

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ АВТОМАТИЗИРОВАННОГО АНАЛИЗА СОСТОЯНИЯ РАСПРЕДЕЛЕННЫХ ПРОГРАММНО-АППАРАТНЫХ КОМПЛЕКСОВ НА ОСНОВЕ МОДИФИЦИРОВАННОГО МАРКОВСКОГО ПРОЦЕССА ПРИНЯТИЯ РЕШЕНИЙ

Р.Г. Астраханцев

Роман Геннадьевич Астраханцев (ORCID 0000-0001-9880-2826)

Ивановский государственный химико-технологический университет, Шереметевский пр., 7. Иваново, 153000, Россия

E-mail: rgastrakhantsev@gmail.com

Разработана математическая модель автоматизированного анализа состояния распределенных программно-аппаратных комплексов на основе модифицированного децентрализованного частично наблюдаемого марковского процесса принятия решений. Предложено формальное определение распределенного программно-аппаратного комплекса как кортежа из четырех элементов. Проведен сравнительный анализ математических аппаратов, применимых для формализации задачи, и обоснован выбор децентрализованного частично наблюдаемого марковского процесса. Изложена классическая модель с девятью элементами, включая функции переходов и наблюдений, байесовское обновление состояния веры, определение политики агента и уравнение Беллмана для оптимальной функции ценности. Предложены три авторских модификации классической модели, адаптирующие ее к специфике задачи анализа. Введено композитное пространство состояний, объединяющее состояние среды и информационное состояние агентской системы. Определено динамическое пространство действий, зависящее от текущего информационного состояния. Предложена функция верификации, заменяющая пошаговое вознаграждение оценкой трассы исполнения. Получена оценка мощности пространства состояний, растущей экспоненциально с числом узлов комплекса, и обоснована необходимость применения приближенных методов решения.

Ключевые слова: распределенный программно-аппаратный комплекс, децентрализованный частично наблюдаемый марковский процесс принятия решений, многоагентная система, математическая модель, пространство состояний, функция верификации, автоматизированный анализ.

MATHEMATICAL MODEL OF AUTOMATED STATE ANALYSIS FOR DISTRIBUTED COMPUTING ENVIRONMENTS BASED ON A MODIFIED DECENTRALIZED PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

R.G. Astrakhantsev

Roman G. Astrakhantsev (ORCID 0000-0001-9880-2826)

Ivanovo State University of Chemistry and Technology, Sheremetevsky avenue, 7, Ivanovo, 153000, Russia

E-mail: rgastrakhantsev@gmail.com

A mathematical model for automated state analysis of distributed computing environments based on a modified decentralized partially observable Markov decision process is developed. A formal definition of a distributed software-hardware complex as a four-element tuple is proposed. A comparative analysis of mathematical formalisms applicable to the problem is conducted, justifying the choice of a decentralized partially observable Markov decision process. The classical nine-element model is presented, including transition and observation functions, Bayesian belief state update, agent policy definition, and the Bellman equation for the optimal value function. Three author's modifications to the classical model are proposed, adapting it to the specifics of the analysis task. A composite state space combining the environment state and the agent system's information state is introduced. A dynamic action space dependent on the

current information state is defined. A verification function replacing step-wise rewards with execution trace evaluation is proposed. An estimate of the state space cardinality, growing exponentially with the number of complex nodes, is obtained, justifying the necessity of approximate solution methods.

Keywords: Distributed computing environment, decentralized partially observable Markov decision process, multi-agent system, mathematical model, state space, verification function, automated analysis.

Для цитирования:

Астраханцев Р.Г. Математическая модель автоматизированного анализа состояния распределенных программно-аппаратных комплексов на основе модифицированного марковского процесса принятия решений. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2026. № 02(68). С. 117-128. DOI: 10.6060/ivecofin.2026682.786

For citation:

Astrakhantsev R.G. Mathematical model of automated state analysis for distributed computing environments based on a modified decentralized partially observable markov decision process. *Ivecofin*. 2026. N 02(68). P. 117-128. DOI: 10.6060/ivecofin.2026682.786 (in Russian)

ВВЕДЕНИЕ

Автоматизированный анализ состояния распределенных программно-аппаратных комплексов относится к классу задач, где несколько автономных программных сущностей должны координировать действия в условиях неполной информации о среде [1, 2]. Распределенный комплекс состоит из территориально разнесенных вычислительных узлов, гетерогенных программных служб и многочисленных конфигурационных параметров. Ни один отдельный агент не способен одновременно наблюдать все компоненты такой среды, что порождает фундаментальную проблему частичной наблюдаемости [3-6]. Практическая потребность в автоматизации анализа подобных сред подтверждается исследованиями в области оценки принадлежности цифровых активов [7] и анализа пользовательской активности в защищенных операционных средах [8].

Формализация подобных задач требует математического аппарата, способного одновременно учитывать множественность агентов, стохастичность переходов между состояниями среды и частичную наблюдаемость каждого отдельного агента. Марковский процесс принятия решений предполагает полную наблюдаемость и единственного агента, что делает его неприменимым [9]. Частично наблюдаемый марковский процесс учитывает неопределенность наблюдений, однако ограничен одним агентом [10]. Децентрализованный марковский процесс моделирует множественность агентов, но не учитывает ограничения наблюдаемости [11].

Единственным аппаратом, объединяющим все три свойства, является децентрализованный частично наблюдаемый марковский процесс принятия решений [3, 4, 12]. Вместе с тем классическая постановка этой модели содержит ряд допущений, несовместимых со спецификой задачи

анализа распределенных комплексов. Пространство состояний классической модели описывает только среду, тогда как при анализе комплексов существенную роль играет информационное состояние самой агентской системы, включающее накопленные результаты обследования. Множество доступных действий в классической модели фиксировано, хотя при анализе распределенных комплексов набор возможных действий расширяется по мере накопления информации. Функция вознаграждения классической модели вычисляется после каждого действия, тогда как результат анализа разумно оценивать однократно, после завершения всего сеанса обследования.

Целью настоящей работы является разработка математической модели автоматизированного анализа состояния распределенных программно-аппаратных комплексов на основе модификации децентрализованного частично наблюдаемого марковского процесса принятия решений.

МАТЕРИАЛЫ И МЕТОДЫ

Основу формализации составляет аппарат децентрализованного частично наблюдаемого марковского процесса принятия решений, разработанный Бернштейном и соавторами [3] и обобщенный в работах Олихука [4] и Бусонь [12]. Этот аппарат позволяет моделировать задачи кооперативного принятия решений множеством автономных агентов, каждый из которых обладает лишь частичным представлением о состоянии среды. Для описания неопределенности используется байесовский вывод, позволяющий рекурсивно обновлять распределение вероятностей по состояниям среды на основе последовательности наблюдений [10, 13]. Аналогичные вероятностные методы успешно применяются для моделирования стохастических процессов в иных предметных областях, в том числе при моделировании проточных трубчатых реакторов

[14]. Для нахождения оптимальной политики применяется принцип оптимальности Беллмана, устанавливающий рекурсивную связь между ценностью текущего состояния и ценностями достижимых состояний [9]. Применение методов машинного обучения для решения задач оптимизации предполагает использование современных подходов, включая применение больших языковых моделей для поддержки принятия решений в условиях неполной информации [15].

Терминологическая база опирается на ГОСТ Р 59277–2020, устанавливающий определения в области многоагентных систем [1,16]. Вычислительная сложность задачи нахождения оптимальной совместной политики децентрализованного частично наблюдаемого марковского процесса принятия решений является сверхэкспоненциальной и принадлежит классу NEXP-полных задач [3], что обосновывает необходимость приближенных подходов [17]. Опыт применения методов машинного обучения для решения смежных задач фильтрации и классификации результатов автоматизированного анализа описан в работе [7].

РЕЗУЛЬТАТЫ

Разработка многоагентной интеллектуальной системы требует строгой формализации решаемой задачи. Автоматизированный анализ состояния распределенных программно-аппаратных комплексов по своей природе является задачей, в которой несколько автономных агентов действуют параллельно, обладая лишь фрагментарной информацией об исследуемой среде, а среда изменяется как под влиянием внешних факторов, так и в результате действий самих агентов.

На основе выявленных характеристик автором предлагается под распределенным программно-аппаратным комплексом понимать сложную техническую систему, образованную множеством территориально разнесенных вычислительных узлов, объединенных сетевой инфраструктурой, на которых функционируют программные службы различных типов, управляемые совокупностью конфигурационных параметров. Формально распределенный программно-аппаратный комплекс предлагается определять как кортеж

$$K = \langle N, E, \Sigma, C \rangle, \quad (1)$$

где $N = \{n_1, n_2, \dots, n|N|\}$ - конечное множество вычислительных узлов, каждый из которых представляет собой физический или виртуальный вычислительный ресурс, обладающий собственной операционной системой, набором установленных служб и сетевым адресом;

$E \subseteq N \times N$ - множество сетевых соединений между узлами, определяющее топологию комплекса и возможные маршруты передачи данных;

$\Sigma = \{ \sigma_1, \sigma_2, \dots, \sigma_{|\Sigma|} \}$ — множество программных служб, функционирующих на узлах, при этом каждая служба σ_j характеризуется кортежем атрибутов (тип, версия, конфигурация, множество предоставляемых интерфейсов);

$C = \{c_1, c_2, \dots, c|C|\}$ - множество конфигурационных параметров, определяющих состояние комплекса в момент времени t .

Графическое представление формальной модели комплекса приведено на рис. 1.

Свойства программно-аппаратного комплекса напрямую обуславливают выбор математического аппарата. Территориальная распределенность и гетерогенность порождают потребность в нескольких параллельно действующих агентах. Динамичность среды требует учета стохастичности переходов между состояниями. Невозможность полного наблюдения каждым агентом всего комплекса создает условия частичной наблюдаемости. Автором проведен сравнительный анализ математических аппаратов, потенциально применимых для формализации задачи. Результаты анализа представлены в табл. 1.

Как следует из табл. 1 единственным аппаратом, одновременно учитывающим частичную наблюдаемость, множественность агентов и стохастичность переходов, является децентрализованный частично наблюдаемый марковский процесс принятия решений. Этот аппарат и положен в основу дальнейшей формализации.

Децентрализованный частично наблюдаемый марковский процесс принятия решений определяется кортежем из девяти элементов

$$M = \langle I, S, \{A_i\}_{i \in I}, T, R, \{\Omega_i\}_{i \in I}, O, \gamma, b_0 \rangle, \quad (2)$$

где $\{A_i\}_{i \in I}$ - множества действий для каждого агента a_i , при этом совместное действие всех агентов определяется как кортеж $a = (a_1, \dots, a_n) \in A = A_1 \times \dots \times A_n$, где $\times$ обозначает декартово произведение;

T - функция переходов, определяющая вероятность перехода системы из одного состояния в другое при выполнении совместного действия;

R - функция вознаграждения, сопоставляющая каждой паре (состояние, действие) вещественное число, характеризующее качество выбранного действия;

$\{\Omega_i\}_{i \in I}$ - множества наблюдений для каждого агента, при этом каждый агент получает только свою компоненту совместного наблюдения;

O - функция наблюдений, задающая вероятность того, что агенты получают совместное

наблюдение $o = (o_1, \dots, o_n)$ после перехода среды в новое состояние s' при выполнении совместного действия a . Каждый агент получает только свою компоненту o_i .

γ - коэффициент дисконтирования, $\gamma \in [0, 1]$, определяющий степень значимости будущих вознаграждений относительно текущих;

b_0 - начальное распределение вероятностей по состояниям, отражающее априорные знания о конфигурации исследуемого комплекса до начала анализа.

Графическое представление модели (2) приведено на рис. 2.

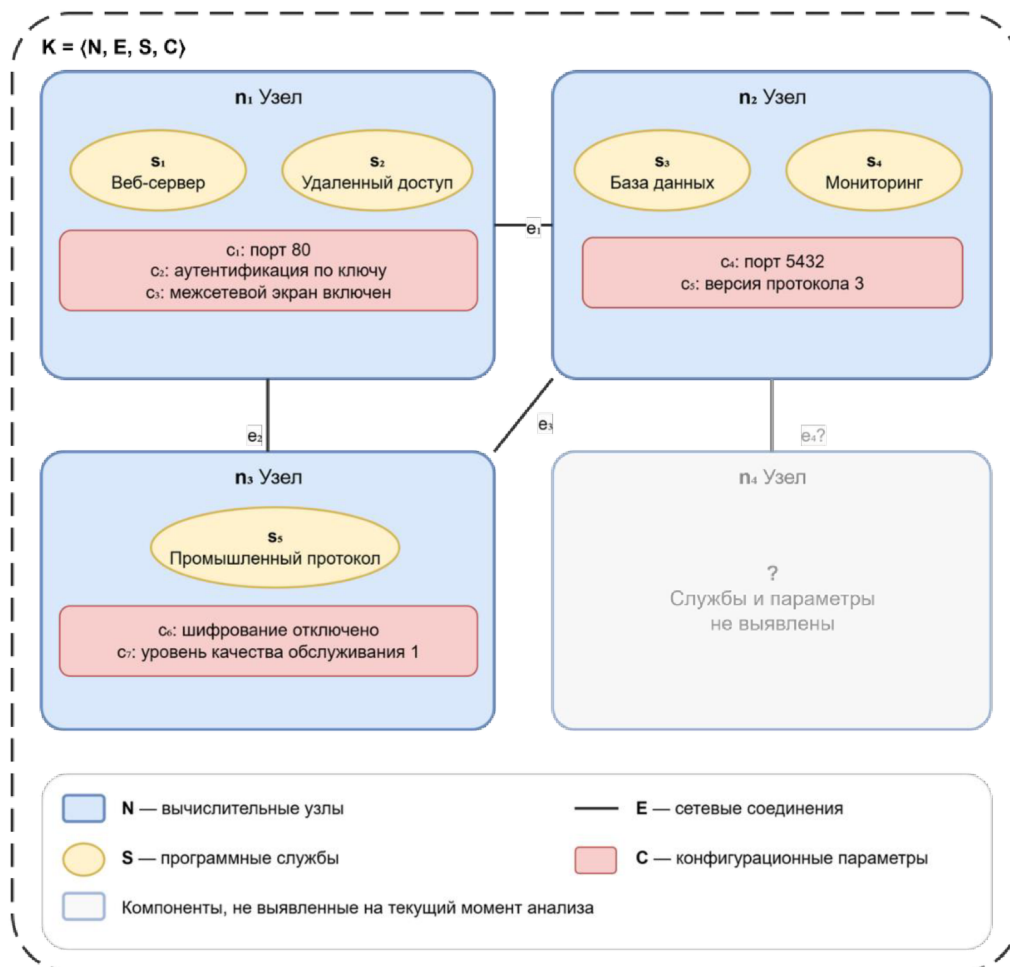


Рисунок 1. Формальное представление распределенного программно-аппаратного комплекса
Figure 1. Formal representation of the distributed software-hardware complex

Источник: разработано автором

Source: compiled by the author

Таблица 1. Сравнительный анализ математических аппаратов для формализации задачи
Table 1. Comparative analysis of mathematical formalisms for the problem

Аппарат	Частичная наблюдаемость	Множественность агентов	Стохастичность переходов
Марковский процесс принятия решений	Нет	Нет	Да
Частично наблюдаемый марковский процесс	Да	Нет	Да
Децентрализованный марковский процесс	Нет	Да	Да
Децентрализованный частично наблюдаемый марковский процесс	Да	Да	Да

Источник: разработано автором

Source: compiled by the author

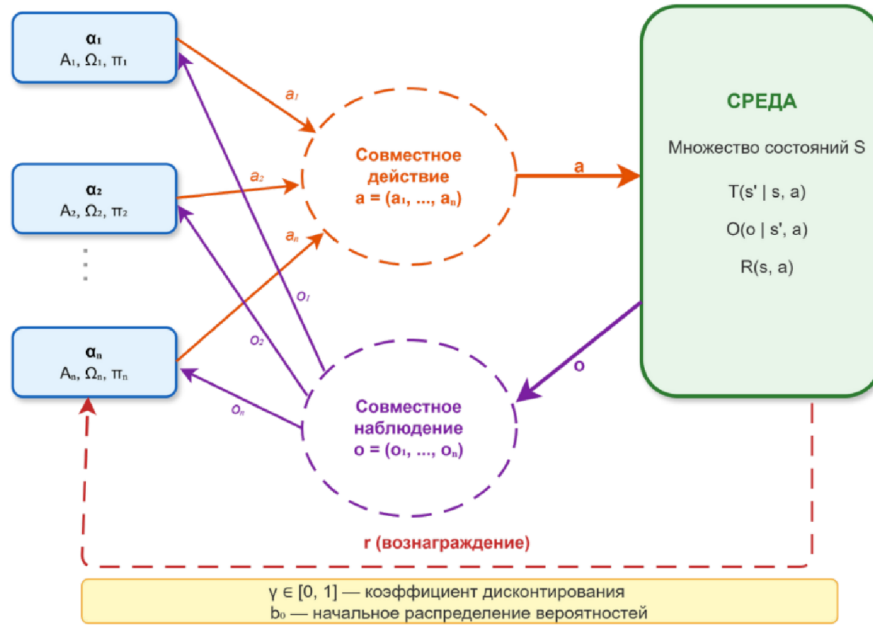


Рисунок 2. Графическое представление децентрализованного частично наблюдаемого марковского процесса принятия решений

Figure 2. Graphical representation of the decentralized partially observable Markov decision process

Источник: разработано автором

Source: compiled by the author

Функция переходов определяет вероятность перехода системы из состояния s в состояние s' при выполнении совместного действия a

$$T(s' | s, a) = P(s_{t+1} = s' | s_t = s, a_t = a), \quad (3)$$

где T - функция переходов, отображающую тройку (текущее состояние, совместное действие, следующее состояние) в вероятность;

$s \in S$ - текущее состояние системы;

$s' \in S$ - состояние, в которое система переходит после выполнения действия;

$a \in A$ - совместное действие всех агентов.

Функция (3) удовлетворяет условию нормировки для всех $s \in S$ и $a \in A$

$$\sum_{s' \in S} T(s' | s, a) = 1. \quad (4)$$

Сумма берется по всем возможным состояниям $s' \in S$, означая, что система с вероятностью один перейдет в какое-либо из допустимых состояний. Это свойство гарантирует замкнутость модели и является фундаментальным требованием к вероятностной мере.

Для задачи анализа состояния распределенного комплекса функция переходов (3) описывает динамику исследуемой системы. Переходы могут быть детерминированными, когда действие агента однозначно определяет следующее состояние, и стохастическими, когда результат действия содержит элемент случайности.

Функция наблюдений определяет вероятность получения совместного наблюдения o по

сле перехода в состояние s' в результате выполнения действия a

$$O(o | s', a) = P(o_{t+1} = o | s_{t+1} = s', a_t = a), \quad (5)$$

где $o = (o_1, o_2, \dots, o_n)$ - совместное наблюдение, полученное всеми агентами;

$s' \in S$ - состояние, в которое перешла система;

$a \in A$ - выполненное совместное действие;

Каждый агент α_i получает только свою компоненту наблюдения o_i .

При анализе распределенных комплексов наблюдения представляют собой ответы служб на диагностические запросы, метрики состояния узлов, содержимое конфигурационных файлов и результаты выполнения команд.

Поскольку истинное состояние s недоступно агентам напрямую, система поддерживает состояние веры $b(s)$, представляющее собой распределение вероятностей по множеству возможных состояний. Состояние веры обновляется на каждом шаге по правилу байесовского вывода

$$b'(s') = \eta \cdot O(o | s', a) \cdot \sum_{s' \in S} T(s' | s, a) \cdot b(s), \quad (6)$$

где $b'(s')$ - обновленное состояние веры, то есть вероятность того, что система находится в состоянии s' после получения нового наблюдения;

η - нормализующую константу, обеспечивающую $\sum b'(s') = 1$;

$O(o | s', a)$ - вероятность получить наблюдение o при нахождении в состоянии s' после действия a ;

$T(s' | s, a)$ - вероятность перехода из s в s' при действии a ;
 $b(s)$ - предыдущее состояние веры.
 Формула (6) представляет собой байесовский вывод, позволяющий агентской системе

рекурсивно обновлять представление о состоянии среды на основе накапливаемого опыта взаимодействия. Схема байесовского обновления приведена на рис. 3.



Рисунок 3. Схема байесовского обновления состояния веры
 Figure 3. Bayesian belief state update scheme

Источник: разработано автором
 Source: compiled by the author

Политика агента α_i определяется как отображение истории наблюдений и действий в распределение вероятностей над множеством действий

$$\pi_i : (\Omega_i \times A_i)^* \times \Omega_i \rightarrow \Delta(A_i), \quad (7)$$

где π_i - политику агента α_i , определяющую правило выбора действий;

$(\Omega_i \times A_i)^*$ - множество всех конечных последовательностей пар наблюдений и действий агента i , где символ $*$ обозначает замыкание Клини;

Ω_i - текущее наблюдение агента;

$\Delta(A_i)$ - симплекс распределений вероятностей над множеством действий A_i ;

совместная политика $\pi = (\pi_1, \dots, \pi_n)$ определяет поведение всей многоагентной системы.

Целью является нахождение оптимальной совместной политики, максимизирующей математическое ожидание суммарного дисконтированного вознаграждения

$$\pi^* = \arg \max_{\pi} E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) \mid b_0 \right], \quad (8)$$

где π^* - оптимальная совместная политика;

$\arg \max_{\pi}$ - оператор, возвращающий политику π , при которой достигается максимум выражения;

E_{π} - математическое ожидание при следовании политике π ;

γ^t - коэффициент дисконтирования в степени t ;

$R(s_t, a_t)$ - вознаграждение, полученное в момент t .

Формулировка (8) предполагает, что вознаграждение вычисляется на каждом шаге взаи-

модействия. Схема взаимодействия многоагентной системы с исследуемой средой представлена на рис. 4.

Функция ценности состояния веры определяет ожидаемую суммарную награду при следовании политике π из состояния веры b

$$V^{\pi}(b) = E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) \mid b_0 = b \right], \quad (9)$$

где $V^{\pi}(b)$ - функция ценности состояния веры b при следовании политике π .

Сумма включает дисконтированные награды, полученные по всей траектории от начального состояния веры.

Оптимальная функция ценности удовлетворяет уравнению Беллмана

$$V^*(b) = \max_{a \in A} \left[\sum_{s \in S} b(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o \mid b, a) \cdot V^*(b'_{a,o}) \right], \quad (10)$$

где $V^*(b)$ - оптимальная функция ценности состояния веры b ;

$\max_a$ - оператор максимизации по всем допустимым совместным действиям из множества A ;

$b(s)$ - вероятность нахождения в состоянии s согласно текущему состоянию веры;

$R(s, a)$ - вознаграждение за выполнение действия a в состоянии s ;

γ - коэффициент дисконтирования;

$P(o \mid b, a)$ - вероятность получить наблюдение o при состоянии веры b и выполнении действия a ;

$V^*(b')$ - оптимальная функция ценности для обновленного состояния веры.

Уравнение (10) устанавливает рекурсивную связь между оптимальной ценностью теку-

щего состояния веры и ценностями состояний, достижимых из него за один шаг.

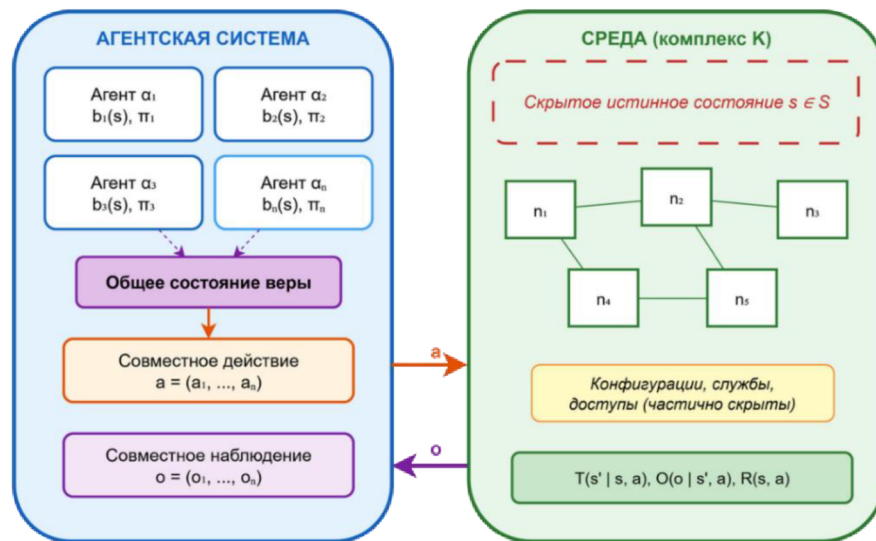


Рисунок 4. Взаимодействие многоагентной системы с исследуемой средой в условиях неопределенности

Figure 4. Multi-agent system interaction with the environment under uncertainty

Источник: разработано автором
Source: compiled by the author

Классический аппарат децентрализованного частично наблюдаемого марковского процесса принятия решений, определяемый кортежем (2), не учитывает ряд существенных особенностей задачи анализа состояния распределенных программно-аппаратных комплексов. Автором предлагается адаптация модели, затрагивающая пространство состояний, пространство действий и механизм оценки результатов.

В классической модели состояние описывает только среду. Автором предлагается композитная структура из двух компонент, позволяющая явно моделировать тот факт, что действия агентов изменяют не только наблюдаемую среду, но и собственное информационное состояние агентской системы

$$S = S_{env} \cdot S_{agent}, \quad (11)$$

где S_{env} - состояние исследуемого комплекса, описываемое компонентами кортежа (1), то есть топологией сети, составом активных служб и текущими конфигурационными параметрами;

S_{agent} - информационное состояние агентской системы, включающее совокупность диагностических сведений, полученных в ходе обследования, и результаты анализа конфигураций.

Отличием от исходной постановки является введение компоненты S_{agent} , отражающей внутреннее информационное состояние агентской системы. Процесс анализа носит преимущественно накопительный характер, поскольку ре-

зультаты обследования фиксируются в хранилище агентской системы.

В классической модели множество доступных действий фиксировано и не зависит от состояния. В задаче анализа распределенных комплексов это допущение не выполняется, поскольку состав доступных действий определяется объемом уже накопленной информации. На начальном этапе агентам доступны только действия начального обследования. По мере накопления сведений о структуре комплекса становятся доступными более специализированные действия, требующие знания о конкретных службах или конфигурациях. Множество доступных действий в момент времени t определяется как функция информационного состояния агентской системы

$$A(t) = f(S_{agent}(t)), \quad (12)$$

где $A(t)$ - множество доступных действий в момент времени t ;

f - функция, определяющую зависимость пространства действий от состояния агентской системы;

$S_{agent}(t)$ - состояние агентской системы в момент t .

Динамика расширения пространства действий отражает процесс постепенного накопления информации о структуре и свойствах комплекса.

Динамика расширения пространства действий в процессе анализа представлена на рис. 5.

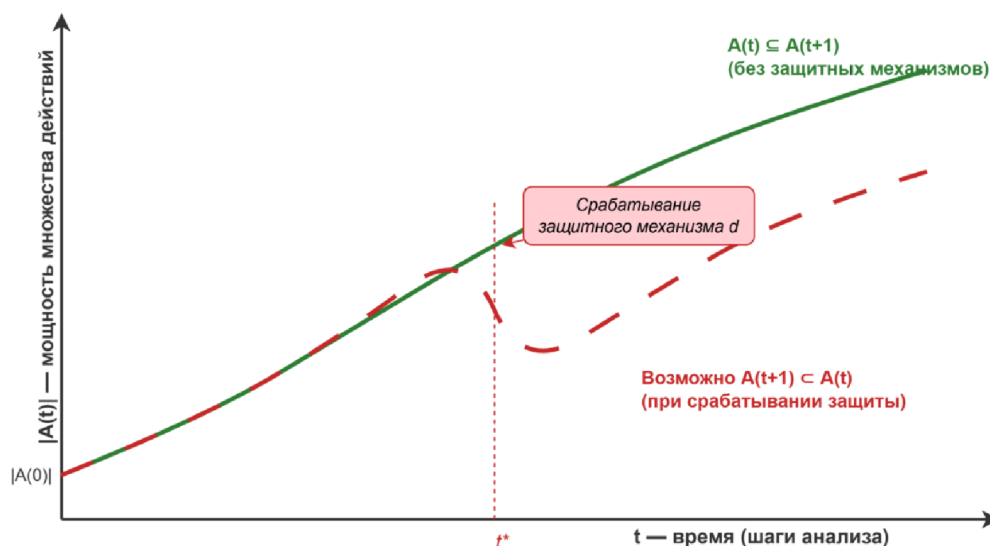


Рисунок 5. Динамика расширения пространства действий в процессе анализа
Figure 5. Dynamics of action space expansion during analysis

Источник: разработано автором
Source: compiled by the author

В исходной модели функция вознаграждения $R(s, a)$ из кортежа (2) вычисляется после каждого действия. В задаче анализа распределенных комплексов такой подход неприменим, поскольку результат анализа характеризуется разнесенностью во времени между действиями агентов и получением полной информации о достижении цели. Промежуточные шаги, такие как отправка диагностического запроса или чтение конфигурационного файла, не обладают самостоятельной ценностью вне их совокупного вклада в итоговый результат анализа.

Последовательность состояний, действий и наблюдений, порождаемая агентской системой в ходе одного сеанса анализа, образует трассу исполнения τ . Трасса фиксирует полную историю взаимодействия агентов с исследуемым комплексом. Для оценки результата анализа вводится функция верификации, отображающая трассу исполнения в степень достижения цели

$$V: T \rightarrow [0, 1], \quad (13)$$

где T - множество трасс исполнения;

$V(\tau) = q$ - степень достижения цели по результатам трассы τ ;

Значение $q=1$ соответствует полному достижению цели, $q=0$ соответствует полному отсутствию прогресса. Промежуточные значения соответствуют частичному результату. Верификация осуществляется однократно, после того как система завершает сеанс анализа.

Верифицирующим механизмом может выступать эталонная модель состояния, экспертная система или среда с автоматической проверкой достижения целевых условий. Аналогичный подход к верификации результатов автоматизированного анализа использован в работе [18] применительно к задаче анализа поведения пользователей.

Автором проведено сравнение классической и авторской моделей. Результаты систематизированы в табл. 2.

Обобщающая схема модификации модели, отражающая взаимосвязь между всеми компонентами, представлена на рис. 6.

Таблица 2. Сравнение классической и авторской моделей
Table 2. Comparison of classical and author's models

Компонент	Классическая модель	Авторская модель
Пространство состояний	S — только среда	$S = S_{env} \times S_{agent}$, формула (11)
Пространство действий	A — фиксированное	$A(t) = f(S_{agent}(t))$, формула (12)
Оценка результата	$R(s, a)$ — после каждого действия	$V(\tau) \in [0, 1]$, формула (13)

Источник: разработано автором
Source: compiled by the author



Рисунок 6. Авторская модификация модели децентрализованного частично наблюдаемого марковского процесса принятия решений

Figure 6. Author's modification of the decentralized partially observable Markov decision process model

Источник: разработано автором
Source: compiled by the author

Предложенная автором модификация затрагивает три компонента исходного кортежа (2). Пространство состояний S, описывающее в классической постановке только среду, расширено до декартова произведения $S = S_{env} \times S_{agent}$ согласно формуле (11), что позволяет явно учитывать информационное состояние агентской системы. Пространство действий A, являющееся в классической модели статическим, заменено динамическим пространством $A(t) = f(S_{agent}(t))$ согласно формуле (12), зависящим от текущего информационного состояния. Функция вознаграждения $R(s, a)$, вычисляемая пошагово, заменена функцией верификации $V(t) \in [0, 1]$ согласно формуле (13), оценивающей полную трассу исполнения.

Задача нахождения оптимальной политики в рамках децентрализованного частично наблюдаемого марковского процесса принятия решений является вычислительно сложной. Время точного решения растет экспоненциально с ростом пространства состояний. Пространство состояний распределенного комплекса растет экспоненциально с увеличением числа узлов. Автором предлагается следующая оценка мощности пространства состояний для комплекса с N узлами и k бинарными признаками состояния каждого узла

$$|S_{env}| = 2^{k \cdot N}, \quad (14)$$

где N - число узлов комплекса;

k - число бинарных признаков, описывающих состояние каждого узла, таких как обнаружен, идентифицирован, диагностирован, инвентаризован.

Произведение $2^{k \cdot N}$ дает общее число различных состояний среды для комплекса заданного размера.

С учетом введенной автором компоненты S_{agent} в формуле (11) общая мощность композиционного пространства состояний составляет

$$|S| = |S_{env}| \cdot |S_{agent}| = 2^{k \cdot N} \cdot 2^{m \cdot N} = 2^{(k+m) \cdot N}, \quad (15)$$

где m - число признаков состояния агентской системы, приходящееся на каждый узел, таких как наличие сведений об узле, наличие доступа к узлу, наличие собранных диагностических данных.

При k = 4 и m = 3 признаках и N = 10 узлах число возможных состояний по формуле (15) достигает 2^{70} , что превышает 10^{21} . Экспоненциальный рост мощности пространства состояний подтверждает сверхполиномиальную сложность задачи оптимизации политики. Фрагмент графа состояний для минимального комплекса из трех узлов представлен на рис. 7.

Даже для минимального комплекса число состояний и переходов оказывается значительным, как видно из рис. 7. Для оценки масштабируемости модели построена зависимость вычислительной сложности от числа узлов комплекса (рис.8).

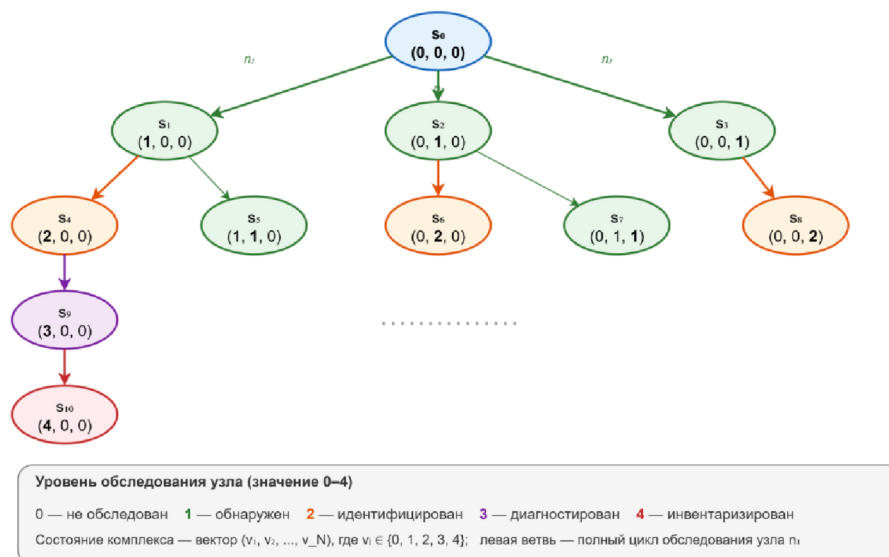


Рисунок 7. Фрагмент графа состояний задачи анализа распределенного комплекса
 Figure 7. Fragment of the state graph for the distributed complex analysis problem

Источник: разработано автором
 Source: compiled by the author

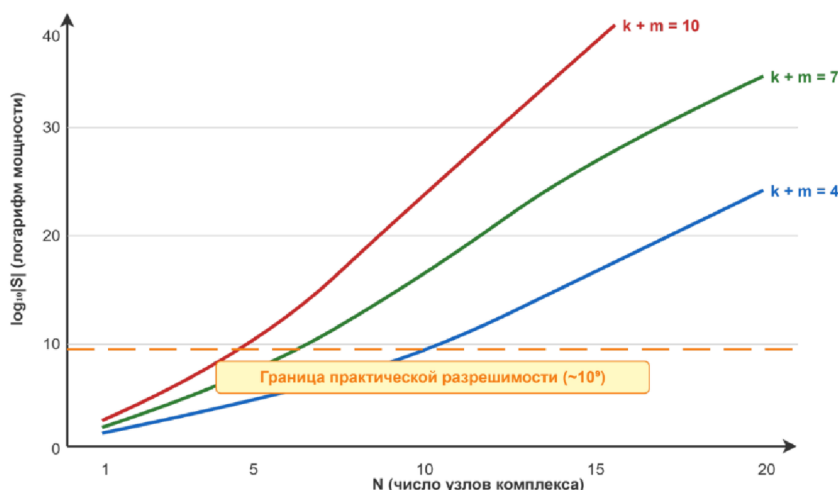


Рисунок 8. Зависимость вычислительной сложности от числа узлов комплекса
 Figure 8. Computational complexity dependence on the number of complex nodes

Источник: разработано автором
 Source: compiled by the author

Для реальных инфраструктур с десятками и сотнями узлов точное решение оптимизационной задачи практически невозможно. Сформулированная задача является оптимизационной. Варьируемыми переменными выступают политики агентов $\pi = (\pi_1, \dots, \pi_n)$, определяемые формулой (7). Критерием оптимальности является максимизация ожидаемого значения функции верификации $V(\tau)$, определенной формулой (13). Вычислительная сложность задачи обосновывает необходимость применения приближенных методов решения. Точные алгоритмы динамического программирования на основе уравнения (10) для данного класса задач обладают сверхэкспоненциальной вычислительной сложностью и применимы лишь для задач малой размерности [3, 17].

ДИСКУССИЯ

Предложенная автором модель развивает классический аппарат децентрализованного частично наблюдаемого марковского процесса принятия решений в нескольких направлениях. Введение композитного пространства состояний по формуле (11) позволяет явно моделировать накопление информации агентской системой, что невозможно в рамках классической постановки, где состояние описывает исключительно среду. Без этого расширения информационное состояние агентов моделировалось бы неявно через историю наблюдений, что затрудняет анализ свойств модели и оценку сложности.

Динамическое пространство действий отражает естественную зависимость набора до-

ступных диагностических операций от объема уже накопленных сведений [19]. Этот аспект специфичен для задач анализа, где первоначально агентам доступны только операции обнаружения, а более глубокие диагностические действия становятся возможными после идентификации конкретных служб и получения необходимых учетных данных. Аналогичная зависимость доступных операций от текущего информационного состояния наблюдается в задачах криптографического анализа распределенных систем [20]. Классическая модель с фиксированным пространством действий вынуждала бы включать все теоретически возможные действия с самого начала, что неоправданно увеличивает размерность задачи оптимизации.

Замена пошагового вознаграждения функцией верификации является существенным отступлением от классической формулы (8). Пошаговое вознаграждение предполагает возможность немедленной оценки качества каждого действия, что не выполняется в задаче анализа, где отдельный диагностический запрос не обладает самостоятельной ценностью вне совокупности результатов обследования. Верификация по трассе позволяет оценивать качество всего сеанса анализа как единого целого, что более адекватно отражает природу задачи.

Полученная оценка мощности пространства состояний имеет принципиальное значение для выбора методов решения. При типичных значениях параметров пространство состояний для комплекса из 10 узлов превышает 10^{21} . Задача нахождения оптимальной совместной политики в рамках децентрализованного частично наблюдаемого марковского процесса принадлежит классу NEXP-полных задач [3], что делает точное решение невозможным для реальных комплексов. Этот результат обосновывает необходимость иерархической декомпозиции задачи и применения приближенных методов, которые являются предметом отдельных исследований автора.

Вместе с тем необходимо указать на ограничения модели. Предполагается дискретность пространств состояний, действий и наблюдений, тогда как в реальных комплексах некоторые параметры являются непрерывными. Функции переходов и наблюдений считаются известными или оцениваемыми, что требует предварительного моделирования поведения среды. Модель определяет оптимизационную задачу, но не устанавливает архитектуру системы, решающей эту задачу, и не содержит экспериментальной валидации.

ЗАКЛЮЧЕНИЕ

Формализация задачи автоматизированного анализа состояния распределенных программно-аппаратных комплексов потребовала выхода за рамки классического децентрализованного частично наблюдаемого марковского процесса принятия решений. Три модификации, внесенные в исходную модель, отражают специфику, которая не укладывается в стандартные допущения.

Композитное пространство состояний позволяет отслеживать не только изменения среды, но и накопление диагностических сведений агентской системой. Динамическое пространство действий устраняет искусственное расширение задачи оптимизации за счет исключения недоступных на ранних этапах операций. Функция верификации переносит оценку качества с отдельных шагов на трассу исполнения целиком, что адекватнее природе обследования.

Оценка мощности пространства состояний выявила принципиальное ограничение. Уже при 10 узлах число состояний превышает 10^{21} , что исключает точное решение задачи оптимизации политики для реальных инфраструктур. Следующим шагом является разработка иерархической архитектуры многоагентной системы, способной декомпозировать задачу на подзадачи управляемой размерности.

*Авторы заявляют об отсутствии
конфликта интересов.*

The authors declare no conflict of interest.

ЛИТЕРАТУРА

1. Астраханцев Р.Г., Бобков С.П. Методологические основы построения многоагентной интеллектуальной системы. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2026. № 1 (67). С. 32–40.
2. Астраханцева И.А., Астраханцев Р.Г., Бобков С.П. Агентные интеллектуальные системы с точки зрения системного анализа. *Современные наукоемкие технологии. Региональное приложение*. 2025. № 4(84). С. 101–109.
3. Bernstein D.S., Givan R., Immerman N., Zilberstein S. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*. 2002. Vol. 27. N 4. P. 819–840.
4. Oliehoek F.A., Spaan M.T.J., Vlassis N. A Concise Introduction to Decentralized POMDPs. *Proceedings of AAMAS*. 2013. P. 1257–1264.
5. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.

REFERENCES

1. Astrakhantsev R.G., Bobkov S.P. Foundations of multi-agent intelligent system design. *Ivecofin*. 2026. N 1(6). P. 32–40. (in Russian).
2. Astrakhantseva I.A., Astrakhantsev R.G., Bobkov S.P. Agent-based intelligent systems from the point of view of system analysis. *Modern High Technologies. Regional Application*. 2025. N 4(84). P. 101–109. (in Russian).
3. Bernstein D.S., Givan R., Immerman N., Zilberstein S. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*. 2002. Vol. 27. N 4. P. 819–840.
4. Oliehoek F.A., Spaan M.T.J., Vlassis N. A Concise Introduction to Decentralized POMDPs. *Proceedings of AAMAS*. 2013. P. 1257–1264.
5. Jennings N.R., Wooldridge M. Agent-Oriented Software Engineering. *Artificial Intelligence*. 2000. Vol. 117. P. 277–296.

6. **Принев М.А., Леденева Т.М., Гаршина В.В.** Много-агентные системы: обзор современных подходов к моделированию и проектированию (Часть 2). *Вестник ВГУ. Серия: Системный анализ и информационные технологии*. 2024. № 4. С. 167–190.
7. **Астраханцева Р.Г., Брюшинин А.О., Астраханцев Р.Г., Котенев Т.Е.** Система фильтрации ложноположительных результатов с использованием методов машинного обучения для оценки принадлежности цифровых активов компаний. *Современные наукоемкие технологии. Региональное приложение*. 2025. № 3(83). С. 57–67.
8. **Самсонова А.И., Суворов И.А., Астраханцев Р.Г.** Интеллектуальная система анализа пользовательской активности в защищенных операционных средах. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2025. № 1(63). С. 79–85.
9. Bellman R. *Dynamic Programming*. Princeton: Princeton University Press. 1957. 339 p.
10. **Kaelbling L.P., Littman M.L., Cassandra A.R.** Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*. 1998. Vol. 101. N 1–2. P. 99–134.
11. **Dorri A., Kanhere S.S., Jurdak R.** Multi-Agent Systems: A Survey. *IEEE Access*. 2018. Vol. 6. P. 28573–28593.
12. **Busoniu L.A., Babuska R., De Schutter B.** Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*. 2008. Vol. 38. N 2. P. 156–172.
13. **Нигматуллин Р.Р.** Системный анализ как направление научных исследований. *Вестник Российской академии наук*. 2007. Т. 77. № 2. С. 101–107.
14. **Бобков С.П., Астраханцева И.А., Астраханцев Р.Г., Шутов Д.А., Гушин А.А., Бобкова Е.С.** Использование дискретного вероятностного подхода для моделирования проточных трубчатых реакторов. *Известия высших учебных заведений. Серия Химия и химическая технология*. 2025. Т. 68. № 2. С. 96–101.
15. **Зимнуров М.Ф., Астраханцева И.А., Грименицкий П.Н.** Системный анализ и оптимизация количественных показателей эффективности в технологических проектах на основе гибких методологий. *Современные наукоемкие технологии. Региональное приложение*. 2023. № 3(75). С. 61–68.
16. ГОСТ Р 59277–2020. Информационные технологии. Многоагентные системы. Термины и определения. Введ. 2020-09-01. М.: Стандартинформ. 2020. 12 с.
17. **Koops W., Junges S., Jansen N.** Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of ICAPS 2024*. 2024. P. 745–753.
18. **Астраханцева И.А., Горев С.В., Астраханцев Р.Г.** Системный подход к анализу фрактальной природы сложных технических систем. *Известия высших учебных заведений. Серия «Экономика, финансы и управление производством» [Ивэкофин]*. 2023. № 3(57). С. 89–97.
19. **Астраханцева И.А., Бобков С.П.** Имитационная модель проточного трубчатого реактора. *Моделирование систем и процессов*. 2024. Т. 17. № 4. С. 16–22.
20. **Astrakhantsev R., Chuhno A., Dmukh A., Kurochkin A., Astrakhantseva I.** Differences with high probability and impossible differentials for the KB-256 cipher. *Journal of Computer Virology and Hacking Techniques*. 2024. Vol. 20. N 3. P. 525–531.
6. **Prinev M.A., Ledeneva T.M., Garshina V.V.** Multi-agent systems: a review of modern approaches to modeling and projecting (Part 2). *Vestnik VGU. Systems Analysis and Information Technologies*. 2024. N 4. P. 167–190. (in Russian).
7. **Astrakhantseva I.A., Bryushinin A.O., Astrakhantsev R.G., Kotenev T.E.** ML-based filtering system for false-positive OSINT results in corporate asset attribution. *Modern High Technologies. Regional Application*. 2025. N 3(83). P. 57–67. (in Russian).
8. **Samsonova A.I., Suvorov I.A., Astrakhantsev R.G.** Intelligent system for analyzing user activities in secure operating environments. *Ivecofin*. 2025. N 1(63). P. 79–85. (in Russian).
9. Bellman R. *Dynamic Programming*. Princeton: Princeton University Press. 1957. 339 p.
10. **Kaelbling L.P., Littman M.L., Cassandra A.R.** Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*. 1998. Vol. 101. N 1–2. P. 99–134.
11. **Dorri A., Kanhere S.S., Jurdak R.** Multi-Agent Systems: A Survey. *IEEE Access*. 2018. Vol. 6. P. 28573–28593.
12. **Busoniu L.A., Babuska R., De Schutter B.** Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*. 2008. Vol. 38. N 2. P. 156–172.
13. **Nigmatullin R.R.** Systems analysis as a research direction. *Vestnik Rossiiskoi Akademii Nauk*. 2007. Vol. 77. N 2. P. 101–107. (in Russian).
14. **Bobkov S.P., Astrakhantseva I.A., Astrakhantsev R.G., Shutov D.A., Gushchin A.A., Bobkova E.S.** Using a discrete probabilistic approach for simulating flow tubular reactors. *ChemChemTech*. 2025. Vol. 68. N 2. P. 96–101. (in Russian).
15. **Zimnurov M.F., Astrakhantseva I.A., Grimenitsky P.N.** System analysis and optimization of quantitative performance indicators in technological projects based on flexible methodologies. *Modern High Technologies. Regional Application*. 2023. N 3(75). P. 61–68. (in Russian).
16. GOST R 59277–2020. Information Technologies. Multi-Agent Systems. Terms and Definitions. Moscow: Standartinform. 2020. 12 p. (in Russian).
17. **Koops W., Junges S., Jansen N.** Approximate Dec-POMDP Solving Using Multi-Agent A*. *Proceedings of ICAPS 2024*. 2024. P. 745–753.
18. **Astrakhantseva I.A., Gorev S.V., Astrakhantsev R.G.** Systems approach to the analysis of the fractal nature of complex technical systems. *Ivecofin*. 2023. N 3(57). P. 89–97. (in Russian).
19. **Astrakhantseva I.A., Bobkov S.P.** Simulation model of a tubular flow reactor. *Modeling of Systems and Processes*. 2024. Vol. 17. N 4. P. 16–22. (in Russian).
20. **Astrakhantsev R., Chuhno A., Dmukh A., Kurochkin A., Astrakhantseva I.** Differences with high probability and impossible differentials for the KB-256 cipher. *Journal of Computer Virology and Hacking Techniques*. 2024. Vol. 20. N 3. P. 525–531.

Поступила в редакцию 20.12.2025
Принята к опубликованию 06.01.2026

Received 20.12.2025
Accepted 06.01.2026