

DOI: 10.17323/2587-814X.2023.2.7.19

Моделирование и оптимизация стратегий принятия индивидуальных решений в многоагентных социально-экономических системах с использованием машинного обучения

А.С. Акопов 

E-mail: akopovas@umail.ru

Центральный экономико-математический институт, Российская академия наук
Адрес: Россия, 117418, г. Москва, Нахимовский проспект, д. 47

Аннотация

В статье представлен новый подход к моделированию и оптимизации стратегий принятия индивидуальных решений в многоагентных социально-экономических системах (МСЭС). В основе такого подхода – синтез методов агентного моделирования, машинного обучения и генетических оптимизационных алгоритмов. Разработана процедура синтеза и обучения искусственных нейронных сетей (ИНС), имитирующих функциональность МСЭС и обеспечивающих аппроксимацию значений ее целевых характеристик. Особенностью данной двухшаговой процедуры является комбинированное использование методов роевой оптимизации (для определения оптимальных значений гиперпараметров) и алгоритма машинного обучения Adam (для вычисления весовых коэффициентов ИНС). Применение подобных, основанных на ИНС суррогатных моделей в параллельных многоагентных генетических алгоритмах вещественного кодирования (МА-RCGA) позволяеткратно повысить временную эффективность процедуры эволюционного поиска оптимальных решений. Проведены численные эксперименты, подтверждающие существенное улучшение производительности МА-RCGA, периодически использующего суррогатную ИНС-модель для аппроксимации значений целевой и фитнес-функции. Спроектирован программный комплекс, который включает оригинальную (эталонную) агентную модель торговых взаимодействий, суррогатную ИНС-модель и генетический алгоритм МА-RCGA. При этом используются программные библиотеки FLAME GPU, OpenNN (Open Neural Networks Library) и др., методы агентного моделирования и машинного обучения.

Разработанная система может быть использована менеджерами, отвечающими, в частности, за формирование оптимальной стратегии торговых взаимодействий.

Ключевые слова: многоагентные социально-экономические системы, метод роя частиц, моделирование случайных продаж, машинное обучение, искусственные нейронные сети, генетические оптимизационные алгоритмы

Цитирование: Акопов А.С. Моделирование и оптимизация стратегий принятия индивидуальных решений в многоагентных социально-экономических системах с использованием машинного обучения // Бизнес-информатика. 2023. Т. 17. № 2. С. 7–19. DOI: 10.17323/2587-814X.2023.2.7.19

Введение

В настоящее время возрастает интерес к исследованию поведения многоагентных социально-экономических систем (МСЭС) и разработки систем поддержки принятия решений (СППР), использующих методы агент-ориентированного моделирования (АОМ), машинного обучения и эвристические (в частности, генетические) оптимизационные алгоритмы.

Большинство современных СППР можно разделить на два укрупненных класса: системы рационального управления на основе имитационного, в том числе, оптимизационного моделирования и экспертные системы поддержки принятия решений.

Примерами СППР первого типа являются: программный комплекс, предназначенный для управления инвестиционной деятельностью крупной нефтяной компании [1], система поддержки принятия решений для эколого-экономического планирования [2], интеллектуальные транспортные системы [3–5] и др.

К наиболее известным СППР второго типа, можно отнести экспертные системы для принятия стратегических решений, использующие метод анализа иерархий [6, 7], системы, предназначенные для приоритизации решений при управлении ИТ-проектами [8], системы, поддерживающие возможность выбора наилучших альтернатив при плохо структурированных исходных данных [9] и др.

В данной работе предлагается система управления МСЭС первого типа, предназначенная, в основном, для формирования оптимальных стратегий принятия индивидуальных решений при множественных торговых взаимодействиях (заключении бартерных и монетарных сделок). Выполнена программная реализация модифицированной модели случайных продаж [10] с использованием методов АОМ [11, 12], ма-

шинного обучения [13, 14], генетических [1, 3, 15] и роевых оптимизационных алгоритмов [16].

Актуальность разработки такой интеллектуальной системы обусловлена, в основном, высокой вычислительной сложностью определения оптимальных моментов времени для заключения бартерных и монетарных сделок в торговых системах со случайными взаимодействиями экономических агентов. В частности, в условиях, когда экономические агенты максимизируют полезность будущего потребления за счёт рационального управления собственными состояниями, разрешающими либо блокирующими парные торговые взаимодействия. Традиционный подход к поиску оптимальных стратегий в подобных многоагентных системах основан на решении задач оптимального управления с использованием классических методов вариационного исчисления и динамического программирования [17]. Однако, из-за высокой размерности моделей подобных МСЭС (т.е. большого числа взаимодействующих агентов) кратно увеличивается вычислительная сложность поиска индивидуальных решений. Поэтому, актуальна разработка программного комплекса, использующего методы машинного обучения и эвристические алгоритмы для приближенного решения задач оптимального управления стратегией торговых взаимодействий в МСЭС.

Цель данной работы состоит в разработке нового подхода к моделированию и формированию стратегий принятия индивидуальных решений в МСЭС с использованием методов машинного обучения, роевых и генетических оптимизационных алгоритмов. Общая методология такого подхода состоит в создании имитационной модели МСЭС, выполнении экспериментов с моделью (класса Монте-Карло) для формирования обучающей выборки, синтезу искусственной нейронной сети

(ИНС) с оптимальной топологией и ее встраиванию в генетический оптимизационный алгоритм для использования в качестве суррогатной модели, существенно ускоряющей процедуру эволюционного поиска решений для всего ансамбля взаимодействующих экономических агентов. При этом, эффективность разработанного подхода и спроектированного программного комплекса исследуется на примере задачи однокритериальной оптимизации характеристик предложенной агентной модели торговых взаимодействий, реализованной использованием системы агентного моделирования FLAME GPU [18] и библиотеки машинного обучения OpenNN [19].

1. Агентная модель торговых взаимодействий

Существенным отличием предложенной агентной модели торговых взаимодействий от ранее известных, является принятие во внимание начального пространственного расположения агентов-продавцов и агентов-покупателей, задаваемого с использованием различных конфигураций, примеры которых представлены на рис. 1.

В модели, в каждый момент времени между каждой произвольной парой агентов, взаимно расположенных в границах зоны торгового взаимодействия (рис. 1), может осуществляется бартерная или монетарная сделка (т.е. обмен товара на товар, либо обмен товара на его денежный эквивалент), если данные агенты, во-первых, находятся в состоянии готовности к подобным сделкам, во-вторых облада-

ют нужным товаром, либо товаром близким к целевому по своим потребительским характеристикам.

Пусть,

$T = \{t_0, t_1, \dots, |T|\}$ – набор моментов времени (по дням), $|T|$ – общее число моментов времени;

$t_0 \in T, t_{|T|} \in T$ – начальный и конечный моменты модели;

$I = \{i_1, i_2, \dots, i_{|I|}\}$ – набор индексов агентов, где $|I|$ – общее число агентов, $\tilde{i} \in I$ – индексы продавцов, $\hat{i} \in I$ – индексы покупателей;

$P = \{p_1, p_2, \dots, p_{|P|}\}$ – набор индексов продуктов, $|P|$ – общее количество продуктов, $p_i(t_k) \in P, i \in I, t_k \in T$ – индекс продукта, имеющегося у i -ого агента, $d_i(t_k) \in P, i \in I, t_k \in T$ – индекс продукта, который необходим i -ому агенту;

$\{b_i(t_k), m_i(t_k)\} \in \{0, 1\}, i \in I$ – состояние готовности агента к заключению бартерных и монетарных сделок, соответственно, в момент t_{k-1} ($t_{k-1} \in T$): 0 – сделки запрещены, 1 – сделки разрешены.

Тогда, расстояние между продуктом, имеющимся у \tilde{i} -го агента продавца ($\tilde{i} \in I$) и продуктом, имеющимся у \hat{i} -ого агента покупателя ($\hat{i} \in I$), измеренного по длине дуги числовой окружности с равномерно распределенными числами 1, 2, ..., $|P|$ в момент t_{k-1} ($t_{k-1} \in T$):

$$\delta_{\tilde{i}\hat{i}}(t_k) = \frac{1}{|P|-1} \min \left\{ |p_{\tilde{i}}(t_k) - d_{\hat{i}}(t_k)|, |P| - |p_{\tilde{i}}(t_k) - d_{\hat{i}}(t_k)| \right\}. \quad (1)$$

При этом, оценка уровня соответствия продукта агента-продавца интересам агента-покупателя может быть задана как:

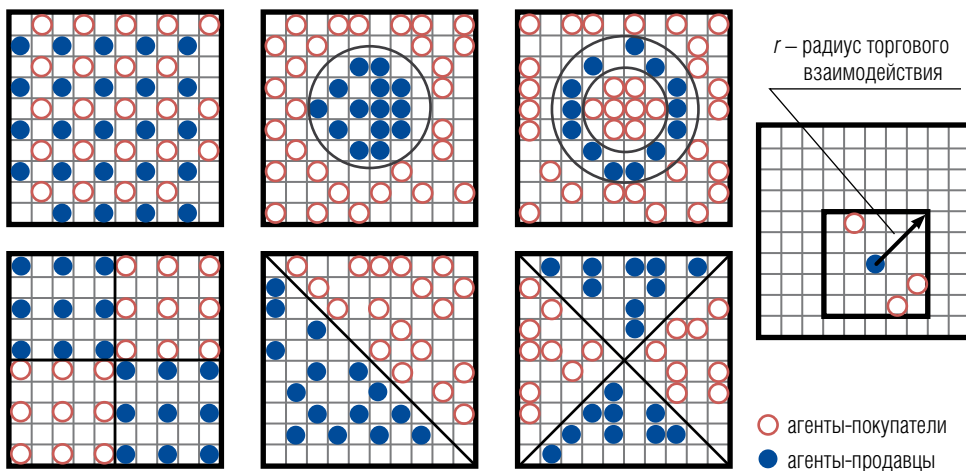


Рис. 1. Возможные конфигурации начального распределения агентов в пространстве МСЭС.

$$\gamma_{ii}(t_k) = \begin{cases} 1, & \text{если } \delta_{ii}(t_k) \leq \varpi, \\ 0, & \text{если } \delta_{ii}(t_k) > \varpi, \end{cases} \quad (2)$$

где $\varpi \geq 0$ — коэффициент порогового соответствия продукта агента-продавца интересам агента-покупателя (коэффициент «контрактности»).

При этом, состояния готовности i -ого агента ($i \in I$) к заключению бартерных и монетарных сделок могут быть сформированы для каждого момента времени t_k ($t_k \in T$), в частности, с помощью логнормальных (*первый способ*) либо бета-распределений (*второй способ*) с заданными характеристиками:

$$b_i(t_k) = \begin{cases} \left\lfloor \frac{\ln N(\mu_b, \sigma_b^2)}{\ln N(\mu_b, \sigma_b^2)} \right\rfloor, & \text{если выполняется I,} \\ 0, & \text{если выполняется II,} \\ \left\lfloor \text{Beta}(\alpha_b, \beta_b) \right\rfloor, & \text{если выполняется III,} \end{cases} \quad (3)$$

$$m_i(t_k) = \begin{cases} \left\lfloor \frac{\ln N(\mu_m, \sigma_m^2)}{\ln N(\mu_m, \sigma_m^2)} \right\rfloor, & \text{если выполняется IV,} \\ 0, & \text{если выполняется V,} \\ \left\lfloor \text{Beta}(\alpha_m, \beta_m) \right\rfloor, & \text{если выполняется VI,} \end{cases} \quad (4)$$

где

- I. при условии использования логнормального распределения для формирования состояний готовности агентов к **бартерным сделкам** и $\ln N(\mu_b, \sigma_b^2) > 0$,
- II. при условии использования логнормального распределения для формирования состояний готовности агентов к **бартерным сделкам** и $\ln N(\mu_b, \sigma_b^2) = 0$,
- III. при условии использования бета-распределения для формирования состояний готовности агентов к **бартерным сделкам**,
- IV. при условии использования логнормального распределения для формирования состояний готовности агентов к **монетарным сделкам** и $\ln N(\mu_m, \sigma_m^2) > 0$,
- V. при условии использования логнормального распределения для формирования состояний готовности агентов к **монетарным сделкам** и $\ln N(\mu_m, \sigma_m^2) = 0$,
- VI. при условии использования бета-распределения для формирования состояний готовности агентов к **монетарным сделкам**.

Здесь

$\ln N(\mu_b, \sigma_b^2)$, $\ln N(\mu_m, \sigma_m^2)$ — случайные величины, имеющие логнормальные распределения с параме-

трами μ_b, σ_b^2 и μ_m, σ_m^2 , где $\mu_b, \mu_m \in [-1, 1]$, $\sigma_b^2, \sigma_m^2 \in (0, 1]$; $\text{Beta}(\alpha_b, \beta_b)$, $\text{Beta}(\alpha_m, \beta_m)$ — случайные величины, имеющие бета-распределения с параметрами α_b, β_b и α_m, β_m , соответственно.

Значение функции полезности i -ого агента ($i \in \{\tilde{i} : \hat{i} \in I, \gamma_{ii}(t_k) = 1\}$) в момент t_k ($t_k \in T$) вычисляется как:

$$u_i(t_k) = \gamma_{ii}(t_k) \left((\delta_{ii}(t_k) + 1)^{-\nu} - \lambda r \right), \quad (5)$$

где $r \in [1, \bar{r}]$ — радиус торгового взаимодействия, т.е. диапазон ячеек дискретного пространства размещения агентов, считающихся соседними, \bar{r} — максимально допустимое расстояние между взаимодействующими агентами;

$\{\nu, \lambda\}$ — коэффициенты, определяющие влияние издержек расстояния между целевым и приобретаем продуктом, а также между покупателем и продавцом, соответственно.

Основными управляющими параметрами такой МСЭС являются: конфигурация начального расположения агентов в пространстве, радиус торгового взаимодействия, коэффициент «контрактности», параметры логнормальных и бета-распределений, используемых для формирования состояний готовности агентов к заключению сделок, вероятность перемещения агентов в пространстве и др.

Каждый агент-потребитель, максимизирует собственную функцию полезности по набору управляющих параметров при ограничениях, имеющих ясный физический и экономический смысл. При этом, интегральным целевым функционалом МСЭС можно считать среднюю (по популяции агентов) полезность будущего потребления:

$$U = \frac{1}{|I|} \sum_{i=1}^{|I|} \sum_{t=1}^{|T|} u_i(t_k). \quad (6)$$

Программная реализация модели (1)–(6) выполнена в среде FLAME GPU с использованием C++ и архитектуры графических процессоров GPU (graphics processing unit), позволяющей, в частности, распараллелить логику поведения агентов посредством специальных функций типа FLAMEGPU_AGENT_FUNCTION. В *таблице 1* представлены ключевые функции, разработанные для рассматриваемой модели со своими входными и выходными параметрами.

Таблица 1.

**Основные функции и процедуры
стохастической модели обмена товарами**

Название функции	Описание	Входные параметры	Выходные параметры
FLAMEGPU_INIT_FUNCTION (init_function)	Инициализация параметров модели. Создание популяции агентов и их размещение в дискретном пространстве.	Нет	Нет
FLAMEGPU_EXIT_CONDITION (exit_condition)	Расчёт целевого функционала и проверка критерия останова.	Нет	Нет
FLAMEGPU_AGENT_FUNCTION (all_agents, flamegpu::MessageNone, flamegpu::MessageArray2D)	Отправка данных о каждом агенте.	Нет	Координаты, тип агента, состояние и др.
FLAMEGPU_AGENT_FUNCTION (all_products, flamegpu::MessageNone, flamegpu::MessageArray2D)	Отправка данных о продукте каждого агента.	Нет	Индекс имеющегося продукта, индекс целевого продукта.
FLAMEGPU_AGENT_FUNCTION (seeking_and_getting_product, flamegpu::MessageArray2D, flamegpu::MessageArray2D)	Поиск и приобретение целевого продукта посредством обмена или за деньги. Поиск ближайшего продавца с желаемым продуктом. Реализация монетарной или бартерной сделки. Пересчёт значения индивидуальной полезности. Отправка данных о приобретаемом продукте и денег в случае монетарной сделки.	Индекс имеющегося продукта, индекс целевого продукта.	Индекс приобретаемого продукта и деньги за продукт (если покупка за деньги).
FLAMEGPU_AGENT_FUNCTION (getting_product_or_money, flamegpu::MessageArray2D, flamegpu::MessageNone)	Получение продукта (при бартере) или денег от покупателя (при монетарной сделке). Пересчёт значения индивидуальной полезности. Завершение торговой операции.	Индекс приобретаемого продукта и деньги за продукт (если покупка за деньги).	Нет
FLAMEGPU_AGENT_FUNCTION (update_agent_state, flamegpu::MessageNone, flamegpu::MessageNone)	Обновление состояния каждого агента и производство нового продукта, если на предыдущем шаге была торговая сделка.	Нет	Нет
FLAMEGPU_AGENT_FUNCTION (update_cell, flamegpu::MessageArray2D, flamegpu::MessageNone)	Обновление состояния каждой ячейки дискретного пространства. Проверка доступности ячейки для занятия агентами.	Координаты, тип агента, состояние и др.	Нет
FLAMEGPU_AGENT_FUNCTION (looking_for_resource, flamegpu::MessageArray2D, flamegpu::MessageArray2D)	Поиск агента, который может быть размещен в данной ячейке с заданной вероятностью.	Координаты, тип агента, состояние и др.	Координаты целевой ячейки. Данные о перемещаемом агенте.
FLAMEGPU_AGENT_FUNCTION (moving_transaction, flamegpu::MessageArray2D, flamegpu::MessageNone)	Случайное перемещение агентов в дискретном пространстве в целевую ячейку.	Координаты целевой ячейки. Данные о перемещаемом агенте.	Нет

2. Процедура синтеза искусственной нейронной сети

В качестве основной конфигурации ИНС для задачи аппроксимации целевого функционала изучаемой МСЭС (агентной модели торговых взаимо-

действий) была выбрана модель MLP (Multilayered perceptron – многослойный перцептрон).

Наиболее важными гиперпараметрами проектируемой ИНС, существенно влияющими на качество аппроксимации являются следующие:

$\mu > 0$ – начальная скорость обучения;

$L = \{l_1, l_2, \dots, l_{|L|}\}$ – количество скрытых слоев в MLP; $n_l > 0, l \in L$ – количество нейронов в каждом из имеющихся скрытых слоев, $n_l \in N$, где N – множество всех нейронов;

$F_l \in \{TANH, ELU, HSig\}, l \in L$ – активационная функция используемая для всех нейронов l -го скрытого слоя (гиперболический тангенс, экспоненциально линейная, «жесткий» сигмоид);

$w_{n_l}, n_l \in N, l \in L$ – весовые коэффициенты n_l -х нейронов l -го скрытого слоя.

При этом, основным критерием качества ИНС в рассматриваемой системе, является ошибка обучения (функция потерь), которая должна быть минимизирована по набору гиперпараметров и весов:

$$\min_{(\bar{r}, |L|, n_l, F_l)} E, n_l \in N, l \in L, \quad (7)$$

где

$$E = \frac{1}{|M|} \sum_{m=1}^{|M|} \left(\tilde{U}_{qm}(\mu, |L|, n_l, F_l, w_{n_l}, X_m) - \hat{U}_m(X_m) \right)^2, \quad (8)$$

где

$M = \{m_1, m_2, \dots, |M|\}$ – множество данных обучающей выборки, где $|M|$ – размер обучающей выборки;

$Q = \{q_1, q_2, \dots, |Q|\}$ – набор итераций алгоритма обучения ИНС, $|Q|$ – все итерации алгоритма машинного обучения;

$\tilde{U}_{qm}, q \in Q, m \in M$ – аппроксимированные значения целевой функции (функции полезности будущего потребления) на выходе ИНС, вычисленные для m -ой выборки данных на q -ой итерации обучения;

$X_m = \{x_{1m}, x_{2m}, \dots, x_{|X_m|m}\}, m \in M$ – набор значений независимых переменных m -ой обучающей выборки (входной слой ИНС);

$\hat{U}_m(X_m), m \in M$ – известные (фактические) значения целевой функции, вычисленные с помощью ранее разработанной агентной модели при заданных значениях входных параметров X_m -ого набора с использованием метода класса Монте-Карло [20, 21].

На рисунке 2 показана блок-схема разработанной двухшаговой процедуры синтеза ИНС, с целью ее дальнейшего использования в качестве суррогатной модели при проведении оптимизационных экспериментов.

На рисунке 2 используются следующие условные обозначения:

$T = \{t_0, t_1, \dots, |T|\}$ – набор итераций роевого алгоритма (PSO), $|T|$ – общее число итераций роевого алгоритма; $t_0 \in T, t_{|T|} \in T$ – начальные и конечные итерации роевого алгоритма;

$Q = \{q_0, q_1, \dots, |Q|\}$ – набор итераций обучающего алгоритма (Adam), $|Q|$ – общее число итераций обучающего алгоритма; $q_0 \in Q, q_{|Q|} \in Q$ – начальные и конечные итерации обучающего алгоритма.

На первом шаге, при относительном малом общем количестве итераций обучающего алгоритма ($|Q| = 1000$), осуществляется оптимизация гиперпараметров ИНС с использованием роевого алгоритма (PSO), агрегированного по целевому функционалу с алгоритмом машинного обучения Adam.

На втором шаге, после того как определены наилучшие значения гиперпараметров ИНС, осуществляется глубокое обучение сформированной ИНС при существенно большем числе итераций обучающего алгоритма – $|Q| = 10000$.

Роевой алгоритм (PSO) [16] обеспечивает перебор гиперпараметров ИНС в направлении улучшения значения целевого функционала, т.е. минимизации ошибки обучения ИНС, вычисляемой с помощью алгоритма машинного обучения (типа Adam [22]). Преимуществом роевого алгоритма является существенно большая высокая временная эффективность, например, в сравнении с классическими генетическими алгоритмами, также используемыми для настройки гиперпараметров ИНС.

В рамках разработанной процедуры, алгоритм PSO агрегирован по целевому функционалу (ошибкой обучения ИНС) с алгоритмом Adam (рис. 2). При использовании роевого алгоритма (PSO) вычисляется вектор скорости изменения значений искомым переменных (гиперпараметров ИНС), определяющий положение i -ых частиц ($i \in I$) в пространстве потенциальных решений в момент t_k ($t_k \in T, k = 1, 2, \dots, K$):

$$\mathbf{v}_i(t_k) = \theta \mathbf{v}_i(t_{k-1}) + c_1 h(0, 1) (\mathbf{x}_i^*(t_{k-1}) - \mathbf{x}_i(t_{k-1})) + c_2 e(0, 1) (\mathbf{x}^g(t_{k-1}) - \mathbf{x}_i(t_{k-1})), \quad (9)$$

$$\mathbf{x}_i(t_k) = \begin{cases} \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}), \\ \text{если } \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}) \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}], \\ \mathbf{x}_i(t_{k-1}), \\ \text{если } \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}) \notin [\underline{\mathbf{x}}, \bar{\mathbf{x}}], \end{cases} \quad (10)$$

где

$I = \{i_1, i_2, \dots, i_{|I|}\}$ – набор индексов частиц PSO, где $|I|$ – общее число частиц;

$\mathbf{x}_i^*(t_{k-1}), \mathbf{x}_i^g(t_{k-1})$ – наилучшие потенциальные значения гиперпараметров ИНС, полученные i -ой частицей PSO за период поиска и всеми частицами в момент t_k ($t_{k-1} \in T$);

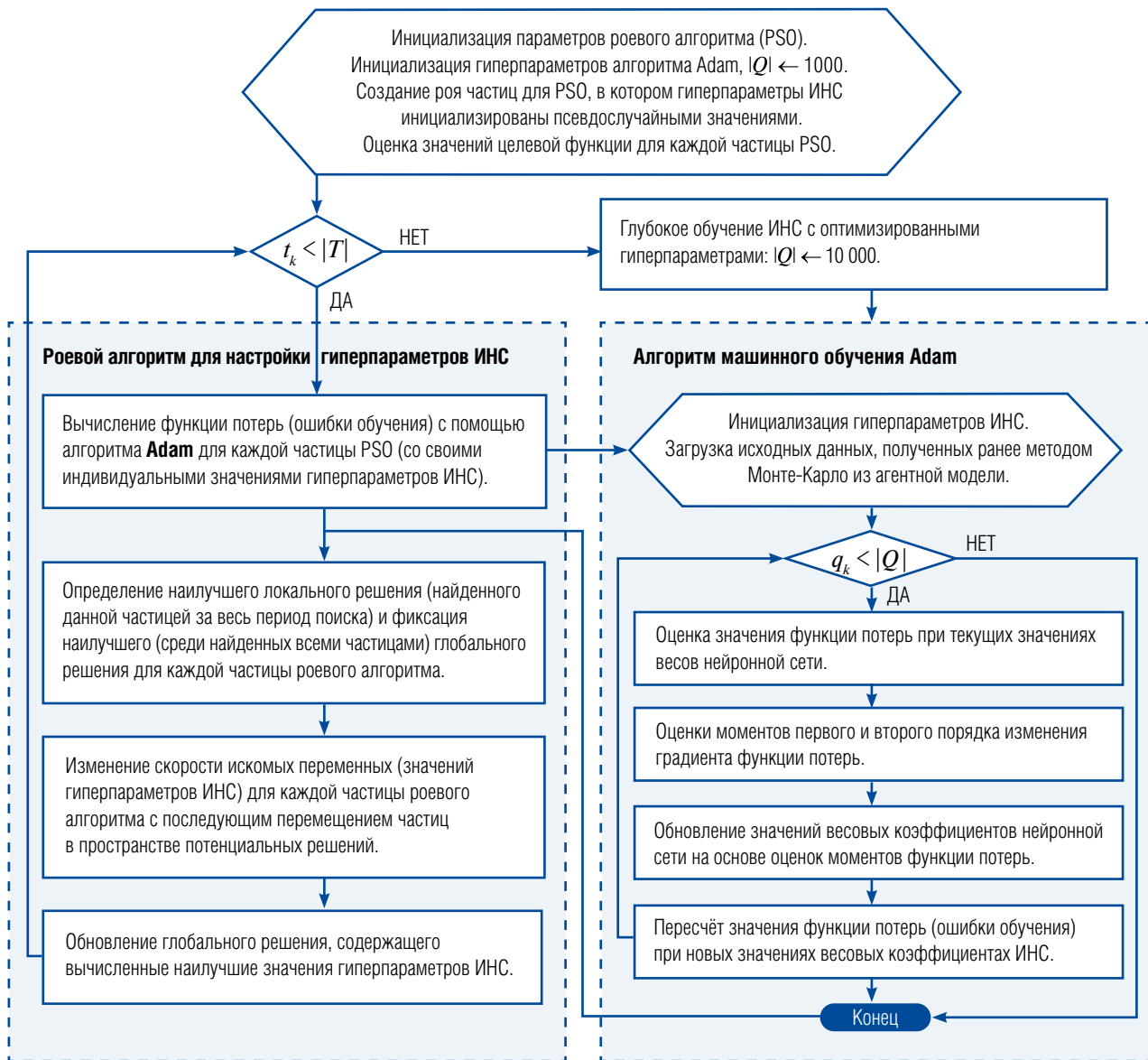


Рис. 2. Процедура синтеза искусственной нейронной сети с использованием метода роя частиц и алгоритма Adam.

$h(0, 1), e(0, 1)$ – случайные величины, равномерно распределенные на отрезке $[0, 1]$;

θ, c_1, c_2 – константы, значения которых, могут быть заданы в следующих диапазонах: $\theta \in [0, 4; 1, 4]$, $c_1 \in [1, 5; 2]$, $c_2 \in [2; 2, 5]$.

Алгоритм машинного обучения Adam [22], обеспечивает вычисление индивидуальных скоростей обучения для разных параметров ИНС (рис. 2), используя оценки первого и второго моментов градиента, чтобы адаптировать скорость обучения для каждого веса нейронной сети.

В алгоритме Adam используется следующее правило обновления значений весовых коэффициентов нейронной сети:

$$w_q = w_{q-1} - \frac{\eta}{\sqrt{\frac{\tilde{\beta}_2 v_{q-1} + (1 - \tilde{\beta}_1)(\nabla_w E_q)^2}{1 - \tilde{\beta}_2} + \epsilon}} \times \frac{\tilde{\beta}_1 m_{q-1} + (1 - \tilde{\beta}_1) \nabla_w E_q}{1 - \tilde{\beta}_1}, \quad (11)$$

где $\nabla_w E_q$ – градиент функции потерь на итерации q ($q \in Q$);

w – весовые коэффициенты нейронной сети;

m_{q-1}, v_{q-1} – оценки моментов первого и второго порядка изменения градиента функции потерь на итерации q ($q \in Q$);

$\tilde{\beta}_1, \tilde{\beta}_2$ – гиперпараметры алгоритма Adam (как правило, задаваемые на уровне 0,9 и 0,99, соответственно);

ε – достаточно малое число.

Предложенная процедура синтеза ИНС с использованием метода роя частиц и алгоритма машинного обучения Adam реализована с использованием C++ и OpenNN. Основанием выбора OpenNN в качестве платформы для машинного обучения является обеспечение достаточно высокого быстродействия, сопоставимого с такими популярными программными библиотеками как TensorFlow и PyTorch при более удобной и простой среде разработки [19].

В результате выполнения данной процедуры обеспечивается автоматическое формирование программных модулей синтезированной ИНС на C++ и Python, которые могут быть использованы в качестве суррогатных моделей в параллельных генетических алгоритмах вещественного кодирования (RCGA) [2, 3, 23], что позволяет принципиально повысить временную эффективность процедуры поиска оптимальных решений для многоагентных социально-экономических систем.

Архитектура ИНС, сформированная с использованием данных, полученных с помощью предложенной агентной модели торговых взаимодействий, состоит из входного слоя, одного скрытого слоя и выходного слоя. Первый слой обеспечивает распределение входных сигналов остальным нейронам. При этом, входными сигналами в данной ИНС являются масштабированные (нормированные) значения управляющих параметров агентной модели, в частности, следующие:

- ◆ конфигурация (числовой эквивалент) начального распределения агентов в МСЭС;
- ◆ параметры логнормального и бета-распределений, применяемых для формирования состояний готовности агентов к сделкам в различные моменты времени;
- ◆ радиус торгового взаимодействия;
- ◆ вероятность перемещения агентов в пространстве и др.

Промежуточный слой состоит из десяти нейронов, использующих в качестве функции активации гиперболический тангенс.

Выходной слой состоит из одного нейрона, обеспечивающего вычисление аппроксимированного значения целевой функции – средней полезности будущего потребления.

Далее, синтезированная ИНС была интегрирована по целевому функционалу с ранее предложенным параллельным многоагентным генетическим алгоритмом MA-RCGA [23], реализованным на платформе FLAME GPU 2 [18], что позволило существенно повысить временную эффективность процедуры эволюционного поиска при решении важной задачи МСЭС по максимизации функции средней полезности будущего потребления (6). Возможности применения суррогатных ИНС-моделей для повышения производительности эволюционных алгоритмов подробно описаны в работе [14].

3. Результаты численных и оптимизационных экспериментов

На *рисунке 3* представлен анализ чувствительности процессного времени и точности полученного решения (т.е. близости решения к известному оптимуму) в зависимости от частоты использования предложенной агентной модели МСЭС для расчета целевой и фитнес-функции вместо соответствующей суррогатной модели. Численные эксперименты были проведены с использованием локального суперкомпьютера ЦЭМИ РАН –

Процессное время поиска решения, сек.

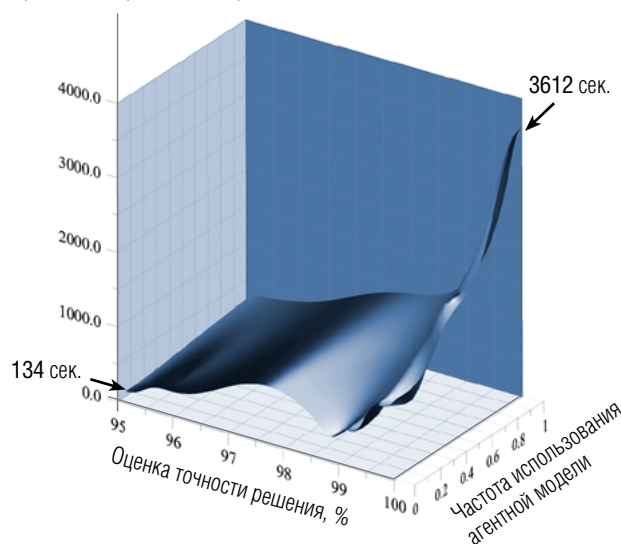


Рис. 3. Анализ чувствительности процессного времени и точности полученного решения в зависимости от частоты использования эталонной агентной модели.

DSWS PRO (2 x Intel Xeon Silver 4114, 1 x NVIDIA QUADRO RTX 600) при использовании 10 параллельных взаимодействующих агентов-процессов, реализующих процедуру эволюционного поиска. Подобная процедура основана известных эвристических операторах селекции, кроссингвера и мутации. При этом, для пересчёта значений целевой и фитнес-функции используется как оригинальная (эталонная) агентная модель торговых взаимодействий, так и соответствующая суррогатная ИНС-модель.

Из рисунка 3 следует, что даже в условиях превалирующего использования суррогатной ИНС-модели, обеспечивается достаточно высокий уровень точности полученного решения (более 95%). При этом, достигается существенно большая временная эффективность процедуры поиска решения в сравнении с подходом, при котором для пересчётов значений целевой и фитнес-функции в MA-RCGA применяется исключительно оригинальная (эталонная) агентная модель (рис. 3). Таким образом, точность оптимиза-

ционного алгоритма при периодическом использовании суррогатной модели непосредственно зависит от аппроксимационных характеристик соответствующей ИНС.

На рисунке 4 представлены результаты оптимизационных экспериментов по максимизации средней (для ансамбля агентов) полезности будущего потребления, выполненные с использованием ранее разработанного генетического алгоритма MA-RCGA [23] и сформированной суррогатной ИНС-модели. При этом, используются различные конфигурации начального расположения агентов в пространстве и различные способы формирования состояний готовности агентов к сделкам.

Из рисунка 4 следует два важных вывода. Во-первых, выбор начальной конфигурации размещения агентов в пространстве влияет на значение целевого функционала исследуемой МСЭС — средней полезности будущего потребления, а во-вторых, влияет на выбор рационального способа формирования состояний готовности к сделкам. В

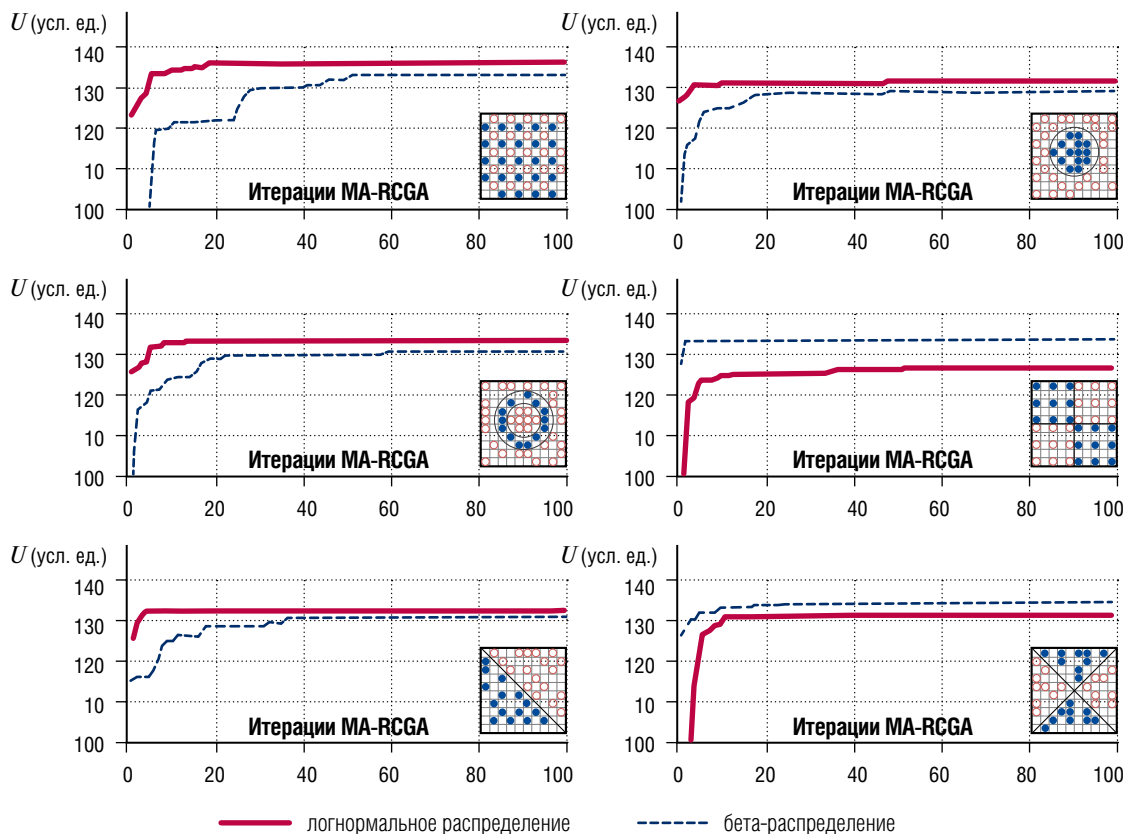


Рис. 4. Результаты оптимизационных экспериментов по максимизации средней полезности будущего потребления (U) при формировании состояний готовности к сделкам.

частности, при некоторых конфигурациях (рис. 4) использование бета-распределения оказывается более предпочтительным по сравнению с логнормальным распределением.

Заключение

В данной статье представлен новый подход к моделированию и оптимизации стратегий принятия индивидуальных решений в крупномасштабных многоагентных социально-экономических системах (МСЭС) на примере предложенной агентной модели торговых взаимодействий. Разработана новая процедуры синтеза и обучения искусственной нейронной сети (ИНС), основанная на комбинированном использовании методов роевой оптимизации (для определения оптимальных значений гиперпараметров) и алгоритма машинного обучения Adam (для вычисления весовых коэффициентов ИНС). Построенная ИНС, относящаяся к классу многослойного перцептрона (MLP), используется в качестве суррогатной модели, встраиваемой в ранее разработанный многоагентный генетический алгоритм (MA-RCGA) для аппроксимации

значений целевой и фитнес-функции – средней (по ансамблю агентов) полезности будущего потребления. В результате проведенных численных исследований показано, что даже в условиях преобладающего использования суррогатной ИНС-модели, обеспечивается достаточно высокий уровень точности полученного решения. При этом, выбор начальной конфигурации размещения агентов в пространстве влияет на значение целевого функционала и оптимальный способ формирования состояний готовности агентов к сделкам.

Дальнейшие исследования будут направлены на создание имитационных моделей крупномасштабных многоагентных социально-экономических систем, использующих методы машинного обучения и генетические оптимизационные алгоритмы для оптимального управления поведением ансамбля взаимодействующих экономических агентов. ■

Благодарности

Исследование выполнено за счет гранта Российского научного фонда (проект № 23-21-00012).

Литература

1. Akopov A.S. Parallel genetic algorithm with fading selection // International Journal of Computer Applications in Technology. 2014. Vol. 49. No. 3–4. P. 325–331. <https://doi.org/10.1504/IJCAT.2014.062368>
2. Акопов А.С., Бекларян А.Л., Тхакур М., Верма Б.Д. Разработка параллельных генетических алгоритмов вещественного кодирования для систем поддержки принятия решений социально-экономического и экологического планирования // Бизнес-информатика. 2019. Т. 13. № 1. С. 33–44. <https://doi.org/10.17323/1998-0663.2019.1.33.44>
3. Akopov A.S., Beklaryan L.A., Thakur M. Improvement of maneuverability within a multiagent fuzzy transportation system with the use of parallel biobjective real-coded genetic algorithm // IEEE Transactions on Intelligent Transportation Systems. 2022. Vol. 23. No. 8. P. 12648–12664. <https://doi.org/10.1109/TITS.2021.3115827>
4. Акопов А.С., Бекларян Л.А. Моделирование динамики дорожно-транспортных происшествий с участием беспилотных автомобилей в транспортной системе «умного города» // Бизнес-информатика. 2022. Т. 16. № 4. С. 19–35. <https://doi.org/10.17323/2587-814X.2022.4.19.35>
5. Khachatryan N.K., Akopov A.S. Model for organizing cargo transportation with an initial station of departure and a final station of cargo distribution // Business Informatics. No. 1 (39). P. 25–35. <https://doi.org/10.17323/1998-0663.2017.1.25.35>
6. Саати Т. Принятие решений. Метод анализа иерархий. Перевод с английского Р. Г. Вачнадзе. М.: Радио и связь, 1993.
7. Кравченко Т.К., Исаев Д.В. Системы поддержки принятия решений. М.: Юрайт, 2017.
8. Кравченко Т.К., Брускин С.Н., Исаев Д.В., Кузнецова Е.В. Приоритизация элементов бэклога ИТ-продукта с применением систем поддержки принятия решений // Информационные технологии. 2020. Т. 26. № 11. С. 631–640. <https://doi.org/10.17587/it.26.631-640>
9. Kravchenko T., Shevgunov T. Equivalent exchange method for decision-making in case of alternatives with incomparable attributes // Inventions. 2023. Vol. 8. No. 1. Article 12. <https://doi.org/10.3390/inventions8010012>
10. Поспелов И.Г. Модель случайных продаж // Математические заметки. 2018. Т. 103. № 3. С. 445–459. <https://doi.org/10.1134/S0001434618030112>
11. Макаров В.Л., Бахтизин А.Р., Эпштейн Д. Агент-ориентированное моделирование для сложного мира. Часть 1 // Экономика и математические методы. 2022. Том 58. № 1. С. 5–26. <https://doi.org/10.31857/S042473880018970-6>
12. Макаров В.Л., Бахтизин А.Р., Эпштейн Д. Агент-ориентированное моделирование для сложного мира. Часть 2 // Экономика и математические методы. 2022. Том 58. № 2. С. 7–21. <https://doi.org/10.31857/S042473880020009-8>

13. Tan R.K., Qian C., Wang M., Ye W. An efficient data generation method for ANN-based surrogate models // *Structural and Multidisciplinary Optimization*. 2022. Vol. 65. Article 90. <https://doi.org/10.1007/s00158-022-03180-6>
14. Díaz-Manríquez A., Toscano G., Barron-Zambrano J.H., Tello-Leal E. A review of surrogate assisted multiobjective evolutionary algorithms // *Computational Intelligence and Neuroscience*. 2016. Vol. 2016. Article 9420460. <https://doi.org/10.1155/2016/9420460>
15. Herrera F., Lozano M., Verdegay J.L. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis // *Artificial Intelligence Review*. 1998. Vol. 12. No. 4. P. 265–319. <https://doi.org/10.1023/A:1006504901164>
16. Bonyadi M.R., Michalewicz Z. Particle swarm optimization for single objective continuous space problems: A review // *Evolutionary Computation*. 2017. Vol. 25. No. 1. P. 1–54.
17. Бекларян Л.А., Флёрова А.Ю., Жукова А.А. Методы оптимального управления: учебное пособие. МФТИ, 2018.
18. Richmond P., Chisholm R., Heywood P., Leach M., Kabiri C.M. FLAME GPU. Zenodo, 2021. <https://doi.org/10.5281/zenodo.5428984>
19. Lopez R. Open NN: An open source neural networks C++ library [software]. 2014. [Электронный ресурс]: <http://www.cimne.com/flood> (дата обращения 10.05.2023).
20. Whitelam S., Selin V., Benlolo I., Casert C., Tamblyn I. Training neural networks using Metropolis Monte Carlo and an adaptive variant // *Machine Learning: Science and Technology*. 2022. Vol. 3. Article 045026. <https://doi.org/10.1088/2632-2153/aca6cd>
21. de Freitas J.F., Niranjani M., Gee A.H., Doucet A. Sequential Monte Carlo methods to train neural network models // *Neural Computation*. 2000. Vol. 12. No. 4. P. 955–93. <https://doi.org/10.1162/089976600300015664>
22. Kingma D.P., Ba J. Adam: A method for stochastic optimization // arXiv preprint. 2014. arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
23. Akopov A.S., Beklaryan L.A., Thakur M., Verma B.D. Parallel multi-agent real-coded genetic algorithm for large-scale black-box single-objective optimisation // *Knowledge-Based Systems*. 2019. Vol. 174. P. 103–122. <https://doi.org/10.1016/j.knosys.2019.03.003>

Об авторе

Акопов Андраник Сумбатович

доктор технических наук, профессор, профессор Российской Академии Наук;

главный научный сотрудник, лаборатория динамических моделей экономики и оптимизации, Центральный экономико-математический институт, Российская академия наук, 117418, г. Москва, Нахимовский проспект, д. 47;

E-mail: akopovas@umail.ru

ORCID: 0000-0003-0627-3037

Modeling and optimization of strategies for making individual decisions in multi-agent socio-economic systems with the use of machine learning

Andranik S. Akopov

E-mail: akopovas@umail.ru

Central Economics and Mathematics Institute, Russian Academy of Sciences
Address: 47, Nakhimovsky Prospect, Moscow 117418, Russia

Abstract

This article presents a new approach to modeling and optimizing individual decision-making strategies in multi-agent socio-economic systems (MSES). This approach is based on the synthesis of agent-based modeling methods, machine learning and genetic optimization algorithms. A procedure for the synthesis and training of artificial neural networks (ANNs) that simulate the functionality of MSES and provide an approximation of the values of its objective characteristics has been developed. The feature of the two-step procedure is the combined use of particle swarm optimization methods (to determine the optimal values of hyperparameters) and the Adam machine learning algorithm (to compute weight coefficients of the ANN). The use of such ANN-based surrogate models in parallel multi-agent real-coded genetic algorithms (MA-RCGA) makes it possible to raise substantially the time-efficiency of the evolutionary search for optimal solutions. We have conducted numerical experiments that confirm a significant improvement in the performance of MA-RCGA, which periodically uses the ANN-based surrogate-model to approximate the values of the objective and fitness functions. A software framework has been designed that consists of the original (reference) agent-based model of trade interactions, the ANN-based surrogate model and the MA-RCGA genetic algorithm. At the same time, the software libraries FLAME GPU, OpenNN (Open Neural Networks Library), etc., agent-based modeling and machine learning methods are used. The system we developed can be used by responsible managers.

Keywords: multi-agent socio-economic systems, particle swarm optimization, modeling random sales, machine learning, artificial neural networks, genetic optimization algorithms

Citation: Akopov A.S. (2023) Modeling and optimization of strategies for making individual decisions in multi-agent socio-economic systems with the use of machine learning. *Business Informatics*, vol. 17, no. 2, pp. 7–19. DOI: 10.17323/2587-814X.2023.2.7.19

References

1. Akopov A.S. (2014) Parallel genetic algorithm with fading selection. *International Journal of Computer Applications in Technology*, vol. 49, nos. 3–4, pp. 325–331. <https://doi.org/10.1504/IJCAT.2014.062368>
2. Akopov A.S., Beklaryan A.L., Thakur M., Verma B.D. (2019) Developing parallel real-coded genetic algorithms for decision-making systems of socio-ecological and economic planning. *Business Informatics*, vol. 13, no. 1, pp. 33–44. <https://doi.org/10.17323/1998-0663.2019.1.33.44>
3. Akopov A.S., Beklaryan L.A., Thakur M. (2022) Improvement of maneuverability within a multiagent fuzzy transportation system with the use of parallel biobjective real-coded genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12648–12664. <https://doi.org/10.1109/TITS.2021.3115827>
4. Akopov A.S., Beklaryan L.A. (2022) Simulation of rates of traffic accidents involving unmanned ground vehicles within a transportation system for the ‘smart city’. *Business Informatics*, vol. 16, no. 4, pp. 19–35. <https://doi.org/10.17323/2587-814X.2022.4.19.35>
5. Khachatryan N.K., Akopov A.S. (2017) Model for organizing cargo transportation with an initial station of departure and a final station of cargo distribution. *Business Informatics*, no. 1 (39), pp. 25–35. <https://doi.org/10.17323/1998-0663.2017.1.25.35>
6. Saaty T. (1993) *Decision making. The Analytic Hierarchy Process*. Translation from English by R.G. Vachnadze. Moscow: Radio and communication (in Russian).
7. Kravchenko T.K., Isaev D.V. (2017) *Decision support systems*. Moscow: Urait (in Russian).
8. Kravchenko T.K., Bruskin S.N., Isaev D.V., Kuznetsova E.V. (2020) Prioritization of IT product backlog items using decision support systems. *Informacionnye Tehnologii*, vol. 26, no. 11, pp. 631–640 (in Russian). <https://doi.org/10.17587/it.26.631-640>
9. Kravchenko T., Shevgunov T. (2023) Equivalent exchange method for decision-making in case of alternatives with incomparable attributes. *Inventions*, vol. 8, no. 1, article 12. <https://doi.org/10.3390/inventions8010012>
10. Pospelov I.G. (2018) A model of random sales. *Mathematical Notes*, vol. 103, pp. 453–465. <https://doi.org/10.1134/S0001434618030112>
11. Makarov V., Bakhtizin A., Epstein J. (2022) Agent-based modelling for a complex world. Part 1. *Economics and the Mathematical Methods*, vol. 58, no. 1, pp. 5–26 (in Russian). <https://doi.org/10.31857/S042473880018970-6>
12. Makarov V., Bakhtizin A., Epstein J. (2022) Agent-based modelling for a complex world. Part 2. *Economics and the Mathematical Methods*, vol. 58, no. 2, pp. 7–21 (in Russian). <https://doi.org/10.31857/S042473880020009-8>
13. Tan R.K., Qian C., Wang M., Ye W. (2022) An efficient data generation method for ANN-based surrogate models. *Structural and Multidisciplinary Optimization*, vol. 65, article 90. <https://doi.org/10.1007/s00158-022-03180-6>

14. Díaz-Manríquez A., Toscano G., Barron-Zambrano J.H., Tello-Leal E. (2016) A review of surrogate assisted multiobjective evolutionary algorithms. *Computational Intelligence and Neuroscience*, vol. 2016, article 9420460. <https://doi.org/10.1155/2016/9420460>
15. Herrera F., Lozano M., Verdegay J.L. (1998) Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319. <https://doi.org/10.1023/A:1006504901164>
16. Bonyadi M.R., Michalewicz Z. (2017) Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54.
17. Beklaryan L.A., Flerova A.Y., Zhukova A.A. (2018) *Optimal control methods: textbook*. MIPT (in Russian).
18. Richmond P., Chisholm R., Heywood P., Leach M., Kabiri C.M. (2021) *FLAME GPU Zenodo*. <https://doi.org/10.5281/zenodo.5428984>
19. Lopez R. (2014) *Open NN: An open source neural networks C++ library [software]*. Available at: <http://www.cimne.com/flood> (accessed 10 May 2023).
20. Whitelam S., Selin V., Benlolo I., Casert C., Tamblyn I. (2022) Training neural networks using Metropolis Monte Carlo and an adaptive variant. *Machine Learning: Science and Technology*, vol. 3, article 045026. <https://doi.org/10.1088/2632-2153/aca6cd>
21. de Freitas J.F., Niranjani M., Gee A.H., Doucet A. (2000) Sequential Monte Carlo methods to train neural network models. *Neural Computation*, vol. 12, no. 4, pp. 955–993. <https://doi.org/10.1162/089976600300015664>
22. Kingma D.P., Ba J. (2014) Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
23. Akopov A.S., Beklaryan L.A., Thakur M., Verma B.D. (2019) Parallel multi-agent real-coded genetic algorithm for large-scale black-box single-objective optimisation. *Knowledge-Based Systems*, vol. 174, pp. 103–122. <https://doi.org/10.1016/j.knsys.2019.03.003>

About the author

Andranik S. Akopov

Dr. Sci. (Tech.), Professor of the Russian Academy of Sciences;

Chief Researcher, Laboratory of Dynamic Models of Economy and Optimisation, Central Economics and Mathematics Institute, Russian Academy of Sciences, 47, Nachimovky Prospect, Moscow 117418, Russia;

E-mail: akopovas@umail.ru

ORCID: 0000-0003-0627-3037