

Федеральное государственное бюджетное образовательное учреждение высшего
образования «Ивановский государственный химико-технологический
университет»

На правах рукописи



Астраханцев Роман Геннадьевич

**Модели и алгоритмы оценки работоспособности распределенных
программно-аппаратных комплексов в условиях неопределенности
и частичной наблюдаемости цифровой среды**

2.3.1 – Системный анализ, управление и обработка информации, статистика
(технические науки)

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Бобков Сергей Петрович

Иваново – 2026

Оглавление

Введение	4
1 Современное состояние в области многоагентных интеллектуальных систем оценки работоспособности распределенных программно-аппаратных комплексов	14
1.1 Многоагентный подход к оценке работоспособности распределенных программно-аппаратных комплексов	14
1.2 Существующие подходы к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов	25
1.3 Постановка задач диссертационного исследования	36
Выводы по главе 1	46
2 Модели многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов	49
2.1 Формальная постановка задачи оценки работоспособности распределенных программно-аппаратных комплексов	49
2.2 Модель многоагентной оценки работоспособности распределенных программно-аппаратных комплексов	63
2.3 Модель архитектуры многоагентной интеллектуальной системы	87
Выводы по главе 2	103
3 Алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов	106
3.1 Метод кросс-уровневого обучения в среде с верифицируемой обратной связью	106
3.2 Оценка завершенной трассы по критериальной модели качества функционирования	119
3.3 Экспериментальные исследования в испытательной среде с верифицируемой обратной связью	138

3.4 Опытная эксплуатация на действующих распределенных программно-аппаратных комплексах	147
Выводы по главе 3.....	157
Заключение	161
Список литературы	167
Приложение А Акты об использовании и внедрении результатов диссертационного исследования	183

Введение

Актуальность темы исследования. Распределенные программно-аппаратные комплексы (ПАК) составляют основу информационной инфраструктуры предприятий и организаций. Корпоративные сети, облачные платформы и промышленные системы управления объединяют от десятков до тысяч территориально рассредоточенных вычислительных узлов с разнородными программными службами и множеством конфигурационных параметров. Работоспособность такого комплекса определяется состоянием его узлов, соединений и служб и оценивается по контролируемым показателям, среди которых пропускная способность соединений, время отклика служб, загрузка узлов, интенсивность обработки заявок и доля задач, решаемых за директивное время. Полное обследование одного комплекса требует от сорока до двухсот часов работы квалифицированного специалиста, тогда как число специалистов не увеличивается пропорционально росту потребности в обследованиях. Объект оценки характеризуется такими свойствами, как территориальная распределенность с заранее неизвестной топологией соединений, гетерогенность аппаратной базы и программных служб, неполная наблюдаемость, при которой сведения о цифровой среде раскрываются поступательно в ходе диагностических действий, и динамичность состояния под действием штатного функционирования, операций администраторов и внешних факторов.

Существующие средства автоматизации оценки работоспособности обладают существенными ограничениями. Сценарные средства выполняют заранее определенные последовательности проверок и не перестраивают логику поведения при встрече с нестандартной конфигурацией. Одноагентные интеллектуальные системы на основе больших языковых моделей ограничены объемом сведений, одновременно доступных единственной модели, и требуют участия специалиста на каждом шаге принятия решения. Многоагентные платформы выполняют полный цикл обследования автономно, однако закрепляют состав агентов

при проектировании, начинают каждый сеанс без учета опыта предыдущих и не сверяют результат с истинным состоянием комплекса.

Возникает противоречие между потребностью в автономной оценке работоспособности заранее неизвестного комплекса, которая совершенствовалась бы от сеанса к сеансу без участия эксперта, и возможностями существующих средств, структура и поведение которых закрепляются при разработке. Таким образом, актуальной научной задачей является разработка моделей и алгоритмов оценки работоспособности распределенных ПАК в условиях неопределенности и частичной наблюдаемости цифровой среды. Не разработаны математическая модель многоагентной оценки работоспособности с композитным пространством состояний и оценкой завершенной трассы, параметрическая модель архитектуры многоагентной интеллектуальной системы со структурой, порождаемой решениями самих агентов, а также алгоритм кросс-уровневого обучения такой системы по собственным завершенным сеансам без экспертной разметки.

Степень разработанности темы исследования. В развитие теории интеллектуальных агентов и многоагентных систем существенный вклад внесли М. Вулдридж, Н. Дженнингс, С. Рассел, П. Норвиг, Я. Шохам, К. Лейтон-Браун. В отечественной науке теория многоагентных систем развивалась в работах В. И. Городецкого, В. Б. Тарасова, Г. В. Рыбиной, С. Н. Васильева.

Методы анализа состояния распределенных систем и графовые модели зависимостей между их компонентами разрабатывались Б. Шнайером, Р. Андерсоном, И. В. Котенко, А. А. Чечулиным, И. Б. Саенко. Математический аппарат децентрализованных частично наблюдаемых марковских процессов принятия решений получил развитие в трудах Д. Бернштейна, Ф. Олихука, К. Амато, Л. Кэлблинг, а его применение к последовательному обследованию сетевых инфраструктур при неопределенности результатов действий обосновано К. Сарро и Э. Милингом.

Обучение с подкреплением систематизировано в работах Р. Саттона и Э. Барто, поиск политики отбором элит по итогу заверщенного сеанса развит Д. Сильвером и Б. Ромера-Паредесом. Применение больших языковых моделей

в качестве ядра агентов и координация агентов через ролевые инструкции исследованы Хонгом и др. и Конгом и др., многоуровневые системы обследования сетевой инфраструктуры на основе глубокого обучения с подкреплением предложены Ли и др., а среды обучения агентов в моделях сетевых инфраструктур предложены Дж. Шварцем и Х. Курниавати, М. Стенденом и др.

Вместе с тем перечисленные работы рассматривают формальные модели принятия решений, архитектуры многоагентных систем и методы обучения агентов изолированно, без объединения в единой системе оценки работоспособности заранее неизвестного комплекса. Аппарат децентрализованных частично наблюдаемых марковских процессов описывает состояние только исследуемого объекта при фиксированном составе действий и пошаговом вознаграждении, тогда как при оценке работоспособности состав допустимых действий определяется накопленными сведениями, а качество обследования устанавливается лишь по завершенной трассе. Многоагентные платформы закрепляют состав агентов и связи между ними при проектировании, а методы поиска политики отбором элит рассчитаны на одного агента с единой политикой и не согласуют политики уровней иерархии. Указанные задачи взаимосвязаны и образуют единый комплекс, решение которого составляет содержание настоящей диссертации.

Объектом исследования служат распределенные программно-аппаратные комплексы.

Предметом исследования выступают модели и алгоритмы оценки работоспособности распределенных программно-аппаратных комплексов.

Целью исследования является разработка моделей и алгоритмов оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды, обеспечивающих автономное обследование заранее неизвестного комплекса многоагентной интеллектуальной системой, обучающейся на собственных завершенных сеансах без экспертной разметки.

Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ существующих подходов к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов и выявить их ограничения, препятствующие автономному обследованию заранее неизвестного комплекса.

2. Разработать формальную математическую модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений, учитывающую неполноту сведений о цифровой среде, множественность агентов, зависимость допустимых действий от накопленных сведений и оценку итога обследования по завершенной трассе исполнения.

3. Разработать параметрическую модель архитектуры многоагентной интеллектуальной системы оценки работоспособности с трехуровневой иерархией агентов на основе единой большой языковой модели, структура которой порождается решениями самих агентов под состав исследуемого комплекса.

4. Разработать алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности в среде с верифицируемой обратной связью, совместно улучшающий политики всех уровней иерархии по оценкам собственных завершенных сеансов без экспертной разметки.

5. Выполнить программную реализацию многоагентной интеллектуальной системы оценки работоспособности и провести экспериментальные исследования в испытательной среде с верифицируемой обратной связью и опытную эксплуатацию на действующих распределенных программно-аппаратных комплексах, включая сопоставление со сторонними системами анализа распределенных комплексов и готовыми большими языковыми моделями.

Методология и методы исследования. Для решения поставленных задач применены методы системного анализа, теории надежности технических систем, теории вероятностей и математической статистики, теории марковских процессов принятия решений, теории многоагентных систем, метод поиска политики отбором элит по оценке заверщенного сеанса, методы планирования экспериментов и статистической обработки результатов. Для экспериментальной проверки

разработанных моделей и алгоритма использован авторский программный комплекс, включающий агентскую систему, испытательную среду с верифицируемой обратной связью и модуль кросс-уровневого обучения.

Научная новизна результатов диссертационного исследования заключается в разработке моделей и алгоритмов оценки работоспособности распределенных программно-аппаратных комплексов многоагентной интеллектуальной системой на основе большой языковой модели, обеспечивающих автономное построение оценки работоспособности заранее неизвестного комплекса посредством формализации процесса многоагентной оценки, порождения структуры системы решениями самих агентов и кросс-уровневого обучения на собственных завершенных сеансах без экспертной разметки в условиях неопределенности и частичной наблюдаемости цифровой среды.

Научная новизна диссертационного исследования состоит в следующем.

1. Разработана математическая модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений, отличающаяся композитным пространством состояний, объединяющим состояние комплекса с накопленными сведениями агентской системы, зависимостью множества допустимых действий от накопленных сведений и оценкой завершенной трассы исполнения вместо пошагового вознаграждения, позволяющая формализовать задачу многоагентной оценки работоспособности цифровой среды (п. 2 паспорта специальности).

2. Разработана параметрическая модель архитектуры многоагентной интеллектуальной системы оценки работоспособности с трехуровневой иерархией агентов на основе единой большой языковой модели, отличающаяся древовидным динамическим графом делегирования, вершины и ребра которого порождаются решениями самих агентов в процессе функционирования под конкретный исследуемый комплекс, обеспечивающая масштабирование системы под состав и размер комплекса при постоянной глубине подчинения (п. 9 паспорта специальности).

3. Разработан алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности в среде с верифицируемой обратной связью, отличающийся единым целевым сигналом в виде оценки завершенной трассы, формируемой критериальной моделью качества функционирования с весовыми коэффициентами, изменяемыми по итерациям обучения, формированием элитного множества трасс с адаптивным пороговым значением и разнообразием трасс, создаваемым температурой генерации и несколькими сеансами на каждом экземпляре комплекса, обеспечивающий совместную оптимизацию политик всех уровней иерархии по собственным завершенным сеансам без экспертной разметки (п. 5 паспорта специальности).

На защиту выносятся следующие положения и результаты, обладающие научной новизной.

1. Разработанная математическая модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений, включающая композитное пространство состояний, объединяющее состояние комплекса с накопленными сведениями агентской системы, множество допустимых действий, зависящее от накопленных сведений, и оценку завершенной трассы исполнения вместо пошагового вознаграждения, формализует задачу многоагентной оценки работоспособности цифровой среды при неполных сведениях о ней с сохранением марковости и однородности процесса, существования оптимальной совместной политики и совместимости оценки завершенной трассы с уравнением оптимальности. В опытной эксплуатации на действующих распределенных программно-аппаратных комплексах время от регистрации нарушения работоспособности до установления его источника по заключению системы сократилось в 1,5 раза.

2. Разработанная параметрическая модель архитектуры многоагентной интеллектуальной системы оценки работоспособности с трехуровневой иерархией агентов на основе единой большой языковой модели, включающая древовидный динамический граф делегирования, вершины и ребра которого порождаются

решениями самих агентов в процессе функционирования под конкретный исследуемый комплекс, обеспечивает масштабирование системы под состав и размер комплекса при постоянной глубине подчинения. Система обследует комплексы величиной от 6 до 30 узлов без изменения своей структуры и снижает трудоемкость обследования одного комплекса с 62 до 14 часов инженера.

3. Разработанный алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности в среде с верифицируемой обратной связью, включающий единый целевой сигнал в виде оценки завершенной трассы, формируемой критериальной моделью качества функционирования с весовыми коэффициентами, изменяемыми по итерациям обучения, формирование элитного множества трасс с адаптивным пороговым значением и разнообразие трасс, создаваемое температурой генерации и несколькими сеансами на каждом экземпляре комплекса, обеспечивает совместную оптимизацию политик всех уровней иерархии по собственным завершенным сеансам без экспертной разметки. После обучения на собственных сеансах без участия специалиста ошибка оценки работоспособности на испытательных стендах снизилась в 1,3 раза, а на действующих комплексах доля верно установленных источников нарушения работоспособности выросла с 60 до 75 %.

Соответствие паспорту специальности. Диссертационная работа соответствует паспорту научной специальности 2.3.1 «Системный анализ, управление и обработка информации, статистика». Содержание работы соответствует следующим пунктам паспорта специальности. Пункт 2 — формализация и постановка задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта (математическая модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений); пункт 5 — разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта (алгоритм кросс-уровневого обучения многоагентной

интеллектуальной системы оценки работоспособности); пункт 9 — разработка проблемно-ориентированных систем управления, принятия решений и оптимизации технических объектов (параметрическая модель архитектуры многоагентной интеллектуальной системы оценки работоспособности).

Теоретическая значимость работы заключается в том, что разработанные модели и алгоритм вносят вклад в развитие теории многоагентных интеллектуальных систем и методов системного анализа распределенных программно-аппаратных комплексов. Математическая модель с композитным пространством состояний, множеством допустимых действий, зависящим от накопленных сведений, и оценкой завершенной трассы исполнения расширяет аппарат формализации задач многоагентной оценки при неполных сведениях о среде. Показано, что модифицированный процесс сохраняет марковость и однородность, а оптимальная совместная политика существует. Параметрическая модель архитектуры с динамическим графом делегирования развивает подходы к проектированию многоагентных систем, структура которых порождается в ходе функционирования без закрепления при проектировании. Алгоритм кросс-уровневого обучения расширяет класс методов поиска политики отбором элит на иерархические системы с единой обучаемой моделью. Разработанные модели и алгоритм могут быть адаптированы для задач многоагентного обследования в смежных предметных областях, включая мониторинг промышленных систем управления и аудит облачных инфраструктур.

Практическая значимость работы определяется внедрением результатов в системы автоматизированной оценки работоспособности распределенных программно-аппаратных комплексов. Разработанные модели и алгоритм реализованы в двух программных продуктах, программном комплексе «Модели и методы автоматизированного анализа распределенных программно-аппаратных комплексов на основе многоагентной интеллектуальной системы» (свидетельство № 2026680346) и программе «Поиск невозможных дифференциалов для блочного шифра КБ256-3» (свидетельство № 2024614939). Программный комплекс выполняет полный цикл обследования от сбора сведений до формирования

заклучения без поэтапного участия специалиста. В опытной эксплуатации на действующих комплексах время от регистрации нарушения работоспособности до установления его источника сократилось в 1,5 раза, трудоемкость обследования одного комплекса снизилась с 62 до 14 часов инженера, а после обучения системы на собственных сеансах ошибка оценки работоспособности на испытательных стендах снизилась в 1,3 раза и доля верно установленных источников нарушения работоспособности выросла с 60 до 75 %. Результаты внедрены в ООО «Интеркомтел» (г. Иваново), ООО «Профи-Групп» (г. Иваново) и ООО «Национальный оператор парковок» (г. Москва), что подтверждено соответствующими актами. Результаты диссертационного исследования используются в учебном процессе Ивановского государственного химико-технологического университета при преподавании дисциплин «Архитектура информационных систем», «Технология обработки информации», «Информационные технологии», «Модели информационных процессов и систем», что подтверждено актом внедрения.

Достоверность и обоснованность научных положений и выводов обеспечивается корректностью применения математического аппарата децентрализованных частично наблюдаемых марковских процессов принятия решений и доказательствами свойств модели, а также экспериментальной проверкой на испытательной среде, где истинная работоспособность каждого стенда известна по построению, и опытной эксплуатацией на 3 действующих комплексах. Испытания проведены на 3 стендах от 6 до 18 вычислительных узлов, обучение повторено 5 раз независимо по 30 итераций каждое, качество оценки проверено на комплексах, не участвовавших в обучении, и сопоставлено с готовыми большими языковыми моделями без дообучения в той же системе.

Апробация результатов исследования. Положения и выводы диссертации прошли апробацию на международных и всероссийских научно-практических конференциях. Всероссийская школа-конференция молодых ученых «Дни науки в ИГХТУ» (22–26 апреля 2024 г., 20–24 апреля 2026 г., ИГХТУ); V и VI Всероссийские научно-практические конференции «Актуальные проблемы

функционирования финансовой системы России» (27 февраля 2025 г., 26 февраля 2026 г., Финансовый университет); Международная научно-практическая конференция «Теория и практика применения вычислительных и информационных технологий в отраслевых системах» (COMSYSTEM 2026) (13–14 марта 2026 г., МТУСИ); III Всероссийская научно-техническая конференция «Вопросы обеспечения безопасности в киберпространстве» (6–7 мая 2026 г., Дагестанский государственный технический университет); Всероссийская научно-техническая конференция «Информационные технологии в науке, промышленности и образовании» (28–29 мая 2026 г., ИжГТУ имени М. Т. Калашникова).

Публикации. По теме диссертационного исследования опубликовано 13 научных работ, в том числе 3 в журналах, индексируемых в международных базах данных Scopus и Web of Science, и 10 в журналах, рекомендованных ВАК РФ (общим объемом 6,5 печатных листов); получено 2 свидетельства о государственной регистрации программ для ЭВМ.

Личный вклад. Все результаты, изложенные в диссертации, получены автором лично. Постановка цели и задач, обсуждение планов исследований и полученных результатов выполнены совместно с научным руководителем.

Структура и объем диссертации. Диссертация состоит из введения, трех глав с выводами, заключения и списка литературы. Работа содержит 182 страницы основного текста, 24 рисунка, 11 таблиц, 1 приложение. Список литературы включает 137 наименований.

1 Современное состояние в области многоагентных интеллектуальных систем оценки работоспособности распределенных программно-аппаратных комплексов

1.1 Многоагентный подход к оценке работоспособности распределенных программно-аппаратных комплексов

Распределенные программно-аппаратные комплексы составляют технологическую основу современной информационной инфраструктуры. Корпоративные сети, облачные платформы, промышленные системы управления, телекоммуникационные и энергетические инфраструктуры, а также системы распределенных реестров [3] объединены общим свойством. Все они представляют собой совокупности территориально рассредоточенных вычислительных узлов, связанных сетевой инфраструктурой, на которых функционируют разнородные программные службы, управляемые множеством конфигурационных параметров [21; 61].

Понятие программно-аппаратного комплекса в научной и инженерной литературе описывает интегрированную совокупность технических средств и программного обеспечения, функционирующих как единое целое для решения определенного класса задач [27]. В отечественных стандартах по системной и программной инженерии такая система рассматривается через призму качества взаимодействия аппаратных и программных компонентов, определяемого метриками надежности, безопасности и адаптивности [21]. Стандарт ГОСТ Р 59898-2021 определяет понятийный аппарат систем искусственного интеллекта, в рамках которого сложная техническая система представляется объектом, допускающим автоматизированную оценку работоспособности с применением интеллектуальных методов [20].

В настоящем исследовании под распределенным программно-аппаратным комплексом понимается сложная техническая система, состоящая из

территориально рассредоточенных вычислительных узлов, связанных сетевой инфраструктурой, на которых функционируют разнородные программные службы, управляемые множеством конфигурационных параметров.

Подобная система обладает четырьмя существенными свойствами, определяющими специфику автоматизированной оценки ее работоспособности.

Территориальная распределенность выражается в том, что вычислительные узлы рассредоточены в сетевой инфраструктуре и связаны множеством соединений, актуальное состояние которых (доступность, задержки, пропускная способность) заранее не известно системе оценки в полном объеме и уточняется в ходе диагностики. Количество узлов и связей между ними может варьироваться от десятков в малых корпоративных средах до тысяч в крупных распределенных инфраструктурах.

Гетерогенность проявляется в неоднородности аппаратной базы узлов, протоколов взаимодействия, а также типов, версий и конфигураций программных служб. На одном узле могут одновременно функционировать десятки служб с различными наборами технологий и интерфейсами управления.

Неполная наблюдаемость заключается в том, что истинное состояние работоспособности комплекса уточняется поступательно, по мере получения результатов измерений. Каждый узел может содержать произвольное число служб, показатели работоспособности которых определяются только по результатам диагностических измерений. В исходных данных представлена лишь малая часть сведений о текущем состоянии комплекса.

Динамичность означает, что конфигурационные параметры изменяются во времени вследствие штатного функционирования комплекса, действий администраторов и внешних факторов. Состояние, зафиксированное в один момент времени, может не соответствовать действительности уже через короткий промежуток.

Под цифровой средой автором понимается исследуемый комплекс в его отношении к системе оценки работоспособности, то есть совокупность узлов, соединений и служб, на которые направлены диагностические действия и от

которых поступают их результаты. Неопределенность цифровой среды складывается из динамичности комплекса и зависимости результатов диагностических действий от неизвестных факторов, а частичная наблюдаемость из неполноты сведений о его составе и состоянии.

Обобщенная структура распределенного программно-аппаратного комплекса представлена на рисунке 1.1.

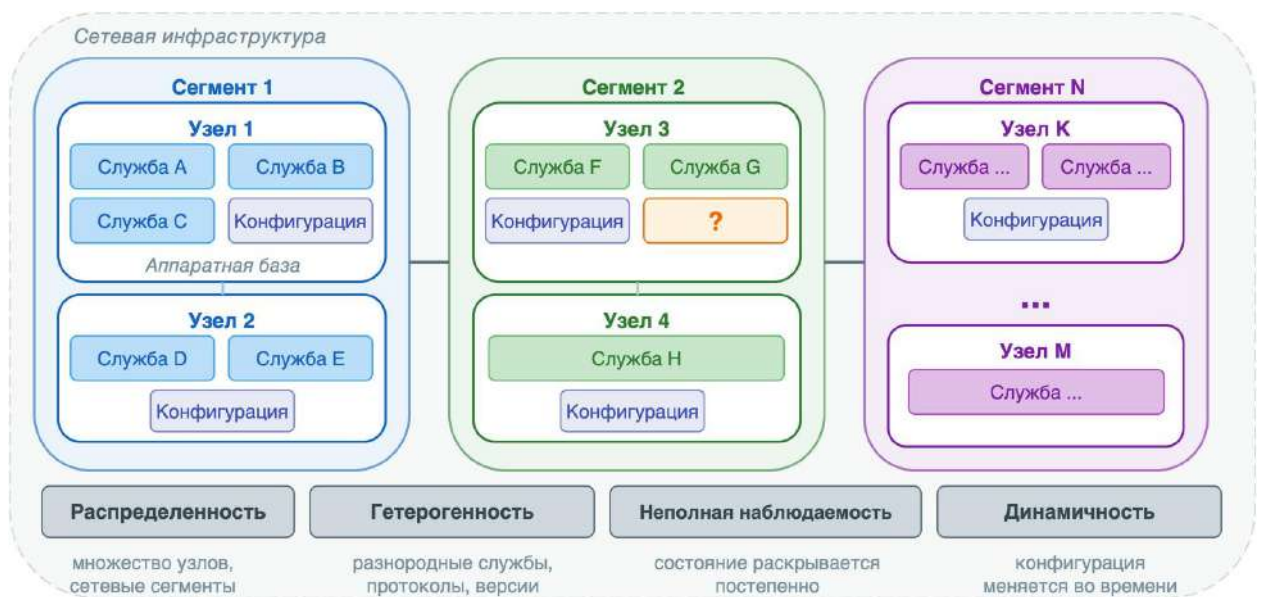


Рисунок 1.1 – Обобщенная структура распределенного программно-аппаратного комплекса

Обследование распределенного программно-аппаратного комплекса представляет собой систематическое получение сведений о состоянии его компонентов, выявление отклонений от штатных параметров и проверку соответствия установленным требованиям. Результатом обследования служит оценка работоспособности комплекса — характеристика степени выполнения им предъявляемых требований, вычисляемая по собранным сведениям. Обследование включает аудит конфигурации, мониторинг состояния служб и выявление аномалий. Сложность задачи определяется распределенностью объекта обследования, разнородностью подзадач и неполнотой информации о среде. Информация об объекте рассредоточена между десятками и сотнями узлов.

Разнородные подзадачи требуют параллельного решения. Полная картина состояния складывается лишь постепенно.

Перечисленные факторы определяют потребность в таком подходе к автоматизации обследования, который по своей структуре соответствовал бы природе обследуемого объекта. Перспективным направлением является многоагентный подход.

Фундаментальным элементом распределенных интеллектуальных систем является понятие интеллектуального агента. Оно сформировалось в работах по искусственному интеллекту и распределенным системам в конце XX века. Под интеллектуальным агентом понимается автономная сущность (программный модуль или устройство), способная воспринимать окружающую среду через сенсоры и воздействовать на нее посредством исполнительных механизмов для достижения заданных целей [115]. Такой агент действует автономно на протяжении длительного времени без вмешательства человека. Реакция агента считается рациональной, если его действия способствуют достижению заданных параметров управляющего процесса.

Интеллектуальный агент характеризуется автономностью, непрерывностью функционирования, реактивностью и социальностью.

Автономность представляет собой способность принимать решения и осуществлять действия без постоянного внешнего управления. Агент самостоятельно определяет цели, выбирает методы их достижения и реализует стратегию поведения с учетом оценки текущего состояния среды.

Непрерывность функционирования определяется постоянным присутствием агента в вычислительной системе и его готовностью к восприятию и обработке событий в режиме реального времени. Агент функционирует как активный элемент среды, обеспечивая реакцию на изменения и поддерживая внутреннюю активность [86].

Реактивность означает способность отслеживать происходящее в системе и реагировать на изменения. Наряду с реактивностью интеллектуальный агент обладает свойством проактивности, то есть инициирует действия, порожденные

внутренними механизмами планирования и прогнозирования, без внешнего побуждения.

Социальность представляет собой способность к координации и взаимодействию с другими агентами в рамках общей вычислительной среды [121] и включает в себя обмен сообщениями, согласование действий, формирование планов и совместное выполнение задач.

М. Вулдридж и Н. Дженнингс в классических работах заложили основы агентно-ориентированных архитектур и выделили указанные свойства как отличительные признаки программных агентов [86; 133; 134]. С. Рассел и П. Норвиг включили агентную парадигму в общий каркас искусственного интеллекта, предложив рассматривать интеллектуальную систему как рационального агента, взаимодействующего со средой для максимизации полезности своих действий [115; 116].

В отечественной науке идеи интеллектуальных агентов развивались в парадигме теории автоматического управления и искусственного интеллекта. С. Н. Васильев развивал методы интеллектуализации автоматизированных систем, формализуя логические процедуры принятия решений для придания им автономности и адаптивности [15]. В. И. Городецкий исследовал модели многоагентной самоорганизации и перспективы применения многоагентных систем в различных предметных областях [16; 17]. В. Б. Тарасов систематизировал методологические основания агентно-ориентированного подхода, рассмотрев его связь с теорией сложных адаптивных систем [52]. Г. В. Рыбина развивала свойства многоагентных интеллектуальных систем и модели координации поведения агентов, подчеркивая необходимость автономности и адаптивности компонентов в гетерогенных средах [45]. Д. А. Пospelov развивал теорию ситуационного управления, в которой принятие решений определяется анализом текущей ситуации [43], что концептуально близко к поведению интеллектуального агента.

Совокупность взаимодействующих интеллектуальных агентов образует многоагентную систему. Согласно ГОСТ Р 59277-2020 многоагентная система определяется как система из множества взаимодействующих интеллектуальных

агентов, способная решать задачи, трудные или нерешаемые для монолитной системы [19]. Такие системы позволяют решать задачи, которые сложно или невозможно решить с помощью одного централизованного алгоритма или отдельного агента [107]. Многоагентные системы обладают высокой степенью автономности компонентов и обеспечивают их постоянное взаимодействие с динамичной средой.

Агентные интеллектуальные системы принципиально отличаются от традиционных автоматизированных систем и классических экспертных систем. Автоматизированная система действует по строго заданному алгоритму и выполняет запрограммированные действия без обратной связи со средой. Интеллектуальные агенты адаптивно реагируют на изменение среды, анализируют результаты своих действий и корректируют поведение, преодолевая ограничения жестко запрограммированных централизованных систем. Классическая экспертная система функционирует на основе статичной базы знаний и логического вывода, при этом не взаимодействует напрямую с динамичной внешней средой. Агентная система, напротив, представляет собой коллектив автономных исполнителей, которые взаимодействуют в реальном времени с окружением и друг с другом [9]. В отечественных стандартах по искусственному интеллекту агентные системы выделены в самостоятельный класс систем [19].

Многоагентный подход применяется в задачах анализа и управления сложными техническими системами различных предметных областей.

В промышленном мониторинге многоагентные системы применяются для отслеживания состояния оборудования на территориально распределенных производственных площадках [34; 37; 49]. Отдельные агенты собирают телеметрию с датчиков, анализируют закономерности деградации и координируют упреждающее обслуживание. Такая архитектура позволяет масштабировать систему мониторинга при росте числа контролируемых объектов без переработки централизованного ядра.

В диагностике сложных технических систем (авиационные двигатели, энергетические сети, телекоммуникационные инфраструктуры) многоагентные

системы обеспечивают декомпозицию диагностической задачи между специализированными агентами, каждый из которых отвечает за анализ определенной подсистемы [52]. Результаты локальной диагностики агрегируются для формирования целостной картины состояния объекта.

В управлении распределенными вычислительными ресурсами (облачные платформы, кластеры) многоагентные системы используются для динамического распределения нагрузки, обнаружения аномалий и автоматического реагирования на отказы [75]. Каждый агент контролирует подмножество ресурсов, при этом коллективное поведение обеспечивает устойчивость инфраструктуры.

В сфере контроля состояния информационных систем многоагентные архитектуры применяются для анализа конфигураций, проверки соответствия требованиям стандартов и автоматизированного обнаружения аномалий [32; 35; 46; 54; 55]. Сбор событий с распределенных узлов, сверка конфигураций с эталоном и выявление нештатных состояний представляют собой задачи, которые естественным образом декомпозируются на подзадачи, решаемые специализированными агентами. К той же области относятся проверка криптографических преобразований, применяемых в службах комплекса, включая схемы электронной подписи [39], и защита программных компонентов на уровне распределения динамической памяти, для которых при участии автора разработаны метод поиска невозможных дифференциалов блочного шифра с реализующей его программой и рандомизированный распределитель памяти [47; 63; 74].

Перечисленные примеры объединяет общая постановка задачи. Объект анализа является распределенным и состоит из множества компонентов. Информация о его состоянии неполна и уточняется постепенно. Во всех случаях предметом обследования выступает работоспособность объекта, а результатом служит ее оценка. Задача такой оценки допускает декомпозицию на относительно независимые подзадачи. Именно при таких условиях многоагентный подход оказывается наиболее естественным и эффективным, поскольку структура решающей системы соответствует структуре исследуемого объекта.

Концептуальная схема многоагентного подхода к оценке работоспособности распределенных программно-аппаратных комплексов представлена на рисунке 1.2. На схеме показано, как группа интеллектуальных агентов, специализирующихся на различных аспектах обследования, взаимодействует с компонентами распределенного программно-аппаратного комплекса. Агенты обмениваются сведениями, координируют действия и формируют коллективное представление о состоянии среды. Распределенная архитектура многоагентной системы отражает распределенную природу исследуемого объекта, что обеспечивает масштабируемость и адаптивность обследования.

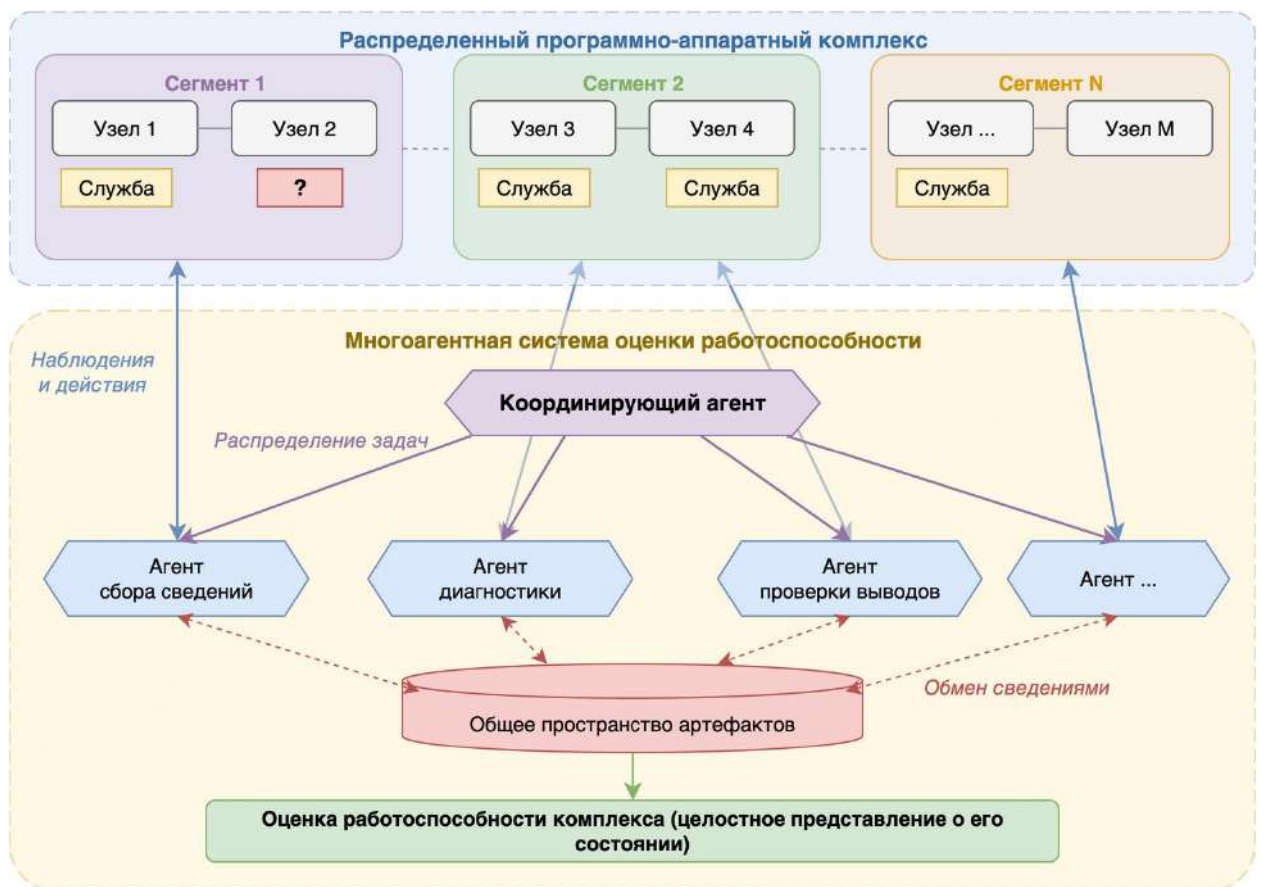


Рисунок 1.2 – Концептуальная схема многоагентного подхода к оценке работоспособности распределенных программно-аппаратных комплексов

Изначально многоагентные системы рассматривались как альтернатива централизованным системам искусственного интеллекта, однако классическая

реализация часто опиралась на заранее заданные модели поведения агентов и фиксированные протоколы коммуникации [123]. В 2020–2025 годах появление больших языковых моделей породило новый класс агентных систем, в которых поведение агента определяется генеративной моделью, способной к рассуждению на естественном языке [14; 56; 91]. Многоагентные платформы на основе больших языковых моделей объединяют специализированных агентов в коллективы, способные к совместному решению сложных задач [99]. Параллельно развивается направление многоагентного обучения с подкреплением, где агенты совершенствуют свое поведение через взаимодействие с общей средой [30; 104; 107; 137]. Агентные имитационные модели находят применение в прогнозировании и управлении сложными процессами [37].

Многоагентные интеллектуальные системы являются сложными иерархическими объектами, поэтому их разработка требует использования аппарата системного анализа [27; 38]. Классический системный подход рассматривает агентную систему как единое целое со сложной внутренней структурой, представимой в терминах состояний, переходов, целей и механизмов адаптации [27]. Агенты и их взаимодействия анализируются в составе целостной системы. Это дает основания включать агентно-ориентированную парадигму в понятийный аппарат системного анализа [8; 9] как научной дисциплины.

Теоретические принципы системного анализа напрямую применимы к исследованию многоагентных интеллектуальных систем. Принцип иерархической декомпозиции позволяет структурировать многоагентную систему на уровни, что особенно актуально для задачи обследования распределенных комплексов, где необходимо разделение между стратегическим планированием, распределением задач и непосредственным взаимодействием [23; 40] с компонентами комплекса. Число уровней такой структуры задается заранее, тогда как наполнение уровней складывается в ходе обследования под состав исследуемого комплекса. Выявление прямых и обратных связей между компонентами обеспечивает анализ информационных потоков между агентами. Рассмотрение системы в окружении и

учет фактора неопределенности критически значимы для задач, где среда является динамичной и непредсказуемой [38].

Принцип эмерджентности особенно актуален для многоагентных систем, поскольку взаимодействуя, агенты порождают функциональности системы, недоступные отдельному агенту [52]. Применительно к оценке работоспособности распределенных программно-аппаратных комплексов эмерджентность проявляется в том, что совокупность агентов, каждый из которых видит лишь фрагмент картины, формирует целостное представление о состоянии комплекса. В отечественной науке Н. Н. Моисеев создал теорию иерархических систем и оптимального управления, предложив методы координации большого числа взаимосвязанных элементов в сложной системе [38]. Эти идеи легли в основу координации агентов в многоагентных системах, где ни один агент не обладает полным знанием и необходимо распределенное управление. В. В. Качала развивал методы системного подхода к техническим и организационным системам, подчеркивая необходимость учета разнообразных связей и факторов при анализе больших систем [27]. Д. А. Новиков исследовал механизмы управления организационными и техническими системами, в том числе задачи координации автономных элементов с неполной информацией [13; 36; 40].

В зарубежной литературе формальным основам многоагентных систем посвящены работы Я. Шохам и К. Лейтон-Брауна по алгоритмическим и игровым аспектам [120; 121], исследования по кооперативному управлению в децентрализованных системах [80; 87], а также обзоры по методам глубокого обучения с подкреплением для многоагентных задач [1; 107; 109]. На стыке системного анализа и агентных технологий формируется методология проектирования интеллектуальных систем, опирающаяся на формализацию предметной области, декомпозицию задачи на подсистемы отдельных агентов, определение структуры их взаимодействия и анализ глобального поведения системы.

На практике системный анализ предусматривает разработку формализованных моделей и требований к системе. Формализация представляет

собой общенаучный метод, обеспечивающий однозначность постановки задачи. Перед разработкой решения необходимо описать цели системы на строгом математическом языке, определить критерии эффективности и формально задать характеристики среды. Для задачи оценки работоспособности распределенных программно-аппаратных комплексов формализованной целью служит построение оценки работоспособности, наиболее близкой к истинному значению. Критериями эффективности выступают полнота обследования, достоверность выводов, число затраченных действий и воздействие на штатное функционирование комплекса. Стандарт ГОСТ Р ИСО/МЭК 25010-2015 задает модели качества программных и программно-аппаратных систем, определяя метрики надежности, безопасности и адаптивности, применимые в том числе к оценке архитектуры [18; 21] многоагентных систем.

К многоагентным системам применяется метод проектирования архитектуры программно-технической системы. Научно обоснованное архитектурное решение определяет, какие виды агентов будут в системе, какие роли они выполняют, как организованы их взаимодействие и управление. В многоагентных интеллектуальных системах выбор между централизованной архитектурой с координатором и распределенной архитектурой, где агенты полностью автономны и договариваются напрямую, влияет на надежность и масштабируемость [11; 75; 123]. Системный подход к построению многоагентных систем позволяет выявить прямые соответствия между базовыми концепциями системного анализа и свойствами агентной архитектуры [9; 45]. Целостность агентной системы обеспечивается посредством протоколов взаимодействия и механизмов координации [28; 33], благодаря чему достигается согласование действий агентов. Система в целом приобретает свойства, которые не присущи отдельным агентам, в том числе способность к коллективному планированию, распределенному обучению или высокую отказоустойчивость за счет дублирования функций.

Таким образом, многоагентный подход является теоретически обоснованным выбором для построения системы автоматизированной оценки работоспособности распределенных программно-аппаратных комплексов. Его обоснованность

определяется соответствием между распределенной природой объекта анализа и распределенной архитектурой решающей системы, возможностью декомпозиции сложной задачи на подзадачи, решаемые специализированными агентами, а также наличием развитого научного аппарата, позволяющего проектировать и исследовать подобные системы с позиций системного анализа. Вместе с тем задача формального описания процесса многоагентной оценки работоспособности при неполной информации о среде требует специализированного математического аппарата, учитывающего частичную наблюдаемость, множественность взаимодействующих агентов, стохастичность переходов между состояниями и кооперативный характер взаимодействия агентов.

1.2 Существующие подходы к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов

Автоматизация оценки работоспособности распределенных программно-аппаратных комплексов прошла несколько этапов развития, каждый из которых расширял возможности предыдущего, но одновременно обнаруживал новые ограничения, требующие принципиально иных архитектурных решений.

Теоретический фундамент для систематического анализа состояния сложных распределенных систем формировался на стыке нескольких направлений. Б. Шнайер предложил способ иерархической декомпозиции сложной цели на подцели, позволяющий формализовать множество возможных путей ее достижения [118]. Р. Андерсон разработал инженерный подход к построению надежных распределенных систем [61] и показал, что качество их функционирования определяется техническими, экономическими и организационными факторами в совокупности [62]. Пендлтон и Рамос с соавторами систематизировали количественные методы оценки, определив метрики для сравнительного анализа различных подходов [57; 101].

В отечественной науке задачи автоматизированного анализа состояния распределенных систем получили системное развитие в работах И. В. Котенко и его научной группы. Ими предложена методология анализа состояния компьютерных сетей на основе моделирования сценариев развития неблагоприятных событий [31], систематизированы интеллектуальные механизмы поддержки принятия решений при управлении сложными сетевыми инфраструктурами [32]. А. А. Чечулин совместно с Котенко разработал методы построения графовых моделей на основе анализа конфигурации сети [53], позволяющие автоматически выявлять цепочки взаимосвязанных событий и оценивать их вероятность. Графовые модели анализа получили широкое развитие в зарубежных исследованиях [124; 135]. Методы обнаружения аномалий на основе интеграции фрактального анализа и машинного обучения развиты Котенко и И. Б. Саенко [35], интеллектуальному анализу данных о функционировании распределенных систем посвящены работы [2; 50; 51]. Для прогнозирования состояния контролируемого объекта по данным текущих наблюдений предложены методы рекуррентной коррекции [112]. Совокупность этих работ сформировала научную базу для перехода от ручного экспертного анализа к автоматизированным интеллектуальным системам.

Исторически первым подходом к автоматизации обследования являлось использование средств с заранее запрограммированной логикой. Модульные инструментарии объединяют готовые модули для диагностики и проверки состояния компонентов распределенного комплекса, однако оператор вручную выбирает последовательность действий и параметры запуска. Платформы имитационного моделирования позволяют автоматически запускать последовательности проверочных действий по заданным профилям и формировать отчетность о результатах [71; 85]. Коммерческие платформы этого класса предоставляют библиотеки сценариев и автоматизированное выполнение, но опираются на заранее запрограммированные последовательности действий и базы известных типов отклонений. Петренко систематизировал методы обнаружения

аномалий функционирования вычислительных систем [42], показав возможности и ограничения существующих средств.

Сценарные средства обладают существенными ограничениями [29; 57; 101]. Большинство таких средств следуют предопределенной логике и не умеют адаптироваться к непредвиденным ситуациям. Они проверяют лишь те отклонения, которые заложены в их базу, и пропускают сложные цепочки взаимосвязанных событий, требующие понимания контекста среды. Существующие инструменты обычно имеют статическую архитектуру и ориентированы на известные типы отклонений, что снижает их эффективность в динамически изменяющихся средах. Если возникает нестандартная ситуация, такая система не способна перестроить логику поведения, поскольку вне заложенного сценария она не имеет механизмов обучения или планирования.

Ограничения сценарной автоматизации стимулировали развитие одноагентных интеллектуальных систем, в которых агент на базе искусственного интеллекта принимает решения самостоятельно [44]. Дэн и др. предложили прототип, в котором большая языковая модель используется для генерации пошаговых действий [70; 111] при анализе состояния распределенной системы. Прототип подтвердил применимость языковых моделей к автоматизации обследования, однако требует значительного вмешательства человека для фильтрации и направления выводов модели. Сюй и др. разработали систему автоматизации отдельных этапов оценки состояния с использованием языковой модели для планирования и генерации программного кода [65]. Карагannis и др. представили платформу для автономного анализа, сочетающую имитационное моделирование проверочных сценариев с валидацией результатов [59; 108].

Общее ограничение одноагентных систем состоит в отсутствии разделения ответственности между специализированными компонентами. Единственный агент вынужден совмещать функции сбора информации, анализа, принятия решений и проверки результатов. При увеличении масштаба анализируемого комплекса это приводит к потере контекста, снижению качества решений и невозможности параллельного обследования различных подсистем.

Дополнительной проблемой является ограничение контекстного окна языковой модели. При анализе крупного распределенного комплекса объем информации, который необходимо учитывать одновременно (топология сети, обнаруженные службы, результаты предыдущих действий), быстро превышает возможности одного агента. Многоагентная архитектура решает эту задачу через распределение контекста между специализированными агентами, каждый из которых хранит и обрабатывает информацию, относящуюся к его роли.

Выявленные ограничения одноагентных систем привели к разработке многоагентных архитектур, в которых несколько специализированных агентов совместно решают задачу анализа. В 2024–2025 годах появились платформы, в которых агенты на основе языковых моделей совместно проводят анализ без постоянного участия человека.

Конг и др. предложили многоагентную систему, моделирующую работу команды аналитиков [131]. Один агент собирает информацию об исследуемой инфраструктуре, другой анализирует обнаруженные отклонения, третий подбирает способы верификации, а координирующий модуль планирует их взаимодействие. В экспериментах эта система превзошла одноагентные решения по полноте обнаружения и сумела автономно выполнить полный цикл анализа реальной тестовой среды. Ли и др. предложили многоуровневую интеллектуальную систему INNES [84], интегрирующую глубокое обучение с подкреплением с модулем сбора информации для анализа состояния сетевой инфраструктуры. Другая платформа этого класса интегрирует взаимодействующих агентов с большими языковыми моделями для обнаружения отклонений, проверки возможных сценариев их развития и формирования отчетов [68].

Графовые модели школы И. В. Котенко [53] позволяют формализовать множество возможных цепочек событий и зависимостей между ними, что создает основу для планирования действий агентов.

Вместе с тем существующие платформы рассчитаны на выполнение одного сеанса обследования и не предусматривают механизма систематического улучшения на основе накопленного опыта. Каждый новый сеанс начинается без

учета знаний, полученных ранее на аналогичных или родственных инфраструктурах. Состав агентов и связи между ними закрепляются при проектировании платформы, тогда как состав обследуемого комплекса до начала работы не известен. Заданная заранее структура либо покрывает наихудший случай избыточным числом агентов, либо оставляет часть комплекса вне обследования. Ни одна из перечисленных платформ не предлагает формальной модели, описывающей процесс обследования в терминах математических объектов и позволяющей строить доказательные оценки качества.

Архитектура взаимодействия агентов в существующих системах, как правило, отражает этапы обследования и разделение задач [75]. Широко применяется модульный подход, когда разные агенты отвечают за различные фазы (сбор сведений, детальное обследование, проверка результатов). Для координации агентов используются механизмы обмена сообщениями и совместного планирования, благодаря чему система избегает потери контекста и дублирования действий. Однако неполнота информации и неопределенность среды остаются центральной проблемой. Агенты редко располагают полным знанием об исследуемой системе, поэтому архитектуры должны строиться с учетом частично наблюдаемых состояний и стохастического характера результатов действий [60; 89; 100; 126]. В существующих системах этот аспект учитывается лишь неявно, на уровне эвристик, без формальных моделей. Сарро и др. [114; 117] показали, что формализация процесса анализа через частично наблюдаемый марковский процесс принятия решений дает преимущество перед детерминированным планированием, поскольку позволяет явно учитывать неопределенность результатов действий. Милинг и др. [100] подтвердили применимость этого аппарата для задач последовательного принятия решений при управлении состоянием крупных сетевых инфраструктур.

Наряду с развитием архитектур многоагентных систем активно исследуются среды обучения интеллектуальных агентов. Для того чтобы агент научился эффективно действовать в распределенной среде, ему необходимо получить опыт взаимодействия с моделью такой среды, поскольку нарушать работу реальной

инфраструктуры недопустимо. Замена натурального объекта имитационной моделью применяется и для технологических аппаратов, для которых при участии автора разработаны дискретные динамические и вероятностные модели процессов [12; 25; 26]. Шварц и Курниавати [119] предложили обучающую среду, предоставляющую абстрактную модель сети с настраиваемыми параметрами топологии и конфигурации. Агент взаимодействует со средой через дискретное пространство действий, включающее обследование сегментов сети и анализ обнаруженных служб. Вероятностная модель успеха действий учитывает неопределенность исхода каждого шага, что приближает моделирование к реальным условиям. Стенден и др. [71; 73] представили среду, в которой две взаимодействующие стороны обучаются одновременно. М. Г. Пантелеев и С. М. Литягин предложили архитектуру среды имитационного моделирования многоагентных систем реального времени [41]. Молина-Маркхэм и др. создали платформу FARLAND, ориентированную на обучение агентов в условиях неполной информации [105]. В отличие от детерминированных сред, FARLAND моделирует неопределенность восприятия агента, когда результаты обследования могут быть неполными или зашумленными, а успех действий зависит от множества факторов. Чоудхари и др. предложили подход к автономному анализу на основе глубокой Q-сети [66], обучаемой в моделируемой среде. Обратная связь в перечисленных средах дается пошаговым вознаграждением обучающегося агента. Проверка итога заверченного сеанса по истинному состоянию экземпляра инфраструктуры, известному по построению, в них не предусмотрена.

Теоретической основой обучения агентов является аппарат обучения с подкреплением. Саттон и Барто [72; 90; 125] систематизировали фундаментальные принципы этого подхода, при котором агент обучается через пробы и ошибки при взаимодействии со средой, постепенно находя политику, максимизирующую накопленную награду. Уоткинс и Даян [132] предложили метод Q-обучения, позволяющий агенту оценивать ожидаемую полезность действий в каждом состоянии без построения модели среды. Мних и др. разработали [83] метод глубокой Q-сети, аппроксимирующей функцию ценности с помощью нейронной

сети, что позволило работать с большими пространствами состояний. Для стабилизации обучения применяются механизмы воспроизведения опыта и целевой сети. Шульман и др. [113] предложили алгоритм оптимизации проксимальной политики, ограничивающий величину обновления на каждом шаге и обеспечивающий стабильное обучение без точной настройки гиперпараметров. Мних и др. также [64] разработали асинхронный алгоритм, использующий несколько параллельных агентов для одновременного исследования среды и обновления общей модели.

Теоретической основой глубоких методов обучения агентов послужили работы по обучению представлений и многослойным нейронным сетям. Бенжио и др. [67] систематизировали методы обучения представлений, позволяющие извлекать иерархические признаки из необработанных данных. Перенос обученных представлений на новые задачи распространен и за пределами агентных систем, например при классификации объектов по изображениям [129]. Хинтон и Салахутдинов [81] показали возможность снижения размерности данных с помощью многослойных нейронных сетей. Лекун, Бенжио и Хинтон [92] обобщили достижения глубокого обучения, определив его как основной инструмент для задач распознавания, классификации и принятия решений в сложных средах. Методы кластеризации и построения наборов данных при этом применимы к экспериментальным данным различной природы, в том числе к спектральным характеристикам флуоресцентных красителей [88; 122]. Эти фундаментальные результаты обеспечили возможность применения глубоких нейросетевых архитектур в агентных системах, где аппроксимация функций ценности и политик требует работы с многомерными пространствами состояний.

Централизованное обучение с децентрализованным выполнением [69; 102; 103; 107; 130; 136] позволяет агентам использовать глобальную информацию во время обучения, но действовать автономно во время выполнения. Нгуен и Редди [106] систематизировали применение глубокого обучения с подкреплением для задач анализа состояния сетевых инфраструктур.

Сходимость классических алгоритмов обучения с подкреплением затруднена из-за большой размерности пространства действий агента. В реальной инфраструктуре имеются сотни узлов и служб, обширное пространство допустимых действий и длинные последовательности шагов. Поэтому наряду с классическим обучением с подкреплением развиваются гибридные подходы. Чен и др. предложили подход, объединяющий обучение с подкреплением с имитационным обучением на основе экспертных трасс [76; 77; 96]. Идея состоит в использовании записей успешных сеансов обследования для ускорения обучения агента и направления его поведения через обратную связь от дискриминатора, отличающего действия агента от действий эксперта. По результатам испытаний этот подход достиг наивысшей эффективности среди рассмотренных методов. Применение глубокого обучения с подкреплением для автоматизации различных этапов анализа исследовано Ху и др. [82], Маэдой и Мимурой [93]. Ганем и Чен [78; 79] предложили иерархическое обучение с подкреплением для анализа крупных сетей, декомпозируя задачу на подзадачи для повышения масштабируемости.

К обучению с подкреплением по оценке заверщенного сеанса относится поиск политики отбором элит, при котором из порожденных решений отбирается лучшая часть, а следующее поколение решений строится на отобранном. Сильвер и др. [94] обучили программу игры в го без человеческих партий, дообучая политику на собственных играх с отбором ходов по итогу партии. Ромера-Паредес и др. [95] применили отбор по проверяемому результату к программам, порождаемым большой языковой моделью, и получили новые математические конструкции. Оба метода рассчитаны на одного агента с единой политикой. Согласованное улучшение политик нескольких уровней иерархии, обучаемых одной моделью, ими не рассматривается.

Отдельным направлением с 2023 года стало применение больших языковых моделей в качестве ядра интеллектуальных агентов [56; 91; 128]. В отличие от обучения с подкреплением, где агент осваивает политику через многократное взаимодействие со средой, языковые модели привносят обширные знания,

полученные при предварительном обучении на больших корпусах текстов. Это позволяет агенту выполнять действия, для которых у него нет специфического обучающего опыта, опираясь на общее понимание предметной области. Хонг и др. [99] предложили платформу MetaGPT, реализующую координацию нескольких агентов через механизм метапрограммирования. Агентам назначаются роли (аналитик, разработчик, верификатор), и система координирует их взаимодействие через структурированный обмен сообщениями.

Вместе с тем поведение агента определяется параметрами языковой модели, обученной на общих данных без учета конкретной задачи обследования распределенного комплекса. Это приводит к генерации правдоподобных, но ошибочных действий. Отсутствует механизм улучшения от сеанса к сеансу, поскольку модель не изменяет параметров по результатам конкретного обследования, что отличает ее от агентов с обучением с подкреплением. Вычислительные затраты на работу языковых моделей существенно выше, чем на выполнение обученной политики, что ограничивает масштабируемость при анализе крупных инфраструктур.

Перспективным направлением является сочетание языковых моделей с обучением с подкреплением по оценке заверщенного сеанса. Языковая модель обеспечивает начальное знание и способность к рассуждению на естественном языке. Обучение с подкреплением по итогу сеанса обеспечивает систематическое улучшение политики по сигналу среды без экспертной разметки. Такое сочетание позволяет создать агентов, которые рассуждают о задаче на высоком уровне абстракции и при этом совершенствуют конкретные навыки от сеанса к сеансу. Реализация такого сочетания требует формальной модели процесса обследования, среды, в которой результат сеанса проверяется без участия человека, и алгоритма обучения, согласующего политики уровней иерархии. Эволюция подходов к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов систематизирована автором на рисунке 1.3.



Рисунок 1.3 – Эволюция подходов к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов

На рисунке отражен переход от сценарной автоматизации через одноагентные интеллектуальные системы к многоагентным архитектурам с самообучением. Каждый этап расширяет возможности предыдущего по осям автономности, адаптивности и масштабируемости, однако вместе с этим обнаруживаются новые научные задачи, не решенные на текущем уровне развития.

Результаты сравнения существующих подходов по критериям автономности обследования, наличия формальной модели процесса, формирования структуры системы под состав комплекса, самообучения на собственных сеансах и автоматической проверки результата сеанса систематизированы автором в таблице 1.1.

Таблица 1.1 – Сравнительный анализ существующих подходов к автоматизированной оценке работоспособности распределенных программно-аппаратных комплексов

Класс подходов	Автономность обследования	Формальная модель процесса	Формирование структуры под состав комплекса	Самообучение на собственных сеансах	Автоматическая проверка результата сеанса
Сценарные средства автоматизации	Низкая, требуется ручное управление	Нет	Нет, фиксированные модули	Нет	По сценарию, для известных отклонений
Платформы имитационного моделирования	Средняя, выполнение заданных профилей	Нет	Нет, заданные профили	Нет	Встроенная по сценарию
Одноагентные системы на основе языковых моделей	Низкая, требуется фильтрация экспертом	Нет	Нет, единственный агент	Нет, параметры модели неизменны	Ручная
Многоагентные платформы	Высокая, автономный полный цикл	Нет, эвристики и инструкции	Нет, состав агентов задан при проектировании	Нет, каждый сеанс с нуля	Нет, результат не сверяется с истинным состоянием

С каждым этапом эволюции подходов возрастает автономность системы и ее способность принимать решения без участия человека, однако ни один из существующих классов подходов не удовлетворяет всем пяти критериям одновременно. Сценарные средства обеспечивают воспроизводимость, но лишены адаптивности. Одноагентные системы приносят гибкость принятия решений, но не справляются с масштабом реальных инфраструктур. Многоагентные платформы решают проблему масштаба и параллельности, однако закрепляют состав агентов при проектировании, начинают каждый сеанс с нуля и не сверяют результат обследования с истинным состоянием комплекса. Существующие среды обучения дают агенту опыт взаимодействия с моделью инфраструктуры, но не проверяют итог заверченного сеанса по истинному состоянию экземпляра, известному по построению. Отсутствие формальной модели процесса обследования препятствует как доказательной оценке качества, так и систематическому улучшению от сеанса к сеансу.

Ни один из рассмотренных подходов не сочетает высокую автономность обследования, формальную модель процесса, формирование структуры системы под состав комплекса, самообучение на собственных сеансах и автоматическую проверку результата без участия эксперта.

1.3 Постановка задач диссертационного исследования

Проведенный анализ теоретических основ многоагентных интеллектуальных систем и существующих подходов к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов позволяет систематизировать выявленные ограничения и определить направления, требующие научной проработки. Несмотря на значительный прогресс в создании специализированных сред моделирования, разработке алгоритмов обучения и применении больших языковых моделей, существующие подходы обладают фундаментальными ограничениями, препятствующими достижению полной автономности и адаптивности обследования.

Первым ограничением остается высокая трудоемкость ручного обследования. Полное обследование одного распределенного программно-аппаратного комплекса требует от сорока до двухсот часов работы квалифицированного специалиста в зависимости от числа узлов, количества программных служб и сложности топологии. Потребность в регулярном проведении такого обследования непрерывно растет, тогда как число специалистов, способных выполнять эту работу на должном уровне, не увеличивается пропорционально. Разрыв между объемом задач и кадровыми ресурсами создает устойчивый запрос на автоматизацию.

Средства сценарной автоматизации частично решают проблему трудоемкости, однако обнаруживают собственное ограничение. Они работают по заранее определенным последовательностям операций и не умеют адаптироваться к нештатным ситуациям. Фиксированные модули и предопределенные сценарии не

учитывают специфику конкретного комплекса и не перестраивают логику поведения при столкновении с нестандартными конфигурациями. При обнаружении конфигурационной особенности, отсутствующей в базе данных инструмента, такая система не способна самостоятельно выработать способ ее обследования.

Следствием статичности сценариев является невозможность построения многошаговых цепочек диагностических действий. Существующие инструменты преимущественно ориентированы на выполнение отдельных проверок и не способны выстраивать последовательности, требующие контекстного понимания всей инфраструктуры. Сложные задачи обследования, включающие получение сведений об узлах разных сегментов сети, уточнение конфигурации смежных узлов, а также согласование результатов нескольких параллельных диагностических потоков, остаются прерогативой экспертов. Распределенные комплексы характеризуются сложными зависимостями между компонентами, при которых результат обследования одного элемента определяет стратегию обследования смежных, и сценарная автоматизация не способна учитывать такие зависимости.

Одноагентные интеллектуальные системы на основе больших языковых моделей привнесли гибкость принятия решений, однако не устранили зависимость от эксперта. Человек по-прежнему должен проверять предложения модели, отсеивать нерелевантные рекомендации и принимать окончательные решения о выполнении действий. Кроме того, единственный агент вынужден совмещать функции сбора сведений, анализа, принятия решений и проверки результатов. При увеличении масштаба обследуемого комплекса объем информации, который необходимо учитывать одновременно, быстро превышает возможности одной модели, что ведет к потере контекста и снижению качества решений.

Многоагентные платформы преодолевают часть перечисленных ограничений за счет параллельной работы специализированных агентов, однако сохраняют закреплённую при проектировании структуру и не накапливают опыт между сеансами. Состав агентов и связи между ними закрепляются при

проектировании, тогда как состав обследуемого комплекса до начала работы не известен и раскрывается поступательно. Заданная заранее структура вынуждена либо покрывать наихудший случай избыточным числом агентов, либо оставлять часть комплекса вне обследования. Механизм систематического улучшения качества работы по результатам предыдущих сеансов обследования в платформах отсутствует. Каждый новый сеанс начинается с нуля, без учета опыта, полученного при обследовании аналогичных или родственных конфигураций, и повышение качества требует прямого участия разработчиков.

Ограничения существующих подходов к автоматизированной оценке работоспособности распределенных программно-аппаратных комплексов систематизированы автором на рисунке 1.4.



Рисунок 1.4 – Систематизация ограничений существующих подходов к автоматизированной оценке работоспособности распределенных программно-аппаратных комплексов

Помимо перечисленных практических ограничений проведенный анализ позволил выявить нерешенные научные задачи, непосредственно определяющие содержание диссертационного исследования. Каждая из этих задач порождена конкретным пробелом в существующем научном аппарате, а их совместное решение необходимо для создания многоагентной интеллектуальной системы автоматизированной оценки работоспособности.

Первая нерешенная задача состоит в отсутствии математической модели многоагентной оценки работоспособности распределенных программно-аппаратных комплексов при неполной информации о среде. Существующие многоагентные системы не имеют математического описания процесса обследования. Действия агентов в них определяются эвристиками или инструкциями языковой модели, тогда как оптимизация формально определенного критерия отсутствует. Без формальной модели невозможно строить доказательные оценки качества, сравнивать стратегии обследования на единой основе и обосновывать архитектурные решения.

Задача оценки работоспособности распределенного комплекса обладает четырьмя существенными особенностями, которые должна учитывать формальная модель. Первая особенность связана с частичной наблюдаемостью. Истинное состояние комплекса недоступно агентам напрямую и уточняется лишь по мере получения результатов измерений на его компонентах. Вторая особенность определяется множественностью агентов, каждый из которых располагает локальной информацией и должен координировать свои действия с другими. Третья особенность обусловлена стохастичностью переходов между состояниями. Результат диагностического действия зависит от неизвестных факторов конфигурации, текущей нагрузки и состояния исследуемых служб. Четвертая особенность состоит в кооперативном характере взаимодействия. Агенты преследуют общую цель обследования, и оценка результата едина для всех участников.

Из формальных аппаратов последовательного принятия решений все четыре особенности одновременно учитывает децентрализованный частично

наблюдаемый марковский процесс принятия решений [110; 127]. Однако классическая постановка этого процесса опирается на три допущения, которые в задаче оценки работоспособности нарушаются. Состояние процесса описывает только исследуемый комплекс, тогда как в задаче оно включает и накопленные агентами сведения. Состав допустимых действий фиксирован и от состояния не зависит, тогда как в задаче он определяется объемом собранных сведений. Углубленная диагностика службы допустима лишь после снятия первичных показателей узла и самой службы, а установленная недоступность службы исключает ее дальнейшую диагностику. Вознаграждение вычисляется после каждого шага, тогда как качество обследования характеризуется близостью построенной оценки меры работоспособности к истинному значению и определяется только по совокупности выполненных действий.

Требуется математическая модель с композитным пространством состояний, объединяющим состояние комплекса с накопленными агентами сведениями, с множеством допустимых действий, зависящим от накопленных сведений, а также с оценкой завершённой трассы исполнения вместо пошагового вознаграждения [6]. Под трассой исполнения понимается последовательность состояний, действий и наблюдений, порождаемая системой за один сеанс обследования. Множество допустимых действий при этом не обязано расширяться по ходу сеанса. Обнаружение соединений узла открывает опрос соседних с ним узлов, а фиксация недоступности службы исключает операции ее углубленной диагностики.

Вторая нерешенная задача состоит в отсутствии параметрической модели архитектуры многоагентной интеллектуальной системы, структура которой формируется в ходе обследования под состав исследуемого комплекса. Исследования многоагентных систем преимущественно проводились на модельных средах с десятками компонентов и заранее заданным составом агентов. Число узлов, перечень служб и значения конфигурационных параметров реального комплекса раскрываются поступательно, по мере целенаправленного взаимодействия с ним. Фиксация состава агентов и связей между ними до начала

обследования означала бы предопределение декомпозиции задачи до получения сведений, от которых эта декомпозиция зависит.

При обследовании крупного распределенного комплекса одновременно решаются задачи стратегического, тактического и операционного масштабов. На стратегическом уровне формируется общая стратегия обследования, определяются приоритетные направления и распределяются ресурсы между ними. На тактическом уровне стратегические указания декомпозируются в конкретные задачи для отдельных агентов, управляются зависимости между задачами и агрегируются промежуточные результаты. На операционном уровне агенты непосредственно взаимодействуют с компонентами комплекса, выполняя диагностические операции, собирая сведения и интерпретируя полученные данные. Перечисленные масштабы не сводятся один к другому и различаются ценой ошибки. Ошибочная стратегия обесценивает все последующие операции, ошибочная операция искажает единственное наблюдение.

Необходима модель архитектуры, параметризованная числом уровней иерархии, в которой три уровня отвечают трем масштабам решений [4; 8]. Политики всех уровней параметризуются единой языковой моделью, а разделение ролей несет конфигурация агента, назначаемая при делегировании задачи и состоящая из системной инструкции и назначенного состава операций. Передача сведений между агентами должна обеспечиваться четырьмя взаимодополняющими механизмами информационного пространства, которыми являются локальная память агента, наследование контекста при делегировании, почтовый ящик и пространство артефактов. Состав действующих агентов и связи подчинения между ними должны порождаться решениями самих агентов по мере накопления сведений о комплексе и описываться графом делегирования, входящим в состояние системы. Порождение дочернего агента есть действие, допустимость которого определяется накопленными сведениями наравне с допустимостью диагностических операций. Глубина графа задается архитектурой и от комплекса не зависит, поэтому система масштабируется под состав и размер комплекса числом агентов при постоянной длине цепочки передачи задач и отчетов.

Третья нерешенная задача состоит в отсутствии алгоритма обучения многоагентной интеллектуальной системы по результатам собственных завершенных сеансов, совместно улучшающего политики всех уровней иерархии без экспертной разметки. Многоагентные платформы используют языковые модели с неизменными параметрами и не извлекают опыт из завершенных сеансов. Известные методы обучения с подкреплением по оценке заверщенного сеанса рассчитаны на одного агента с единой политикой и не согласуют политики уровней иерархии. Обучение иерархической системы осложнено взаимной зависимостью уровней. Качество решений стратегического уровня зависит от того, насколько хорошо агенты тактического и операционного уровней исполняют назначенные задачи, а качество их работы определяется адекватностью задач, поставленных сверху. Изолированное улучшение политики одного уровня при неизменных политиках остальных способно не улучшить или даже ухудшить общий результат.

Обучение без экспертной разметки требует среды с верифицируемой обратной связью, в которой результат сеанса оценивается однозначно без участия человека. Качество обследования характеризуется близостью построенной оценки меры работоспособности к истинному значению, однако для действующего комплекса истинное значение не известно и близость невычислима. Истинное значение известно по построению лишь для экземпляров комплекса, порождаемых программно с заданными составом, состоянием и значениями конфигурационных параметров. Необходима среда, дополняющая порождение экземпляров постановкой цели обследования и функцией верификации, которая по завершенной трассе, экземпляру и цели вычисляет оценку достижения цели.

Оценка достижения цели характеризует итог сеанса одним числом и не различает сеансы с одинаковым итогом и разной ценой, отличающиеся числом затраченных действий, долей обследованных элементов комплекса, надежностью сделанных выводов и воздействием обследования на работу комплекса. Отбор обучающего материала требует единого скалярного целевого сигнала, учитывающего эти различия. Таким сигналом должна служить оценка завершенной трассы, формируемая критериальной моделью качества функционирования [5]. На

уровне одного сеанса критериальная модель включает пять критериев, полноту, достоверность, эффективность, неинвазивность и результативность, объединяемых агрегирующей функцией полезности с настраиваемыми весовыми коэффициентами, что позволяет задавать приоритеты обследования в зависимости от его назначения. Набор весовых коэффициентов допускает изменение по итерациям обучения, от преобладания полноты обследования на первых итерациях к набору, отвечающему назначению обследования. На системном уровне, охватывающем множество сеансов на различных экземплярах комплекса, необходим критерий устойчивости, характеризующий обобщающую способность системы на ранее не встречавшихся конфигурациях.

Необходим алгоритм кросс-уровневого обучения в среде с верифицируемой обратной связью, в котором обучающий материал формируется элитным множеством трасс с адаптивным пороговым значением [7]. Из каждой элитной трассы извлекаются обучающие пары из локальных историй и действий агентов всех уровней иерархии одновременно, а параметры единой модели обновляются на объединенном множестве пар, что обеспечивает совместную оптимизацию политик всех уровней. Отбор не должен ограничиваться полностью успешными сеансами, поскольку на ранних итерациях обучения их число мало или равно нулю. Адаптивное пороговое значение, вычисляемое по распределению оценок текущей итерации, включает в обучающую выборку частично успешные трассы и повышает требования к материалу по мере совершенствования системы. Отбор по порогу возможен лишь при разбросе оценок внутри итерации. При детерминированной генерации действий сеансы на одном экземпляре повторяют друг друга, и в обучающую выборку не поступают новые образцы поведения. Разнообразие порождаемых трасс должно обеспечиваться температурой генерации действий и выполнением нескольких сеансов на каждом экземпляре комплекса. Проверка результатов обучения должна вестись по критерию устойчивости на контрольных экземплярах, не участвовавших в обучении.

Перечисленные нерешенные задачи взаимосвязаны и образуют единый комплекс. Математическая модель задает основу архитектуры, поскольку

структура многоагентной системы входит в состояние модели, а порождение агента подчинено правилу допустимых действий. Модель архитектуры задает алгоритму обучения разбиение обучающего материала по уровням иерархии и единую модель, параметры которой обновляются. Оценка завершенной трассы, введенная математической моделью и конкретизированная критериальной моделью, служит алгоритму обучения целевым сигналом, а среда с верифицируемой обратной связью замыкает контур, доставляя оценку достижения цели без участия эксперта. Взаимосвязь нерешенных задач представлена на рисунке 1.5.

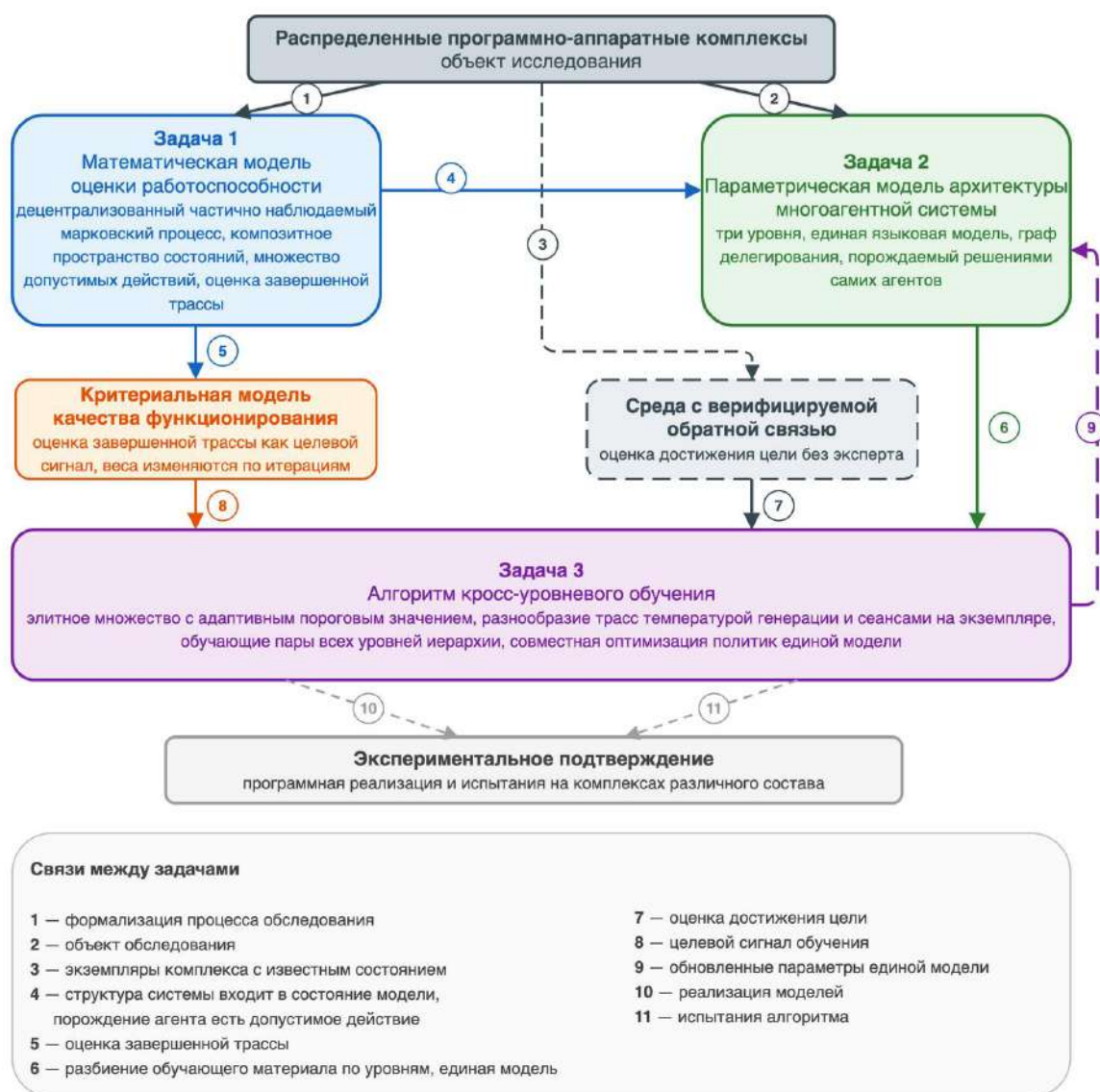


Рисунок 1.5 – Взаимосвязь нерешенных научных задач диссертационного исследования

Решение перечисленных задач нуждается в экспериментальном подтверждении на распределенных программно-аппаратных комплексах различного состава. Для этого необходима программная реализация предложенных моделей и алгоритма в виде комплекса взаимодействующих программ, а также проведение серии экспериментальных исследований с количественной оценкой результатов. Программный комплекс должен включать реализацию многоагентной системы на основе иерархической архитектуры, среду с верифицируемой обратной связью и модуль кросс-уровневого обучения [48]. Экспериментальные исследования должны подтверждать эффективность предложенных решений количественными показателями на экземплярах комплекса, не использовавшихся при обучении, и включать сопоставление с существующими системами обследования.

Выявленные задачи определяют потребность в разработке моделей и алгоритмов оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды. Средством решения выступает многоагентная интеллектуальная система, которая должна опираться на математическую модель с композитным пространством состояний и оценкой завершенной трассы, обладать архитектурой со структурой, порождаемой решениями самих агентов, и совершенствоваться алгоритмом кросс-уровневого обучения по оценкам завершенных сеансов в среде с верифицируемой обратной связью. Решение перечисленных задач позволит создать научную основу для разработки многоагентных интеллектуальных систем оценки работоспособности распределенных программно-аппаратных комплексов, обеспечивающих повышение полноты и точности обследования по сравнению с существующими подходами.

Выводы по главе 1

В первой главе исследованы свойства распределенных программно-аппаратных комплексов как объектов автоматизированной оценки работоспособности и существующие подходы к ее автоматизации, сформулированы нерешенные научные задачи. Получены следующие основные результаты.

1. Исследованы свойства распределенных программно-аппаратных комплексов как объектов автоматизированной оценки работоспособности. Показано, что территориальная рассредоточенность, неоднородность компонентов, ограниченный доступ к внутреннему состоянию и изменчивость конфигурации порождают требования, которым в наибольшей мере отвечает многоагентная парадигма. Структурное подобие между объектом оценки и коллективом специализированных агентов, каждый из которых решает отдельный фрагмент общей задачи, обеспечивает масштабируемость и адаптивность обследования. Обосновано, что для строгого описания такого процесса необходим математический аппарат, способный отразить неполноту сведений о цифровой среде, множественность взаимодействующих участников, вероятностную природу результатов диагностических действий и общую цель участников.

2. Проведен анализ четырех классов подходов к автоматизации оценки работоспособности (сценарные средства, платформы имитационного моделирования, одноагентные интеллектуальные системы, многоагентные платформы) и выполнено их сопоставление по пяти критериям, которыми являются автономность обследования, наличие формальной модели процесса, формирование структуры системы под состав комплекса, самообучение на собственных сеансах и автоматическая проверка результата сеанса. Выявлено, что каждый класс покрывает лишь часть указанных критериев. Сценарные средства ограниченно адаптируются к нестандартным конфигурациям, одноагентные решения теряют контекст при росте масштаба, а многоагентные платформы закрепляют состав

агентов при проектировании, не накапливают опыт между сеансами и располагают лишь неформальным описанием процесса.

3. Исследованы среды моделирования для обучения агентов и спектр применяемых алгоритмов обучения. Выявлено, что имеющиеся среды дают агенту опыт взаимодействия с моделью инфраструктуры, однако не проверяют итог заверщенного сеанса по истинному состоянию экземпляра, известному по построению. Показано, что значительная размерность пространства действий затрудняет применение классических алгоритмов обучения с подкреплением. Наиболее перспективным признано сочетание генеративных языковых моделей, привносящих способность к рассуждению, с обучением с подкреплением по оценке заверщенного сеанса, обеспечивающим поступательное повышение качества решений без экспертной разметки.

4. На основании проведенного анализа выявлен комплекс из трех нерешенных научных задач, составляющих предмет диссертационного исследования. Не разработана математическая модель многоагентной оценки работоспособности при неполноте сведений о среде с композитным пространством состояний, множеством допустимых действий, зависящим от накопленных сведений, и оценкой завершенной трассы вместо пошагового вознаграждения. Не разработана параметрическая модель архитектуры со структурой, порождаемой решениями самих агентов под состав исследуемого комплекса при постоянной глубине графа делегирования. Не разработан алгоритм кросс-уровневого обучения по оценкам завершенных сеансов в среде с верифицируемой обратной связью, совместно улучшающий политики всех уровней иерархии по единому целевому сигналу, формируемому критериальной моделью качества функционирования с весовыми коэффициентами, изменяемыми по итерациям обучения, при отборе элитного множества трасс по адаптивному пороговому значению и разнообразию трасс, создаваемом температурой генерации.

5. Установлено, что сформулированные задачи взаимно обусловлены. Математическая модель лежит в основе архитектуры, поскольку структура системы входит в состояние модели и порождается допустимыми действиями

агентов. Модель архитектуры задает алгоритму обучения разбиение обучающего материала по уровням иерархии и единую модель. Оценка завершенной трассы служит алгоритму целевым сигналом, а среда с верифицируемой обратной связью замыкает контур обучения без участия эксперта. Их совокупное решение составляет научную основу для создания самообучающейся многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов.

Результаты анализа современного состояния, формализация программно-аппаратного комплекса как объекта исследования и обоснование выбора многоагентного подхода получены автором лично и были частично опубликованы в [8; 9; 10; 11; 46; 74], в том числе в соавторстве (вклад автора более 80 %).

2 Модели многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов

2.1 Формальная постановка задачи оценки работоспособности распределенных программно-аппаратных комплексов

Автоматизированная оценка работоспособности распределенных программно-аппаратных комплексов представляет собой задачу последовательного принятия решений в условиях значительной неопределенности. Разрабатываемая многоагентная интеллектуальная система, именуемая далее агентской системой, обладает лишь частичной информацией о состоянии исследуемого комплекса, а каждое ее действие приносит новые, зачастую неполные сведения и потенциально изменяет наблюдаемую картину.

Распределенный программно-аппаратный комплекс определяется автором как сложная техническая система, состоящая из территориально рассредоточенных вычислительных узлов, связанных сетевой инфраструктурой, на которых функционируют разнородные программные службы, управляемые множеством конфигурационных параметров.

Подобная система обладает четырьмя существенными свойствами. Территориальная распределенность выражается в том, что вычислительные узлы рассредоточены в сетевой инфраструктуре и связаны множеством соединений, актуальное состояние которых заранее не известно агентской системе и уточняется в ходе обследования. Гетерогенность проявляется в неоднородности аппаратной базы узлов, протоколов взаимодействия, а также типов, версий и конфигураций программных служб. Неполная наблюдаемость заключается в том, что истинное состояние комплекса устанавливается поступательно, по мере получения результатов диагностических измерений. Динамичность означает, что текущее состояние комплекса изменяется во времени вследствие штатного функционирования, действий администраторов и внешних факторов. Отказывают

и восстанавливаются узлы и службы, колеблются характеристики соединений, перенастраиваются значения конфигурационных параметров.

Формально распределенный программно-аппаратный комплекс определяется как кортеж (2.1).

$$K = \langle N, E, \Sigma, C \rangle, \quad (2.1)$$

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 программного обеспечения 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; каждая служба σ_k характеризуется типом, версией и множеством предоставляемых интерфейсов;

$C = \{c_1, c_2, \dots, c_{|C|}\}$ — множество конфигурационных параметров: выделенные вычислительные ресурсы узлов, параметры сетевых соединений и маршрутизации, конфигурации программных служб.

Состав комплекса агентской системе заранее не известен и устанавливается в ходе обследования. В пределах сеанса обследования — одного заверщенного цикла работы агентской системы от постановки задачи до вычисления итоговой оценки — состав считается неизменным, а изменчивость комплекса относится к его состоянию, то есть к фактическим режимам работы узлов, соединений и служб и текущим значениям конфигурационных параметров.

Формальное представление комплекса приведено на рисунке 2.1.

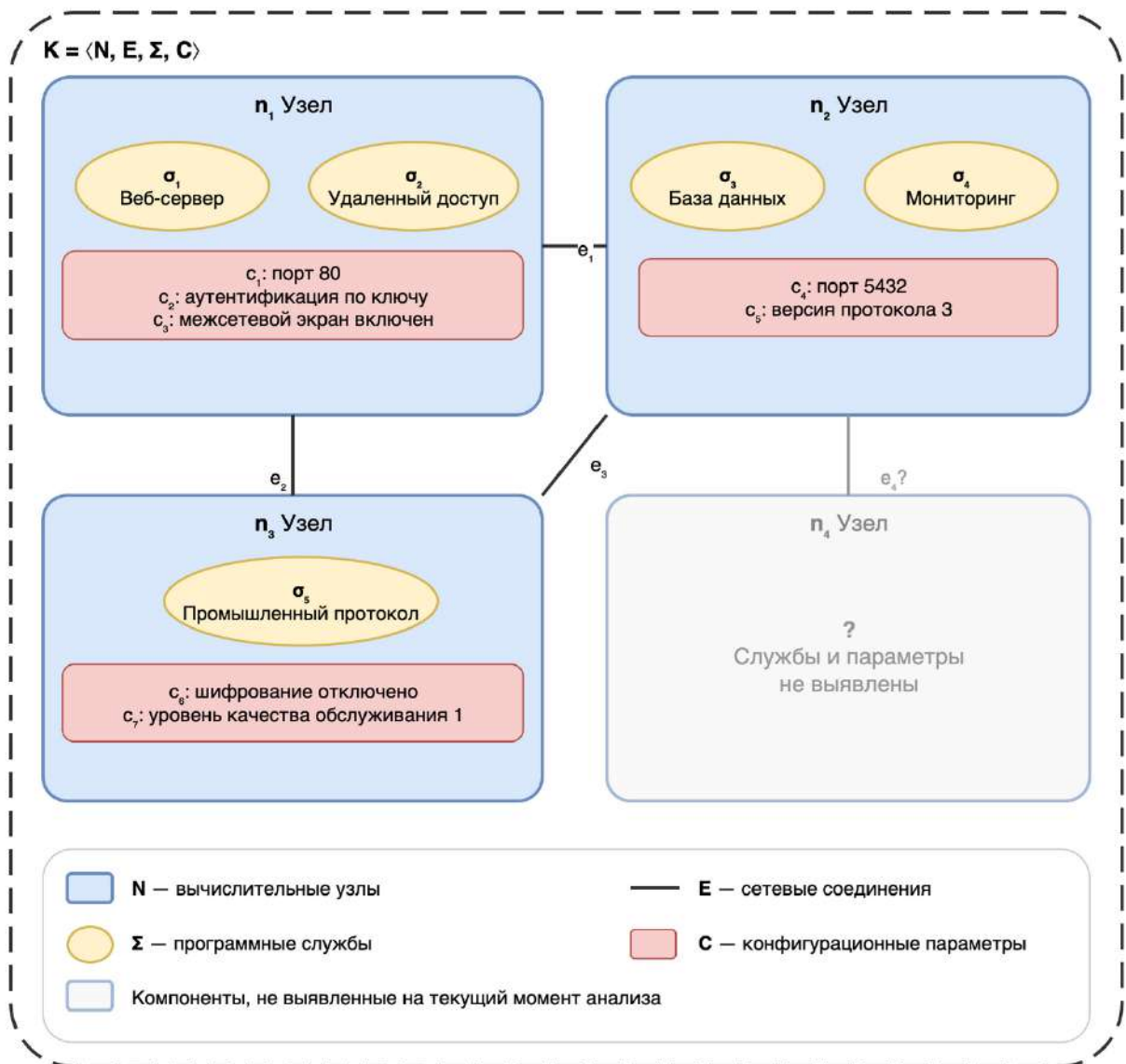


Рисунок 2.1 – Формальное представление распределенного программно-аппаратного комплекса

Целевой характеристикой обследования принимается работоспособность, то есть способность комплекса выполнять заданные функции с показателями, соответствующими установленным требованиям.

Для распределенного комплекса типична частичная деградация, при которой недоступность отдельного узла или службы снижает темп обработки заявок, но не прекращает выполнения функций комплекса в целом. Двухзначная работоспособность классической теории надежности, формализуемая бинарной структурной функцией со значениями $\{0, 1\}$, таких режимов не различает. Поэтому работоспособность распределенного комплекса вводится как мера со значениями в

отрезке $[0, 1]$, выражающая степень выполнения требований на всем диапазоне состояний. Переход от двузначной характеристики к мере известен для многоуровневых систем и для единых мер производительности и надежности деградирующих систем. Автором строится конструкция меры под задачу обследования, состав показателей, степень соответствия требованиям и способ их объединения в одну величину.

Мера строится на составе контролируемых показателей нижнего уровня, объединяемых в одну величину. Контролируемым показателем называется в том смысле, что его значение определяется составом и текущим состоянием самого комплекса и устанавливается обследованием.

Показатели измеряются на уровне комплекса в целом, на уровне вычислительного узла, на уровне сетевого соединения и на уровне программной службы. Поэлементная величина определена для каждого экземпляра своего уровня и записывается с указанием элемента, например $x_3(\sigma_k)$ — время отклика службы σ_k . Запись без аргумента обозначает величину уровня комплекса; для поэлементных показателей это агрегат — сводное значение по всем обнаруженным экземплярам, вид которого закреплен в наименовании показателя. Состав показателей систематизирован автором и представлен в таблице 2.1.

Таблица 2.1 – Контролируемые показатели нижнего уровня распределенного программно-аппаратного комплекса

Обозначение	Наименование показателя	Единица измерения	Уровень измерения	Характер
x_1	Число вычислительных узлов	шт.	комплекс	структурный
x_2	Наименьшая пропускная способность сетевого соединения	Мбит/с	соединение	паспортный
x_3	Наибольшее время отклика программной службы	мс	служба	эксплуатационный
x_4	Наименьшая производительность процессора узла	млн операций/с	узел	паспортный

Обозначение	Наименование показателя	Единица измерения	Уровень измерения	Характер
x_5	Наименьший объем оперативной памяти узла	ГБ	узел	паспортный
x_6	Наименьший объем дисковой памяти узла	ГБ	узел	паспортный
x_7	Наибольшая загрузка процессора узла	%	узел	эксплуатационный
x_8	Наибольшая загрузка оперативной памяти узла	%	узел	эксплуатационный
x_9	Наибольшая доля потерь пакетов в соединении	%	соединение	эксплуатационный
x_{10}	Доля доступных программных служб	%	служба	эксплуатационный
x_{11}	Интенсивность обработки заявок	заявок/с	комплекс	производительностный
x_{12}	Доля задач, решаемых за директивное время	%	комплекс	производительностный

Производительностные показатели x_{11} и x_{12} измеряют выполнение комплексом его функций непосредственно, темп работы с поступающим потоком заявок и своевременность обработки. Структурный показатель x_1 совпадает с числом обнаруженных узлов $|N|$, паспортные показатели x_2 , x_4 – x_6 характеризуют ресурсы, выделенные элементам комплекса, эксплуатационные x_3 , x_7 – x_{10} возникают только при функционировании и характеризуют его текущие режимы.

Каждому показателю сопоставляется требование x_j^* — положительное значение, отделяющее соответствие назначению комплекса от несоответствия. Требования устанавливаются проектной документацией либо заданием на обследование. Требования вида «не более» установлены для показателей x_3 , x_7 – x_9 , остальные требования имеют вид «не менее».

Вид агрегата закреплён в наименовании показателя и определяется направлением требования. При требовании вида «не более» берется наибольшее из поэлементных значений, при требовании вида «не менее» — наименьшее, поэтому выполнение требования агрегатом означает его выполнение каждым элементом. При требованиях, различающихся по ролям узлов, поэлементные значения

предварительно нормируются на поэлементные требования. Доля доступных служб x_{10} вычисляется отношением числа доступных служб к числу обнаруженных, а время отклика x_3 вычисляется по доступным службам, поскольку у недоступной службы времени отклика не существует и сам факт недоступности учтен показателем x_{10} . Недоступность отдельной службы поэтому не обнуляет агрегатов, и частичная деградация остается различимой на уровне комплекса.

Степень соответствия показателя требованию вычисляется как (2.2).

$$g_j(x_j) = \begin{cases} \min(1, x_j/x_j^*), & \text{требование вида «не менее»;} \\ \min(1, x_j^*/x_j), & \text{требование вида «не более», } x_j > 0; \\ 1, & \text{требование вида «не более», } x_j = 0 \end{cases} \quad (2.2)$$

где

$g_j(x_j)$ — степень соответствия j -го показателя требованию, безразмерная величина;

x_j — текущее значение j -го показателя уровня комплекса в единицах измерения таблицы 2.1; для показателей уровня элемента подставляется агрегат;

x_j^* — требование к j -му показателю в тех же единицах.

Значения $g_j(x_j)$ лежат в отрезке $[0, 1]$. Единица соответствует выполнению требования, значения меньше единицы выражают недовыполнение в долях требования, а нулевая степень достигается только на ветви «не менее» и отвечает полному отсутствию измеряемой способности. Усечение $\min(1, \cdot)$ означает, что превышение требования дополнительного вклада в меру не дает.

Конструкция (2.2) составлена из стандартных приемов количественной оценки качества. Отношение фактического значения к требуемому и обратное отношение для показателей противоположного направления образуют относительные показатели дифференциального метода квалиметрии, а нормировка откликов в отрезок $[0, 1]$ с насыщением за целевым значением лежит в основе функций желательности Харрингтона.

Для показателей-загрузок x_7 и x_8 , ограниченных потолком 100 %, обратное отношение обладает малой различающей способностью. При требовании «не более 80 %» степень соответствия не опускается ниже 0,8 даже при полной загрузке. Малая чувствительность принята осознанно. Показатели-загрузки удерживают в

мере сигнал о перегрузке ресурсов, а тонкое различие деградировавших режимов обеспечивают производительные показатели, непосредственно реагирующие на последствия перегрузки.

Мера работоспособности комплекса определяется как взвешенное геометрическое среднее степеней соответствия (2.3).

$$W(K) = \prod_j g_j(x_j)^{\lambda_j}, \quad (2.3)$$

где

$W(K)$ — мера работоспособности комплекса K , безразмерная величина;

$g_j(x_j)$ — степени соответствия показателей требованиям, вычисляемые по (2.2);

λ_j — весовые коэффициенты показателей, $\lambda_j > 0$, $\sum_j \lambda_j = 1$; произведение и сумма распространяются на все показатели таблицы 2.1.

Значения $W(K)$ лежат в отрезке $[0, 1]$. Единица достигается при полном выполнении всех требований, причем благодаря усечению в (2.2) запас по одному показателю недовыполнения другого не компенсирует. Ноль означает полное несоответствие хотя бы по одному показателю с требованием вида «не менее», промежуточные значения выражают степень частичной работоспособности. Недоступность отдельной службы мере не обнуляет, поскольку по построению агрегатов проявляется снижением x_{10} и производительных показателей, то есть частичной деградацией.

Весовой коэффициент λ_j имеет смысл эластичности меры по степени соответствия. Улучшение степени g_j на 1 % изменяет меру на λ_j %. Значения весов могут назначаться методом анализа иерархий с контролем согласованности парных сравнений при настройке агентской системы под класс обследуемых комплексов, а ограничение $\lambda_{11} + \lambda_{12} > 0,5$ закрепляет приоритет производительных показателей и заранее сужает произвол назначения.

В испытательной среде, то есть в программном окружении, порождающем экземпляры комплекса с заданными составом, состоянием и значениями конфигурационных параметров, истинное состояние каждого экземпляра известно по построению, и истинное значение $W(K)$ вычисляется по нему непосредственно

по (2.2) и (2.3). Для действующего комплекса истинные значения показателей заранее не известны, и обследование дает измеренные значения $\hat{x}_j$ — оценки истинных значений, получаемые чтением конфигурационных данных и активными диагностическими измерениями.

Целью обследования является построение оценки $\hat{W}(K)$ меры работоспособности. Агентская система устанавливает состав элементов комплекса, измеряет контролируемые показатели и вычисляет оценку подстановкой измеренных значений в те же соотношения (2.2) и (2.3), $\hat{W}(K) = \prod_j g_j(\hat{x}_j)^{\lambda_j}$. Расхождение оценки с истинным значением порождается необнаруженными элементами комплекса, неполнотой и зашумленностью измерений, а также устареванием накопленных данных к моменту вычисления меры. Качество решения задачи характеризуется близостью оценки $\hat{W}(K)$ к истинному значению $W(K)$. В испытательной среде эта близость измерима точно, что составляет основу обучения агентской системы на верифицируемой обратной связи, то есть обратной связи, вычислимой сопоставлением оценки с истинным значением меры без участия эксперта.

Оценка в решаемой задаче ведется в двух контурах, относящихся к различным объектам, и разделение исследуемого комплекса и агентской системы как объекта и инструмента формализуется композитным пространством состояний. Контур объекта принадлежит комплексу, мера $W(K)$ характеризует работоспособность комплекса и существует независимо от способа обследования. Контур системы принадлежит инструменту, качество функционирования агентской системы оценивается по итогам завершеного обследования отдельной критериальной моделью качества функционирования. Контуры измеряют разное и друг друга не подменяют, некачественное обследование оставляет работоспособность комплекса неизменной и лишь снижает точность оценки $\hat{W}(K)$, а связь контуров устанавливается верификацией, то есть сопоставлением оценки с истинным значением меры.

Территориальная распределенность и гетерогенность комплекса обуславливают многоагентную организацию системы, неполная наблюдаемость

исключает модели с полным доступом к состоянию среды, а динамичность требует стохастического описания переходов. Простейшая модель последовательного принятия решений, марковский процесс принятия решений, предполагает, что агент в каждый момент времени обладает полной и достоверной информацией о состоянии среды [125]. Однако агентская система исчерпывающими сведениями о текущем состоянии комплекса не располагает, и сведения уточняются по результатам диагностических измерений. Частично наблюдаемый марковский процесс принятия решений обобщает марковский процесс на случай, когда агенту доступны лишь косвенные наблюдения, по которым строятся вероятностные предположения об истинном состоянии среды [89].

Разрабатываемая система при этом строится как многоагентная, и каждый из специализированных агентов обладает только локальной информацией. Стохастические игры моделируют множество агентов при стохастичности переходов, однако их классическая постановка предполагает полную наблюдаемость состояния, а интересы сторон в общем случае не совпадают [121]. Кооперативный случай с общим вознаграждением при частичной наблюдаемости выделен в самостоятельный аппарат — децентрализованный частично наблюдаемый марковский процесс принятия решений [110; 127]. Этот аппарат принимается в работе, поскольку одновременно учитывает частичную наблюдаемость, множественность агентов, стохастичность переходов и кооперативный характер взаимодействия.

Децентрализованный частично наблюдаемый марковский процесс принятия решений определяется кортежем (2.4).

$$M = \langle I, S, \{A_i\}_{i \in I}, T, R, \{\Omega_i\}_{i \in I}, O, \gamma, b_0 \rangle, \quad (2.4)$$

где

I — конечное множество агентов, $|I| = n$; агенты обозначаются $\alpha_1, \dots, \alpha_n$, каждый из них является автономной программной сущностью, принимающей решения на основе локальной информации;

S — конечное множество состояний среды, охватывающее возможные состояния исследуемого комплекса;

$\{A_i\}_{i \in I}$ — множества действий для каждого агента α_i ; совместное действие всех агентов определяется как кортеж $a = (a_1, \dots, a_n) \in A = A_1 \times \dots \times A_n$;

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

R — функция вознаграждения, сопоставляющая каждой паре из состояния и совместного действия вещественное число; вознаграждение едино для всех агентов, что выражает кооперативный характер модели;

$\{\Omega_i\}_{i \in I}$ — 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 ;

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 обследования.

Схема взаимодействия агентской системы со средой приведена на рисунке 2.2.

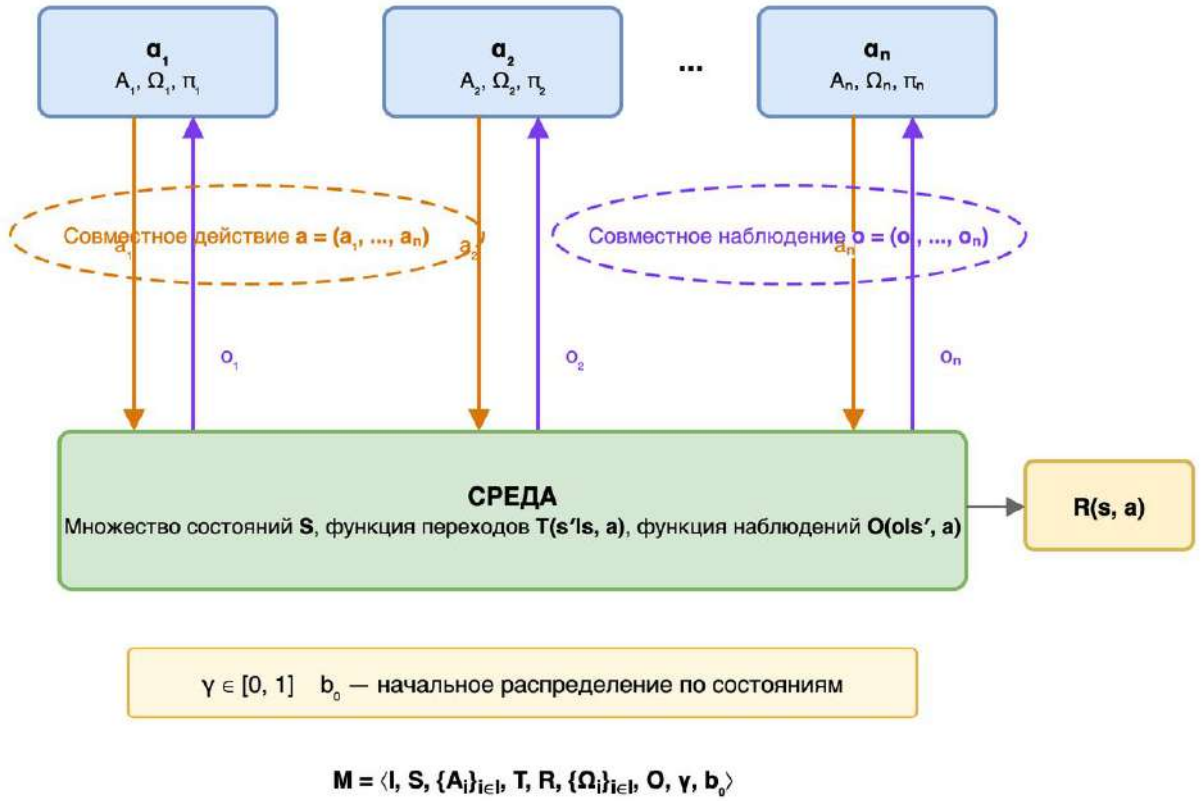


Рисунок 2.2 – Графическое представление децентрализованного частично наблюдаемого марковского процесса принятия решений как схемы взаимодействия агентской системы с исследуемой средой в условиях неопределенности

Агенты формируют совместное действие из индивидуальных компонент, среда возвращает совместное наблюдение, распределяемое по агентам покомпонентно, а сигнал вознаграждения передается по отдельному каналу.

Функция переходов определяет вероятность перехода системы из состояния s в состояние s' при выполнении совместного действия a (2.5).

$$T(s'|s, a) = P(s_{t+1} = s' | s_t = s, a_t = a), \quad (2.5)$$

где

$P(\cdot|\cdot)$ — условная вероятность;

s_t, s_{t+1} — состояния в моменты времени t и $t + 1$;

a_t — действие, выполненное в момент t .

Функция переходов удовлетворяет условию нормировки при всех $s \in S$ и $a \in A$ (2.6).

$$\sum_{s' \in S} T(s'|s, a) = 1. \quad (2.6)$$

Переходы могут быть детерминированными, как чтение конфигурационного файла, оставляющее состояние комплекса неизменным, либо стохастическими, когда результат действия зависит от не наблюдаемых агентом факторов, таких как сетевая нагрузка или изменение конфигурации маршрутизации.

Функция наблюдений определяет вероятность получения совместного наблюдения o после перехода в состояние s' в результате выполнения действия a (2.7).

$$O(o|s', a) = P(o_{t+1} = o | s_{t+1} = s', a_t = a), \quad (2.7)$$

где

o_{t+1} — наблюдение, полученное в момент времени $t + 1$.

Функция наблюдений удовлетворяет условию нормировки, аналогичному (2.6), а каждый агент α_i получает только свою компоненту o_i совместного наблюдения. При обследовании распределенных комплексов наблюдениями служат ответы служб на диагностические запросы, метрики состояния узлов, данные журналов работы служб и результаты анализа конфигурационных файлов; наблюдения могут быть неполными, зашумленными либо отсутствовать вовсе.

Истинное состояние s недоступно агентам напрямую, поэтому в централизованном описании процесса вводится состояние веры b — распределение вероятностей по множеству состояний, значение $b(s)$ которого выражает вероятность нахождения среды в состоянии s . После выполнения действия a и получения наблюдения o состояние веры обновляется по правилу Байеса (2.8).

$$b'(s') = \eta \cdot O(o|s', a) \cdot [\sum_{s \in S} T(s'|s, a) \cdot b(s)], \quad (2.8)$$

где

$b'(s')$ — обновленное состояние веры;

η — нормализующий множитель, обеспечивающий $\sum_{s' \in S} b'(s') = 1$;

$b(s)$ — предыдущее состояние веры.

Каждое новое наблюдение перераспределяет вероятности между состояниями, и больший вес получают состояния, совместимые с полученными диагностическими данными.

В централизованной постановке с единственным агентом состояние веры служит достаточной статистикой истории взаимодействия, и политика

определяется как отображение вер в действия. В децентрализованной постановке отдельный агент не может поддерживать совместную веру, поскольку обновление (2.8) требует совместного действия a и совместного наблюдения o , а агенту доступны только собственные компоненты a_i и o_i . Локальная политика поэтому строится на локальной истории наблюдений и действий, и политика агента α_i определяется как отображение (2.9).

$$\pi_i: (\Omega_i \times A_i)^* \times \Omega_i \rightarrow \Delta(A_i), \quad (2.9)$$

где

π_i — политика агента α_i , определяющая правило выбора действий;

$(\Omega_i \times A_i)^*$ — множество всех конечных последовательностей пар наблюдений и действий агента α_i , где символ $*$ обозначает замыкание Клини; элемент этого множества представляет собой историю взаимодействия агента со средой;

Ω_i — множество наблюдений агента α_i ; последний сомножитель $\times \Omega_i$ отвечает текущему наблюдению $o_i \in \Omega_i$, полученному перед выбором действия;

$\Delta(A_i)$ — симплекс распределений вероятностей над множеством действий A_i .

История агента к шагу t имеет явный вид $h_i^t = (o_i^1, a_i^1, \dots, o_i^{t-1}, a_i^{t-1}, o_i^t)$ и складывается в решаемой задаче из выполненных агентом диагностических запросов и полученных на них ответов. На первом шаге прошлых пар еще нет, и история состоит из единственного наблюдения, а пустая последовательность пар покрывается замыканием Клини. Совместная политика $\pi = (\pi_1, \dots, \pi_n)$ определяет поведение всей агентской системы.

В классической постановке целью является нахождение оптимальной совместной политики, максимизирующей математическое ожидание суммарного дисконтированного вознаграждения (2.10).

$$\pi^* = \operatorname{argmax}_{\pi} E_{\pi} [\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) | b_0], \quad (2.10)$$

где

π^* — оптимальная совместная политика;

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

γ^t — коэффициент дисконтирования в степени t , уменьшающий вклад будущих вознаграждений; при бесконечном горизонте предполагается $\gamma < 1$;

$R(s_t, a_t)$ — вознаграждение, полученное в момент t при нахождении в состоянии s_t и выполнении действия a_t ;

b_0 — начальное состояние веры, относительно которого вычисляется ожидание.

Формулировка (2.10) предполагает, что вознаграждение вычисляется на каждом шаге взаимодействия. В решаемой задаче качество обследования устанавливается лишь по завершении сеанса, что требует замены пошагового вознаграждения оценкой завершённого сеанса обследования.

Функция ценности $V^\pi(b)$ определяет ожидаемое суммарное вознаграждение при следовании политике π из состояния веры b (2.11) и даёт критерий сравнения политик из любой точки процесса.

$$V^\pi(b) = E_\pi[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) | b_0 = b], \quad (2.11)$$

где

$b_0 = b$ — условие, что начальное состояние веры равно b .

Максимизируемое в (2.10) выражение есть значение функции ценности в начальном состоянии веры, $\pi^* = \operatorname{argmax}_\pi V^\pi(b_0)$. Оптимальная функция ценности удовлетворяет уравнению Беллмана (2.12).

$$V^*(b) = \max_{a \in A} [\sum_{s \in S} b(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o | b, a) \cdot V^*(b'_{a,o})], \quad (2.12)$$

где

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

$\Omega = \Omega_1 \times \dots \times \Omega_n$ — множество совместных наблюдений;

$P(o | b, a)$ — вероятность получить наблюдение o при состоянии веры b и выполнении действия a , определяемая через функции переходов T и наблюдений O ;

$V^*(b'_{a,o})$ — оптимальная функция ценности состояния веры $b'_{a,o}$, полученного пересчетом (2.8) после действия a и наблюдения o .

Уравнение Беллмана связывает оптимальную ценность текущего состояния веры с ценностями состояний, достижимых из него за один шаг; на этой рекурсии строятся методы динамического программирования. Однако в децентрализованной постановке совместная вера отдельным агентам недоступна, а пошаговое

вознаграждение, на котором стоят соотношения (2.10)–(2.12), в задаче оценки работоспособности отсутствует. Необходимо модифицировать классический аппарат под решаемую задачу.

2.2 Модель многоагентной оценки работоспособности распределенных программно-аппаратных комплексов

Классический аппарат строится на следующих трех допущениях [110]. Состояние описывает исключительно среду, состав допустимых действий фиксирован и от состояния не зависит, а вознаграждение вычисляется после каждого шага. В задаче оценки работоспособности распределенных программно-аппаратных комплексов состояние процесса включает и накопленные агентами диагностические сведения, состав допустимых действий изменяется по мере получения новой информации, а результат обследования формируется по совокупности выполненных действий и на отдельном шаге определен быть не может.

Автором предлагается модификация трех компонентов кортежа M . В модель вводится композитное пространство состояний, фиксированный состав действий заменяется стационарным правилом допустимых действий, а пошаговое вознаграждение — оценкой завершенной трассы. Сопутствующие уточнения функции наблюдений и начального состояния веры согласуют остальные компоненты кортежа с композитной структурой состояния. Корректность модификации устанавливается по ходу построения.

Действия агентов изменяют и наблюдаемую картину комплекса, и информационные возможности самой агентской системы. Пространство состояний поэтому определяется композитным (2.13).

$$S = S_{env} \times S_{agent}, \quad (2.13)$$

где

S_{env} — множество состояний исследуемого комплекса, описываемых фактическими режимами работы узлов, соединений и служб и актуальными значениями конфигурационных параметров;

S_{agent} — множество информационных состояний агентской системы: накопленные диагностические сведения, результаты анализа конфигураций, остаток бюджета действий — наибольшего допустимого числа шагов сеанса.

Элементы композитного пространства записываются парами $s = (S_{env}, S_{agent})$; номер шага для компонент указывается в скобках, а композитное состояние шага t обозначается нижним индексом, $s_t = (S_{env}(t), S_{agent}(t))$. Признак относится к компоненте S_{env} , если его значение определяется самим комплексом безотносительно к процессу обследования, и к компоненте S_{agent} , если он фиксирует знания агентской системы или ход обследования.

Состав компоненты S_{env} подчинен требованию полноты — она обязана содержать все признаки комплекса, от которых зависят исходы диагностических действий. При невключенном влияющем признаке вероятности исходов при совпадающих состояниях оказывались бы различными, то есть закона переходов над таким состоянием не существовало бы.

Однородность процесса принимается как допущение модели. Закон переходов T_{env} в пределах сеанса не изменяется, и вероятности переходов не зависят от номера шага. Динамичность комплекса при этом выражается движением состояния — отказы и восстановления узлов, перенастройка конфигурационных параметров и колебания характеристик соединений суть случайные переходы, разыгрываемые по этому закону.

Авторским отличием от исходной постановки является введение компоненты S_{agent} , отражающей внутреннее информационное состояние агентской системы. Получение журналов узла не изменяет состояния комплекса, но обогащает информационный ресурс системы. Эволюция информационной компоненты задается правилом записи (2.14).

$$s_{agent}(t + 1) = u(s_{agent}(t), a_t, o_{t+1}), \quad (2.14)$$

где

u — фиксированное детерминированное правило записи, единое для всех шагов сеанса.

Правило записи ведет ограниченную сводку, но не полную историю наблюдений. Оно дописывает новые диагностические сведения, помечает устаревшие записи, сжимает ранние при превышении допустимого объема хранения и списывает выполненный шаг с бюджета сеанса. Аргументами правила служат только текущее информационное состояние, действие и наблюдение, поэтому накопительный характер обследования и снижение актуальности сведений выражаются внутри правила u без обращения к предыстории процесса.

Конечность модели принимается как допущение — конечны множества действий A_i и наблюдений Ω_i с фиксированным составом операций и различными ответов, конечна компонента S_{env} с конечными наборами значений признаков, конечна и компонента S_{agent} , объем сводки которой ограничен правилом записи.

Функция наблюдений в композитной модели обуславливается компонентой комплекса и принимает вид (2.15).

$$O(o|s'_{env}, a) = P(o_{t+1} = o|s_{env}(t+1) = s'_{env}, a_t = a), \quad (2.15)$$

где

s'_{env} — компонента комплекса состояния, в которое перешла система.

Информационная компонента является собственной памятью агентской системы и наблюдений не порождает. Начальное состояние веры b_0 также задает вероятности только по компоненте комплекса и сосредоточено на единственном начальном значении информационной компоненты. В начальный момент хранилище сведений пусто, бюджет действий полон, и значение $s_{agent}(0)$ известно агентской системе точно.

В решаемой задаче состав осмысленных операций определяется объемом собранных сведений — углубленная диагностика службы определена лишь после снятия первичных показателей узла и самой службы, то есть после фиксации этих сведений в информационной компоненте. Автором вводится стационарное правило допустимых действий (2.16).

$$A(s) = f(s_{agent}), \quad (2.16)$$

где

$A(s)$ — множество совместных действий, допустимых в состоянии $s = (S_{env}, s_{agent})$, сохраняющее структуру декартова произведения $A(s) = A_1(s) \times \dots \times A_n(s)$, где $A_i(s)$ определяется той частью информационного состояния, которая доступна агенту α_i ; состав доступных агентам частей задается механизмами информационного пространства агентской системы;

f — фиксированное, не зависящее от времени отображение $f: S_{agent} \rightarrow 2^A \setminus \{\emptyset\}$, сопоставляющее каждому информационному состоянию непустое подмножество множества совместных действий.

Состав допустимых операций изменяется только записью сведений в информационную компоненту. Операция становится допустимой, когда правило записи (2.14) фиксирует открывающее ее сведение, и перестает быть допустимой, когда зафиксировано исключающее. Изменчивость допустимого множества по ходу сеанса совместима со стационарностью правила — на каждом шаге применяется одно и то же отображение f , изменяется лишь его аргумент $s_{agent}(t)$. В отличие от систем с переменной структурой управления, где со временем изменяется сам закон, правило (2.16) фиксировано на протяжении сеанса.

Пополнение информационной компоненты способно и расширять, и сужать допустимое множество. Обнаружение соединений узла открывает операции опроса соседних с ним узлов, а фиксация недоступности службы исключает операции ее углубленной диагностики. Допустимость при этом отделена от исхода — правило (2.16) определяет, какие операции принимаются к исполнению, а результат принятой операции, в том числе неуспешный, описывается законом переходов и функцией наблюдений. Зависимость правила (2.16) только от информационной компоненты существенна, поскольку при зависимости от компоненты S_{env} допустимое множество определялось бы ненаблюдаемой величиной, недоступной агентской системе.

Наблюдение 2.1 (о роли информационной компоненты). Информационная компонента неустранима из состава состояния — без нее допустимое множество перестает быть функцией состояния. Пусть состояние сведено к компоненте комплекса, а состав допустимых действий определяется накопленными

сведениями, как того требует существо задачи. Два сеанса, приводящие к одному и тому же состоянию комплекса S_{env} , могут различаться ходом обследования. В первом обнаружены узлы n_1 и n_2 , во втором только n_1 , и диагностика узла n_2 допустима в первом сеансе и недопустима во втором. Одному состоянию комплекса отвечают различные допустимые множества, то есть над S_{env} допустимое множество не является функцией состояния; задача не допускает постановки в виде марковского процесса принятия решений с зависящими от состояния множествами действий, и динамическое программирование над S_{env} лишено основания. Изъятие информационной компоненты закона переходов T_{env} не затрагивает — нарушена только связь допустимости действий с состоянием. Включение информационной компоненты разводит такие сеансы по разным композитным состояниям и восстанавливает зависимость допустимости только от текущего состояния. Тем самым композитное состояние служит условием корректности стационарного правила допустимых действий (2.16).

Всякий сеанс обследования должен завершаться, и модель дополняется условием завершения. Допустимое множество $A(s)$ в каждом состоянии содержит останавливающее совместное действие $\hat{a}$, единое для всех состояний, выбор которого завершает сеанс. Число шагов сеанса ограничено бюджетом N_{max} ; остаток бюджета входит в информационную компоненту состояния, и при его исчерпании допустимое множество сокращается до останавливающего действия. Момент завершения k , равный числу выполненных за сеанс шагов без учета останавливающего, определен при любом поведении системы и удовлетворяет $k \leq N_{max}$. Состояние, в котором выбрано останавливающее действие, называется терминальным. Условие завершения содержательно оправдано конечностью комплекса и объема извлекаемой информации и ограниченностью практического бюджета времени и диагностических запросов.

Совместное распределение нового состояния и наблюдения при текущем состоянии s и действии a принимается за определение одношагового перехода (2.17).

$$P(s', o | s, a) = T_{env}(s'_{env} | s_{env}, a) \cdot O(o | s'_{env}, a) \cdot \mathbf{1}[s'_{agent} = u(s_{agent}, a, o)], \quad (2.17)$$

где

T_{env} — закон переходов состояния комплекса, заменяющий в композитной модели функцию переходов (2.5);

$\mathbf{1}[\cdot]$ — индикатор: информационная компонента принимает то значение, которое выдает правило записи, и никакое другое.

Сомножители правой части (2.17) отвечают последовательным событиям одного шага — переходу компоненты комплекса по закону T_{env} , порождению наблюдения новой компонентой комплекса и пересчету информационной компоненты правилом записи. Покажем, что произведение (2.17) задает распределение вероятностей на парах (s', o) . Правая часть неотрицательна, и проверке подлежит нормировка.

$$\begin{aligned} \sum_{s' \in S} \sum_o P(s', o | s, a) &= \sum_{s'_{env}} \sum_o \sum_{s'_{agent}} T_{env}(s'_{env} | s_{env}, a) \cdot O(o | s'_{env}, a) \cdot \\ \mathbf{1}[s'_{agent} = u(s_{agent}, a, o)] &= \sum_{s'_{env}} T_{env}(s'_{env} | s_{env}, a) \cdot \left(\sum_o O(o | s'_{env}, a) \cdot \right. \\ \left. \left(\sum_{s'_{agent}} \mathbf{1}[s'_{agent} = u(s_{agent}, a, o)] \right) \right) &= \sum_{s'_{env}} T_{env}(s'_{env} | s_{env}, a) \cdot \\ \left(\sum_o O(o | s'_{env}, a) \right) &= \sum_{s'_{env}} T_{env}(s'_{env} | s_{env}, a) = 1. \end{aligned}$$

Сумма по парам записана тройной суммой, затем из внутренних сумм вынесены сомножители, не зависящие от их переменных; скобки фиксируют вложенность, поскольку наблюдение входит в аргумент индикатора и сумма по наблюдениям обязана его охватывать. При каждом наблюдении индикатор отличен от нуля на единственном значении s'_{agent} , вычисляемом правилом записи, и внутренняя сумма равна единице. Остальное сворачивают нормировки функции наблюдений (2.15) при каждом s'_{env} и закона T_{env} .

Формула (2.17) задает отдельный шаг; процесс в целом определяется распределением траекторий. Вероятность полной истории h_t вместе с выбранным действием a_t равна произведению начального распределения, вероятностей выбора действий и одношаговых множителей (2.17) и составляет (2.18).

$$P(h_t, a_t) = b_0(s_0) \cdot \prod_{k=0}^{t-1} [\pi(a_k | h_k) \cdot P(s_{k+1}, o_{k+1} | s_k, a_k)] \cdot \pi(a_t | h_t). \quad (2.18)$$

Полная история $h_t = (s_0, a_0, o_1, s_1, \dots, o_t, s_t)$ перечисляет состояния, действия и наблюдения сеанса от начального момента до текущего; от локальных историй h_i^t отдельных агентов она отличается полнотой. Вероятность $\pi(a_k | h_k)$ отвечает выбору

совместного действия политикой, зависимость выбора от всей накопленной истории допускается. Факторизация повторяет хронологию сеанса, каждый множитель отвечает одному событию.

Утверждение 2.1 (о марковости и однородности композитного процесса). При фиксированных T_{env} , O и u композитный управляемый процесс $\{s_t = (s_{env}(t), s_{agent}(t))\}$ обладает марковским свойством и однороден по времени. Индуцированный закон переходов имеет вид (2.19).

$$T(s'|s, a) = T_{env}(s'_{env}|s_{env}, a) \cdot \sum_{o: u(s_{agent}, a, o) = s'_{agent}} O(o|s'_{env}, a). \quad (2.19)$$

Доказательство. Индуцированный закон переходов описывает распределение нового состояния без учета полученного наблюдения и вычисляется суммированием совместного закона (2.17) по наблюдениям.

$$\begin{aligned} T(s'|s, a) &= \sum_o P(s', o|s, a) = \sum_o T_{env}(s'_{env}|s_{env}, a) \cdot O(o|s'_{env}, a) \cdot \mathbf{1}[s'_{agent} = \\ &u(s_{agent}, a, o)] = T_{env}(s'_{env}|s_{env}, a) \cdot (\sum_o O(o|s'_{env}, a) \cdot \mathbf{1}[s'_{agent} = \\ &u(s_{agent}, a, o)]) = T_{env}(s'_{env}|s_{env}, a) \cdot \sum_{o: u(s_{agent}, a, o) = s'_{agent}} O(o|s'_{env}, a). \end{aligned}$$

Цепочка подставляет произведение (2.17) и выносит переходный сомножитель, не зависящий от переменной суммирования; скобки фиксируют, что сумма по наблюдениям охватывает наблюдательный и индикаторный сомножители. Индикатор обращает в нуль слагаемые с $u(s_{agent}, a, o) \neq s'_{agent}$, остальные входят с единичным множителем, и сумма по всем наблюдениям переходит в сумму по наблюдениям, которые правило записи приводит к значению s'_{agent} . Правая часть совпадает с (2.19), закон переходов построен. Его нормировка следует из нормировки (2.17). Суммирование закона (2.19) по новым состояниям возвращает сумму совместного закона по всем парам, равную единице.

Марковское свойство составляет равенство $P(s_{t+1} = s' | h_t, a_t) = T(s' | s_t, a_t)$ при всякой полной истории. По факторизации распределения траекторий вероятность истории, продолженной наблюдением o' и состоянием s' , равна $P(h_t, a_t) \cdot P(s', o' | s_t, a_t)$; суммирование по наблюдению дает числитель условной вероятности.

$$P(s_{t+1} = s' | h_t, a_t) = \frac{P(h_t, a_t) \cdot \sum_{o'} P(s', o' | s_t, a_t)}{P(h_t, a_t)} = \sum_{o'} P(s', o' | s_t, a_t) = T(s' | s_t, a_t).$$

Сокращается вся предыстория — начальное распределение, вероятности прошлых выборов и переходов, включая множитель $\pi(a_t | h_t)$, — поэтому марковское свойство выполняется и при политике, зависящей от всей истории.

Остаток зависит только от текущей пары, поскольку одношаговый множитель (2.17) есть функция (s_t, a_t) . Однородность следует из отсутствия номера шага в (2.17) и (2.19), все шаги сеанса подчинены одной формуле. Что и требовалось доказать.

Неопределенность в композитной модели распределена по компонентам несимметрично — информационная компонента известна агентской системе точно, и неопределенность сохраняется только по компоненте комплекса. Точный смысл несимметрии придает устройство состояния веры. Вера в композитной модели пересчитывается как условное распределение нового состояния при выполненном действии a_t и полученном наблюдении o_{t+1} , предыдущая вера взвешивает одношаговые множители (2.17). Запись принимается за определение пересчета веры.

$$b_{t+1}(s') = \eta \cdot \sum_{s \in S} b_t(s) \cdot P(s', o_{t+1} | s, a_t).$$

Множитель η нормирует пересчитанную веру на единицу. Для классической модели (2.4) выписанное определение совпадает с правилом (2.8) — там одношаговый переход распадается в произведение функций переходов (2.5) и наблюдений (2.7), и вынос наблюдательного множителя за знак суммы дает (2.8). В композитной модели одношаговый множитель задан произведением (2.17), и пересчет наследует его структуру.

Утверждение 2.2 (о расщеплении состояния веры). При всяком шаге t существует распределение $\tilde{b}_t \in \Delta(S_{env})$, для которого состояние веры имеет вид (2.20).

$$b_t(s_{env}, s_{agent}) = \tilde{b}_t(s_{env}) \cdot \mathbf{1}[s_{agent} = s_{agent}(t)], \quad (2.20)$$

где

$\tilde{b}_t$ — распределение веры по компоненте комплекса;

$s_{agent}(t)$ — фактическое значение информационной компоненты на шаге t , определяемое действиями и наблюдениями сеанса по правилу записи (2.14) от начального значения $s_{agent}(0)$.

Доказательство. Индукция ведется по номеру шага. В начальный момент вера имеет вид (2.20), поскольку состояние веры b_0 сосредоточено на единственном

значении $s_{\text{agent}}(0)$. Распределение $\tilde{b}_0$ задается равенством $\tilde{b}_0(s_{\text{env}}) = b_0(s_{\text{env}}, s_{\text{agent}}(0))$; его полная масса равна единице, и оно принадлежит $\Delta(S_{\text{env}})$.

Пусть на шаге t вера имеет вид (2.20). Подстановка веры в определение пересчета дает цепочку

$$\begin{aligned} b_{t+1}(s') &= \eta \cdot \sum_{s_{\text{env}}} \sum_{s_{\text{agent}}} b_t(s_{\text{env}}, s_{\text{agent}}) \cdot P(s', o_{t+1} | (s_{\text{env}}, s_{\text{agent}}), a_t) = \eta \cdot \sum_{s_{\text{env}}} \tilde{b}_t(s_{\text{env}}) \cdot P(s', o_{t+1} | (s_{\text{env}}, s_{\text{agent}}(t)), a_t) \\ &= \eta \cdot \sum_{s_{\text{env}}} \tilde{b}_t(s_{\text{env}}) \cdot T_{\text{env}}(s'_{\text{env}} | s_{\text{env}}, a_t) \cdot O(o_{t+1} | s'_{\text{env}}, a_t) \cdot \mathbf{1}[s'_{\text{agent}} = u(s_{\text{agent}}(t), a_t, o_{t+1})] \\ &= \eta \cdot O(o_{t+1} | s'_{\text{env}}, a_t) \cdot \left(\sum_{s_{\text{env}}} \tilde{b}_t(s_{\text{env}}) \cdot T_{\text{env}}(s'_{\text{env}} | s_{\text{env}}, a_t) \right) \cdot \mathbf{1}[s'_{\text{agent}} = u(s_{\text{agent}}(t), a_t, o_{t+1})]. \end{aligned}$$

Сумма по состояниям записана двойной суммой по компонентам, и внутренняя сумма свернута по предположению индукции — при каждом s_{env} вера b_t отлична от нуля лишь при $s_{\text{agent}} = s_{\text{agent}}(t)$, слагаемые с иными значениями информационной компоненты нулевые, а оставшееся входит с множителем $\tilde{b}_t(s_{\text{env}})$. Затем подставлено произведение (2.17) и из суммы вынесены сомножители, не зависящие от переменной суммирования s_{env} . Наблюдательный свободен от нее вследствие обусловливания (2.15), индикаторный — вследствие детерминированности правила записи, аргументы которого фиксированы; скобка удерживает под знаком суммы сомножители, зависящие от s_{env} .

Аргумент индикатора одинаков во всех слагаемых и по правилу записи (2.14) равен $s_{\text{agent}}(t + 1)$. Сомножители перед индикатором составляют байесовский пересчет вида (2.8) по одной компоненте комплекса — с законом T_{env} в роли функции переходов и функцией наблюдений (2.15); результат пересчета принимается за $\tilde{b}_{t+1}(s'_{\text{env}})$. Итог цепочки принимает вид

$$b_{t+1}(s'_{\text{env}}, s'_{\text{agent}}) = \tilde{b}_{t+1}(s'_{\text{env}}) \cdot \mathbf{1}[s'_{\text{agent}} = s_{\text{agent}}(t + 1)],$$

то есть расщепление (2.20) на шаге $t + 1$ воспроизведено. Принадлежность $\tilde{b}_{t+1}$ симплексу $\Delta(S_{\text{env}})$ следует из нормировки. Вне значения $s'_{\text{agent}} = s_{\text{agent}}(t + 1)$ вера b_{t+1} нулевая, при нем совпадает с $\tilde{b}_{t+1}$, поэтому суммы обеих величин по своим аргументам равны и один множитель η нормирует обе. Что и требовалось доказать.

Расщепление (2.20) ограничивает пространство неопределенности. Вера, номинально принадлежащая симплексу $\Delta(S_{\text{env}} \times S_{\text{agent}})$, пробегает лишь множество $\Delta(S_{\text{env}}) \times S_{\text{agent}}$ — пары из распределения по компоненте комплекса и фактического значения информационной компоненты. Введение информационной компоненты

увеличивает номинальную мощность пространства состояний, но не увеличивает пространства неопределенности.

В исходной модели функция вознаграждения $R(s, a)$ вычисляется после каждого действия, тогда как качество обследования характеризуется близостью итоговой оценки меры к истинному значению и определяется только по совокупности выполненных действий. Пошаговое вознаграждение поэтому заменяется оценкой завершеного обследования. Последовательность состояний, действий и наблюдений, порождаемая агентской системой за один сеанс — от начального состояния до останавливающего действия, — образует трассу исполнения τ ; длина трассы равна моменту завершения k , множество завершённых трасс обозначается D . Целевой функционал задачи ставится над завершёнными сеансами и принимает эпизодическую форму (2.21).

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi} E_{\pi}[R(\tau)|b_0], \quad (2.21)$$

где

Π — множество допустимых совместных политик, то есть политик, распределения которых сосредоточены на допустимых множествах $A(s_i)$;

$R(\tau)$ — оценка завершённой трассы, вещественное число.

Совместная политика в классе Π устроена децентрализованно, каждый агент выбирает свою компоненту действия самостоятельно, по доступным ему сведениям. Запись (2.9) отвечает классической модели, в которой сведения агента исчерпываются наблюдениями; в композитной модели агент располагает и записанными сведениями, поэтому аргумент политики определяется заново. Локальной историей агента α_i к шагу t служит доступная ему часть информационного состояния $s_{\text{agent}}(t)$, обозначаемая h_i^t . В нее входят выполненные агентом действия и полученные на них наблюдения, назначенная ему задача, сведения, доставленные ему механизмами информационного пространства агентской системы, и остаток бюджета действий. Формируется эта часть правилом записи (2.14) наравне с остальным информационным состоянием, и ею же по правилу (2.16) определяется сомножитель $A_i(s)$. Политика π_i отображает локальную историю в распределение над действиями A_i ; совместная политика называется

детерминированной, если каждое распределение $\pi_i(\cdot | h_i)$ вырождено, то есть сосредоточено на единственном действии.

Утверждение 2.3 (о существовании оптимальной совместной политики). Пусть оценка $R(\tau)$ определена на всех завершенных трассах. Тогда максимум в (2.21) достигается, причем достигается на детерминированной совместной политике.

Доказательство. Длина трассы не превосходит $N_{\max}$, а состояние, действие и наблюдение каждого шага принадлежат конечным множествам, поэтому множество завершенных трасс конечно. Оценка $R(\tau)$ принимает конечное число значений, и ожидаемая оценка определена и конечна при всякой политике.

Завершенная трасса τ состоит из полной истории h_k и останавливающего действия a_k , поэтому ее вероятность равна значению распределения траекторий $P(h_k, a_k)$. Ожидаемая оценка принимает вид

$$E_{\pi}[R(\tau)|b_0] = \sum_{\tau \in D} R(\tau) \cdot P(h_k, a_k) = \sum_{\tau \in D} R(\tau) \cdot b_0(s_0) \cdot (\prod_{t=0}^{k-1} [\pi(a_t|h_t) \cdot P(s_{t+1}, o_{t+1}|s_t, a_t)]) \cdot \pi(a_k|h_k) = \sum_{\tau \in D} R(\tau) \cdot b_0(s_0) \cdot (\prod_{t=0}^{k-1} P(s_{t+1}, o_{t+1}|s_t, a_t)) \cdot (\prod_{t=0}^k \pi(a_t|h_t)) = \sum_{\tau \in D} R(\tau) \cdot b_0(s_0) \cdot (\prod_{t=0}^{k-1} P(s_{t+1}, o_{t+1}|s_t, a_t)) \cdot (\prod_{t=0}^k \prod_{i \in I} \pi_i(a_i^t|h_i^t)).$$

Ожидание раскрыто суммой по завершенным трассам, в которую подставлено распределение траекторий (2.18) при $t = k$; состояния, действия, наблюдения и момент завершения k определяются трассой τ . Затем сомножители конечного произведения переставлены — одношаговые множители (2.17) собраны в первое произведение, вероятности выбора во второе, и выбор останавливающего действия $\pi(a_k | h_k)$ вошел во второе сомножителем с номером $t = k$; начальное распределение и одношаговые множители от политики не зависят, зависимость от нее собрана во втором произведении. Последний переход факторизует каждый выбор по агентам, $\pi(a_t|h_t) = \prod_{i \in I} \pi_i(a_i^t|h_i^t)$: агенты выбирают компоненты совместного действия независимо, каждый по своей локальной истории, а она определяется полной историей правилом записи (2.14).

Локальная история принимает значения в конечном множестве вместе с компонентой S_{agent} , поэтому детерминированные допустимые политики задают конечное число различных наборов значений. Хотя бы один допустимый набор

существует вследствие условия завершения. Останавливающее совместное действие принадлежит каждому допустимому множеству, покомпонентно — каждому множеству $A_i(s)$, и выбор останавливающей компоненты при всякой локальной истории допустим. Вещественная функция на конечном непустом множестве достигает максимума, поэтому максимум ожидаемой оценки по детерминированным допустимым политикам достигается. Остается показать, что рандомизация значения не увеличивает.

Зафиксируем агента α_i и его локальную историю $\bar{h}$. Остаток бюджета действий входит в локальную историю и убывает с каждым выполненным шагом. Вдоль одной трассы равенство $h_i^t = \bar{h}$ выполняется поэтому не более чем при одном t , и сомножитель вида $\pi_i(\cdot | \bar{h})$ входит в произведение не более одного раза. Разобьем завершенные трассы по выбору агента α_i при истории $\bar{h}$: класс D_a составляют трассы, в которых история $\bar{h}$ реализовалась и выбрано действие a , класс D_0 — трассы, в которых история $\bar{h}$ у агента α_i не встречается; классы не пересекаются и исчерпывают D . Ожидаемая оценка принимает вид

$$E_\pi[R(\tau)|b_0] = \sum_{\tau \in D_0} R(\tau) \cdot P(h_\kappa, a_\kappa) + \sum_{a \in A_i} \pi_i(a|\bar{h}) \cdot y(a) = y_0 + \sum_{a \in A_i} \pi_i(a|\bar{h}) \cdot y(a).$$

В сумме, разбитой по классам, в вероятность каждой трассы класса D_a сомножитель $\pi_i(a | \bar{h})$ входит ровно один раз и общим множителем вынесен за знак суммы по классу; суммой $y(a)$ обозначено оставшееся — произведения оценки трассы и всех прочих сомножителей ее вероятности, а суммой y_0 — сумма по классу D_0 , вероятности трасс которого распределения $\pi_i(\cdot | \bar{h})$ не содержат. От распределения $\pi_i(\cdot | \bar{h})$ величины $y(a)$ и y_0 не зависят.

Значения распределения $\pi_i(\cdot | \bar{h})$ на действиях в сумме дают единицу, поэтому

$$y_0 + \sum_{a \in A_i} \pi_i(a|\bar{h}) \cdot y(a) = \sum_{a \in A_i} \pi_i(a|\bar{h}) \cdot (y_0 + y(a)) = \sum_{a: \pi_i(a|\bar{h}) > 0} \pi_i(a|\bar{h}) \cdot (y_0 + y(a)) \leq \sum_{a: \pi_i(a|\bar{h}) > 0} \pi_i(a|\bar{h}) \cdot \max_{a': \pi_i(a'|\bar{h}) > 0} (y_0 + y(a')) = \max_{a': \pi_i(a'|\bar{h}) > 0} (y_0 + y(a')).$$

Величина y_0 внесена под знак суммы, поскольку веса $\pi_i(a | \bar{h})$ в сумме дают единицу; нулевые слагаемые отброшены. Неравенство заменяет каждую величину $y_0 + y(a)$ наибольшей из них по действиям с положительной вероятностью — веса неотрицательны, и почленная замена сумму не уменьшает. Максимум вынесен за

знак суммы как не зависящий от индекса суммирования, а сумма весов по действиям с положительной вероятностью равна единице.

Ограничение максимума действиями с положительной вероятностью существенно для допустимости. Распределения допустимой политики сосредоточены на допустимых множествах, а множество $A_i(s)$ определяется по правилу (2.16) той же историей $\bar{h}$. Действие с $\pi_i(a | \bar{h}) > 0$ поэтому допустимо всюду, где эта история реализуется. Для действия с нулевой вероятностью такой гарантии нет, и замена на него могла бы вывести политику из класса Π .

Пар из агента и локальной истории конечное число. При всякой такой паре и фиксированных остальных распределениях максимум в правой части последнего неравенства достигается на некотором действии a^* . Замена распределения $\pi_i(\cdot | \bar{h})$ вырожденным, сосредоточенным на a^* , допустима, а подстановка вырожденного распределения в представление ожидаемой оценки дает значение $y_0 + y(a^*)$, поскольку величины y_0 и $y(a)$ от заменяемого распределения не зависят и при замене сохраняются. По доказанному неравенству значение $y_0 + y(a^*)$ не меньше ожидаемой оценки исходной политики, то есть замена оценки не уменьшает. Выполним такие замены последовательно по всем парам; величины y_0 и $y(a)$ на каждом шаге вычисляются по текущей политике, каждая замена затрагивает единственное распределение, и ранее замененные остаются вырожденными. Значения политики на нереализуемых историях заменяются произвольными вырожденными распределениями, поскольку такие истории не встречаются ни в одной трассе, и оценка сохраняется. Итог замен — детерминированная допустимая политика $\tilde{\pi}$, для которой $E_{\pi}[R(\tau)|b_0] \leq E_{\tilde{\pi}}[R(\tau)|b_0]$.

Обозначим через π^* детерминированную допустимую политику с наибольшей ожидаемой оценкой; она существует, поскольку максимум по детерминированным допустимым политикам достигается, как установлено выше. Политика $\tilde{\pi}$ детерминирована и допустима, поэтому

$$\forall \pi \in \Pi: E_{\pi}[R(\tau)|b_0] \leq E_{\tilde{\pi}}[R(\tau)|b_0] \leq E_{\pi^*}[R(\tau)|b_0].$$

Левое неравенство накоплено заменами, правое выражает выбор π^* как наилучшей детерминированной допустимой политики. Правая часть от π не

зависит, а π^* принадлежит классу Π , поэтому максимум в (2.21) равен $E_{\pi^*}[R(\tau)|b_0]$ и достигается на детерминированной совместной политике π^* . Что и требовалось доказать.

Установленный максимум относится к произвольной оценке завершенной трассы. Конкретный вид оценки определяется составом критериев качества обследования и способом их объединения в одну величину.

Оценка по завершенной трассе обобщает пошаговое вознаграждение. Недоступность пошаговых значений в ходе сеанса не исключает разложимости итоговой оценки на вклады шагов. Оценка трассы совместима с пошаговой постановкой, если допускает разложение (2.22).

$$R(\tau) = \sum_{t=0}^{k-1} \gamma^t \cdot R(s_t, a_t) + \gamma^k \cdot \zeta(s_k), \quad (2.22)$$

где

γ — коэффициент дисконтирования кортежа (2.4), $\gamma \in [0, 1]$;

$R(s, a)$ — вклад отдельного шага, отождествляемый с функцией вознаграждения кортежа (2.4);

ζ — завершающий вклад, определенный на терминальных состояниях.

Разложение (2.22) понимается как тождество на всякой допустимой трассе с произвольным начальным состоянием; допустимой считается трасса, действия которой принадлежат допустимым множествам $A(s_t)$, а переходы имеют положительную вероятность. Правая часть (2.22) определяется состояниями и действиями трассы, поэтому при выполнении разложения наблюдения трассы на ее оценку не влияют. Всюду принято $\gamma^0 = 1$; при нулевой длине трассы сумма в (2.22) пуста, и оценка равна завершающему вкладу начального состояния. Выбор останавливающего действия в состоянии s завершает сеанс и дает завершающий вклад $\zeta(s)$, причем само останавливающее действие, не входящее в число k выполненных шагов, множителя γ не добавляет. Для оценки завершенной трассы, недисконтированной по своей природе, естественен выбор $\gamma = 1$, допустимый ввиду конечности сеанса.

Совместная вера отдельным агентам недоступна, поэтому уравнение оптимальности ставится над политиками централизованного планирования [110;

126]. Историей планирования служит последовательность пар «действие — наблюдение» из совместных действий и наблюдений; в начале сеанса она пуста. Политика планирования μ отображает такую историю в распределение над совместными действиями, сосредоточенное на допустимом множестве текущего информационного состояния. Это состояние вычисляется по истории правилом записи (2.14) от начального информационного состояния сеанса; при старте из веры b_t начальным служит единственное значение $s_{\text{agent}}(t)$ на ее носителе. Класс таких политик обозначается $\bar{\Pi}$. По расщеплению (2.20) вера всякого шага сосредоточена на состояниях с информационной компонентой $s_{\text{agent}}(t)$, поэтому допустимое множество на носителе веры общее и по правилу (2.16) равно $f(s_{\text{agent}}(t))$; общим является и остаток бюджета действий. Множество продолжающих действий — допустимых, кроме останавливающего, — обозначается $A_t^\circ = f(s_{\text{agent}}(t)) \setminus \{\hat{a}\}$. Ожидаемая оценка сеанса, начатого из веры b_t , обозначается $V^\mu(b_t) = E_\mu[R(\tau)|b_t]$: начальное состояние разыгрывается по b_t , трасса порождается политикой μ , а общий остаток бюджета ограничивает число шагов. Оптимальное значение составляет $V^*(b_t) = \sup_{\mu \in \bar{\Pi}} V^\mu(b_t)$.

Вероятность наблюдения при вере вычисляется взвешиванием одношагового множителя (2.17) состоянием веры,

$$P(o|b_t, a) = \sum_{s \in S} b_t(s) \cdot \sum_{s' \in S} P(s', o|s, a),$$

а пересчитанная вера $b'_{a,o}$ определена пересчетом перед утверждением 2.2; суммирование определения пересчета по новым состояниям дает $\eta \cdot P(o | b_t, a) = 1$ при $P(o | b_t, a) > 0$. При нулевой вероятности наблюдения пересчет не определен. Вера $b'_{a,o}$ доопределяется произвольно, а всякое слагаемое с множителем $P(o | b_t, a) = 0$ нулевое.

Утверждение 2.4 (о совместимости трассовой оценки с уравнением оптимальности). Пусть оценка $R(\tau)$ определена на всех завершенных трассах и допускает разложение (2.22) при фиксированном $\gamma \in [0, 1]$. Тогда уравнение

оптимальности вида (2.12) принимает над состоянием веры композитной модели вид

$$V^*(b_t) = \max \left\{ \sum_{s \in S} b_t(s) \cdot \zeta(s), \max_{a \in A_t^\circ} \left(\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o}) \right) \right\}.$$

Максимум сравнивает две величины — немедленный останов, то есть выбор действия $\hat{a}$ с ожидаемым завершающим вкладом, и наилучшее продолжение среди действий множества A_t° . При пустом A_t° , в частности при исчерпанном бюджете, вторая строка отсутствует, и максимум равен ожидаемому завершающему вкладу.

Доказательство. Уравнение (2.12) описывает оптимизацию ожидаемой суммы дисконтированных пошаговых вкладов, тогда как целевой функционал (2.21) ставится над ожидаемой оценкой завершённой трассы; разложение (2.22) уравнивает обе величины. Оно есть тождество на всякой допустимой трассе, а при всякой допустимой политике вероятность недопустимой трассы равна нулю. Сосредоточенность распределений политики на допустимых множествах обнуляет вероятности выбора недопустимых действий, а трасса с переходом нулевой вероятности содержит нулевой одношаговый множитель (2.17). Ожидаемая оценка поэтому раскрывается цепочкой

$$\forall \pi \in \Pi: E_\pi[R(\tau)|b_0] = \sum_{\tau \in D} R(\tau) \cdot P(h_\kappa, a_\kappa) = \sum_{\tau \in D} \left(\sum_{t=0}^{\kappa-1} \gamma^t \cdot R(s_t, a_t) + \gamma^\kappa \cdot \zeta(s_\kappa) \right) \cdot P(h_\kappa, a_\kappa) = E_\pi \left[\sum_{t=0}^{\kappa-1} \gamma^t \cdot R(s_t, a_t) + \gamma^\kappa \cdot \zeta(s_\kappa) | b_0 \right].$$

Крайние части цепочки — одно и то же представление ожидания суммой по завершённым трассам с весами-вероятностями, введенное в доказательстве утверждения 2.3: под знаком ожидания в обоих случаях стоит функция трассы, и момент завершения κ , состояния и действия слагаемых определяются трассой τ . Среднее равенство заменяет оценку каждой трассы правой частью разложения (2.22) — на трассах положительной вероятности замена тождественна, а слагаемые нулевой вероятности нулевые в обеих суммах. Оптимизация ожидаемой оценки трассы тем самым совпадает с оптимизацией ожидаемой суммы вкладов, и место пошагового вознаграждения в уравнении занимают вклад шага $R(s, a)$ и завершающий вклад ζ из разложения (2.22).

Для трассы, начатой останавливающим действием $\hat{a}$, момент завершения $\kappa = 0$ и оценка равна $\zeta(s_0)$. Для допустимой трассы τ с начальным состоянием s_0 , начатой

действием $a_0 \neq \hat{a}$, момент завершения удовлетворяет $\kappa \geq 1$, и отделение вклада нулевого шага в (2.22) дает цепочку

$$R(\tau) = R(s_0, a_0) + \sum_{t=1}^{\kappa-1} \gamma^t \cdot R(s_t, a_t) + \gamma^\kappa \cdot \zeta(s_\kappa) = R(s_0, a_0) + \gamma \cdot \left(\sum_{j=0}^{\kappa-2} \gamma^j \cdot R(s_{j+1}, a_{j+1}) + \gamma^{\kappa-1} \cdot \zeta(s_\kappa) \right) = R(s_0, a_0) + \gamma \cdot R(\tilde{\tau}).$$

В (2.22) отделено слагаемое нулевого шага, входящее с множителем $\gamma^0 = 1$, затем заменой индекса суммирования $t = j + 1$ вынесен общий множитель γ ; скобка фиксирует, что множитель вынесен и из завершающего вклада. Выражение в скобке в точности повторяет разложение (2.22) для трассы $\tilde{\tau}$, полученной из τ отбрасыванием нулевого шага. Ее шаг с номером j есть пара (s_{j+1}, a_{j+1}) , момент завершения равен $\kappa - 1$, терминальное состояние совпадает с s_κ . Разложение применимо к укороченной трассе. Ее действия и переходы взяты из допустимой трассы τ , поэтому она допустима; тождество (2.22) действует при произвольном начальном состоянии, в том числе при ее начальном состоянии s_1 .

Первое действие сеанса политика μ выбирает по пустой истории планирования. Запись $\mu(a)$ обозначает вероятность выбрать действие a первым шагом, в частности $\mu(\hat{a})$ — вероятность немедленного останова. Через $\mu_{a,o}$ обозначается продолжение политики μ после первого действия a и наблюдения o — оно управляет хвостовым сеансом; при $\mu(a) = 0$ продолжение выбирается произвольно, соответствующие слагаемые нулевые. Развертка ожидаемой оценки строится разбиением трасс на классы по первому действию, вычислением вклада каждого класса и сложением вкладов. Разбиение трасс по первому действию a_0 дает цепочку

$$V^\mu(b_t) = E_\mu[R(\tau)|b_t] = \sum_{\tau \in D} R(\tau) \cdot P_\mu(\tau|b_t) = \sum_{\tau: a_0 = \hat{a}} R(\tau) \cdot P_\mu(\tau|b_t) + \sum_{a \in A_t^\circ} \sum_{\tau: a_0 = a} R(\tau) \cdot P_\mu(\tau|b_t).$$

Ожидаемая оценка раскрыта представлением ожидания суммой по завершенным трассам; вероятность трассы дается распределением траекторий (2.18) с начальным распределением b_t и множителями выбора μ вместо множителей π . Завершенная трасса τ определяет историю h_κ и останавливающее действие a_κ , поэтому эта вероятность записывается как $P_\mu(\tau|b)$ с указанием начального распределения b после черты. Затем трассы разбиты по первому действию. Классы не пересекаются и исчерпывают трассы положительной вероятности, поскольку

трассы с первым действием вне допустимого множества входят с нулевым весом вследствие сосредоточенности распределения политики.

Класс останавливающего действия исчерпывается трассами из начального состояния и останова,

$$\sum_{\tau: a_0 = \hat{a}} R(\tau) \cdot P_\mu(\tau|b_t) = \mu(\hat{a}) \cdot \sum_{s \in S} b_t(s) \cdot \zeta(s).$$

Такая трасса определяется начальным состоянием s : ее вероятность равна $b_t(s) \cdot \mu(\hat{a})$, а оценка по разложению (2.22) при $\kappa = 0$ равна $\zeta(s)$.

Ожидаемая оценка сеанса из фиксированного начального состояния вводится тем же представлением. Для политики μ и начального состояния s положим

$$E_\mu[R(\tau)|s] = \sum_{\tau \in D} R(\tau) \cdot P_\mu(\tau|s),$$

где

$P_\mu(\tau|s)$ — распределение траекторий (2.18) с начальным распределением, вырожденным в s , и множителями выбора политики μ . Для продолжения $\mu_{a,o}$ и состояния s' это дает ожидаемую оценку хвостового сеанса $E_{\mu_{a,o}}[R(\tilde{\tau})|s']$: множители (2.17) не содержат номера шага, поэтому хвост исходного сеанса распределен именно так, а его бюджет списан правилом записи.

Ожидаемая оценка при начальной вере есть взвешивание таких ожиданий. При всякой политике $\mu \in \bar{\Pi}$ и всякой вере b выполнено

$$V^\mu(b) = \sum_{\tau \in D} R(\tau) \cdot P_\mu(\tau|b) = \sum_{\tau \in D} R(\tau) \cdot \left(\sum_{s \in S} b(s) \cdot P_\mu(\tau|s) \right) = \sum_{s \in S} b(s) \cdot \sum_{\tau \in D} R(\tau) \cdot P_\mu(\tau|s) = \sum_{s \in S} b(s) \cdot E_\mu[R(\tau)|s].$$

Цепочка раскрывает ожидание суммой по трассам и использует линейность распределения траекторий по начальному распределению. Начальное распределение входит в (2.18) единственным множителем при начальном состоянии трассы, при вырожденном распределении этот множитель равен единице на своем состоянии и нулю на остальных, и внутренняя сумма оставляет ровно $b(s_0) \cdot P_\mu(\tau|s_0)$ — вероятность той же трассы при вере b . Перестановка конечных сумм с выносом $b(s)$, от трассы не зависящей, и сворачивание внутренней суммы по определению ожидания при начальном состоянии завершают вывод.

Трассы класса продолжающего действия $a \in A_t^\circ$ перечисляются началом и хвостом,

$$\sum_{\tau: a_0=a} R(\tau) \cdot P_\mu(\tau|b_t) = \sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot \sum_{\tilde{\tau} \in D} (R(s, a) + \gamma \cdot R(\tilde{\tau})) \cdot P_{\mu_{a,o}}(\tilde{\tau}|s').$$

Трасса с первым действием a распадается на начальную четверку (s, a, s', o) и хвостовую трассу $\tilde{\tau}$ из состояния s' ; ее вероятность факторизуется по виду (2.18) в произведение $b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot P_{\mu_{a,o}}(\tilde{\tau}|s')$, а оценка равна $R(s, a) + \gamma \cdot R(\tilde{\tau})$ по потраекторному сдвигу; суммирование по классу и есть суммирование по четверкам и хвостам.

Внутренняя сумма по хвостам вычисляется цепочкой

$$\sum_{\tilde{\tau} \in D} (R(s, a) + \gamma \cdot R(\tilde{\tau})) \cdot P_{\mu_{a,o}}(\tilde{\tau}|s') = R(s, a) \cdot \sum_{\tilde{\tau} \in D} P_{\mu_{a,o}}(\tilde{\tau}|s') + \gamma \cdot \sum_{\tilde{\tau} \in D} R(\tilde{\tau}) \cdot P_{\mu_{a,o}}(\tilde{\tau}|s') = R(s, a) + \gamma \cdot E_{\mu_{a,o}}[R(\tilde{\tau})|s'].$$

Скобка раскрыта, и вклад $R(s, a)$, от хвоста не зависящий, вынесен. Обе суммы свернуты — сумма вероятностей хвостовых трасс равна единице, поскольку хвостовой сеанс завершается и завершённые трассы исчерпывают его исходы, а сумма оценок с весами-вероятностями есть $E_{\mu_{a,o}}[R(\tilde{\tau})|s']$ по выписанному определению.

Подстановка внутренней суммы и раскрытие скобки распадаются вклад класса на два слагаемых,

$$\sum_{\tau: a_0=a} R(\tau) \cdot P_\mu(\tau|b_t) = \sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot (R(s, a) + \gamma \cdot E_{\mu_{a,o}}[R(\tilde{\tau})|s']) = \sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot R(s, a) + \gamma \cdot \sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot E_{\mu_{a,o}}[R(\tilde{\tau})|s'].$$

Скобка раскрыта почленно, общий множитель γ второго слагаемого вынесен за суммы.

Слагаемое вклада шага сворачивают нормировки,

$$\sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o|s, a) \cdot R(s, a) = \mu(a) \cdot \sum_{s \in S} b_t(s) \cdot R(s, a) \cdot \left(\sum_{s' \in S} \sum_{o \in \Omega} P(s', o|s, a) \right) = \mu(a) \cdot \sum_{s \in S} b_t(s) \cdot R(s, a).$$

Множители вынесены из внутренних сумм — $\mu(a)$ не зависит ни от одной переменной суммирования, $b_t(s)$ и $R(s, a)$ не содержат s' и o ; при каждом s внутренняя скобка равна единице по нормировке (2.17).

Слагаемое хвостовых оценок перегруппировкой сумм приводится к ожиданиям при пересчитанных верах,

$$\begin{aligned} \sum_{s \in S} \sum_{s' \in S} \sum_{o \in \Omega} b_t(s) \cdot \mu(a) \cdot P(s', o | s, a) \cdot E_{\mu_{a,o}}[R(\tilde{\tau}) | s'] &= \mu(a) \cdot \sum_{o \in \Omega} \sum_{s' \in S} (\sum_{s \in S} b_t(s) \cdot P(s', o | s, a)) \cdot E_{\mu_{a,o}}[R(\tilde{\tau}) | s'] \\ &= \mu(a) \cdot \sum_{o \in \Omega} \sum_{s' \in S} P(o | b_t, a) \cdot b'_{a,o}(s') \cdot E_{\mu_{a,o}}[R(\tilde{\tau}) | s'] \\ &= \mu(a) \cdot \sum_{o \in \Omega} P(o | b_t, a) \cdot V^{\mu_{a,o}}(b'_{a,o}). \end{aligned}$$

Конечные суммы переставлены — суммирование по s внесено внутрь, суммы по o и s' стали внешними, множитель $\mu(a)$ вынесен, а ожидание $E_{\mu_{a,o}}[R(\tilde{\tau}) | s']$, от s не зависящее, осталось при внешних суммах. Внутренняя сумма опознана как пересчет веры. Она равна $P(o | b_t, a) \cdot b'_{a,o}(s')$ по определению пересчета и равенству $\eta \cdot P(o | b_t, a) = 1$, слагаемые с $P(o | b_t, a) = 0$ нулевые. Вынос $P(o | b_t, a)$, от s' не зависящей, и сворачивание суммы по s' по выписанному тождеству взвешивания дают правую часть.

Сумма обоих слагаемых дает вклад класса

$$\sum_{\tau: a_0 = a} R(\tau) \cdot P_\mu(\tau | b_t) = \mu(a) \cdot (\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o | b_t, a) \cdot V^{\mu_{a,o}}(b'_{a,o})).$$

Подстановка вкладов обоих классов в разбиение дает развертку ожидаемой оценки

$$\begin{aligned} V^\mu(b_t) &= \mu(\hat{a}) \cdot \sum_{s \in S} b_t(s) \cdot \zeta(s) + \sum_{a \in A_t^\circ} \mu(a) \cdot (\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \\ &\sum_{o \in \Omega} P(o | b_t, a) \cdot V^{\mu_{a,o}}(b'_{a,o})). \end{aligned}$$

Уравнение доказывается индукцией по остатку бюджета действий веры. При нулевом остатке допустимое множество сокращается до останавливающего действия — множество A_t° пусто. Всякая политика немедленно завершает сеанс, $V^\mu(b_t) = \sum_s b_t(s) \cdot \zeta(s)$ при каждой μ , грань равна этой величине и достигается, а вторая строка максимума отсутствует — уравнение выполнено.

Пусть при остатке бюджета, меньшем v , грань конечна и достигается при всякой вере шага. Для веры b_t с остатком v пересчитанная вера $b'_{a,o}$ есть вера следующего шага с остатком $v - 1$: правило записи списывает выполненный шаг. По определению точной верхней грани $V^{\mu_{a,o}}(b'_{a,o}) \leq V^*(b'_{a,o})$, и замена хвостовых оценок гранями в развертке с оценкой выпуклой комбинации дает

$$\begin{aligned} \forall \mu \in \bar{P}: V^\mu(b_t) &\leq \mu(\hat{a}) \cdot \sum_{s \in S} b_t(s) \cdot \zeta(s) + \sum_{a \in A_t^\circ} \mu(a) \cdot (\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \\ &\sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o})) \leq \\ \max &\begin{cases} \sum_{s \in S} b_t(s) \cdot \zeta(s), \\ \max_{a \in A_t^\circ} (\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o})) \end{cases}. \end{aligned} \quad (2.23)$$

Почленная замена хвостовых оценок гранями законна — множитель γ и вероятности неотрицательны. Далее выпуклая комбинация оценена наибольшей строкой максимума. Каждая из взвешиваемых величин не превосходит ее, а веса $\mu(\hat{a})$ и $\mu(a)$ неотрицательны и в сумме дают единицу вследствие сосредоточенности распределения политики на допустимом множестве. Правая часть цепочки совпадает с правой частью уравнения и от политики не зависит; неравенство сохраняется при переходе слева к точной верхней грани по политикам класса.

Обратное неравенство устанавливается построением политики. Максимум правой части достигается. Внешний сравнивает две величины, а внутренний — когда вторая строка присутствует — берется по конечному непустому множеству. Если он доставляется верхней строкой, политика немедленного останова принадлежит классу $\bar{P}$ и дает $V^{\hat{\mu}}(b_t) = \sum_s b_t(s) \cdot \zeta(s)$ — значение совпадает с правой частью уравнения. Иначе пусть a^* — действие, доставляющее максимум второй строки; политика $\hat{\mu}$ выбирает a^* первым шагом, а после наблюдения o следует политике, достигающей грани $V^*(b'_{a^*,o})$ и существующей по предположению индукции — пересчитанная вера имеет остаток $v - 1$; продолжения при наблюдениях нулевой вероятности выбираются произвольно. Политика $\hat{\mu}$ принадлежит классу $\bar{P}$: первое действие принадлежит $f(s_{\text{agent}}(t))$, продолжения допустимы по построению. Подстановка $\hat{\mu}$ в развертку дает цепочку

$$\begin{aligned} V^{\hat{\mu}}(b_t) &= \sum_{s \in S} b_t(s) \cdot R(s, a^*) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a^*) \cdot V^*(b'_{a^*,o}) = \\ \max &\begin{cases} \sum_{s \in S} b_t(s) \cdot \zeta(s), \\ \max_{a \in A_t^\circ} (\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o})) \end{cases}. \end{aligned}$$

Подстановка $\hat{\mu}$ в развертку оставляет единственное слагаемое с весом единица, а хвостовые оценки равны граням по выбору продолжений; правая часть выражает выбор a^* как максимайзера второй строки и превосходство второй строки над первой в разбираемом случае. В обоих случаях значение политики $\hat{\mu}$ равно

правой части уравнения, и по определению точной верхней грани — $\hat{\mu}$ принадлежит классу $\bar{P}$ — выполнено

$$V^*(b_t) \geq V^{\hat{\mu}}(b_t) = \max \left\{ \sum_{s \in S} b_t(s) \cdot \zeta(s), \max_{a \in A_t} \left(\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o}) \right) \right\}. \quad (2.24)$$

Неравенства (2.23) — после перехода к точной верхней грани по политикам класса — и (2.24) вместе дают

$$V^*(b_t) = \max \left\{ \sum_{s \in S} b_t(s) \cdot \zeta(s), \max_{a \in A_t} \left(\sum_{s \in S} b_t(s) \cdot R(s, a) + \gamma \cdot \sum_{o \in \Omega} P(o|b_t, a) \cdot V^*(b'_{a,o}) \right) \right\}$$

то есть уравнение при остатке v выполнено, а грань конечна — как максимум конечного числа конечных величин — и достигается на политике $\hat{\mu}$. Что и требовалось доказать.

Для децентрализованного исполнения уравнение оптимальности характеризует централизованное планирование. Оно описывает наилучшее поведение системы в целом, но не предписывает способа его распределенного вычисления. Существование оптимальной совместной политики в децентрализованной постановке установлено утверждением 2.3 независимо от уравнения; для конечного горизонта тот же вывод составляет стандартный факт теории децентрализованных частично наблюдаемых марковских процессов принятия решений, изложенный Олихуком и Амато. Точная характеристика децентрализованного оптимума дается в схеме централизованного планирования совместной политики при децентрализованном исполнении. Достаточной статистикой планирования служит распределение по состояниям и совместным историям наблюдений, и конечно-горизонтная децентрализованная задача сводится к марковскому процессу принятия решений над непрерывным пространством таких статистик; сведение принадлежит Дибанге, Амато, Бюффе и Шарпилле, оптимальная функция ценности этого процесса кусочно-линейна и выпукла, а динамическое программирование сохраняет корректность.

Взаимосвязь трех модификаций с компонентами исходного кортежа приведена на рисунке 2.3.



Рисунок 2.3 – Авторская модификация модели

Задача нахождения оптимальной политики в модифицированной модели вычислительно сложна, поскольку пространство состояний распределенного комплекса растет экспоненциально с числом узлов. Мощность пространства состояний комплекса с $|N|$ узлами и k бинарными признаками на каждый узел составляет (2.25).

$$|S_{env}| = 2^{k \cdot |N|}, \quad (2.25)$$

где

k — число бинарных признаков состояния узла, таких как доступность узла, штатное функционирование его служб, допустимость значений его конфигурационных параметров, активность его сетевых соединений; параметризация укрупненная — признаки соединений и служб отнесены к узлам, которым они принадлежат.

С учетом компоненты S_{agent} общая мощность композитного пространства состояний составляет (2.26).

$$|S| = |S_{env}| \cdot |S_{agent}| = 2^{k \cdot |N|} \cdot 2^{m \cdot |N|} = 2^{(k+m) \cdot |N|}, \quad (2.26)$$

где

m — число бинарных признаков информационного состояния агентской системы на каждый узел (наличие сведений об узле, активного сеанса диагностики, собранных данных о его состоянии).

При $k = 4$ и $m = 3$ признаках и $|N| = 10$ узлах число возможных состояний достигает 2^{70} , что превышает 10^{21} . Оценка занижена — множитель остатка бюджета в информационной компоненте не учтен, — но и ее достаточно для вывода о практической недостижимости точного решения. Согласно расщеплению (2.20) неопределенность сосредоточена на компоненте комплекса, поэтому эффективная сложность пространства веры определяется мощностью $|S_{env}| = 2^{k \cdot |N|}$, однако и она остается за пределами возможностей точного решения при реальных размерах комплекса.

Варьируемыми переменными сформулированной оптимизационной задачи выступают политики агентов, критерием служит ожидаемая оценка завершенной трассы (2.21). Задача точного решения в децентрализованной постановке полна для класса недетерминированного экспоненциального времени согласно результату Бернштейна, Гивана, Иммермана и Зильберштейна, а точные алгоритмы динамического программирования имеют сложность двойной экспоненты от горизонта планирования по оценке из монографии Олихука и Амато. Существование оптимальной политики, установленное утверждением 2.3, следует отличать от ее точной вычислимости. Приближенные методы приближают гарантированно существующий оптимум, сама же постановка задачи при переходе к ним не изменяется.

Таким образом, предложенная автором модификация трех компонентов кортежа M составляет основной теоретический вклад в формальную модель, а утверждения 2.1–2.4 устанавливают ее корректность. Марковость и однородность сохраняются на композитном пространстве состояний, состояние веры расщепляется на фактическое значение информационной компоненты и распределение по компоненте комплекса, оптимальная совместная политика существует и достигается на детерминированных политиках, а трассовая оценка при разложении (2.22) совместима с уравнением оптимальности.

Экспоненциальная сложность пространства состояний при этом исключает точные методы и обосновывает переход к приближенным алгоритмам на основе машинного обучения [127; 80].

Предложенная модель определяет оптимизационную задачу нахождения совместной политики агентов. Для практической реализации необходимо определить модель архитектуры агентской системы, включая состав агентов, распределение функций между ними и механизмы координации.

2.3 Модель архитектуры многоагентной интеллектуальной системы

Модель архитектуры определяет состав агентской системы, распределение функций между агентами, механизмы их координации и организацию накопленных сведений. Автором многоагентная интеллектуальная система определяется как организованная в виде иерархической или сетевой структуры совокупность автономных программных сущностей, именуемых агентами [133]. Агенты обладают способностью к восприятию и интерпретации внешней среды, принятию решений, целенаправленному действию, а также взаимодействию друг с другом. Система функционирует в стохастических условиях с частичной или полной неопределенностью и реализует процессы адаптации, самообучения и коллективного поведения для достижения индивидуальных и коллективных целей.

Проектирование архитектуры такой системы опирается на принципы иерархической декомпозиции, модульности и распределенного управления. Иерархическая декомпозиция разбивает задачу на подзадачи убывающего масштаба, модульность закрепляет за каждым агентом ограниченную зону ответственности, а распределенное управление проводит через отдельного агента решения одного масштаба, но не все решения системы.

Автором иерархическая архитектура многоагентной интеллектуальной системы рассматривается как параметрическая модель с ℓ уровнями, где $\ell \geq 1$ [4;

8]. Множество агентов I кортежа (2.4) разбивается на ℓ непересекающихся подмножеств по уровням иерархии (2.27).

$$I = I_1 \cup I_2 \cup \dots \cup I_\ell, \quad \forall j, k \in \{1, \dots, \ell\}, j \neq k: I_j \cap I_k = \emptyset, \quad (2.27)$$

где

I_k — множество агентов k -го уровня иерархии, $k = 1, \dots, \ell$.

Условие пустого пересечения закрепляет за каждым агентом единственный уровень. Агенты верхних уровней, с меньшими значениями k , отвечают за формирование целей и планирование, агенты нижних уровней — за непосредственное взаимодействие с исследуемым комплексом. Задача поступает агенту от вышестоящего уровня; агенту верхнего уровня задача ставится извне как цель обследования. При $\ell = 1$ иерархия вырождается в плоскую организацию равноправных агентов, а каждый дополнительный уровень вводит промежуточную ступень между постановкой цели и элементарной операцией.

Поведение агента α_i определяется его политикой π_i , введенной при целевом функционале (2.21), — отображением локальной истории агента в распределение над множеством действий A_i . Множество A_i включает создание дочерних агентов, вызов инструментов диагностики, обмен сообщениями и формирование отчетов. Множество I фиксировано кортежем модели, тогда как состав действующих агентов изменяется по ходу обследования. Агент становится действующим с делегированием ему задачи и выбывает из работы с ее исполнением либо с прекращением работы. Действующие агенты составляют переменное подмножество множества I , формализуемое далее графом делегирования.

Общая ℓ -уровневая архитектура приведена на рисунке 2.4.

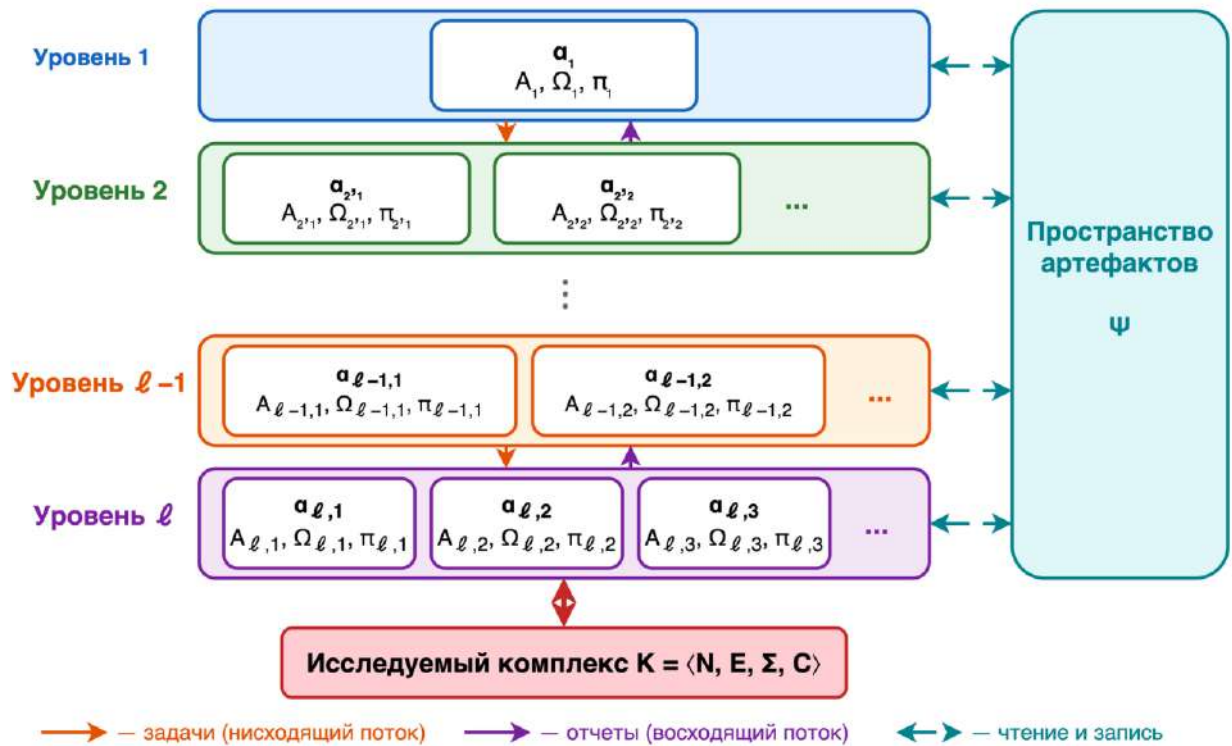


Рисунок 2.4 – Общая l -уровневая архитектура многоагентной интеллектуальной системы

Число уровней соответствует глубине естественной декомпозиции задачи. Недостаточное число уровней перегружает отдельных агентов задачами разного масштаба, избыточное увеличивает задержки и потери сведений при межуровневой передаче.

В задаче оценки работоспособности распределенного программно-аппаратного комплекса решения принимаются на несводимых один к другому масштабах. Глобальное целеполагание охватывает формирование стратегии обследования, назначение приоритетов и распределение ресурсов между направлениями — таково, например, решение о выборе между оценкой пропускной способности сетевых соединений и углубленным обследованием программных служб на узлах. Тактическое управление сводит стратегическую задачу к последовательности конкретных операций в пределах одного направления, к таким решениям относится порядок диагностики программных служб выбранного вычислительного узла. Исполнение отвечает отдельной элементарной операции взаимодействия с комплексом — составу параметров диагностического запроса к

определенной службе. Масштабы опираются на различные сведения и различаются ценой ошибки. Ошибочная стратегия обесценивает все последующие операции, ошибочная операция искажает единственное наблюдение.

При $\ell = 2$ верхний уровень совмещает глобальное целеполагание с тактическим управлением, и агент, погруженный в детали отдельного направления, утрачивает целостное представление о задаче по мере накопления частных сведений. При $\ell \geq 4$ хотя бы один из масштабов делится между смежными уровнями, а каждая лишняя ступень передачи задач и отчетов увеличивает задержки и потери сведений при ретрансляции. Трехуровневая организация закрепляет за каждым масштабом отдельный уровень. В теории иерархических систем Месаровича, Мако и Такахары ей отвечает многоуровневая организация с разделением координации, адаптации и исполнения, а в теории управления организационными системами Буркова и Новикова трехуровневая иерархия признается минимально достаточной для систем, совмещающих планирование, оперативное управление и исполнение.

Автором предлагается трехуровневая архитектура как частный случай общей модели при $\ell = 3$ (2.28).

$$I = I_{str} \cup I_{tac} \cup I_{oper}, \quad |I_{str}| = 1, \quad (2.28)$$

где

I_{str} — множество агентов стратегического уровня, состоящее из единственного агента α_1 ;

I_{tac} — множество агентов тактического уровня;

I_{oper} — множество агентов операционного уровня.

Единственность агента стратегического уровня обеспечивает согласованность стратегии обследования, решения глобального масштаба принимаются в одной точке. Числа действующих агентов тактического и операционного уровней изменяются по ходу сеанса вслед за составом решаемых задач.

Агент стратегического уровня, называемый далее координатором, реализует глобальное целеполагание. Координатор анализирует поставленную цель

обследования, выделяет направления обследования, формирует тактические задачи, создает под них агентов тактического уровня и распределяет между ними приоритеты. Инструменты диагностики в состав его операций не назначаются — сведения о комплексе поступают к координатору в отчетах нижестоящих агентов, что освобождает его контекст от низкоуровневых деталей. Решение о завершении сеанса, выбор останавливающего действия $\hat{a}$, закреплено за координатором его конфигурацией.

Агенты тактического уровня, называемые далее агентами-руководителями, реализуют тактическое управление. Каждый агент-руководитель создается координатором под определенное направление обследования — исследование сетевой инфраструктуры, диагностику класса служб, работу с определенной операционной системой. Агент-руководитель разбивает назначенную задачу на элементарные операции, создает под них агентов операционного уровня, агрегирует полученные результаты и передает координатору отчет по направлению.

Агенты операционного уровня, называемые далее исполнителями, реализуют элементарные операции. Исполнитель получает от агента-руководителя элементарную задачу, формирует диагностический запрос, интерпретирует ответ и возвращает наблюдение в отчете. Вызовы инструментов диагностики проходят через стандартизированный протокольный интерфейс, единый для всех инструментов; состав назначаемых агенту инструментов определяет его создатель при делегировании, и задача, не требующая обращения к комплексу, инструментов не получает. Исполнители, созданные одним агентом-руководителем, образуют команду; прямого канала между командами разных направлений нет, согласование направлений выполняется на тактическом уровне.

Политики агентов всех уровней параметризуются единой языковой моделью, поскольку требуемые уровням навыки — декомпозиция задач, формирование инструкций, интерпретация результатов, координация — по существу совпадают [56; 128].

Разделение ролей при единой модели несет конфигурация агента, назначаемая создателем в момент делегирования, а для координатора — при развертывании системы, и неизменная до завершения работы агента. Тексты, которыми обмениваются агенты и которыми описываются задачи и обнаруженные факты, составляют конечное множество $\mathcal{E}$ допустимых текстов ограниченной длины над конечным алфавитом. Конфигурация агента α_i определяется как пара (2.29).

$$\rho_i = \langle l_i, \tilde{A}_i \rangle, \quad (2.29)$$

где

ρ_i — конфигурация агента α_i ;

$l_i \in \mathcal{E}$ — системная инструкция агента: описание его роли, правил поведения и границ ответственности, подаваемое языковой модели при каждой генерации;

$\tilde{A}_i \subseteq A_i$ — назначенный состав операций: подмножество множества действий, открытое агенту его создателем под назначенную задачу.

Инструкции агентов-руководителей и исполнителей вместе с назначаемым составом операций порождаются создателями по ходу сеанса — состав направлений обследования заранее не известен, и роли формулируются по накопленным сведениям.

Координация агентов сочетает вертикальную передачу задач и отчетов с обменом накопленными сведениями через информационное пространство системы — локальную память агента, наследование контекста при делегировании, почтовый ящик и пространство артефактов. Ведущие их структуры входят в информационную компоненту S_{agent} композитного состояния и собственных законов изменения не имеют, всякое их пополнение выполняется правилом записи (2.14) по действию и наблюдению шага.

Локальной памятью агента α_i служит доступная ему часть информационного состояния — локальная история h_i^t , аргумент его политики. Имя «локальная память» употребляется в описании архитектуры, имя «локальная история» — в формальных соотношениях. Память складывается из назначенной агенту задачи и унаследованного при делегировании контекста, выполненных им действий и

полученных на них наблюдений, прочитанных сообщений и извлеченных описаний пространства артефактов; к ней отнесены и служебные сведения — назначенный состав операций его конфигурации и остаток бюджета действий. Память образует основной контекст принятия решений, языковая модель получает ее на вход при выборе каждого действия. Новые записи дописываются в конец последовательности, а при превышении допустимого объема хранения ранние сжимаются в сводку с сохранением ключевых фактов — теми же операциями, которыми правило записи (2.14) ведет ограниченную сводку информационного состояния в целом.

Наследование контекста связывает память создателя с памятью дочернего агента. Создатель вызывает операцию делегирования, передавая ей описание задачи и отобранный фрагмент собственной памяти, из которых складывается начальная память дочернего агента. Снимок контекста фиксируется в момент создания и впоследствии не синхронизируется — памяти создателя и дочернего агента развиваются независимо, без согласования копий и конкурентного доступа, свойственных архитектурам с разделяемой памятью.

Почтовый ящик обеспечивает адресный обмен сообщениями. Каждому агенту α_i сопоставляется входящая очередь (2.30).

$$Q_i \in (I \times E \times \{0,1\})^{\leq N_{\max}}, \quad (2.30)$$

где

Q_i — почтовый ящик агента α_i : упорядоченная по поступлению последовательность троек из агента-отправителя, текста сообщения и признака прочтения, принимающего значение 0 для непрочитанного сообщения и 1 для прочитанного.

Длина очереди ограничена бюджетом $N_{\max}$, поскольку отправка сообщения расходует бюджет наравне с прочими операциями.

Отправка сообщения агентом α_i агенту α_j дописывает в очередь Q_j тройку с нулевым признаком прочтения; чтение выполняется отдельной операцией получателя, которая выдает непрочитанные сообщения и переводит признак в единицу. Отправитель не ожидает ни готовности получателя, ни ответа —

сообщение сохраняется в очереди до чтения, чем обеспечивается асинхронность обмена. Агенты-руководители обмениваются сообщениями между направлениями и с координатором, исполнители — внутри своей команды и с агентом-руководителем своего направления. Отчет о завершении задачи — отдельная операция агента, дописывающая структурированные результаты в почтовый ящик создателя; туда же доставляется извещение об отказе дочернего агента, условия которого определяются далее при состояниях жизненного цикла. Делегирование выполняется в синхронном режиме, когда создатель приостанавливает работу до поступления отчета либо извещения об отказе от одного из своих дочерних агентов, или в асинхронном, когда он продолжает работу.

Память, наследование и сообщения переносят сведения вдоль связей создания и известных адресов, а задачу горизонтального обмена между ветвями решают не полностью — исполнитель одной ветви не осведомлен о существовании исполнителей других ветвей и не может ни адресовать им сообщение, ни узнать их результаты, а сведения чужого направления доступны лишь через опрос его агента-руководителя.

Задачу горизонтального обмена решает вводимое автором пространство артефактов. Под артефактом понимается текстовое описание факта, обнаруженного агентами при обследовании комплекса, а под пространством артефактов Ψ — общее хранилище таких описаний с накоплением записей и поиском по семантической близости [99]. Пространство артефактов доступно агентам всех уровней независимо от положения в иерархии. Записи суть элементы множества $\mathcal{E}$; их сопоставление выполняется по векторным представлениям, которые сопоставляет описаниям фиксированное отображение, причем близость векторных представлений выражает семантическую близость описаний.

Состоянием пространства артефактов служит конечное множество записанных описаний — подмножество множества $\mathcal{E}$, входящее в информационную компоненту; операция записи добавляет новое описание, и по шагам сеанса множество не убывает. Операция поиска принимает запрос из $\mathcal{E}$ и возвращает записанные описания, упорядоченные по близости векторных

представлений запроса и записи. Поиск не требует ни точной терминологии агента-источника, ни его адреса — запись, оставленная исполнителем одной ветви, извлекается агентом-руководителем другой по семантической близости запроса. Пространство артефактов служит хранилищем фактов с общим поиском, адресный обмен остается за почтовыми ящиками.

Перечисленные механизмы определяют, какая часть накопленных в сеансе сведений доступна отдельному агенту. У разных агентов эти части различны — каждый располагает своей памятью и своим почтовым ящиком, а общими служат записи пространства артефактов и остаток бюджета действий. Этой частью по правилу (2.16) определяется сомножитель $A_i(s)$ произведения $f(s_{\text{agent}})$, допустимые операции открываются каждому агенту по доступным ему сведениям.

Работа отдельного агента организована итеративным циклом «действие — наблюдение — обновление памяти» [56]. На каждой итерации языковая модель получает локальную память агента и его конфигурацию ρ_i и генерирует действие — вызов инструмента, чтение ящика, отправку сообщения, запись или поиск в пространстве артефактов, создание дочернего агента либо передачу отчета. Результат действия дописывается в память, после чего генерируется следующее действие. Цикл продолжается до передачи отчета о завершении задачи либо до приостановки при синхронном делегировании.

Наблюдениями в смысле функции (2.15) служат при этом только ответы комплекса на диагностические операции. Результат внутрисистемной операции — чтения ящика, поиска в пространстве артефактов — наблюдения не образует, он есть детерминированная функция информационного состояния и выполненного действия, поскольку и очереди, и записи пространства артефактов входят в информационную компоненту. Правило записи (2.14) переносит такой результат в память, не выходя за пределы своих аргументов.

Схема основного цикла работы агента приведена на рисунке 2.5.

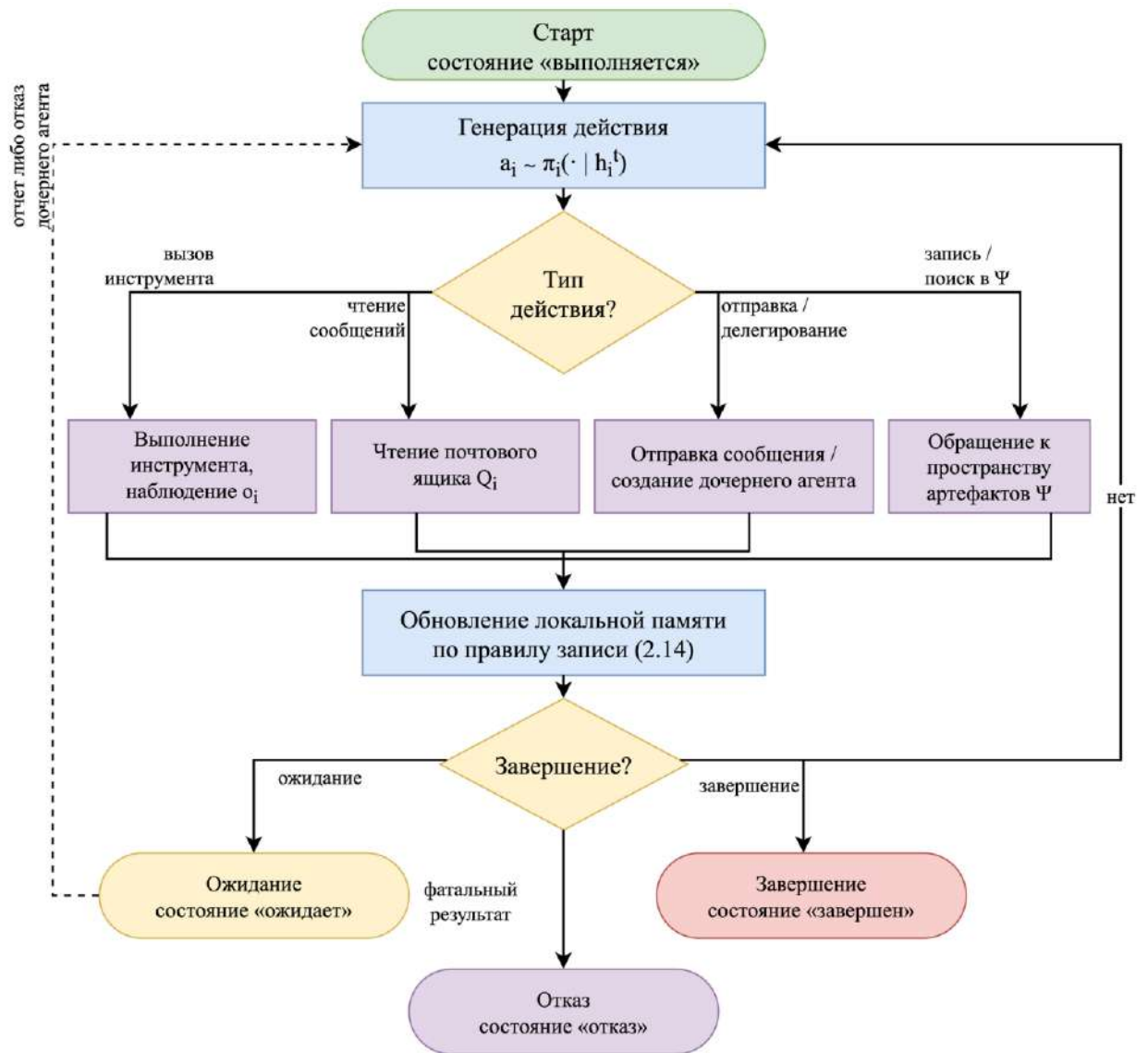


Рисунок 2.5 – Основной цикл работы агента

Выходы из цикла отвечают передаче отчета о завершении, переходу в ожидание результата дочернего агента и прекращению работы при фатальном результате операции.

Состояние агента в жизненном цикле принимает значения из множества $\tilde{S} = \{\text{«выполняется»}, \text{«ожидает»}, \text{«завершен»}, \text{«отказ»}\}$. Состояние «выполняется» отвечает генерации действий, «ожидает» — приостановке при синхронном делегировании, «завершен» — переданному отчету о завершении, «отказ» — прекращению работы вследствие фатального результата операции, при котором продолжение работы агента невозможно. Извещение об отказе формируется

правилом записи (2.14) и действий отказавшего агента не требует. Генерацию выполняют лишь агенты в состоянии «выполняется».

Ход сеанса фиксируется журналом сеанса — последовательностью совместных действий и наблюдений от начального момента до завершения; журнал заверщенного сеанса замыкается останавливающим действием. Журнал составляет наблюдаемую часть трассы исполнения, в отличие от трассы он не содержит состояний процесса, недоступных прямому наблюдению.

Формальным бездействием называется выделенный элемент множества A_i , вводимый архитектурой и не порождающий операций; содержательным называется действие агента, отличное от формального бездействия. Формальным наблюдением называется выделенный элемент множества Ω_i , его получает компонентой совместного наблюдения агент, не выполнявший на шаге содержательного действия. Функция наблюдений (2.15) приписывает формальному наблюдению вероятность единица при всяком значении компоненты комплекса, поэтому в наблюдательном сомножителе одношагового перехода (2.17) оно дает один и тот же множитель во всех состояниях и при нормировке пересчитанной веры сокращается. Отсутствие ответа комплекса на выполненную диагностическую операцию формальным наблюдением не является — это одно из значений наблюдения, различимое среди прочих ответов и изменяющее веру наравне с ними. Обе служебные величины вводятся архитектурой независимо от конфигурации агента.

Совместному действию a композитной модели отвечает на шаге набор компонент всех агентов. Агенты в состоянии «выполняется» вносят содержательные компоненты, агенты в состояниях «ожидает», «завершен» и «отказ» — формальное бездействие. Одновременность операций разных агентов выражена тем самым совместностью действия, а декартова структура $A(s)$ правила (2.16) сохраняется при переменном составе действующих агентов.

Структура агентской системы в решаемой задаче не может быть задана при проектировании, поскольку состав исследуемого комплекса до обследования не известен. Число узлов, перечень служб и значения конфигурационных параметров

раскрываются поступательно, по мере целенаправленного взаимодействия со средой. Фиксация состава агентов и связей заранее означала бы предопределение декомпозиции задачи до получения сведений, от которых эта декомпозиция зависит, — архитектура с заданной структурой вынуждена либо покрывать наихудший случай избыточным числом агентов, либо оставлять часть комплекса вне обследования. Неполная наблюдаемость исключает и промежуточный вариант с однократным вычислением структуры по результатам предварительного обследования, поскольку каждое диагностическое действие способно изменить представление о том, какие направления обследования необходимы и сколько исполнителей они требуют. Единственным носителем актуальных сведений о потребной декомпозиции выступает информационная компонента состояния, формируемая в ходе самого обследования.

Стационарное правило допустимых действий (2.16) задает эту зависимость на уровне отдельного шага. Создание дочернего агента принадлежит множеству действий, поэтому допустимость порождения нового исполнителя определяется накопленными сведениями наравне с допустимостью диагностических операций. Совокупным итогом таких шагов оказывается динамика состава системы — множество действующих агентов в каждый момент есть накопленный итог решений о делегировании, принятых по ходу поступления сведений. Число уровней при этом закреплено проведенным анализом масштабов и в ходе работы не изменяется, изменяются наполнение тактического и операционного уровней и связи создания. Структура системы входит в компоненту S_{agent} композитного состояния наравне с диагностическими сведениями, она порождается действиями самих агентов и изменяется от шага к шагу. Формализацией этой переменной состояния служит граф делегирования.

Автором граф делегирования формализуется как размеченный ориентированный граф, входящий в информационную компоненту состояния (2.31).

$$G = \langle I_G, E_G, \chi \rangle, \quad (2.31)$$

где

$I_G \subseteq I$ — множество вершин: агенты, действующие или действовавшие в сеансе к текущему шагу;

$E_G \subseteq I_G \times I_G$ — множество ориентированных ребер делегирования: ребро (α_i, α_j) означает, что агент α_i создал агента α_j и делегировал ему задачу;

χ — разметка вершин.

Значение графа на шаге t читается из состояния $s_{\text{agent}}(t)$. Штрих у обозначения отвечает значению на следующем шаге, как в записи (2.17). Разметка сопоставляет каждой вершине назначенную задачу и текущее состояние жизненного цикла (2.32).

$$\chi: I_G \rightarrow \mathcal{E} \times \tilde{\mathcal{S}}, \quad \chi(\alpha_i) = (z_i, \tilde{s}_i), \quad (2.32)$$

где

$z_i \in \mathcal{E}$ — описание задачи, назначенной агенту α_i : цель обследования для координатора, задача направления для агента-руководителя, задача отдельной операции для исполнителя;

$\tilde{s}_i \in \tilde{\mathcal{S}}$ — состояние жизненного цикла агента α_i .

Вышестоящим агентам сведения разметки доступны через отчеты, извещения об отказе и сообщения. Состояние «завершен» дочернего агента открывает обработку его отчета, состояние «отказ» — создание замены либо пересмотр направления.

Начальное значение графа задано постановкой сеанса (2.33).

$$G_0 = \langle \{\alpha_1\}, \emptyset, \chi_0 \rangle, \quad \chi_0(\alpha_1) = (z_1, \text{«выполняется»}), \quad (2.33)$$

где

$z_1 \in \mathcal{E}$ — цель обследования, поставленная сеансу извне.

Изменение графа на шаге задается присоединением приращений (2.34).

$$I'_G = I_G \cup I_G^+, \quad E'_G = E_G \cup E_G^+, \quad (2.34)$$

где

I_G^+ — множество агентов, созданных операцией делегирования шага: операция входит компонентой в совместное действие a и определяет создателя — агента, которому принадлежит компонента, — создаваемого агента и назначаемые ему задачу и конфигурацию (2.29); при отсутствии такой компоненты приращения пусты;

E_G^+ — множество ребер, добавленных этой операцией: ребро ведет от создателя к созданному агенту.

Правило (2.34) второго закона переходов не образует — оно задает правило записи (2.14), суженное на графовую компоненту информационного состояния. Аргументами служат текущее значение графа и событие шага, номер шага в них не входит.

Каждый создаваемый агент не принадлежит графу до создания, а создатель принадлежит; создаваемый агент получает ровно одно входящее ребро (2.35).

$$I_G^+ \cap I_G = \emptyset, \quad E_G^+ \subseteq I_G \times I_G^+, \quad \forall w \in I_G^+ \exists! v \in I_G: (v, w) \in E_G^+. \quad (2.35)$$

Создание допустимо только на смежном нижнем уровне. Координатор создает агентов-руководителей, агенты-руководители — исполнителей, исполнители дочерних агентов не создают (2.36).

$$E_G^+ \subseteq (I_{str} \times I_{tac}) \cup (I_{tac} \times I_{oper}). \quad (2.36)$$

Задача вершины неизменна на протяжении жизненного цикла, $z'_i = z_i$ для всех $\alpha_i \in I_G$. Состояние жизненного цикла агента $\alpha_i \in I_G$ обновляется событиями шага по правилу (2.37).

$$\tilde{s}'_i = \begin{cases} \text{«отказ»,} & U_1; \\ \text{«завершен»,} & U_2; \\ \text{«ожидает»,} & U_3; \\ \text{«выполняется»,} & U_4; \\ \tilde{s}_i, & \text{в остальных случаях,} \end{cases} \quad (2.37)$$

где

U_1 — результат операции агента α_i фатальный;

U_2 — компонента a агента α_i есть передача отчета о завершении;

U_3 — компонента a агента α_i есть синхронное делегирование;

U_4 — агент α_i создан на текущем шаге, либо $\tilde{s}_i = \text{«ожидает»}$ и хотя бы один подчиненный агент w , $(\alpha_i, w) \in E_G$, передал отчет либо отказал.

Строки (2.37) применяются в порядке записи, состояние определяет верхняя из строк с истинным условием, чем переход задан однозначно. Условие отказа читает результат операции агента, условия завершения и ожидания — компоненту его действия; у агента, не выполнявшего на шаге содержательного действия, результата операции нет, и строка отказа на нем не срабатывает. Строка

«выполняется» возвращает создателя в работу по событию любого его дочернего агента w — передача отчета есть компонента действия агента w , отказ есть фатальный результат его операции; обработку поступившего и решение о дальнейшем ожидании создатель выполняет очередной генерацией. Задача создаваемой вершины назначается операцией делегирования, ее начальное состояние дает строка «выполняется» по условию принадлежности множеству I_G^+ .

Динамика формирования графа делегирования показана на рисунке 2.6.

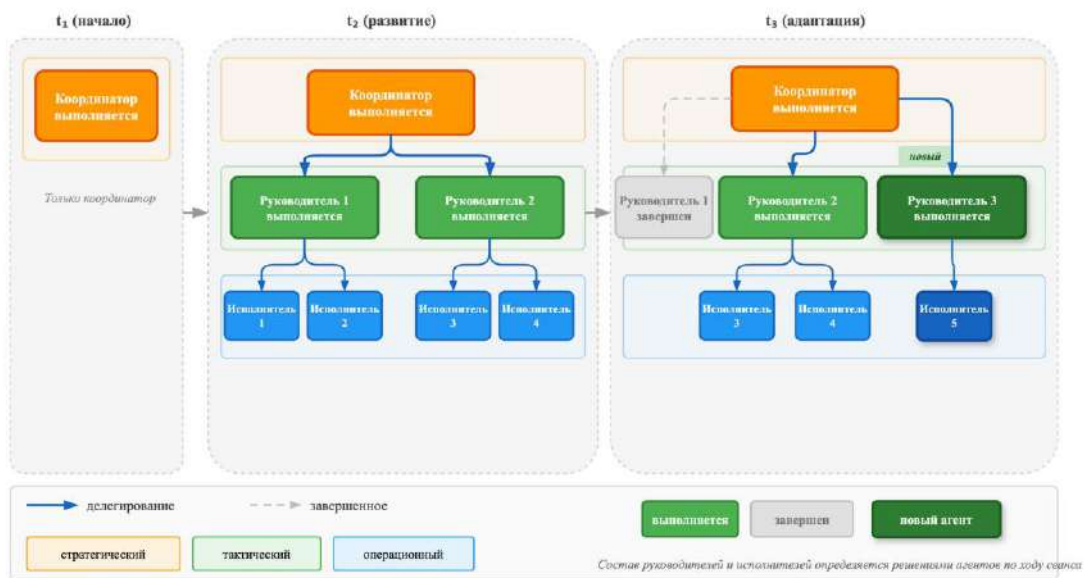


Рисунок 2.6 – Динамика формирования графа делегирования

Рост графа вширь следует за структурой исследуемого комплекса и ограничен бюджетом $N_{\max}$ — операция делегирования составляет содержательное действие и списывается с бюджета наравне с диагностическими операциями. Глубина графа задана архитектурой и от комплекса не зависит, поскольку создание допустимо только на смежном нижнем уровне (2.36), и цепочка от координатора к исполнителю содержит не более двух связей делегирования. Система масштабируется под размер объекта числом агентов, сохраняя постоянную длину цепочки передачи задач и отчетов.

Модель архитектуры конкретизирует композитное пространство состояний. Компоненту S_{agent} составляют локальные памяти агентов, почтовые ящики,

конфигурации действующих агентов, содержимое пространства артефактов, граф делегирования с разметкой и остаток бюджета действий; компонента S_{env} описывает состояние исследуемого комплекса K . Эволюция всех перечисленных структур — дописывание памяти, доставка сообщений и извещений, назначение конфигурации при делегировании, пополнение пространства артефактов, приращение графа, переходы разметки (2.37), списание шага с бюджета — выполняется правилом записи (2.14) по действию и наблюдению шага.

Принятая ранее конечность компоненты S_{agent} обеспечивается ограниченностью памятей и очередей и конечностью множества Ξ , элементы которого суть тексты ограниченной длины над конечным алфавитом. Тем же обеспечивается и конечность множеств действий A_i : параметрами операций — системными инструкциями, описаниями задач, сообщениями и описаниями артефактов — служат элементы того же множества Ξ .

Правило f открывает агенту содержательные операции только из назначенного состава $\tilde{A}_i$ его конфигурации; служебные компоненты доступны независимо от конфигурации, причем останавливающая входит в каждое допустимое множество согласно условию завершения. Совместная политика $\pi = (\pi_i)_{i \in I}$ содержит по компоненте на каждого агента множества I — фактическое участие агента в сеансе выражается его присутствием в графе делегирования, а компоненты недействующих агентов на выполненных шагах сводятся к формальному бездействию.

Модель архитектуры замыкает формальную постановку на исполняющую систему. Уровни иерархии отвечают масштабам целеполагания, тактического управления и исполнения, локальная память агента, наследование контекста при делегировании, почтовый ящик и пространство артефактов обеспечивают накопление и перенос сведений, а граф делегирования формализует структуру системы как переменную состояния, порождаемую решениями самих агентов.

Модель архитектуры определяет структуру агентской системы и механизмы координации агентов, но не устанавливает, каким образом оценивается заверченный сеанс и как по этой оценке дообучается единая модель. Оценка нужна

для каждого сеанса, на котором ведется обучение, поэтому она должна вычисляться без участия эксперта. Для решения этой задачи необходимо формализовать среду, в которой результат сеанса проверяется автоматически.

Выводы по главе 2

Во второй главе разработаны математическая модель оценки работоспособности распределенных программно-аппаратных комплексов и параметрическая модель архитектуры многоагентной интеллектуальной системы. Получены следующие основные результаты.

1. Выполнена формальная постановка задачи оценки работоспособности распределенных программно-аппаратных комплексов. Комплекс определен кортежем $K = \langle N, E, \Sigma, C \rangle$ из множеств вычислительных узлов, сетевых соединений, программных служб и конфигурационных параметров. Работоспособность введена автором как мера $W(K)$ со значениями в отрезке $[0, 1]$, вычисляемая как взвешенное геометрическое среднее степеней соответствия контролируемых показателей требованиям, а качество обследования характеризуется близостью оценки $\hat{W}(K)$ к истинному значению меры. Обоснован выбор децентрализованного частично наблюдаемого марковского процесса принятия решений как аппарата, одновременно учитывающего частичную наблюдаемость, множественность агентов, стохастичность переходов и кооперативный характер взаимодействия, и показано, что в классическом виде он задачу не описывает.

2. Разработана математическая модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений, отличающаяся композитным пространством состояний, объединяющим состояние комплекса с накопленными сведениями агентской системы, зависимостью множества допустимых действий от накопленных сведений и

оценкой завершенной трассы исполнения вместо пошагового вознаграждения. Информационная компонента ведется фиксированным правилом записи, допустимое множество $A(s) = f(s_{\text{agent}})$ изменяется только записью сведений, а условие завершения с бюджетом действий N_{max} гарантирует конечность сеанса.

3. Показано, что композитный процесс сохраняет марковское свойство и однородность по времени (утверждение 2.1), а состояние веры расщепляется на распределение по компоненте комплекса и фактическое значение информационной компоненты (утверждение 2.2). Установлено, что оптимальная совместная политика существует и достигается на детерминированной политике (утверждение 2.3), а оценка завершенной трассы, допускающая разложение на вклады шагов, совместима с уравнением оптимальности над состоянием веры композитной модели (утверждение 2.4), чем подтверждена корректность модификации.

4. Получена оценка мощности композитного пространства состояний $2^{(k+m) \cdot |N|}$, при $k = 4$ признаках состояния узла и $m = 3$ признаках информационного состояния на каждый из $|N| = 10$ узлов превышающая 10^{21} . Вместе с полнотой задачи точного решения для класса недетерминированного экспоненциального времени это обосновывает переход к приближенным алгоритмам на основе машинного обучения.

5. Разработана параметрическая модель архитектуры многоагентной интеллектуальной системы оценки работоспособности с трехуровневой иерархией агентов на основе единой большой языковой модели. Уровням отвечают три несводимых масштаба принятия решений, глобальное целеполагание единственного координатора, тактическое управление агентов-руководителей и исполнение операций исполнителями, а разделение ролей несет конфигурация агента из системной инструкции и назначенного состава операций. Обмен сведениями ведется через локальную память, наследование контекста при делегировании, почтовый ящик и пространство артефактов.

6. Формализован как переменная состояния древовидный динамический граф делегирования, вершины и ребра которого порождаются решениями самих агентов в процессе функционирования под конкретный исследуемый комплекс. Правила

порождения допускают создание агента только на смежном нижнем уровне с единственным входящим ребром, а операция делегирования списывается с бюджета действий, поэтому рост графа вширь ограничен бюджетом, цепочка от координатора к исполнителю содержит не более двух связей делегирования, и система масштабируется под состав и размер комплекса при постоянной глубине подчинения.

Результаты формальной постановки задачи, модификации модели и модели архитектуры получены автором лично и были частично опубликованы в [4; 6; 8; 9; 10; 11; 46], в том числе в соавторстве (вклад автора более 80 %).

3 Алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов

3.1 Метод кросс-уровневого обучения в среде с верифицируемой обратной связью

Испытательная среда порождает экземпляры комплекса программно, с заданными составом, состоянием и значениями конфигурационных параметров, поэтому истинное состояние каждого экземпляра и истинное значение меры работоспособности $W(K)$ известны по построению. Качество обследования характеризуется близостью оценки $\hat{W}(K)$, построенной агентской системой за сеанс, к истинному значению. Для действующего комплекса истинное значение не известно, и близость невычислима. В испытательной среде она вычисляется сопоставлением двух значений, без участия эксперта. На такой обратной связи и ведется дообучение единой модели.

Под средой с верифицируемой обратной связью понимается программная среда, в которой результат сеанса оценивается однозначно, без участия человека-эксперта. Испытательная среда, дополненная генератором целей и функцией верификации, образует такую среду и формально определяется кортежем (3.1) [71; 119].

$$Env = \langle K_{gen}, G_{gen}, V \rangle, \quad (3.1)$$

где

K_{gen} — генератор экземпляров, распределение на множестве экземпляров комплекса; экземпляр $K \sim K_{gen}$ есть состав $\langle N, E, \Sigma, C \rangle$ вместе с истинными значениями показателей;

G_{gen} — генератор целей, условное распределение на множестве целей при заданном экземпляре, $z \sim G_{gen}(\cdot | K)$;

V — функция верификации, сопоставляющая завершенной трассе τ , экземпляру K и цели z оценку достижения цели $q = V(\tau, K, z) \in \mathbb{R}$.

Верифицируемость обратной связи обеспечивается тем, что истинное состояние экземпляра известно по построению, поэтому оценка достижения цели вычисляется по трассе, этому состоянию и цели без участия человека. Автоматическая верификация допускает произвольное число сеансов и оценивает их единообразно. Генератор целей выдает лишь цели, для которых функция верификации определена. Верификация выполняется однократно, после завершения сеанса, а ее вещественные значения позволяют упорядочивать сеансы и могут отражать частичное достижение цели. В пределах сеанса экземпляр и цель фиксированы, поэтому далее пишется $V(\tau)$; при фиксированных экземпляре и цели функция верификации есть вещественная функция на множестве завершенных трасс, как и оценка завершенной трассы.

В решаемой задаче экземпляры порождаются с варьируемыми составом и состоянием, а множество целей состоит из одной цели, которая ставится координатору как задача построить оценку $\hat{W}(K)$ меры работоспособности. Эталоном служит истинное значение $W(K)$, вычисляемое по истинным значениям показателей экземпляра. Оценка достижения цели вычисляется как (3.2)

$$q = V(\tau, K, z) = 1 - |\hat{W}(K) - W(K)|, \quad (3.2)$$

где

$\hat{W}(K)$ — оценка меры работоспособности, построенная агентской системой за сеанс с трассой τ ;

$W(K)$ — истинное значение меры для экземпляра K .

Оба значения лежат в отрезке $[0, 1]$, поэтому и $q \in [0, 1]$, и значения q для сеансов на разных экземплярах сопоставимы между собой. Значение 1 достигается при совпадении оценки с истинным значением. Значение 0 отвечает наибольшему расхождению, оценке 1 для экземпляра с нулевой мерой или оценке 0 для экземпляра с единичной мерой. Сеансу, завершённому без построения оценки, приписывается $q = 0$. Способ порождения экземпляров, состав множества целей и

вид функции верификации кортежем не закреплены, компоненты заменяются независимо друг от друга.

Автором разработана схема среды с верифицируемой обратной связью, отражающая взаимодействие агентской системы с генераторами экземпляров и целей и функцией верификации. Данная схема представлена на рисунке 3.1.



Рисунок 3.1 – Среда с верифицируемой обратной связью

Генераторы экземпляров K_{gen} и целей G_{gen} порождают экземпляр комплекса $K \sim K_{gen}$ и цель $z \sim G_{gen}(\cdot | K)$. Агентская система выполняет сеанс в условиях частичной наблюдаемости. Сеанс образует завершенную трассу τ , которая вместе с экземпляром и целью поступает в функцию верификации V . Функция верификации возвращает оценку достижения цели $q = V(\tau, K, z)$, с которой трасса поступает в контур обучения.

Обучение агентской системы с иерархической архитектурой принципиально отличается от обучения одноагентной системы или плоской совокупности равноправных агентов. Качество работы стратегического агента зависит от того, насколько хорошо тактические и операционные агенты исполняют назначенные задачи. Одновременно качество работы нижестоящих агентов определяется тем, насколько адекватные задачи формулирует вышестоящий уровень. Эта взаимная

зависимость делает изолированное обучение каждого уровня малоэффективным. Улучшение политики одного уровня при неизменных политиках остальных может не привести к улучшению общего результата или даже ухудшить его, поскольку оптимальное поведение на данном уровне зависит от поведения смежных уровней.

Автором предлагается метод кросс-уровневого обучения, основанный на совместной оптимизации политик всех уровней иерархии [7]. Все уровни архитектуры параметризуются единой обучаемой моделью M_θ с параметрами θ . Обучение производится на полных трассах исполнения, содержащих решения и действия всех уровней, а не на изолированных фрагментах поведения отдельных агентов. Отбор обучающего материала осуществляется по оценке завершённой трассы, что устраняет необходимость в экспертной разметке или ручном создании эталонных решений.

Модель M_θ реализует политику π_i каждого агента, определённую в разделе 2.2 как отображение локальной истории в распределение над действиями, и действие a выбирается случайно из распределения $P_\theta(\cdot \mid h; T)$, которое модель порождает по локальной истории h при температуре генерации T . Под температурой генерации понимается стандартный параметр генерации текста языковой моделью, задающий разброс этого распределения. При $T = 1$ распределение совпадает с распределением $P_\theta(\cdot \mid h)$, которое модель присваивает действиям, при $T \rightarrow 0$ генерация становится детерминированной, с ростом T распределение выравнивается. Случайность выбора действий при $T > 0$ даёт трассам разброс, G сеансов на одном экземпляре завершаются разными трассами с разными значениями оценки, и этот разброс делает возможным отбор по оценке. Температура T и число сеансов на экземпляр G входят в гиперпараметры метода [58].

Оценка завершённой трассы $R(\tau)$ есть целевая величина решаемой задачи; метод пользуется ею только через упорядочение трасс, и её конкретный вид методом не фиксируется. Среда сама вычисляет единственную оценку сеанса, оценку достижения цели, поэтому далее оценкой завершённой трассы служит вердикт среды, $R(\tau) = V(\tau)$. Не все трассы одинаково полезны для обучения. Трассы

с низкими значениями $R(\tau)$ содержат преимущественно неудачные стратегии, воспроизведение которых нежелательно. Трассы с высокими значениями $R(\tau)$ содержат успешные стратегии, которые модель должна научиться воспроизводить и развивать. Для отбора полезного обучающего материала автором предлагается механизм формирования элитного множества.

Формально элитное множество определяется как (3.3)

$$D_{elite} = \{\tau \in D_{all} | R(\tau) \geq R_{threshold}\}, \quad (3.3)$$

где

D_{elite} — множество элитных трасс, отобранных для обучения;

D_{all} — множество трасс текущей итерации, по G сеансов на каждом экземпляре итерации. Множество D_{all} формируется заново на каждой итерации и не включает трассы из предыдущих итераций;

$R(\tau)$ — оценка завершенной трассы τ , вещественное число;

$R_{threshold}$ — пороговое значение, определяющее границу отбора.

Автором предлагается адаптивное определение порогового значения на основе распределения оценок текущей итерации (3.4)

$$R_{threshold} = Percentile(R(D_{all}), \beta), \quad (3.4)$$

где

$Percentile(R(D_{all}), \beta)$ — β -я процентиль распределения значений $R(\tau)$ по множеству D_{all} , где $R(D_{all}) = \{R(\tau) | \tau \in D_{all}\}$ — мультимножество оценок всех трасс текущей итерации;

$\beta \in (0, 1)$ — параметр селективности, определяющий долю трасс, отбираемых в элитное множество.

При $\beta = 0,75$ в элитное множество попадает верхняя четверть трасс по оценке завершенной трассы. Преимущество адаптивного порога перед фиксированным проявляется на разных стадиях обучения. На ранних итерациях обучения, когда агенты еще не освоили эффективные стратегии и значения $R(\tau)$ в целом низкие, адаптивный порог автоматически снижается, что позволяет включать в обучающую выборку даже частично успешные трассы. На поздних итерациях,

когда общее качество поведения возрастает, порог автоматически повышается, стимулируя дальнейшее совершенствование.

Включение частично успешных трасс в обучающую выборку является принципиальным решением, отличающим предлагаемый метод от подходов, использующих только полностью успешные сеансы. Обоснование данного решения связано со стохастической природой процесса обучения. На начальных этапах агенты не владеют достаточными навыками для полного решения задачи. Полностью успешные трассы на ранних итерациях могут отсутствовать. Частичные результаты при этом содержат ценную информацию о промежуточных стратегиях, приводящих к локальному прогрессу. Исключение таких трасс приводит к ситуации, когда обучающая выборка оказывается пустой и обучение невозможно.

После формирования элитного множества выполняется структурирование отобранных трасс по уровням иерархии. Трасса содержит на каждом шаге t состояние s_t и совместное действие a_t ; информационная компонента состояния включает локальные истории h_i^t всех агентов, а совместное действие содержит компоненту a_i^t каждого агента. Обучающее подмножество уровня составляют пары «локальная история, действие» агентов этого уровня, выполнивших на шаге содержательное действие (3.5)

$$D_k = \{(h_i^t, a_i^t) : \tau \in D_{elite}, 0 \leq t < \kappa, i \in I_k, a_i^t \text{ содержательное}\}, \quad k \in \{str, tac, oper\}, \quad (3.5)$$

где

D_k — обучающее подмножество уровня k ;

h_i^t — локальная история агента α_i к шагу t , читаемая из состояния s_t трассы τ ;

a_i^t — компонента агента α_i в совместном действии a_t той же трассы;

κ — момент завершения трассы τ ;

I_k — множество агентов уровня k .

Модель обучается воспроизводить действие по локальной истории. Результат действия в пару не входит. Наблюдение для исполнителя, отчет или найденная

запись для агентов верхних уровней дописываются в локальную историю и входят в контекст последующих пар.

Каждая обучающая пара представляет собой фрагмент успешного поведения. Для стратегического уровня пара «локальная история, действие» описывает ситуацию, в которой координатор выбрал действие, приведшее к общему успеху сеанса. Для тактического уровня аналогичная пара описывает ситуацию, в которой агент-руководитель выбрал способ декомпозиции задачи, оказавшийся эффективным. Для операционного уровня пара описывает ситуацию, в которой исполнитель выбрал действие, приведшее к получению полезной информации.

Совокупное обучающее множество формируется как объединение подмножеств всех уровней (3.6)

$$D_{train} = D_{str} \cup D_{tac} \cup D_{oper}. \quad (3.6)$$

Такое объединение позволяет обучать единую модель M_θ на примерах успешного поведения со всех уровней иерархии одновременно, что составляет основу метода кросс-уровневого обучения.

Дообучение модели выполняется на совокупном множестве D_{train} . Поскольку все уровни архитектуры параметризуются единой моделью M_θ с параметрами θ , обновление параметров осуществляется на основании единой функции потерь (3.7)

$$L(\theta) = -\frac{1}{|D_{train}|} \cdot \sum_{(h,a) \in D_{train}} \log P_\theta(a|h), \quad (3.7)$$

где

$L(\theta)$ — функция потерь;

$P_\theta(a | h)$ — условная вероятность, которую модель с параметрами θ присваивает действию a при локальной истории h .

Функция потерь представляет собой стандартную отрицательную логарифмическую правдоподобность, широко применяемую при дообучении языковых моделей [125]. Минимизация $L(\theta)$ увеличивает вероятность генерации тех действий, которые наблюдались в элитных трассах, при соответствующих локальных историях. Распределение $P_\theta(a | h)$ отвечает температуре $T = 1$, поскольку

температура действует при генерации трасс, а дообучение изменяет само распределение модели.

Существенно, что обучающие пары различных уровней входят в единое множество D_{train} без дополнительного разграничения. Модель не получает явного указания, к какому уровню относится данная пара. Различие между уровнями закодировано в содержании локальной истории h и действия a . Локальная память координатора содержит его системную инструкцию, цель обследования и отчеты агентов-руководителей. Локальная память агента-руководителя содержит его системную инструкцию, назначенную задачу и отчеты исполнителей. Локальная память исполнителя содержит его системную инструкцию, элементарную задачу и назначенный состав инструментов. Модель обучается различать роли по содержанию локальной памяти без внешней метки уровня.

Обновление параметров осуществляется методом стохастической оптимизации. На каждом шаге из D_{train} случайным образом извлекается подмножество (мини-пакет) фиксированного размера B , по которому вычисляется приближенный градиент функции потерь и выполняется шаг оптимизации (3.8)

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L_B(\theta_t), \quad (3.8)$$

где

θ_t — параметры модели на шаге t ;

η — скорость обучения;

$\nabla_{\theta} L_B(\theta_t)$ — градиент функции потерь, вычисленный по мини-пакету B .

Роль обучающего сигнала в предлагаемом методе выполняет не внешний учитель в классическом понимании, а механизм отбора элит. Модель обучается воспроизводить собственное наиболее успешное поведение, отобранное по оценке завершенной трассы. Этот подход является аналогом эволюционных стратегий в оптимизации, где селективное давление формирует адаптивное поведение без явного учителя [94; 95]. В отличие от классического обучения с подкреплением, где пошаговое вознаграждение используется для прямого обновления политики через градиент вознаграждения, предлагаемый метод использует оценку завершенной трассы опосредованно. Оценка $R(\tau)$ определяет, какие трассы

попадают в обучающую выборку, но не участвует непосредственно в вычислении градиента. Градиент вычисляется по функции потерь, имеющей стандартную форму правдоподобности.

Итерация кросс-уровневого обучения начинается с порождения трасс и заканчивается дообучением модели, итерации повторяются заданное число раз.

На первом этапе производится генерация экземпляров и выполнение сеансов. Генератор экземпляров K_{gen} порождает множество экземпляров комплекса, генератор целей G_{gen} ставит для каждого из них цель. Агентская система, параметризованная текущей моделью M_θ , выполняет при температуре генерации T по G сеансов на каждом экземпляре. Каждый сеанс образует завершенную трассу τ_j .

На втором этапе каждой трассе приписывается оценка завершенной трассы $R(\tau_j)$.

На третьем этапе производится формирование элитного множества. Адаптивное пороговое значение $R_{\text{threshold}}$ вычисляется как β -я перцентиль распределения оценок. Трассы, удовлетворяющие условию (3.3), включаются в D_{elite} .

На четвертом этапе производится структурирование по уровням. Из каждой элитной трассы извлекаются пары «локальная история, действие» агентов каждого уровня по формуле (3.5). Формируется совокупное обучающее множество D_{train} по формуле (3.6).

На пятом этапе производится дообучение модели. Параметры модели θ обновляются путем минимизации функции потерь на множестве D_{train} методом стохастической оптимизации.

Формально одна итерация обучения определяется как отображение (3.9)

$$Iter: \theta_k \rightarrow \theta_{k+1}, \quad (3.9)$$

где

θ_k — параметры модели на k -й итерации;

θ_{k+1} — обновленные параметры после выполнения пяти этапов цикла.

Последовательность итераций порождает траекторию в пространстве параметров (3.10)

$$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \dots \rightarrow \theta_K, \quad (3.10)$$

где

θ_0 — начальные параметры модели до обучения;

K — общее число итераций обучения.

Автором разработана схема полного цикла кросс-уровневого обучения. Данная схема представлена на рисунке 3.2.

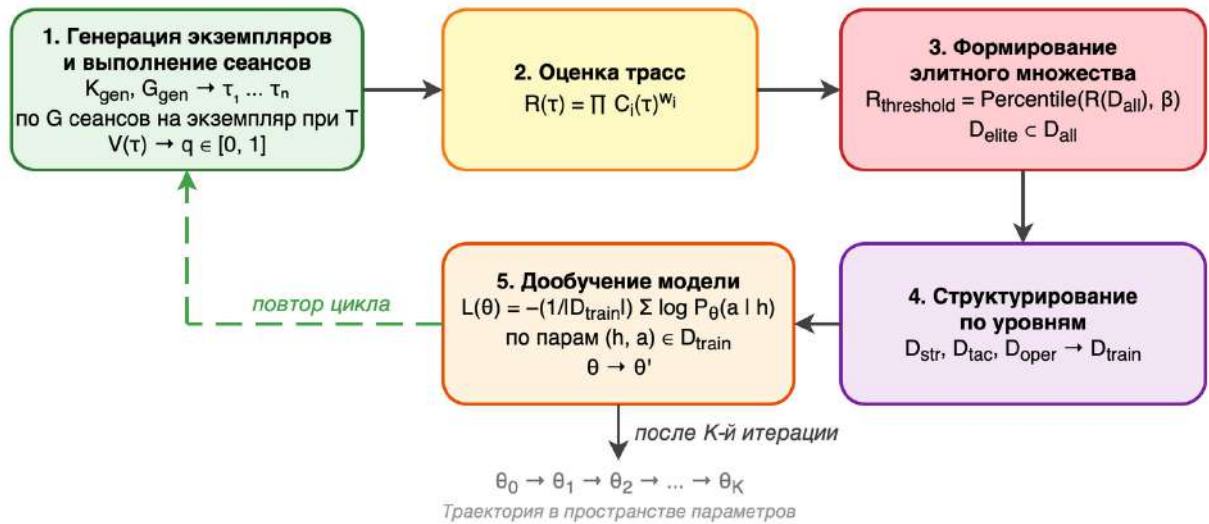


Рисунок 3.2 – Цикл кросс-уровневого обучения

Кросс-уровневый характер обучения обеспечивается тем, что совокупное множество D_{train} содержит пары всех уровней. Улучшение, достигнутое моделью на стратегическом уровне (например, более эффективное распределение задач между тактическими агентами), автоматически отражается в контекстах тактического уровня на последующих итерациях. Улучшение на операционном уровне (например, более точный выбор инструментов) приводит к более успешным сеансам, что повышает оценки элитных трасс и, следовательно, качество обучающего материала для всех уровней. Эта взаимная зависимость формирует положительную обратную связь между уровнями иерархии.

Для практического применения метода кросс-уровневого обучения необходимо определить значения гиперпараметров.

Параметр селективности β определяет долю трасс, отбираемых в элитное множество. Значение $\beta = 0,75$ (верхняя четверть) обеспечивает баланс между объемом обучающего материала и его качеством. При $\beta < 0,5$ в элитное множество попадает более половины трасс, что снижает избирательность и замедляет улучшение. При $\beta > 0,9$ элитное множество оказывается слишком малым для эффективного дообучения, что приводит к переобучению на малой выборке.

Число итераций обучения K определяется скоростью сходимости, которая зависит от начального качества модели и сложности задачи. Для типичных задач оценки работоспособности распределенных комплексов значение K составляет от 3 до 5 итераций.

Скорость обучения η должна быть достаточно малой для предотвращения катастрофического забывания навыков, накопленных моделью до начала дообучения. Типичные значения составляют от 10^{-5} до 10^{-4} , что соответствует стандартным рекомендациям для дообучения больших языковых моделей.

Температура генерации T определяет разброс трасс внутри итерации. При T , близкой к нулю, сеансы на одном экземпляре повторяют друг друга, элитное множество составляется из повторяющихся стратегий, и дообучение не получает новых образцов поведения. При больших T выбор действий отдаляется от распределения модели, растет доля сеансов, завершенных без построения оценки, и порог (3.4) снижается вместе с качеством элитных трасс. Значения T от 0,7 до 1,3 обеспечивают баланс между разнообразием и качеством трасс.

Число сеансов на экземпляр G определяет состав трасс одного экземпляра в D_{all} . При $G = 1$ каждая трасса представляет свой экземпляр, и различие оценок внутри итерации определяется прежде всего различием экземпляров. При $G > 1$ в D_{all} входят разные реализации поведения на одном экземпляре, и отбор по порогу оставляет из них лучшие. Рост G во столько же раз увеличивает число сеансов итерации. Значение G подбирается так, чтобы обеспечить баланс между числом реализаций на экземпляре и стоимостью итерации.

Автором проведена систематизация гиперпараметров метода с указанием рекомендуемых значений и диапазонов. Результаты систематизации представлены в таблице 3.1.

Таблица 3.1 – Гиперпараметры метода кросс-уровневого обучения

Гиперпараметр	Обозначение	Рекомендуемое значение	Допустимый диапазон	Влияние
Селективность	β	0,75	[0,5; 0,95]	Баланс объема и качества выборки
Число трасс на итерацию	$ D_{all} $	50–100	[20; 500]	Статистическая представительность выборки
Число сеансов на экземпляр	G	–	[1; 100]	Число реализаций поведения на одном экземпляре
Температура генерации	T	0,7–1,3	[0,3; 2]	Разброс трасс внутри итерации
Число итераций	K	3–5	[1; 10]	Глубина обучения
Скорость обучения	η	$2 \cdot 10^{-5}$	$[10^{-5}; 10^{-4}]$	Стабильность дообучения
Размер мини-пакета	B	8	[4; 32]	Точность градиента

Автором исследован общий поток данных в методе кросс-уровневого обучения от генерации экземпляров до обновления параметров модели. Результаты представлены на обобщающей схеме (рисунок 3.3).

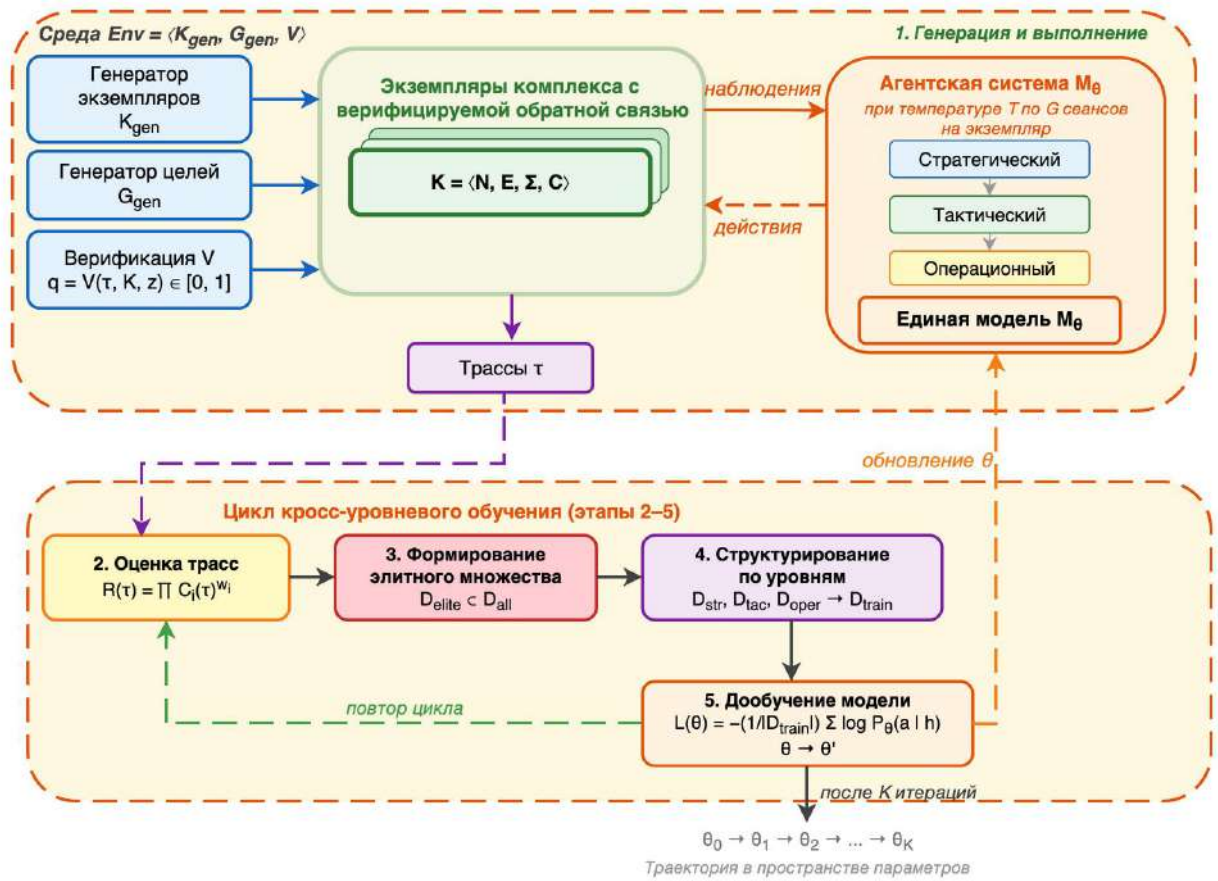


Рисунок 3.3 – Обобщающая схема метода кросс-уровневого обучения

Среда и цикл обучения обмениваются только трассами и обновленными параметрами θ' , и процесс завершается после K итераций.

В частных случаях, обусловленных ограничениями вычислительных ресурсов или спецификой прикладной задачи, допускается использование нескольких специализированных моделей, по одной на каждый уровень архитектуры. В этом случае обучающие подмножества D_{str} , D_{tac} , D_{oper} используются для дообучения соответствующих моделей отдельно. Однако отбор элитных трасс по-прежнему осуществляется по общей для всех уровней оценке завершенной трассы $R(\tau)$, что сохраняет кросс-уровневый характер метода. Решения стратегического уровня, попавшие в элитную трассу, являются элитными именно потому, что привели к успешному исполнению на тактическом и операционном уровнях.

Предлагаемый метод кросс-уровневого обучения в среде с верифицируемой обратной связью формирует замкнутый контур совершенствования агентской системы. Среда обеспечивает объективную автоматическую оценку достижения цели. Механизм отбора элит формирует обучающую выборку из наиболее успешного опыта. Дообучение модели обновляет политики всех уровней иерархии.

При $R(\tau) = V(\tau)$ отбор ведется по вердикту среды, а он характеризует только итог сеанса, поэтому два сеанса с одинаковой оценкой меры получают одно и то же значение q независимо от того, сколько действий потребовалось, какая часть комплекса обследована и как обследование сказалось на его работе, и порог (3.3) не различает сеансы с одинаковым итогом и разной ценой. Метод определяет, как оценка превращается в обучение, но не устанавливает, что считать качественной работой агентской системы. Для этого необходима критериальная модель качества функционирования.

3.2 Оценка завершенной трассы по критериальной модели качества функционирования

При $R(\tau) = V(\tau)$ элитное множество (3.3) формируется по одному вердикту среды, а вердикт не учитывает цену результата, и два сеанса с одинаковой оценкой достижения цели получают одну оценку независимо от того, сколько действий потребовалось, какая часть комплекса обследована и как обследование сказалось на его работе, поэтому в элиту попадает и сеанс, добившийся результата ценой всего бюджета.

Автором предлагается рассматривать несколько критериев, оценивающих сеанс с разных сторон, и объединять их в итоговую оценку завершенной трассы. Набор критериев вместе со способом их объединения составляет критериальную модель качества функционирования [5].

Рассмотрим формальное определение, обоснование и свойства критериев, выделенных автором.

К первому критерию относится полнота анализа, оценивающая степень охвата элементов исследуемого комплекса в ходе одного сеанса. Пусть $K = \langle N, E, \Sigma, C \rangle$ описывает исследуемый комплекс. Обозначим через $U_{total} = N \cup E \cup \Sigma$ полное множество элементов комплекса, подлежащих анализу, где N — множество узлов, E — множество связей, Σ — множество служб. Обозначим через $U(\tau) \subseteq U_{total}$ подмножество элементов, обнаруженных агентами в ходе трассы τ .

Формально полнота анализа определяется как (3.11)

$$C_{cov}(\tau) = |U(\tau)|/|U_{total}|, \quad (3.11)$$

где

$C_{cov}(\tau)$ — значение полноты анализа для трассы τ ;

$|\cdot|$ — мощность множества;

$U(\tau)$ — множество элементов комплекса, обнаруженных агентами в ходе трассы τ ;

U_{total} — полное множество элементов исследуемого комплекса.

Значения $C_{cov}(\tau)$ лежат в отрезке $[0, 1]$. Значение 1 соответствует полному исследованию всех элементов комплекса. Значение 0 соответствует ситуации, когда ни один элемент не обнаружен. Промежуточные значения отражают долю исследованного пространства.

Высокий показатель полноты свидетельствует о том, что агентская система способна к планомерному исследованию пространства состояний и не склонна к «застреванию» на отдельных компонентах комплекса. При этом полнота характеризует широту анализа, а не его глубину. Система, зафиксировавшая наличие множества компонентов, но не исследовавшая ни один из них подробно, будет обладать высокой полнотой, но низкой результативностью.

Ко второму критерию относится достоверность, характеризующая надежность выводов, полученных агентской системой в ходе анализа. Под выводами понимаются утверждения агентской системы о состоянии комплекса, такие как обнаруженные характеристики компонентов, выявленные несоответствия конфигураций, зафиксированные состояния узлов и служб. Формально достоверность определяется как (3.12)

$$C_{acc}(\tau) = |F_{verified}(\tau)|/|F_{total}(\tau)|, \quad (3.12)$$

где

$C_{acc}(\tau)$ — значение достоверности для трассы τ ;

$|\cdot|$ — мощность множества;

$F_{verified}(\tau)$ — подмножество выводов, корректность которых подтверждена сопоставлением с истинным состоянием экземпляра, известным в среде по построению;

$F_{total}(\tau)$ — множество всех выводов, сформированных агентами в ходе трассы τ ; при пустом множестве выводов достоверность принимается равной нулю.

К третьему критерию относится эффективность использования ресурсов, определяющая долю неиспользованного вычислительного бюджета сеанса. Формально эффективность определяется как (3.13)

$$C_{eff}(\tau) = 1 - N_{actions}(\tau)/N_{max}, \quad (3.13)$$

где

$C_{eff}(\tau)$ — значение эффективности использования ресурсов для трассы τ ;

$N_{actions}(\tau)$ — общее число действий, выполненных агентами всех уровней в ходе сеанса;

N_{max} — максимально допустимое число действий в одном сеансе (бюджет), определяемое параметрами конфигурации.

Значения C_{eff} лежат в $[0, 1]$. Значение 1 соответствует сеансу, завершённому без единого действия (теоретический предел). Значение 0 соответствует полному исчерпанию бюджета.

Эффективность измеряет экономность использования ресурсов безотносительно к достигнутому результату. Высокое значение критерия соответствует выполнению сеанса с минимальным числом операций. Система с низкой эффективностью расходует непропорционально много операций, что делает ее непригодной для практического применения в условиях ограниченного вычислительного бюджета.

К четвертому критерию относится неинвазивность, отражающая способность системы получать оценку при минимальном снижении работоспособности

исследуемого комплекса. Снижение работоспособности вызывают отдельные действия обследования, поэтому его объем за сеанс измеряется числом таких действий в долях бюджета сеанса N_{max} . Формально неинвазивность определяется как (3.14)

$$C_{ninv}(\tau) = 1 - N_{impact}(\tau)/N_{max}, \quad (3.14)$$

где

$C_{ninv}(\tau)$ — значение неинвазивности для трассы τ ;

$N_{impact}(\tau)$ — число действий, вызвавших наблюдаемое снижение работоспособности исследуемого комплекса.

Значения $C_{ninv}(\tau)$ лежат в отрезке $[0, 1]$. Значение 1 соответствует сеансу, ни одно действие которого не снизило работоспособность комплекса; сеанс без действий также получает единицу. Значение 0 соответствует сеансу, весь бюджет которого израсходован на действия, снизившие работоспособность. Критерий существен для задач, в которых оценка проводится на работающем комплексе и не должна нарушать его функционирование.

Действий, снизивших работоспособность, не больше, чем действий вообще, поэтому неинвазивность и эффективность связаны неравенством (3.15)

$$C_{ninv}(\tau) = 1 - \frac{N_{impact}(\tau)}{N_{max}} \geq 1 - \frac{N_{actions}(\tau)}{N_{max}} = C_{eff}(\tau), \quad (3.15)$$

в котором оба равенства суть определения (3.14) и (3.13), а неравенство следует из $N_{impact}(\tau) \leq N_{actions}(\tau)$. Разность $C_{ninv}(\tau) - C_{eff}(\tau)$ равна доле бюджета, израсходованной на действия без снижения работоспособности. Оба критерия измеряются в долях одного бюджета, эффективность учитывает каждое действие, а неинвазивность только действия, снизившие работоспособность.

К пятому критерию относится результативность, выражающая достижение цели по завершении трассы. Под результативностью понимается степень реализации запланированной деятельности и достижения запланированных результатов. Результативность формализуется функцией верификации среды (3.1), которая в решаемой задаче задана оценкой (3.2) и принимает значения в отрезке $[0, 1]$. Формально результативность определяется как (3.16)

$$C_{res}(\tau) = V(\tau) = q \in [0,1], \quad (3.16)$$

где

$C_{res}(\tau)$ — значение результативности для трассы τ ;

V — функция верификации из кортежа среды (3.1);

q — оценка достижения цели для трассы τ .

Значение 1 соответствует полному достижению цели. Значение 0 соответствует ситуации, когда цель не достигнута. Промежуточные значения отражают частичное достижение цели. Критерий результативности принципиально отличается от остальных четырех тем, что оценивает итог сеанса, но не качество процесса.

Между пятью критериями одной трассы существуют взаимные компромиссы, которые необходимо учитывать при настройке системы.

Большинство связей между критериями одной трассы проходит через число действий сеанса. Повышение полноты анализа C_{cov} требует выполнения большего числа действий $N_{actions}$, что при фиксированном бюджете N_{max} снижает эффективность C_{eff} ; обе величины линейны по числу действий, поэтому компромисс между ними линеен, пока остаются необнаруженные элементы. Стремление к высокой неинвазивности C_{ninv} ограничивает набор допустимых методов анализа, поскольку действия, дающие наибольший прирост полноты, обычно и сильнее прочих нагружают комплекс, снижая на время оценки его работоспособность; сама неинвазивность линейна по числу таких действий. Достоверность C_{acc} и эффективность C_{eff} находятся в положительной взаимосвязи, так как агент с высокой достоверностью реже совершает бесполезные действия, основанные на ложных гипотезах, что сокращает общее число операций. Результативность C_{res} положительно связана с полнотой C_{cov} , поскольку более широкое исследование компонентов комплекса повышает вероятность достижения цели. Вместе с тем результативность находится в отрицательной зависимости от эффективности C_{eff} , так как высокая результативность, как правило, требует большего числа действий. Достоверность C_{acc} также положительно влияет на результативность C_{res} , поскольку точные диагностические решения приближают

систему к цели быстрее, чем действия, основанные на ошибочных гипотезах. Взаимные зависимости и компромиссы между критериями показаны на рисунке 3.4.

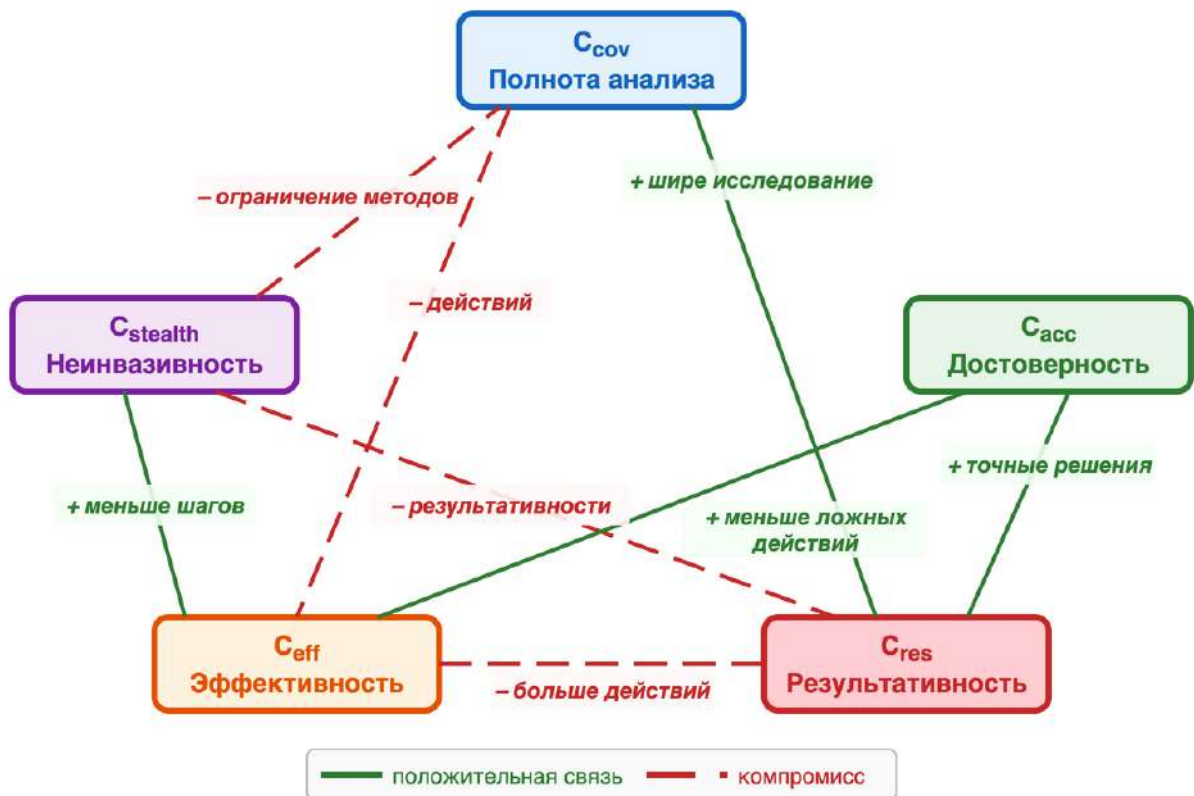


Рисунок 3.4 – Зависимости между критериями одной трассы

Пять рассмотренных критериев оценивают различные стороны сеанса, тогда как отбор элитных трасс (3.3) сравнивает трассы по одному числу. Набор из пяти критериев без объединения задает многоцелевую задачу, не имеющую единственного решения без дополнительных предпочтений; весовые коэффициенты при критериях задают эти предпочтения явно. Для решаемой задачи автором разработана агрегирующая функция полезности, взвешенное геометрическое среднее пяти критериев с настраиваемыми весовыми коэффициентами, далее она называется оценкой завершенной трассы. Той же формой задана мера работоспособности комплекса. Оценка завершенной трассы для решаемой задачи определяется как (3.17)

$$R(\tau) = C_{cov}(\tau)^{w_1} \cdot C_{acc}(\tau)^{w_2} \cdot C_{eff}(\tau)^{w_3} \cdot C_{ninv}(\tau)^{w_4} \cdot C_{res}(\tau)^{w_5}, \quad (3.17)$$

где

$R(\tau)$ — оценка завершенной трассы τ ;

$C_{cov}(\tau)$, $C_{acc}(\tau)$, $C_{eff}(\tau)$, $C_{ninv}(\tau)$, $C_{res}(\tau)$ — значения соответствующих критериев;

w_1, w_2, w_3, w_4, w_5 — весовые коэффициенты, показатели степени при критериях; сомножитель при нулевом весовом коэффициенте принимается равным единице.

Далее критерии в том же порядке обозначаются $C_i(\tau)$, $i = 1, \dots, 5$. Соотношение (3.17) принимается за определение; оно корректно при всех значениях критериев из отрезка $[0, 1]$, поскольку неопределенность вида 0^0 снята соглашением о нулевом весе. На весовые коэффициенты накладывается ограничение нормировки (3.18)

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1, \quad w_i \geq 0, \quad i = 1, \dots, 5. \quad (3.18)$$

При выполнении (3.18) значения $R(\tau)$ лежат в отрезке $[0, 1]$, поскольку каждый сомножитель лежит в этом отрезке. Значение 1 достигается только при единичных значениях всех критериев с положительными весами. Если все критерии равны одному числу, оценка равна этому числу, поскольку показатели степени в сумме дают единицу. Отдельные критерии при этом сохраняют диагностическую ценность, значения C_{cov} , C_{acc} , C_{eff} , C_{ninv} и C_{res} анализируются независимо при интерпретации результатов, тогда как $R(\tau)$ служит только целевым сигналом при обучении.

Мультипликативная форма (3.17) выбрана вместо аддитивной, взвешенной суммы тех же критериев с теми же весами, из-за обнуления и из-за поведения оценки на компромиссе между критериями, и то и другое видно на отборе элит.

Обнуление состоит в том, что оценка (3.17) не допускает взаимной компенсации критериев, а взвешенная сумма допускает ее полностью. По взвешенной сумме сеанс, не обнаруживший ни одного элемента и не давший оценки меры, получает ненулевую оценку за неизрасходованный бюджет и отсутствие воздействий на комплекс и при большом весе этих критериев попадает

в элитное множество (3.3). У оценки (3.17) нулевое значение любого критерия с положительным весом обнуляет оценку при любом наборе весов, удовлетворяющем (3.18), поэтому такой сеанс не попадает в элиту, какими бы ни были остальные критерии, а веса управляют вкладом частичных значений, но не этим требованием. Вне нулевой области оценка (3.17) строго возрастает по каждому критерию с положительным весом, поскольку возрастает соответствующий сомножитель; внутри нее, при нулевом критерии, различия остальных критериев на оценке не сказываются.

Поведение оценки на компромиссе между критериями показывает пример двух критериев, полноты и эффективности, при фиксированных остальных. Пусть каждое действие обнаруживает один элемент комплекса, N_0 — число действий, необходимое для полного охвата, а $N \leq \min(N_0, N_{\max})$ — число выполненных действий. Тогда полнота и эффективность линейны по N и принимают вид (3.19)

$$C_{cov} = \frac{N}{N_0}, \quad C_{eff} = 1 - \frac{N}{N_{\max}}. \quad (3.19)$$

Взвешенная сумма с весами w_{cov} и w_{eff} линейна по N ,

$$w_{cov}C_{cov} + w_{eff}C_{eff} = w_{eff} + \left(\frac{w_{cov}}{N_0} - \frac{w_{eff}}{N_{\max}}\right)N, \quad (3.20)$$

поэтому ее максимум достигается на краю отрезка. При $w_{cov}/N_0 > w_{eff}/N_{\max}$ он лежит при наибольшем N , при обратном неравенстве при $N = 0$, а при равенстве все значения N дают одинаковую оценку. Отбор элит по такой оценке выделяет трассы одной крайней стратегии, пустого сеанса или полного охвата, и переходит от одной к другой при малом изменении отношения весов. Оценка (3.17) ведет себя иначе. При фиксированных остальных критериях от N зависят только два ее сомножителя, а их произведение достигает максимума там же, где его логарифм (3.21)

$$\ln(C_{cov}^{w_{cov}} C_{eff}^{w_{eff}}) = w_{cov} \ln \frac{N}{N_0} + w_{eff} \ln \left(1 - \frac{N}{N_{\max}}\right). \quad (3.21)$$

Производная правой части по N равна $w_{cov}/N - w_{eff}/(N_{\max} - N)$ и обращается в ноль в единственной точке

$$N^* = N_{\max} \frac{w_{cov}}{w_{cov} + w_{eff}}, \quad (3.22)$$

а вторая производная $-w_{\text{cov}}/N^2 - w_{\text{eff}}/(N_{\text{max}} - N)^2$ отрицательна. Максимум внутренний, если N^* не превосходит N_0 , и зависит от весов непрерывно, изменение отношения весов сдвигает его, но не переводит отбор через край отрезка. Элиту при оценке (3.17) образуют трассы с числом действий вблизи N^* , и веса задают это число прямо.

Автором построены графики обеих оценок на компромиссе (3.19) при $N_0 = 50$ и $N_{\text{max}} = 100$, единичных значениях остальных критериев и $w_{\text{cov}} + w_{\text{eff}} = 1$. Графики представлены на рисунке 3.5.

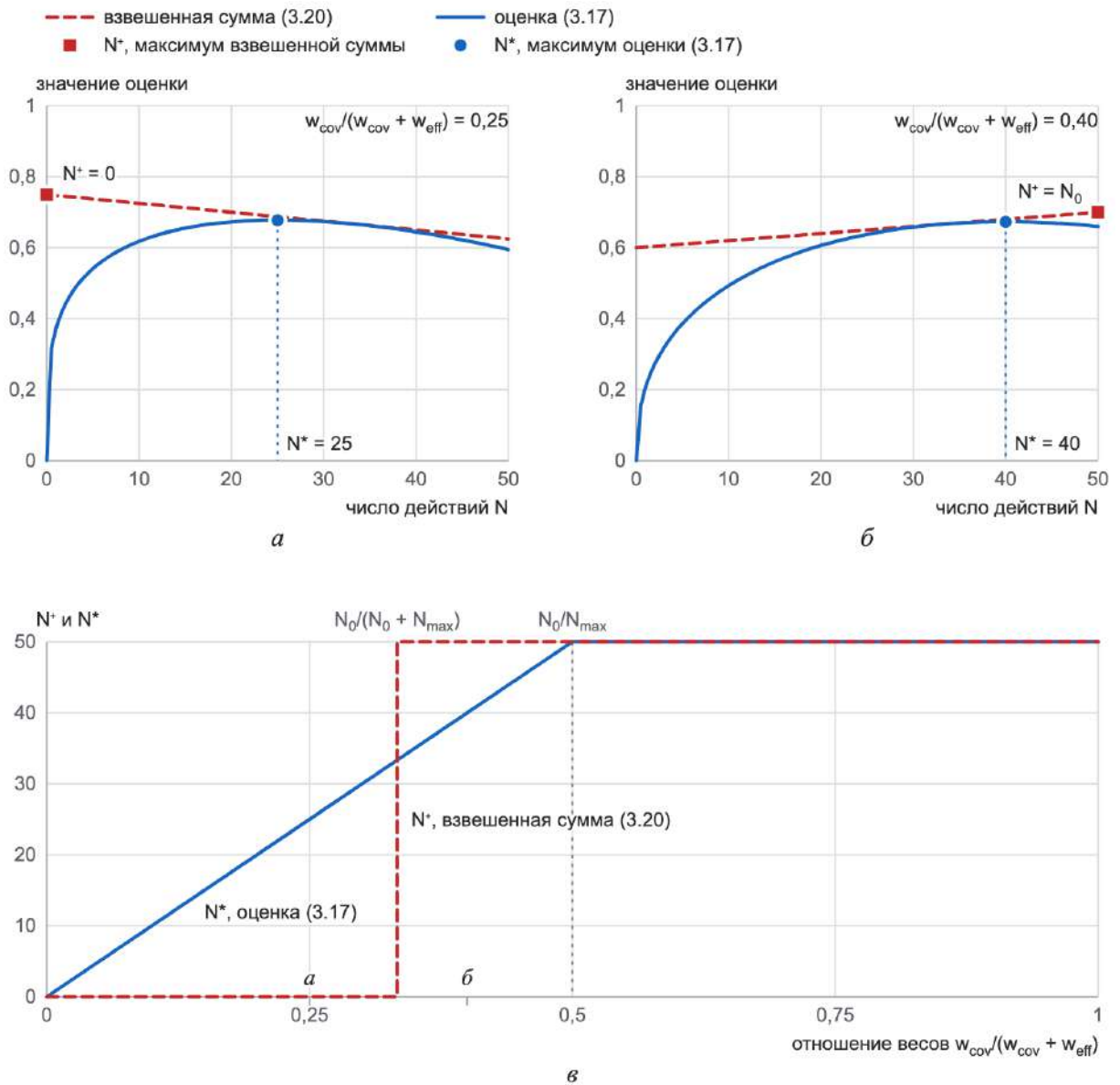


Рисунок 3.5 – Взвешенная сумма (3.20) и оценка (3.17) на компромиссе полноты и эффективности: а, б – зависимость от числа действий при отношении весов

$w_{\text{cov}}/(w_{\text{cov}} + w_{\text{eff}})$, равном 0,25 и 0,40; в – положение максимума в зависимости от отношения весов

При отношении весов 0,25 максимум взвешенной суммы лежит на пустом сеансе, $N^+ = 0$, при отношении 0,40 на полном охвате, $N^+ = N_0$, тогда как максимум оценки (3.17) внутренний, $N^* = 25$ и $N^* = 40$ по (3.22). Сумма всюду не ниже оценки, поскольку взвешенное среднее арифметическое двух чисел не меньше их взвешенного среднего геометрического, и касается ее при $N = N_0 \cdot N_{\text{max}}/(N_0 + N_{\text{max}})$, около 33, где полнота равна эффективности. Точка касания не зависит от весов, N^+ лежит на краю отрезка, а N^* движется вместе с отношением весов. Ступень N^+ переходит от 0 к N_0 скачком при отношении $N_0/(N_0 + N_{\text{max}})$, равном 1/3, при котором все значения N дают одинаковую сумму, а прямая $N^* = N_{\text{max}} \cdot w_{\text{cov}}/(w_{\text{cov}} + w_{\text{eff}})$ достигает N_0 при отношении N_0/N_{max} , равном 0,5.

Различие обеих оценок на компромиссе (3.19) допускает геометрическое прочтение. Пары $(C_{\text{cov}}, C_{\text{eff}})$, достижимые при N от нуля до N_0 , образуют отрезок в пространстве двух критериев, а наибольшее значение оценки на отрезке достигается в точке, которой касается ее наивысшая линия уровня. Автором построены линии уровня обеих оценок при отношении весов 0,40. Линии уровня представлены на рисунке 3.6.

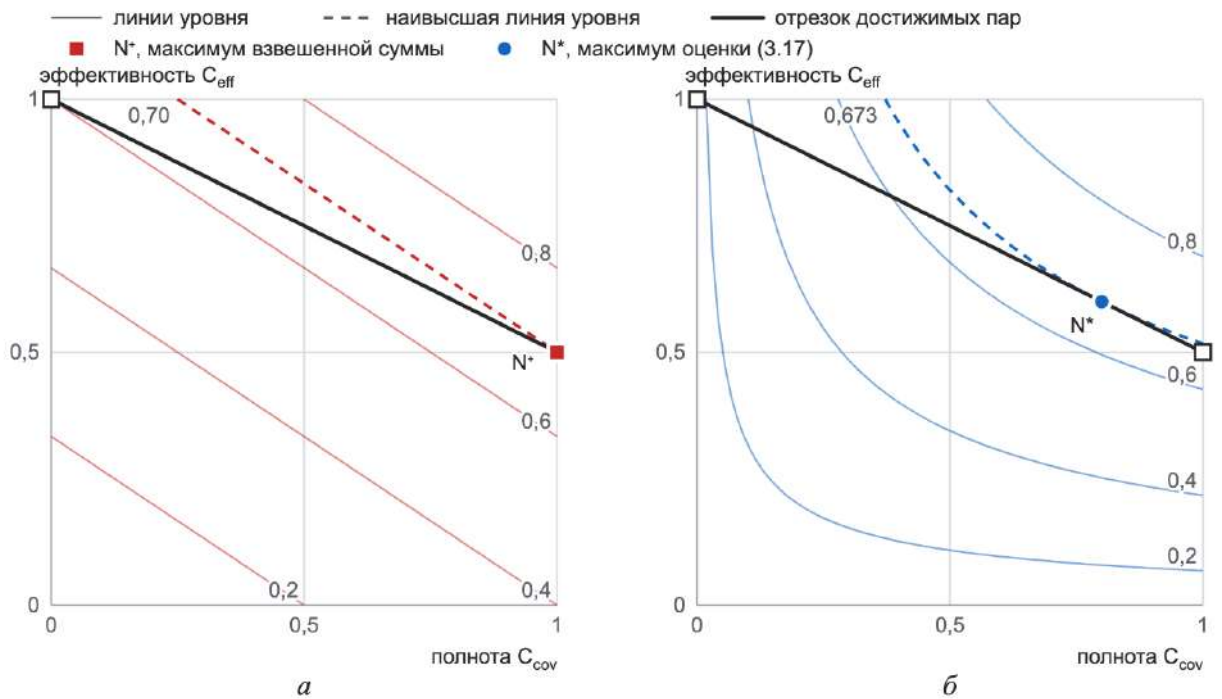


Рисунок 3.6 – Линии уровня в пространстве критериев полноты и эффективности при отношении весов 0,40: а – взвешенная сумма (3.20); б – оценка (3.17)

Линии уровня взвешенной суммы суть параллельные прямые, и наивысшая из них, уровня 0,70, касается отрезка только в его конце (1; 0,5), отвечающем $N^+ = N_0$, поэтому максимум суммы лежит в углу при любом отношении весов, кроме критического, при котором линия уровня совпадает с отрезком. Линии уровня оценки (3.17) обращены выпуклостью к началу координат, и наивысшая из них, уровня 0,673, касается отрезка во внутренней точке (0,8; 0,6), отвечающей $N^* = 40$.

Помимо обнуления, из логарифмической записи (3.23) оценки (3.17)

$$\ln R(\tau) = w_1 \ln C_1(\tau) + w_2 \ln C_2(\tau) + w_3 \ln C_3(\tau) + w_4 \ln C_4(\tau) + w_5 \ln C_5(\tau), \quad (3.23)$$

справедливой при положительных критериях, видны относительная компенсация критериев и малая чувствительность к погрешности весов вблизи полного качества.

Компенсация между критериями носит относительный характер. Пусть значения C_i и C_j двух критериев с положительными весами изменены на C_i' и C_j' при неизменных остальных, и оценка сохранилась. Тогда из логарифмической записи (3.24)

$$w_i \ln C'_i + w_j \ln C'_j = w_i \ln C_i + w_j \ln C_j \quad (3.24)$$

следует условие компенсации (3.25)

$$\ln \frac{C'_j}{C_j} = -\frac{w_i}{w_j} \ln \frac{C'_i}{C_i}. \quad (3.25)$$

В первом порядке по относительным приращениям снижение C_i на 1 % возмещается ростом C_j на w_i/w_j %. Веса имеют смысл эластичностей, изменение критерия на 1 % изменяет оценку на w_i % при любых значениях остальных критериев. Компенсация невозможна для критерия, равного нулю, что согласуется с обнулением.

Чувствительность оценки к погрешности назначения весов видна из той же логарифмической записи. Перенос веса Δw с критерия C_j на критерий C_i дает оценку $R'(\tau)$, для которой выполняется (3.26)

$$\ln R'(\tau) - \ln R(\tau) = \Delta w (\ln C_i(\tau) - \ln C_j(\tau)). \quad (3.26)$$

Вблизи полного качества, при всех критериях, близких к единице, логарифмы близки к нулю, и оценка малочувствительна к погрешности весов; чувствительность растет с разбросом критериев при глубоком провале одного из них. Порядок двух трасс меняется только там, где различие их оценок меньше сдвига, вызванного переносом веса.

Весовые коэффициенты являются параметрами модели и задаются в зависимости от целевого назначения анализа. Автором предложены и систематизированы три типовые конфигурации весовых коэффициентов, представленные в таблице 3.2.

Таблица 3.2 – Типовые конфигурации весовых коэффициентов оценки завершенной трассы

Тип задачи	w_1	w_2	w_3	w_4	w_5	Приоритет
Комплексный аудит	0,25	0,25	0,15	0,10	0,25	Широта, надежность и результат
Оценка под рабочей нагрузкой	0,15	0,15	0,15	0,35	0,20	Минимальное воздействие
Быстрая точечная оценка	0,10	0,15	0,30	0,10	0,35	Скорость достижения цели

Для задач комплексного аудита приоритет отдается полноте, достоверности и результативности, что обеспечивает широкий охват компонентов с надежной верификацией достигнутого состояния. Для задач оценки комплекса под рабочей нагрузкой возрастает значимость неинвазивности, что приближает условия оценки к штатному режиму работы. Для задач быстрой точечной оценки приоритет отдается результативности и эффективности, поскольку требуется достичь цели за минимальное число действий. Поскольку веса суть эластичности, набор весов задает, на сколько процентов оценка отзывается на процент каждого критерия, и этим задает направление отбора элит.

Оценка завершенной трассы характеризует один сеанс. Сохранение качества при переходе между экземплярами комплекса по одной трассе не оценивается; для этого автором вводится шестой критерий, устойчивость, вычисляемый по множеству трасс на различных конфигурациях исследуемого комплекса. Пять критериев одной трассы образуют первую категорию модели, устойчивость образует вторую. Обозначим через N_{trials} общее число проведенных сеансов, а через $N_{success}$ число сеансов, для которых оценка достижения цели не ниже минимального порога, то есть $V(\tau) \geq q_{min}$.

Формально устойчивость определяется как (3.27)

$$C_{robust} = N_{success} / N_{trials}, \quad (3.27)$$

где

C_{robust} — значение устойчивости;

$N_{success}$ — число сеансов, для которых $V(\tau) \geq q_{min}$;

$q_{min} \in (0, 1]$ — наименьшая оценка достижения цели, при которой сеанс считается успешным;

N_{trials} — общее число проведенных сеансов на различных конфигурациях.

Множество конфигураций порождается генератором K_{gen} среды (3.1) варьированием числа узлов, состава служб, топологии связей и значений конфигурационных параметров.

Устойчивость характеризует обобщающую способность обученных агентов и их применимость к ранее не встречавшимся конфигурациям комплекса [20]. Система с высокой устойчивостью успешно справляется с анализом комплексов, существенно отличающихся от тех, на которых проводилось обучение. Система с низкой устойчивостью может показывать высокие значения критериев на знакомых конфигурациях, но терять работоспособность при встрече с новыми.

Для обеспечения статистической значимости оценки устойчивости число испытаний N_{trials} должно быть достаточно большим. Автором предлагается определять минимальное число испытаний из условия (3.28)

$$N_{\text{trials}} \geq 1/(\varepsilon^2 \cdot \delta), \quad (3.28)$$

где

ε — допустимая погрешность оценки;

δ — допустимая вероятность превышения погрешности.

При $\varepsilon = 0,05$ и $\delta = 0,05$ минимальное число испытаний составляет 400, что обеспечивает оценку устойчивости с точностью $\pm 0,05$ с вероятностью не менее 0,95.

Автором предлагается также доверительный интервал для оценки устойчивости (3.29)

$$C_{\text{robust}} \in [\hat{p} - z_{\alpha/2} \cdot \sqrt{\frac{\hat{p}(1 - \hat{p})}{N_{\text{trials}}}}, \hat{p} + z_{\alpha/2} \cdot \sqrt{\frac{\hat{p}(1 - \hat{p})}{N_{\text{trials}}}}], \quad (3.29)$$

где

$\hat{p} = N_{\text{success}} / N_{\text{trials}}$ — точечная оценка устойчивости;

$z_{\alpha/2}$ — квантиль стандартного нормального распределения для уровня значимости α .

Данный интервал позволяет оценить точность полученного значения устойчивости и провести корректное сравнение различных конфигураций системы.

Критерии первой категории и системный критерий оценивают качественно различные свойства системы и вычисляются на различных данных. Критерии первой категории вычисляются по одной трассе и характеризуют поведение агентов в конкретной среде. Системный критерий вычисляется по множеству трасс

и характеризует обобщающую способность. Данное разделение устраняет проблему смешения уровней оценки, выявленную при анализе существующих подходов.

Автором проведена оценка трех гипотетических трасс исполнения поведения агентов при анализе одного и того же комплекса. Трассы различаются числом действий и числом действий, снизивших работоспособность, при бюджете $N_{\max} = 100$. Широкое исследование выполняет 75 действий, из них 45 со снижением работоспособности; целенаправленный анализ выполняет 30 действий, из них 20 со снижением; неинвазивный анализ выполняет 35 действий без единого снижения. Эффективность и неинвазивность вычислены по (3.13) и (3.14), оценки завершённой трассы — по (3.17) при равных весах и при наборе «оценка под рабочей нагрузкой» из таблицы 3.2. Результаты представлены в таблице 3.3.

Таблица 3.3 – Оценка трех трасс поведения по критериям первой категории

Трасса	C_{cov}	C_{acc}	C_{eff}	C_{ninv}	C_{res}	$R(\tau)$ при $w_i = 0,20$	$R(\tau)$ при $w_4 = 0,35$
Широкое исследование	0,90	0,75	0,25	0,55	0,85	0,602	0,601
Целенаправленный анализ	0,45	0,80	0,70	0,80	0,90	0,711	0,736
Неинвазивный анализ	0,40	0,75	0,65	1,00	0,85	0,698	0,758

При равных весах наивысшую оценку $R(\tau) = 0,711$ получает трасса целенаправленного анализа τ_2 за счет высокой результативности $C_{\text{res}} = 0,90$ и эффективности $C_{\text{eff}} = 0,70$. Трасса неинвазивного анализа τ_3 занимает второе место с оценкой 0,698, а трасса широкого исследования τ_1 получает наименьшую оценку 0,602, поскольку ее эффективность 0,25 есть наименьший критерий в таблице, и относительная компенсация высокой полнотой 0,90 и результативностью 0,85 его не покрывает.

При наборе «оценка под рабочей нагрузкой» лидером становится трасса τ_3 с оценкой 0,758, поскольку вес неинвазивности 0,35 делает ее единичное значение решающим, тогда как трасса τ_2 с оценкой 0,736 смещается на второе место, а трасса

τ_1 остается последней с оценкой 0,601. Смена набора весов меняет лидера, малое возмущение весов внутри набора его не меняет. При весе неинвазивности от 0,32 до 0,38 с пропорциональной перенормировкой остальных весов трасса τ_3 опережает τ_2 не менее чем на 0,016, а при равных весах с тем же возмущением трасса τ_2 опережает τ_3 не менее чем на 0,008. Лидер сменяется при весе неинвазивности около 0,26, между двумя наборами.

Весовые коэффициенты назначаются либо одним набором на все обучение, либо расписанием по итерациям. При одном наборе весов из таблицы 3.2 отбор элит на каждой итерации цикла обучения ведется по одной и той же оценке, и обученная политика отвечает выбранному назначению обследования. Расписание весов, предлагаемое автором, на первых итерациях отдает преобладание весу полноты, и система прежде всего учится обследовать комплекс, затем веса переводятся к целевому набору, и отбор начинает требовать экономности и сохранения работоспособности комплекса. Формально набор весов k -й итерации определяется как (3.30)

$$\begin{aligned} w_{i,k} &= \left(1 - \frac{k}{K_w}\right) w_{i,0} + \frac{k}{K_w} w_{i,K_w}, \quad 0 \leq k \leq K_w; & w_{i,k} \\ &= w_{i,K_w}, \quad k > K_w, \end{aligned} \quad (3.30)$$

где

$w_{i,k}$ — вес i -го критерия на k -й итерации цикла обучения;

K_w — длина расписания, число итераций перехода от стартового набора к целевому;

$w_{i,0}$ — стартовый набор весов, в котором преобладает вес полноты;

w_{i,K_w} — целевой набор весов, один из наборов таблицы 3.2.

Нормировка (3.18) сохраняется на каждой итерации, поскольку сумма весов выпуклой комбинации двух нормированных наборов равна единице по (3.31)

$$\begin{aligned} \sum_{i=1}^5 w_{i,k} &= \left(1 - \frac{k}{K_w}\right) \sum_{i=1}^5 w_{i,0} + \frac{k}{K_w} \sum_{i=1}^5 w_{i,K_w} = 1 - \frac{k}{K_w} + \frac{k}{K_w} \\ &= 1, \end{aligned} \quad (3.31)$$

а неотрицательность весов наследуется от обоих наборов. Оценка (3.17) с весами $w_{i,k}$ остается взвешенным геометрическим средним, и на каждой итерации сохраняются обнуление и относительная компенсация.

В цикл обучения расписание встраивается без изменения его этапов. Оценка завершенной трассы приписывается трассам итерации заново, поэтому расписание меняет только веса, по которым она вычисляется; отбор элитного множества (3.3) по порогу (3.4), извлечение обучающих пар по уровням и шаг дообучения модели не меняются. Стартовый набор и длина расписания K_w дополняют гиперпараметры метода. Безопасность перехода между наборами следует из непрерывности внутреннего оптимума по весам, на компромиссе полноты и эффективности оптимальное число действий N^* движется вместе с весами, и промежуточные наборы расписания не переводят отбор через край компромисса.

Критериальная модель определяет количественные показатели качества функционирования агентской системы и их взвешенное геометрическое среднее, задающее оценку завершенной трассы. Устойчивость (3.27) служит показателем качества обученной модели, а оценка завершенной трассы с набором весов текущей итерации подставляется в условие отбора элит (3.3) вместо оценки (3.2).

Критериальная модель и метод кросс-уровневого обучения вместе образуют алгоритм с определенными входом, выходом и условием остановки. Обучению подлежит единая языковая модель, реализующая политики всех трех уровней иерархии. Ввиду того, что итог каждого сеанса проверяется средой без участия эксперта, обучающим материалом служат собственные завершенные сеансы агентской системы, и внешняя разметка не требуется. На вход алгоритм принимает исходную модель, среду с генераторами экземпляров и целей, наборы весов критериальной модели и гиперпараметры метода, на выходе выдает параметры модели после последней итерации [5; 7]. Псевдокод алгоритма представлен в алгоритме 3.1.

Алгоритм 3.1 – Кросс-уровневое обучение единой модели в среде с верифицируемой обратной связью

	Вход: θ_0 — начальные параметры модели M_0 ; среда $Env = \langle K_{\text{gen}}, G_{\text{gen}}, V \rangle$ (3.1); стартовый набор весов $w_{i,0}$, целевой набор w_{i,K_w} и длина расписания K_w (3.30); гиперпараметры $\beta, D_{\text{all}} , G, T, K, \eta, B$ (Таблица 3.1).
	Выход: параметры θ_K модели после K итераций.
1	для $k = 0, 1, \dots, K - 1$ выполнять
2	$w_{i,k} \leftarrow$ набор весов k -й итерации по (3.30), $i = 1, \dots, 5$
3	для $j = 1, \dots, D_{\text{all}} / G$ выполнять
4	$K_j \sim K_{\text{gen}}; z_j \sim G_{\text{gen}}(\cdot K_j)$
5	выполнить G сеансов агентской системы с моделью M_0 при параметрах θ_k на экземпляре K_j с целью z_j при температуре T
6	$D_{\text{all}} \leftarrow$ множество завершенных трасс всех сеансов итерации
7	для каждой трассы $\tau \in D_{\text{all}}$ выполнять
8	$C_{\text{cov}}(\tau), C_{\text{acc}}(\tau), C_{\text{eff}}(\tau), C_{\text{ninv}}(\tau) \leftarrow$ по (3.11)–(3.14) из трассы τ и истинного состояния экземпляра
9	$C_{\text{res}}(\tau) \leftarrow V(\tau)$ по (3.16), (3.2)
10	$R(\tau) \leftarrow$ оценка завершенной трассы по (3.17) с весами $w_{i,k}$
11	$R_{\text{threshold}} \leftarrow \text{Percentile}(R(D_{\text{all}}), \beta)$ по (3.4)
12	$D_{\text{elite}} \leftarrow \{\tau \in D_{\text{all}} \mid R(\tau) \geq R_{\text{threshold}}\}$ по (3.3)
13	$D_{\text{str}}, D_{\text{tac}}, D_{\text{oper}} \leftarrow$ пары (h_i^t, a_i^t) агентов трех уровней из трасс D_{elite} по (3.5)
14	$D_{\text{train}} \leftarrow D_{\text{str}} \cup D_{\text{tac}} \cup D_{\text{oper}}$ по (3.6)
15	$\theta \leftarrow \theta_k$
16	повторять до сходимости $L(\theta)$ по (3.7): извлечь из D_{train} мини-пакет размера B ; $\theta \leftarrow \theta - \eta \cdot \nabla_{\theta} L_B(\theta)$ по (3.8)
17	$\theta_{k+1} \leftarrow \theta$
18	вернуть θ_K

Из метода кросс-уровневого обучения в алгоритм входят порождение трасс (строки 3–6), отбор элит (строки 11–12), извлечение обучающих пар (строки 13–14) и дообучение единой модели (строки 15–17). Из критериальной модели входят расписание весов (строка 2) и вычисление критериев с оценкой завершенной трассы (строки 8–10). Постоянный набор весов является частным случаем расписания при совпадающих стартовом и целевом наборах, при $w_{i,0} = w_{i,K_w}$ формула (3.30) дает один и тот же набор на каждой итерации.

Обученная модель оценивается устойчивостью (3.27) на контрольном множестве экземпляров. Под контрольным множеством понимается множество экземпляров, порожденных генератором K_{gen} и не использованных при порождении трасс ни на одной итерации; число контрольных сеансов N_{trials} задается по (3.28). Значения устойчивости для моделей с параметрами $\theta_1, \dots, \theta_K$, вычисленные на

одном контрольном множестве, образуют зависимость качества от номера итерации, по которой задается число итераций K .

Автором разработана схема алгоритма, отражающая путь данных от экземпляров комплекса к параметрам единой модели. Схема представлена на рисунке 3.7.

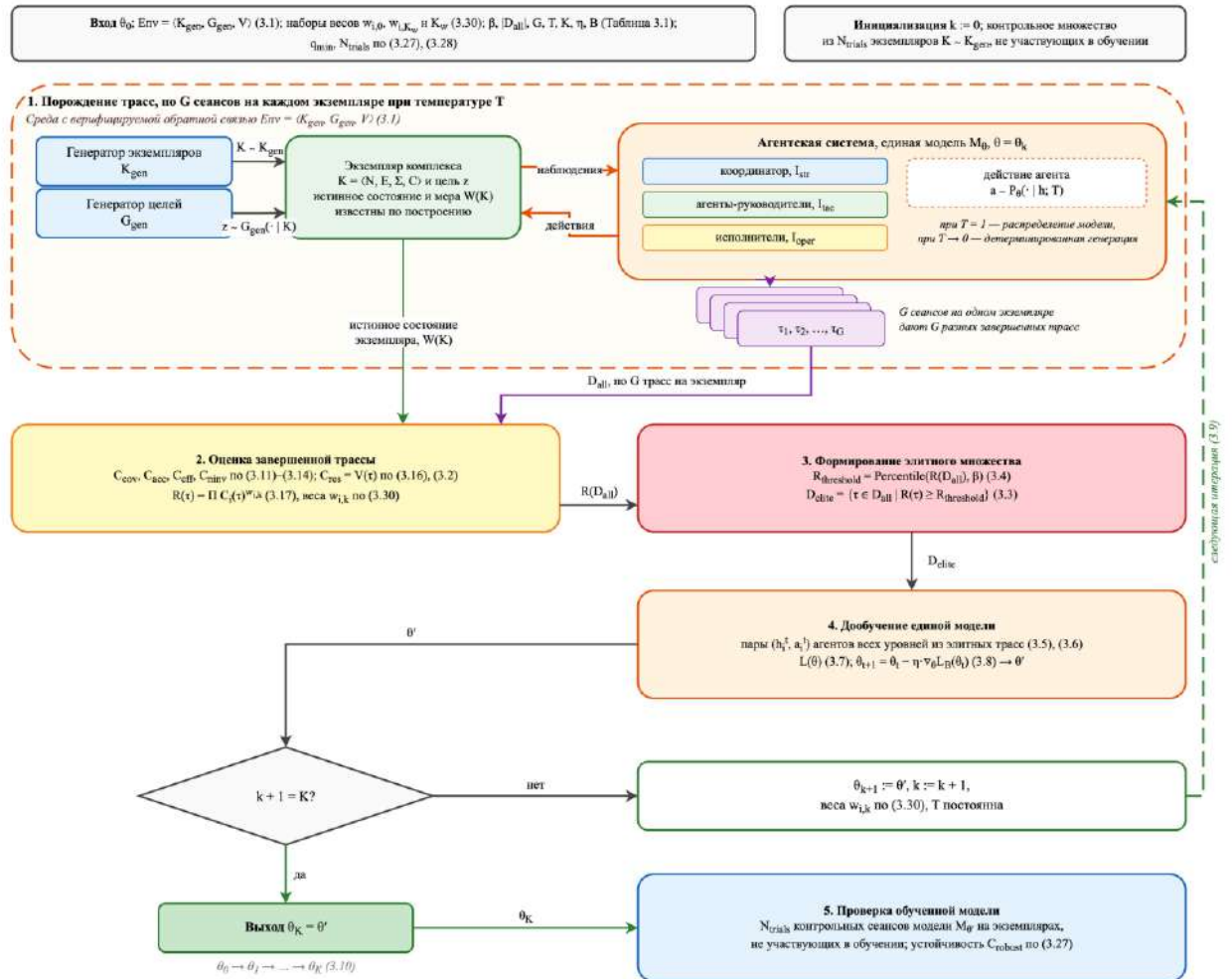


Рисунок 3.7 – Алгоритм кросс-уровневого обучения

Алгоритм замыкает контур самообучения агентской системы. Среда проверяет итог каждого сеанса без участия эксперта, критериальная модель преобразует проверенный итог и ход сеанса в единый целевой сигнал, а отбор элит с дообучением единой модели переносит успешный опыт сеансов в политики всех уровней иерархии.

3.3 Экспериментальные исследования в испытательной среде с верифицируемой обратной связью

Программная реализация выполнена в виде комплекса из трех взаимодействующих модулей, каждый из которых предназначен для самостоятельной задачи и заменяется альтернативной реализацией без изменения остальных. Модулем агентской системы оценка работоспособности комплекса строится трехуровневой иерархией агентов. Модулем испытательной среды с верифицируемой обратной связью порождаются экземпляры распределенного программно-аппаратного комплекса и автоматически проверяется полученная оценка. Модулем кросс-уровневого обучения выполняется дообучение базовой модели, единой для всех уровней иерархии, на отобранных трассах. Взаимодействие модулей ведется через фиксированный программный интерфейс, потоки данных однонаправлены, чем обеспечивается независимое развитие и замена каждого модуля. Архитектура комплекса приведена на рисунке 3.8.

Ядром комплекса служит модуль агентской системы. Уровень агента определяется его положением в иерархии. Корневой агент создается как стратегический и играет роль координатора; порожденные им агенты относятся к тактическому уровню, порожденные тактическими — к операционному. Стратегический уровень включает одного координатора, численность тактического и операционного уровней определяется в ходе обследования. Работа каждого агента организована циклически. На каждом шаге по накопленной истории формируется запрос к большой языковой модели, в ответ возвращается вызов инструмента, инструмент исполняется, результат добавляется в историю. Цикл повторяется до вызова инструмента завершения или исчерпания бюджета действий. Координатору задается глобальная цель обследования, задачи распределяются между тактическими агентами, решение о завершении принимается на стратегическом уровне. На тактическом уровне полученная задача

декомпозируется и порождаются операционные агенты. На операционном уровне комплекс непосредственно обследуется средствами сбора сведений об узлах, соединениях и службах, измерения контролируемых показателей, чтения конфигурационных данных и опроса служб. Все средства исполняются в изолированном окружении, чем обеспечивается воспроизводимость и безопасность экспериментов. По завершении обследования оценка меры работоспособности вычисляется подстановкой измеренных значений показателей.

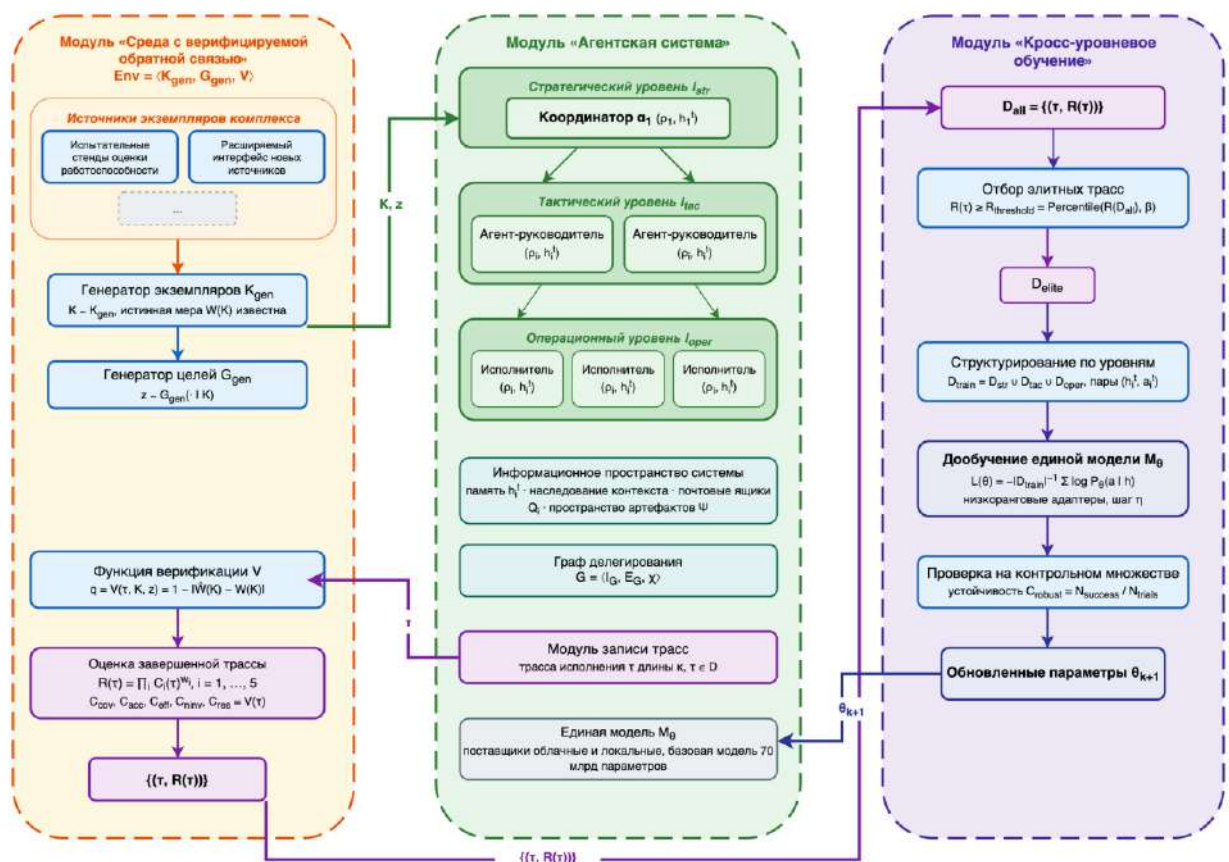


Рисунок 3.8 – Архитектура программного комплекса

В комплексе поддерживается подключение различных поставщиков больших языковых моделей через единый интерфейс. Облачные модели подключаются по адресу сервера и ключу доступа, локальные — по адресу сервера вывода. При переключении между поставщиками изменение кода агентов не требуется, чем допускается подстановка в неизменную архитектуру различных базовых

моделей. Базовой моделью в проведенных экспериментах служит готовая открытая большая языковая модель Qwen2.5-72B-Instruct с 72 млрд параметров. Исходные веса базовой модели при дообучении не изменяются, обучаются низкоранговые адаптеры, а одна и та же модель с адаптером обслуживает все три уровня иерархии.

Модулем испытательной среды с верифицируемой обратной связью порождаются экземпляры комплекса с заданными составом, состоянием и значениями конфигурационных параметров, поэтому истинная мера работоспособности каждого экземпляра известна по построению. Агентской системе предоставляется доступ к экземпляру, а по завершении обследования построенная оценка сопоставляется с истинным значением меры, чем без участия эксперта вычисляется степень достижения цели. Архитектурой среды допускается подключение различных источников экземпляров через фиксированный интерфейс, чем обеспечивается расширение состава испытательных стендов без изменения остальных модулей.

Модулем кросс-уровневого обучения принимаются оцененные трассы, отбирается элитное подмножество по адаптивному пороговому значению, формируются обучающие пары с разделением по уровням иерархии и выполняется дообучение адаптера базовой модели. Порождение трасс на каждой итерации ведется с управляемой температурой, повышаемой на ранних итерациях для разнообразия трасс и понижаемой на поздних для устойчивого воспроизведения освоенного поведения. Обновленная модель проверяется на контрольном множестве экземпляров, после чего цикл повторяется.

Разработанные модели и алгоритм реализованы в программном комплексе, на который получено свидетельство о государственной регистрации программы для электронных вычислительных машин. В реализованном комплексе воплощены математическая модель оценки работоспособности, параметрическая модель архитектуры многоагентной интеллектуальной системы с трехуровневой иерархией агентов и алгоритм кросс-уровневого обучения в среде с верифицируемой обратной связью; комплекс служит инструментом их экспериментальной проверки.

Экспериментальная проверка предложенных моделей и алгоритма требует среды, в которой истинная работоспособность исследуемого комплекса известна заранее, а результат каждого сеанса обследования устанавливается без участия эксперта. Такой средой служит испытательная среда с верифицируемой обратной связью. Средой порождаются экземпляры распределенного программно-аппаратного комплекса с заданными составом, состоянием и значениями конфигурационных параметров, поэтому истинное состояние каждого экземпляра известно по построению. По известному состоянию непосредственно вычисляется истинная мера работоспособности $W(K)$ как взвешенное геометрическое среднее степеней соответствия контролируемых показателей комплекса их требованиям. Агентской системой экземпляр обследуется, устанавливается состав его элементов, измеряются контролируемые показатели и строится оценка $\hat{W}(K)$ той же сверткой по измеренным значениям. Качество обследования определяется близостью оценки к истинному значению меры.

Готового общедоступного набора задач под оценку работоспособности распределенных программно-аппаратных комплексов не существует. Существующие наборы ориентированы на выявление отдельных программных дефектов и не содержат ни градуированной меры работоспособности, ни известного по построению истинного состояния комплекса, без которого автоматическая верификация оценки невозможна. Поэтому автором построена собственная испытательная среда, воспроизводящая типовые распределенные комплексы, и все экспериментальные сравнения проводятся на ней. На единой среде с известной мерой работоспособности сопоставляются конфигурации разработанной системы и готовые большие языковые модели, чем обеспечивается сопоставимость и воспроизводимость результатов.

Организации, предоставившие вычислительные мощности, выделяли несколько виртуальных машин, на которых допускалось развертывание произвольных конфигураций исследуемых комплексов, а наращивание объема оперативной памяти, дискового пространства и числа процессорных ядер производилось по предъявлению первых результатов. Такой порядок определил

посильный масштаб среды, которую один исследователь разворачивает без промышленной инфраструктуры. Число узлов удержано небольшим, а типовые конфигурации, среди них служба каталога и сетевые сервисы, разворачиваются автоматически.

Каждый из трех стендов задан составом узлов, сетевой топологией, набором служб и числовыми значениями аппаратных характеристик, по которым вычисляются контролируемые показатели меры работоспособности. Характеристики стендов приведены в таблице 3.4.

Таблица 3.4 – Характеристики испытательных стендов

Характеристика	Малый	Средний	Корпоративный
Число вычислительных узлов	6	12	18
Число сетевых сегментов	1	2	4
Операционные системы	Linux	Linux, Windows	Linux, Windows, специализированные
Число программных служб	9	20	40
Процессорных ядер на узел	2	4	от 4 до 8
Оперативная память узла, ГБ	4	8	от 8 до 16
Дисковая память узла, ГБ	50	100	от 100 до 250
Пропускная способность соединений, Гбит/с	1	1	от 1 до 10
Директивное время обработки заявки, с	2	2	1
Средства обнаружения и защиты	нет	системы обнаружения вторжений	управление событиями безопасности, защита конечных точек

Стенд малого масштаба содержит шесть узлов в одном сетевом сегменте под управлением операционных систем семейства Linux с веб-сервером, сервером базы данных, файловым хранилищем и службой удаленного доступа и воспроизводит инфраструктуру малого предприятия. Стенд среднего масштаба содержит двенадцать узлов в двух сегментах с контролируемой маршрутизацией под управлением операционных систем семейств Linux и Windows, состав служб расширен серверами каталога, почтовыми серверами и системами управления

содержимым, добавлены средства обнаружения вторжений. Стенд корпоративного масштаба содержит восемнадцать узлов в четырех сегментах с многоуровневой сегментацией, службой каталога предприятия, распределенными базами данных с репликацией и средствами защиты. В каждом комплексе поступающий поток заявок обрабатывается прикладными службами, а работоспособность характеризуется способностью обслуживать этот поток с показателями, соответствующими требованиям, чем задается содержательный смысл интенсивности обработки заявок и доли задач, решаемых за директивное время. Единый состав контролируемых показателей делает меру работоспособности сопоставимой между стендами.

Вычислительные ресурсы экспериментов разделены между стендами, сервером модели и дообучением. Стенды развернуты на сервере виртуализации с 32 процессорными ядрами, 256 ГБ оперативной памяти и 4 ТБ на твердотельных накопителях, одновременно разворачивается один экземпляр комплекса, ресурсы его узлов заданы таблицей 3.4. Базовая модель со сжатием весов до четырех бит выполняется на сервере с двумя графическими ускорителями по 24 ГБ памяти, 16 процессорными ядрами, 128 ГБ оперативной памяти и 2 ТБ на твердотельном накопителе при окне контекста 8 тыс. единиц текста модели (токенов), двух параллельных сеансах и скорости около 19 единиц в секунду на сеанс; агентская система работает на виртуальной машине с 8 ядрами, 32 ГБ оперативной памяти и 300 ГБ дисковой памяти.

Сеанс обследования одного экземпляра длится от 6 до 20 минут в зависимости от стенда, по медиане 11 минут, и содержит около 30 вызовов модели. Дообучение адаптеров выполняется в облачном сервисе на одном ускорителе с 80 ГБ памяти, одна итерация занимает около трех часов, тридцать итераций пяти прогонов заняли около 450 часов машинного времени; сервер модели, версии инструментов и стенды за время экспериментов не менялись.

Оценка ведется в контуре объекта и в контуре инструмента. Контур объекта принадлежит комплексу, мера работоспособности $W(K)$ характеризует его состояние и существует независимо от способа обследования, а качество

обследования оценивается близостью оценки $\hat{W}(K)$ к истинному значению меры и устойчивостью, равной доле экземпляров, на которых близость не ниже порогового значения. Контур инструмента принадлежит агентской системе, качество ее функционирования характеризуется оценкой завершенной трассы $R(\tau)$, которая служит целевым сигналом дообучения. Прямое сравнение конфигураций по оценке завершенной трассы некорректно, ведь дообученная конфигурация превосходит прочие по построению, будучи оптимизированной именно на этой величине. Итоговое сравнение проводится по контуру объекта на контрольном множестве экземпляров, не участвовавших в дообучении, что отделяет проверяемое свойство системы от оптимизируемой при обучении величины и устраняет замкнутость оценки.

Сопоставлением двух конфигураций разработанной системы, трехуровневой иерархии на базовой модели без дообучения и той же иерархии на той же модели после дообучения, выделяется вклад алгоритма при неизменной архитектуре. Внешнее сопоставление выполняется подстановкой в неизменную архитектуру готовых больших языковых моделей облегченного класса Claude Haiku и Gemini Flash без дообучения, сопоставимых с базовой по числу параметров. Базовая модель до дообучения сама принадлежит к готовым моделям, поэтому ее исходное качество служит одной из точек диапазона готовых моделей, а вклад алгоритма измеряется превышением над всем этим диапазоном после дообучения; достигнутое качество относится к предложенному алгоритму и не сводится к исходным возможностям базовой модели. Все сравнения ведутся на одной испытательной среде и оцениваются по контуру объекта на контрольных экземплярах.

Пороговое значение близости, при котором экземпляр засчитывается в устойчивость, принято равным 0,85. Дообучение выполнено пятью независимыми прогонами по тридцать итераций при одной исходной модели и одном контрольном множестве, прогоны различаются порождением трасс. Динамика средней близости оценки и устойчивости по итерациям приведена на рисунке 3.9, каждая тонкая линия отвечает одному прогону, утолщенная линия

— среднему по прогонам. На исходной модели средняя близость составляет 0,83, устойчивость 0,57. Ход отдельного прогона не монотонен. Устойчивость прогона между соседними итерациями изменяется на величину до 0,14, снижение наблюдается почти на половине переходов, у каждого прогона есть участки снижения длиной в две-три итерации, на ранних итерациях прогоны опускаются ниже исходного значения, до 0,51, а отдельные значения достигают 0,76. Колебания порождаются случайным выбором действий при температуре генерации, ограниченным объемом элитного множества одной итерации и конечным числом экземпляров контрольного множества. Направленный рост проявляется в среднем по прогонам, причем и среднее снижается на 13 переходах из 30, поэтому достигнутый уровень оценивается средним за последние десять итераций. Средние по прогонам за итерации с первой по десятую, с одиннадцатой по двадцатую и с двадцать первой по тридцатую составляют 0,84, 0,86 и 0,87 по близости и 0,60, 0,63 и 0,65 по устойчивости; за последние десять итераций средние отдельных прогонов лежат в пределах от 0,86 до 0,87 по близости и от 0,60 до 0,69 по устойчивости.

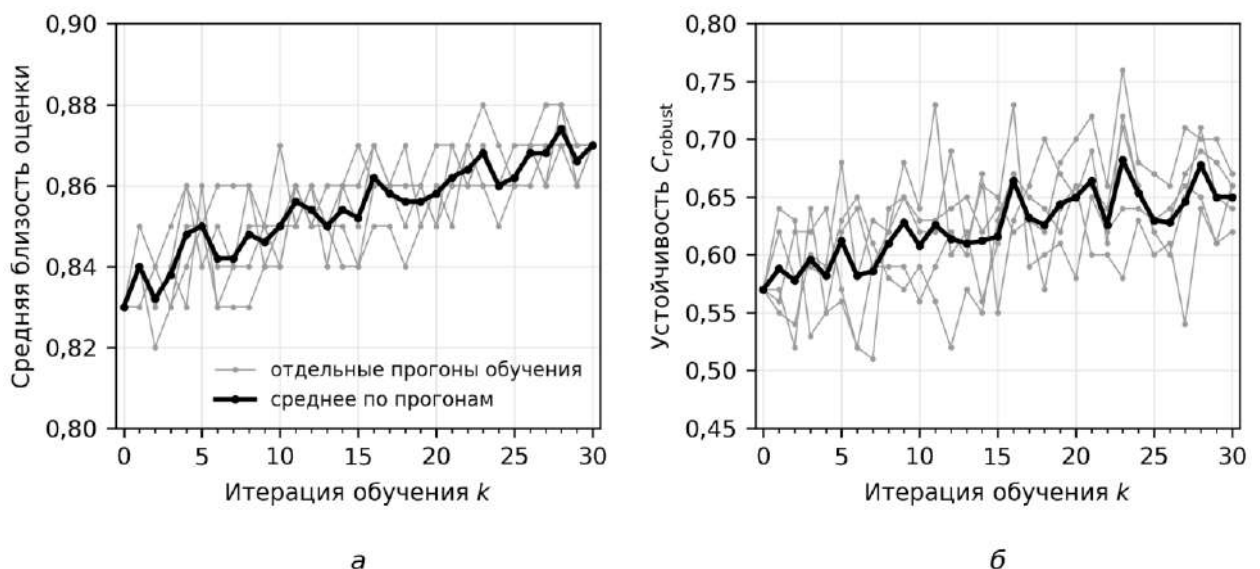


Рисунок 3.9 – Качество оценки работоспособности по итерациям обучения: а – средняя близость оценки; б – устойчивость

Вклад предложенного алгоритма выделяется подстановкой в неизменную архитектуру готовых больших языковых моделей без дообучения, сопоставимых с базовой по числу параметров. Результаты приведены в таблице 3.5 и на рисунке 3.10.

Таблица 3.5 – Сравнение базовых моделей в неизменной архитектуре

Базовая модель	Устойчивость	Средняя близость оценки
Qwen2.5-72B-Instruct после кросс-уровневого обучения	0,65	0,87
Qwen2.5-72B-Instruct без дообучения	0,57	0,83
Claude Haiku без дообучения	0,60	0,84
Gemini Flash без дообучения	0,58	0,83

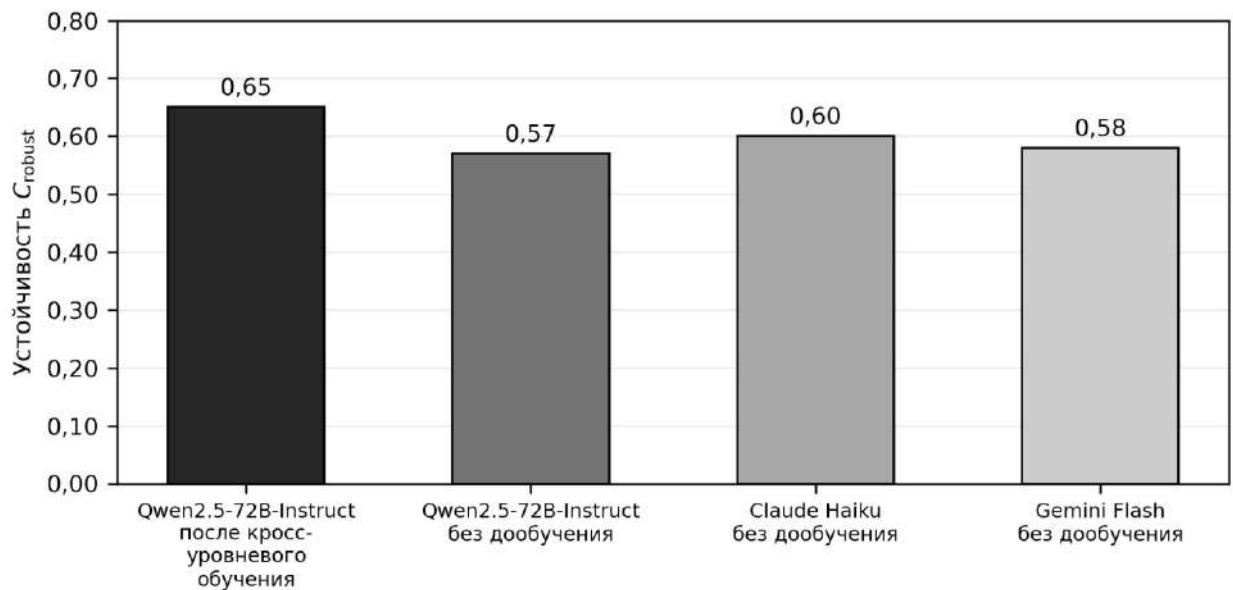


Рисунок 3.10 – Сравнение базовых моделей в неизменной архитектуре по устойчивости

У базовой модели без дообучения устойчивость составляет 0,57, у готовых моделей облегченного класса без дообучения 0,58 и 0,60. У той же базовой модели после кросс-уровневого обучения устойчивость составляет 0,65, что выше любой

из готовых моделей без дообучения. Разность устойчивости 0,08 между моделью после обучения и той же моделью до обучения получена при неизменной архитектуре и одной базовой модели, поэтому относится к алгоритму кросс-уровневого обучения; превышение 0,05 над лучшей из готовых моделей по устойчивости и снижение ошибки оценки относительно нее с 0,16 до 0,13, в 1,2 раза, показывают, что достигнутое качество не сводится к выбору базовой модели.

3.4 Опытная эксплуатация на действующих распределенных программно-аппаратных комплексах

Опытная эксплуатация системы проведена на трех действующих распределенных программно-аппаратных комплексах, обследование которых входит в производственный процесс сопровождающих их организаций. Истинная мера работоспособности действующего комплекса неизвестна, поэтому близость оценки и устойчивость в эксплуатации не вычисляются, проверяется применимость результатов работы. Организации обозначены площадками А, Б и В, акты о внедрении приведены в приложении. Характеристики площадок сведены автором в таблице 3.6.

Таблица 3.6 – Площадки опытной эксплуатации

Признак	Площадка А	Площадка Б	Площадка В
Комплекс	Сеть автоматизированных парковочных объектов, 44 объекта по 6–10 узлов и ядро из 8 узлов в центре обработки данных	Сеть оператора связи, около 300 участков и около 3 тыс. управляемых устройств	34 информационные системы заказчиков по 8–30 узлов
Единица обследования	Парковочный объект со службами ядра	Участок сети, домовый узел с агрегацией и затронутыми службами, 6–18 узлов	Информационная система заказчика
Единица работы	Зарегистрированное нарушение	Наряд второй линии на сетевую диагностику	Плановое обследование по

Признак	Площадка А	Площадка Б	Площадка В
Период	работоспособности объекта 6 недель до включения заклучений, 22 недели после	6 недель до включения заклучений, 27 недель после	договору сопровождения Пилотный квартал, основной режим два квартала
Базовая модель и обучение	Qwen2.5-72B-Instruct без дообучения, сервер вывода облачного поставщика	DeepSeek-R1-Distill-Llama-70B на сервере организации, обучение на стендовой копии в периметре организации, 20 итераций	Qwen2.5-72B-Instruct без дообучения, сервер вывода организации
Улучшенный показатель	Время от регистрации нарушения до установления источника несоответствия	Медиана времени закрытия наряда, нарядов на инженера в вечернюю смену	Доля систем, обследованных за квартал, при известных точности и полноте находок

Под единицей обследования понимается комплекс от 6 до 30 узлов, обследуемый за один сеанс или за обход из нескольких сеансов. Заключение содержит оценку меры работоспособности единицы, показатель с наименьшей степенью соответствия, элемент, на котором он измерен, и подтверждающие результаты инструментов. Опорным исходом служит источник несоответствия, установленный персоналом при закрытии единицы работы, заключение считается совпавшим с исходом при совпадении показателя и элемента.

На площадках А и Б первые шесть недель заключения записывались с отметкой времени, но персоналу не показывались, и на этих неделях измерена правильность заключений: скрытое заключение сравнивалось с источником несоответствия, который персонал установил без подсказки. После включения заключения открыты персоналу, показатели площадок сравниваются до включения и после включения, решение персонала по каждому заключению записывается в введенное на период поле со значениями «принято», «принято с изменениями» и «отклонено». Повторный сигнал по той же единице в контрольном окне означает, что несоответствие не устранено, и дает верхнюю границу ошибок системы.

Площадка А объединяет 44 парковочных объекта, каждый с объектным шлюзом, камерами распознавания номеров, контроллерами шлагбаумов, стойками

оплаты и коммутатором, с ядром объект связан основным и резервным каналами. Состав контролируемых показателей задан процессом парковки и лишь частью совпадает с составом показателей меры работоспособности распределенного программно-аппаратного комплекса. Из него взяты доля доступных служб распознавания, управления шлагбаумом, оплаты и фискализации (x_{10}), время отклика интерфейса ядра (x_3), потери и пропускная способность канала объекта (x_9 , x_2). Показатели самого процесса, время открытия шлагбаума от фиксации номера до команды контроллеру с требованием не более 5 с, доля распознанных номеров, доля сессий оплаты, закрытых в ядре к выезду, и задержка доставки списков допуска на объект, введены с требованиями регламента оператора. Степень соответствия требованию и взвешенное геометрическое среднее вычисляются без изменений. Нарушение регистрируется диспетчером, когда стандартное действие не устранило вызов со стойки или поступил сигнал мониторинга, около 6 нарушений в сутки и 1 170 за период. Источник установлен внутри комплекса в 52 % случаев, остальные источники лежат в механике привода, нечитаемом номере или поврежденной петле и требуют выезда при любом заключении.

Обследование запускается при регистрации нарушения, между стандартным действием диспетчера и решением о выезде. Координатор выделяет направления канала объекта, цепочки проезда от камеры к шлагбауму, оплаты и служб ядра. Исполнители читают журналы шлюза, контроллера и распознавания, измеряют потери и переход на резервный канал, сверяют списки допуска с эталоном. Пробы ограничены синтетическим запросом к ядру и эталонным снимком для службы распознавания, команд шлагбаумам система не подает.

Улучшенным показателем служит время от регистрации нарушения до установления источника. Автором построен ряд недельных медиан этого времени до включения заключений и после включения. Ряд представлен на рисунке 3.11.

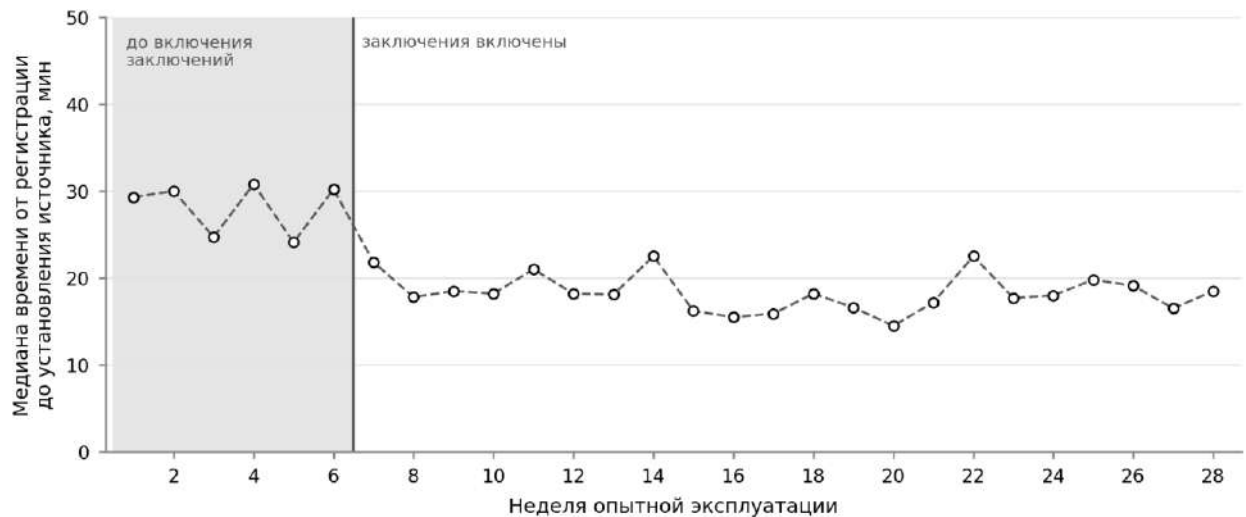


Рисунок 3.11 – Медиана времени от регистрации нарушения до установления источника несоответствия по неделям на площадке А

Затемненная полоса отвечает шести неделям до включения заключений, вертикальная линия — включению на седьмой неделе. Ступенька на седьмой неделе отделяет время до включения от времени после включения, далее недельные медианы колеблются вокруг нового уровня.

До включения заключений медиана времени до установления источника составляла днем 24 мин, ночью 55 мин, после включения 16 и 17 мин. Ночное сокращение больше дневного, поскольку ночью в смене один диспетчер, до получения заключения он открывает шлагбаум удаленно, а выезд назначается на следующий день. За шесть недель до включения скрытое заключение совпало с источником, установленным диспетчером, в 185 случаях из 250, то есть в 74 %. Исходы после включения заключений на 920 нарушениях сведены автором в таблице 3.7.

Таблица 3.7 – Исходы после включения заключений на площадке А

Показатель	Значение
Заключение принято, принято с изменениями, отклонено	653, 147 и 120, принято 71 %
Сеансов без заключения, объект без связи, оборудование без доступа, истечение времени ожидания	7 %

Показатель	Значение
Нарушений, устраненных без выезда, до и после	34 и 47 % при 52 % источников внутри комплекса
Повторный сигнал по объекту в течение 72 часов, до и после	9 и 8 %
Отклоненное заключение задержало выезд	5 случаев, медиана задержки 40 мин
Оценка меры объекта до и после устранения, медиана	0,62 и 0,97
Время сеанса, медиана и 90-й перцентиль	16 и 31 мин

Доля устраненных без выезда ограничена сверху долей источников внутри комплекса. Рост оценки меры после устранения подтверждает, что показатель с наименьшей степенью соответствия в заключении отвечает устраненному несоответствию. Базовая модель Qwen2.5-72B-Instruct без дообучения вызывается на сервере вывода облачного поставщика, адаптер испытательных стендов на нем не размещается. Агентская система работает на виртуальной машине в том же центре обработки данных, что и ядро, версия модели и инструменты за период эксплуатации не менялись.

Площадка Б включает около 300 участков сети оператора связи. Участок объединяет домовый узел доступа с источником бесперебойного питания, узел агрегации и затронутые службы адресации, аутентификации и учета подключений абонентов, разрешения имен, телевизионного вещания и расчетов с абонентами. Ведущими показателями служат доля потерь пакетов с требованием не более 0,1 % для групп вещания (x_9), доля доступных служб (x_{10}), пропускная способность к абоненту (x_2), доля аутентификаций и выдач адресов за директивное время (x_{12}) и время отклика личного кабинета (x_3). Наряд второй линии возникает, когда порт и подключение в норме, а абонент наблюдает потери или замирание вещания, либо когда мониторинг указывает на нарушение выше домового коммутатора, от 610 до 740 нарядов в месяц, вечерний пик дает 40 % суточных. Наряд закрывается кодом источника и действием, измеряется время от создания до закрытия или передачи бригаде.

Обследование запускается при создании наряда, до того как инженер берет его в работу, поэтому в вечерний пик заключения готовы раньше, чем инженер

приступает к нарядам, в этом состоит механизм выигрыша по пропускной способности смены. Наряды, закрытые инженером до готовности заключения, из сравнения исключены. Сведения о сети не могут передаваться третьим лицам, поэтому модель размещена на сервере организации с одним ускорителем на 80 ГБ. По выбору организации развернута открытая модель DeepSeek-R1-Distill-Llama-70B в четырехбитном квантовании.

На испытательных стендах эта модель не проверялась, а адаптер испытательных стендов к ней неприменим по построению. В шесть недель до включения заключений модель без дообучения соблюдала протокол взаимодействия агентов неустойчиво, 50 % сеансов завершались без заключения. Инженерные меры, образцы вызовов в инструкциях агентов, ограничение числа инструментов у исполнителя, принудительная доставка отчета агенту-руководителю и завершение агента после трех ответов без вызова инструмента, снизили долю до 24 %, остаток снят итерациями обучения.

Средой обучения служит стендовая копия участка сети в периметре организации: восемь виртуальных машин со службами участка и обезличенной копией системы расчетов, три коммутатора доступа и один коммутатор агрегации из запасного фонда, генератор подключений абонентов. Материалом для обучения служит корпус из 700 примеров несоответствий, взятых из закрытых нарядов второй линии и воспроизводимых на копии: несоответствие вносится контроллером стенда и записывается в журнал, поэтому истинное состояние каждого экземпляра известно и алгоритм кросс-уровневого обучения применяется без изменений. Примеры охватывают шесть типов несоответствий: питание, узел доступа и порт, канал к агрегации, адресация и аутентификация, вещание, подключение абонента; вещание и подключения абонентов на копии имитируются. Журналы сеансов эксплуатации в среду обучения не передаются.

На модели без дообучения большая часть сеансов заканчивалась без заключения, и пригодных для обучения трасс набиралось мало, поэтому обучение построено на объеме. На каждой итерации из корпуса случайно выбираются 70 примеров, на каждом система проводит по 10 сеансов, каждой

трассе по журналу стенда приписывается оценка завершенной трассы, в обучение идет лучшая четверть трасс, на первых итерациях это трассы лишь с частичным результатом. Адаптер каждой итерации дообучается на этих трассах, продолжая адаптер предыдущей, и проверяется на небольшом контрольном множестве из 72 экземпляров, по которому видно, улучшается ли модель от итерации к итерации. Выполнено 20 итераций, после чего модель с адаптером введена в эксплуатацию, до этого заключения строила модель без дообучения с инженерными мерами. Автором построен ряд медиан времени закрытия наряда по неделям. Ряд представлен на рисунке 3.12.

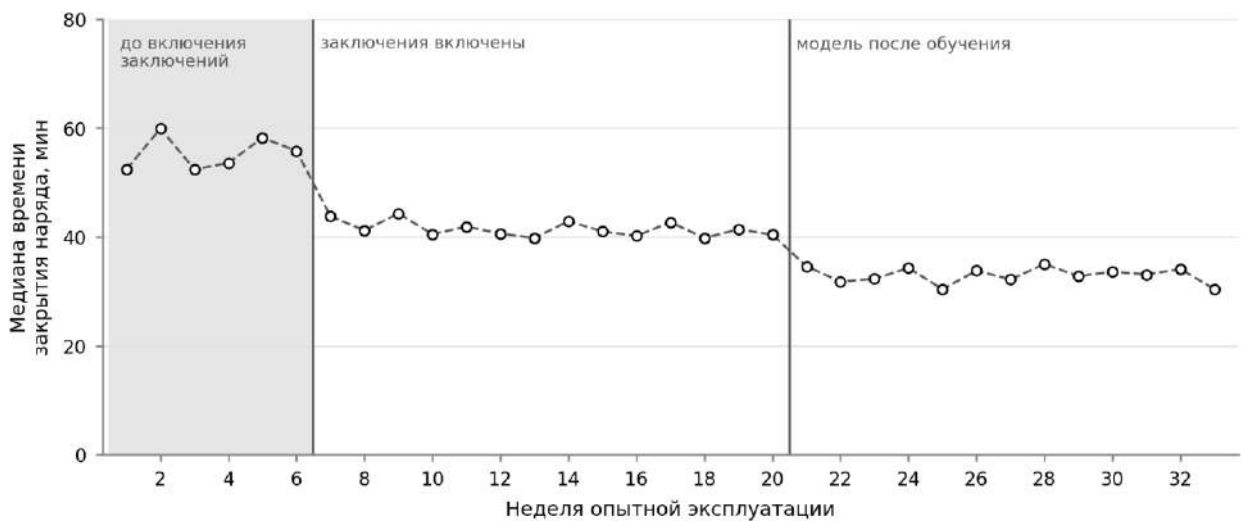


Рисунок 3.12 – Медиана времени закрытия наряда по неделям на площадке Б

Затемненная полоса отвечает шести неделям до включения заключений, первая вертикальная линия — включению на седьмой неделе, вторая — вводу модели после обучения на двадцать первой. Показатели до включения заключений и последних 12 недель, с моделью после обучения, сведены автором в таблице 3.9.

Таблица 3.9 – Показатели площадки Б до включения заключений и в последние 12 недель с моделью после обучения

Показатель	До	После	Примечание
Время закрытия наряда, медиана, мин	54	33	–
Время закрытия наряда, 90-й перцентиль, мин	230	140	хвост образуют наряды с передачей бригаде
Нарядов на инженера в вечернюю смену	4,1	5,4	по графику смен
Повторные обращения в течение 7 суток, %	6,2	5,1	–
Оценка меры участка до и после закрытия наряда, медиана	0,71	0,98	по показателям при обследовании и при повторной проверке
Сеансов без заключения, %	50	9	базовая модель и после 20 итераций
Заключение принято среди сеансов с заключением, %	41	77	–

После обучения доля верно установленных источников нарушения работоспособности на площадке Б выросла с 60 до 75 %. На контрольном множестве расхождение оценки с истинным значением снижалось по итерациям для всех шести типов. Единственный случай влияния проб, проба потерь на канале к агрегации в час пик, поднял загрузку канала на 3 %, после чего пробы ограничены ночными часами.

Площадка В объединяет 34 информационные системы заказчиков из серверов приложений и баз данных, рабочих станций ключевых ролей, средств защиты информации и коммутаторов, от 8 до 30 узлов и от 15 до 40 служб под управлением Linux и Windows. Договор сопровождения предусматривает ежеквартальное обследование со сверкой состава узлов и служб с учетом (x_1), проверкой конфигурационных параметров, резервного копирования, сертификатов и учетных записей, измерением ресурсов узлов (x_4 – x_8), доступности и отклика служб (x_{10} , x_3). Под находкой понимается установленное при обследовании несоответствие требованиям договора, перечень находок образует заключение вместе с оценкой меры. Вручную обследование занимает от 40 до 120 часов на систему при медиане 62, поэтому ежеквартально обследовались восемь или девять наиболее ответственных систем, остальные реже раза в год.

Обследование выполняется ночью с переносного сервера, подключенного к служебному сегменту по согласованию с администратором заказчика. Детерминированные сборщики для Linux, Windows и средств защиты передают модели сжатые таблицы, обход занимает от 6 до 12 сеансов при двух одновременных, всего три или четыре часа. Базовая модель Qwen2.5-72B-Instruct без дообучения размещена на сервере организации, система подключена к его программному интерфейсу. Переносной сервер несет агентскую систему и сборщики. После пилотного квартала введено сжатие таблиц сборщиков, перечень находок вынесен в начало заключения, оба изменения учтены при сравнении кварталов.

Улучшенным показателем служит доля систем, обследованных за квартал, при известных точности и полноте находок. Автором построен ряд обследований по месяцам опытной эксплуатации. Ряд представлен на рисунке 3.13.

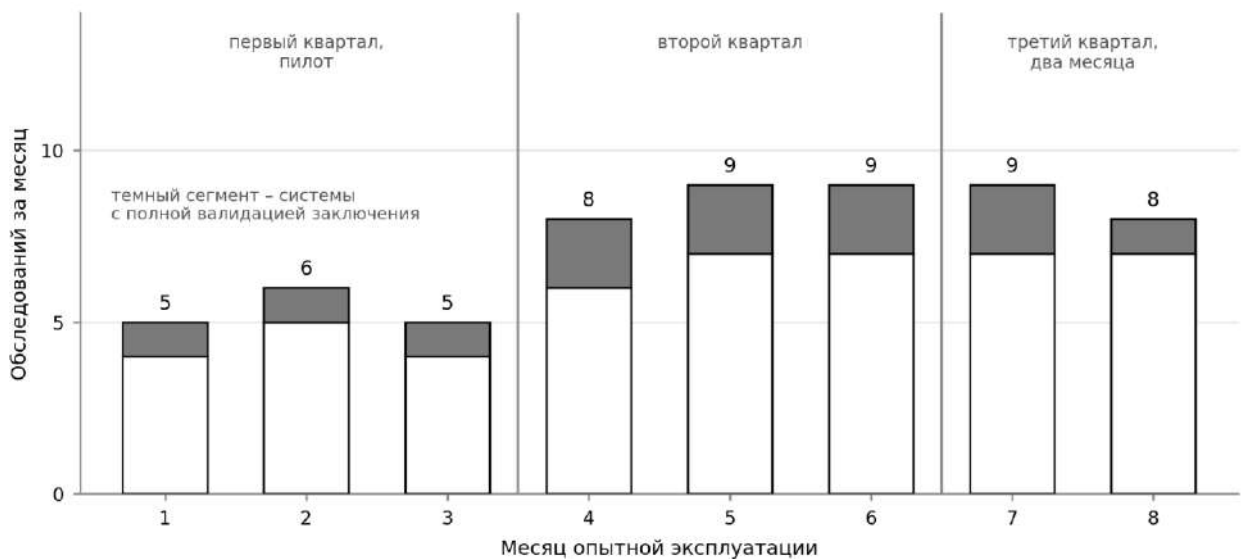


Рисунок 3.13 – Обследования информационных систем заказчиков по месяцам опытной эксплуатации на площадке В

Темный сегмент столбца отвечает системам с полной ручной валидацией заключения, третий квартал показан за два месяца, со второго квартала число

обследований держится на уровне восьми-девяти в месяц. Показатели площадки В сведены автором в таблице 3.10.

Таблица 3.10 – Показатели площадки В до и в период опытной эксплуатации

Показатель	До	После	Примечание
Обследований в квартал	8–9	26	пилотный квартал 16, два месяца третьего квартала 17
Часы инженера на систему, медиана	62	14	с дорогой, установкой сервера и проверкой находок, при полной валидации 48
Часы инженеров за квартал	около 530	около 560	покрытие в 3 раза при тех же часах
Давность последнего обследования, медиана, дней	210	75	до опытной эксплуатации и на конец периода
Сеансов без заключения, %	–	около 14	сжатие контекста, недоступные сегменты

Качество находок измерено на 12 системах с полной ручной валидацией. Из 55 находок системы инженером подтверждены 38, точность около 69 %, из 69 находок инженера система назвала 38, полнота около 55 %. Пропущено около 5 % элементов состава.

Единица обследования площадок А и Б обследуется вручную за время от 8 до 24 часов, поэтому базовая трудоемкость полного обследования от сорока до двухсот часов относится только к площадке В, где медиана ручного обследования составляет 62 часа.

Ограничения достоверности заданы устройством сравнений. Сравнения до и после разделены во времени и не имеют контрольной группы, влияние состава случаев, сезона и организационных мер снижено рядами по неделям, но не устранено. Доля принятых заключений дает нижнюю границу ошибок системы, повторные сигналы и обращения верхнюю. Близость оценки и устойчивость в эксплуатации не вычисляются, единственным местом лабораторной метрики служит контрольное множество стендовой копии площадки Б. Результаты площадок А и В получены на готовой модели без дообучения и вклад алгоритма кросс-уровневого обучения не характеризуют. Область

применимости выводов ограничена комплексами от 6 до 30 узлов с перечисленными типами служб.

Результаты опытной эксплуатации подтверждают применимость разработанных моделей и алгоритма к действующим комплексам. Заключение по мере работоспособности принято персоналом без изменений в 71 % случаев на площадке А и в 77 % на площадке Б, оценка меры после устранения несоответствия возрастает с 0,62 до 0,97 и с 0,71 до 0,98. Трехуровневая архитектура обследует комплексы до 30 узлов при машинном времени обхода от трех до четырех часов. Алгоритм кросс-уровневого обучения на стендовой копии в периметре организации снижает долю сеансов без заключения у базовой модели, не проверявшейся на испытательных стендах, с 50 до 9 % за 20 итераций. На площадках А и В система работает на готовой модели без дообучения. На площадке А состав контролируемых показателей задан процессом парковки, конструкция меры и система применены к нему без изменений, поэтому результаты работы применимы и к задачам с иным составом требований к комплексу. О практической значимости результатов свидетельствует сокращение времени установления источника несоответствия днем в 1,5 раза и ночью в 3,2 раза, времени закрытия нарядов в 1,6 раза и трудоемкости плановых обследований в 4,4 раза при неизменной численности персонала.

Выводы по главе 3

В третьей главе разработан алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности распределенных программно-аппаратных комплексов, выполнена его экспериментальная проверка в испытательной среде с верифицируемой обратной связью и проведена опытная эксплуатация системы на действующих комплексах. Получены следующие основные результаты.

1. Формализована среда с верифицируемой обратной связью как кортеж $Env = \langle K_{gen}, G_{gen}, V \rangle$ из генератора экземпляров, генератора целей и функции верификации, вычисляющей оценку достижения цели по завершенной трассе без участия эксперта, поскольку истинное состояние экземпляра известно по построению. Разработан метод кросс-уровневого обучения в этой среде, в котором все уровни иерархии параметризуются единой моделью $M\theta$, обучающим материалом служат собственные завершенные трассы агентской системы, а отбор ведется по оценке завершенной трассы. Элитное множество формируется по адаптивному пороговому значению, из элитных трасс извлекаются пары «локальная история, действие» агентов всех уровней, и единая модель дообучается на их объединении. Разнообразие трасс создается температурой генерации и несколькими сеансами на каждом экземпляре.

2. Разработана критериальная модель качества функционирования, в которой сеанс оценивается полнотой анализа, достоверностью, эффективностью, неинвазивностью и результативностью, а оценка завершенной трассы $R(\tau)$ задана их взвешенным геометрическим средним с нормированными весовыми коэффициентами. Показано, что в отличие от взвешенной суммы такая оценка обнуляется при нулевом значении любого критерия с положительным весом и на компромиссе полноты и эффективности имеет внутренний максимум, непрерывно зависящий от весов. Введено расписание весов по итерациям обучения.

3. Разработан алгоритм кросс-уровневого обучения, объединяющий метод и критериальную модель в процедуру с определенными входом, выходом и числом итераций. На каждой итерации порождаются трассы, вычисляется оценка завершенной трассы с весами текущей итерации, отбирается элитное множество и дообучается единая модель, обученная модель оценивается устойчивостью на контрольном множестве, чем замыкается контур самообучения без экспертной разметки.

4. Выполнена программная реализация в виде комплекса из модулей агентской системы, испытательной среды с верифицируемой обратной связью и кросс-уровневого обучения, допускающего подстановку различных больших

языковых моделей в неизменную архитектуру, построена испытательная среда из трех стендов от 6 до 18 вычислительных узлов с известной по построению мерой работоспособности.

5. Экспериментально установлено, что за тридцать итераций кросс-уровневого обучения в пяти независимых прогонах средняя близость оценки к истинному значению меры на контрольном множестве возрастает с 0,83 до 0,87, то есть ошибка оценки снижается в 1,3 раза, а устойчивость, равная доле экземпляров с близостью не ниже 0,85, с 0,57 до 0,65 по средним за последние десять итераций. Ход отдельного прогона не монотонен, направленный рост проявляется в среднем по прогонам. Базовая модель Qwen2.5-72B-Instruct без дообучения по устойчивости лежит внутри диапазона готовых моделей того же класса Claude Haiku и Gemini Flash, от 0,57 до 0,60, а после обучения превосходит лучшую из них по устойчивости и по ошибке оценки, сниженной в 1,2 раза, поэтому достигнутое качество относится к алгоритму кросс-уровневого обучения и не сводится к выбору базовой модели.

6. Проведена опытная эксплуатация системы на трех действующих распределенных программно-аппаратных комплексах с единицами обследования от 6 до 30 узлов. Истинная мера работоспособности действующего комплекса неизвестна, поэтому проверялась применимость результатов работы. Заключение по мере работоспособности принято персоналом без изменений в 71 и 77 % случаев на двух площадках, оценка меры после устранения несоответствия возрастает с 0,62 до 0,97 и с 0,71 до 0,98, а время от регистрации нарушения работоспособности до установления его источника по заключению системы сократилось в 1,5 раза. Там, где состав контролируемых показателей задан процессом парковки, конструкция меры и система применены без изменений, поэтому результаты применимы и к задачам с иным составом требований к комплексу. Система обследует комплексы величиной от 6 до 30 узлов без изменения своей структуры при машинном времени обхода от трех до четырех часов и снижает трудоемкость обследования одного комплекса с 62 до 14 часов инженера. На двух площадках система работает на готовой модели без дообучения, а на третьей алгоритм кросс-

уровневого обучения на стендовой копии в периметре организации снижает долю сеансов без заключения у базовой модели, не проверявшейся на испытательных стендах, с 50 до 9 % за 20 итераций, при этом доля верно установленных источников нарушения работоспособности после обучения выросла с 60 до 75 %.

Результаты получены автором лично и были частично опубликованы в [5; 7; 8; 48], в том числе в соавторстве (вклад автора более 80 %).

Заключение

В настоящем диссертационном исследовании решена актуальная научная задача разработки моделей и алгоритмов оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды, обеспечивающих автономное обследование заранее неизвестного комплекса многоагентной интеллектуальной системой, обучающейся на собственных завершенных сеансах без экспертной разметки. Существующие средства автоматизации оценки работоспособности не располагают формальной моделью многоагентной оценки как задачи принятия решений при неполных сведениях о цифровой среде, закрепляют состав агентов и связи между ними при проектировании и не совершенствуются по собственным завершенным сеансам без участия эксперта.

В ходе выполнения работы получены следующие основные результаты.

1. Проведен анализ существующих подходов к автоматизации оценки работоспособности распределенных программно-аппаратных комплексов и выявлены их ограничения, препятствующие автономному обследованию заранее неизвестного комплекса. Обосновано, что территориальной распределенности, гетерогенности, неполной наблюдаемости и динамичности состояния комплекса в наибольшей мере отвечает многоагентная парадигма. Сценарные средства, платформы имитационного моделирования, одноагентные интеллектуальные системы и многоагентные платформы сопоставлены по автономности обследования, наличию формальной модели процесса, формированию структуры системы под состав комплекса, самообучению на собственных сеансах и автоматической проверке результата сеанса. Ни один класс не покрывает эти критерии целиком, а многоагентные платформы закрепляют состав агентов при проектировании, не накапливают опыт между сеансами и не сверяют результат с истинным состоянием комплекса. Сформулирован комплекс нерешенных научных задач, составляющих предмет диссертационного исследования.

2. Разработана математическая модель оценки работоспособности распределенных программно-аппаратных комплексов на основе децентрализованного частично наблюдаемого марковского процесса принятия решений. Работоспособность комплекса введена как мера со значениями от нуля до единицы, вычисляемая как взвешенное геометрическое среднее степеней соответствия контролируемых показателей требованиям. Качество обследования характеризуется близостью построенной оценки к истинному значению меры. Модель расширяет классическую постановку композитным пространством состояний, объединяющим состояние комплекса с накопленными сведениями агентской системы, зависимостью множества допустимых действий от накопленных сведений и оценкой завершенной трассы исполнения вместо пошагового вознаграждения. Показано, что композитный процесс сохраняет марковское свойство и однородность по времени, оптимальная совместная политика существует и достигается на детерминированной политике, а оценка завершенной трассы, допускающая разложение на вклады шагов, совместима с уравнением оптимальности над состоянием веры. Мощность композитного пространства состояний растет экспоненциально по числу узлов комплекса, что обосновывает переход к приближенным алгоритмам на основе машинного обучения.

3. Разработана параметрическая модель архитектуры многоагентной интеллектуальной системы оценки работоспособности с трехуровневой иерархией агентов на основе единой большой языковой модели. Уровням отвечают глобальное целеполагание единственного координатора, тактическое управление агентов-руководителей и исполнение операций исполнителями. Разделение ролей несет конфигурация агента из системной инструкции и назначенного состава операций, а обмен сведениями ведется через локальную память, наследование контекста при делегировании, почтовый ящик и пространство артефактов. Введен как переменная состояния древовидный динамический граф делегирования, вершины и ребра которого порождаются решениями самих агентов в процессе функционирования под конкретный исследуемый комплекс. Агент создается

только на смежном нижнем уровне с единственным входящим ребром, а делегирование списывается с бюджета действий, поэтому рост графа вширь ограничен бюджетом, цепочка от координатора к исполнителю содержит не более двух связей делегирования, и система масштабируется под состав и размер комплекса при постоянной глубине подчинения.

4. Разработан алгоритм кросс-уровневого обучения многоагентной интеллектуальной системы оценки работоспособности в среде с верифицируемой обратной связью. Среда оценивает итог каждого сеанса без участия эксперта по близости построенной оценки меры работоспособности к истинному значению, известному по построению экземпляра. Единым целевым сигналом алгоритма служит оценка завершенной трассы, вычисляемая критериальной моделью качества функционирования как взвешенное геометрическое среднее полноты обследования, достоверности выводов, эффективности использования ресурсов, неинвазивности и результативности с весовыми коэффициентами, изменяемыми по итерациям обучения от преобладания полноты к целевому назначению обследования. Мультипликативная форма обнуляет оценку при нулевом значении любого критерия с положительным весом и имеет на компромиссе критериев внутренний оптимум, непрерывный по весам, тогда как взвешенная сумма выделяет крайние стратегии. Итерация алгоритма включает порождение трасс при температуре генерации с несколькими сеансами на каждом экземпляре комплекса, отбор элитного множества по адаптивному пороговому значению, процентилю оценок текущей итерации, извлечение из элитных трасс пар «локальная история, действие» агентов всех уровней в единое обучающее множество и дообучение единой модели по отрицательному логарифмическому правдоподобию действий. Адаптивный порог пропускает частично успешные трассы на ранних итерациях и повышает требования к обучающему материалу по мере роста качества. Пары всех уровней обновляют одни и те же параметры, чем обеспечивается совместная оптимизация политик всех уровней иерархии по собственным завершенным сеансам без экспертной разметки.

5. Выполнена программная реализация многоагентной интеллектуальной системы оценки работоспособности в виде комплекса взаимодействующих модулей, каждый из которых решает самостоятельную задачу и заменяется альтернативной реализацией без изменения остальных. Модуль агентской системы строит оценку работоспособности комплекса трехуровневой иерархией агентов. Модуль испытательной среды с верифицируемой обратной связью порождает экземпляры комплекса с известным по построению истинным состоянием и без участия эксперта сопоставляет построенную оценку с истинным значением меры. Модуль кросс-уровневого обучения отбирает элитное множество трасс, формирует обучающие пары агентов всех уровней и дообучает единую модель. Все уровни иерархии параметризованы одной открытой большой языковой моделью, дообучаемой низкоранговыми адаптерами, а единый интерфейс поставщиков моделей позволяет подставлять в неизменную архитектуру различные базовые модели. Программный комплекс выполняет полный цикл обследования от сбора сведений до формирования заключения без поэтапного участия специалиста, на него получено свидетельство о государственной регистрации программы для ЭВМ.

6. Проведены экспериментальные исследования в испытательной среде с верифицируемой обратной связью на трех стендах от 6 до 18 вычислительных узлов с известной по построению мерой работоспособности. Итоговое сравнение выполнено по близости оценки к истинной мере работоспособности и по устойчивости, равной доле экземпляров с близостью не ниже 0,85, на контрольном множестве экземпляров, не участвовавших в дообучении. За тридцать итераций кросс-уровневого обучения в пяти независимых прогонах средняя близость оценки возрастает с 0,83 до 0,87, то есть ошибка оценки снижается в 1,3 раза, а устойчивость с 0,57 до 0,65 по средним за последние десять итераций, при этом ход отдельного прогона не монотонен и направленный рост проявляется в среднем по прогонам. Базовая модель Qwen2.5-72B-Instruct без дообучения по устойчивости лежит внутри диапазона готовых моделей того же класса Claude Haiku и Gemini Flash, от 0,57 до 0,60, а после обучения

превосходит лучшую из них по устойчивости и по ошибке оценки, сниженной в 1,2 раза, поэтому достигнутое качество относится к алгоритму кросс-уровневого обучения и не сводится к выбору базовой модели.

7. Проведена опытная эксплуатация системы на трех действующих распределенных программно-аппаратных комплексах, сети автоматизированных парковочных объектов, сети оператора связи и информационных системах заказчиков на сопровождении. Истинная мера работоспособности действующего комплекса неизвестна, поэтому проверялась применимость результатов работы. Заключение по мере работоспособности принято персоналом без изменений в 71 и 77 % случаев на двух площадках, оценка меры после устранения несоответствия возрастает с 0,62 до 0,97 и с 0,71 до 0,98, а время от регистрации нарушения работоспособности до установления его источника по заключению системы сократилось в 1,5 раза. Там, где состав контролируемых показателей задан процессом эксплуатации, конструкция меры и система применены без изменений, поэтому результаты работы применимы к задачам с иным составом требований к комплексу. Система обследует комплексы величиной от 6 до 30 узлов без изменения своей структуры при машинном времени обхода от трех до четырех часов и снижает трудоемкость обследования одного комплекса с 62 до 14 часов инженера. На двух площадках система работает на готовой модели без дообучения, а на третьей алгоритм кросс-уровневого обучения на стендовой копии в периметре организации снижает долю сеансов без заключения у базовой модели, не проверявшейся на испытательных стендах, с 50 до 9 % за 20 итераций, при этом доля верно установленных источников нарушения работоспособности после обучения выросла с 60 до 75 %. Внедрение результатов в трех организациях и в учебный процесс университета подтверждено актами, приведенными в приложении А.

Таким образом, все поставленные в диссертационном исследовании задачи решены, а основная цель, состоящая в разработке моделей и алгоритмов оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды,

обеспечивающих автономное обследование заранее неизвестного комплекса многоагентной интеллектуальной системой, обучающейся на собственных завершенных сеансах без экспертной разметки, достигнута.

Направлениями дальнейших исследований являются масштабирование системы на комплексы с сотнями узлов и многоуровневой сегментацией, разработка методов дообучения по журналам эксплуатации без разделения сред обучения и эксплуатации, перенос разработанных моделей и алгоритма на мониторинг промышленных систем управления и аудит облачных инфраструктур, а также исследование взаимодействия агентской системы со специалистом для повышения управляемости автономного обследования.

Все результаты, представленные в диссертационной работе, получены автором лично и опубликованы, в том числе в соавторстве (вклад автора более 80 %), в рецензируемых научных изданиях, рекомендованных ВАК РФ [4; 5; 6; 7; 8; 9; 10; 11; 46; 50], и в изданиях, индексируемых в базах данных Scopus и Web of Science [63; 74; 129]. Получены два свидетельства о государственной регистрации программ для ЭВМ № 2026680346 [48] и № 2024614939 [47].

Список литературы

1. Акопов, А.С. Моделирование и оптимизация стратегий принятия индивидуальных решений в многоагентных социально-экономических системах с использованием машинного обучения / А.С. Акопов // Бизнес-информатика. – 2023. – Т. 17, № 2. – С. 7–19.
2. Аналитический обзор подходов к обнаружению вторжений, основанных на федеративном обучении : преимущества использования и открытые задачи / Е.С. Новикова, Е.В. Федорченко, И.В. Котенко, И.И. Холод // Информатика и автоматизация. – 2023. – Т. 22, № 5. – С. 1034–1082.
3. Астраханцев, Р.Г. Анализ современных тенденций развития технологии «блокчейн» и цифровых валют / Р.Г. Астраханцев, А.Б. Лось, Р.Ш. Мухамадиева // Вопросы кибербезопасности. – 2019. – № 5 (33). – С. 57–62.
4. Астраханцев, Р.Г. Архитектура многоагентной интеллектуальной системы на основе параметрической модели для анализа распределенных программно-аппаратных комплексов / Р.Г. Астраханцев, С.П. Бобков // Современные наукоемкие технологии. Региональное приложение. – 2026. – № 2 (86). – С. 70–81.
5. Астраханцев, Р.Г. Критериальная модель оценки качества функционирования многоагентной интеллектуальной системы анализа распределенных программно-аппаратных комплексов / Р.Г. Астраханцев, С.П. Бобков // Современные наукоемкие технологии. Региональное приложение. – 2026. – № 3 (87). – С. 77–83.
6. Астраханцев, Р.Г. Математическая модель автоматизированного анализа состояния распределенных программно-аппаратных комплексов на основе модифицированного марковского процесса принятия решений / Р.Г. Астраханцев // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2026. – № 2 (68). – С. 117–128.

7. Астраханцев, Р.Г. Метод кросс-уровневого обучения интеллектуальной многоагентной системы в среде с верифицируемой обратной связью / Р.Г. Астраханцев, С.П. Бобков // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2026. – № 3 (69). – С. 83–90.
8. Астраханцев, Р.Г. Методологические основы построения многоагентной интеллектуальной системы / Р.Г. Астраханцев, С.П. Бобков // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2026. – № 1 (67). – С. 127–136.
9. Астраханцева, И.А. Агентные интеллектуальные системы с точки зрения системного анализа / И.А. Астраханцева, Р.Г. Астраханцев, С.П. Бобков // Современные наукоемкие технологии. Региональное приложение. – 2025. – № 4 (84). – С. 101–109.
10. Астраханцева, И.А. Системный подход к анализу фрактальной природы сложных технических систем / И.А. Астраханцева, С.В. Горев, Р.Г. Астраханцев // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2023. – № 3 (57). – С. 89–97.
11. Астраханцева, И.А. Фрактальный анализ в оценке эффективности и надежности сложных технических систем / И.А. Астраханцева, С.В. Горев, Р.Г. Астраханцев // Современные наукоемкие технологии. Региональное приложение. – 2023. – № 4 (76). – С. 60–68.
12. Бобков, С.П. Исследование структуры потоков в технологических аппаратах с использованием дискретных динамических моделей / С.П. Бобков, Р.Г. Астраханцев, Е.А. Павлова // Современные наукоемкие технологии. Региональное приложение. – 2024. – № 1 (77). – С. 95–101.
13. Бурков, В.Н. Механизмы управления в сетевых структурах / В.Н. Бурков, Н.А. Кузнецов, Д.А. Новиков // Автоматика и телемеханика. – 2002. – № 12. – С. 96–115.

14. Бурый, А.С. Агентные модели принятия решений на основе генеративного искусственного интеллекта / А.С. Бурый, О.С. Цаплина // Правовая информатика. – 2025. – № 1. – С. 101–109.
15. Васильев, С.Н. От классических задач регулирования к интеллектуальному управлению / С.Н. Васильев // Известия Российской академии наук. Теория и системы управления. – 2001. – № 1. – С. 5–22.
16. Городецкий, В.И. Многоагентные системы : современное состояние исследований и перспективы применения / В.И. Городецкий // Новости искусственного интеллекта. – 1996. – № 1. – С. 44–59.
17. Городецкий, В.И. Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации / В.И. Городецкий // Известия Российской академии наук. Теория и системы управления. – 2012. – № 2. – С. 92–120.
18. ГОСТ Р 59276–2020. Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения : национальный стандарт Российской Федерации : дата введения 2021-03-01. – Москва : Стандартиформ, 2021. – 12 с.
19. ГОСТ Р 59277–2020. Системы искусственного интеллекта. Классификация систем искусственного интеллекта : национальный стандарт Российской Федерации : дата введения 2021-03-01. – Москва : Стандартиформ, 2021. – 16 с.
20. ГОСТ Р 59898–2021. Оценка качества систем искусственного интеллекта. Общие положения : национальный стандарт Российской Федерации : дата введения 2022-03-01. – Москва : Российский институт стандартизации, 2021. – 20 с.
21. ГОСТ Р ИСО/МЭК 25010–2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов : национальный стандарт Российской Федерации : дата введения 2016-06-01. – Москва : Стандартиформ, 2015. – 36 с.

22. Грушо, А.А. Теоретические основы защиты информации / А.А. Грушо, Е.Е. Тимонина. – Москва : Яхтсмен, 1996. – 187, [1] с.
23. Гулай, А.В. Адаптивное управление многоагентной интеллектуальной системой : алгоритмический аспект / А.В. Гулай, В.М. Зайцев // Наука и техника. – 2025. – Т. 24, № 3. – С. 167–180.
24. Имитационная модель движения вещества в технологическом аппарате / С.П. Бобков, Р.Г. Астраханцев, А.А. Самарский, Е.А. Павлова // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2024. – № 2 (60). – С. 32–40.
25. Имитационное моделирование химического реактора трубчатого типа / С.П. Бобков, Р.Г. Астраханцев, Е.А. Павлова, И.А. Астраханцева // Имитационное моделирование. Теория и практика (ИММОД-2025) : труды конференции. – Санкт-Петербург, 2025. – С. 607–615.
26. Использование дискретного вероятностного подхода для моделирования проточных трубчатых реакторов / С.П. Бобков, И.А. Астраханцева, А.А. Гушин [и др.] // Известия высших учебных заведений. Серия: Химия и химическая технология. – 2025. – Т. 68, № 2. – С. 96–101.
27. Качала, В.В. Основы теории систем и системного анализа / В.В. Качала. – 2-е изд. – Москва : Горячая линия – Телеком, 2012. – 210 с.
28. Колесников, А.В. Гибридные интеллектуальные системы с самоорганизацией : координация, согласованность, спор / А.В. Колесников, И.А. Кириков, С.В. Листопад. – Москва : ИПИ РАН, 2014. – 189 с.
29. Концептуальные основы оценки уровня защищенности автоматизированных систем на основе их уязвимости / А.О. Ефимов, И.И. Лившиц, Т.В. Мещерякова, Е.А. Рогозин // Безопасность информационных технологий. – 2023. – Т. 30, № 2. – С. 63–79.
30. Копосов, А.С. Управление с итеративным обучением стохастическими мультиагентными системами с изменяемой желаемой траекторией и топологией / А.С. Копосов, П.В. Пакшин // Автоматика и телемеханика. – 2023. – № 6. – С. 79–99.

31. Котенко, И.В. Анализ защищенности компьютерных сетей на основе моделирования действий злоумышленников и построения графа атак / И.В. Котенко, М.В. Степашкин // Труды Института системного анализа Российской академии наук. – 2007. – Т. 31. – С. 126–207.
32. Котенко, И.В. Интеллектуальные механизмы управления кибербезопасностью / И.В. Котенко // Труды Института системного анализа Российской академии наук. – 2009. – Т. 41. – С. 74–103.
33. Листопад, С.В. Метод на основе нечётких правил для управления конфликтами агентов в гибридных интеллектуальных многоагентных системах / С.В. Листопад, И.А. Кириков // Информатика и её применения. – 2023. – Т. 17, № 1. – С. 66–72.
34. Методика интеграции многоагентных моделей в технологию управления многоспутниковыми космическими системами мониторинга / А.Е. Привалов, А.М. Зубачев, Р.П. Власов, Б.А. Данилюк // Известия Тульского государственного университета. Технические науки. – 2023. – № 2. – С. 193–197.
35. Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения / И.В. Котенко, И.Б. Саенко, О.С. Лаута, А.М. Крибель // Информатика и автоматизация. – 2022. – Т. 21, № 6. – С. 1328–1358.
36. Мишин, С.П. Оптимальное стимулирование в многоуровневых иерархических структурах / С.П. Мишин // Автоматика и телемеханика. – 2004. – № 5. – С. 96–119.
37. Многоагентная динамическая модель многокритериального информационного взаимодействия структурных элементов самоорганизующейся сети передачи данных наземно-воздушной системы мониторинга / В.В. Бородин, Д.А. Клецков, А.В. Тимошенко, В.А. Щевцов // Известия Российской академии наук. Теория и системы управления. – 2023. – № 1. – С. 123–136.

38. Моисеев, Н.Н. Математические задачи системного анализа / Н.Н. Моисеев. – Москва : Наука, 1981. – 488 с.
39. Нестеренко, А.Ю. Об одном подходе к разработке пороговой схемы электронной подписи ГОСТ Р 34.10 и ее расширений / А.Ю. Нестеренко, Р.Г. Астраханцев // Алгебра, теория чисел, дискретная геометрия и многомасштабное моделирование : современные проблемы, приложения и проблемы истории : материалы XXI Международной конференции, посвященной 85-летию со дня рождения А.А. Карацубы. – Тула, 2022. – С. 170–172.
40. Новиков, Д.А. Иерархические модели управления / Д.А. Новиков // Программа 5-й Российской мультиконференции по проблемам управления. – Санкт-Петербург : ЦНИИ «Электроприбор», 2012. – С. 19–20.
41. Пантелеев, М.Г. Архитектура среды имитационного моделирования многоагентных систем реального времени в задачах командного противодействия / М.Г. Пантелеев, С.М. Литягин // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2025. – № 11. – С. 93–98.
42. Петренко, С.А. Методы обнаружения вторжений и аномалий функционирования киберсистем / С.А. Петренко // Труды Института системного анализа РАН. – 2009. – Т. 41. – С. 194–202.
43. Пospelов, Д.А. Ситуационное управление : теория и практика / Д.А. Пospelов. – Москва : Наука, 1986. – 288 с.
44. Пророков, В.А. Новый подход к безопасности ПО : искусственный интеллект в анализе и устранении уязвимостей / В.А. Пророков, М.С. Щеголева, Р.Г. Астраханцев // Сборник научных трудов вузов России «Проблемы экономики, финансов и управления производством». – 2024. – № 54. – С. 163–165.
45. Рыбина, Г.В. Исследование подходов к проектированию архитектуры многоагентной системы в рамках моделей Индустрия 4.0 для цифровизации управления городским парковочным пространством / Г.В. Рыбина,

- В.Ю. Степаньков // Приборы и системы. Управление, контроль, диагностика. – 2023. – № 6. – С. 24–33.
46. Самсонова, А.И. Интеллектуальная система анализа пользовательской активности в защищенных операционных средах / А.И. Самсонова, И.А. Суворов, Р.Г. Астраханцев // Известия высших учебных заведений. Серия: Экономика, финансы и управление производством. – 2025. – № 1 (63). – С. 79–85.
 47. Свидетельство о государственной регистрации программы для ЭВМ № 2024614939 Российская Федерация. Поиск невозможных дифференциалов для блочного шифра КБ256-3 : № 2024613936 : заявл. 28.02.2024 : опубл. 29.02.2024 / Астраханцева И.А., Астраханцев Р.Г.
 48. Свидетельство о государственной регистрации программы для ЭВМ № 2026680346 Российская Федерация. Модели и методы автоматизированного анализа распределенных программно-аппаратных комплексов на основе многоагентной интеллектуальной системы : № 2026667925 : заявл. 29.05.2026 : опубл. 10.07.2026 / Астраханцев Р.Г., Бобков С.П.
 49. Симонова, Л.А. Разработка интеллектуального модуля предиктивной аналитики на основе многоагентной системы на примере производства труб из полимерных материалов / Л.А. Симонова, О.Ю. Бочаров // Научно-технический вестник Поволжья. – 2023. – № 12. – С. 540–544.
 50. Система фильтрации ложноположительных результатов OSINT с использованием методов машинного обучения для оценки принадлежности цифровых активов компаний / И.А. Астраханцева, А.О. Брюшинин, Р.Г. Астраханцев, Т.Е. Котенев // Современные наукоемкие технологии. Региональное приложение. – 2025. – № 3 (83). – С. 57–67.
 51. Стрижко, М.А. Модуль интеллектуального анализа сетевого трафика в системах обнаружения вторжений на базе совокупности нейронных сетей / М.А. Стрижко, В.В. Червинский, В.В. Турупалов // Вестник Донецкого

- национального университета. Серия Г: Технические науки. – 2023. – № 1. – С. 23–33.
52. Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям : философия, психология, информатика / В.Б. Тарасов. – Москва : Эдиториал УРСС, 2002. – 352 с.
 53. Чечулин, А.А. Построение графов атак на основе моделей нарушителей и данных об уязвимостях и шаблонах атак / А.А. Чечулин, И.В. Котенко // Региональная информатика «РИ-2012» : материалы юбилейной XIII Санкт-Петербургской международной конференции. – Санкт-Петербург : СПОИСУ, 2012. – С. 129–130.
 54. Штеренберг, С.И. Метод построения архитектуры интеллектуальной системы обнаружения вторжений на основе квазибиологической парадигмы / С.И. Штеренберг // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2024. – № 1. – С. 82–90.
 55. Язов, Ю.К. Проблемные вопросы управления защитой информации от утечки по техническим каналам с применением многоагентных систем / Ю.К. Язов, А.О. Авсентьев // Вопросы кибербезопасности. – 2024. – № 6 (64). – С. 85–97.
 56. A Survey on Large Language Model Based Autonomous Agents / L. Wang, C. Ma, X. Feng [et al.] // Frontiers of Computer Science. – 2024. – Vol. 18, No. 6. – Art. 186345.
 57. A Survey on Systems Security Metrics / M. Pendleton, R. Garcia-Lebron, J.-H. Cho, S. Xu // ACM Computing Surveys. – 2016. – Vol. 49, No. 4. – P. 1–35.
 58. Absolute Zero : Reinforced Self-play Reasoning with Zero Data / A. Zhao, Y. Wu, Y. Yue [et al.] // arXiv preprint arXiv:2505.03335. – 2025. – URL: <https://arxiv.org/abs/2505.03335> (дата обращения: 12.09.2026).
 59. AI-Powered Penetration Testing Using Shennina : From Simulation to Validation / S. Karagiannis, C. Fusco, L. Agathos [et al.] // Proceedings of the 19th

- International Conference on Availability, Reliability and Security (ARES 2024). – New York : ACM, 2024. – P. 1–7.
60. Amato, C. Scalable Planning and Learning for Multiagent POMDPs / C. Amato, F.A. Oliehoek // Proceedings of the AAAI Conference on Artificial Intelligence. – 2015. – Vol. 29, No. 1. – P. 1995–2002.
 61. Anderson, R. Security Engineering : A Guide to Building Dependable Distributed Systems / R. Anderson. – 3rd ed. – Indianapolis : Wiley, 2020. – 1232 p.
 62. Anderson, R. Why Information Security is Hard : An Economic Perspective / R. Anderson // Proceedings of the 17th Annual Computer Security Applications Conference. – Los Alamitos : IEEE Computer Society, 2001. – P. 358–365.
 63. Astrakhantseva, I.A. Randomized C/C++ Dynamic Memory Allocator / I.A. Astrakhantseva, R.G. Astrakhantsev, A.V. Mitin // Journal of Physics: Conference Series. – 2021. – Vol. 2001, No. 1. – Art. 012006.
 64. Asynchronous Methods for Deep Reinforcement Learning / V. Mnih, A. Puigdomenech Badia, M. Mirza [et al.] // Proceedings of the 33rd International Conference on Machine Learning (ICML). – 2016. – Vol. 48. – P. 1928–1937.
 65. AutoAttacker : A Large Language Model Guided System to Implement Automatic Cyber-Attacks / J. Xu, J.W. Stokes, G. McDonald [et al.] // arXiv preprint arXiv:2403.01038. – 2024. – URL: <https://arxiv.org/abs/2403.01038> (дата обращения: 12.09.2026).
 66. Autonomous Security Analysis and Penetration Testing (ASAP) / A. Chowdhary, D. Huang, J.S. Mahendran [et al.] // 2020 16th International Conference on Mobility, Sensing and Networking (MSN). – Piscataway : IEEE, 2020. – P. 508–515.
 67. Bengio, Y. Representation Learning : A Review and New Perspectives / Y. Bengio, A. Courville, P. Vincent // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2013. – Vol. 35, No. 8. – P. 1798–1828.
 68. BreachSeek : A Multi-Agent Automated Penetration Tester / I. Alshehri, A. Alshehri, A. Almalki [et al.] // arXiv preprint arXiv:2409.03789. – 2024. – URL: <https://arxiv.org/abs/2409.03789> (дата обращения: 12.09.2026).

69. Counterfactual Multi-Agent Policy Gradients / J. Foerster, G. Farquhar, T. Afouras [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. – 2018. – Vol. 32, No. 1. – P. 2974–2982.
70. Curtis, J.A. The Role of AI in Modern Penetration Testing / J.A. Curtis, N.U. Eisty // arXiv preprint arXiv:2512.12326. – 2025. – URL: <https://arxiv.org/abs/2512.12326> (дата обращения: 12.09.2026).
71. CybORG : A Gym for the Development of Autonomous Cyber Agents / M. Standen, M. Lucas, D. Bowman [et al.] // arXiv preprint arXiv:2108.09118. – 2021. – URL: <https://arxiv.org/abs/2108.09118> (дата обращения: 12.09.2026).
72. Deep Reinforcement Learning : A Brief Survey / K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath // IEEE Signal Processing Magazine. – 2017. – Vol. 34, No. 6. – P. 26–38.
73. Developing Optimal Causal Cyber-Defence Agents via Cyber Security Simulation / A. Andrew, S. Spillard, J. Collyer, N. Dhir // arXiv preprint arXiv:2207.12355. – 2022. – URL: <https://arxiv.org/abs/2207.12355> (дата обращения: 12.09.2026).
74. Differences with High Probability and Impossible Differentials for the KB-256 Cipher / R.G. Astrakhantsev, A. Chuhno, A. Dmukh [et al.] // Journal of Computer Virology and Hacking Techniques. – 2024. – Vol. 20, No. 3. – P. 525–531.
75. Dorri, A. Multi-Agent Systems : A Survey / A. Dorri, S.S. Kanhere, R. Jurdak // IEEE Access. – 2018. – Vol. 6. – P. 28573–28593.
76. GAIL-PT : An Intelligent Penetration Testing Framework with Generative Adversarial Imitation Learning / J. Chen, S. Hu, H. Zheng [et al.] // Computers & Security. – 2023. – Vol. 126. – Art. 103055.
77. Generative Adversarial Nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza [et al.] // Advances in Neural Information Processing Systems. – 2014. – Vol. 27. – P. 2672–2680.
78. Ghanem, M.C. Hierarchical Reinforcement Learning for Efficient and Effective Automated Penetration Testing of Large Networks / M.C. Ghanem, T.M. Chen, E.G. Nepomuceno // Journal of Intelligent Information Systems. – 2023. – Vol. 60. – P. 281–303.

79. Ghanem, M.C. Reinforcement Learning for Efficient Network Penetration Testing / M.C. Ghanem, T.M. Chen // Information. – 2020. – Vol. 11, No. 1. – Art. 6.
80. Goldman, C.V. Decentralized Control of Cooperative Systems : Categorization and Complexity Analysis / C.V. Goldman, S. Zilberstein // Journal of Artificial Intelligence Research. – 2004. – Vol. 22. – P. 143–174.
81. Hinton, G.E. Reducing the Dimensionality of Data with Neural Networks / G.E. Hinton, R.R. Salakhutdinov // Science. – 2006. – Vol. 313, No. 5786. – P. 504–507.
82. Hu, Z. Automated Penetration Testing Using Deep Reinforcement Learning / Z. Hu, R. Beuran, Y. Tan // 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). – Piscataway : IEEE, 2020. – P. 2–10.
83. Human-Level Control through Deep Reinforcement Learning / V. Mnih, K. Kavukcuoglu, D. Silver [et al.] // Nature. – 2015. – Vol. 518, No. 7540. – P. 529–533.
84. INNES : An intelligent network penetration testing model based on deep reinforcement learning / Q. Li, M. Hu, H. Hao [et al.] // Applied Intelligence. – 2023. – Vol. 53. – P. 27110–27127.
85. Intelligent, Automated Red Team Emulation / A. Applebaum, D. Miller, B. Strom [et al.] // Proceedings of the 32nd Annual Conference on Computer Security Applications. – New York : ACM, 2016. – P. 363–373.
86. Jennings, N.R. On Agent-Based Software Engineering / N.R. Jennings // Artificial Intelligence. – 2000. – Vol. 117, No. 2. – P. 277–296.
87. Jiang, J. Fully Decentralized Cooperative Multi-Agent Reinforcement Learning : A Survey / J. Jiang, K. Su, Z. Lu // arXiv preprint arXiv:2401.04934. – 2024. – URL: <https://arxiv.org/abs/2401.04934> (дата обращения: 12.09.2026).
88. K-Means Analysis of Spectral Properties of BODIPY Dye During Compression on Air-Water Interface : Construction of Dataset for Effective Clustering / I.A. Astrakhantseva, R.G. Astrakhantsev, S.D. Usoltsev [et al.] // Liquid Crystals and their Application. – 2024. – Vol. 24, No. 2. – P. 43–53.

89. Kaelbling, L.P. Planning and Acting in Partially Observable Stochastic Domains / L.P. Kaelbling, M.L. Littman, A.R. Cassandra // Artificial Intelligence. – 1998. – Vol. 101, No. 1–2. – P. 99–134.
90. Kaelbling, L.P. Reinforcement Learning : A Survey / L.P. Kaelbling, M.L. Littman, A.W. Moore // Journal of Artificial Intelligence Research. – 1996. – Vol. 4. – P. 237–285.
91. Large Language Model Based Multi-Agents : A Survey of Progress and Challenges / T. Guo, X. Chen, Y. Wang [et al.] // arXiv preprint arXiv:2402.01680. – 2024. – URL: <https://arxiv.org/abs/2402.01680> (дата обращения: 12.09.2026).
92. LeCun, Y. Deep Learning / Y. LeCun, Y. Bengio, G. Hinton // Nature. – 2015. – Vol. 521, No. 7553. – P. 436–444.
93. Maeda, R. Automating Post-Exploitation with Deep Reinforcement Learning / R. Maeda, M. Mimura // Computers & Security. – 2021. – Vol. 100. – Art. 102108.
94. Mastering the game of Go without human knowledge / D. Silver, J. Schrittwieser, K. Simonyan [et al.] // Nature. – 2017. – Vol. 550, No. 7676. – P. 354–359.
95. Mathematical discoveries from program search with large language models / B. Romera-Paredes, M. Barekatin, A. Novikov [et al.] // Nature. – 2024. – Vol. 625, No. 7995. – P. 468–475.
96. Maximum Entropy Inverse Reinforcement Learning / B.D. Ziebart, A. Maas, J.A. Bagnell, A.K. Dey // Proceedings of the AAAI Conference on Artificial Intelligence. – 2008. – Vol. 23, No. 1. – P. 1433–1438.
97. Mesarovic, M.D. Multilevel Systems and Concepts in Process Control / M.D. Mesarovic // Proceedings of the IEEE. – 1970. – Vol. 58, No. 1. – P. 111–125.
98. Mesarovic, M.D. Two Coordination Principles and Their Application in Large Scale Systems Control / M.D. Mesarovic, D. Macko, Y. Takahara // Automatica. – 1970. – Vol. 6, No. 2. – P. 261–270.
99. MetaGPT : Meta Programming for a Multi-Agent Collaborative Framework / S. Hong, M. Zhuge, J. Chen [et al.] // arXiv preprint arXiv:2308.00352. – 2023. – URL: <https://arxiv.org/abs/2308.00352> (дата обращения: 12.09.2026).

100. Miehl, E. A POMDP Approach to the Dynamic Defense of Large-Scale Cyber Networks / E. Miehl, M. Rasouli, D. Teneketzis // IEEE Transactions on Information Forensics and Security. – 2018. – Vol. 13, No. 10. – P. 2490–2505.
101. Model-Based Quantitative Network Security Metrics : A Survey / A.L. Ramos, M. Lazar, R. Holanda Filho, J.J.P.C. Rodrigues // IEEE Communications Surveys & Tutorials. – 2017. – Vol. 19, No. 4. – P. 2704–2734.
102. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning / T. Rashid, M. Samvelyan, C. Schroeder de Witt [et al.] // Journal of Machine Learning Research. – 2020. – Vol. 21, No. 178. – P. 1–51.
103. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments / R. Lowe, Y. Wu, A. Tamar [et al.] // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 6379–6390.
104. Multiagent Cooperation and Competition with Deep Reinforcement Learning / A. Tampuu, T. Matiisen, D. Kodelja [et al.] // PLoS ONE. – 2017. – Vol. 12, No. 4. – Art. e0172395.
105. Network Environment Design for Autonomous Cyberdefense / A. Molina-Markham, C. Minitier, B. Powell, A. Ridley // arXiv preprint arXiv:2103.07583. – 2021. – URL: <https://arxiv.org/abs/2103.07583> (дата обращения: 12.09.2026).
106. Nguyen, T.T. Deep Reinforcement Learning for Cyber Security / T.T. Nguyen, V.J. Reddi // IEEE Transactions on Neural Networks and Learning Systems. – 2023. – Vol. 34, No. 8. – P. 3779–3795.
107. Nguyen, T.T. Deep Reinforcement Learning for Multiagent Systems : A Review of Challenges, Solutions, and Applications / T.T. Nguyen, N.D. Nguyen, S. Nahavandi // IEEE Transactions on Cybernetics. – 2020. – Vol. 50, No. 9. – P. 3826–3839.
108. NIG-AP : A New Method for Automated Penetration Testing / T.-Y. Zhou, Y.-C. Zang, J.-H. Zhu, Q.-X. Wang // Frontiers of Information Technology & Electronic Engineering. – 2019. – Vol. 20, No. 9. – P. 1277–1288.

109. Ning, Z. A Survey on Multi-Agent Reinforcement Learning and Its Application / Z. Ning, L. Xie // *Journal of Automation and Intelligence*. – 2024. – Vol. 3, No. 2. – P. 73–91.
110. Oliehoek, F.A. A Concise Introduction to Decentralized POMDPs / F.A. Oliehoek, C. Amato. – Cham : Springer, 2016. – XX, 134 p.
111. PentestGPT : An LLM-Empowered Automatic Penetration Testing Tool / G. Deng, Y. Liu, V. Mayoral-Vilches [et al.] // *arXiv preprint arXiv:2308.06782*. – 2023. – URL: <https://arxiv.org/abs/2308.06782> (дата обращения: 12.09.2026).
112. Prediction of the Information Security State of the Protected Object Using Recurrent Correction / Y.M. Monakhov, M.Y. Monakhov, A.V. Telny, A.P. Kuznetsova // 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT). – Piscataway : IEEE, 2020. – P. 602–605.
113. Proximal Policy Optimization Algorithms / J. Schulman, F. Wolski, P. Dhariwal [et al.] // *arXiv preprint arXiv:1707.06347*. – 2017. – URL: <https://arxiv.org/abs/1707.06347> (дата обращения: 12.09.2026).
114. Pynadath, D.V. The Communicative Multiagent Team Decision Problem : Analyzing Teamwork Theories and Models / D.V. Pynadath, M. Tambe // *Journal of Artificial Intelligence Research*. – 2002. – Vol. 16. – P. 389–423.
115. Russell, S. Artificial Intelligence : A Modern Approach / S. Russell, P. Norvig. – 4th ed. – Hoboken : Pearson, 2021. – 1132 p.
116. Russell, S. Rationality and Intelligence / S. Russell // *Artificial Intelligence*. – 1997. – Vol. 94, No. 1–2. – P. 57–77.
117. Sarraute, C. POMDPs Make Better Hackers : Accounting for Uncertainty in Penetration Testing / C. Sarraute, O. Buffet, H. Hoffmann // *Proceedings of the AAAI Conference on Artificial Intelligence*. – 2012. – Vol. 26, No. 1. – P. 1816–1824.
118. Schneier, B. Attack Trees : Modeling Security Threats / B. Schneier // *Dr. Dobb's Journal*. – 1999. – Vol. 24, No. 12. – P. 21–29.

119. Schwartz, J. Autonomous Penetration Testing Using Reinforcement Learning / J. Schwartz, H. Kurniawati // arXiv preprint arXiv:1905.05965. – 2019. – URL: <https://arxiv.org/abs/1905.05965> (дата обращения: 12.09.2026).
120. Shoham, Y. If Multi-Agent Learning Is the Answer, What Is the Question? / Y. Shoham, R. Powers, T. Grenager // Artificial Intelligence. – 2007. – Vol. 171, No. 7. – P. 365–377.
121. Shoham, Y. Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations / Y. Shoham, K. Leyton-Brown. – Cambridge : Cambridge University Press, 2009. – 504 p.
122. Solvatochromic Sensitivity of BODIPY Probes : A New Tool for Selecting Fluorophores and Polarity Mapping / F.Yu. Telegin, V.S. Karpova, A.O. Makshanova [et al.] // International Journal of Molecular Sciences. – 2023. – Vol. 24, No. 2. – Art. 1217.
123. Stone, P. Multiagent Systems : A Survey from a Machine Learning Perspective / P. Stone, M. Veloso // Autonomous Robots. – 2000. – Vol. 8, No. 3. – P. 345–383.
124. Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing / J. Zeng, S. Wu, Y. Chen [et al.] // Security and Communication Networks. – 2019. – Vol. 2019. – Art. 2031063.
125. Sutton, R.S. Reinforcement Learning : An Introduction / R.S. Sutton, A.G. Barto. – 2nd ed. – Cambridge : MIT Press, 2018. – 552 p.
126. Taming Decentralized POMDPs : Towards Efficient Policy Computation for Multiagent Settings / R. Nair, M. Tambe, M. Yokoo [et al.] // Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI). – 2003. – P. 705–711.
127. The Complexity of Decentralized Control of Markov Decision Processes / D.S. Bernstein, R. Givan, N. Immerman, S. Zilberstein // Mathematics of Operations Research. – 2002. – Vol. 27, No. 4. – P. 819–840.
128. The Rise and Potential of Large Language Model Based Agents : A Survey / Z. Xi, W. Chen, X. Guo [et al.] // arXiv preprint arXiv:2309.07864. – 2023. – URL: <https://arxiv.org/abs/2309.07864> (дата обращения: 12.09.2026).

129. Transfer Learning for Road-Based Location Classification of Non-Residential Property / L. Mizgirev, E. Galiaskarov, I. Astrakhantseva [et al.] // CEUR Workshop Proceedings. ITIDMS 2021. – 2021. – Vol. 2843. – Paper 42.
130. Value-Decomposition Networks for Cooperative Multi-Agent Learning Based on Team Reward / P. Sunehag, G. Lever, A. Gruslys [et al.] // Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS). – 2018. – P. 2085–2087.
131. VulnBot : Autonomous Penetration Testing for A Multi-Agent Collaborative Framework / H. Kong, D. Hu, J. Ge [et al.] // arXiv preprint arXiv:2501.13411. – 2025. – URL: <https://arxiv.org/abs/2501.13411> (дата обращения: 12.09.2026).
132. Watkins, C.J.C.H. Q-Learning / C.J.C.H. Watkins, P. Dayan // Machine Learning. – 1992. – Vol. 8, No. 3–4. – P. 279–292.
133. Wooldridge, M. An Introduction to MultiAgent Systems / M. Wooldridge. – 2nd ed. – Chichester : Wiley, 2009. – 484 p.
134. Wooldridge, M. Intelligent Agents : Theory and Practice / M. Wooldridge, N.R. Jennings // The Knowledge Engineering Review. – 1995. – Vol. 10, No. 2. – P. 115–152.
135. Zenitani, K. Attack Graph Analysis : An Explanatory Guide / K. Zenitani // Computers & Security. – 2023. – Vol. 126. – Art. 103081.
136. Zhang, K. Multi-Agent Reinforcement Learning : A Selective Overview of Theories and Algorithms / K. Zhang, Z. Yang, T. Basar // Handbook of Reinforcement Learning and Control. – Cham : Springer, 2021. – P. 321–384.
137. Zhu, C. A Survey of Multi-Agent Deep Reinforcement Learning with Communication / C. Zhu, M. Dastani, S. Wang // Autonomous Agents and Multi-Agent Systems. – 2024. – Vol. 38. – Art. 4.

Приложение А

(справочное)

Акты об использовании и внедрении результатов диссертационного исследования

Акты об использовании и внедрении результатов диссертационного исследования, а также свидетельства о государственной регистрации программ для ЭВМ:

– Акт о внедрении результатов диссертационного исследования в сопровождение сетевой инфраструктуры ООО «Интеркомтел», г. Иваново;

– Акт о внедрении результатов диссертационного исследования в плановые обследования информационных систем заказчиков ООО «Профи-Групп», г. Иваново;

– Акт о внедрении результатов диссертационного исследования в диспетчерское сопровождение автоматизированных парковочных объектов ООО «Национальный оператор парковок» (ООО «А1Парк»), г. Москва;

– Акт о внедрении результатов диссертационного исследования в учебный процесс ФГБОУ ВО «Ивановский государственный химико-технологический университет»;

– Свидетельство о государственной регистрации программы для ЭВМ № 2026680346 «Модели и методы автоматизированного анализа распределенных программно-аппаратных комплексов на основе многоагентной интеллектуальной системы»;

– Свидетельство о государственной регистрации программы для ЭВМ № 2024614939 «Поиск невозможных дифференциалов для блочного шифра КБ256-3».



общество с ограниченной ответственностью «Интеркомтел»
153006, г. Иваново, ул. 6-я Меланжевая, д.1
ОГРН 1063702023835
ИНН/КПП 3702087103/370201001

В диссертационный совет 24.2.281.04
при ВлГУ имени А.Г. и Н.Г. Столетовых

АКТ
о внедрении результатов диссертационного исследования
Астраханцева Романа Геннадьевича

Настоящим подтверждаем, что результаты диссертационного исследования Астраханцева Романа Геннадьевича на тему «Модели и алгоритмы оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды» использованы в деятельности ООО «Интеркомтел» при сопровождении сетевой инфраструктуры, обеспечивающей предоставление услуг связи абонентам в городе Иваново и Ивановской области, при выполнении нарядов второй линии на сетевую диагностику.

Многоагентная интеллектуальная система оценки работоспособности построена на разработанных автором математической модели оценки работоспособности на основе децентрализованного частично наблюдаемого марковского процесса принятия решений и параметрической модели архитектуры с трехуровневой иерархией агентов и динамическим графом делегирования, а базовая большая языковая модель системы дообучена алгоритмом кросс-уровневого обучения в среде с верифицируемой обратной связью. Система выполняет полный цикл обследования от сбора сведений до формирования заключения без поэтапного участия специалиста.

Единицей обследования служит участок сети, включающий домовой узел доступа, узел агрегации и затронутые службы адресации, аутентификации, разрешения имен, телевизионного вещания и расчетов с абонентами. Обследование запускается при создании наряда второй линии, до того как инженер берет наряд в работу, поэтому в вечерний пик заключение готово раньше, чем инженер приступает к наряду. Заключение с оценкой меры работоспособности участка, показателем с наименьшей степенью соответствия, элементом, на котором он измерен, и подтверждающими результатами инструментов вписывается в наряд, закрывает наряд инженер кодом источника несоответствия.

Сведения о сети и подключениях абонентов периметр организации не покидают, поэтому большая языковая модель размещена на сервере ООО «Интеркомтел», а ее дообучение выполнено на стендовой копии участка сети в периметре организации, где вносимые несоответствия записываются в журнал стенда и истинное состояние каждого экземпляра известно. Модель с адаптером введена в эксплуатацию после завершения обучения.

В первые недели опытной эксплуатации заключения записывались без показа персоналу, затем открыты. Медиана времени закрытия наряда второй линии снизилась с 54 до 33 минут, число нарядов на одного инженера в вечернюю смену возросло на треть, доля сеансов обследования без заключения, составлявшая у базовой модели половину, после дообучения не превышает десятой части.

Результаты диссертационного исследования Р.Г. Астраханцева сокращают время закрытия нарядов второй линии и повышают пропускную способность вечерней смены без увеличения штата, а алгоритм кросс-уровневого обучения делает базовую модель, размещенную в периметре организации, пригодной для обследования без передачи сведений о сети третьим лицам.

Принято решение об использовании в инфраструктуре компании результатов диссертации.

Генеральный директор
ООО «Интеркомтел»



 /Кузьмин Е.Ю./



ООО «Профи-Групп»

Иваново, ул. 8 Марта, д. 32А, оф 7
ИНН 3702146743 КПП 370201001
8 (4932) 58-44-88

В диссертационный совет 24.2.281.04
при ВлГУ имени А.Г. и Н.Г. Столетовых

АКТ

о внедрении результатов диссертационного исследования
Астраханцева Романа Геннадьевича

Настоящим подтверждаем, что результаты диссертационного исследования Астраханцева Романа Геннадьевича на тему «Модели и алгоритмы оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды» использованы в деятельности ООО «Профи-Групп» при выполнении плановых обследований информационных систем заказчиков компании по договорам сопровождения.

Многоагентная интеллектуальная система оценки работоспособности построена на разработанных автором математической модели оценки работоспособности на основе децентрализованного частично наблюдаемого марковского процесса принятия решений и параметрической модели архитектуры с трехуровневой иерархией агентов и динамическим графом делегирования. Система выполняет полный цикл обследования от сбора сведений до формирования заключения без поэтапного участия специалиста.

Единицей обследования служит информационная система заказчика, включающая серверы приложений и баз данных, рабочие станции ключевых ролей, средства защиты информации и сетевое оборудование, до трех десятков узлов. Обследование выполняется в ночное время с переносного сервера, подключаемого к служебному сегменту системы по согласованию с администратором заказчика, сбор фактов ведут детерминированные сборщики. Заключение содержит перечень несоответствий требованиям договора сопровождения по составу узлов и служб, значениям конфигурационных параметров, состоянию резервного копирования, срокам сертификатов и учетным записям, а также оценку меры работоспособности системы, валидируется инженером сопровождения и передается администратору заказчика.

Система работает на готовой большой языковой модели без дообучения, размещенной на сервере ООО «Профи-Групп», переносной сервер на площадке заказчика несет агентскую систему и сборщики, сведения информационных систем заказчиков третьим лицам не передаются.

После пилотной эксплуатации система работает в основном режиме на всех сопровождаемых системах. Число обследований за квартал возросло втрое при прежней численности инженеров сопровождения, медиана трудозатрат инженера на одну систему снизилась с 62 до 14 часов с учетом дороги, установки сервера и проверки находок. При полной ручной валидации заключений инженером подтверждены около двух третей находок системы, а система называет более половины находок инженера.

Результаты диссертационного исследования Р.Г. Астраханцева положительно отражаются на качестве разрабатываемых программных решений, сокращают трудоемкость ручного сопровождения при анализе состояния программно-аппаратных комплексов и расширяют круг задач, которые могут быть решены средствами автоматизированной многоагентной интеллектуальной системы.

Директор ООО «Профи-Групп»

Машин И.В.

М.П.





Национальный Оператор Парковок
Россия, 121087, Москва,
Барклай д. 6 стр. 25
1 этаж антресоль, комн. 1
Тел. +7 (495) 646 87 02
Email: info@a1park.com

В диссертационный совет 24.2.281.04
при ВлГУ имени А.Г. и Н.Г. Столетовых

АКТ

о внедрении результатов диссертационного исследования
Астраханцева Романа Геннадьевича

Настоящим подтверждаем, что результаты диссертационного исследования Астраханцева Романа Геннадьевича на тему «Модели и алгоритмы оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды» использованы в деятельности организации при круглосуточном диспетчерском сопровождении автоматизированных парковочных объектов и разборе нарушений их работоспособности.

Многоагентная интеллектуальная система оценки работоспособности построена на разработанных автором математической модели оценки работоспособности на основе децентрализованного частично наблюдаемого марковского процесса принятия решений и параметрической модели архитектуры с трехуровневой иерархией агентов и динамическим графом делегирования. Система выполняет полный цикл обследования от сбора сведений до формирования заключения без поэтапного участия специалиста.

Единицей обследования служит парковочный объект со службами ядра, включающий объектный шлюз, камеры распознавания номеров, контроллеры шлагбаумов, стойки оплаты и коммутатор, всего более сорока объектов при ядре из восьми узлов в центре обработки данных. Состав контролируемых показателей задан регламентом оператора, степень соответствия требованиям и мера работоспособности вычисляются по нему без изменений. Обследование запускается при регистрации нарушения работоспособности объекта, которое не устранено стандартным действием диспетчера, до решения о выезде. Заключение с оценкой меры работоспособности объекта, показателем с наименьшей степенью соответствия, элементом, на котором он измерен, и подтверждающими результатами инструментов передается диспетчеру, который устанавливает источник несоответствия и принимает решение о выезде.

Система работает на готовой большой языковой модели без дообучения, агентская система размещена в том же центре обработки данных, что и ядро. Пробы на действующем комплексе ограничены синтетическим запросом к ядру и эталонным снимком для службы распознавания, команд шлагбаумам система не подает.

В первые недели опытной эксплуатации заключения записывались без показа диспетчерам, затем открыты. После включения заключений медиана времени от регистрации нарушения до установления источника несоответствия сократилась днем в полтора раза, а ночью, при одном диспетчере в смене, с 55 до 17 минут, доля нарушений, устраненных без выезда, возросла с трети почти до половины. Результаты диссертационного исследования Р.Г. Астраханцева сокращают время установления источника несоответствия и число выездов на объекты.

Принято решение о дальнейшем использовании программного комплекса в работе компании.

С Уважением,
Генеральный директор ООО «А1 ПАРК»



Швецов Р.М.

Национальный Оператор Парковок
Россия, 121087, Москва, Барклай д. 6 стр. 25
1 этаж антресоль, комн. 1
Тел. +7 (495) 646 87 02 Email: info@a1park.com
www.a1park.com

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение
высшего образования

**«ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ
ХИМИКО-ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»
(«ИГХТУ»)**

В Диссертационный Совет 24.2.281.04
при ВлГУ имени А.Г. и Н.Г. Столетовых

пр. Шереметевский, д. 7, Иваново, 153000
тел. (4932) 32-92-41, факс (4932) 41-79-95
E-mail: rector@isuct.ru, http://www.isuct.ru

ИНН/КПП 3728012818 / 370201001

25.05.2026 № 01-11/140

на № _____ от _____

АКТ

о внедрении результатов диссертационного исследования
Астраханцева Романа Геннадьевича в учебный процесс ИГХТУ

Результаты диссертационного исследования Астраханцева Романа Геннадьевича на тему «Модели и алгоритмы оценки работоспособности распределенных программно-аппаратных комплексов в условиях неопределенности и частичной наблюдаемости цифровой среды», выполненного на кафедре «Информационные технологии и цифровая экономика» ИГХТУ, используются в учебном процессе Ивановского государственного химико-технологического университета. Основные результаты диссертационного исследования внедрены при подготовке учебно-методического материала и проведении лабораторных работ по следующим дисциплинам.

По дисциплинам «Архитектура информационных систем», «Технология обработки информации» и «Информационные технологии» (программа бакалавриата по направлению 09.03.02 «Информационные системы и технологии») используются разработанные автором математическая модель оценки работоспособности на основе децентрализованного частично наблюдаемого марковского процесса принятия решений и параметрическая модель архитектуры многоагентной интеллектуальной системы с трехуровневой иерархией агентов и динамическим графом делегирования. Результаты применяются в качестве примеров при изучении методов проектирования распределенных интеллектуальных систем, а также при выполнении лабораторных работ по реализации компонентов многоагентных платформ.

По дисциплине «Модели информационных процессов и систем» (программа магистратуры по направлению 09.04.02 «Информационные системы и технологии») используются разработанные автором критериальная модель качества функционирования многоагентной интеллектуальной системы, объединяющая пять критериев одного сеанса в оценку завершённой трассы взвешенным геометрическим средним с настраиваемыми весовыми коэффициентами, и алгоритм кросс-уровневого обучения в среде с верифицируемой обратной связью по собственным завершённым сеансам без экспертной разметки. Материалы диссертации применяются при изучении методов моделирования информационных процессов в распределенных интеллектуальных системах и построения критериальных моделей оценки качества.

Реализация результатов диссертационного исследования Р.Г. Астраханцева в учебном процессе способствует повышению качества подготовки обучающихся и формированию профессиональных компетенций в области обработки информации и интеллектуального управления техническими системами.

Ректор



Гордина Н.Е.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2026680346

**Модели и методы автоматизированного анализа
распределенных программно-аппаратных комплексов
на основе многоагентной интеллектуальной системы**

Правообладатель: *Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ивановский государственный химико-технологический
университет» (RU)*

Авторы: *Астраханцев Роман Геннадьевич (RU), Бобков
Сергей Петрович (RU)*



Заявка № 2026667925

Дата поступления 29 мая 2026 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 10 июля 2026 г.

Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 00a570e4f7d-38d531b4b-8818e75f29506
Владелец **Зубов Юрий Сергеевич**
Действителен с 04.07.2025 по 28.11.2026

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024614939

Поиск невозможных дифференциалов для блочного шифра КБ256-3

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный химико-технологический университет» (RU)*

Авторы: *Астраханцева Ирина Александровна (RU), Астраханцев Роман Геннадьевич (RU)*



Заявка № 2024613936

Дата поступления 28 февраля 2024 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 29 февраля 2024 г.

Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 4209b60fe3853164baf96f83b73b4aa7
Владелец: **Зубов Юрий Сергеевич**
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов