УДК 004.021
ББК 3.30

А. И. Звезинцев, И. Ю. Квятковская

ПРИМЕНЕНИЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ИДЕНТИФИКАЦИИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПУТЁМ РАСШИРЕНИЯ ОБУЧАЮЩЕГО МНОЖЕСТВА ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТЬЮ

A. I. Zvezintsev, I. Yu. Kvyatkovskaya

APPLICATION OF MODIFIED GENETIC PROGRAMMING ALGORITHM FOR IDENTIFICATION OF MATHEMATICAL MODELS THROUGH THE EXPANSION OF THE TRAINING SET BY NEURAL NETWORK

Рассмотрено понятие математической идентификации, область её применения и этапы проведения. Проанализированы методы идентификации математических моделей: регрессионный анализ, гармонический анализ, метод группового учёта аргументов, генетическое программирование. Рассмотрено ограничение использования метода генетического программирования для идентификации математической модели неизученного процесса при наличии шумовой составляющей в экспериментальных данных. Предлагается модификация метода генетического программирования способом предварительной аппроксимации и расширения обучающего множества искусственной нейронной сетью. Приведены интерфейсы разработанного программного продукта и результаты тестирования предлагаемого метода.

Ключевые слова: математическая идентификация, генетическое программирование, искусственная нейронная сеть, аппроксимация, извлечение знаний, математическая модель.

The concept of mathematical identification, its scope and stages of implementation are considered. The methods of identification of mathematical models: regression analysis, harmonic analysis, group method of data handling, genetic programming are analyzed. The restriction of the use of genetic programming method for the identification of the mathematical model of unexplored process in the presence of the noise component in the experimental data is studied. Proposes a modification of the method of genetic programming using the method of pre-approximation and expanding the training set by artificial neural network. The interfaces of the developed software product and the test results of the proposed method are presented.

Key words: mathematical identification, genetic programming, artificial neural network, approximation, knowledge extraction, mathematical model.

Введение

К методам математической идентификации прибегают каждый раз, когда требуется изучить новый процесс или систему: определить аналитический вид функциональной зависимости, скрытой «внутри» процесса, произвести предсказание состояния системы в некоторый момент времени либо при наступлении определённых условий, выработать стратегию управления процессом посредством воздействий на доступные «рычаги». Исследования физических, социальных, экономических процессов требуют использования методов математической идентификации [1–5].

Процесс идентификации математических моделей, как правило, разделяется на два последовательных этапа: проведение структурной и параметрической идентификации.

При реализации первого этапа – *структурной идентификации*, исходя из набора теоретических предположений о природе изучаемого процесса выделяют классы функциональных зависимостей, которые потенциально могут описывать данный процесс. Из них строят общий аналитический вид зависимости, не уточняя коэффициенты.

Второй этап – *параметрическая идентификация* – позволяет определить коэффициенты при всех членах выбранного аналитического представления изучаемого процесса.

Разбор существующих методов. Преимущества, недостатки, ограничения

Существует ряд методов, с разной степенью успешности справляющихся с задачей идентификации математических моделей различной природы. Рассмотрим некоторые из них.

Методы регрессионного анализа – полиномиальная регрессия, экспоненциально-степенная, логарифмическая регрессия – требуют знания аналитического вида исследуемой зависимости [6]. Перед применением методов данной категории исследователь должен сделать предположение о виде исследуемой зависимости (линейная, квадратичная или иного вида). С помощью регрессионного анализа находятся численные значения параметров формулы выбранной зависимости. По сути, методы регрессионного анализа позволяют решить лишь задачу параметрической идентификации, но не задачу структурной идентификации, решение которой представляет большую сложность. Теоретически исследователь имеет возможность выдвинуть целый ряд предположений о виде исследуемой зависимости и, подобрав параметры с помощью методов регрессионного анализа, выбрать наиболее подходящий вид зависимости. Но на практике возможен вариант, когда исследователь не может выдвинуть подобные предположения, потому что у него недостаточно информации о внутренней структуре изучаемого процесса. Ведь многие процессы, в частности в области экономики, имеют значительную структурную сложность стоящей за ними математической зависимости, и заранее, с помощью каких-либо методов анализа, выявить её формулу не представляется возможным.

При анализе процессов с помощью методов гармонического анализа изучение ведётся путём разложения исследуемой зависимости в ряды Фурье. Дальнейший анализ гармоник различных порядков позволяет извлечь ценные сведения о вкладе каждого из гармонических колебаний, которые при суммарном воздействии образуют исследуемый гармонический процесс.

Важнейший недостаток метода гармонического анализа заключается в том, что он подходит лишь для изучения процессов, имеющих периодическую, колебательную природу. Кроме того, гармонический анализ позволяет решить задачу лишь параметрической идентификации.

Отличным от прочих методом идентификации математических моделей является *метод группового учёта аргументов* [7]. Он позволяет проводить и структурную, и параметрическую идентификацию. Работа метода базируется на итеративном усложнении построенной математической модели до тех пор, пока не будет получена модель, с нужной степенью достоверности отражающая изучаемый процесс.

Работа метода группового учета аргументов в общем случае состоит из 4 шагов.

Шаг 1. Строятся предположения о том, какие примитивные функциональные зависимости (функции) могут лежать в основе исследуемого процесса. Выбирается способ комбинирования выбранных функций в итоговую модель, называемую опорной функцией. Чаще всего для этого используют полином Колмогорова – Габора.

Шаг 2. Случайным образом генерируют всевозможные модели путём комбинирования различным образом выбранных функций в опорную функцию. Затем с помощью регрессионного анализа определяют коэффициенты в сгенерированных моделях.

Шаг 3. Производят тестирование полученных моделей. Если не выполнено условие останова работы алгоритма (получена достаточно хорошая модель либо достигнута максимально допустимая сложность модели), то наилучшие модели отбирают для следующего – 4 шага алгоритма. Если условие останова выполнено – выход из алгоритма, лучшую полученную за время работы модель принимают как результат работы алгоритма.

Шаг 4. Происходит возврат на шаг 2, но теперь в роли аргументов для функций в опорной функции выступают отобранные модели.

Таким образом, в методе группового учета аргументов с каждым этапом работы алгоритма итеративно получают всё более сложные модели, которые всё лучше описывают изучаемый процесс.

Главным ограничением при работе метода группового учёта является значительный объём перебираемых результатов, следствием чего является медленная сходимость метода и значительное время его работы.

Перспективы использования генетического программирования с предварительным расширением обучающего множества нейросетью. Возможные преимущества, недостатки, ограничения

Генетическое программирование является методологией идентификации математических моделей, позволяющей представлять модели в аналитическом виде. Отметим, что при этом выполняется и структурная, и параметрическая идентификация моделей [8, 9].

Генетическое программирование относится к классу эволюционных алгоритмов и отчасти похоже на метод группового учета аргументов. В отличие от генетических алгоритмов, метод генетического программирования оперирует не битовыми строками, а программами или функциями, представленными, как правило, в виде деревьев.

Рассмотрим упрощённый вариант генетического программирования, когда хромосомы являются не программами, а математическими функциями, представленными в виде деревьев разбора [10]. В качестве функций в хромосомах могут использоваться любые математические функции и их комбинации: алгебраические, трансцендентные, специальные функции, функции теории чисел, гамма-функции и др.

Хромосома обычно реализована в виде функции на языке программирования, аргументами которой являются входные переменные описываемой математической функции, а возвращаемым значением – результат преобразования входных переменных данной математической функцией. Например, хромосома может описывать операцию «сложение двух чисел», и в этом случае она реализуется в виде программной функции, принимающей на вход два числа и возвращающей их сумму.

В общем случае работу генетического программирования можно представить в виде последовательности из 4 шагов.

Шаг 1. На основе теоретических предположений и каких-либо знаний об исследуемом процессе выделяют классы функций, которые потенциально могут содержаться в математической модели исследуемого процесса.

Шаг 2. Случайным образом строят начальную популяцию хромосом, состоящую из нескольких десятков или даже нескольких тысяч функций.

Шаг 3. Проводят отбор функций, наилучшим образом справляющихся с поставленной задачей, из текущей популяции, а также их видоизменение путём проведения мутаций и «размножение» путём проведения кроссовера между ними.

Шаг 4. Если не выполнено условие останова (найдено достаточно хорошее решение, либо время работы алгоритма превысило допустимое) – возвращаются на шаг 2, но вместо случайно созданных функций используют те функции, которые получены на шаге 3. Если условие останова выполнено – выход из алгоритма, лучшую полученную за время работы модель принимают как результат работы алгоритма.

Таким образом, итеративно происходит получение функций, всё более успешно справляющихся с поставленной задачей.

Методологию генетического программирования можно применять в различных областях: создание программ, проявляющих необходимое поведение, конструирование объектов с необходимыми характеристиками, а также построение математических моделей изучаемых процессов [8].

Нами предлагается модификация метода генетического программирования при его использовании для задачи идентификации математических моделей: предварительное «сглаживание» и дальнейшее расширение обучающего множества для генетического программирования с помощью предварительной аппроксимации обучающего множества искусственной нейронной сетью. Концептуальная схема работы предлагаемого метода выглядит следующим образом.

Шаг 1. Внесение в программу исходных экспериментальных данных, а именно набора входных значений и соответствующих им выходных: N пар векторов вход-выход.

Шаг 2. Обучение на внесённых данных искусственной нейронной сети – многослойного персептрона.

Шаг 3. Использование обученной искусственной нейронной сети для генерации расширенного множества входных значений и соответствующих им выходных: M пар векторов вход-выход, причем $M > N$.

Шаг 4. Построение модели исследуемого процесса с помощью метода генетического программирования, причём в качестве множества пар вход-выход для оценивания полученных моделей будем использовать множество, сгенерированное на шаге 3.

Как известно, искусственная нейронная сеть является аппроксиматором функции многих переменных, она фактически позволяет аппроксимировать сколь угодно сложные нелинейные многомерные зависимости. Поэтому, обучив нейронную сеть на ограниченном наборе данных, можно с её помощью получать дополнительное множество значений аппроксимируемой функции.

При решении задачи математической идентификации неизученного процесса полученные в ходе исследования этого процесса экспериментальные данные, как правило, имеют «шумовую» составляющую, объясняемую несовершенством методов измерений. Наличие шума в экспериментальных данных, которые являются обучающим множеством для методов идентификации моделей, осложняет построение качественных моделей методами математической идентификации.

Предположительно предлагаемый подход позволит генетическому программированию лучше справляться с задачей идентификации математических моделей при работе на зашумленном наборе экспериментальных данных.

У предлагаемого подхода, кроме того, имеется потенциальное ограничение: чтобы генетическое программирование лучше справилось с задачей на расширенном наборе входных данных, искусственная нейронная сеть должна хорошо саппроксимировать экспериментальные данные. Отсюда ограничение: исходных экспериментальных данных должно быть достаточно для проведения качественного обучения нейронной сети.

Для проверки преимуществ и ограничений работы предлагаемого модифицированного подхода к идентификации средствами генетического программирования был разработан программный продукт, основные возможности которого:

– выполнять аппроксимацию набора экспериментальных данных средствами искусственной нейронной сети (многослойный персептрон, алгоритм обратного распространения ошибки);

– выполнять идентификацию математической модели в аналитическом виде по экспериментальным данным методом генетического программирования.

Разработанное программное обеспечение обладает также рядом дополнительных возможностей.

1. Использование в качестве исходных экспериментальных данных (при обучении нейросети и идентификации модели методом генетического программирования) пользовательской функции, представляемой пользователем в аналитическом виде (в том числе с возможностью автоматического «зашумления»).

2. Всесторонняя настройка нейросети и генетического программирования.

3. Использование в качестве обучающего множества для генетического программирования расширенного множества, полученного от искусственной нейронной сети после аппроксимации ею исходных экспериментальных данных.

4. Визуализация построенной с помощью генетического программирования модели в виде дерева разбора.

5. Механизм автоматического тестирования предлагаемого в данной статье модифицированного метода генетического программирования.

Апробация: интерфейс разработанного программного обеспечения, результаты использования на различных наборах данных

Пример интерфейса главного окна разработанного программного обеспечения приведён на рисунке.

Алгоритм работы пользователя с разработанным программным продуктом:

1. Задание исходной функциональной зависимости на языке C# для генерации обучающего набора входных и выходных данных с возможностью автоматического зашумления.

2. Генерация набора входных и выходных данных обучающего множества.

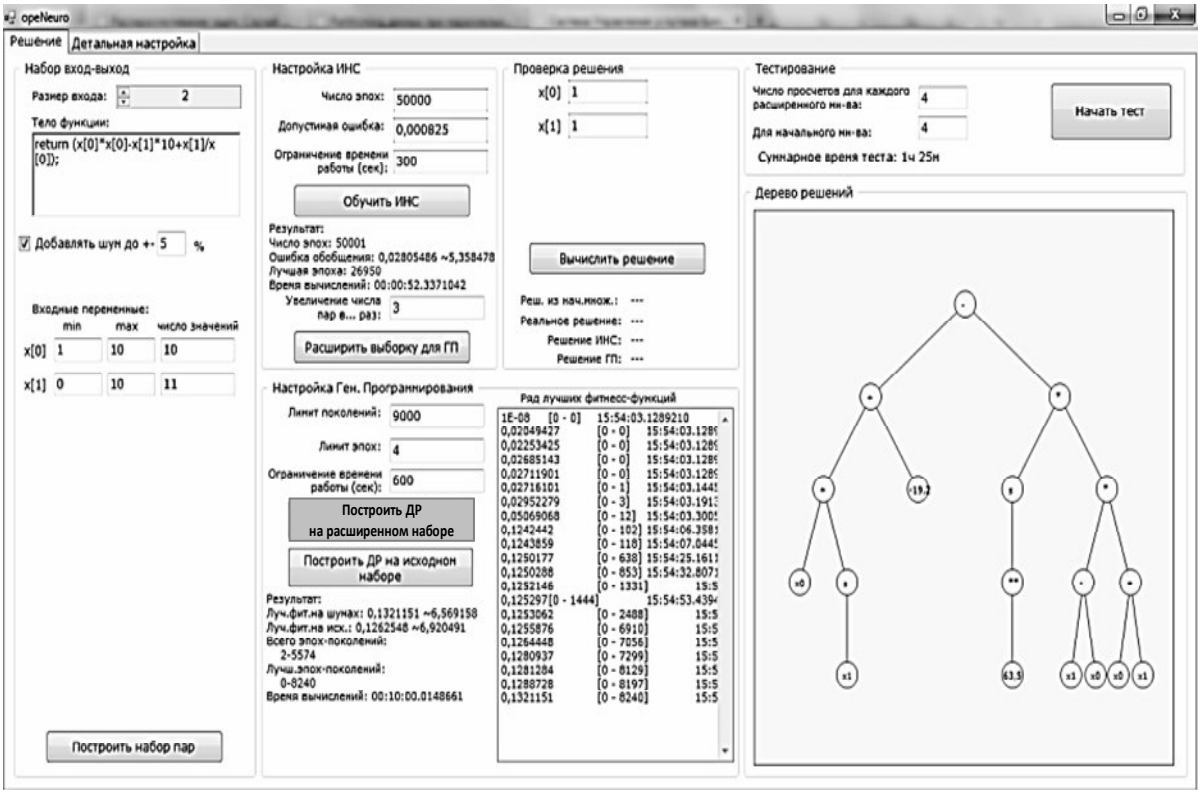
3. Настройка параметров искусственной нейронной сети.

4. Обучение искусственной нейронной сети.

5. Расширение набора входных и выходных данных с помощью обученной искусственной нейронной сети для использования расширенного набора в качестве обучающего набора для генетического программирования (шаг можно пропустить, тогда генетическое программирование будет работать на исходном наборе данных).

6. Настройка параметров алгоритма генетического программирования.

7. Получение с помощью генетического программирования результирующей математической модели в виде дерева разбора.



Интерфейс главного окна разработанного программного обеспечения

Было проведено тестирование, целью которого являлось получить предварительный вывод о целесообразности проведения предлагаемой модификации метода генетического программирования. Конкретная цель – получить предварительный ответ на вопрос: «В каком случае генетическое программирование на зашумленных исходных данных работает лучше: в случае применения генетического программирования на исходном наборе данных или же в случае применения его на расширенном наборе данных, полученном с помощью аппроксимации исходных экспериментальных данных силами искусственной нейронной сети и дальнейшего их расширения?».

Был проведён набор тестов на обучающих множествах с различной степенью «зашумленности», позволяющих получить предварительный ответ на поставленный вопрос. Функция, используемая для генерации исходных экспериментальных данных:

$$f(x_1, x_2) = 7 \cdot x_1^{1,5} - \sin(x_2) / x_1 \cdot 100 + x_2^2, x_1 \in [1; 4], x_2 \in [-8; 8].$$

Объём генерируемых данных: 102 значения в узлах сетки с равноудалёнными друг от друга узлами, включая пограничные значения [1; –8], [1; 8], [4; –8], [4; 8].

Задачей генетического программирования является идентификация именно этой исходной функции.

Результаты тестов на исходном, не зашумленном наборе данных, представлены в табл. 1.

Фитнесс-функция вычисляется по формуле $fitness = 1/(1+d)$, где d – средняя ошибка полученного решения при тестировании на исходных данных. Лучшие полученные решения характеризуются большим значением фитнесс-функции. Данный способ определения значений фитнесс-функции является общепринятым для генетического программирования и позволяет наглядно отображать качество полученного решения: чем ближе значение фитнесс-функции к единице, тем выше качество; чем ближе значение фитнесс-функции к нулю, тем качество ниже [9].

Таблица 1

Результаты тестирования модифицированного и оригинального метода генетического программирования на исходном наборе данных

№ теста	Обучающее множество для генетического программирования	Лучшее значение фитнесс-функции генетического программирования	Степень расширения обучающего множества для генетического программирования
1	Исходный набор	0,202	–
	Расширенный набор после искусственной нейросети	0,206	2,5*x*
2	Исходный набор	0,539	–
	Расширенный набор после искусственной нейросети	0,201	4,25*x*
3	Исходный набор	0,333	–
	Расширенный набор после искусственной нейросети	0,19	4,25*x*

Как можно видеть из табл. 1, при работе на исходных данных без зашумления модифицированный метод генетического программирования работает сравнительно также или хуже оригинального метода генетического программирования – до 63 % хуже (пояснение – если рассмотреть тест № 2, то $(|0,539-0,201|/0,539)\cdot 100\ \% \approx 62,7\ \%$).

Результаты тестов на зашумленном наборе данных представлены в табл. 2.

Таблица 2

Результаты тестирования модифицированного и оригинального метода генетического программирования на зашумленном наборе данных

№ теста	Обучающее множество для генетического программирования	Шум, %	Лучшее значение фитнесс-функции генетического программирования	Степень расширения обучающего множества для генетического программирования
1	Исходный набор	4	0,147	–
	Расширенный набор после искусственной нейросети	4	0,113	2,5 *x*
2	Исходный набор	8	0,094	–
	Расширенный набор после искусственной нейросети	8	0,095	4,25 *x*
3	Исходный набор	15	0,052	–
	Расширенный набор после искусственной нейросети	15	0,085	4,25 *x*

При работе на зашумленном наборе исходных данных модифицированный метод генетического программирования работает сравнительно также или чуть лучше оригинального метода генетического программирования: как можно видеть из табл. 2, в случае наличия 4 % шума в исходных данных модифицированный метод сработал хуже оригинального на 23 % (пояснение: $(|0,147-0,113|/0,147)\cdot 100\ \% \approx 23,1\ \%$); в случае шума 8 % – сработал также; в случае шума 15 % – сработал на 63 % лучше (пояснение: $(|0,052-0,085|/0,052)\cdot 100\ \% \approx 63,5\ \%$). При этом заметно, что качество работы модифицированного метода по сравнению с оригинальным возрастает при увеличении степени зашумленности исходного набора данных.

Заключение

Полученные результаты говорят в пользу того, что при идентификации предлагаемой аналитической зависимости модифицированный метод генетического программирования не даёт устойчиво лучших результатов.

Для получения однозначного вывода о целесообразности применения предлагаемого модифицированного метода генетического программирования, в дальнейшем требуется провести дополнительный набор тестов для изучения влияния степени расширения обучающего набора на качество идентификации.

Необходимо также изучение влияния степени сложности исходной функции на качество идентификации оригинальным и модифицированным методом генетического программирования.

Дополнительной задачей является разработка математического объяснения, в каких случаях (и имеются ли такие случаи) модифицированный метод генетического программирования будет работать устойчиво качественнее оригинального метода генетического программирования.

СПИСОК ЛИТЕРАТУРЫ

1. *Матвеев М. Г.* Модели и методы искусственного интеллекта. Применение в экономике / М. Г. Матвеев, А. С. Свиридов, Н. А. Алейникова. – М.: Финансы и статистика; ИНФРА-М, 2008. – 448 с.

2. *Френкель М. Б.* Методика оценки результатов технического анализа на рынке ценных бумаг в условиях нечеткости и неуверенности / М. Б. Френкель, И. Ю. Квятковская // Вестн. Астрахан. гос. техн. ун-та. – 2006. – № 6 (35). – С. 258–265.

3. *Полумордвинова А. О.* Информационная система поиска оптимального управленческого решения / А. О. Полумордвинова, И. Ю. Квятковская // Вестн. Астрахан. гос. техн. ун-та. Сер.: Морская техника и технология. – 2009. – № 2. – С. 61–64.

4. *Умеров А. Н.* Методы и программные средства аппроксимации экспериментальных данных / А. Н. Умеров, В. Ф. Шуршев // Вестн. Астрахан. гос. техн. ун-та. – 2005. – № 1 (24). – С. 97–104.

5. *Шуршев В. Ф.* Исследование алгоритма комплексного эволюционного метода, применяемого в компьютерной системе поддержки принятия решения о выборе состава холодильных агентов, с помощью вычислительных экспериментов / В. Ф. Шуршев, Н. В. Демич // Вестн. Астрахан. гос. техн. ун-та. – 2006. – № 1 (30). – С. 141–146.

6. *Норман Р. Дрейпер.* Прикладной регрессионный анализ / Норман Р. Дрейпер, Гарри Смит. – М.: Вильямс: Диалектика, 2007. – 912 с.

7. *Ивахненко А. Г.* Моделирование сложных систем по экспериментальным данным / А. Г. Ивахненко, Ю. П. Юрачковский. – М.: Радио и связь, 1987. – 119 с.

8. *Koza John R.* Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems) / Koza John R. – A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 1992. – 813 с.

9. *Николенко С. И.* Самообучающиеся системы / С. И. Николенко, А. Л. Тулупьев. – М.: МЦНМО, 2009. – 288 с.

10. *Сегаран Т.* Программируем коллективный разум / Т. Сегаран. – СПб.: Символ-Плюс, 2008. – 368 с.

REFERENCES

1. Matveev M. G., Sviridov A. S., Aleinikova N. A. *Modeli i metody iskusstvennogo intellekta. Primenenie v ekonomike* [Models and methods of artificial intellect. Application in economics]. Moscow, Finansy i statistika; INFRA-M Publ., 2008. 448 p.

2. Frenkel' M. B., Kviatkovskaia I. Iu. Metodika otsenki rezul'tatov tekhnicheskogo analiza na rynke tsennykh bumag v usloviiakh nechetkosti i neuverennosti [Methods of assessment of the results of technical analysis on the equity market in conditions of fuzziness and uncertainty]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta*, 2006, no. 6 (35), pp. 258–265.

3. Polumordvinova A. O., Kviatkovskaia I. Iu. Informatsionnaia sistema poiska optimal'nogo upravlencheskogo resheniia [Information system of searching optimal managerial decision]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriia: Morskaia tekhnika i tekhnologiia*, 2009, no. 2, pp. 61–64.

4. Umerov A. N., Shurshev V. F. Metody i programmnye sredstva approksimatsii eksperimental'nykh dannykh [Methods and program means of approximation of experimental data]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta*, 2005, no. 1 (24), pp. 97–104.

5. Shurshev V. F., Demich N. V. Issledovanie algoritma kompleksnogo evoliutsionnogo metoda, primeniaemogo v komp'iuternoi sisteme podderzhki priniatiia resheniia o vybore sostava kholodil'nykh agentov, s pomoshch'iu vychislitel'nykh eksperimentov [Study of the algorithm of complex evolutionary method used in computer system of decision making support on selection of refrigerant agents composition with the help of computational experiments]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta*, 2006, no. 1 (30), pp. 141–146.

6. Norman R. Dreiper, Garri Smit. *Prikladnoi regressionnyi analiz* [Applied regression analysis]. Moscow, Vil'iams, Dialektika Publ., 2007. 912 p.

7. Ivakhnenko A. G., Iurachkovskii Iu. P. *Modelirovanie slozhnykh sistem po eksperimental'nym dannym* [Modeling of complex systems on experimental data]. Moscow, Radio i sviaz' Publ., 1987. 119 p.

8. Koza John R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).* A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 1992. 813 p.

9. Nikolenko S. I., Tulup'ev A. L. *Samoobuchaiushchiesia sistemy* [Self-training systems]. Moscow, MTsNMO, 2009. 288 p.

10. Segaran T. *Programmiruem kollektivnyi razum* [Programme collective mind]. Saint Petersburg, Simvol-Plius, 2008. 368 p.

Статья поступила в редакцию 3.03.2013,
в окончательном варианте – 26.03.2013

ИНФОРМАЦИЯ ОБ АВТОРАХ

Звезинцев Андрей Игоревич – Астраханский государственный технический университет; аспирант кафедры «Прикладная информатика в экономике»; AndZvezintsev@gmail.com.

Zvezintsev Andrey Igorevich – Astrakhan State Technical University; Postgraduate Student of the Department "Applied Informatics in Economics"; AndZvezintsev@gmail.com.

Квятковская Ирина Юрьевна – Астраханский государственный технический университет; д-р техн. наук, профессор; директор института информационных технологий и коммуникаций; i.kvyatkovskaya@astu.org.

Kvyatkovskaya Irina Yurievna – Astrakhan State Technical University; Doctor of Technical Sciences, Professor; Director of the Institute of Information Technologies and Communications; i.kvyatkovskaya@astu.org.