

А.А. Орлов

## Принципы построения архитектуры программной платформы для реализации алгоритмов метода группового учета аргументов

Рассмотрены принципы построения архитектуры программной платформы для реализации алгоритмов МГУА. Приведен аналитический обзор программных продуктов для индуктивного моделирования. Предложена объектно-ориентированная структура платформы, позволяющая реализовать алгоритмы МГУА, базисы, методы обучения и критерии селекции моделей.

Principles of construction of the software framework architecture for the implementation of the algorithms of a group method of data handling are considered. The analytic survey of the inductive modeling software is given. An object-oriented platform structure allowing to implement the algorithms, model bases, training methods and selection criteria of the models is suggested.

Розглянуто принципи побудови архітектури програмної платформи для реалізації алгоритмів методу МГУА. Подано аналітичний огляд програмних продуктів для індуктивного моделювання. Запропоновано об'єктно-орієнтовану структуру платформи, що дозволяє реалізувати алгоритми МГУА, базиси, методи навчання та критерії селекції моделей.

**Введение.** Известно множество успешных применений метода группового учета аргументов (МГУА) в самых различных областях: распознавание образов, нахождение физических и нефизических закономерностей, идентификация нелинейных систем, краткосрочное и долгосрочное прогнозирование стационарных и нестационарных процессов, управление сложными техническими объектами и др. [1–3]. При этом каждый из проектов в каждой из указанных областей выдвигает собственные требования к конкретной практической реализации технологии МГУА. Поэтому исследователям необходимо иметь разнообразие доступных алгоритмов метода, базисов, методов обучения и критериев селекции моделей [3–5] для выработки наиболее эффективного их сочетания в рамках решаемой задачи. Это обстоятельство, а также тот факт, что список алгоритмов метода постоянно расширяется, обуславливает необходимость реализации гибкой и универсальной программной платформы.

В работе [6] предложено реализовывать подобную программную платформу в виде пакета прикладных программ. Были сформулированы три группы критериев: обязательные, целесообразные, желаемые – исчерпывающим образом описывающие характеристики и свойства программной платформы с точки зрения ее

пользователя. Однако в указанной работе не раскрыт принцип построения программной платформы, позволяющий достичь выполнения этих критериев.

В данной статье автор предлагает рассмотреть принципы построения такой архитектуры (внутренней структуры) платформы, которая бы гарантировала удовлетворение всех указанных в работе [9] критериев, актуальных и сегодня.

### Требования к архитектуре программной платформы

Сформулируем требования к данной архитектуре:

- *гибкость* характеризует такую организацию архитектуры программного продукта, которая поддерживает добавление новых алгоритмов, базисов функций, критериев и других функций на различных уровнях программной платформы: как путем расширения исходных кодов программы (статически), так и путем подключения дополнительных модулей в процессе выполнения (динамически); при этом добавление новых элементов не должно приводить к необходимости существенного и/или трудоемкого изменения архитектуры программной платформы;

- *универсальность* предполагает возможность использования программной платформы в качестве как независимой системы индуктивного моделирования (диалоговой, интерак-

тивной и т.д.), так и элемента другой системы (в том числе и в составе технической системы), с использованием унифицированных интерфейсов доступа к *внутренним возможностям* программного продукта;

- *производительность* определяет время получения результата при проведении индуктивного моделирования. Это свойство особенно критично для программных продуктов, реализующих алгоритмы МГУА, по своей природе являющихся переборными. При этом такие средства достижения высокой производительности, как оптимизация исходных кодов (среда выполнения, оптимизации под процессорные архитектуры и т.д.) и распараллеливание вычислений, *на уровне архитектуры* реализуются наиболее эффективно.

Отметим, что обозначенные критерии гибкости и универсальности определяют как уровень удобства использования программной платформы (на уровне разработчика и пользователя), так и объем трудозатрат, необходимых для решения конкретных задач (от локальной настройки до существенной доработки).

### Обзор аналогов

Существует программное обеспечение, реализующее алгоритмы метода МГУА. Результаты анализа и оценки известных программных продуктов представлены в табл. 1. Для каждого из них приведена информация о поддерживаемых алгоритмах, базисах, методах обучения и критериях. Цветом в таблице отмечен оцениваемый уровень трудозатрат, необходимых для расширения возможностей платформы до требуемого уровня (серым – средняя трудоемкость, черным – высокая трудоемкость), причем в качестве минимальных требований выбрана поддержка комбинаторного алгоритма и полиномиальной нейронной сети, полиномиального базиса, метода наименьших квадратов (МНК) в качестве метода обучения и критериев точности и несмещенности для селекции моделей [4].

Также в таблице приведена оценка соответствия каждого продукта критериям гибкости, универсальности и производительности. Зна-

ком «+» отмечено соответствие критерию, знак «-» определяет несоответствие критерию, устранение которого либо невозможно, либо связано с существенными трудозатратами. Знаком «?» отмечены ячейки, информация по которым недостаточна.

Исходя из анализа, делаем следующие выводы.

- Наиболее популярным алгоритмом, реализуемым в системах моделирования, есть *PNN* (восемь из девяти продуктов), причем все системы моделирования обладают возможностью генерации полиномиальных моделей.

- Наиболее часто оптимизация параметров сгенерированных моделей производится по МНК, однако существуют реализации нелинейных процедур оптимизации (*FAKE GAME*, *GEvoM*).

- Классические критерии селекции моделей реализованы в большинстве программных продуктов. Существует возможность выбора используемого при моделировании критерия.

- Все продукты обладают интерактивным пользовательским интерфейсом, что позволяет использовать их в качестве системы моделирования. Программные продукты «Пакет прикладных программ метода группового учета аргументов» (ППП МГУА), *FAKE GAME* и *Knowledge Miner* реализуют программные интерфейсы, позволяющие включать их в состав технических систем, поэтому они удовлетворяют критерию универсальности.

- Лучшим по гибкости есть программный продукт *FAKE GAME*, расширение возможностей которого осуществимо путем изменения исходных кодов. *GMDH Shell* поддерживает динамическое подключение исполняемых модулей, частично удовлетворяя критерию гибкости. ППП МГУА реализован на языке ФОРТРАН-77 для среды *MS DOS*, поэтому расширение на уровне исходных кодов и подключение динамических модулей крайне трудоемко, что не удовлетворяет критерию гибкости.

- *Knowledge Miner* и *GMDH Shell* поддерживают распараллеливание вычислений, однако являются программным обеспечением с закрытым

Таблица 1. Системы моделирования на основе МГУА

Система моделирования	МГУА				Гибкость		Универсальность		Производительность	
	Алгоритмы	Базисы	Методы обучения	Критерии	На уровне исходных кодов	Подключение динамических модулей	Как система моделирования	Как элемент технической системы	Оптимизация исходных кодов	Параллельные вычисления
АСТРИД [7]	<i>COMBI MULTI PNN</i>	ПБ и др.	Много	Много	?	?	+	?	+	–
<i>FAKE GAME</i> [8]	<i>PNN</i> + усоверш. ГА	ПБ ГБ ЛБ ЗБ РБ	Много	Много	+	Возможно	+	+	–	?
<i>Knowledge Miner</i> [9]	<i>2MLAN AC Self-Organizing Fuzzy Rule Induction</i>	ПБ	?	<i>PSS MAPE</i>	–	–	+	Возможно ( <i>MS Excel XLL, DLL</i> )	+	+
<i>PCombi</i> [10, 11]	<i>COMBI</i>	ПБ	МНК	?	Затруднительно	–	+	Затруднительно	–	+
ППП МГУА [6, 12]	<i>COMBI MIA</i>	ПБ	Метод Брандона. Метод Гаусса–Ньютона	<i>CR, BS, баланса</i>	Затруднительно	Затруднительно	+	+	+	–
<i>Virtual Comput. Chemistry Lab. PNN</i> [13]	<i>PNN</i>	ПБ	МНК	<i>CR</i>	–	–	+	–	?	?
<i>GEvoM</i> [14]	<i>PNN (ISP, PSD, ED, Evolutionary Method)</i>	ПБ	<i>SVD SNE</i>	?	–	–	+	–	?	?
<i>GMDH Shell</i> [15, 16]	<i>COMBI MIA</i>	ПБ	?	<i>RMSE MAE Hit%</i>	–	+	+	–	+	+
<i>PNN Discovery Client</i> [17]	<i>PNN</i>	ПБ	МНК (Гаусс)	Стандарт	–	–	+	–	?	?

*COMBI* – комбинаторный алгоритм, *MULTI* – комбинаторно-селекционный алгоритм, *PNN* – полиномиальная нейронная сеть МГУА, *MIA* – многорядный алгоритм МГУА, *2MLAN* – дважды многорядная сеть с активными нейронами, *AC* – алгоритм комплексования аналогов. ПБ – полиномиальный базис, ГБ – гармонический базис, ЛБ – логистический базис, ЭБ – экспоненциальный базис, РБ – рациональный базис.

исходным кодом, что противоречит критерию гибкости. А продукт *FAKE GAME* реализован на платформе *Java* и не поддерживает параллельные вычисления, поэтому не удовлетворяет критерию производительности.

Таким образом, несмотря на широкий спектр существующего программного обеспечения, оно не удовлетворяет в полной мере всем обозначенным выше требованиям к архитектуре программной платформы (это связано с тем,

что перед авторами данных программных продуктов не ставилась задача следования всем этим критериям). Исходя из этого, принято решение о разработке собственной программной платформы.

### Разработка архитектуры программной платформы

Для реализации программной платформы выбрана объектно-ориентированная парадигма анализа, проектирования и программирования.

Во-первых, применение объектно-ориентированного анализа особенно эффективно благодаря тому, что концепция метода группового учета аргументов оперирует набором понятий с ясными связями между собой. Во-вторых, объектно-ориентированные программные системы более удобны для использования в качестве базовых с учетом их гибкости, расширяемости и переносимости. В-третьих, объектно-ориентированный процесс разработки рекомендуется при реализации больших сложных систем [18].

Согласно классике объектно-ориентированного стиля программирования Гради Буч [18], разработка объектных систем начинается с объектно-ориентированного анализа. Цель анализа – выявление сущностей предметной области, которыми могут быть классы либо объекты. Объекты при этом обладают собственным состоянием, поведением и идентичностью, а классы описывают структуру и поведение схожих объектов.

Предметной областью программной платформы есть МГУА. Словарь данной предметной области устоявшийся и включает в себя такие понятия, как модель, базис, критерий, алгоритм, обучающая и проверочная выборки. Семантические связи между данными сущностями представлены на рис. 1.

Центральное место в системе занимает понятие модели как сущности, отражающей некую взаимосвязь или зависимость в данных. При этом модель принадлежит некоторому базису. Базис модели определяет ее структуру. Структура и параметры модели определяют ее уникальность среди других моделей.

Алгоритм МГУА – это сущность, обозначающая некоторую последовательность действий по выявлению модели оптимальной сложности. Для этого она, с одной стороны, организует структурную и параметрическую оптимизацию моделей, выделенную в виде абстракции «обучение». С другой стороны, отбор моделей проводится с помощью одного или нескольких критериев. Сущности критериев способны вычислять некое численное значение, характеризующее степень оптимальности мо-

дели (связь модель–критерий). Одна из известных особенностей МГУА – разделение выборки на обучающую и проверочную части – также нашла свое отражение на диаграмме.

Следующим за анализом шагом объектно-ориентированного процесса разработки есть объектно-ориентированное проектирование, в ходе которого выявленные связи между сущностями уточняются, а система представляется в виде совокупностей классов и объектов. Результат данного этапа – диаграммы классов и объектов, отражающие как семантику выбранной предметной области, так и архитектуру (структуру) и свойства проектируемой программной системы.

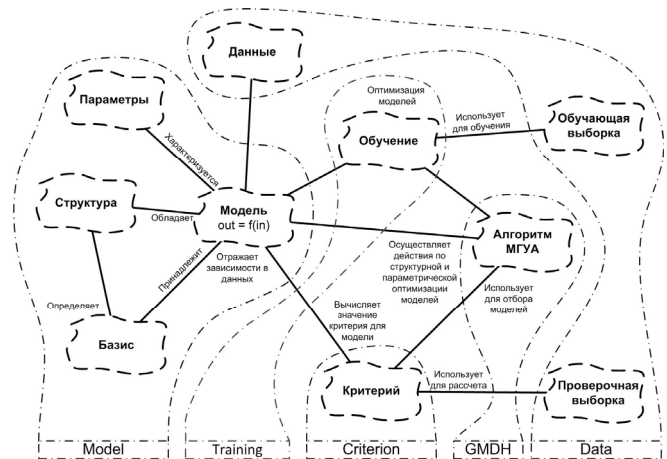


Рис. 1. Сущности предметной области и семантические связи

Сформулированный критерий гибкости требует, чтобы реализация *любого* алгоритма, базиса моделей и прочих сущностей МГУА была возможна на базе единой архитектуры. В рамках МГУА реализованы более 15-ти различных алгоритмов – от классических [4, 5] до гибридных, в которых используются сочетания МГУА с другими техниками, такими как генетические алгоритмы *Particle Swarm Optimization (PSO)*, вейвлеты и др. Результаты анализа известных алгоритмов относительно сущностей, выявленных на этапе объектно-ориентированного анализа, представлены в табл. 2–4. Таблицы позволяют оценить требуемый уровень гибкости архитектуры программной платформы относительно сущностей предметной области, необходимый для реализации *каждого* из алгоритмов МГУА.

Таблица 2. Параметрические алгоритмы МГУА

Название алгоритма (в терминологии авторов)	Особенности реализации сущностей предметной области				
	Алгоритм	Данные	Модель	Обучение	Критерии
Комбинаторный (COMBI) [4]	Последовательный полный перебор моделей и выбор модели оптимальной сложности	Функция многих аргументов	ПБ. Возрастание сложности модели есть увеличение числа слагаемых и степени полинома	Метод наименьших квадратов (МНК)	Возможность последовательного применения набора критериев для отбора оптимальных моделей
Комбинаторно-селекционный (MULTI) [19]			Полигармонический базис. Возрастание сложности модели есть увеличение числа составляющих гармоник		
Гармонический (HARM) [4]			Критерии регулярности (точности) CR (MSE). Критерий несмещенности BS и др.		
Многорядный итерационный (MIA) [4]	Итеративное построение сетевой структуры оптимальной сложности	Функция многих аргументов	ПБ. Структура полинома фиксирована (полином 2-й степени от 2-х аргументов)	Нелинейная регрессия; округление и/или дискретизация параметров	Критерии регулярности (точности) CR (MSE). Критерий несмещенности BS и др.
Расширенный многорядный итерационный (eMIA) [20]			Совершенство базисов: ПБ, гармонический, логарифмический, экспоненциальный и т.д.		
Модифицированная полиномиальная НС (MPNN) [21]			ПБ (полиномы 2, 3, 4 аргументов 1 и 2 степени фиксированной структуры)	МНК	Критерии регулярности (точности) CR (MSE)
Робастная полиномиальная НС (RPNN) [22]			ПБ, фиксированная структура полиномов	Метод робастных M-оценок	AR – робастный критерий точности CR
Нечеткая НС МГУА (FuzzyGMDH) [23, 24]			ПБ, треугольная, гауссова и колоколообразная нечеткая функции принадлежности	Линейное программирование (минимизация интервала функции принадлежности)	Критерии регулярности (точности) CR (MSE). Критерий несмещенности RB
Основанная на правилах нечеткая НС (RFPNN) [25]			ПБ (полиномы 2, 3, 4 аргументов 1, 2 степени фиксированной структуры), набор функций принадлежности	МНК (полиномиальная НС). ГА (нечеткая НС)	Критерии регулярности (точности) CR (MSE). Критерий абсолютной ошибки MAE
Полиномиальная НС с ОС (FB PNN) [26]			ПБ, экспоненциальный базис, фиксированные структуры моделей	Пошаговый регрессионный анализ	Критерий Акаике (Akaike's Information Criterion AIC)
Нечеткая НС МГУА с ОС (NF-GMDH-IFL) [27]			ПБ, гауссова нечеткая функция принадлежности	Гибридный проекционный метод (Hybrid Projection Method, HPM)	Критерии регулярности (точности) CR (MSE)
С применением ГА (GA-GMDH) [28]			ПБ (полиномы двух аргументов 1, 2 и 3 степени фиксированной структуры)	МНК	
Многокритериальная Парето-оптимальная полиномиальная НС МГУА с применением ГА (MO-GA-GMDH) [29]			ПБ (полиномы фиксированной структуры 2-й степени от 2-х аргументов)	МНК Singular Value Decomposition (SVD)	Критерий регулярности (точности) CR (MSE). Критерий Акаике (Akaike's Information Criterion AIC). Критерий оптимальности по Парето и др.
С применением метода роя частиц (PSO-MIA) [30]	ПБ (структура полинома оптимизируется с помощью PSO)	Метод роя частиц (particle swarm optimization, PSO)	Критерий точности CR и др.		

НС – нейронная сеть, ПНС – полиномиальная нейронная сеть, МНК – метод наименьших квадратов, ОС – обратная связь, ГА – генетический алгоритм, ПБ – полиномиальный базис моделей.

Рассмотрим каждую сущность в отдельности.

• **Данные.** Алгоритмы предназначены для моделирования как функций многих аргументов, так и многомерных временных рядов. Более того, модели различных базисов предполагают специфичные методы обработки данных. Эти факторы требуют инкапсуляции процесса обработки выборки данных и (опционально) оптимизированного ее хранения.

• **Модель.** Применяются как модели с фиксированной структурой, так и различного вида переборные способы генерации структур моделей, что обуславливает необходимость инкапсуляции процесса генерации некоторой последовательности структур моделей по заданным параметрам.

• **Обучение.** Используется как независимое, так и совместное обучение совокупностей па-

Таблица 3. Непараметрические алгоритмы МГУА

Название алгоритма (в терминологии авторов)	Особенности реализации сущностей предметной области				
	Алгоритм	Данные	Модель	Обучение	Критерии
Объективной компьютерной кластеризации (OCC) [5]	Итеративный поиск оптимального числа кластеров	Функция многих аргументов	Модель определяет способ разбиения на заданное число кластеров	Кластеризация / классификация выборки на заданное число кластеров	Критерий сходства разбиений на кластеры
Комплексирования аналогов (AC) [31]	Итеративный поиск ближайших аналогов и их оптимального числа	Многомерная временная последовательность	Полиномиальный базис – модель трансформирования аналога, модель комбинирования аналогов	МНК для трансформации аналогов	Критерий сходства аналогов

Таблица 4. Многорядные алгоритмы МГУА

Название алгоритма (в терминологии авторов)	Особенности реализации сущностей предметной области				
	Алгоритм	Данные	Модель	Обучение	Критерии
Дважды многорядная сеть МГУА (2ML) [ 4]	Итеративное построение сетевой структуры оптимальной сложности, нейроны которой реализуют алгоритм(ы) МГУА	Функция многих аргументов. Многомерная временная последовательность	Определяется используемым алгоритмом (или совокупностью алгоритмов)		
Многорядный алгоритм с последовательным перебором/выделением трендов (Trend) [ 4]					
Объективный системный анализ (Objective System Analysis, OSA) [32]					

раметрических моделей, однако распараллеливание вычислений должно осуществляться в обоих случаях.

• **Критерий.** Общее количество применяемых внешних критериев селекции моделей превышает 10.

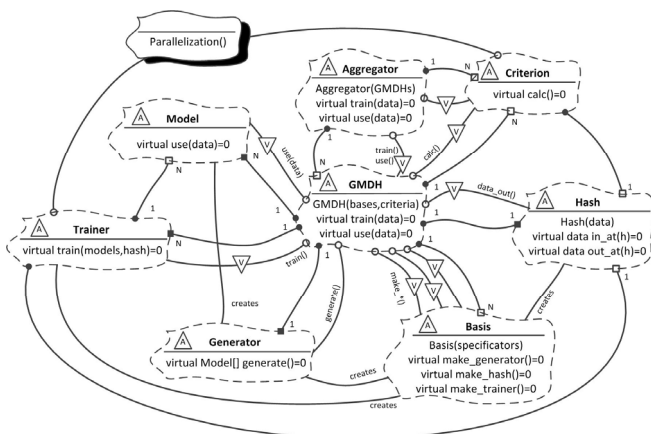


Рис. 2. Базовая диаграмма классов программной платформы

Проведенный обзор позволил уточнить семантику выделенных сущностей МГУА. Результирующая диаграмма классов, определяющая архитектуру программной платформы, представлена на рис. 2. Рассмотрим специфику некоторых выделенных классов.

• **Basis.** Данный класс представляет унифицированный интерфейс для генерации и обучения моделей, а также специфичной обработки выборок данных.

• **Организация процесса индуктивного моделирования** в классе *GMDH* проводится следующим образом. Принимая на вход выборку данных (функция *train* с параметром *data*), объект класса *GMDH* проводит обработку данных (с помощью *Hash*), генерацию (используя класс *Generator*), обучение (класс *Trainer*) и селекцию моделей заданного базиса (класс *Basis*) согласно внешнему критерию (класс *Criterion*) или набору критериев.

• **Aggregator** – класс реализует многорядные алгоритмы МГУА, принимая список используемых алгоритмов МГУА в качестве параметра конструктора.

• **Parallelization** – это утилита, предназначенная для организации параллельных вычислений с целью повышения производительности переборных и итеративных частей алгоритмов МГУА (на диаграмме используется классами *Trainer* и *Criterion*).

На диаграмме видно, что все связи реализуются посредством механизма виртуального вы-

зова функций абстрактных классов–интерфейсов. Конкретные реализации данных классов возможны путем наследования от базовых классов (интерфейсов), что обеспечивает независимость реализации элементов друг от друга и гибкость переконфигурирования в процессе выполнения программы, что позволяет динамически задавать не только выборку данных, но и набор используемых критериев, параметры и тип базиса функций, а также другие параметры, влияющие на результат моделирования.

**Заключение.** Предложенная в статье архитектура программной платформы удовлетворяет всем сформулированным требованиям. Она обладает достаточной гибкостью для реализации приведенных алгоритмов, оставаясь при этом структурно неизменной в каждой конкретной реализации. Универсальность достигается путем экспортирования (предоставления доступа) к интерфейсам базовых классов как на уровне исходных кодов, так и динамически в процессе выполнения. Следование требованию производительности также нашло свое отражение на уровне архитектуры системы в виде явного указания распараллеливаемых частей архитектуры – процессов обучения и селекции моделей в каждом алгоритме МГУА.

1. *Ivakhnenko A.G., Ivakhnenko G.A.* The Review of Problems Solvable by Algorithms of the Group Method of Data Handling // Int. J. of Patt. Recog. and Image Analysis: Advanced in Mathem. Theory and Appl. – 1995. – 5, N 4. – P. 527–535.
2. *GMDH* – Examples of Applications. URL: [http://www.gmdh.net/GMDH\\_exa.htm](http://www.gmdh.net/GMDH_exa.htm) (дата обращения 12.12.2012).
3. *Anastasakis L., Mort N.* The development of Self-Organization techniques in modelling: A review of the Group Method of Data Handling (GMDH). // Res. Rep. N 813, The University of Sheffield, United Kingdom, 2001. – 38 p.
4. *Ивахненко А.Г.* Индуктивный метод самоорганизации моделей сложных систем. – Киев: Наук. думка, 1982 \. – 296 с.
5. *Madala H.R., Ivakhnenko A.G.* Inductive Learning Algorithms for Complex System Modeling // Publ. CRC Press, 1994. – 368 p.
6. *Степануко В.С.* Основные требования к функциональной структуре ППП МГУА для персональных ЭВМ / Управление в технических системах. – Киев: ИК АН УССР, 1990. – С. 27–34.

7. *ASTRID* – программная система. – URL: <http://www.mgua.irtc.org.ua/ru/index.php?page=astrid>
8. *Kordik P.* Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution // PhD thesis. – Prague, 2006. – 136 p.
9. *Muller J.A., Lemke F.* Self-Organising Data Mining. An Intelligent Approach to Extract Knowledge from Data, Dresden, Berlin, 1999. – 225 p.
10. *Koshulko O.A., Koshulko A.I.* Adaptive parallel implementation of the Combinatorial GMDH algorithm // Proc. of Int. Workshop on Inductive Modelling (IWIM), Prague, 2007. – P. 71–77.
11. *Parallel-combi* – openGMDH. – URL: <http://opengmdh.org/wiki/parallel-combi>
12. *Степануко В.С., Семенов Н.А., Михеев В.Н.* Диалоговый пакет прикладных программ моделирования на основе МГУА (ППП МГУА) / Искусственный интеллект – основа новой информационной технологии. – Калинин: НПО ЦПС, 1990. – С. 105–116.
13. *Polynomial Neural Network for Linear and Non-linear Model Selection in Quantitative-Structure Activity Relationship Studies on the Internet / I.V. Tetko, T.I. Akse-nova, V.V. Volkovich et al.* // SAR and QSAR in Environ. Res., 2000. – Issue 3–4, 11. – P. 263–280.
14. *Genetic Design of GMDH-type Neural Networks for Modelling of Thermodynamically Pareto Optimized Turbojet Engines / K. Atashkari, N. Nariman-Zadeh, A. Darvizeh et al.* // WSEAS Transactions on COMPUTERS, Issue 3, 3, July 2004.
15. *Koshulko O.A., Koshulko A.I.* Acceleration of GMDH combinatorial search with HPC clusters // Proc. of 2nd Int. Conf. on Inductive Modelling, Sept. 15–19, Kyiv, Ukraine, 2008. – P. 164–167.
16. *GMDH Shell* – Forecasting Software for Professionals. – URL: <http://www.gmdhshell.com>
17. *PNN Software Company* – Polynomial Neural Network Solutions. – URL: <http://pnn.pnnsoft.com>
18. *Object-Oriented Analysis and Design with Applications / G. Booch, R.A. Maksimchuk, M.W. Engel et al.* // Addison-Wesley Professional, 3/E, 2007. – 720 p.
19. *Степануко В.С., Костенко Ю.В.* Исследование свойств комбинаторно–селекционного (многоэтапного) алгоритма МГУА // Сб. научн. тр. МНУЦИТ и С НАНУ. Моделирование и управление состоянием эколого–экономических систем региона. – Киев, 2001. – С. 96–100.
20. *Taušer J., Buryan P.* Exchange Rate Predictions in International Financial Management by Enhanced GMDH Algorithm // Prague Economic Papers N 3, University of Economics. – 2011. – P. 232–249.
21. *Devilopoulos E., Theocharis J.B.* A modified PNN algorithm with optimal PD modeling using the orthogonal least squares method // Int. J. Information Sciences. – 2004. – N 168. – P. 133–170.

Окончание на с. 88

22. *Aksenova T., Volkovich V., Villa A.* Robust Structural Modeling and Outlier Detection with GMDH-Type Polynomial Neural Networks // Proc. of 15th Int. Conf. Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Warsaw, Lecture Notes in Comp. Sci. – **3697**, Springer, 2005. – P. 881–886.
23. *Исследование* разных видов функций принадлежности параметров нечетких прогнозирующих моделей в нечетком методе группового учета аргументов / Ю.П. Зайченко, И.О. Заец, О.В. Камоцкий и др. // УСИМ. – 2003. – № 2. – С. 56–67.
24. *Zaychenko Yu.* The Investigations of Fuzzy Group Method of Data Handling with Fuzzy Inputs in the Problem of Forecasting in Financial Sphere // Proc. of 2nd Int. Conf. on Inductive Modelling, Kiev, 2008. – P. 129–133.
25. *Park B.-J., Lee D.-Y., Oh S.-K.* Rule-Based Fuzzy Polynomial Neural Networks in Modeling Software Process Data // Int. J. of Control, Automation and Systems. – 2003. – **1**, N 3.– P. 321–331.
26. *Kondo T., Pandya A.* GMDH-type Neural Networks with a Feedback Loop and their Application to the Identification of Large-spatial Air Pollution Patterns // Proc. of the 39th SICE Annual Conf. Int. Session Papers, 112A-4, Iizuka, Japan, 2000. – P. 1–6.
27. *Zhao X., Song Z., Li P.* A Novel NF-GMDH-IFL and Its Application to Identification and Prediction of Nonlinear Systems // Proc. of IEEE Region 10 Conf. on Comp., Communications, Control and Power Engineering (TENCON), Beijing, China, 2002. – **3**. – P. 1286–1289.
28. *Farzi S.* A New Approach to Polynomial Neural Networks based on Genetic Algorithm // In. J. of Comp. Syst. Sci. and Engin. – 2008. – P. 180–187.
29. *Nariman-Zadeh N., Jamali A.* Pareto Genetic Design of GMDH-type Neural Networks for Nonlinear Systems // Proc. of Int. Workshop on Inductive Modeling 2007, Prague, Czech Republic, 2007. – P. 96–103.
30. *Hybrid Particle Swarm Optimization and Group Method of Data Handling for Inductive Modeling / G. Onwubolu, A. Sharma, A. Dayal et.al.* // Proc. of the II Int. Conf. on Inductive Modelling ICIM-2008, 15–19 Sept. 2008, Kiev, Ukraine. – Kiev: IRTC ITS NANU, 2008. – P. 96–103.
31. *Ivakhnenko A.G.* An inductive sorting method for the forecasting of multidimensional random processes and events, with the help of analogue forecast complexing // Pattern recognition and Image Analysis, 1991. – **1**, N 1. – P. 99–107.
32. *Ивахненко А.Г., Мюллер Й.А.* Самоорганизация прогнозирующих моделей // Киев: Техніка, 1985; Берлин: ФЭБ Ферлаг Техник, 1984. – 223 с.

Тел. для справок: +7 952 893-4743 (Томск)  
 E-mail: d1scnc@gmail.com  
 © А.А. Орлов, 2013