

Л.Ф. Гуляницкий, А.И. Павленко

## Разработка и исследование генетических алгоритмов для прогнозирования временных рядов

Разработан основанный на генетическом алгоритме метод прогнозирования временных рядов, исследованы подходы к построению процедур обучения. Предложены способы представления популяции в алгоритме, а также процедуры рекомбинации, формирования ниш, расчета приспособленности и разрешения конфликтов. Приведены результаты вычислительного эксперимента с использованием реального временного ряда.

Genetic-based time-series forecasting algorithm is developed and suggested. The approaches to the construction of training procedures are described. The methods of representation of the population in the algorithm, as well as procedures recombination, formation of the niches, the calculation of fitness, conflict resolution are presented. The results of computer simulation using the real time series are shown.

Розроблено метод прогнозування часових рядів на основі генетичного алгоритму, досліджено підходи до побудови процедур навчання. Запропоновано способи подання популяції в алгоритмі, а також процедури рекомбінації, формування ніш, розрахунку пристосованості та розв'язання конфліктів. Подано результати обчислювального експерименту з використанням реального часового ряду.

**Введение.** Развитие теории прогнозирования в последние десятилетия привело к разработке многих методов, процедур и приемов построения прогнозов, различных по своим параметрам. По оценкам специалистов, уже насчитывается более двухсот методов прогнозирования, поэтому при решении конкретных практических задач возникает проблема выбора методов, которые давали бы наиболее точные прогнозы. К основным трудностям в получении точных прогнозов общих показателей экономической деятельности относятся неожиданные и существенные изменения в ключевых экономических факторах. Эмпирические исследования точности прогнозов показали, что ни один метод экстраполяции не дает одинаково удовлетворительных результатов для всех типов данных и разных прогнозных горизонтов; проблема также – в учете возникающих тенденций.

Перспективные подходы к прогнозированию временных рядов возникают в области искусственного интеллекта, где наблюдается тенденция заимствования идеи у природы. В частности, исследования нервной системы и биологическая эволюция повлияли на разработку таких мощных инструментов, как генетические алгоритмы

(ГА). Последние относятся к классу методов поиска, итерационно улучшающих качество вариантов решения процедурами рекомбинации и отбором для выживания. Благодаря универсальности вычислительной схемы, возможностям параллельной реализации и устойчивости к шуму, ГА находят успешное практическое применение при решении многих сложных нелинейных многомерных задач оптимизации. ГА широко применяются в научных и инженерных задачах – комбинаторной оптимизации, распознавании образов, компьютерном видении и др. [1, 2].

С другой стороны, прогнозирование на основе правил может основываться на использовании нескольких временных рядов и выявлять существующие причинно-следственные связи в сложных экономических процессах, ход которых в значительной степени зависит от различных факторов. Еще один положительный аспект применения этого подхода – гибкость задания целевой функции. Традиционно ГА используются в сфере оптимизации, но при условии некоторой модификации могут служить для классификации и прогнозирования [3].

В области эволюционных вычислений одна из наиболее сложных проблем – это поиск зна-

чений оптимальных параметров из-за их существенного влияния на эффективность алгоритма.

Предлагаемый метод прогнозирования базируется на ГА, исходная идея которого предложена в [2], также исследуются вопросы его реализации и практического применения.

### Общая схема генетического алгоритма для решения задач прогнозирования

Предлагается разделить процесс прогнозирования на основе правил на несколько шагов [4]:

1. Этап *обучения* (*learning phase*) – построение базы правил, направленное на выявление существенных зависимостей во временном ряде на основе использования выбранного алгоритма обучения – ГА.

2. На этапе *сопоставления* (*match phase*) правила проверяются на допустимость применения для прогнозирования в рассматриваемый момент времени и формируется подмножество конфликтующих правил (при их наличии).

3. Этап *разрешения конфликтов* предназначен для определения среди конфликтующих правил тех, которые будут применяться для прогнозирования.

4. На этапе *применения* (*action phase*) строится прогноз для заданного периода.

ГА является общим подходом и имеет многочисленные реализации для разных задач, но в целом его вычислительная схема представлена на рис. 1. Есть два основных подхода для представления набора правил при применении ГА [1]. В питтсбургском подходе одна хромосома (решение) кодирует одно правило, поэтому увеличивается сложность обрабатываемых структур. В мичиганском подходе обычно одно правило представлено набором хромосом, следовательно, возникает проблема разрешения конфликтов.

Разработан ряд различных стратегий разрешения конфликтов. Одним из самых простых способов считается выбор такого правила, которое попало в набор конфликтов первым или последним. Другой способ решения конфликта – выбрать правило, наиболее специфичное, т.е. предпочтение отдается правилу с наибольшей условной частью [5]. В стратегии приоритетов выбираются правила, наиболее важные в

некотором смысле (обычно приоритеты определяют экспертным путем). Иногда предпочтение отдается правилам, которые выполняются для более новых данных временного ряда. Считается, что правило с актуальными данными наиболее релевантно. Еще один подход – игнорирование уже использованных правил. Часто для построения прогноза используют комбинации различных стратегий. Разрешение конфликтов – существенный аспект функционирования подобных систем прогнозирования на основе вывода по правилам, поскольку этим определяется чувствительность и стабильность всей системы.

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

Рис. 1. Псевдокод общей схемы ГА

Работа ГА обычно начинается с инициализации популяции – набора вариантов решений задачи (хромосом в терминах эволюционных алгоритмов), оценивается качество и определяется значение фитнеса всех ее элементов. На следующих этапах из популяции выбираются хромосомы, к которым применяются генетические операторы – скрещивание и мутация с целью создания хромосом лучшего качества, чем предшественники. На каждой итерации алгоритма популяция пополняется новыми хромосомами. Цикл продолжается до тех пор, пока не выполнится некоторое условие остановки – например, достижение максимального количества итераций или же не происходит улучшение качества хромосом на протяжении некоторого времени и др. Результатом работы ГА для поставленной задачи будет набор хромосом – множество правил, согласно которому будет строиться прогноз [6].

### Представление правил

В работе отдается предпочтение питтсбургскому подходу представления популяции, т.е. правило имеет вид *if-then* и состоит из произ-

вольного числа условий  $\omega$  (атомарных подправил) и итогового значения  $\omega_c$ . Атомарное подправило  $\omega$  есть утверждением, что прирост в произвольных точках временного ряда принадлежит некоторому интервалу ранжирования [7]:

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

где  $F_t$  – значение временного ряда в момент времени  $t$ ;  $t_1, t_2$  – произвольные моменты времени,  $t_1 < t_2$ ;  $y_b, y_{b+1}$  – действительные числа, которые используются для ранжирования приростов,

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

При этом

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

где  $F_{\max}$  – максимальное абсолютное значение прогноза (положительное или отрицательное), определяемое экспертным путем для конкретной модели.

В результате процедуры ранжирования строится  $2m$  интервалов (для отрицательных и положительных приростов). Следует отметить, что алгоритм прогнозирования на основе ГА предполагает возможность оперирования несколькими временными рядами для учета причинно-следственных факторов в сложных системах. В таком случае параметры  $F_{\max}$  и  $m$  необходимо настраивать индивидуально для каждого ряда [4].

Полное правило имеет вид

$$\Omega_j = \text{if} \left( \bigwedge_{i=1}^z \omega_i \right) \text{ then } \omega_c,$$

где  $\omega_i$  – атомарное подправило условной части,  $i$  – номер атомарного подправила,  $i = \overline{1, z}$ ,  $z$  – длина правила (количество атомарных подправил в правиле),  $\omega_c$  – атомарное подправило, используемое в случае выполнения условия *if*.

Отметим, что в правилах используется относительная индексация, а не абсолютные значения моментов времени  $t$ , т.е. все индексы подправил определяются расстоянием относительно начального нулевого индекса. Таким образом, все правила представляют собой шаблон,

применяемый к временному ряду последовательным сдвигом  $\hat{\omega} = \Delta(s, k) \in [y_b, y_{b+1})$ , где  $s, k$  – индексы, определяющие расстояние относительно начального индекса полного правила.

Рассмотрим графически интерпретацию полного правила с одним подправилом и его применение к временному ряду (рис. 2). Правило

$$\text{if} (\Delta(s_0, s_7) \in [0, 50)) \text{ then } (\Delta(s_7, s_{16}) \in [0, 50))$$

говорит о том, что если прирост функции в двух точках, расстояние между которыми равно семи, будет в пределах от нуля до 50, то прирост между седьмой и 16-й точками будет в пределах 250–300. Шаблон (правило) проверяется для всех  $t$ . Очевидно, что правило выполняется при  $s_0 = 10, s_1 = 17, s_2 = 26$ .

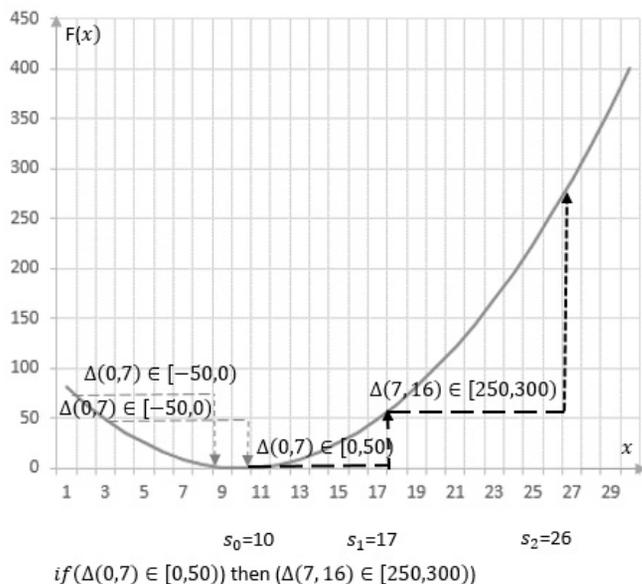


Рис. 2. Графическая интерпретация применения правила

### Фитнес-функция

Качество каждой хромосомы популяции (правила) оценивается значением фитнес-функции  $\phi$ , которая учитывает информацию о частоте выполнения правила  $\phi$  на временном ряду  $F$  и длину правила  $z$ :

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

где  $\delta$  – параметр алгоритма для контроля минимально допустимой частоты выполнения правила,  $\alpha$  – параметр алгоритма для регули-

рования веса значения длины правила и частоты выполнения при расчете значения фитнес-функции.

Несмотря на определение правила как строгой конъюнкции условий, считаем, что оно выполняется в момент времени  $t$ , если процент выполненных атомарных подправил  $\omega_i$  выше определенного параметра алгоритма  $\varpi$  и полностью выполняется  $\omega_c$ .

Частота выполнения правила  $\varphi(\Omega_j)$  рассчитывается последовательным сдвигом индексов правила по временному ряду и проверкой выполнения правила в каждый момент времени  $t$ :

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

где  $\Omega_j^t$  – значение выполнения правила  $\Omega_j$  ( $\Omega_j^t = 1$ , если правило выполняется в момент  $t$ ,  $\Omega_j^t = 0$  в других случаях).

### Определение подобных правил. Инициализация

Когда ГА используются для оптимизации, то целью, как правило, является возвращение единственного значения – наилучшего найденного решения. Обычно, если традиционный ГА выполняет достаточное количество итераций, вся популяция обычно сходится к окрестности одного решения [8]. Для прогнозирования такой подход непригоден, поскольку необходимо получить достаточно широкую базу правил для построения финального прогноза. Поэтому актуальна разработка ГА, которые используют процедуры построения ниш и способны находить и использовать различные правила из одной популяции. Основная идея построения ниш – одновременная оптимизация в нескольких областях пространства поиска путем снижения конкуренции между достаточно разнородными хромосомами. В финансовом прогнозировании различные правила в пределах одной популяции ГА могут выполнять прогнозирование в различных условиях рынка и конкретных компаниях.

Существуют различные стратегии построения ниш: *обмен фитнесом* [9] – снижение зна-

чения фитнеса у подобных хромосом популяции; в процедурах *вытеснения* новые хромосомы замещают существующие элементы популяции и пр.

Для реализации процедуры построения ниш авторами предложено ввести понятие подобия правил – степень сходства схемы хромосом. Два правила подобны, если процент подобных подправил  $\tilde{\omega}$  больше заданного параметра модели  $\sigma$ ,  $\sigma \in \{1, \dots, 100\}$ . Подправила  $\omega_1 = \Delta(s_1, k_1) \in [y_{b_1}, y_{b_1+1})$  и  $\omega_2 = \Delta(s_2, k_2) \in [y_{b_2}, y_{b_2+1})$  считаются подобными, если они характеризуются похожим сдвигом относительно начального индекса правила, покрывают отрезок похожего размера на временном ряде и принадлежат похожему интервалу значений:

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

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

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

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

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

где  $s_1, s_2$  – начальные индексы подправила,  $k_1, k_2$  – конечные индексы подправил,  $\vartheta$  – параметр, характеризующий допустимую погрешность сдвига начальных индексов подобных подправил,  $\varsigma$  – параметр, характеризующий погрешность длины отрезка, покрываемый правилом на временном ряде,  $b_1, b_2$  – индексы, определяющие интервалы (ранги), которым принадлежит подправило,  $\gamma$  – параметр, характеризующий допустимую погрешность рангов подобных подправил.

Два правила считаются подобными, если процент подобных подправил выше определенного значения параметра алгоритма  $\rho$ .

Процедура поиска подобия используется для исключения избыточности правил на этапе рекомбинации, при добавлении правил в популяцию на этапе отбора к выживанию с помощью процедуры вытеснения, на этапе отбора к рекомбинации для способов инбридинг и аутбридинг и т.д.

Начальная популяция имеет фиксированный размер, заданный параметром алгоритма, и строится по заданному временному ряду с учетом минимального допустимого размера правила, заданного параметром алгоритма.

Подправила строятся путем случайного выбора моментов на временном ряду и сопоставления приростов в этих моментах. Таким образом, начальная популяция всегда содержит правила с минимальным количеством выполнения, равным единице. При этом для каждого нового правила выполняется процедура проверки подобности существующих правил и оценка фитнеса.

### Отбор для рекомбинации. Процедуры скрещивания, мутации

Отбором для рекомбинации определяются родительские пары для дальнейшего скрещивания и мутации. В алгоритме реализованы такие способы отбора для рекомбинации: панмиксия, инбридинг, аутбридинг и рулетка [10]. Способ отбора неизменный на протяжении всего запуска алгоритма и определяется в начале работы параметром модели.

*Панмиксия (panmixia)* – отбор, в котором любые две хромосомы имеют одинаковую возможность образовать случайным образом «брачную пару». *Инбридинг (inbreeding)* – отбор, в котором во время образования «брачной пары» предпочтение отдается хромосомам с генетически похожим кодированием. *Аутбридинг (autobreeding)* – отбор, в котором во время образования «брачной пары» предпочтение отдается хромосомам с генетически непохожими кодировками. *Турнирный отбор (tournament selection)* – отбор, в котором из популяции выбираются случайно  $1 \leq d < N$  вариантов (где  $N$  – размер популяции), среди которых «брачную пару» образуют лучшие. *Рулетка (roulette-wheel selection)* – стохастический отбор на основе пропорциональной приспособленности хромосом.

Для реализации инбридинга и аутбридинга используются процедуры поиска подобных правил. В качестве сравнительной характеристики подобных правил используется процент подобных подправил.

Одна из целей данной статьи – исследование влияния этих видов отбора на результаты.

Рекомбинация состоит из двух этапов: скрещивания и мутации. Операторы скрещивания реализуются по-разному в зависимости от представления данных, однако при их разработке стараются обеспечить наследование черт обоих родителей. Представление правила в виде неупорядоченного множества атомарных подправил позволяет определять точку сечения произвольно. В алгоритме используется одноточечное скрещивание, т.е. для двух выбранных правил  $\Omega^1$  и  $\Omega^2$  произвольно выбирается точка скрещивания  $p$  (индекс подправила), и формируются два наследника  $\Omega^{12}$  и  $\Omega^{21}$ :

если

$$\Omega^1 = \text{if} \left( \omega_1^1 \wedge \omega_2^1 \wedge \dots \wedge \omega_p^1 \wedge \omega_{p+1}^1 \wedge \dots \wedge \omega_{z-1}^1 \wedge \omega_z^1 \right)$$

then  $\omega_c^1$ ,

$$\Omega^2 = \text{if} \left( \omega_1^2 \wedge \omega_2^2 \wedge \dots \wedge \omega_p^2 \wedge \omega_{p+1}^2 \wedge \dots \wedge \omega_{z-1}^2 \wedge \omega_z^2 \right)$$

then  $\omega_c^2$ ,

то

$$\Omega^{12} = \text{if} \left( \omega_1^1 \wedge \omega_2^1 \wedge \dots \wedge \omega_p^1 \wedge \omega_{p+1}^2 \wedge \dots \wedge \omega_{z-1}^2 \wedge \omega_z^2 \right)$$

then  $\omega_c^2$ ,

$$\Omega^{21} = \text{if} \left( \omega_1^2 \wedge \omega_2^2 \wedge \dots \wedge \omega_p^2 \wedge \omega_{p+1}^1 \wedge \dots \wedge \omega_{z-1}^1 \wedge \omega_z^1 \right)$$

then  $\omega_c^1$ .

С определенной вероятностью  $\nu$  – параметр модели, проводится процедура мутации. Мутировать может произвольное подправило  $\tilde{\omega} = \Delta(s, k) \in [y_b, y_{b+1})$  – в нем могут изменяться  $s, k$  или ранг  $b$  согласно допустимому пространству значений.

После операций рекомбинации проводятся процедуры перестроения индексов относительно начального, проверка подобных подправил и оценка фитнеса потомков.

### Отбор для выживания. Правило остановки работы генетического алгоритма

После этапа рекомбинации потомки добавляются в популяцию, если подобных хромосом нет, или замещают существующие подобные хромосомы, если обладают более высоким значением фитнеса. Следует отметить, что в дан-

ной реализации существует порог минимального значения количества выполнений правил на временном ряде, при котором потомки могут попасть в популяцию.

Критерий остановки алгоритма – проведение заданного количества итераций  $I$ . Результатом работы ГА есть популяция правил – множество правил, обладающих определенной оценкой качества – значением фитнес-функции, по которым строится прогноз.

### Построение прогноза

Этап *согласования*. Каждое правило базы проверяется на допустимость применения для построения прогноза для каждой прогнозируемой позиции  $\tilde{t}$ . Правила проверяются для таких значений временного ряда, где условная часть принадлежит известным значениям временного ряда, а часть *then* принадлежит периоду опережения. Если условная часть некоторого правила выполняется для некоторой позиции  $t$ , то посредством части *then* строится возможный прогноз.

В результате этапа согласования каждой прогнозируемой позиции  $\tilde{t}$  соответствует свой набор конфликтов  $\{ \langle \Omega_j, \phi, offset \rangle \}$  длины  $h_{\tilde{t}}$ , содержащих информацию о правиле  $\Omega_j$ , его степени приспособленности  $\phi$  и смещение правила на временном ряде  $offset$ . Таким образом, каждой позиции  $\tilde{t}$  соответствует  $h_{\tilde{t}}$  альтернативных значений  $\hat{G}$  прогноза.

Этап *разрешения конфликтов*. Предложено определять агрегированное значение по всем конфликтующим значениям  $\hat{G}$  с учетом приоритетности каждого правила, которая определяется степенью его приспособленности. Описанная интерпретация значения приспособленности содержит информацию о специфичности правила и его частоте выполнения на временном ряде.

Этап *выполнения*. На вход алгоритма прогнозирования подаются набор конфликтов, данные временного ряда, количество итераций планирования, т.е. время опережения. Тогда прогнозируемое значение рассчитывается по формуле:

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

где  $\tilde{t}$  – позиция в периоде опережения, для которой рассчитывается прогноз,  $\tilde{t} = \overline{1, P}$ ,  $P$  – период упреждения,  $\hat{F}_{\tilde{t}}$  – результирующее значение прогноза для позиции  $\tilde{t}$ ,  $\hat{G}_i$  – конфликтующее значение прогноза для позиции  $\tilde{t}$ ,  $h_{\tilde{t}}$  – количество конфликтов для позиции  $\tilde{t}$ ,  $\phi_i$  – степень приспособленности правила, спровоцировавшего конфликт  $\hat{G}_i$ ,  $i = \overline{1, h_{\tilde{t}}}$ .

Особенность предлагаемого подхода – это участие в построении прогноза всех правил из набора конфликтов. Отметим также, что каждое следующее прогнозируемое значение учитывает значения, полученные ранее.

### Исследование практического применения разработанного алгоритма прогнозирования

На основе предложенного подхода к прогнозированию был разработан специализированный программный комплекс на языке C#. Расчеты проводились на виртуальной машине *Microsoft Azure A8*, предназначенной для ресурсоемких вычислений, имеющей следующие характеристики: процессор *Intel Xeon 2.5 Gh*, 8 ядер, 56 Гб оперативной памяти. Для оценки эффективности созданных программно-алгоритмических средств был проведен вычислительный эксперимент по прогнозированию на основе использования реального временного ряда (рис. 3). Для разработки этого прогноза использован помесечный временной ряд объемов фактических продаж медицинских препаратов в аптеках Украины за период 2004–2013 гг., выраженный в гривнях. Для достижения цели вычислительного эксперимента временной ряд был разделен на учебную выборку (2004–2012 гг.). Эти данные использовались для обучения с применением разработанного ГА и построения базы правил; данные тестовой выборки (2012 г.) были известны алгоритму при

обучении, а затем на основе этих данных был построен прогноз для оценки ошибки; данные контрольной выборки (2013 г.) были недоступны алгоритму обучения.

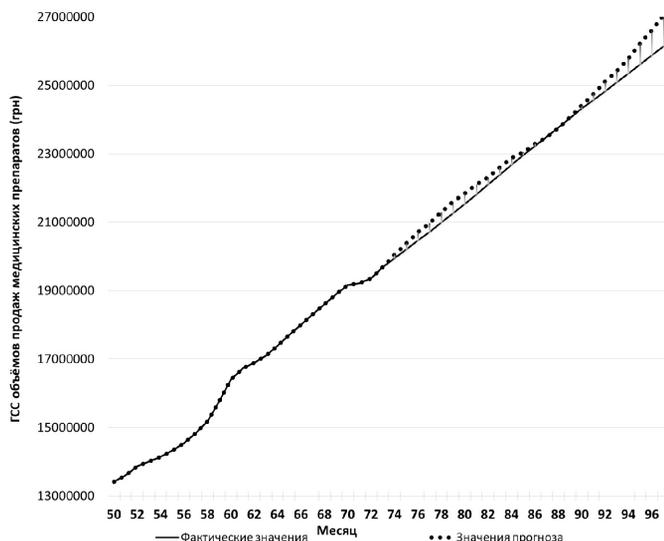


Рис. 3. Прогноз годовых скользящих сумм объемов продаж медицинских препаратов

Оценка ошибки выполняется на основе использования известных критериев  $MPE$ ,  $MAPE$  [11]:

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

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

где  $\hat{F}_t$  – спрогнозированное значение,  $F_t$  – фактическое значение,  $N$  – количество точек для сравнения (обычно – имеющийся временной ряд),  $t$  – момент времени,  $t = \overline{1, N}$ .

В таблице приведены фактические данные за 2012 г., прогнозные значения на 2012 г. (тестовая выборка), фактические данные за 2013 г., прогнозные значения на 2013 г. (контрольная выборка), полученные в результате вычислительного эксперимента.

Из результатов эксперимента следует, что усредненные значения ошибок прогноза по критерию  $MPE$  на тестовой и контрольной выборках составляют  $-1,1\%$  и  $-1,2\%$  соответственно, а по критерию  $MAPE$  –  $1,1\%$  и  $1,2\%$ , что можно считать приемлемым на практике.

#### Результаты прогнозирования

№	Фактическое значение на тестовой выборке	Прогнозируемое	Фактическое значение на контрольной выборке	Прогнозируемое
1	19949504	20059232	23228344	23295940
2	20206460	20387146	23494700	23487183
3	20463903	20712021	23765372	23762372
4	20721069	20981187	24051883	24069264
5	20993146	21297783	24323290	24412651
6	21261310	21593159	24567628	24754267
7	21530600	21851015	24828381	25108251
8	21817053	22098811	25091582	25418312
9	22105440	22312336	25350755	25794917
10	22393529	22604263	25609683	26224947
11	22679450	22882337	25876747	26606814
12	22961491	23044748	26141320	27063485
<b>MAPE</b>		<b>1,1%</b>		<b>1,2%</b>
<b>MPE</b>		<b>-1,1 %</b>		<b>-1,2%</b>

#### Настройка алгоритмов прогнозирования

Проведен эксперимент для определения оптимальных параметров алгоритма. Количество экспериментов  $K = 50$  для каждой комбинации значений параметров, период опережения прогноза составляет 12 месяцев. Для эксперимента использовался сглаженный ряд годовых скользящих сумм объемов продаж медицинских препаратов.

На рис. 4 представлена зависимость ошибки  $MPE$  от способа отбора для рекомбинации и минимального размера хромосомы  $z$  на этапе инициализации при зафиксированных остальных значениях параметров алгоритма ( $\vartheta = 3$ ,  $\varsigma = 3$ ,  $\rho = 70\%$ ,  $\gamma = 3$ ,  $m = 100$ ,  $\alpha = 0,8$ ,  $\delta = 1$ ,  $I = 100$ ,  $\upsilon = 20\%$ ). Очевидно, что при росте начального размера хромосомы ошибка увеличивается. Это можно объяснить тем, что при росте размера хромосомы количество выполненных правил уменьшается и они теряют обобщающую способность. Необходимо дополнительно исследовать возможность одновременного увеличения точности начальных правил (размера хромосом), размера всей начальной популяции и количества итерации обучения. Однако исследование этих параметров есть одним из наиболее ресурсоемких, поэтому может стать темой отдельного исследования.

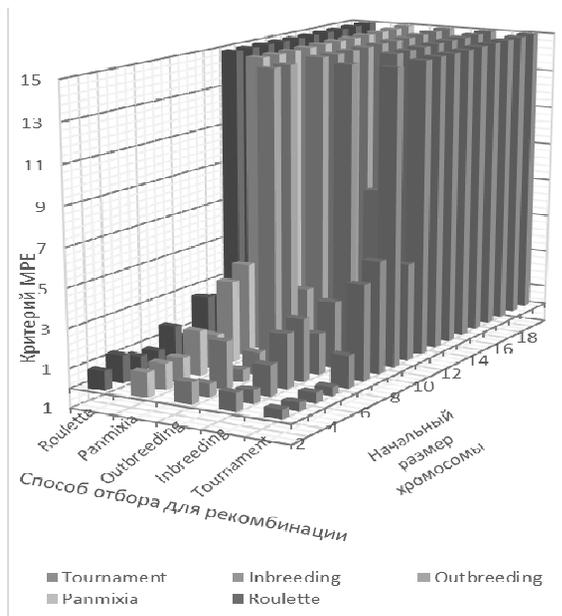


Рис. 4. Зависимость ошибки *MPE* от способа отбора для рекомбинации и размера хромосомы

На рис. 5 отображена агрегированная погрешность прогнозов в зависимости от способа отбора для рекомбинации.

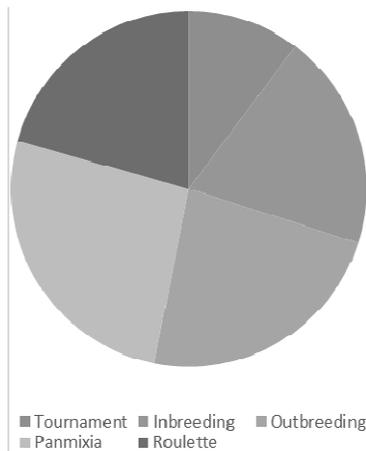


Рис. 5. Зависимость ошибки *MPE* от способа отбора для рекомбинации и размера хромосомы

На рис. 6 и 7 представлена зависимость критерия *MAPE* от способа отбора для рекомбинации и количества итераций при зафиксированных остальных значениях параметров алгоритма ( $\vartheta=3$ ,  $\zeta=3$ ,  $\rho=70\%$ ,  $\gamma=3$ ,  $m=100$ ,  $\alpha=0,8$ ,  $\delta=1$ ,  $z=3$ ,  $\nu=20\%$ ).

Из результатов, представленных на рис. 5, следует, что наилучшее агрегированное значение *MAPE* достигается при  $I=600$ . При этом данные на рис. 6 свидетельствуют о том, что

разным значениям длительности обучения подходят различные способы отбора для рекомбинации: для более длительного обучения больше подходит аутбридинг и инбридинг.

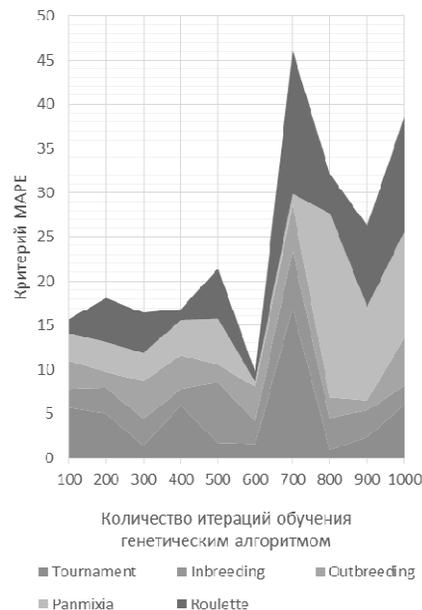


Рис. 6. Зависимости критерия *MAPE* от способа отбора для рекомбинации и количества итераций обучения генетическим алгоритмом

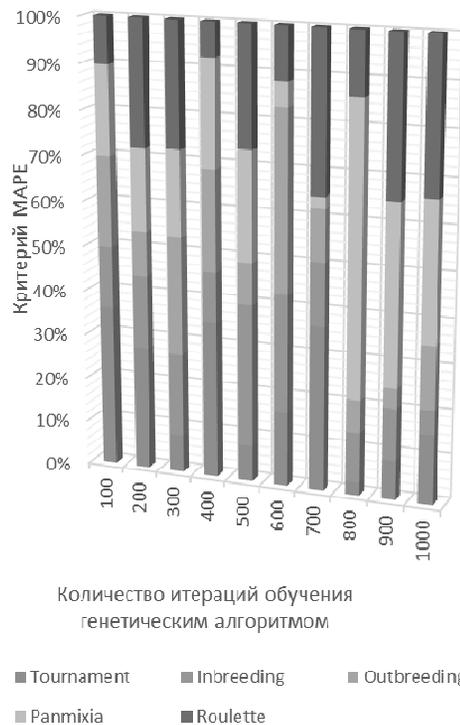


Рис. 7. Зависимость критерия *MAPE* от способа отбора для рекомбинации и количества итераций обучения генетическим алгоритмом

**Заключение.** Предложенный способ представления данных для задачи прогнозирования позволяет применять разработанный алгоритм с несколькими временными рядами. Способ оценки качества хромосомы с учетом ее длины и частоты выполнения позволяет учитывать качество правила на этапе разрешения конфликта и построения финального прогноза. Предложен механизм для диверсификации в пространстве поиска. Описаны генетические операторы с учетом предложенного способа представления данных.

Вычислительные эксперименты проведены для выбора оптимальных значений параметров алгоритма.

Выбор оптимальных значений параметров алгоритма построения прогноза с учетом влияния нескольких факторов, представленных разными временными рядами, требуют дополнительного исследования.

1. *Eiben A.E., Smith J.E.* Introduction to Evolutionary Computing. Natural Computing Series. – Berlin, Heidelberg, N.Y.: Springer, 2007. – 300 p.
2. *Mahfoud S., Mani G.* Financial forecasting using genetic algorithms // Applied Artificial Intelligence. – 1996. – 10, N.6. – P. 543–560.

3. *Гуляницкий Л.Ф.* Разработка методов прогнозирования на основе эволюционных алгоритмов // Компьютерная математика. – 2012. – 1. – С. 69–77.
4. *Павленко А.И.* Застосування генетичного алгоритму для прогнозування на основі бази правил // Компьютерная математика. – 2014. – № 2. – С. 74–80.
5. *A Practical Introduction to Rule Based Expert Systems / M. Sasikumar, S. Ramani, S. Muthu Raman et al.* – New Delhi: Narosa Publishing House, 2007. – 294 p.
6. *De Jong K.A., Spears W.M., Gordon D.F.* Using genetic algorithms for concept learning // Machine Learning, 1993. – P. 161–188.
7. *Гуляницкий Л.Ф., Павленко А.И.* Прогнозирование на основе генетического алгоритма обучения // Proc. XX Int. Conf. «Knowledge-Dialogue-Solution», Sept. 8–10, 2014, Kyiv. – Kyiv–Sofia: ITHEA, 2014. – P. 41–42.
8. *Mahfoud S.W.* Niching methods for genetic algorithms. – Dissert. Abstracts Int. – University Microfilms, 1995. – 251 p.
9. *Goldberg D.E., Richardson J.* Genetic algorithms with sharing for multimodal function optimization / Genetic algorithms and their applications // Proc. of the Second Int. Conf. on genetic algorithms, 1987. – P. 41–49.
10. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – М.: ФИЗМАТЛИТ, 2006. – 320 с.
11. *Ханк Д.Э., Уичерн Д.У., Раймс А.Дж.* Бизнес-прогнозирование. – М.: Вильямс, 2003. – 656 с.

Поступила 02.02.2015

E-mail: [leonhul.icyb@gmail.com](mailto:leonhul.icyb@gmail.com), [dmitrieva.anya@gmail.com](mailto:dmitrieva.anya@gmail.com)  
© Л.Ф. Гуляницкий, А.И. Павленко, 2015

Окончание статьи А.П. Сарычева

1. *Ивахненко А.Г.* Индуктивный метод самоорганизации моделей сложных систем. – Киев: Наук. думка, 1982. – 296 с.
2. *Self-organizing methods in modelling: GMDH type algorithms / Ed. by S.J. Farlow.* – New York, Basel: Marcel Decker Inc., 1984. – 350 p.
3. *Ивахненко А.Г., Степанко В.С.* Помехоустойчивость моделирования. – Киев: Наук. думка, 1985. – 216 с.
4. *Ивахненко А.Г., Мюллер Й.А.* Самоорганизация прогнозирующих моделей. – Киев: Техніка, 1985. – 223 с.
5. *Ивахненко А.Г., Юрачковский Ю.П.* Моделирование сложных систем по экспериментальным данным. – М.: Радио и связь, 1987. – 120 с.
6. *Madala H.R., Ivakhnenko A.G.* Inductive Learning Algorithms for Complex System Modeling. – London, Tokyo: CRC Press Inc., 1994. – 370 p.
7. *Muller J.-A., Lemke F.* Self-organizing Data Mining. Extracting Knowledge from Data. – Hamburg: Libri, 2000. – 250 p.

8. *Сарычев А.П.* Идентификация состояний структурно-неопределенных систем. – Днепропетровск: Ин-т техн. механики НАН и НКА Украины, 2008. – 268 с.
9. *Ермаков С.М., Жиглявский А.А.* Математическая теория оптимального эксперимента. – М.: Наука, 1987. – 320 с.
10. *Себер Дж.* Линейный регрессионный анализ. – М.: Мир, 1980. – 456 с.
11. *Сарычев А.П.* Решение проблемы разбиения в МГУА при расчете критерия регулярности в условиях активного эксперимента // Автоматика. – 1989. – № 4. – С. 19–27.
12. *Сарычев А.П.* Определение J-оптимального множества регрессоров по повторным выборкам наблюдений // Там же. – 1993. – № 3. – С. 58–66.

Поступила 05.12.2014

Тел. для справок: +38 0562 46-5149 (Днепропетровск)  
E-mail: [Sarychev@prognoz.dp.ua](mailto:Sarychev@prognoz.dp.ua)  
© А.П. Сарычев, 2015