотрены (табл. 1) возможные трансформации исходных признаков и соответствующие им области значений, которые применялись при проведении исследований.

Также добавлялись признаки из даты и времени: месяц, порядковый номер недели в году и месяце, порядковый номер дня в году, месяце и неделе, день недели, порядковый номер часа в 24-часовом формате, порядковый номер минуты в часе.

На рис. 4 (см. третью сторону обложки) представлен пример использования метода автоматической генерации признакового пространства при выполнении трансформации только лагового значения температуры окружающего воздуха.

На рис. 4 можно заметить, что минимум целевой функции, в данном случае нормализованной среднеквадратической ошибки, наблюдается при использовании лагового значения 37 температуры окружающего воздуха. Также десять красных точек обозначают итерации, которые были выполнены в процессе оптимизации. Потенциальная следующая итерация была бы сделана в интервале самой большой неопределенности, которая обозначены красной вертикальной чертой.

Число проделанных итераций отражено на рис. 5. В данном случае третья итерация оказалась самой эффективной, и была достигнута минимальная среднеквадратическая ошибка при прогнозировании потребления электроэнергии на основе исходных дополнительных признаков.

Таблица 1

Трансформации признаков

№	Трансформации признаков	Область значений
1	Взятие лаговых значений	[1, 100]
2	Преобразование с помощью тригонометрических функций	[sin, cos, tg, ctg]
3	Обработка на основе скользящего окна (размер окна)	[5, 48]
4	Перемножение бинарных признаков (выбор двух признаков)	[0, 1, 2, 3, 4]
5	Дифференцирование с лаговым значением	[1, 72]

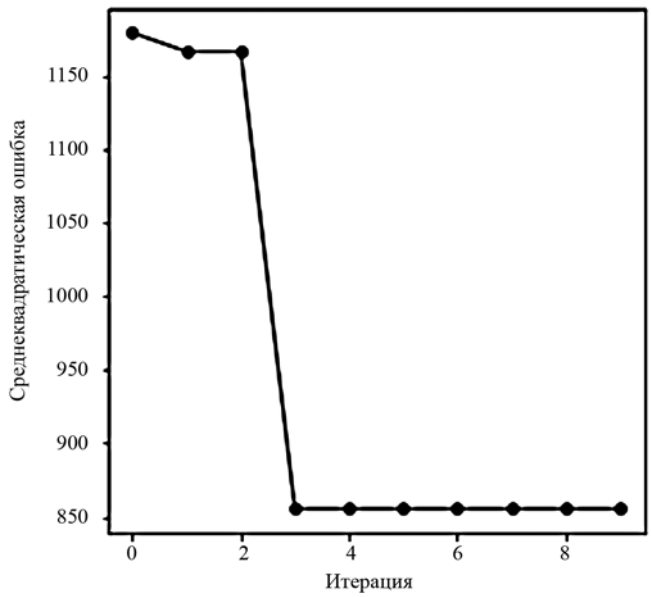


Рис. 5. Число итераций при генерации признака

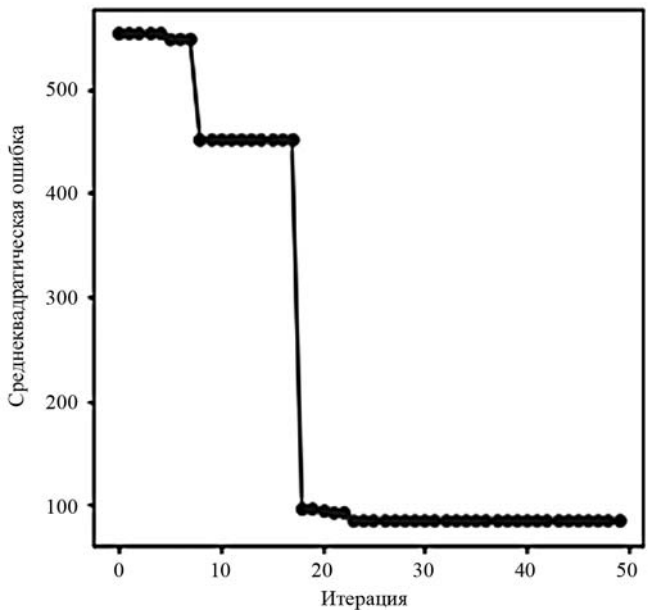


Рис. 6. Число итераций при генерации признакового пространства

На рис. 6 представлен пример использования метода автоматической генерации признакового пространства при выполнении 50 итераций на основе возможных трансформаций из табл. 1.

Среднеквадратическая ошибка сильно уменьшилась при генерации дополнительных признаков, что положительно влияет на точность прогнозирования потребления электроэнергии.

Безусловно, время работы метода автоматической генерации признакового пространства напрямую зависит от числа итераций оптимизатора и рассматриваемых трансформаций, что обязательно нужно учитывать при решении подобных задач.

Обучение и тестирование моделей прогноза потребления электроэнергии. В результате предыдущих этапов формируется подготовленный набор данных для обучения и тестирования модели прогноза потребления электроэнергии. Признаковое пространство включало в себя как исходные данные, так и сгенерированную информацию, что в сумме составляло 156 объектов.

В процессе проведения экспериментов были применены различные технологии прогнозирования временных рядов, но лучшие результаты на имеющихся данных показали градиентый бустинг и нейронные сети типа LSTM [9,10,26]. Модели прогноза строили на основе двух подходов, и усредненное значение использовали для финального ответа. В качестве основных показателей точности исследовали коэффициент детерминации и среднеквадратическую ошибку.

Тестирование происходило с помощью кросс-валидации, где число фолдов равнялось пяти [14, 17]. Также финальная модель проходила настройку гиперпараметров на основе байесовской оптимизации, что позволяло улучшить точность прогнозирования объемов потребления электроэнергии.

Результаты прогнозирования. Тестирование разработанного подхода проводили в течение двух месяцев с ежедневным построением прогноза потребления электроэнергии регионами РФ и еженедельным переобучением модели на основе метода автоматической генерации признакового пространства.

В процесс испытаний также была включена интегрированная модель авторегрессии и скользящего среднего *ARIMA*, которая широко популярна при решении задачи прогнозирования временных рядов, что позволило провести более комплексную верификацию описываемого в данной работе метода.

Таблица 2

Результаты прогнозирования

Регион	Средние абсолютные процентные отклонения прогноза от фактического потребления энергопотребления, %			
	Использовалась ручная генерация дополнительных признаков		Использовался метод автоматической генерации признаков	
	Ансамбль из градиентого бустинга (*Xgboost*) и нейронной сети *LSTM*	*ARIMA*	Ансамбль из градиентого бустинга (*Xgboost*) и нейронной сети *LSTM*	*ARIMA*
Орел	1,8	3,2	1,7	2,5
Томск	2,4	2,4	2,4	2,2
Курск	2,08	2,5	1,7	2,5
Мурманск	3,1	5,2	2,9	5,3
Смоленск	1,99	2,6	1,8	2,2
Тверь	2,12	4,2	2,3	3,9

В табл. 2 представлены средние абсолютные отклонения прогноза от фактического потребления электроэнергии в период тестирования. Можно заметить, что автоматическая генерация признакового пространства в большинстве случаев добавляла прирост в точности прогнозирования.

На рис. 7 (см. третью сторону обложки) отображены результаты прогнозирования потребления электроэнергии на основе ансамбля градиентного бустинга и нейронной сети *LSTM*. Значимое уменьшение средних процентных отклонений можно наблюдать в Курсе — 1,7 % по сравнению с 2,08 %. Для модели этого региона метод автоматической генерации признакового пространства создал дополнительные временные ряды, которые оказали влияние на построение прогноза и помогли улучшить точность.

Результаты прогнозирования с помощью ARIMA-модели представлены на рис. 8 (см. третью сторону обложки). Средние процентные отклонения значительные выше по сравнению с моделью ансамбля деревьев решений и нейронной сети *LSTM*. Тем не менее, автоматическая генерация признаков позволила увеличить точность прогноза.

Наблюдаются регионы, где автоматическая генерация признакового пространства про-

демонстрировала результат хуже, чем ручная. В первую очередь, это связано со спецификой и влияющими факторами самого региона на потребление электроэнергии. Расширяя область значений возможных трансформаций и число итераций оптимизатора, можно ожидать повышение точности прогноза.

Заключение

Разработанный метод автоматической генерации признакового пространства на основе байесовской оптимизации ориентирован на построение моделей краткосрочного прогноза и включает ряд улучшений по сравнению с существующими реализациями: позволяет формировать нетривиальные признаки, повышающие точность прогноза; нивелирует выбросы и пропущенные значения в данных.

Метод учитывает особенности временных рядов, наблюдаемых в сфере энергопотребления, опирается на исторические данные и принимает в расчет сезонный компонент, тренд, цикличность.

Результаты экспериментального исследования разработанного метода в течение двух месяцев на данных о потреблении электроэнергии по регионам России при ежедневном использовании и еженедельном переобучении модели прогноза показали существенное улучшение точности прогнозирования. На исследуемой выборке она составила 96...99 %.

Возможности разработанного метода позволяют провести его дальнейшее совершенствование в направлении уменьшения продолжительности работы оптимизатора за счет обучения и тестирования модели прогноза на сокращенной выборке, а также оценки качества найденных трансформаций признаков с помощью статистических алгоритмов. С незначительными модификациями, учитывающими особенности временных рядов, метод может быть успешно использован в моделях прогнозирования временных рядов во многих других областях, связанных с потреблением и распределением ресурсов.

Список литературы

1. **Esling P., Agon C.** Time-series data mining // ACMComput. Surv. 2012. Vol. 45, N. 1. Article 12. P. 1—34.

2. **riyamvada, Wadhvani R.** Review on various models for time series forecasting // 2017 International Conference on Inventive Computing and Informatics (ICICI). Coimbatore. 2017. P. 405—410. doi: 10.1109/ICICI.2017.

3. **Chirag D., Fan Z., Junjing Y., Siew E. L., Kwok W. S.** A review on time series forecasting techniques for building energy consumption // Renewable and Sustainable Energy Reviews. 2017. Vol. 74, Iss. C. P. 902—924.

4. **Eliana V., Héctor A., Rodrigo S.** A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score // Entropy. 2020. Vol. 22, N. 12. 1412. URL: https://doi.org/10.3390/e22121412.

5. **Wu J., Zhong L., Li L., Lu A.** A Prediction Model Based on Time Series Data in Intelligent Transportation System // Information Computing and Applications. ICICA 2013. Communications in Computer and Information Science. Vol 392. Springer, Berlin, Heidelberg. URL: https://doi.org/10.1007/978-3-642-53703-5_43.

6. **Lin W. T.** Multiple Time Series Approach to the Analysis of Hospital Patient Movements // Third International Conference on System Science in Health Care. Health Systems Research. Springer, Berlin, Heidelberg, 1984. URL: https://doi.org/10.1007/978-3-642-69939-9_225.

7. **Luo L., Zhang, X.** et al. Hospital daily outpatient visits forecasting using a combinatorial model based on ARIMA and SES models // BMC Health Serv Res. 2017. N. 17, 469. URL: https://doi.org/10.1186/s12913-017-2407-9.

8. **Troy D., Aaron S. H.** Recession forecasting using Bayesian classification // International Journal of Forecasting. 2019, Vol. 35, Iss. 3. P. 848—867.

9. **Nameer A. K., Mahdi J., Peter S.** Application of Deep Learning Long Short-Term Memory in Energy Demand Forecasting // Engineering Applications of Neural Networks. 2019. P. 31—42.

10. **Waddah W., Rozaida G.** Multi-step Time Series Forecasting Using Ridge Polynomial Neural Network with Error-Output Feed-backs // Soft Computing in Data Science. Second International Conference. 2016.

11. **Manuel M., Jesus R.** Electricity Load Forecasting Using Self Organizing Maps // International Conference on Artificial Neural Networks. ICANN 2006. Artificial Neural Networks — ICANN. 2006. P. 709—716.

12. **ValiakhetovaYu. I., FilippovaA. S., KaramovaL. M.** Applied operational research problems // Вестник УГАТУ. 2013. Т. 17, № 6 (59). С. 83—87.

13. **Chandu S.** Time Series Forecasting. What is time series forecasting and what are the basic steps?, URL: https://dzone.com/articles/time-series-forecasting.

14. **Jason B.** A Gentle Introduction to XGBoost for Applied Machine Learning. URL: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ (дата обращения: 26.06.2019).

15. **Леонова Ю. В.,Федотов А. М.** К проблеме построения модели поискового образа документа // Междунар. конф. "Современные проблемы математики, информатики и биоинформатики". 2011. С. 2.

16. **Han L. S.** A robust functional time series forecasting method // Journal of Statistical Computation and Simulation. 2019. Vol. 89, N. 5. P. 795—814. DOI: 10.1080/00949655.2019.1572146.

17. **Безручко Б. П., Смирнов Д. А.** Статистическое моделирование по временным рядам. Саратов: Издательство ГосУНЦ "Колледж", 2000. С. 6.

18. **Сидоров С. Г., Никологорская А. В**. Анализ временных рядов как метод построения прогноза потребления электроэнергии // Вестник ИГЭУ. 2010. № 3. URL: https://cyberleninka.ru/article/n/analiz-vremennyh-ryadov-kak-metod-postroeniya-prognoza-potrebleniya-elektroenergii (дата обращения: 05.08.2021).

19. **Щербаков М. В., Бребельс А., Щербакова Н. Л., Тюков А. П.** Обзор оценок качества моделей прогнозирования. URL: http://www.mtas.ru/bitrix/ components/bitrix/forum.interface/show_file.php?fid=6450 (дата обращения: 05.07.2019).

20. **Marco L., Jiri O.** Bayesian Optimization Algorithms for Multi-objective Optimization // Parallel Problem Solving from Nature — PPSN VII. 2002. P. 298—307.

21. **Aggarwal C. C., Yu P. S.** Outlier Detection for High Dimensional Data // ACM SIGMOD Record (SIGMOD REC). February 2010. P. 10.

22. **Крючин О. В., Козадаев А. С., Дудаков В. П.** Прогнозирование временных рядов с помощью искусственных нейронных сетей и регрессионных моделей на примере прогнозирования котировок валютных пар // Электронный научный журнал "ИсследованоРоссии". 2010. С. 354—362.

23. **Ümit C. B., Şeyda E.** Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition // Neurocomputing. 2019, Oct. P. 151—163

24. **Makridakis S., Spiliotis E., Assimakopoulos V.** Statistical and Machine Learning forecasting methods: Concerns and ways forward // PLOS ONE. 2018. Vol. 13, N. 3. URL: https://doi.org/10.1371/journal.pone.0194889.

25. **Jasper S., Hugo L., Ryan P.** Practical Bayesian optimization of machine learning algorithms // In Proceedings of the 25th International Conference on Neural Information Processing Systems — Volume 2 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA. P. 2951—2959.

26. **Matthew N., Zhongyang Z., Caisheng W., Carol J. Mi.** Improving Short-Term Electricity Price Forecasting Using Day-Ahead LMP with ARIMA Models // 2017 IEEE Power & Energy Society General Meeting. 2017.

K. V. Danilov[1, 2], Lead Data Scientist, danilovkostya@yandex.ru,
S. V. Maltseva[1], Dr. of Tech., Professor, smaltseva@hse.ru,
[1] National Research University Higher School of Economics, Moscow,101000, Russian Federation,
[2] PJSC NLMK, Moscow, 119017, Russian Federation

Automated Feature Engineering Method in the Problem of Forecasting Energy Consumption

The automated feature engineering method in the problem of forecasting energy consumption is considered. The algorithm of the method and the scheme of the forecasting model construction are stated. The proposed approach was tested on data about electricity consumption in Russian regions. The results of the computational experiments carried out using the described method demonstrate an increase in the efficiency of the developed forecasting model and improvement of accuracy.

Keywords: *automated machine learning, automated feature engineering, forecasting energy consumption, Bayesian optimization*

Acknowledgements: The work was supported by the grant of the Russian Foundation for Basic Research 20-07-00651 A "Study of the stability of distributed decentralized digital systems based on models of the dynamics of social networks".

DOI: 10.17587/it.27.550-560

References

1. **Esling P., Agon C.** Time-series data mining, *ACMComput. Surv.*, 2012, vol. 45, no. 1, article 12, pp. 1—34.

2. **Priyamvada, Wadhvani R.** Review on various models for time series forecasting, *2017 International Conference on Inventive Computing and Informatics (ICICI),* Coimbatore, 2017, pp. 405—410, doi: 10.1109/ICICI.2017.8365383.

3. **Chirag D., Fan Z., Junjing Y., Siew E. L., Kwok W. S.** A review on time series forecasting techniques for building energy consumption, *Renewable and Sustainable Energy Reviews*, 2017, vol. 74, iss. C, pp. 902—924.

4. **Eliana V., Héctor A., Rodrigo S.** A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score, *Entropy,* 2020, vol. 22, no. 12, 1412; available at: https://doi.org/10.3390/e22121412.

5. **Wu J., Zhong L., Li L., Lu A.** A Prediction Model Based on Time Series Data in Intelligent Transportation System, *Information Computing and Applications. ICICA 2013. Communications in Computer and Information Science*, 2013, vol. 392. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-53703-5_43.

6. **Lin W. T.** Multiple Time Series Approach to the Analysis of Hospital Patient Movements, *Third International Conference on System Science in Health Care. Health Systems Research*, 1984, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-69939-9_225.

7. **Luo L., Zhang X.** et al. Hospital daily outpatient visits forecasting using a combinatorial model based on ARIMA and SES models, *BMC Health Serv Res.,* 2017, vol. 17, 469, https://doi.org/10.1186/s12913-017-2407-9.

8. **Troy D., Aaron S. H.** Recession forecasting using Bayesian classification, *International Journal of Forecasting,* July—September 2019, vol. 35, iss. 3, pp. 848—867.

9. **Nameer A. K., Mahdi J., Peter S.** Application of Deep Learning Long Short-Term Memory in Energy Demand Forecasting, *Engineering Applications of Neural Networks,* 2019, pp. 31—42.

10. **Waddah W., Rozaida G.** Multi-step Time Series Forecasting Using Ridge Polynomial Neural Network with Error-Output Feed-backs, *Soft Computing in Data Science. Second International Conference.* 2016.

11. **Manuel M., Jesus R.** Electricity Load Forecasting Using Self Organizing Maps, *International Conference on Artificial Neural Networks. ICANN 2006: Artificial Neural Networks — ICANN*, 2006, pp. 709—716.

12. **Valiakhetova Yu. I., Filippova A. S., Karamova L. M.** Applied operational research problems, *USATU Bulletin, Science Magazine*, 2013, vol. 17, no. 6 (59), pp. 83—87.

13. **Chandu S.** Time Series Forecasting. What is time series forecasting and what are the basic steps?, available at: https://dzone.com/articles/time-series-forecasting

14. **Jason B.** A Gentle Introduction to XGBoost for Applied Machine Learning. available at: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ (date accessed: 26.06.2019).

15. **Leonova Yu. V., Fedotov A. M.** *Intern. conf. "Modern problems of mathematics, computer science and bioinformatics",* 2011, pp. 2.

16. **Han L. S.** A robust functional time series forecasting method, *Journal of Statisti-cal Computation and Simulation*, 2019, vol. 89, no. 5, pp.795—814, DOI: 10.1080 / 00949655.2019.1572146.

17. **Bezruchko B. P., Smirnov D. A.** Statistical modeling by time series, Saratov,Publishing House of the State Scientific Center "College", 2000, 6 p.

18. **Sidorov S. G., Nikologorskaya A. V.** Analysis of time series as a method for constructing a forecast of electricity consumption, *Bulletin of ISUE*, 2010, no. 3, available at: https://cyberleninka.ru/article/n/analiz-vremennyh-ryadov-kak-metod-postroeniya-prognoza-potrebleniya-elektroenergii (date of access: 05.08.2021).

19. **Shcherbakov M. V., Brebels A., Shcherbakova N. L., Tyukov A. P.** Review of estimates of the quality of forecasting models, available at: http://www.mtas.ru/bitrix/components/bitrix/forum.interface/show_file.php?Fid=6450 (date accessed: 05.07.2019).

20. **Marco L., Jiri O.** Bayesian Optimization Algorithms for Multi-objective Optimization, *Parallel Problem Solving from Nature — PPSN VII*, 2002, pp. 298—307.

21. **Aggarwal C. C., Yu P. S.** Forecasting Time Series Using Artificial Neural Networks and Regression Models Using the Example of Forecasting Currency Pairs Quotes, *Scientific Journal "Investigated in Russia",* 2010, pp. 10.

22. **Kryuchin O. V., Kozadaev A. S., Dudakov V. P.** Forecasting time series using artificial neural networks and regression models on the example of forecasting quotes of currency pairs, *Electronic scientific journal "IssledovanovRussia",* 2010, pp. 354—362.

23. **Ümit C. B., Seyda E.** Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition, *Neurocomputing*, 2019, Oct, pp. 151—163.

24. **Makridakis S., Spiliotis E., Assimakopoulos V.** Statistical and Machine Learning forecasting methods: Concerns and ways forward, *PLOS ONE*, 2018, vol. 13, no. 3, available at: https://doi.org/10.1371/journal.pone.0194889.

25. **Jasper S., Hugo L., Ryan P.** Practical Bayesian optimization of machine learning algorithms, *In Proceedings of the 25th International Conference on Neural Information Processing Systems — Volume 2 (NIPS'12),* Curran Associates Inc., Red Hook, NY, USA, pp. 2951—2959.

26. **Matthew N., Zhongyang Z., Caisheng W., Carol J. Mi.** Improving Short-Term Electricity Price Forecasting Using Day-Ahead LMP with ARIMA Models, *2017 IEEE Power & Energy Society General Meeting,* 2017.

Адрес редакции:

107076, Москва, Матросская тишина, 23с2

Телефон редакции журнала **+7 916 392 2167**
E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.
Корректор *М. Ю. Безменова*.
Сдано в набор 10.08.2021. Подписано в печать 21.09.2021. Формат 60×88 1/8. Бумага офсетная.
Усл. печ. л. 8,86. Заказ IT1021. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: **www.aov.ru**

Рисунок к статье К. В. Данилова, С. В. Мальцевой

«МЕТОД АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ПРИЗНАКОВОГО ПРОСТРАНСТВА В ЗАДАЧЕ ПРОГНОЗИРОВАНИЯ ПОТРЕБЛЕНИЯ ЭЛЕКТРОЭНЕРГИИ»

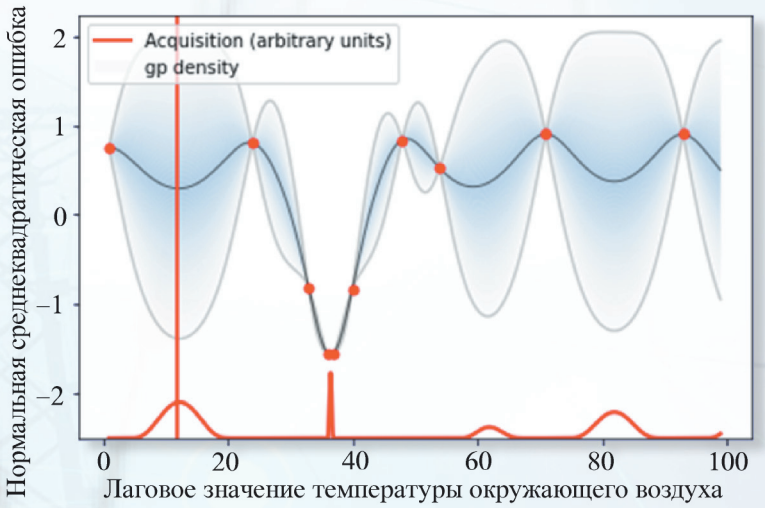


Рис. 4. Автоматическая генерация признака

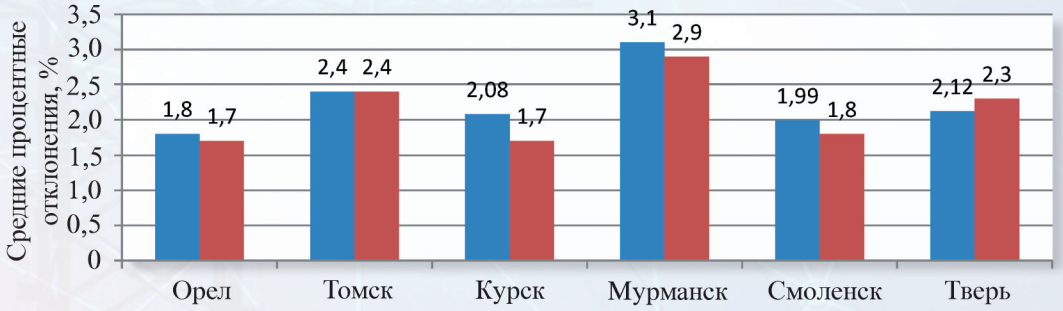


Рис. 7. Средние процентные отклонения в результате прогнозирования на основе ансамбля из градиентного бустинга и нейронной сети:
■ – Ручная генерация признаков; ■ – Автоматическая генерация признаков

Рис. 8. Средние процентные отклонения в результате прогнозирования на основе ARIMA-технологии:
■ – Ручная генерация признаков; ■ – Автоматическая генерация признаков

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

выпускает *научно-технические журналы*

Ежемесячный теоретический
и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656

Подписной индекс по Объединенному каталогу
«Пресса России» – 79963

Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопрос разработки и внедрения нанои микросистем в различные области науки, технологии и производства.

Подписной индекс по Объединенному каталогу
«Пресса России» – 79493

Подписной индекс по Объединенному каталогу
«Пресса России» – 79492

Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автомати ческого и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Теоретический
и прикладной
научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях челове ческой деятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765

Адрес редакции журналов для авторов и подписчиков:
107076, Москва, Матросская тишина, д. 23, стр. 2, оф. 45. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: +7 (499) 270-16-52. E-mail: antonov@novtex.ru

ISSN 1684-6400. Информационные технологии. 2021. Том 27. № 10. 505–560. Индекс: 72656 (Объединенный каталог «Пресса России»)