ISSN 2712-8687

ПРОБЛЕМЫ УПРАВЛЕНИЯ

2/2025

CONTROL SCIENCES

CONTROL SCIENCES
Научно-технический журнал

6 номеров в год

ISSN 1819-3161 (Print)

ISSN 2712-8687 (Online)

Издается с 2003 года

УЧРЕДИТЕЛЬ и ИЗДАТЕЛЬ

Федеральное государственное бюджетное учреждение науки Институт проблем управления им. В.А. Трапезникова РАН

Адрес редакции
117997, ГСП-7, Москва,
ул. Профсоюзная, д. 65, к. 410

Тел./факс (495) 198-17-20, доб. 1410

E-mail: pu@ipu.ru

Интернет: http://pu.mtas.ru
http://controlsciences.org

Опубликовано: 16 мая 2025 г.

Свидетельство о регистрации
ПИ № ФС 77-49203 от 30 марта 2012 г.
выдано Министерством Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций

Свидетельство о регистрации
Эл № ФС 77-80482 от 17 февраля 2021 г.
выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций

Журнал входит в RSCI на платформе Web of Science и Перечень рецензируемых научных изданий ВАК

Журнал включен в Российский индекс научного цитирования (РИНЦ). На сайте Научной электронной библиотеки (www.elibrary.ru) доступны полные тексты статей.

ПРОБЛЕМЫ УПРАВЛЕНИЯ

2.2025

СОДЕРЖАНИЕ

CONTROL SCIENCES
Scientific Technical Journal

6 issues per year

ISSN 1819-3161 (Print)

ISSN 2712-8687 (Online)

Published since 2003

FOUNDER and PUBLISHER
V.A. Trapeznikov
Institute of Control Sciences
of Russian Academy of Sciences

Editor-in-Chief
D.A. Novikov, RAS Academician

Deputy Editor-in-Chief
F.F. Pashchenko

Executive Editor-in-Chief
N.E. Maximova

Editor
L.V. Petrakova

Editorial address
65 Profsoyuznaya st., office 410,
Moscow 117997, Russia

☎/🖷 +7(495) 198-17-20, ext. 1410

✉ pu@ipu.ru

URL: http://pu.mtas.ru
http://controlsciences.org

Published: May 16, 2025

Registration certificate of
ПИ № ФС 77-49203 of 30 March 2012
issued by the Ministry of Press,
Broadcasting, and Mass Media
of the Russian Federation

Registration certificate of
Эл № ФС 77-80482 of 17 February 2021
issued by the Federal Service
for Supervision of Communications,
Information Technology, and Mass Media

The Journal is indexed in RSCI (Russian Science Citation Index) on the platform Web of Science and in the list of peer-reviewed scientific publications of HAC

On the website of the Scientific electronic library (www.elibrary.ru) full texts of articles are available

CONTROL SCIENCES

2.2025

CONTENTS

УДК 519.72+629.7.05

МЕТОДЫ МОНИТОРИНГА ТЕХНИЧЕСКОГО СОСТОЯНИЯ СИСТЕМ В ЦЕЛЯХ УПРАВЛЕНИЯ ИХ ИЗБЫТОЧНОСТЬЮ. Ч. 1. Встроенные средства контроля и разбиение на классы

В. Н. Буков*, А. М. Бронников**, А. С. Попов***, В. А. Шурман****

*АО «Бортовые аэронавигационные системы», г. Москва,
**Московский государственный технический университет им. Н. Э. Баумана,
***ВУНЦ ВВС «Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина», г. Воронеж,
****АО «НИИ авиационного оборудования», г. Жуковский

*✉ v_bukov@mail.ru, **✉ bronnikov_a_m@mail.ru, ***✉ saga30@yandex.ru, ****✉ shurman@niiao.ru

Аннотация. Управление избыточностью технической системы предполагает использование процедуры мониторинга – контроля текущего состояния ее составных частей с целью реконфигурирования системы для обеспечения повышения эксплуатационных характеристик и автономности их применения. В обзоре, состоящем из четырех частей, приводятся современные методы мониторинга в интересах управления избыточностью. Первая часть обзора посвящена в основном анализу наиболее распространенных в современных технических системах методов мониторинга на основе схем голосования, правил достоверности, контрольных кодов, программного контроля, на использовании которых чаще всего основаны современные встроенные средства контроля. Наряду с ними здесь анализируются давно известные, но менее распространенные в технике методы мониторинга: диагностирование с разделением на классы, а также диагностирование на основе алгебраических инвариантов.

Ключевые слова: мониторинг технического состояния, управление избыточностью, диагностирование, встроенные средства контроля, контрольные коды, разбиение на классы, алгебраические инварианты.

ВВЕДЕНИЕ

Возросшие возможности информационного и математического обеспечения процессов управления сложными динамическими системами и комплексами оборудования позволяют принципиально по-новому подойти к удовлетворению постоянно ужесточаемых требований к их отказоустойчивости, в том числе на основе управляемой избыточности [1, 2]. Одной из задач управления избыточностью перспективных систем является выполнение процедуры мониторинга, т. е. слежения за текущими изменениями [3] готовности к функционированию составных частей таких систем [4, 5] с целью реконфигурирования последних по мере необходимости.

Мониторинг технического состояния объекта (ТСО) заключается в процессе наблюдения за ТСО на протяжении интервала времени жизни объекта (например, на протяжении полета воздушного судна) и базируется на определенной иерархии методов определения ТСО, т. е. технического диагностирования [6]. Согласно изложенному в работе [7] мониторинг является составной частью технического обслуживания.

Техническое диагностирование обычно представляет собой дискретную последовательность

технических диагнозов (результатов диагностирования, привязанных к определенным моментам времени) [6].

Диагностирование технического состояния технических систем является весьма сложной проблемой, требующей привлечения широкого спектра алгоритмических решений. Содержание задачи контроля (диагностирования) систем с инженерной точки зрения заключается в обнаружении (поиске) отказов по доступным для этой цели признакам.

С точки зрения перспектив развития данного направления наиболее сильных результатов в области мониторинга следует ожидать на позициях системного подхода[1], основанного прежде всего на триединстве таких ключевых направлений, как, во-первых, надежность и работоспособность, во-вторых, рациональность (ограничение привлекаемых ресурсов, существенных, например, для самопроверяемых схем [8]), и, в третьих, обоснованная глубина («зернистость») проектирования системы в целом, оптимизирующая баланс крупнофрагментарного и мелкофрагментарного разбиений системы на компоненты.

На рисунке показана классификация применяемых и в настоящее время развиваемых методов диагностирования динамических систем, где указаны не распространенные названия методов, а характеризующие их ключевые слова. Штриховая связь указывает на то обстоятельство, что, с одной стороны, тестовый и функциональный контроль имеют много общего в применяемых методах и процедурах, а с другой стороны, обладают и характерными особенностями, не раскрываемыми в данном обзоре.

Далее кратко изложены положения основных подходов к задаче диагностирования отказов компонентов авиационных комплексов бортового оборудования (КБО). Обзор не претендует на полноту. Он содержит заявку на системность, отражает точку зрения авторов и призван проиллюстрировать возможности, типовые ограничения распространенных подходов и тенденции их развития. Изложение ведется в порядке от наиболее простых (очевидных) методов к все более сложным, требующим от разработчика владения специальным математическим аппаратом.

Классификация методов диагностирования

[1] Этот тезис предложен одним из рецензентов статьи, за что авторы выражают искреннюю благодарность. Вместе с тем научное обоснование и раскрытие соответствующих позиций является предметом отдельной публикации.

1. МОНИТОРИНГ НА ОСНОВЕ ВСТРОЕННЫХ СРЕДСТВ КОНТРОЛЯ

Встроенные средства контроля (ВСК) [9, 10] относятся к наиболее распространенным современным решениям обеспечения требуемого уровня отказобезопасности различных технических систем. В системах же повышенной опасности, к которым, в частности, относятся авиационные КБО, встроенный контроль внедряется во все составляющие компоненты (системы, подсистемы, узлы, модули и даже микросхемы).

ВСК представляют собой совокупность аппаратных или программных компонентов, специально вводимых в состав систем или комплектующих изделий, функциональных узлов (ФУз). Они, как правило, не участвуют в работе функциональных модулей (ФМ) системы или ее ФУз по назначению, а собирают и обобщают различные данные, объективно отражающие, по мнению разработчика, работоспособность этих модулей. В перспективе ВСК могут быть основаны на применении различных методов мониторинга и их сочетаний, освещаемых во всех четырех частях обзора. Здесь ограничимся рассмотрением наиболее распространенных подходов, нашедших широкую реализацию в современных ВСК.

Применяются два различных подхода к организации работы ВСК: тестовый и функциональный контроль.

1.1. Тестовый контроль

При тестовом контроле [11, 12] проверку узлов, устройств и системы в целом осуществляют с помощью специального оборудования – генераторов тестовых воздействий и анализаторов выходных реакций. Необходимость в дополнительном оборудовании и сложность совмещения с нормальным функционированием ограничивают применение тестовых методов исключительно периодами, когда объект контроля не используется по прямому назначению.

В бортовых условиях тестирование осуществляется с использованием специализированных тестов встроенного контроля (ТВК) в фоновом режиме во время специальных интервалов (слотов), выделяемых операционной системой реального времени. Содержание ТВК – сравнение результатов обращения (записи и чтения) к программно-доступным ресурсам вычислителя, включая специально организованные контрольные тракты.

Тестирование с имитацией штатных воздействий осуществляется специальным генератором тестов, выходные реакции сравниваются с эталонными при помощи анализатора.

Вероятностное тестирование основано на использовании тестового генератора псевдослучайных воздействий и статистической обработки выходных данных со сравнением результатов с эталонными, полученными предварительно.

Тестирование с подсчетом числа переключений предполагает генерирование последовательности тестовых наборов сигналов на входе схемы и подсчет переключений на ее выходе, результат сравнивается с эталонным.

При *сигнатурном тестировании* стимуляция объекта контроля осуществляется с помощью генератора псевдослучайных воздействий, а выходные реакции сжимаются посредством сигнатурного анализатора; полученные сигнатуры сравниваются с эталонными.

1.2. Функциональный контроль

Функциональный контроль производится в процессе функционирования объекта контроля по назначению, и его реализация базируется обычно на применении двух основных принципов: схем голосования и правил достоверности.

Принципиальной особенностью *схем сравнения (голосования)* является одновременное использование нескольких одинаковых по назначению и исполнению технических устройств (подсистем, узлов, модулей). Система же диагностирования сводится к средствам, осуществляющим сравнение данных этих систем и выбор предпочтения по заданному правилу сопоставления их выходов. Распространенным вариантом являются так называемые кворум-элементы (КЭ), выделяющие неисправные модули на основе обработки результатов голосования нескольких подключенных ФМ. Суждение о работоспособности ФМ выносится на основе значительного (наибольшего или превышающего пороговое значение) отклонения его выхода от других однотипных модулей [13–15].

Основные особенности метода кворумирования включают:

– предположение о неизменности технического состояния ФМ в пределах цикла;

– предположение о том, что КЭ может быть только исправным[2];

– применимость к числу ФМ, превышающему 2 (при двух ФМ объектом оценки работоспособно-

[2] Речь идет о концептуальном решении, однако известны многоуровневые схемы мажорирования, которые сдвигают это ограничение на более высокие уровни сравнения результатов сравнения.

сти становится пара ФМ, а не каждый ФМ в отдельности);

– предположение о том, что с учетом правил голосования (равноправное, взвешенное, с дискриминациями и пр.) исправные ФМ внутри каждого цикла доминируют над неисправными и реализована возможность отключения неисправных;

– общий поток данных для всех ФМ.

Своеобразная форма сравнения широко реализуется в так называемых самопроверяемых системах [8], в которых совокупность однотипных модулей, подверженных одинаковым входным воздействиям, разбивается на пары, и выходы внутри каждой пары сравниваются между собой. Пара с совпадающими выходами считается работоспособной, в противном случае оба модуля пары полагаются неработоспособными.

Принципиальной особенностью систем, использующих *правила достоверности (ПД)*, является наличие диагностируемого устройства в единственном числе. В зависимости от конкретных условий и решений в качестве правил достоверности могут выступать: сравнение с эталонными (электронными) моделями, фиксирование нарушений заданных временны́х и (или) параметрических интервалов (контроль по допуску на параметр [13]), проверка логических и др. соотношений, вычисление инвариантов разных порядков и пр.

Основные особенности метода использования ПД:

– в пределах цикла работоспособность ФМ не изменяется;

– предполагается, что элемент, реализующий ПД, может быть только работоспособным, в том числе при наличии эталонной модели она может быть только работоспособной;

– применимость к любому числу ФМ;

– предполагается, что входные и выходные данные в достаточной степени информативны;

– каждый ФМ имеет свой поток данных.

Несколько особняком стоит недавно возникшее направление [16], которое можно назвать мониторингом ФМ по его эксплуатационным данным с записью условий применения. Подразумевается, что непосредственно с ФМ конструктивно и функционально связан специальный элемент (чип), собирающий и накапливающий данные об условиях его использования и хранения. В число параметров, хранимых и выдаваемых таким чипом в модуль мониторинга, входят различные данные о ФМ, включая:

– паспортные данные,

– результаты испытаний на разных стадиях жизненного цикла,

– статистику эксплуатационных показателей и характеристик (оценки достигаемой точности, остаток ресурса, энергетические показатели и пр.),

– статистику внешних воздействий во время использования по назначению, при хранении и регламентных работах.

На модуль мониторинга возлагается анализ поступающих данных и формирование на основе этого анализа суждения о возможной работоспособности ФМ.

1.3. Преимущества и условия использования встроенных средств контроля

Можно отметить в разной степени распространенные особенности (ограничения) ВСК:

– наличие слабых[3] предположений о неизменности работоспособности проверяемых устройств внутри цикла мониторинга;

– наличие сильных[4] предположений о работоспособности систем контроля или их основных устройств,

– ограничение минимального или требование большого числа ФМ для случая кворумирования (мажоритарного контроля),

– требование доминирования работоспособных ФМ над неработоспособными,

– возможность оперативного отключения неисправных ФМ,

– требование информативности процессов в ФМ[5].

Основное же преимущество непосредственного использования ВСК (как есть) для мониторинга компонентов в избыточном КБО заключается в отлаженности технологий их создания и практического применения.

2. ИСПОЛЬЗОВАНИЕ КОНТРОЛЬНЫХ КОДОВ

Специфическим направлением в рамках общей задачи встроенного контроля цифровой техники является использование контрольных кодов для обнаружения и исправления ошибок в цифровых данных [11, 17–23].

Наибольшее распространение получили блочные коды, у которых последовательность символов

[3] Удовлетворение предположения не влияет на практическое применение подхода.

[4] Удовлетворение предположения значительно сужает прикладные возможности подхода.

[5] Необходимость такого требования иллюстрируется в ч. 3 обзора.

на выходе источника разбивается на блоки (кодовые слова, кодовые комбинации), содержащие одинаковое число символов. Любой код способен обнаруживать и исправлять ошибки (является помехоустойчивым), если не все кодовые слова этого кода используются для передачи информации [24], иначе говоря, помехоустойчивый код должен быть избыточным. Тем не менее различают два вида помехоустойчивых кодов: коды с обнаружением ошибок и коды с исправлением ошибок (корректирующие коды).

Сущность обнаружения ошибки связана с обнаружением превращения полученного или считанного (разрешенного) кодового слова в так называемое запрещенное. При этом ошибки, связанные с его превращением в другое разрешенное кодовое слово, не обнаруживаются.

Исправление ошибок представляет собой более сложную операцию, и для ее выполнения все запрещенные кодовые слова разбиваются на непересекающиеся подмножества, каждое из которых приписывается одному из разрешенных кодовых слов. Таким образом, принадлежность возникшего запрещенного кодового слова какому-либо подмножеству истолковывается как соответствующее разрешенное кодовое слово. Если же возникшее запрещенное кодовое слово не принадлежит ни одному из подмножеств, то ошибка обнаруживается, но не исправляется.

Свойства кодов по обнаружению и исправлению ошибок характеризуются коэффициентами обнаружения ($K_{\text{об}}$) и исправления ($K_{\text{ис}}$), имеющими вероятностную природу. При этом всегда $K_{\text{об}} > K_{\text{ис}}$.

К обнаруживающим относятся, например, следующие распространенные коды.

Код с контролем по запрещенным комбинациям заключается в обнаружении комбинаций значений разрядов в кодовом слове, объявленных недопустимыми (например, обращение к несуществующему адресу).

Код с проверкой четности может рассматриваться как частный случай кода с контролем по запрещенным комбинациям и образуется путем добавления к информационным разрядам, хранящим кодовое слово, одного неинформационного разряда (свертки по mod2, дополнения количества единиц в коде до нечетности, проверяемой при каждом обмене между регистрами). Код с проверкой четности прост в технической реализации, позволяет обнаруживать ошибки нечетной кратности.

Итеративные коды относятся к классу *кодов произведения* и могут быть представлены в виде прямоугольных матриц или таблиц (могут строиться на основе матриц и более высоких размерностей); при этом информационные символы, записываемые по строкам и столбцам, могут кодироваться либо одним типом помехоустойчивого кода, либо разными.

В частном случае итеративные коды представляют собой развитие кода проверки четности в том смысле, что применяются для раздельной проверки четности по строкам, столбцам и другим структурам хранимых и передаваемых массивов данных. Такие коды отличаются простотой и эффективностью обнаружения многократных ошибок. В качестве иллюстрации представлен принцип контроля двумерного матричного массива, включающий проверку четности по строкам, столбцам и главной диагонали [22]:

1	0	1	1	1	$\text{чет}_{1j}=0$
0	0	1	0	1	$\text{чет}_{2j}=0$
1	1	1	0	0	$\text{чет}_{3j}=1$
0	1	0	0	1	$\text{чет}_{4j}=0$
1	1	1	0	1	$\text{чет}_{5j}=0$
$\text{чет}_{i1}=1$	$\text{чет}_{i2}=1$	$\text{чет}_{i3}=0$	$\text{чет}_{i4}=1$	$\text{чет}_{i5}=0$	$\text{чет}_{ii}=1$

.

Корреляционные коды предусматривают введение дополнительного контрольного разряда для каждого информационного разряда слова таким образом, что исходному информационному значению «0» соответствует запись «01», а значению «1» – запись «10». При этом коды «00» и «11» являются признаками искажения данных.

DS-кодирование [23] представляет собой разновидность применения корреляционных кодов и реализуется посредством двух параллельных каналов последовательной передачи кода. Если в основном канале *D* происходит изменение разряда с 0 на 1 или с 1 на 0, то в контрольном канале *S* не происходит смены значения. И наоборот, если в канале *D* нет изменения, то в канале *S* происходит изменение с 0 на 1 или с 1 на 0. Приемник информации контролирует отсутствие одновременной смены или неизменности значений разрядов в обоих каналах. Если это условие нарушается, фиксируется наличие ошибки передачи данных. Данный тип кодирования отличается минимальной задержкой обнаружения ошибки (один такт).

Код с простым повторением предусматривает повторную передачу кодовых слов, совпадение которых принимается в качестве подтверждения отсутствия ошибки. В противном случае принимается решение о наличии ошибок.

Инверсный код является модификацией кода с простым повторением, заключающейся в том, что в случае нечетного числа единиц в исходном слове к нему добавляется его инверсия. Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в основном слове. Если число единиц четное, то контрольные разряды принимаются без изменения, если нечетное, то контрольные разряды инвертируются. На втором этапе контрольные и информационные разряды суммируются по модулю два. Нулевая сумма говорит об отсутствии ошибок. При ненулевой сумме принятое слово бракуется.

Равновесные коды относятся к простейшим блочным кодам и представляют собой коды, разрешенные слова которых содержат фиксированное число единиц. Используются в основном при передаче данных по каналам связи.

Каскадное кодирование. Преимущества разных способов кодирования можно объединить, применив каскадное кодирование. При этом информация сначала кодируется одним кодом, а затем другим, в результате получается *код-произведение*.

К корректирующим кодам относятся, в частности, следующие.

Код Хэмминга – один из наиболее широко известных классов линейных кодов [17, 20, 24, 25]. Он конструируется следующим образом: разряды с номерами, представляющими собой степень двойки, являются контрольными, остальные – информационными. Обычно поступают таким образом: исходя из числа контрольных разрядов, определяют максимально возможное число информационных. Кодов Хэмминга с одним контрольным разрядом не существует, код Хэмминга с двумя контрольными разрядами содержит один информационный разряд и т. д. Результативность достигается путем многократной проверки принятой комбинации на четность. Каждая проверка должна охватывать часть информационных разрядов и один из избыточных. При передаче (записи в память) данных формируются и записываются значения контрольных разрядов, а при приеме (чтении) этих данных контрольные разряды вновь формируются и сопоставляются с исходными. Если все вновь вычисленные контрольные разряды совпадают с полученными, то сообщение не содержит ошибок. В противном случае выводится сообщение об ошибке, и, если представляется такая возможность, ошибка исправляется.

В *сверточных кодах* [26] формирование контрольных разрядов производится на основе нескольких информационных слов, а декодирование – на основе нескольких кодовых слов. Принцип работы сверточных кодов можно сопоставить с исправлением ошибок по смыслу. Сверточные коды называют также решетчатыми или трилистными.

В *циклических кодах* [17–19] в результате циклической перестановки любого кодового слова, содержащего как информационные, так и контрольные разряды, получается слово, принадлежащее этому же циклическому коду. При построении циклических кодов используют формализм операций над многочленами: многочлены, соответствующие принятым кодовым словам, должны делиться на их порождающий без остатка. Наличие остатка свидетельствует об ошибке. Если число ошибок не превышает расчетную величину, то остаток от деления зависит от конфигурации ошибок и может быть использован для их исправления. Использование циклических кодов позволяет упростить схемную реализацию кодирующих и декодирующих устройств путем использования сдвиговых регистров.

3. ПРОГРАММНЫЕ МЕТОДЫ КОНТРОЛЯ ВЫПОЛНЕНИЯ АЛГОРИТМА

К достоинствам программных средств относятся универсальность, гибкость и относительно невысокая стоимость. В то же время для их реализации требуются специфические пакеты программного обеспечения.

Контроль методом *многократного счета* предполагает, что решение соответствующей задачи выполняется два или более раз. В простейшем случае совпадение результатов считается признаком правильности решения задачи. При более глубоком подходе могут использоваться различные алгоритмы обработки результатов, включая варианты голосования (мажорированные оценки). Требуются дополнительные объемы памяти и времени счета.

Контроль по методу *усеченного алгоритма* призван снизить затраты многократного счета и применим в случаях, когда у решаемой задачи может существовать упрощенный (сокращенный) алгоритмический вариант, позволяющий сформировать пусть менее точный, но содержательно близкий результат. Такой вариант может не устраивать заказчика в качестве решения поставленной задачи, но вполне приемлем для оценки правильности функционирования полного алгоритма.

Метод проверки предельных значений[6]. Применим в задачах, допускающих априорное оценивание допустимых диапазонов искомого решения.

[6] Разновидность контроля по допуску на параметр [12].

Часто такие диапазоны определяются для отдельных контрольных точек (мест в последовательности действий, составляющих алгоритм). Этот метод можно считать вариантом усеченного алгоритма в случае, когда с его помощью вычисляются границы возможных решений.

Способ подстановки. Если проверяемый алгоритм осуществляет решение математических уравнений, то традиционно эффективным способом проверки решения является подстановка полученного решения в исходные уравнения. Приемлемая малость невязки (различия левых и правых частей) уравнений позволяет судить о правильности полученного решения. В отличие от многократного счета подстановка позволяет выявить систематические (внесенные при программировании) ошибки. Кроме того, подстановка, как правило, менее трудоемка по сравнению с многократным счетом.

Метод обратного счета. В некоторых задачах возможно определение исходных данных по полученному результату. Тогда проверка правильности счета может быть выполнена использованием данной возможности. При этом затраты иногда могут оказаться меньше затрат на прямой счет.

Проверка с помощью *дополнительных связей*. Такой метод предполагает возможность введения связей между различными параметрами основной задачи, описываемых точными или статистическими соотношениями. Вообще говоря, он является упрощенной версией аналитических методов: метода инвариантов и метода избыточных переменных, излагаемого в ч. 2 обзора.

Метод контрольных сумм заключается в суммировании всех слов обрабатываемого массива (команды, данные) с последующим сохранением суммы в определенной части массива. В интересах контроля выполняют повторное суммирование и сравнение с сохраненной контрольной суммой. Применяется в основном при передаче (загрузке) данных и программ.

Метод счета записей подразумевает подсчитывание и запоминание выполняемых записей, т. е. точно установленных наборов данных. В дальнейшем при работе с данными подсчет повторяется и результат сравнивается с первоначальным. Позволяет обнаружить потерю или пропуск обработки данных.

Контроль маркерных импульсов позволяет отследить прохождение вычислительным процессом определенных позиций или завершение счета в целом. Генерация соответствующих меток времени (маркеров) должна быть предусмотрена в реализуемом алгоритме. В случае нарушения предписанной последовательности маркеров или завышенного времени их ожидания счет прерывается и принимается решение о дальнейших действиях (повторное вычисление, использование резервных вариантов, останов задачи).

Специфическим частным случаем маркерного метода можно полагать многократную (от трех до пяти раз) отправку *запроса на получение данных*. Ошибка фиксируется при отсутствии отклика на все отправленные запросы. В случае получения ответа на любой из запросов процесс передачи[7] данных считается безотказным.

Контроль последовательности выполнения команд и программных модулей осуществляется путем разбиения программ на участки. Затем применятся один из способов:

- для каждого участка вычисляется свертка (путем счета числа операторов, методом сигнатурного анализа, использованием кодов), которая сравнивается с заранее рассчитанной;
- каждому участку присваивается определенное кодовое слово (ключ участка), которое записывается в выбранную ячейку ОЗУ перед началом выполнения участка и проверяется при окончании участка. Узлы разветвляющихся программ проверяются с помощью ключей, а циклических участков – числом повторений цикла.

Излагаемые далее методы диагностирования, в отличие от описанных выше эвристических, опираются на относительно более глубокий математический анализ диагностируемой системы.

4. ДИАГНОСТИРОВАНИЕ С РАЗБИЕНИЕМ НА КЛАССЫ

Математическую постановку задача контроля получила в работе [27], где осуществлялся поиск неисправностей в электрической схеме. Основным элементом постановки задачи явилась прямоугольная таблица неисправностей, содержащая S строк-признаков и D столбцов-состояний. Далее фиксировалось некоторое подмножество столбцов R_D. Если R_D – разбиение множества столбцов-состояний на классы[8], то формально задача заключалась в определении такого разбиения R_S на множестве строк-признаков, чтобы выполнялось взаимно однозначное соответствие

$$R_D \Leftrightarrow R_S. \qquad (1)$$

Содержание условия (1) интуитивно ясно: на элементах из множества D в виде теста строится хотя бы один элемент из R_D с последующим при-

[7] Достоверность данных контролируется отдельно.

[8] По определению классы либо не пересекаются, либо совпадают полностью.

своением ему, в силу импликации «тогда и только тогда», элемента из R_S.

Получили распространение различные модификации такого подхода [28].

В основе методов этого подхода лежит модель диагностируемого объекта в виде таблицы функций неисправности R_j^i (в общем виде она приведена ниже). Здесь D – множество технических состояний объекта; $d_0 \in D$ – его безотказное (исправное) состояние; $d_i \in D$, $i = \overline{1, n}$, – отказные (неисправные) состояния. Каждому неисправному состоянию соответствует определенный отказ (дефект) $s_i \in S$ и наоборот.

Функции неисправностей

R		D				
		d_0	…	d_i	…	d_n
S	s_1	R_1^0	…	R_1^i	…	R_1^n
	⋮	…	…	…	…	…
	s_j	R_j^0	…	R_j^i	…	R_j^n
	⋮	…	…	…	…	…
	s_k	R_k^0	…	R_k^i	…	R_k^n

В приведенной таблице S – множество признаков, $s_j \in S$, $j = \overline{1,k}$; R_j^i – функция неисправности, которая кратко записывается формулой

$$R_j^i = \Psi^i(s_j) \qquad (2)$$

и является аналитической, графической, табличной или другой формой представления поведения системы. Более подробная запись включает входные и выходные сигналы, перечни элементарных проверок, а также начальные условия, если объект динамический [28].

Общая запись (2) может быть детализирована в свете достижения желаемой глубины диагностирования. Часто явную запись (2) используют только для безотказного состояния R_j^0, относительно которого описывают отказные состояния.

При этом предполагается, что все отказы обладают свойствами обнаруживаемости и различения, т. е. все отказы могут быть однозначно определены и сепарированы от других отказов посредством некоторой совокупности элементарных проверок Π. В этом случае простым перебором элементарных проверок $\pi_k \in \Pi$ можно разбить таблицу неисправностей на непересекающиеся подмножества D_υ, $\upsilon = \overline{1, \lambda}$:

$$\bigcup_{\upsilon=1}^{\lambda} D_\upsilon = D, \quad D_\upsilon \cap D_\mu = \varnothing, \quad \upsilon \neq \mu. \qquad (3)$$

Таблицы отказов используются для построения как алгоритмов диагностирования, так и физической модели объекта, обеспечивающей реализацию схем диагностирования.

По одной и той же таблице функций неисправностей и при заданном разбиении множества D на подмножества D_υ можно построить в общем случае несколько полных неизбыточных тестов $T \subset \Pi$ (совокупностей элементарных проверок π). Содержание многих решений данного подхода связано с минимизацией числа элементарных проверок. Для этого используются различные инструменты: разделение задач диагностирования на прямые (определение технического состояния d_i по заданной элементарной проверке π_k) и обратные (определение множества проверок $\{\pi_k\}$, различающих заданную пару отказов d_m и d_n), построение деревьев отказов, расщепление входов и выходов и пр.

5. ДИАГНОСТИРОВАНИЕ НА ОСНОВЕ АЛГЕБРАИЧЕСКИХ ИНВАРИАНТОВ

Контроль с вычислением алгебраических инвариантов [28] относится к аналитическим методам, поскольку использует аналитическую информацию (математическое описание) о функционировании объекта контроля. Такой контроль заключается в проверке некоторых алгебраических соотношений (контрольных условий), которым должна удовлетворять совокупность выходных сигналов объекта, дополненная при необходимости одним или несколькими избыточными сигналами. Инвариантность контрольных условий состоит в том, что при отсутствии отказов они обязаны выполняться для любых входных сигналов и в любой момент времени.

Вкратце устройство диагностирования работает следующим образом. На устройство поступают входные U и выходные Y сигналы проверяемого объекта. На их основе вырабатываются вспомогательные сигналы Z, удовлетворяющие совместно с сигналами Y алгебраическому уравнению, разрешенному относительно переменной Δ, называемой синдромом:

$$\Delta = \Phi(Y, Z) = 0. \qquad (4)$$

Синдром инвариантен по отношению к вектору входных сигналов U. Если условие (4) нарушается, т. е. если $\Delta \neq 0$, то выносится суждение о возникновении отказа. Практически из-за наличия допустимых погрешностей измерений и вычислений контрольное условие (4) выполняется приближенно и диагностирование производится по неравенству $|\Delta| \leq \varepsilon$, где ε – допуск на выходной сигнал устройства диагностирования. Объекты, обладающие такими алгебраическими инвариантами, называются объектами с естественной избыточностью. Примерами здесь могут служить объекты, которые в безотказном состоянии обязаны двигаться по определенной, в частности фазовой, траектории (на сфере, в плоскости и пр.). Достоинством таких систем является минимум необходимой априорной информации об объекте и дополнительных средств диагностирования. Большинство же схем обнаружения отказов обладают искусственно созданной избыточностью.

Проанализируем чувствительность синдрома (4) к отказам. Пусть синдром представляет собой m-мерный вектор $\Delta = [\Delta_1 \ \cdots \ \Delta_m]^{\mathrm{T}}$, а возможные отказы объекта формализуются n-мерным вектором $F = [f_1 \ \cdots \ f_n]^{\mathrm{T}}$. При этом уравнение (4) следует переписать в виде

$$\Delta = \Phi(Y, Z, F) = 0. \qquad (5)$$

Очевидно, что для проявления реакции синдрома Δ на единичный отказ $f_i \neq 0$ достаточно выполнения условия ненулевой чувствительности

$$\exists j : \frac{\partial \Phi_j}{\partial f_i} \neq 0,$$

которое удовлетворяется конструктивными решениями. В случае же множественного отказа, когда одновременно присутствуют несколько отказов f_i, требуется выполнение дополнительного условия невозможности взаимной компенсации (5) эффектов этих отказов

$$\begin{bmatrix} \partial\Phi_1 / \partial f_1 & \cdots & \partial\Phi_1 / \partial f_n \\ \vdots & \ddots & \vdots \\ \partial\Phi_m / \partial f_1 & \cdots & \partial\Phi_m / \partial f_n \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \neq 0,$$

откуда видно, что при $m = n$ и максимальном ранге матрицы Якоби все одновременные формализованные отказы f_i обнаруживаются. Если же число m компонент Δ_j синдрома меньше числа n формализованных отказов f_i, возможна взаимная компенсация эффектов отказов и как результат – их пропуск.

ЗАКЛЮЧЕНИЕ

Предложен вариант классификации методов диагностирования и описаны содержание и особенности встроенных средств контроля. Рассмотрены инженерные эвристические методы мониторинга. Кратко охарактеризованы варианты мониторинга на основе схем голосования, правил достоверности, контрольных кодов, программных средств функционального контроля, таблиц неисправностей, а также алгебраических инвариантов. Во второй части обзора будут рассмотрены методы диагностирования, основанные на классическом моделировании отказов диагностируемой системы. В третьей части работы будут изложены методы диагностирования, основанные на нейронных сетях, нечетких и структурных моделях и моделях в виде множеств. Четвертая часть посвящена новым подходам к диагностированию и объединению различных моделей и методов.

ЛИТЕРАТУРА

1. *Буков В.Н., Бронников А.М., Агеев А.М.* и др. Концепция управляемой избыточности комплексов бортового оборудования // Науч. чтения по авиации, посвящ. пам. Н.Е. Жуковского: Матер. XVI Всерос. науч.-практ. конф. – Москва, 2019. – С. 17–33. [*Bukov, V. N., Bronnikov, A.M., Ageev, A.M.,* et al. Concept of controlled redundancy of onboard equipment complexes // Scientific readings on aviation, dedicated to the memory of N.E. Zhukovsky: Proc. of the XVI All-Russian scientific and practical conf. – Moscow, 2019. – P. 17–33. (In Russian)]
2. *Буков В.Н., Агеев А.М., Евгенов А.В., Шурман В.А.* Управление избыточностью технических систем. Супервизорный способ управления конфигурациями. – М.: ИНФРА-М, 2023. [*Bukov, V.N., Ageev, A.M., Evgenov, A.V., Shurman, V.A.* Redundancy management of technical systems. Supervisory method of configuration management. – M.: INFRA-M, 2023. (In Russian)]
3. *Большая* российская энциклопедия 2004–2017. – URL: https://old.bigenc.ru/economics/text/2227291?ysclid=lt6tp1tio4782244915 (дата обращения 29.02.2024). [*The Great* Russian Encyclopedia 2004–2017. – URL: https://old.bigenc.ru/economics/text/2227291?ysclid=lt6tp1tio4782244915 (accessed February 29, 2024). (In Russian)]
4. *Pouliezos, A.D., Stavrakakis, G.S.* Real time fault monitoring of industrial processes. – Dordrecht: Kluwer Acad. Publishers, 1994.
5. *DO-297*. Integrated modular avionics (IMA) development guidance and certification considerations. – Washington: RTCA Inc., 2005.
6. *Науменко А.П.* Теория и методы мониторинга и диагностики: Материалы лекций. – Омск: ОмГТУ, 2017. [*Naumenko, A.P.* Theory and methods of monitoring and diagnostics: Lecture materials. – Omsk: OmskGTU, 2017. (In Russian)]
7. *ГОСТ* Р 27.605–2013. Надежность в технике. Ремонтопригодность оборудования. Диагностическая проверка. – М.: Стандартинформ, 2014. – 28 с. [*GOST* R 27.605–2013. Reli-

ability in engineering. Equipment maintainability. Diagnostic testing. – Moscow: Standartinform, 2014. – 28 p. (In Russian)]
8. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. – Москва: Радио и связь, 1989. [*Sogomonian, E.S., Slabakov, E.V.* Self-checking devices and fault-tolerant systems. – Moscow: Radio and Communications, 1989. (In Russian)]
9. *ГОСТ Р* 56079-2014. Изделия авиационной техники. Безопасность полета, надежность, контролепригодность, эксплуатационная и ремонтная технологичность. Номенклатура показателей. – М.: Стандартинформ, 2014. [*GOST R 56079-2014*. Aircraft items. Flight safety, reliability, testability and maintainability indices. Indices nomenclature. – Moscow: Standartinform, 2014. (In Russian.)]
10. *Долбня Н.А.* Встроенные средства контроля бортовой вычислительной системы под управлением операционной системы реального времени как итеративный агрегированный объект // Вест. Самарского гос. аэрокосмич. ун-та. – 2012. – № 5 (36). – С. 224–228. [*Dolbnya, N.A*. Built-in means of control of the on-board computing system under the control of the real-time operating system as an iterative aggregated object // Bulletin of the Samara State Aerospace University. – 2012. – No. 5 (36). – P. 224–228. (In Russian)]
11. *Arikan, E.* Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels // IEEE Transactions on Information Theory. – 2009. – Vol. 55, no. 7. – P. 3051–3073.
12. *Бугаева А.А., Денисенко В.В*. Процесс тестирования, методы и тиры тестирования программного обеспечения // Синергия Наук. – 2022. – № 72. – С. 92–102. [*Bugaeva, A.A., Denisenko, V.V.* Protsess testirovaniya, metody i tiry testirovaniya programmnogo obespecheniya // Sinergiya Nauk. – 2022. – No. 72. – S. 92–102. (In Russian)]
13. *Диагностирование* и прогнозирование технического состояния авиационного оборудования: уч. пособ. для вузов гражд. авиации / под ред. И. М. Синдеева. – М.: Транспорт, 1984. [*Diagnostics* and forecasting of the technical condition of aviation equipment: teaching aid for civil aviation universities / Ed. by I. M. Sindeev. – Moscow: Transport, 1984. (In Russian)]
14. *Земляный Е.С., Тектов М.В.* Кворум-элемент для определения параметрического отказа курсовертикалей // Известия Тульского государственного университета. Технические науки. – 2023. – № 6. – С. 104–109. [*Zemlyanyy, E.S., Tektov, M.V.* Quorum element for determining the parametric failure of heading and vertical reference systems // Bulletin of Tula State University. Technical sciences. – 2023. – No. 6. – P. 104–109. (In Russian)]
15. *Савина М.Г., Мусонов В.М., Худоногов В.П., Сеславин В.С.* Применение кворум-элементов для контроля однотипных датчиков // Решетневские чтения. – 2013. – Т. 1. – С. 377–379. [*Savina, M.G., Musonov, V.M., Khudonogov, V.P., Seslavin, V.S.* Application of quorum elements for monitoring similar sensors // Reshetnev readings. – 2013. – Vol. 1. – P. 377–379. (In Russian)]
16. *Джанджгава Г.И., Дядищев А.В., Гарифов Р.Ш.* О концепции мониторинга технического состояния изделий авионики на основе применения средств и методов физической диагностики // Идеи и новации. – 2018. – Т. 6, № 3. – С. 64–68. [*Dzhandzhgava, G.I., Dyadishchev, A.V., Garifov, R.Sh.* On the concept of monitoring the technical condition of avionics products based on the use of physical diagnostics tools and methods // Ideas and Innovations. – 2018. – Vol. 6, No. 3. – P. 64–68. (In Russian)]
17. *Кудряшов Б.Д.* Основы теории кодирования: учеб. пособие. – СПб.: БХВ-Петербург, 2016. [*Kudryashov, B.D.* Fundamentals of coding theory: textbook. – St. Petersburg: BKhV-Petersburg, 2016. (In Russian)]
18. *Тельпухов Д.В., Деменева А.И., Жукова Т.Д., Хрущев Н.С.* Исследование и разработка систем автоматизированного проектирования схем функционального контроля комбинационных логических устройств // Электронная техника. Серия 3: Микроэлектроника. – 2018. – № 1 (169). – С. 15–22. [*Telpukhov, D.V., Demeneva, A.I., Zhukova, T.D., Khrushchev, N.S.* Research and development of automated design systems for functional control circuits of combinational logic devices // Electronic engineering. Series 3: Microelectronics. – 2018. – No. 1 (169). – P. 15–22. (In Russian)]
19. *Mytsko, E.A., Osokin, A.N., Malchukov, A.N.* CRC checksum. A study of CRC checksum calculation algorithms. – Saarbrucken: LAP LAMBERT Academic Publishing GmbH & Co KG, 2013. – 180 p.
20. *Павлов А.А., Царьков А.Н., Сорокин Д.Е.* и др. Анализ эффективности тестово-кодовой коррекции ошибок оперативных запоминающих устройств // Известия Института инженерной физики. – 2015. – № 4 (38). – С. 47–57. [*Pavlov, A.A., Tsarkov, A.N., Sorokin, D.E.,* et al. Analysis of the efficiency of test-code error correction of random access memory devices // Bulletin of the Institute of Engineering Physics. – 2015. – No. 4 (38). – P. 47–57. (In Russian)]
21. *Селиванов Е.П., Митин Д.Д.* Вариант методики построения помехоустойчивого кода Хемминга // Современные информационные технологии. – 2006. – № 3. – С. 34–37. [*Selivanov, E.P., Mitin, D.D.* A variant of the method for constructing a noise-resistant Hamming code // Modern information technologies. – 2006. – No. 3. – P. 34–37. (In Russian)]
22. *Павлов А.А., Царьков А.Н., Романенко А.Ю., Михеев А.А.* Метод коррекции ошибок в устройствах передачи и переработки информации систем телекоммуникаций при воздействии внешних и внутренних помех // Авиакосмическое приборостроение. – 2021. – № 8. – С. 13–24. [*Pavlov, A.A., Tsar'kov, A.N., Romanenko, A.Yu., Mikheev, A.A.* Metod korrektsii oshibok v ustroistvakh peredachi i pererabotki informatsii sistem telekommunikatsii pri vozdeistvii vneshnikh i vnutrennikh pomekh // Aviakosmicheskoe priborostroenie. – 2021. – No. 8. – S. 13–24. (In Russian)]
23. *ГОСТ* Р 70020-2022 Космическая техника. Интерфейсы и протоколы высокоскоростного межприборного информационного обмена и комплексирования бортовых систем космических аппаратов. SpaceWire-RUS. – М.: Российский институт стандартизации, 2022. – 118 с. [*GOST* R 70020-2022 Kosmicheskaya tekhnika. Interfeisy i protokoly vysokoskorostnogo mezhpribornogo informa-tsionnogo obmena i kompleksirovaniya bortovykh sistem kosmicheskikh apparatov. SpaceWire-RUS. – Moscow: Rossiiskii institut standartizatsii, 2022. – 118 s. (In Russian)].
24. *Asokan, R., Vijayakumar, T.* Design of extended hamming code technique encryption for audio signals by double code error prediction // Information Technology and Digital World. – 2021. – Vol. 3, no. 3. – P. 179–192.
25. *Bae, W., Han, J.W., Yoon, K.J.* In-memory hamming error-correcting code in mersister crossbar // IEEE Trans. on Electron. Devices. – 2022. – Vol. 69, no. 7. – P. 3700–3707.

26. *Журавлев В.Г., Куранова Н.Ю., Евсеева Ю.Ю.* Помехоустойчивые коды: учебное пособие. – Владимир: Изд-во ВлГУ, 2013. [*Zhuravlev, V.G., Kuranova, N.Yu., Evseeva, Yu.Yu.* Noise-immune codes: a tutorial. – Vladimir: VlSU Publishing House, 2013. (In Russian)]
27. *Чегис И.А., Яблонский С.В.* Логические способы контроля работы электрических схем // Тр. МИАН СССР. – 1958. – Т. 51. – С. 270–360. [*Chegis, I.A., Yablonsky, S.V.* Logical methods for monitoring the operation of electrical circuits // Proceedings of the Steklov Mathematical Institute of the USSR. – 1958. – Vol. 51. – P. 270–360. (In Russian)]
28. *Основы* технической диагностики. В 2-х кн. / Под ред. П. П. Пархоменко. – М.: Энергия, 1976. [*Fundamentals* of technical diagnostics. In 2 books. / Ed. by P.P. Parkhomenko. – M.: Energy, 1976. (In Russian)]

Статья представлена к публикации членом редколлегии В.Г. Лебедевым.

Поступила в редакцию 10.11.2024,
после доработки 26.03.2025.
Принята к публикации 10.04.2025.

Буков Валентин Николаевич – д-р техн. наук, АО «Бортовые аэронавигационные системы», г. Москва,
✉ v_bukov@mail.ru,
ORCID iD: https://orcid.org/0000-0002-5194-8251

Бронников Андрей Михайлович – д-р техн. наук, МГТУ им. Н.Э. Баумана, ✉ bronnikov_a_m@mail.ru,
ORCID iD: https://orcid.org/0009-0009-1216-3521

Попов Александр Сергеевич – канд. техн. наук, ВУНЦ ВВС «Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина», г. Воронеж, ✉ saga30@yandex.ru,

Шурман Владимир Александрович – АО «НИИ авиационного оборудования», г. Жуковский, ✉ shurman@niiao.ru

TECHNICAL CONDITION MONITORING METHODS TO MANAGE THE REDUNDANCY OF SYSTEMS. PART I: Built-in Control and Partition into Classes

V. N. Bukov*, A. M. Bronnikov**, A. S. Popov***, and V. A. Shurman****

*Airborne Aeronautical Systems, Moscow, Russia
**Bauman Moscow State Technical University, Moscow, Russia
***Zhukovsky–Gagarin Air Force Academy, Voronezh, Russia
****Institute of Aircraft Equipment, Zhukovsky, Russia

*✉ v_bukov@mail.ru, **✉ bronnikov_a_m@mail.ru, ***✉ saga30@yandex.ru, ****✉ shurman@niiao.ru

Abstract. Redundancy management of a technical system involves a monitoring procedure (control of the current state of its components) to reconfigure the system and improve the performance and autonomy of its application. This paper initiates a four-part survey of the state-of-the-art monitoring methods for redundancy management. Part I is mainly devoted to the analysis of voting schemes, fidelity rules, control codes, and program control, representing the most widespread monitoring methods in modern technical systems and built-in control. In addition, we examine long-known, albeit less common, monitoring methods: diagnosis with partition into classes and diagnosis based on algebraic invariants.

Keywords: technical condition monitoring, redundancy management, diagnosis, built-in control, control codes, partition into classes, algebraic invariants.

УДК 65.012+005

РЕШЕНИЕ СЛОЖНЫХ ЗАДАЧ УПРАВЛЕНИЯ РЕСУРСАМИ: ОТ КЛАССИЧЕСКОЙ ОПТИМИЗАЦИИ И ТЕОРИИ ИГР – К МУЛЬТИАГЕНТНЫМ ТЕХНОЛОГИЯМ ДЛЯ ПОИСКА КОНСЕНСУСА[#]

А. В. Леонидов*, П. О. Скобелев**

*Физический институт им. П. Н. Лебедева РАН, г. Москва;
Московский физико-технический институт, г. Долгопрудный, Московская область
**Самарский федеральный исследовательский центр РАН; Самарский государственный технический университет

*✉ leonidovav@lebedev.ru, **✉ p.skobelev@kg.ru

Аннотация. Рассмотрены вызовы и проблемы управления ресурсами предприятий и предлагаемые решения этих задач. Дан обзор существующих моделей, методов и средств планирования ресурсов предприятия, приведены новые требования к адаптивному многокритериальному управлению ресурсами в реальном времени. Обсуждена концепция автономных систем искусственного интеллекта (ИИ) для адаптивного управления ресурсами на основе мультиагентных технологий. Показана эволюция подхода к решению сложных задач управления ресурсами: от традиционной оптимизации одной целевой функции с игнорированием индивидуальных интересов участников – к теории игр с конкуренцией и кооперацией участников. Предложено развитие подхода для поиска и поддержания «конкурентного равновесия» – консенсуса между программными агентами участников решения задачи, обладающими собственными интересами, путем выявления конфликтов и проведения переговоров с взаимными уступками для их разрешения. Представлены базовая модель мультиагентной сети потребностей и возможностей на основе виртуального рынка и метод компенсаций для нахождения консенсуса для адаптивного управления ресурсами. Рассмотрены функциональность и архитектура интеллектуальных систем адаптивного управления ресурсами. Показаны результаты внедрения ИИ-решений на промышленных предприятиях и возможность повышения эффективности использования их ресурсов. Обсуждаются уроки, извлеченные из опыта разработок, и перспективы развития предлагаемого подхода.

Ключевые слова: управление ресурсами, сложность, искусственный интеллект, сети потребностей и возможностей, автономные системы, адаптивность, мультиагентные технологии, самоорганизация, экономика реального времени.

ВВЕДЕНИЕ

Растущая сложность выработки и реализации оптимальных решений в современной экономике во многом объясняется резким усложнением динамики спроса и предложения, когда разнообразные возмущающие события становятся скорее нормой, чем исключением [1], и связанной с этой необходимостью быстрой адаптации предприятий к меняющимся условиям осуществления экономической деятельности.

При этом растущая сложность в управлении предприятиями все чаще обусловлена увеличением числа и разнообразием целей и характерных свойств участников, вовлекаемых в процессы принятия согласованных решений и обладающих своими индивидуальными предпочтениями и ограничениями, например, в сложных международных

[#] Исследования Скобелева П. О. выполнены при поддержке Министерства науки и высшего образования Российской Федерации.

или национальных сетях поставок. К числу непредвиденных событий относят как масштабные события, связанные с появлением новых крупных заказчиков, партнеров или конкурентов, с разработкой новых продуктов и технологий или с изменениями в каналах поставок товаров, так и ежедневные события, например, поломки оборудования и задержки в исполнении операций.

Обычной реакцией менеджмента компаний на плохо предсказуемые события в бизнесе является привлечение дополнительных ресурсов, например, найм новых менеджеров, увеличение запасов товаров на складах и размеров самих складов. При этом в рамках традиционной системы принятия решений возрастает время реакции на возникающие события, включая коллективную выработку, согласование, принятие и исполнение решений. Как следствие, происходит снижение качества обслуживания клиентов, рост простоев в использовании ресурсов, теряются заказы или увеличиваются затраты, и, наконец, происходит общее снижение эффективности и конкурентоспособности бизнеса [2].

Одной из причин складывающейся ситуации является применение традиционных моделей, методов и средств планирования и оптимизации ресурсов, по-прежнему ориентированных на централизованное многоуровневое иерархическое управление предприятиями и пакетную обработку данных. При таком подходе учет важных для осуществляемой деятельности индивидуальных особенностей, предпочтений и ограничений участников процессов управления предприятиями затруднен, и они часто игнорируются.

Решение рассматриваемой проблемы требует разработки новой парадигмы создания максимально автономных интеллектуальных систем управления ресурсами, в которой не человек, а сама система принимает решения по текущему управлению ресурсами предприятия. Такая парадигма ориентирована на функционирование в рамках формирующейся сетевой экономики реального времени, для которой характерен высокий уровень автономности управления, что, в свою очередь, требует и высокой адаптивности управления ресурсами при возникновении разного рода непредвиденных событий [3].

В настоящее время становится возможным решить эту проблему с помощью систем искусственного интеллекта (ИИ), работающих непрерывно и способных автономно (самостоятельно) принимать решения, обеспечивая распределение ресурсов, планирование, оптимизацию, мониторинг и контроль результатов в режиме реального времени, а также адаптивно перестраивать планы по событиям.

Однако современные научно-исследовательские проекты в области ИИ-систем по-прежнему в основном сфокусированы только на автономных роботах, беспилотных летательных аппаратах и беспилотных транспортных средствах [4]. Реализуемые проекты в других областях технологий ИИ включают анализ больших и малых данных, распознавание образов и машинное зрение, машинное обучение и т. д. Технологии ИИ для управления ресурсами, как ни странно, пока еще не включаются в этот список, хотя задачи использования ИИ для автономного адаптивного управления с целью повышения эффективности предприятий при увеличении объемов заказов и разнообразия привлекаемых ресурсов являются весьма актуальными и значимыми.

В настоящей статье представлены теоретические основы и практические результаты решения сложных задач адаптивного управления ресурсами с использованием ИИ-систем, основанных на мультиагентной технологии, позволяющей создавать самоорганизующиеся расписания заказов и ресурсов, более открытые и гибкие к изменениям по сравнению с традиционными подходами.

В § 1 статьи исследуются причины растущей сложности и динамики современного управления производственными ресурсами. В § 2 дается краткий анализ ограничений существующих методов и средств планирования и оптимизации ресурсов, включая как классические и эвристические методы оптимизации, так и методы на основе теории игр. В § 3 рассматривается концепция автономной ИИ-системы для адаптивного управления ресурсами на основе понятия мультиагентной сети потребностей и возможностей (ПВ-сети) и виртуального рынка программных агентов заказов, операций, ресурсов и продуктов. Показывается, что решение сложной задачи управления ресурсами при этом может строиться через выявление и разрешение конфликтов с помощью аукционоподобных многоитерационных переговоров с использованием функций удовлетворенности, бонусов и штрафов агентов и метода компенсаций при взаимных уступках. Функциональность и архитектура решений автономного ИИ для адаптивного управления ресурсами представлены в § 4. Результаты промышленных внедрений ИИ представлены в § 5, где демонстрируется рост эффективности использования ресурсов предприятия. Уроки, извлеченные из разработки и внедрения промышленных систем, и преимущества этих систем для бизнеса показаны в § 6. В заключении формулируются основные выводы и

даются предложения по дальнейшим направлениям разработок рассматриваемых систем управления ресурсами.

1. СЛОЖНОСТЬ СОВРЕМЕННОГО УПРАВЛЕНИЯ РЕСУРСАМИ

Примеры современных задач управления ресурсами предприятий весьма разнообразны и могут включать управление флотилией грузовиков, цехами машиностроительных предприятий, цепочками поставок, движением поездов, группировками спутников и дронов и другие приложения.

Ряд примеров таких задач уже рассматривался в работах одного из авторов данной статьи [5]. Накопленный за прошедшее время опыт позволяет выделить их основные особенности и более точно сформулировать требования к применяемым подходам.

Ключевыми факторами сложности, типичными для этих задач, являются: большое количество ежедневных заказов, многокритериальное управление ресурсами (максимизация качества обслуживания, минимизация финансовых затрат и времени доставки, максимизация прибыли), индивидуальный подход к заказам и ресурсам и их многочисленные особенности (разделяемые заказы, повторно используемые ресурсы, возобновляемые ресурсы и т. д.), взаимозависимости между решаемыми задачами, специфика применяемых ресурсов, общие или разделяемые затраты, гибкие или фиксированные цены и т. д.

Одним из главных факторов сложности управления ресурсами является то, что на практике люди сталкиваются со множеством конфликтных требований, диктуемых множеством участников процессов ведения бизнеса, от стратегических целей предприятия в целом до тактических целей его подразделений, а также оперативных целей исполнителей «на земле»: водителей грузовиков, рабочих, логистов, диспетчеров, экономистов и других специалистов. По распространенному мнению опытных диспетчеров, «хорошее расписание – это хорошо сбалансированное расписание», учитывающее предпочтения и ограничения всех участников в каждой конкретной ситуации. Таким образом, ИИ-система должна формировать расписания, которые в каждой ситуации и в каждый конкретный момент времени отражают баланс многих конфликтных интересов, предпочтений и ограничений, что крайне сложно и трудоемко для решения традиционными подходами к планированию ресурсов.

Более того, такие расписания часто не являются «однородными» – разные фрагменты расписания различаются в зависимости от актуальных на конкретный момент времени критериев, которые могут меняться при поступлении новых заказов и других событий в процессе проведения вычислений. Подчеркнем, что достигнутый баланс интересов всегда зависит от развития ситуации, но относится к конкретному моменту времени, поэтому согласованное «оптимальное» расписание уже в следующий момент времени может не быть оптимальным и даже стать в принципе нереализуемым.

Такая «скользящая оптимизация», предполагающая в действительности гармонизацию интересов участников в каждой ситуации, в режиме реального времени требует интерактивного взаимодействия с лицами, принимающими решения, которые могут не только добавлять новые события, но и изменять свои предпочтения и ограничения, утверждать или отклонять решения и вносить встречные предложения.

В этой связи адаптивность должна рассматриваться как одна из наиболее важных функций таких решений, которая может быть определена как способность ИИ-системы частично перестраивать расписание, разрешая внутренние конфликты путем переговоров, без останова системы, и гибко маневрировать ресурсами для достижения поставленных целей в условиях неопределенности из-за постоянного возникновения событий, изменяющих ситуацию в заранее непредвиденные моменты времени.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ УПРАВЛЕНИЯ РЕСУРСАМИ

Хорошо известны традиционные пакетные методы и средства планирования и оптимизации ресурсов, основанные на линейном, динамическом программировании или программировании в ограничениях [6, 7].

Однако большинство этих методов и инструментов разработаны для постановки задачи, где все заказы и ресурсы известны заранее и не изменяются в режиме реального времени. Поэтому в области планирования ресурсов предприятий (ERP) классические пакетные планировщики, предлагаемые SAP, Oracle, Manugistic, i2, ILOG, J-Log и другими компаниями, по-прежнему доминируют на рынке, но на практике в этих пакетах, как правило, из-за роста размерности задачи реализуются в основном учетные функции, а встроенные модули распределения ресурсов, планирования,

оптимизации и коммуникации с участниками бизнеса имеют ограниченное применение.

Чтобы уменьшить сложность комбинаторного поиска, на практике применяются методы с использованием эвристических и метаэвристических правил, позволяющих принимать приемлемые решения в более разумное время благодаря сокращению области поиска решений [8, 9]:

– жадные алгоритмы локального поиска на основе эвристических правил предметной области;

– методы ИИ на основе нейронных сетей и нечеткой логики;

– метаэвристика: генетические алгоритмы, поиск по табу;

– моделирование, включая имитацию отжига и т. д.;

– стохастические методы, такие как метод Монте-Карло;

– алгоритмы оптимизации колоний муравьев и роя частиц;

– сочетание параллельных эвристических алгоритмов оптимизации и др.

Однако эти методы также используют пакетную обработку и не обеспечивают адаптацию расписаний в режиме реального времени по ходу поступления событий.

Анализ вышеуказанных решений позволяет выявить следующие проблемы:

– отсутствие моделей, методов и инструментов для адаптивного управления ресурсами;

– при изменении спецификаций задач требуется переработка применяемых методов и привлечение специалистов для перепрограммирования системы;

– имеющиеся системы поддерживают централизованное управление на основе команд сверху вниз, без учета мнений и интересов, предпочтений и ограничений исполнителей;

– иерархическая жесткость систем не позволяет оперативно и гибко реагировать на события, лишь частично перестраивая расписания;

– внутренняя пассивность системы и функционирование в пакетном режиме только по запросу пользователя;

– сосредоточение на данных, а не на корпоративных предметных знаниях, необходимых для автоматического принятия решений;

– избыточная стандартизация бизнес-процессов, игнорирующая индивидуальные предпочтения и ограничения лиц, принимающих решения.

Высокая сложность и динамика рассматриваемых задач приводят к тому, что традиционные централизованные иерархически организованные последовательные методы и алгоритмы комбинаторного поиска или эвристики не позволяют эффективно решить проблему адаптивного управления ресурсами с приемлемым качеством и в требуемое время, что сдерживает внедрение рассматриваемых ИИ-систем управления предприятиями на практике.

3. НОВЫЕ МОДЕЛИ И МЕТОДЫ ПОИСКА КОНСЕНСУСА ДЛЯ АДАПТИВНОГО УПРАВЛЕНИЯ РЕСУРСАМИ

Одним из ключевых направлений в области ИИ являются мультиагентные технологии; информацию о них можно найти в монографиях [10, 11] и статьях [12, 13].

В последнее время мультиагентные технологии часто связывают с ИИ-агентами и LLM-моделями (*Large Language Models*), но ключевым свойством мультиагентных технологий по-прежнему остается возможность создания самоорганизующихся систем, где каждый элемент сам принимает решения, что делает такие системы более открытыми к изменениям, гибкими и эффективными для решения различных сложных задач.

В этой связи мультиагентные технологии являются одним из возможных методов решения оптимизационных задач [14]. В последнее десятилетие на их основе были разработаны новые модели и методы распределенного решения задач планирования и оптимизации ресурсов. Описание таких моделей и ссылки на соответствующую литературу можно найти в обзорах [15–20].

Отметим, что переход к мультиагентной технологии адаптивного построения локально оптимального расписания отражает существенное изменение парадигмы, в которой рассматривается задача, по сравнению с подходом, использующим стандартные пакетные оптимизационные технологии, в которых решение строится централизованным последовательным детерминированным алгоритмом. Расписание в мультиагентном подходе, напротив, является распределенным и динамическим объектом, в котором решение задачи планирования строится недетерминированным путем, поскольку предполагает наличие параллельных и асинхронных процессов вычислений, развивающихся над общей структурой данных, зеркально отражающих состояние ресурсов предприятия в любой момент времени. При этом каждое событие инициирует переходный процесс из одного неравновесного состояния в другое, что реализуется путем частичного адаптивного перестроения расписания заказов и ресурсов, т. е. допускается пересмотр принятых ранее решений и перераспределение ранее распределенных заказов по ресурсам.

Таким образом, возникает задача оперативного перестроения расписания за ограниченное время, которое определяет характеристики целевого пространства состояний системы, достижимых в течение заданного интервала времени из заданного начального состояния в рамках рассматриваемого метода.

Идея использования в задачах управления ресурсами моделей и методов, основанных на самоорганизации агентов, выглядит для разработчиков программного обеспечения весьма привлекательной. Многие полезные свойства таких алгоритмов уже хорошо изучены: они интуитивно понятны, могут учитывать индивидуальные критерии, предпочтения и ограничения всех участников, достоверно корректны, естественно распараллеливаются, подходят для развертывания в распределенных системах, в большом числе случаев являются устойчивыми к изменениям спецификации задачи и т. д. Особый интерес представляет систематическое сравнение результатов мультиагентного и пакетного оптимизационного подходов, приведенное, например, в работах [21, 22], которое необходимо для «разметки» эффективности мультиагентных алгоритмов в зависимости от характерных режимов задачи.

В целом, архитектура мультиагентных моделей распределенной оптимизации делится на два больших класса: модели с автономными агентами и модели с дополнительным участием агентов-посредников. Ключевым элементом мультиагентной технологии является протокол переговоров, обеспечивающий запуск и развитие процесса достижения договоренности между программными агентами спроса (например, необходимые действия) и предложения (ресурсы) соответственно. В моделях с автономными агентами соответствующие агенты действуют независимо, а в моделях с участием посредников возможно их ограниченное управляющее вмешательство.

В большинстве работ используются различные версии протокола контрактной сети (англ. *Contract Net Protocol*) [23, 24], регулирующего процесс подачи и рассмотрения заявок. Обсуждение таких протоколов и их сравнительный анализ представлены в работе [25].

Реализация протокола установления баланса спроса и предложения ведется с использованием рыночного ценового механизма, подразумевающего наличие внутренних виртуальных денег. Тем самым в мультиагентных моделях реализуется концепция виртуального рынка (ВР), на котором агенты ведут итерационные переговоры, заключая и пересматривая контракты между собой, обмениваясь задачами, а также деньгами. Поиск решения каждый агент начинает с некоторого начального набора задач, возможно, пустого, а затем вступает в процесс согласования новых решений. Важной составляющей поиска решения является совместное рассмотрение планирования и составления расписания. Изучению этого вопроса посвящен обзор [16] и цитируемая в нем литература. В результате поиск оптимального расписания выполняется в рамках процесса динамической самоорганизации сети агентов, конечной целью которого для моделей с автономными агентами с общетеоретической точки зрения является достижение состояния «конкурентного равновесия» (консенсуса), когда ни один из агентов более не может улучшить результат для системы в целом. Как уже отмечалось выше, ключевым фактором, определяющим качество полученного решения, является конечность времени, отпущенного на его принятие и, как следствие, возможное отличие выработанного за это время решения от оптимального.

Реализуемая в мультиагентных моделях концепция виртуального рынка естественно вписывается в общую концепцию формулировки задач оптимизации в терминах виртуальной экономики взаимодействующих агентов [26–28] и, в частности, в рамках теории игр [29–32].

Для моделей автономных агентов фундаментом общего анализа возможных исходов их взаимодействия является теория игр и проводимый в ее рамках анализ теоретико-игровых равновесий Нэша в мультиагентных системах, а также алгоритмов их нахождения.

Для задачи построения расписания анализ равновесий Нэша рассматривался, в частности, в работах [19, 33, 34]. Основным результатом является формальное доказательство того факта, что нахождение равновесий Нэша в подобных задачах является NP-трудным. Заметим, что рассматриваемые задачи не являются в этом смысле чем-то исключительным: поиск равновесия Нэша в чистых стратегиях за исключением узкого класса специальных задач относится как минимум к классу PPAD (*Polynomial Parity Argument, Directed*) [11]. На качественном уровне это означает, что время построения решения экспоненциально по параметру (параметрам), отражающим гетерогенность системы. В условиях ограниченного времени на построение измененного расписания его может не хватить для выявления соответствующего равновесия Нэша.

Понятно, что NP-трудный характер теоретико-игровых равновесий подчеркивает значимость временны́х ограничений, имеющихся при построении расписания. Важным обстоятельством, которое потенциально имеет решающий характер для

классификации соответствующих режимов, является наличие фазового перехода по вычислительной цене решения NP-трудных задач, в частности, задач построения расписания [35, 36], разделяющего фазы с легко и трудно находимыми решениями.

Что касается алгоритмов нахождения теоретико-игрового равновесия, то ситуация осложняется тем, что на сегодняшний день *универсальных* алгоритмов такого рода не существует. Например, для одного из наиболее естественных и привлекательных алгоритмов поиска равновесия, основанного на теории аукционов [11, 37, 38], продемонстрировано наличие конфигураций, в которых решение не может быть найдено [39]. Известно также, что в некоторых случаях конкурентные мультиагентные модели не находят удовлетворительного решения и попадают в ловушку (*deadlock*) [40]. Попадание в ловушку отражает недостаточность того или иного протокола, используемого в конкурентных мультиагентных системах, для разрешения конфликтов. В этой связи существенный интерес представляют мультиагентные системы с посредниками (*mediators*) [17, 18]. Общая конструкция мультиагентной архитектуры с посредниками описана в работе [41]. В ней, кроме базового конкурентного слоя агентов, вводятся дополнительные агенты-посредники, к которым конкурентные агенты могут обратиться для разрешения конфликтов. В отличие от автономных агентов, агентам-посредникам доступен существенно больший объем информации, позволяющей точнее планировать динамическое перестроение расписания. Сравнение конкурентной архитектуры и архитектуры с посредниками, проведенное в работах [42–44], показало, что введение посредников может улучшить контрольные показатели качества решения. Детальное описание различных мультиагентных моделей с посредниками дано в статьях [40, 44–48].

Существенным вопросом построения архитектуры мультиагентных систем является анализ иерархических и холонических архитектур, что обсуждается в работах [49–51].

С теоретической точки зрения, кроме описанных выше архитектур взаимодействия агентов, существенный интерес представляют также кооперативные модели, в которых протокол взаимодействия агентов описывается в рамках кооперативной теории игр [11, 41, 44]. Этот интерес связан, в частности, с тем обстоятельством, что, за исключением специальных случаев, конкурентное равновесие в теории игр неэффективно с точки зрения общего качества решения. В рассматриваемом контексте эта проблема отражена в обзоре [47].

В монографии [5] описаны проекты, в которых начиная с 1999 г. был развит аналогичный подход к разработке программного обеспечения для реализации мультиагентных решений оптимизационных задач. В частности, привлекательные свойства таких алгоритмов проявились уже в первом мультиагентном прототипе системы для завода Volkswagen, в которой решалась задача поставки и замены деревянных деталей для оформления салона автомобилей класса «люкс». Проблема состояла в том, что уже готовый к отправке дорогой автомобиль часто не проходил контроль качества из-за того, что цвет или рисунок на деревянных частях салона не отвечали стандарту. Такой автомобиль отгонялся на стоянку и требовалось много времени, чтобы найти, доставить и установить замену, что приводило к существенным потерям в экономике. При этом производственная система SAP требовала от 12 до 24 часов для решения проблемы, и на практике руководители подразделений завода просто начинали звонить друг другу и решать вопрос путем переговоров. Необходимо было разработать систему, которая быстро и адаптивно перестраивала бы расписание, используя данные из SAP. Разработанная мультиагентная система позволяла в течение нескольких секунд (до минуты) решать указанную проблему.

В следующий период разработанная мультиагентная технология была усовершенствована в соответствии с концепцией холонических систем, в которой были реализованы базовые агенты продуктов, ресурсов, заказов и штабного агента (или агента системы в целом) в соответствии с референтной архитектурой PROSA [48]. В дальнейшем в разработанной технологии был сделан важный шаг детализации агентов до уровня не только агентов бизнес-процессов, но каждой отдельной задачи, и представлены классы и роли агентов, которые формируют мультиагентные сети потребностей и возможностей (ПВ-сети), представляющие самоорганизующиеся расписания с проактивностью и выплатой взаимных компенсаций при разборе конфликтов. Для агентов ПВ-сетей был предложен метод адаптивного принятия решений с выплатой компенсаций при взаимных уступках заказов и ресурсов на виртуальном рынке, основанный на функциях удовлетворенности и бонусов-штрафов, удобных для обеспечения эластичности в принятии решений при разрешении конфликтов и формировании нового консенсуса агентов ПВ-сети [52-54].

В разработанном методе агенты заказов и ресурсов, задач и продуктов сначала выбирают лучшие бесконфликтные варианты, а затем разрешают

конфликты до тех пор, пока система не будет сбалансирована до состояния нового консенсуса, и ни один из новых вариантов не сможет улучшить общую целевую функцию системы, например, прибыль.

Этот процесс отражает имеющуюся практику опытных менеджеров и диспетчеров, которые формируют сложные расписания, разрешая конфликты и находя баланс конфликтующих интересов всех сторон, участвующих в принятии решений.

Формализованная постановка задачи и описание разработанного метода приведены в работе [55].

В последнее время интерес к ИИ-системам для управления предприятиями существенно вырос благодаря массовому внедрению электронных карт, ERP-систем и интернета вещей, сотовым телефонам и другим устройствам, переводящим бизнес в цифровую модель, отражающую состояние ресурсов в реальном времени [56–59]. При этом ИИ-возможности связываются в основном с прогнозированием, планированием и извлечением знаний в ходе обучения в комбинации с классическим планированием и оптимизацией ресурсов и различными эвристиками. Задача формирования динамического самоорганизующегося расписания, оперативно, гибко и эффективно перестраиваемого по событиям в реальном времени, пока не ставилась.

4. ФУНКЦИОНАЛЬНОСТЬ И АРХИТЕКТУРА РЕШЕНИЯ

Функциональность разрабатываемых ИИ-систем для адаптивного управления ресурсами направлена на поддержку полного цикла автономного управления ресурсами, включающего:

- сбор новых событий с помощью датчиков, внешних систем и мобильных устройств;
- распределение заказов по ресурсам путем выявления наиболее подходящих;
- планирование заказов и ресурсов – вычисление наилучшей возможной последовательности и определение времени начала и окончания задачи (операции) для выполнения заказов;
- оптимизация заказов и ресурсов (если есть время) – постоянный процесс улучшения целевых функций всех агентов, участвующих в управлении ресурсами;
- прогнозирование новых событий (новых заказов или отказов), которые будут обрабатываться как виртуальные события, для предварительного динамического резервирования критических ресурсов;
- онлайн-общение с пользователями: утверждение системных рекомендаций, изменение предпочтений или предоставление встречных предложений, исправление фактов и т. д.;
- мониторинг и контроль выполнения плана – сравнение запланированных и фактических результатов, выявление пробелов и запуск события перепланирования для высшего руководства;
- адаптивное перепланирование в случае растущего разрыва между планом и реальностью, например, если пользователь игнорирует рекомендации и выходит из заданных временны́х рамок;
- обучение на основе опыта – кластеризация событий, сравнение планового и фактического времени выполнения задач, например, для анализа производительности труда работников;
- моделирование в режиме реального времени «что, если» – параллельно с основной траекторией выполнения плана в режиме реального времени могут выполняться несколько линий моделирования для изучения будущего;
- эволюционная перестройка бизнес-сети – генерация предложений о том, как улучшить качество и эффективность операций (выбрать лучшее место для хранения и т. д.).

Разработанный подход может быть обобщен в виде концепта автономной системы для интеллектуального управления ресурсами в реальном времени ИСУР-РВ [60, 61], где выделяются следующие виды пользователей (рис. 1):

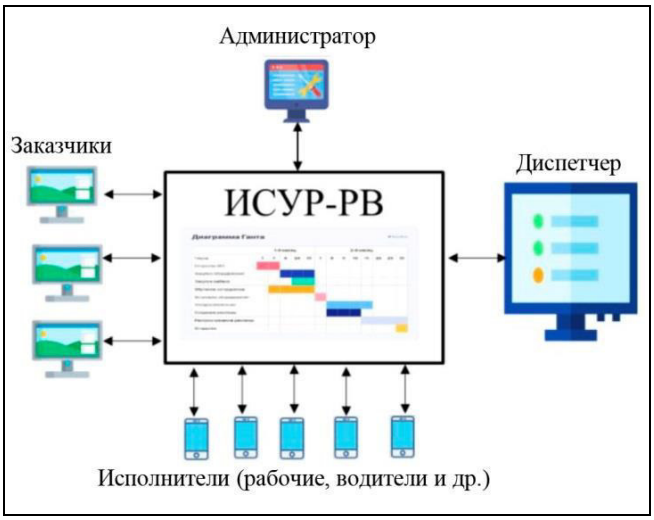

Рис. 1. Концепт автономной интеллектуальной системы управления ресурсами в реальном времени

– заказчики – задают требуемые заказы, согласовывают получаемые предложения и далее видят пошаговое исполнение своих заказов;

– диспетчеры – задают критерии планирования и утверждают построенные системой планы, корректируют результаты и решают оставшиеся проблемы;

– исполнители – получают сменные задания и делают отметки их исполнения при необходимости, а также вводят непредвиденные события, которые вызывают адаптивное изменение планов.

– администраторы – выдают логины и пароли для авторизации пользователей, сохраняют базы данных системы и т. д.

Основные виды пользователей системы ИСУР-РВ и их возможности могут изменяться в зависимости от ее применения, но указанный базовый функционал сохраняется для различных систем.

В архитектуре ИСУР-РВ выделяются следующие основные компоненты (рис. 2):

– веб-системы пользователей – предназначены для поддержки бизнес-процессов работы пользователей;

– база знаний на основе онтологий – содержит формализованные знания, представленные в виде классов понятий и отношений, для поддержки принятия решений в реальном времени;

– онто-МАС – онтологически настраиваемая мультиагентная система управления ресурсами в реальном времени;

– интеграция – модули интеграции с традиционными учетными системами (1С и др.).

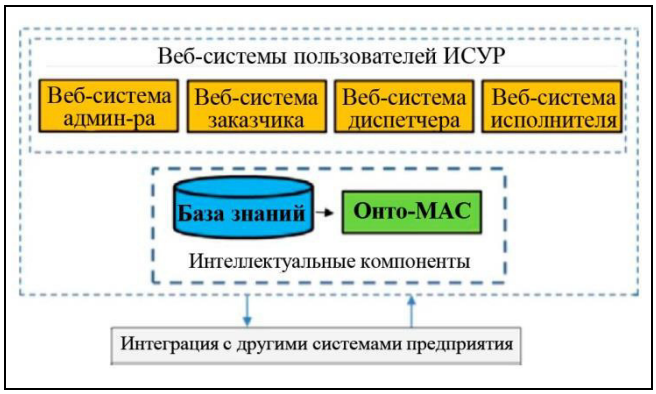

Рис. 2. Основные компоненты автономной интеллектуальной системы управления ресурсами в реальном времени

Принятые решения в виде текущих планов, указаний или команд передаются на мобильные телефоны исполнителей или оборудование предприятия с запросом принятия или подтверждения, но при изменении ситуации в любой момент времени могут быть адаптивно пересмотрены по генерируемым системой или вводимым пользователями событиям.

5. РЕЗУЛЬТАТЫ ПРОМЫШЛЕННЫХ ВНЕДРЕНИЙ

На основе представленного подхода в период с 2000 по 2008 г. разработано 15 промышленных прототипов и полномасштабных мультиагентных систем для адаптивного управления ресурсами, в том числе по управлению танкерами, корпоративным такси, грузовыми перевозками с консолидациями, и довольно много различных прототипов и малых приложений (адаптация плана питания или плана тренировок, заказа товаров для дома и т. д.).

В ходе разработки и внедрения рассматриваемых систем была выработана методика оценки повышения эффективности использования ресурсов, которая оценивает два типа расходов:

• Прямые расходы: сокращается время перевозки грузов или исполнения производственных заказов, использование материалов, машин и станков, уменьшается объем оплаты труда рабочих и т. д.

• Накладные расходы: сокращается штат управленцев – менеджеров нижнего звена, логистов, диспетчеров, экономистов и т. д.

При расчете экономического эффекта также учитывается сокращение сложности и трудоемкости выполнения управленческих операций, сокращение времени на обработку непредвиденных событий, снижение затрат на обучение персонала и т. п.

Более подробно постановка задач и результаты внедрений представлены в работе [5], а в настоящей статье кратко перечислены основные результирующие преимущества для бизнеса:

– увеличение числа выполненных заказов при том же или сокращенном объеме ресурсов;

– сокращение времени исполнения заказов;

– сокращение простоя на каждый ресурс в год;

– повышение эффективности использования ресурсов;

– знания предметной области, используемые в принятии решений, собраны, формализованы и систематизированы;

– сокращение пеней и штрафов за задержку исполнения заказов;

– снижение сложности и трудоемкости работы для диспетчеров, менеджеров, логистов и экономистов;

– снижение затрат на обучение управленческого персонала.

Указанные преимущества в среднем обеспечивают возврат инвестиций на разработку рассмат-

риваемых систем в срок от трех месяцев до одного года.

Часть разработанных решений использовалась в качестве инструментов моделирования и поддержки принятия решений, но большая часть получила полномасштабное внедрение и работает до настоящего времени.

На следующем этапе в 2009–2024 гг. разработанный подход был существенно развит и его применение расширилось с производственных и транспортных предприятий на новые сферы управления, в частности на управление железнодорожным движением пассажирских и грузовых поездов, управление группировками спутников, управление цепочками поставок напитков, управление распределением вагонов для перевозки угля и рядом других видов ресурсов.

Выделим дополнительные преимущества для бизнеса, которые были выявлены на этом этапе, подробно они представлены в работах [62–64]:

– сокращение затрат на выполнение заказов;

– сокращение численности управленческого персонала;

– повышение оперативности и гибкости принятия решений;

– возможность моделирования развития бизнеса одновременно с оперативным управлением.

На первом этапе внедрения в базу знаний вносится довольно много дополнений, которые выявляются лишь в ходе сопоставления результатов планирования ресурсов системой с работой практиков. Когда же качество принимаемых решений системой превосходит 50 %, по сравнению с человеком, т. е. ИИ-система начинает принимать больше правильных решений, чем опытные пользователи, можно говорить о начале перехода к автономному ИИ для «безлюдного» управления предприятиями.

Главным результатом этого периода стало более органичное объединение возможностей адаптивного планирования и оптимизации ресурсов с мониторингом и контролем исполнения заказов, что позволило создавать «цифровых двойников» подразделений предприятий, работающих параллельно и асинхронно с предприятиями и синхронизируемых с ними по событиям в реальном времени.

И пусть пока еще окончательные управленческие решения предлагаются пользователям на согласование и утверждение, но уже видна растущая тенденция постепенного перехода к автономным системам, предназначенным для указанного выше безлюдного управления.

В среднем доказанный теоретически и подтвержденный эффект от внедрения рассматриваемых систем может достигать 15–40 % [62], позволяя предприятиям с помощью одного и того же объема производственных ресурсов выполнять больше заказов, т. е. существенно повышать их эффективность.

6. ИЗВЛЕЧЕННЫЕ УРОКИ И ОСНОВНЫЕ ПРЕИМУЩЕСТВА

В результате проведенного анализа был выявлен ряд проблем, возникающих при практическом внедрении ИИ-систем управления предприятиями:

• Разработка рассматриваемых ИИ-систем требует участия высококвалифицированных экспертов и программистов, занимает много времени, требует продолжительного тестирования и т. д.

• Разработка самоорганизующихся решений для бизнес-пользователей является сложной задачей:

– часто трудно оценить, насколько полученный системой результат далек от «оптимального» решения;

– результаты зависят от истории возникновения событий;

– «эффект бабочки»: небольшие изменения приводят к неожиданно большой реакции;

– реакция системы может замедлиться в случае перехода между состояниями равновесия;

– в случае перезапуска системы результат планирования может быть другим;

– взаимодействие с пользователями в режиме реального времени становится более сложным и динамичным;

– решение иногда трудно объяснить пользователю (потеря причинности) и т. д.

• Управление корпоративными ресурсами является критически важным для бизнеса, поэтому данная область все еще очень консервативна в принятии новых решений ИИ.

• Большая часть корпоративных знаний для принятия решений обычно не осознается и скрыта в головах экспертов – выявление и формализация этих знаний требуют прямой связи с диспетчерами, инженерами, рабочими, водителями и т. д.

• Значительная часть усилий связана с разработкой сетевых пользовательских интерфейсов, которые должны быть настраиваемыми и недорогими.

• Дальнейшее развитие для более широкого круга малых и средних предприятий видится как разработка цифровых платформ SaaS (*Software as as Service*) для экосистемы услуг и дополнительных решений, которые могут быть интегрированы с существующими системами.

На практике эти трудности поддаются решению, но требуют специальных инструментов для первоначального анализа данных клиента и интеграции с обычно сильно устаревшими системами, данные в которых не всегда актуальны и корректны.

Обсуждаемые трудности компенсируются преимуществами предлагаемых ИСУР-РВ, так как они:

– повышают эффективность использования ресурсов путем перехода к принятию решений в режиме реального времени;

– решают сложные задачи планирования, заменяя комбинаторный поиск анализом конфликтов и поиском компромиссов;

– обеспечивают адаптивное перепланирование с быстрой реакцией на события;

– предлагают индивидуальный подход к каждому заказу, задаче, продукту и ресурсу;

– поддерживают активное двустороннее взаимодействие с пользователями для скоординированной командной работы;

– снижают роль человеческого фактора в принятии решений;

– уменьшают затраты на разработку благодаря повторному использованию кода в новых приложениях;

– позволяют моделировать сценарий «если – то» и прогноз для улучшения решений;

– создают новую цифровую платформу для поддержки роста бизнеса без пропорционального роста численности управленческого персонала.

Результаты разработок могут быть применены для решения широкого спектра задач управления ресурсами в рамках конйцепций Industry 5.0 и Society 5.0, ориентированных на цифровизацию знаний и переход к системам автономного коллективного интеллекта [65].

ЗАКЛЮЧЕНИЕ

Новый класс автономных интеллектуальных систем для безлюдного управления ресурсами предприятия открывает новые возможности для роста эффективности управления бизнесом, повышая удовлетворенность клиентов, делая бизнес более гибким, снижая себестоимость исполнения заказов, сокращая время исполнения заказов и риски.

Временные ограничения на выработку оптимальных решений требуют теоретического осмысления и пересмотра существующих подходов. Фактически речь идет о разработке новой методики «направляемой самоорганизации» и «умной оптимизации» для выработки квазиоптимальных решений экспоненциально трудных задач с учетом таких ограничений, в рамках которых система сама оценивает результаты и принимает решения о завершении вычислений или о том, какие ветви оптимизации дополнительно следует исследовать.

Разработанные промышленные приложения доказывают, что мультиагентная технология обеспечивает возможность решения широкого круга задач управления ресурсами в условиях высокой неопределенности, сложности и динамики. Адаптивность управления ресурсами помогает повысить эффективность бизнеса, сократить время отклика и улучшить качество обслуживания новых заказов, а также повысить коэффициент полезного использования ресурсов.

В качестве следующего шага предполагается создание цифровой сетецентрической платформы и экосистемы цифровых двойников предприятий для решения сложных многоуровневых задач управления ресурсами крупных промышленных предприятий, транспортных и сервисных компаний и т. д.

Дальнейшие работы, как предполагается, будут сочетать адаптивное планирование с обучением на основе опыта с помощью нейронных сетей и взаимодействия с пользователями на основе LLM-моделей как для формирования базы знаний предприятия, так и для диалога с пользователями на естественном языке, способных также объяснять и согласовывать принятые решения.

ЛИТЕРАТУРА

1. *Capitalizing* on Complexity? Insights from the Global Chief Executive Officer Study. – USA: IBM, 2010. – 70 p. – URL: http://www-935.ibm.com/services/us/ceo/ceostudy2010/index.html (дата обращения 04.01.2025). [Accessed January 4, 2025].
2. *Skobelev, P., Trentesaux, D*. Disruptions Are the Norm: Cyber-Physical Multi-Agent Systems for Autonomous Real Time Resource Management / In: Service Orientation in Holonic and Multi-agent Manufacturing, series “Studies in Computational Intelligence”. Ed. by T. Borangiu, D. Trentesaux, A. Thomas, et al. – Vol. 694. – Switzerland: Springer, 2017. – P. 287–294.
3. *GARTNER*. Top Strategic Predictions for 2016 and Beyond: The Future Is a Digital Thing. – Stamford: Gartner, Inc., 2015. – URL: https://www.gartner.com/en/documents/3142020 (дата обращения 05.01.2025). [Accessed January 5, 2025].
4. *Perez, J.A., Deligianni, F., Ravi, D., Yang, G.-Z*. Artificial Intelligence and Robotics. UK-RAS White Paper. – London: Imperial College, 2017. – 48 p. – URL: https://arxiv.org/

pdf/1803.10813 (дата обращения 05.01.2025). [Accessed January 5, 2025].
5. *Rzevski, G., Skobelev, P*. Managing Complexity. – London-Boston: WIT Press, 2014. – 216 p.
6. *Handbook* of Scheduling: Algorithms, Models and Performance Analysis / Ed. by J. Y.-T. Leung. – London-New York: Chapman & Hall/CRC, 2004. – 1224 p.
7. *Vos,* S. Meta-heuristics: The State of the Art. Local Search for Planning and Scheduling / In: Lecture Notes in Computer Science. Ed. by A. Nareyek. – Berlin: Springer-Verlag, 2001. – Vol. 2148. – P. 1–23.
8. *Binitha, S., Sathya, S.S*. A Survey of Bio inspired Optimization Algorithms // International Journal of Soft Computing and Engineering. – 2012. – Vol. 2, iss. 2. – P. 137–151.
9. *Handbook* of Constraint Programming / Ed. by F. Rossi, P. Van Beek, T. Walsh. – Amsterdam: Elsevier, 2006. – 978 p.
10. *Wooldridge, M.* An Introduction to Multi-Agent Systems. – Hoboken: John Wiley & Sons, 2009. – 488 p.
11. *Shoham, Y., Leyton-Brown, K*. Multi-agent Systems: Algorithmic, Game Theoretic and Logical Foundations. – Cambridge: Cambridge Univ. Press, 2009. – 483 p.
12. *Словохотов Ю.Л., Новиков Д.А.* Распределенный интеллект мультиагентных систем. Ч. 1. Основные характеристики и простейшие формы // Проблемы управления. – 2023. – № 5. – С. 3–22. [*Slovokhotov, Yu.L., Novikov, D.A*. Distributed Intelligence of Multi-Agent Systems. Part I: Basic Features and Simple Forms // Control Sciences. – 2023. – No. 5. – P. 2–17.]
13. *Словохотов Ю.Л., Новиков Д.А.* Распределенный интеллект мультиагентных систем. Ч. 2. Коллективный интеллект социальных систем // Проблемы управления. – 2023. – № 6. – С. 3–21. [*Slovokhotov, Yu.L., Novikov, D.A*. Distributed Intelligence of Multi-Agent Systems. Part II: Collective Intelligence of Social Systems // Control Sciences. – 2023. – No. 6. – P. 2–17.]
14. *Davidsson, P., Persson, J., Holmgren, J*. On the Integration of Agent-Based and Mathematical Optimization Techniques / In: Agent and Multi-Agent Systems: Technologies and Applications. KES-AMSTA 2007. Lecture Notes in Computer Science. Ed. by N.T. Nguyen, A. Grzech, R.G. Howlett, L.C. Jain. – Berlin, Heidelberg: Springer, 2007. – Vol. 4496. – P. 1–10.
15. *Shen, W., Norrie, D.H*. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey // Knowledge and Information Systems. – 1999. – Vol. 1. – P. 129–156.
16. *Shen, W., Wang, L., Qi, H*. Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey // IEEE Transactions on Systems, Man and Cybernetics. – 2006. – Vol. 36. – P. 563–577.
17. *Quelhadj, D., Petrovich, S.* A Survey of Dynamic Scheduling in Manufacturing Systems // Journal of Scheduling. – 2009. – Vol. 12. – P. 417–431.
18. *Barbati, M., Bruno, M., Genovese, A*. Applications of Agent-Based Models for Optimization Problems: A Literature Review // Expert Systems with Applications. – 2012. – Vol. 39. – P. 6020–6028.
19. *Agnestis, A*. Multiagent Scheduling Problems // INFORMS Tutorials on Operational Research. – 2014. – P. 151–170.
20. *Lin, G. Y.-J., Solberg, J.J*. Intergated Shop Floor Control Using Autonomous Agents // IIE Transactions. – 1992. – Vol. 24. – P. 57–71.
21. *Frey, D., Nimis, J., Worn, H., Lockemann, P*. Benchmarking and Robust Multi-agent-Based Production and Control // Engineering Applications of Artificial Intelligence. – 2003. – Vol. 16. – P. 307–320.
22. *Mes, M., van der Heijen, M., van Haarten, A*. Comparison of Agent-Based Scheduling to Look-Ahead Heuristics for Real-Time Transportation Problems // European Journal of Operational Research. – 2007. – Vol. 181. – P. 59–75.
23. *Smith, R.G*. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver // IEEE Transactions on Computers. – 1980. – Vol. 29. – P. 1104–1113.
24. *Davis, R., Smith, R.G*. Negotiation as a Metaphor for Distributed Problem Solving // Artificial Intelligence. – 1983. – Vol. 20. – P. 63–109.
25. *Reaidy, J., Masotte, P., Diep, D*. Comparison of Negotiation Protocols in Dynamic Agent-Based Manufacturing Systems // International Journal of Production Economics. – 2006. – Vol. 99. – P. 117–130.
26. *Ferguson, D., Yemini, Y., Nikolaou, C*. Microeconomic Algorithms for Load Balancing in Distributed Computer Systems // Proc. of the 8th International Conference on Distributed Computing Systems. – San Jose, 1988. – P. 491–499.
27. *Waldspurger, C.A., Hogg, T., Huberman, B.A.,* et al. Spawn: A Distributed Computational Economy // IEEE Transactions on Software Engineering. – 1992. – Vol. 18. – P. 103–117.
28. *Huberman, B.A., Hogg, T*. Distributed Computation as an Economic System // Journal of Economic Perspectives. – 1995. – Vol. 9. – P. 141–152.
29. *Wang, J., Hong, Y., Xu J.,* et al. Cooperative and Competitive Multi-Agent Systems: From Optimization to Games // IEEE/CAA Journal of Automatica Sinica. – 2022. – Vol. 9. – P. 763–783.
30. *Renna, P.* A Review of Game Theory Models to Support Production Planning, Scheduling, Cloud Manufacturing and Sustainable Production Systems // Designs. – 2024. – Vol. 8, iss. 2. – Art. no. 26.
31. *Yang, B., Johansson, M*. Distributed Optimization and Games: A Tutorial Overview / In: Networked Control Systems. Lecture Notes in Control and Information Sciences. Ed. by A. Bemporad, M. Heemels, M. Johansson. – London: Springer, 2010. – Vol. 406. – P. 109–148.
32. *Madsen, J.R., Shamma, J.S*. Game Theory and Distributed Control // Handbook of Game Theory and Applications. – 2015. – Vol. 4. – P. 861–899.
33. *Briand, C., Ngueveu, S.U., Sucha, P*. Finding an Optimal Nash Equilibrium to the Multi-agent Scheduling Problem // Journal of Scheduling. – 2017. – Vol. 20. – P. 475–491.
34. *Agnetis, A., Briand, C., Ngueveu, S.U., Sucha, P.* Price of Anarchy and Price of Stability in Multi-agent Project Scheduling // Annals of Operations Research. – 2020. – Vol. 285. – P. 97–119.
35. *Hogg, T., Huberman, B.A., Williams, C.P*. Phase Transitions and the Search Problem // Artificial Intelligence. – 1996. – Vol. 81. – P. 1–15.
36. *Herroelen, W., De Reyck, B*. Phase Transitions in Project Scheduling // Journal of the Operational Research Society. – 1999. – Vol. 50. – P. 148–156.
37. *Easley, D., Kleinberg, J*. Networks, Crowds, and Markets: Reasoning about a Highly Connected World. – Cambridge:

Cambridge University Press, 2010. – URL: http://www.cs.cornell.edu/home/kleinber/networks-book/ (дата обращения 06.01.2025.) [Accessed January 6, 2025].
38. *Wellman, M., Walsh, W.E., Wurman, P., Makkie-Mason, K*. Auction Protocols for Decentralized Scheduling // Games and Economic Behavior. – 2001. – Vol. 35. – P. 291–303.
39. *Hall, N.G., Liu, Z*. On Auction Protocols for Decentralized Scheduling // Games and Economic Behavior. – 2011. – Vol. 72. – P. 583–585.
40. *Chen, W., Maturana, F., Norrie, D.H*. MetaMorph II: An Agent-Based Architecture for Distributed Intelligent Design and Manufacturing // Journal of Intelligent Manufacturing. – 2000. – Vol. 11. – P. 237–251.
41. *Munich, L.* Schedule Situations and Their Cooperative Game Theoretic Representations // European Journal of Operational Research. – 2024. – Vol. 316. – P. 767–778.
42. *Cavalieri, S., Garetti, M., Macchi, M., Taisch, M*. An Experimental Benchmarking of Two Multi-agent Architectures for Production Scheduling and Contro*l* // Computers in Industry. – 2000. – Vol. 43. – P. 139–152.
43. *Brennan, R.W., Norrie, D.H*. Evaluating the Performance of Reactive Control Architectures for Manufacturing Production Control // Computers in Industry. – 2001. – Vol. 46. – P. 235–245.
44. *Messie, D., Oh, J.C*. Cooperative Game Theory within Multi-agent Systems for Systems Scheduling // Proc. of the 4th International Conference on Hybrid Intelligent Systems (HIS'04). – Kitakyushu, Japan, 2004. – P. 166–171.
45. *Ramos, C*. An Architecture and a Negotiation Protocol for the Dynamic Scheduling of Manufacturing Systems // Proc. of the 1994 IEEE International Conference on Robotics and Automation. – San Diego, 1994. – Vol. 4. – P. 3161–3166.
46. *Maturana, F., Shen, W., Norrie, D.H*. MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing // International Journal of Production Research. – 1999. – Vol. 37. – P. 2159–2173.
47. *Paccagnan, D., Chandan, R., Marsden, J.R*. Utility and Mechanism Design in Multi-agent Systems: An Overview // Annual Reviews in Control. – 2022. – Vol. 53. – P. 315–328.
48. *Brussel, H.V., Wyns, J., Valckenaers, P*. Reference Architecture for Holonic Manufacturing Systems: PROSA // Computer in Industry. – 1998. – Vol. 37, no. 3. – P. 255–274.
49. *Bongaerts, L., Monostori, L., Mcfarlane, D., Kadar, B*. Hierarchy in Distributed Shop Control // Computers in Industry. – 2000. – Vol. 43. – P. 123–137.
50. *Rabelo, R.J., Camarinha-Matos, L.M*. Negotiation in Multi-agent Based Dynamic Scheduling // Robotics & Computer-Integrated Manufacturing. – 1994. – Vol. 11. – P. 303–309.
51. *Gou, L., Luh, P.B., Kyoya, Y*. Holonic Manufacturing Scheduling: Architecture, Cooperation Mechanism, and Implementation // Computers in Industry. – 1998. – Vol. 37. – P. 213–231.
52. *Skobelev, P*. Open Multi-agent Systems for Decision-Making Support // Avtometriya. – 2002. – No. 6. – P. 45–61.
53. *Скобелев П.О., Виттих В.А.* Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах // Автоматика и телемеханика. – 2003. – № 1. – С. 177–185. [*Skobelev, P.O., Vittikh, V.A.* Multiagent Interaction Models for Constructing the Needs-and-Means Networks in Open Systems // Automation and Remote Control. – 2003. – Vol. 64. – P. 162–169.]
54. *Vittikh, V., Skobelev, P*. The Compensation Method of Agents Interactions for Real Time Resource Allocation // Avtometriya. – 2009. – No. 2. – P. 78–87.
55. *Skobelev, P*. Multi-Agent Systems for Real Time Adaptive Resource Management / In: Industrial Agents: Emerging Applications of Software Agents in Industry. Ed. by P. Leitão, S. Karnouskos. – Amsterdam: Elsevier, 2015. – P. 207–230.
56. *Peretz-Andersson, E., Tabares, S., Mikalef, P., Parida, V*. Artificial Intelligence Implementation in Manufacturing SMEs: A Resource Orchestration Approach // International Journal of Information Management. – 2024. – Vol. 77, no. 1. – Art. no. 102781. – DOI: 10.1016/j.ijinfomgt.2024.102781
57. *Dwivedi, Y.K*. et al. Artificial Intelligence (AI): Multidisciplinary Perspectives on Emerging Challenges, Opportunities, and Agenda for Research, Practice and Policy // International Journal of information management. – 2021. – Vol. 57. – Art. no. 101994.
58. *Yang, W., Li, W., Cao, Y.,* et al. An Information Theory Inspired Real-Time Self-Adaptive Scheduling for Production-Logistics Resources: Framework, Principle, and Implementation // Sensors. – 2020. – Vol. 20. – Art. no. 7007. – DOI: https://doi.org/10.3390/s20247007
59. *Mourtzis,* D. Advances in Adaptive Scheduling in Industry 4.0 // Front. Manuf. Technol. – 2022. – Vol. 2. – Art. no. 937889. – DOI: 10.3389/fmtec.2022.937889
60. *Leitão, P., Colombo, A., Karnouskos, S*. Industrial Automation Based on Cyber-physical Systems Technologies: Prototype Implementations and Challenges // Computers in Industry. – 2016. – Vol. 81. – P. 11–25.
61. *Городецкий В.И., Ларюхин В.Б., Скобелев П.О.* Концептуальная модель цифровой платформы для киберфизического управления современными предприятиями. Часть I. Цифровая платформа и цифровая экосистема // Мехатроника, автоматизация, управление. – 2019. – Т. 20, № 6. – С. 323–332. [*Gorodeckij, V.I., Laryuhin, V.B., Skobelev, P.O.* Konceptual'naya model' cifrovoj platformy dlya kiber-fizicheskogo upravleniya sovremennymi predpriyatiyami. Part I. Cifrovaya platforma i cifrovaya ekosistema // Mekhatronika, avtomatizaciya, upravlenie. – 2019. – Vol. 20, no. 6. – P. 323–332. (In Russian)]
62. *Rzevski, G., Skobelev, P., Zhilyaev, A*. Emergent Intelligence in Smart Ecosystems: Conflicts Resolution by Reaching Consensus in Resource Management // Mathematics. – 2022. – Vol. 10, iss. 11. – Art. no. 1923.
63. *Galuzin, V., Galitskaya, A., Grachev, S.,* et al. The Autonomous Digital Twin of Enterprise: Method and Toolset for Knowledge-Based Multi-Agent Adaptive Management of Tasks and Resources in Real Time // Mathematics. – 2022. – Vol. 10, iss. 10. – Art. no. 1662.
64. Грачев С.П., Жиляев А.А., Ларюхин В.Б. Методы и средства построения интеллектуальных систем для решения сложных задач адаптивного управления ресурсами в реальном времени // Автоматика и телемеханика. – 2021. – № 11. – С. 30–67. [*Grachev, S.P., Zhilyaev, A.A., Laryukhin, V.B.,* et al. Methods and Tools for Developing Intelligent Systems for Solving Complex Real-Time Adaptive Resource Management Problems // Automation and Remote Control. – 2021. – Vol. 82, no. 11. – P. 1857–1885.]
65. *Skobelev, P.O., Borovik, S.Y*. On the Way from Industry 4.0 to Industry 5.0: From Digital Manufacturing to Digital Society // Industry 4.0. – 2017. – Vol. 2, iss. 6. – P. 307–311.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 27.01.2025,
после доработки 24.04.2025.
Принята к публикации 29.04.2025.

Леонидов Андрей Владимирович – д-р физ.-мат. наук, Физический институт им. П.Н. Лебедева РАН, Москва; Московский физико-технический институт, г. Долгопрудный, Московская область
✉ leonidovav@lebedev.ru
ORCID iD: https://orcid.org/0000-0002-6714-6261

Скобелев Петр Олегович – д-р техн. наук, Самарский федеральный исследовательский центр РАН, Самарский государственный технический университет, Самара,
✉ p.skobelev@kg.ru
ORCID iD: https://orcid.org/0000-0003-2199-9557

SOLVING COMPLEX RESOURCE MANAGEMENT PROBLEMS: FROM CLASSICAL OPTIMIZATION AND GAME THEORY TO MULTI-AGENT TECHNOLOGIES FOR REACHING CONSENSUS

A. V. Leonidov* and P. O. Skobelev**

*Lebedev Physical Institute, Russian Academy of Sciences, Moscow, Russia
*Moscow Institute of Physics and Technology, Dolgoprudny, Russia
**Samara Federal Research Center, Russian Academy of Sciences, Samara, Russia
**Samara State Technical University, Samara, Russia

*✉ leonidovav@lebedev.ru, **✉ p.skobelev@kg.ru

Abstract. Challenges and complex problems arising in the resource management of modern enterprises are considered. The existing resource planning models, methods and tools for enterprises are reviewed, and new requirements for adaptive multicriteria resource planning in real time are presented. The concept of autonomous artificial intelligence (AI) systems for adaptive resource planning based on multi-agent technologies is discussed. The evolution of the approach to solving complex resource management problems is described: from traditional optimization of a single objective function, ignoring the individual interests of participants, to game theory with their competition and cooperation. The approach to finding and maintaining a competitive equilibrium (consensus) between participants is further developed via conflict identification and negotiations for conflict resolution with mutual trade-offs. A basic model of a multi-agent demand-supply network with a virtual market and a compensation method for reaching consensus for adaptive resource planning are presented. The functionality and architecture of intelligent adaptive resource planning systems are considered. The implementation results of AI solutions for industrial applications are provided, and the possibility of improving the effectiveness of resource usage by enterprises is shown. Finally, the lessons learned from the experience in R&D work and the prospects of this approach are discussed.

Keywords: resource management, complexity, artificial intelligence, demand-supply networks, autonomous systems, adaptability, multi-agent technologies, self-organization, real-time economics.

Acknowledgments. The work of P.O. Skobelev was supported by the Ministry of Science and Higher Education of the Russian Federation.

УДК 519

ВЛИЯНИЕ ВНУТРЕННЕЙ СТРУКТУРЫ СЛОЖНОЙ СИСТЕМЫ НА ЕЕ ИНТЕГРАЛЬНЫЙ РИСК НА ПРИМЕРЕ ЗАДАЧИ МИНИМИЗАЦИИ РИСКА В ДРЕВОВИДНОЙ СТРУКТУРЕ

А. А. Широкий*, А. О. Калашников**

*,**Институт проблем управления им. В. А. Трапезникова РАН, г. Москва

*✉ shiroky@ipu.ru, **✉ aokalash@ipu.ru

Аннотация. Одной из наиболее общих математических постановок задачи управления рисками является задача «Защитник – Атакующий». Ее суть состоит в том, что указанные игроки с противоположными целями назначают элементам рассматриваемой системы некоторые объемы ресурсов из ограниченного пула таким образом, чтобы значение наперед заданной функции риска было, соответственно, минимальным или максимальным. В предположении независимости элементов системы эта задача исследована достаточно подробно. Однако элементы сложных систем связаны и влияют друг на друга, что приводит к значительным отклонениям измеряемого риска от прогнозируемого значения. Модели риска, учитывающие взаимное влияние элементов системы друг на друга, периодически встречаются в литературе, но системного понимания характера и степени влияния структуры сложной системы на ее интегральный риск пока не сформировано. С этой целью авторами запланирована публикация серии работ, в которых изучается влияние структур все возрастающей сложности на интегральный риск защищаемой системы. В ранее опубликованных работах были рассмотрены простая цепная и звездообразная структуры. В настоящей работе ранее полученные результаты обобщены на случай произвольной древовидной структуры. Поставлена задача оптимального с точки зрения минимизации риска размещения элементов в древовидной структуре, рассчитаны верхние оценки относительной погрешности ее приближенного алгоритмического решения для деревьев с небольшим числом ветвей и листьев, проанализировано поведение полученных оценок при увеличении числа листьев и ветвей. Показано, что полученные значения оценок не превосходят аналогичных, полученных для звездообразных структур в предшествующих работах авторов.

Ключевые слова: сложные системы, риск, структура системы, управление рисками, алгоритмы минимизации риска, задача оптимального размещения элементов.

ВВЕДЕНИЕ

Сложность дисциплины управления рисками обусловлена, в первую очередь, ее мультидисциплинарностью. Так, в книге [1] выделены 15 «размерностей» (*dimensions*) управления рисками, включающих в себя как сравнительно узкие предметные области (управление риском в цепочках поставок, управление финансовыми рисками), так и глобальные – например, вопросы этики в управлении рисками. Во второй части той же книги обсуждаются шесть кросс-дисциплин, частично перекрывающих все предметные области, а именно (в скобках даны оригинальные названия на английском языке): риск-культура (*risk culture*), принятие решений на основе риска (*risk-based decision making*), управление рисками в сложных системах (*risk leadership in complexity*), устойчивость (*resilience*), информационная неопределенность (*communication uncertainty*), риск в управлении ор-

ганизационными изменениями (*organizational change management and risk*). Приведенная выше классификация не является ни полной, ни единственно возможной. Она иллюстрирует тот факт, что управление рисками можно обсуждать в привязке к специфике конкретной управляемой системы, а можно – применительно к процессам и свойствам, характерным для целых классов систем. При этом будут отличаться не только применяемые модели и методы, но и используемая терминология (в частности, многие фигурирующие в книге [1] термины не имеют устоявшегося русскоязычного эквивалента), и даже само определение риска.

В условиях отсутствия общепринятой универсальной модели управления рисками объединяющую роль играют базовые принципы, верные для любой управляемой системы. Этот факт нашел отражение в стандарте ГОСТ Р ИСО 31000–2019: «Менеджмент риска. Принципы и руководство» [2], который также предлагает достаточно общее определение риска как следствия влияния неопределенности на достижение поставленных целей. При этом отмечается, что под следствием влияния неопределенности необходимо понимать отклонение от ожидаемого результата или события (позитивное и (или) негативное). Но для использования такого определения на практике необходимо научиться *измерять* цели, неопределенность и вызываемые ею отклонения. Отсюда следует необходимость исследования количественных соотношений для управления рисками с применением соответствующего математического аппарата.

Исходя из природы управления рисками, заключающегося в минимизации отклонений, математическая задача управления риском должна принадлежать к классу задач оптимизации (в случае наличия в системе игроков со стратегическим поведением постановка может также быть теоретико-игровой – см., например, работы [3–5]). Однако попытка поиска исследований, посвященных математическим моделям управления рисками, не привязанным к конкретному объекту, их классу или предметной области, скорее всего, будет неудачной. Причиной этого является то, что если работа посвящена математическим моделям оптимизации, применяемым в риск-менеджменте, то она будет относиться к соответствующему разделу математики, а не к управлению рисками. Если же речь идет именно об управлении рисками, то в работе будет обозначен управляемый объект, что привяжет работу к его специфике.

Тем не менее, вполне можно предположить существование моделей, достаточно универсальных, чтобы с их помощью можно было количественно рассматривать общие принципы управления рисками, но не привязанных к конкретным управляемым объектам, системам или их классам. По всей видимости, в наибольшей степени этому критерию отвечает модель защищаемой системы, представляющая собой взвешенный ориентированный граф, вершины которого являются элементами системы (объектами произвольной природы), а дуги с назначенными весами характеризуют направление и силу связей между элементами, имеющих значение для задачи управления рисками.

Отметим, что в общем случае такой граф является сложной сетью произвольной топологии. Моделируемый объект управления – защищаемая система – может представлять собой социальную сеть [6], сеть организаций [7], компьютерную сеть [8] или относиться к иному классу. Необходимо отметить, что в настоящей статье авторы работают с чисто математической постановкой задачи. Это означает, что структура защищаемой системы вовсе не обязательно отражает именно физическую или организационную структуру объекта, моделью которого является. В то же время дуги орграфа показывают взаимное влияние элементов друг на друга. Например, при решении задачи снижения рисков наступления авиационных происшествий в регионе модель будет скорее отражать структуру причинно-следственных связей между типами происшествий, их предпосылками и факторами влияния, нежели связи между элементами инфраструктуры управления воздушным движением. Если с точки зрения передачи риска друг другу все элементы системы независимы, то адекватной моделью такой ситуации будет частный случай безреберного графа.

В общем виде задачу управления рисками в сложных сетях можно сформулировать следующим образом.

Рассмотрим защищаемую систему, состоящую из конечного множества элементов (объектов, пока произвольной природы): $S = \{s_1, ..., s_i, ..., s_n\}$, $i \in N = \{1, ..., n\}$, $n \in \mathbb{N}$. Предположим, что существуют два субъекта (также пока произвольной природы), которых будем называть игрок *A* (иначе, Атакующий, *Attacker*) и игрок *D* (иначе, Защитник, *Defender*), имеющие несовпадающие интересы относительно состояния системы *S*.

Будем считать, что игрок D располагает некоторым объемом ресурса $X \geq 0$, который он может произвольным образом распределять между элементами системы S: $x = (x_1, \ldots, x_n)$, $x_i \geq 0$, $i \in N$, $\sum_{i=1}^{n} x_i \leq X$. Аналогично будем считать, что игрок A также располагает некоторым объемом ресурса $Y \geq 0$, который он может произвольным образом распределять между элементами системы S: $y = (y_1, \ldots, y_n)$, $y_i \geq 0$, $i \in N$, $\sum_{i=1}^{n} y_i \leq Y$.

В рамках рассматриваемой модели под ресурсом будем понимать любой измеримый и произвольно делимый ресурс, который может быть представлен неотрицательным действительным числом. В качестве ресурсов, в зависимости от контекста, могут пониматься финансовые, трудовые, временны́е, производственные и иные ресурсы (затраты).

Допустим, что воздействие в части передачи рисков описано взвешенным орграфом $G(S, W)$, $W \subseteq S \times S$, $w_{ij} = (s_i, s_j) \in W$, $i, j \in N$. Положим, что на $G(S, W)$ заданы функции

$$\rho : s \to \mathbb{R}_+^0, \ \sigma : W \to \mathbb{R}_+^0,$$

где ρ_i, $i \in N$ – «вес» узлов, отражающий текущее значение локального риска, а σ_{ij}, $i, j \in N$, – «вес» дуг, отражающий «интенсивность» передачи риска между элементами рассматриваемой системы. Матрица $\Sigma = \|\sigma_{ij}\|$ отражает степень (или силу) влияния i-го элемента системы S на j-й. Начальное (при $t = 0$) значение функций $\rho_i = \rho_i(x, y, t) = \rho_i(x, y, t = 0)$ определяется распределениями ресурсов x и y. Значения весов ρ_i при $t > 0$ зависят только от предшествующих по времени значений этих функций. Изменение значений весов элементов в результате их взаимного влияния друг на друга будет описываться следующим выражением:

$$\rho_i(t+1) = \rho_i(t) + \sum_{k=1}^{n} \sigma_{ik} \left(\rho_i(t) - \rho_i(t-1) \right), \quad t = 0, 1, \ldots; \ \rho_i(t = 0) = \tilde{\rho}_i. \tag{1}$$

Аргументы x, y в записи выше пропущены для компактности.

Обозначим $\mathcal{X}(X)$ множество допустимых распределений ресурса X между элементами системы S игроком D, а $\mathcal{Y}(Y)$ – множество допустимых распределений ресурса Y между элементами системы S игроком A:

$$\mathcal{X}(X) = \left\{ (x_1, \ldots, x_n) \in \mathbb{R}_+^n : x_i \geq 0, \, i \in N, \, \sum_{i=1}^{n} x_i \leq X \right\},$$

$$\mathcal{Y}(Y) = \left\{ (y_1, \ldots, y_n) \in \mathbb{R}_+^n : y_i \geq 0, \, i \in N, \, \sum_{i=1}^{n} y_i \leq Y \right\}.$$

Тогда задача игрока D («задача Защитника») заключается в нахождении распределения ресурса $x^* \in X$, минимизирующего интегральный риск (т. е. риск, характеризующий уязвимость системы в целом), и формально может быть записана в следующем виде:

$$x^* = \operatorname*{Argmin}_{x \in X} \lim_{t \to \infty} \rho(x, y, t) = \operatorname*{argmin}_{x \in X} \sum_{i=1}^{n} \lim_{t \to \infty} \rho_i(x, y, t). \tag{2}$$

Решение этой задачи с ограничениями на собственные значения матрицы взаимного влияния элементов приведено в работе [8]. При этом необходимо отметить, что для такой постановки задачи требуется с достаточной точностью идентифицировать не только текущие значения локальных рисков и функциональные зависимости $\rho(x, y, \cdot)$, но еще и количественно охарактеризовать взаимное влияние локальных рисков. Для реально существующих систем это может оказаться крайне трудоемко и даже невозможно. По этой причине актуальной становится задача нахождения общих принципов управления рисками системы, имеющей сложную сетевую структуру, которые позволили бы достигать эффекта снижения риска даже в условиях неполной информации. Решению этой задачи для древовидных структур и посвящена предлагаемая статья.

Структура изложения материала в работе следующая. В § 1 содержится краткий обзор математических моделей распространения отказов в сложных сетях. В § 2 приведена общая постановка задачи управления риском сложной системы с древовидной структурой. В § 3 предложено субоптимальное решение этой задачи. В заключении обсуждаются дальнейшие перспективы исследования.

1. КРАТКИЙ ОБЗОР МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ РАСПРОСТРАНЕНИЯ ОТКАЗОВ В СЛОЖНЫХ СЕТЯХ

В самом общем случае можно считать, что структура сложной системы является сложной сетью произвольной топологии. Успешной атакой

(или, по-другому, отказом) некоторого элемента системы будем считать наступление события, при котором этот элемент системы перестает выполнять свою функцию. Для простоты в настоящей работе будем рассматривать только бинарный случай – когда элемент функционален либо нефункционален полностью. Для исследования различных деструктивных эффектов (включая целенаправленные атаки на узлы и ребра) в таких сетях разработано достаточно много моделей и постоянно появляются новые. Весьма широко применяются модели оценки риска распространения отказов при исследовании сложных систем различной природы – в частности, киберфизических [9–17], вычислительных [18–19] и медико-социальных [20–22].

Ранние модели описывали развитие отказов, вызванных нецеленаправленными (например, случайными) воздействиями. Наиболее известными из них являются модель устойчивости к ошибкам [23–25], модель распространения лесного пожара [26–28] и ее производные, модели на базе клеточных автоматов [29–32], а также модели перколяции со случайными атаками [33]. Отметим, что последние имеют ряд модификаций, предполагающих, что деструктивные воздействия на узлы и ребра сети являются целенаправленными. К таковым относятся собственно перколяции с целенаправленными атаками [34–36], а также перколяции с локализованными атаками [37–40] и перколяции с k-ядром [41–43].

Упомянутые выше модели распространения отказов хорошо сочетаются с классическими моделями управления рисками в сложных сетях «Защитник – Атакующий» [44–46]. Напомним, что такие модели описывают конфликт между двумя игроками – Защитником и Атакующим – имеющими противоположные цели относительно рассматриваемой системы. Атакующий расходует ресурсы из некоторого доступного ему ограниченного пула с целью вывести систему из строя. Защитник, в свою очередь, пытается противостоять действиям Атакующего. В классических постановках Защитник решает задачу оптимального распределения ресурсов среди элементов системы с целью минимизации ее интегрального риска. Но он может выбрать и другой путь, а именно – модифицировать структуру самой системы с той же самой целью. Для описания такого сценария требуются другие модели.

Учет изменения структуры подразумевают, например, модели каскадного распространения ошибки [47, 48], однако в их рамках такие изменения не предполагаются целенаправленными. Возможность намеренного изменения структуры предусмотрена в моделях, модифицированных для случая двух взаимосвязанных сетей [49–51], но именно в отношении ребер, связывающих сети между собой.

Таким образом, для решения задач управления структурой сложной системы, в том числе с целью минимизации ее интегрального риска, существующего аппарата моделирования недостаточно. В настоящей (и ряде предшествующих) работ авторы сконцентрировались на том, чтобы рассчитать влияние на риск собственно структуры, безотносительно выделяемых ресурсов.

С этой целью базовая задача (2) была переформулирована таким образом, чтобы вместо поиска оптимального назначения ресурсов элементам системы с зафиксированной структурой осуществлялись бы поиск оптимального размещения элементов в некоторой заданной структуре и сравнение структур между собой. Решать эту задачу «в лоб» не получается из-за высокой вычислительной сложности, поэтому авторы ищут приближенные решения для различных типов структур в порядке возрастания их сложности, а именно:

1) простая цепь – получено аналитическое решение, см. работу [52];

2) «звезда» – предложено приближенное решение с гарантированной погрешностью [53];

3) древовидная структура – рассматривается в настоящей работе;

4) произвольная структура – решение будет построено на основе обобщения результатов, полученных для более простых структур.

Отметим, что в общем виде задача эффективного управления сложной системой в условиях неопределенности эквивалентна задаче минимизации риска, где под риском понимается измеримое отклонение от максимально эффективного (целевого) режима функционирования рассматриваемой системы [2] – математическая эквивалентность постановок показана, например, в работе [54]. Поэтому, хотя проблема синтеза или совершенствования структуры, в частности, организационной системы управления, несомненно, не сводится только лишь к распределению ресурсов, а уровень риска – лишь один из ключевых показателей эффективности ее функционирования, в смысле обеспечения достижения целей системы управления его, по-видимому, следует считать наиболее важным.

2. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ РИСКОМ СЛОЖНОЙ СИСТЕМЫ С ДРЕВОВИДНОЙ СТРУКТУРОЙ

Предположим, что защищаемая система включает в себя n элементов $s_1,..., s_n \in S$, $n \in \mathbb{N}$. Пред-

положим также, что каждому из них сопоставлены два числа: $p_i^0 \in (0, 1]$ – удельная вероятность успешной атаки на i-й элемент; $u_i > 0, u \in \mathbb{R}^+$ – величина ущерба, который будет нанесен в случае успешной атаки i-го элемента.

Определение 1. Удельным риском i-го элемента назовем величину $\rho_{s_i}^0 = u_i p_i^0$. ♦

Зададим структуру $W_m = \langle G(V, E), T \rangle, T \subseteq V$, где $G(V, E)$ – ориентированный граф с множеством вершин V и множеством дуг E, а T – подмножество V, которое будем называть *периметром*. В настоящей работе будем рассматривать структуры с периметром, состоящим ровно из одной вершины.

Определение 2. Будем говорить, что последовательность векторов

$$B = \{(b_{01}, \ldots, b_{0q_0}), (b_{11}, \ldots, b_{1q_1}), \ldots, (b_{l1}, \ldots, b_{lq_l}), \ldots, (b_{L1}, \ldots, b_{Lq_L})\},$$
$$b, L, q_l \in \mathbb{N}, l \in \mathbb{N} \cup \{0\}$$

определяет направленное дерево с m листьями, если:

- число $b_{li}, i \in \{1, \ldots, q_l\}$, обозначает число дуг, выходящих из соответствующей вершины;
- число L равно длине максимального пути;
- число $q_l, l \le L$, определяет число вершин яруса l (длина пути из корня до которых равна l);
- $q_0 = 1; q_l = \sum_{i=1}^{q_{l-1}} b_{(l-1)i} \; \forall l > 0; q_L = m;$
- $b_{L1} = b_{L2} = \ldots = b_{Lq_L} = 0$. ♦

Определение 3. Будем говорить, что порожденная последовательностью B структура системы имеет тип «дерево с m листьями», и записывать $W_m = \langle G(V, E), T \rangle$, если

$$V = \left\{ \{v_0\} \cup \bigcup_{j=1}^{b_{01}} \{v_{0j}\} \cup \bigcup_{i=1}^{q_1} \bigcup_{j=1}^{b_{1i}} \{v_{0ij}\} \cup \ldots \right.$$
$$\left. \ldots \cup \bigcup_{i=1}^{q_{L-1}} \bigcup_{j=1}^{b_{(L-1)i}} \{v_{0\ldots ij}\} \right\};$$

$$E = \left\{ \bigcup_{j=1}^{b_{01}} \{(v_0, v_{0j})\} \cup \bigcup_{i=1}^{q_1} \bigcup_{j=1}^{b_{1i}} \{(v_{0i}, v_{0ij})\} \cup \ldots \right.$$
$$\left. \ldots \cup \bigcup_{i=1}^{q_{L-1}} \bigcup_{j=1}^{b_{(L-1)i}} \{(v_{0\ldots i}, v_{0\ldots ij})\} \right\}; \; T = \{v_0\}. \blacklozenge$$

Пример дерева с четырьмя листьями на третьем ярусе приведен на рис. 1. Отметим, что частным случаем дерева, когда $b_{li} = 1, q_l = m \;\; \forall l < L, l \ne 0$, является структура типа «звезда с m лучами». Соответствующие типы структур рассматривались авторами ранее в работе [53].

Определение 4. Взаимно однозначное отображение $M^{-1} : S \to V$ будем называть размещением элементов S в древовидной структуре W_m. Соответствующее обратное отображение $M : V \to S$ будем называть проекцией дерева W_m на множество элементов S. ♦

Отметим, что для существования такого отображения необходимо равенство между числом вершин графа $G(V, E)$ и числом элементов защищаемой системы. В случае бесконечного числа вершин множества V и S должны быть счетными.

Определение 5. Интегральным риском системы со множеством элементов S, размещенных в древовидной структуре W_m с помощью взаимно однозначного отображения $M^{-1} : S \to V$, будем называть величину

$$\rho\left(S, W_m, M^{-1}\right) = \rho_{M(v_o)} + \sum_{j=1}^{b_{01}} \rho_{M(v_{0j})} + \sum_{i=1}^{q_1} \sum_{j=1}^{b_{1i}} \rho_{M(v_{0ij})} + \ldots + \sum_{i=1}^{q_{L-1}} \sum_{j=1}^{b_{(L-1)i}} \rho_{M(v_{0\ldots ij})}. \blacklozenge \quad (3)$$

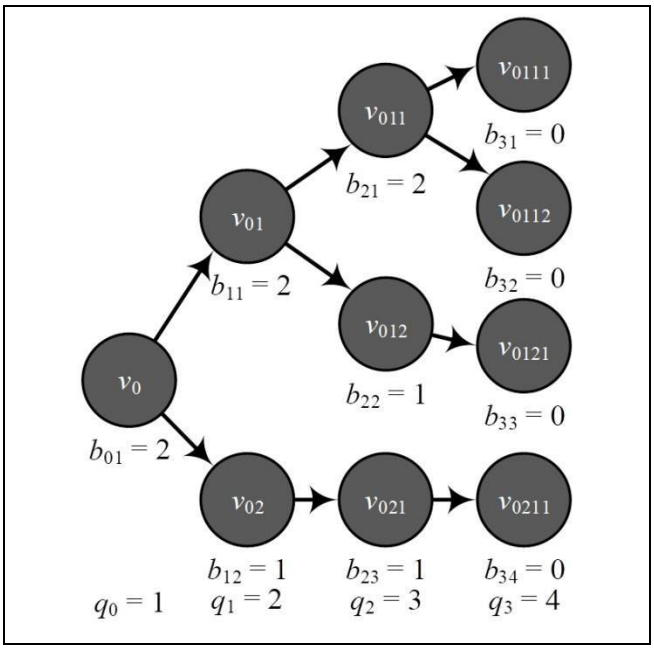

Рис. 1. Пример дерева с $m = 4$, $L = 3$. Индексы вершин являются уникальными и отражают простой путь до периметра – в него входят все вершины с индексами, являющимися подстроками индекса рассматриваемой вершины

Пусть защищаемая система включает в себя множество элементов $S=\{s_1, s_2,..., s_n\}$, $n\in\mathbb{N}$, с соответствующими им удельными вероятностями успешной атаки $P=\{p^0_{s_1}, p^0_{s_2}, p^0_{s_n}\}$ и ущербами $U=\{u_{s_1}, u_{s_2},..., u_{s_n}\}$. Предположим также, что возможные пути атаки задаются древовидной структурой $W_m=\langle G(V, E), T\rangle$, где $\sum_{l=1}^{L} q_l = n$. Тогда задача минимизации интегрального риска защищаемой системы заключается в нахождении такого размещения M^{-1} элементов S в структуре W_m, что

$$\rho(S, W_m, M^{-1})\to\min. \qquad (4)$$

Для частного случая $m=1$ в работе [52] описано точное решение. Для случая $q_1=m\ \forall l<L, l\neq 0$ в работе [53] получено субоптимальное решение с априорной оценкой относительной погрешности. Далее аналогичным образом найдем такую оценку для рассматриваемых в настоящей работе древовидных структур.

3. ПРИБЛИЖЕННОЕ РЕШЕНИЕ ЗАДАЧИ ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ В ДРЕВОВИДНОЙ СТРУКТУРЕ

Предположим, что величины ущербов в случае успешной атаки для всех элементов системы оцениваются одинаково, т. е. $u_{s_i}=u\ \forall i\in\{1,..., n\}$. Тогда задача (4) принимает следующий вид:

$$\rho\left(S, W_m, M^{-1}\right)=u\left(p_{M(v_0)}+\sum_{k=0}^{L-1}\sum_{i=1}^{q_k}\sum_{j=1}^{b_{ki}}\left(p_{M(v_{0...ij})}\cdot p_{M(v_{0...i})}\cdot....\cdot p_{M(v_0)}\right)\right)\to\min. \qquad (5)$$

Потребуем также, чтобы выражение

$$p_{M(v_0)}+\sum_{k=0}^{L-1}\sum_{i=1}^{q_k}\sum_{j=1}^{b_{ki}}\left(p_{M(v_{0...ij})}p_{M(v_{0...i})}\cdot....\cdot p_{M(v_0)}\right)$$

являлось конечным для любых значений L и $m=q_L$. С этой целью ограничим удельные риски элементов величиной, называемой предельно допустимым удельным риском и определяемой следующим образом (см. также работу [53]).

Определение 6. Предельно допустимым удельным риском элемента защищаемой системы, размещенного в структуре $W_m=\langle G(V, E), T\rangle$, назовем величину

$$\rho^0_{\max}=\frac{u}{1+\sqrt{m}}.\ \blacklozenge$$

Отметим, что при выполнении ограничения $p_i\le\frac{1}{1+\sqrt{m}}=p^0_{\max}$ справедливо следующее неравенство:

$$\rho\left(S, W_m, M^{-1}\right)\le \le u\left(p^0_{\max}+\sum_{k=0}^{L-1}\sum_{i=1}^{q_k}\sum_{j=1}^{b_{ki}}\left(p^0_{\max}\right)^{k+1}\right)\le u. \qquad (6)$$

Равенство в формуле (6) достигается в случае $L=\infty$ для любого конечного m. Важным обстоятельством является то, что в силу построения величины (6), полученные ранее для звездообразной структуры [53, *табл. 2*] верхние оценки приращения значения интегрального риска по мере удаления от периметра остаются верными и для древовидных структур. Это позволяет рассчитывать на то, что приведенные в той же работе верхние оценки относительного отклонения от оптимального решения в случае произвольного размещения в структуре элементов на фиксированном удалении от периметра удастся использовать для деревьев.

Для того, чтобы это проверить, проведем серию численных экспериментов по аналогии с работой [53]. Введем следующие ограничения:

$$\begin{cases} u=1; \\ 0<p^0_{M(v_{0...i})}\le p^0_{M(v_{0...ij})} \\ \quad \forall i\in\{1,..., q_k\}, j\in\{1,..., b_{ki}\}, k\in\{0,..., L-1\}; \\ p^0_{M(v_{0...ij})}\le\frac{u}{1+\sqrt{m}} \\ \quad \forall i\in\{1,..., q_k\}, j\in\{1,..., b_{ki}\}, k\in\{0,..., L-2\}; \\ p^0_{M(v_{0...ij})}\le\sum_{l=L}^{\infty}\left(\frac{u}{1+\sqrt{m}}\right)^{l+1} \\ \quad \forall i\in\{1,..., q_{L-1}\}, j\in\{1,..., b_{(L-1)i}\}. \end{cases}$$

Будем генерировать выражения вида (3) для всех размещений, получаемых путем перестановки элементов, находящихся на первом и более дальних ярусах дерева, т. е. на фиксированном расстоянии k от периметра, начиная с $k=1$. Для каждого k потребуется рассмотреть случаи $q_k=2,..., m$, соответствующие деревьям с q_k вершинами k-го яруса. Затем рассмотрим все возможные модули разности этих выражений, для каждого из которых проведем поиск глобального максимума. Разделив найденное значение на минимум разности этих двух выражений, получим величину относительного отклонения. Максимум таких отклонений даст

верхнюю оценку относительной погрешности решения задачи (5).

Результаты вычисления величин относительной погрешности для небольших деревьев приведены в таблице.

На рис. 2 показано поведение полученных значений относительной погрешности на ярусах 1–4 в зависимости от числа выходящих из вершин (речь идет, соответственно, о подмножествах $\{v_{0j}\}_{j=1}^{b_{01}}$ (рис. 2, *а*), $\left\{\{v_{0ij}\}_{j=1}^{b_{1i}}\right\}_{i=1}^{q_1}$ (рис. 2, *б*), $\left\{\{v_{0...ij}\}_{j=1}^{b_{2i}}\right\}_{i=1}^{q_2}$ (рис. 2, *в*) и $\left\{\{v_{0...ij}\}_{j=1}^{b_{3i}}\right\}_{i=1}^{q_3}$ (рис. 2, *г*)) текущего яруса дуг и числа листьев в дереве. Отметим, что оно сходно с поведением, демонстрируемым величинами максимального прироста риска при заданной величине предельного собственного риска (подробнее см. табл. 2 из работы [53]). А именно, на первом ярусе погрешность растет вместе с числом листьев в дереве. На втором и более дальних ярусах, при числе листьев $m \geq 5$, погрешность монотонно убывает. При небольшом числе листьев монотонность нарушается. Природа этого явления кратко описана в работе [53]. Более подробно исследовать этот вопрос авторы не планируют, поскольку их задачей является разработка методов управления рисками в сложных сетевых структурах с тысячами вершин и ребер.

Отметим, что поскольку полученные в эксперименте значения относительного отклонения монотонно убывают с удалением от периметра, для построения системы с интегральным риском, не превышающим минимально возможный более, чем на 6,07 % (оценка на рис. 2, *б*), достаточно оптимальным образом выбрать элемент системы для помещения в вершину-периметр, а также элементы для размещения в вершинах первого яруса. При введенном ранее условии $u_{s_i} = u \ \forall i \in \{1,...,n\}$ это будут вершины с наименьшими удельными локальными рисками. Такая погрешность допустима для достаточно широкого класса систем. В случае, когда требуется более высокий уровень защищенности, потребуется дополнительно отобрать из оставшихся неразмещенными элементов q_2 с наименьшими удельными локальными рисками, а оставшиеся разместить в вершинах $V \setminus \left\{\{v_0\} \cup \bigcup_{j=1}^{b_{01}} \{v_{0j}\} \cup \bigcup_{i=1}^{q_1} \bigcup_{j=1}^{b_{1i}} \{v_{0ij}\}\right\}$ произвольным образом. Затем потребуется найти оптимальное

Численные оценки относительной погрешности решения задачи оптимального размещения элементов в подмножествах вершин древовидной структуры, округление до четвертого знака с избытком

Число вершин в подмножестве	Подмножество вершин			
	$\{v_{0j}\}_{j=1}^{b_{01}}$	$\left\{\{v_{0ij}\}_{j=1}^{b_{1i}}\right\}_{i=1}^{q_1}$	$\left\{\{v_{0...ij}\}_{j=1}^{b_{2i}}\right\}_{i=1}^{q_2}$	$\left\{\{v_{0...ij}\}_{j=1}^{b_{3i}}\right\}_{i=1}^{q_3}$
Три листа (m = 3)				
2	0,3095	0,0585	0,0119	0,0030
3	0,1548	0,0434	0,0088	0,0022
Четыре листа (m = 4)				
2	0,3750	0,0571	0,0107	0,0025
3	0,2500	0,0461	0,0086	0,0020
4	0,2000	0,0607	0,0107	0,0024
Пять листьев (m = 5)				
2	0,4223	0,0553	0,0097	0,0021
3	0,3168	0,0468	0,0082	0,0018
4	0,2563	0,0591	0,0098	0,0021
5	0,1709	0,0515	0,0085	0,0018
Шесть листьев (m = 6)				
2	0,4588	0,0535	0,0089	0,0018
3	0,3671	0,0466	0,0077	0,0016
4	0,2997	0,0574	0,0090	0,0018
5	0,2248	0,0512	0,0080	0,0016
6	0,1899	0,0607	0,0091	0,0018

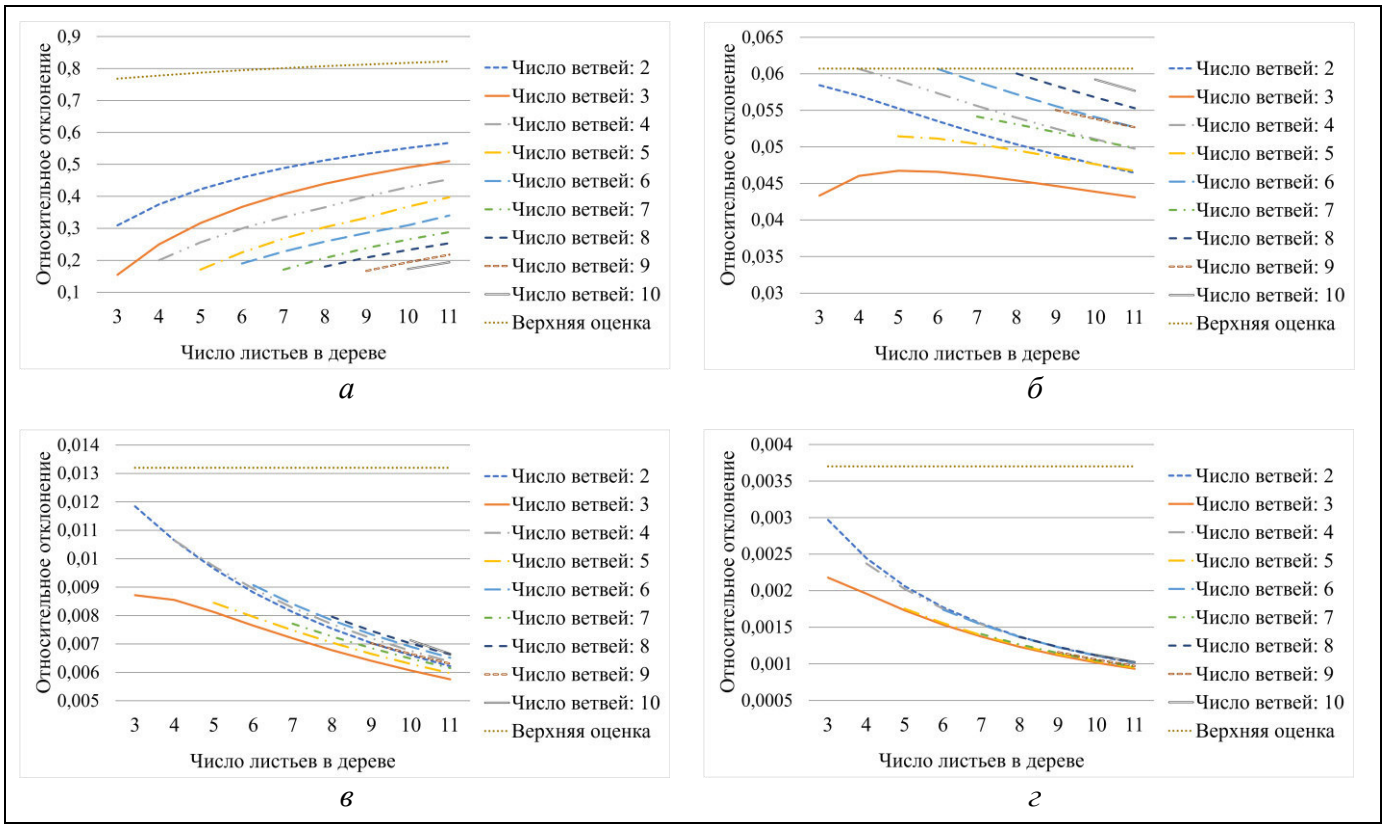

Рис. 2. Поведение численных оценок относительного отклонения от решения задачи оптимального размещения элементов в подмножествах древовидной структуры в зависимости от числа листьев в дереве: значения верхней оценки в подмножестве вершин первого яруса (*а*), соответствуют максимальному приращению риска в звездообразной структуре, т. е. при совпадающем числе ветвей и листьев; значения верхней оценки для подмножеств вершин ярусов 2–4 (*б–г*) равны значениям, полученным в работе [53] для звездообразной структуры с четырьмя лучами (для *б*), и двумя лучами (для *в* и *г*)

размещение отобранных элементов в вершинах $\bigcup_{i=1}^{q_1}\bigcup_{j=1}^{b_{1i}}\{v_{0ij}\}$, например, путем вычисления $q_3!$ значений интегрального риска для всех возможных перестановок элементов, размещенных в вершинах второго яруса. В этом случае погрешность полученного решения составит менее 1,32 %.

ЗАКЛЮЧЕНИЕ

Настоящая статья является частью серии работ, посвященных изучению влияния внутренней структуры сложной системы на ее интегральный риск. Для достижения цели исследования была сформулирована задача об оптимальном размещении элементов защищаемой системы внутри заданной структуры. Такая постановка позволяет изучать влияние структуры на риск безотносительно выделяемых для изменения последнего ресурсов (как это предусмотрено в классической постановке «Атакующий – Защитник»). В связи с тем, что прямой путь к решению поставленной задачи не просматривался, авторы приняли решение последовательно рассматривать отдельные виды структур в порядке возрастания сложности.

В предшествующих работах были рассмотрены цепные структуры [52], для которых было найдено общее решение в виде критерия предпочтения выбора того или иного элемента системы для размещения в вершине простой цепи в зависимости от ее положения относительно периметра. Для звездообразной структуры, включающей в себя единственную вершину-периметр и произвольное конечное число исходящих из нее простых цепей (в том числе бесконечной длины), были найдены [53] верхние оценки относительной погрешности решения поставленной задачи в случае, если начиная с некоторого расстояния от периметра размещение элементов будет произвольным.

В настоящей статье полученные для звездообразных структур оценки были обобщены на случай произвольных древовидных структур. Для этого была введена система обозначений вершин в дереве, в явном виде указывающих путь к текущей вершине от периметра, поставлена задача оптимального размещения элементов в древовидной структуре, рассчитаны численные оценки относи-

тельной погрешности решения этой задачи для деревьев с небольшим числом ветвей и листьев. Было проанализировано поведение полученных оценок при увеличении числа листьев и ветвей, сделан вывод о том, что погрешности решения не превосходят верхних оценок, полученных ранее для звездообразных структур.

Полученные результаты могут быть применены, например, при решении задач управления рисками в компьютерных сетях с переменной топологией, таких как туманные вычислители [55] или беспроводные mesh-сети [56], при проектировании систем охраны [57] и многих других. Предложенный подход позволяет оценить, в какой мере перестроение топологии компьютерной сети (или, как в другом примере, структуры системы охраны) влияет на ее защищенность в целом, а полученные верхние оценки позволяют быстро оценить величину интегрального риска рассматриваемой системы.

Следующим этапом работы станет рассмотрение структур произвольной топологии с одновершинным периметром.

ЛИТЕРАТУРА

1. *The Risk* Management Handbook: A Practical Guide to Managing the Multiple Dimensions of Risk / Ed by D. Hillson. – London, UK: Kogan Page Publishers, 2023. – 416 p.
2. *ГОСТ* Р ИСО 31000–2019: «Менеджмент риска. Принципы и руководство». – М.: Стандартинформ, 2020. – I–IV, 13 с. [*ISO* 31000: Risk Management-Principles and Guidelines. – Geneva, Switzerland: International Organization for Standardization, 2018. – 24 p.]
3. *Rass, S.* On Game-Theoretic Risk Management (Part One) – Towards a Theory of Games with Payoffs that are Probability-Distributions. – arXiv:1506.07368, 2015. – DOI: https://doi.org/10.48550/arXiv.1506.07368
4. *Rass, S.* On Game–Theoretic Risk Management (Part Two) – Algorithms to Compute Nash-Equilibria in Games with Distributions as Payoffs. – arXiv:1511.08591, 2015. – DOI: https://doi.org/10.48550/arXiv.1511.08591
5. *Rass S.* On Game–Theoretic Risk Management (Part Three) – Modeling and Applications. – arXiv:1711.00708, 2017. – DOI: https://doi.org/10.48550/arXiv.1711.00708
6. *Остапенко А.Г., Паринов А.В., Калашников А.О.,* и др. Социальные сети и деструктивный контент / под. ред. Д.А. Новикова. – М.: Горячая линия – Телеком, 2017. – 276 с. [*Ostapenko, A.G., Parinov, A.V., Kalashnikov, A.O.,* et al. Sotsial'nye seti i destruktivnyi kontent / pod. red. D.A. Novikova. – M.: Goryachaya liniya – Telekom, 2017. – 276 s. (In Russian)]
7. *Калашников А.О.* Модели и методы организационного управления информационными рисками корпораций. – М.: ИПУ РАН, 2011. – 312 с. [*Kalashnikov, A.O.* Modeli i metody organizatsionnogo upravleniya informatsionnymi riskami korporatsii. – M.: IPU RAN, 2011. – 312 s. (In Russian)]
8. *Калашников А.О., Аникина Е.В.* Управление информационными рисками сложной системы с использованием механизма «когнитивной игры» // Информация и безопасность. – 2020. – Т. 38, № 4. – С. 2–10. [*Kalashnikov, A.O., Anikina, E.V.* Upravlenie informatsionnymi riskami slozhnoi sistemy s ispol'zovaniem mekhanizma «kognitivnoi igrY» // Cybersecurity Issues. – 2020. – Vol. 38, no 4. – P. 2–10. (In Russian)]
9. *Deng, S., Zhang, J., Wu, D.,* et al. A Quantitative Risk Assessment Model for Distribution Cyber-Physical System Under Cyberattack // IEEE Transactions on Industrial Informatics. – 2023. – Vol. 19, no. 3. – P. 2899–2908.
10. *Hu, B., Zhou, C., Tian, Y.–C.,* et al. Attack Intention Oriented Dynamic Risk Propagation of Cyberattacks on Cyber-Physical Power Systems // IEEE Transactions on Industrial Informatics. – 2023. – Vol. 19, no. 3. – P. 2453–2462.
11. *Xiaoxiao, G., Tan, Y., Wang, F.* Modeling and Fault Propagation Analysis of Cyber-Physical Power System // Energies. – 2020. – Vol. 13, no. 3. – Art. no. e539.
12. *Gao, X., Peng, M., Tse, C.K., Zhang, H.* A Stochastic Model of Cascading Failure Dynamics in Cyber-Physical Power Systems // IEEE Systems Journal. – 2020. – Vol. 14, no. 3. – P. 4626–4637.
13. *Marashi, K., Sarvestani, S.S., Hurson, A.R.* Identification of Interdependencies and Prediction of Fault Propagation for Cyber-Physical Systems // Reliability Engineering & System Safety. – 2021. – Vol. 215. – Art. no. e107787.
14. *Yan, K., Liu, X., Lu, Y., Qin, F.* A Cyber-Physical Power System Risk Assessment Model Against Cyberattacks // IEEE Systems Journal. – 2023. – Vol. 17, no. 2. – P. 2018–2028.
15. *Pelissero, N., Laso, P.M., Puentes, J.* Impact Assessment of Anomaly Propagation in a Naval Water Distribution Cyber–Physical System // Proceedings of 2021 IEEE International Conference on Cyber Security and Resilience (CSR). – Rhodes, Greece, 2021. – P. 518–523.
16. *Islam, M.Z., Lin, Y., Vokkarane, V.M., Venkataramanan, V.* Cyber–Physical Cascading Failure and Resilience of Power Grid: A Comprehensive Review // Frontiers in Energy Research. – 2023. – Vol. 11. – Art. no. e1095303.
17. *Zhang, C., Xu, X., Dui, H.* Analysis of Network Cascading Failure Based on the Cluster Aggregation in Cyber-Physical Systems // Reliability Engineering & System Safety. – 2020. – Vol. 202. – Art. no. e106963.
18. *Xing, L.* Cascading Failures in Internet of Things: Review and Perspectives on Reliability and Resilience // IEEE Internet of Things Journal. – 2021. – Vol. 8, no. 1. – P. 44–64.
19. *Wang, Q., Jia, G., Jia, Y, Song, W.* A New Approach for Risk Assessment of Failure Modes Considering Risk Interaction and Propagation Effects // Reliability Engineering & System Safety. – 2021. – Vol. 216. – Art. no. e108044.
20. *Khoshakhlagh, A., Moradi Hanifi, S., Laal, F.,* et al. A Model to Analyze Human and Organizational Factors Contributing to Pandemic Risk Assessment in Manufacturing Industries: FBN–HFACS Modelling // Theoretical Issues in Ergonomics Science. – 2023. – Vol. 25, no. 4. – P. 369–390.
21. *Moore, S., Rogers, T.* Predicting the Speed of Epidemics Spreading in Networks // Physical Review Letters. – 2020. – Vol. 124, no. 6. – P. 685–689.
22. *Nasution, H., Jusuf, H., Ramadhani, E., Husein, I.* Model of Spread of Infectious Diseases // Systematic Reviews in Pharmacy. – 2020. – Vol. 11, no. 2. – P. 685.

23.*Albert, R., Jeong, H., Barabasi, A.-L.* Error and Attack Tolerance of Complex Networks // Nature. – 2000. – Vol. 406. – P. 378–382.

24.*Artime, O., Grassia, M., De Domenico, M.,* et al. Robustness and Resilience of Complex Networks // Nature Reviews Physics. – 2024. – Vol. 6, no. 2. – P. 114–131.

25.*Ming, L., Run-Ran, L., Linyuan, L.,* et al. Percolation on Complex Networks: Theory and Application // Physics Reports. – 2021. – Vol. 907. – P. 1–68.

26.*Bak, P., Chen, K., Tang, C.* A Forest-Fire Model and Some Thoughts on Turbulence // Physics Letters A. – 1990. – Vol. 147, no. 5–6. – P. 297–300.

27.*Palmieri, L., Jensen, H.J.* The Forest Fire Model: The Subtleties of Criticality and Scale Invariance // Frontiers in Physics. – 2020. – Vol. 8. – Art. no. e00257.

28.*Rybski, D., Butsic, V., Kantelhardt, J.W.* Self-organized Multistability in the Forest Fire Mode // Physical Review E. – 2021. – Vol. 104, no. 1. – Art. no. eL012201.

29.*Newman, D.E., Nkei, B., Carreras, B.A.,* et al. Risk Assessment in Complex Interacting Infrastructure Systems // Proceedings of 38th Annual Hawaii International Conference on System Sciences (HICSS'05). – Big Island, HI, USA, 2005. – DOI: 10.1109/HICSS.2005.524

30.*Li, X., Ji, L., Zhu, H.,* et al. Cellular Automata–Based Simulation of Cross-space Transmission of Energy Local Area Network Risks: A Case Study of a Power Supply Station in Beijing // Sustainable Energy, Grids and Networks. – 2021. – Vol. 27. – Art. no. e100521.

31.*Torres, M.A., Chávez-Cifuentes, J.F., Reinoso, E.* A Conceptual Flood Model Based on Cellular Automata for Probabilistic Risk Applications // Environmental Modelling & Software. – 2022. – Vol. 157. – Art. no. e105530.

32.*Sequeira, J.G.N., Nobre, T., Duarte, S.,* et al. Proof–of–Principle That Cellular Automata Can Be Used to Predict Infestation Risk by *Reticulitermes grassei* (Blattodea: Isoptera) // Forests. – 2022. – Vol. 13, no. 2. – Art. no. e237.

33.*Gallos, L.K., Cohen, R., Argyrakis, P.,* et al. Stability and Topology of Scale-Free Networks under Attack and Defense Strategies // Physical Review Letters. – 2005. – Vol. 94, no. 18. – Art. no. e188701.

34.*Gallos, L.K., Cohen, R., Argyrakis, P.,* et al. Network Robustness and Fragility: Percolation on Random Graphs // Physical Review Letters. – 2000. – Vol. 85, no. 25. – Art. no. e5468.

35.*Wang, F., Dong, G., Tian, L., Stanley, H.E.* Percolation Behaviors of Finite Components on Complex Networks // New Journal of Physics. – 2022. – Vol. 24, no. 4. – Art. no. e043027.

36.*Dong, G., Luo, Y., Liu, Y.,* et al. Percolation Behaviors of a Network of Networks under Intentional Attack with Limited Information // Chaos, Solitons & Fractals. – 2022. – Vol. 159. – Art. no. e112147.

37.*Shao, S., Huang, X., Stanley, H.E., Havlin, S.* Percolation of Localized Attack on Complex Networks // New Journal of Physics. – 2015. – Vol. 17, no. 2. – Art. no. e023049.

38.*Dong, G., Xiao, H., Wang, F.,* et al. Localized Attack on Networks with Clustering // New Journal of Physics. – 2019. – Vol. 21, no. 1. – Art. no. e013014.

39.*Shang, Y.* Percolation of Attack with Tunable Limited Knowledge // Physical Review E. – 2021. – Vol. 103, no. 4. – Art. no. e042316.

40.*Qing, T., Dong, G., Wang, F.,* et al. Phase Transition Behavior of Finite Clusters under Localized Attack // Chaos: An Interdisciplinary Journal of Nonlinear Science. – 2022. – Vol. 32, no. 2. – Art. no. e023105.

41.*Goltsev, A.V., Dorogovtsev, S.N., Mendes, J.F.F.* K-Core (Bootstrap) Percolation on Complex Networks: Critical Phenomena and Nonlocal Effects // Physical Review E. – 2006. – Vol. 73, no. 5. – Art. no. e056101.

42.*Burleson-Lesser, K., Morone, F., Tomassone, M.S., Makse, H.A.* K-core Robustness in Ecological and Financial Networks // Scientific Reports. – 2020. – Vol. 10, no. 1. – Art. no. 3357.

43.*Shang, Y.* Generalized K-cores of Networks under Attack with Limited Knowledge // Chaos, Solitons & Fractals. – 2021. – Vol. 152. – Art. no. e111305.

44.*Al Mannai, W.I., Lewis, T.G.* A General Defender-Attacker Risk Model for Networks // The Journal of Risk Finance. – 2008. – Vol. 9, no. 3. – P. 244–261.

45.*Peng, R., Wu, D., Sun, M., Wu, S.* An Attack-Defense Game on Interdependent Networks // Journal of the Operational Research Society. – 2021. – Vol. 72, no. 10. – P. 2331–2341.

46.*Ren, J., Liu, J., Dong, Y.,* et al. An Attacker-Defender Game Model with Constrained Strategies // Entropy. – 2024. – Vol. 26, no. 8. – Art. no. e26080624.

47.*He, S., Zhou, Y., Yang, Y.,* et al. Cascading Failure in Cyber-Physical Systems: A Review on Failure Modeling and Vulnerability Analysis // IEEE Transactions on Cybernetics. – 2024. – P. 1–19. – DOI: 10.1109/TCYB.2024.3411868

48.*Zhou, F., Xu X., Trajcevski, G., Zhang, K.* A Survey of Information Cascade Analysis: Models, Predictions, and Recent Advances // ACM Computing Surveys (CSUR). – 2021. – Vol. 54, no. 2. – P. 1–36.

49.*Cui, P., Zhu, P., Wang, K.,* et al. Enhancing Robustness of Interdependent Network by Adding Connectivity and Dependence Links // Physica A. – 2018. – Vol. 497. – P. 185–197.

50.*Xu, X., Fu, X.* Analysis on Cascading Failures of Directed-Undirected Interdependent Networks with Different Coupling Patterns // Entropy. – 2023. – Vol .25, no. 3. – Art. no. e471.

51.*Yang, X.H., Feng, W.H., Xia, Y.,* et al. Improving Robustness of Interdependent Networks by Reducing Key Unbalanced Dependency Links // IEEE Transactions on Circuits and Systems II: Express Briefs. – 2020. – Vol. 67, no. 12. – P. 3187–3191.

52.*Shiroky, A., Kalashnikov, A.* Mathematical Problems of Managing the Risks of Complex Systems under Targeted Attacks with Known Structures // Mathematics. – 2021. – Vol. 9, no. 19. – Art. no. e2468.

53.*Shiroky, A., Kalashnikov, A.* Influence of the Internal Structure on the Integral Risk of a Complex System on the Example of the Risk Minimization Problem in a «Star» Type Structure // Mathematics. – 2023. – Vol. 11, no. 4. – Art. no. e998.

54.*Широкий А.А., Калашников А.О.* Применение методов естественных вычислений для управления рисками сложных систем // Проблемы упраления. – 2021. – № 4. – С. 3–20. [*Shiroky, A.A., Kalashnikov, A.O.* Natural Computing with Application to Risk Management in Complex Systems // Control Sciences. – 2021. – No. 4. – P. 2–17.]

55.*Shiroky, A.A.* A Method for Rapid Risk Assessment of a Fog Computing System with a Star–Shaped Topology // Proceedings of 17th International Conference Management of

Large–Scale System Development (MLSD). – Moscow, Russia, 2024. – P. 1–5.
56. *Shiroky, A.A.* Risk Management in the Design of Computer Network Topology / In: Lecture Notes in Computer Science. Ed. by V.M. Vishnevskiy, K.E. Samouylov, and D.V.Kozyrev. – Cham: Springer, 2024. – Vol. 14123. – P. 375–385.
57. *Shiroky, A.A.* Risk Management in the Design of Security Systems with Nested Security Zones // Proceedings of 16th International Conference Management of Large-Scale System Development (MLSD). – Moscow, Russia, 2023. – P. 1–4.

Статья представлена к публикации членом редколлегии В. Н. Бурковым.

Поступила в редакцию 06.11.2024,
после доработки 21.03.2025.
Принята к публикации 21.03.2025.

Широкий Александр Александрович – канд. физ.-мат. наук,
✉ shiroky@ipu.ru
ORCID ID: https://orcid.org/0000–0002–9130–5541

Калашников Андрей Олегович – д-р техн. наук,
✉ aokalash@ipu.ru
ORCID ID: https://orcid.org/0000–0001–5204–1398

Институт проблем управления им. В.А. Трапезникова РАН, г. Москва.

HOW DOES THE INTERNAL STRUCTURE OF A COMPLEX SYSTEM INFLUENCE ITS OVERALL RISK? RISK MINIMIZATION FOR TREES

A. A. Shiroky* and A. O. Kalashnikov**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ shiroky@ipu.ru, **✉ aokalash@ipu.ru

Abstract. The Defender–Attacker problem is often employed as a mathematical framework in risk management. In this problem, the above players with opposite goals allocate limited resources to system elements to minimize or maximize a risk function. It has been well-studied under the assumption of independent system elements. However, in complex systems, elements interact, causing significant differences between the measured and predicted risks. Although models with the interdependence of system elements are regularly considered in the literature, no comprehensive understanding has been formed of how the structure of a complex system influences its overall risk. We address this issue in a series of papers by investigating system structures of increasing complexity. Chains and stars have been analyzed previously; in this paper, the findings are extended to arbitrary trees. We optimize the placement of elements within a tree to minimize risk; derive upper bounds for the relative error of an approximate algorithmic solution of this problem for trees with a few branches and leaves; and explore the dynamics of these bounds when increasing the number of leaves and branches. As demonstrated, the resulting upper bounds do not exceed their counterparts for stars from the previous works.

Keywords: complex systems, risk, system structure, risk management, risk minimization algorithms, the problem of optimal element placement.

УДК 519.7

МЕТОД ФУНКЦИОНАЛЬНОГО ДИАГНОСТИРОВАНИЯ ДИСКРЕТНО-СОБЫТИЙНЫХ СИСТЕМ НА ОСНОВЕ МОДЕЛИ НЕЧЕТКОГО КОНЕЧНОГО АВТОМАТА[#]

А. Е. Шумский*, А. Н. Жирабок**

*,**Дальневосточный федеральный университет, г. Владивосток

*✉ a.e.shumsky@yandex.com, **✉ zhirabok@mail.ru

Аннотация. Рассматривается задача функционального диагностирования дискретно-событийных систем ответственного назначения, описываемых моделью нечеткого конечного автомата. Предлагается метод решения задачи, особенностью которого является использование математического аппарата нечеткой логики. Описываются операции нечеткой логики и вводится понятие детерминизатора нечеткого конечного автомата. Приводится схема диагностирования, позволяющая формировать структурированный вектор невязки. Схема содержит ряд каналов (по числу возможных дефектов в системе). В основе каждого канала лежит наблюдатель в виде детерминизатора специального нечеткого конечного автомата, одновременно учитывающего возможность как правильного перехода автомата при исправном функционировании системы, так и неправильного перехода, обусловливаемого дефектом в системе. Другой частью канала является блок принятия решения. Предлагаются способы синтеза наблюдателя и блока принятия решения. Особенности применения метода иллюстрируются на примере задачи мониторинга ошибок операторов IT-системы.

Ключевые слова: дискретно-событийные системы, нечеткая логика, нечеткие конечные автоматы, детерминизатор, функциональное диагностирование, IT-системы, мониторинг.

ВВЕДЕНИЕ

Удовлетворение жестких требований, предъявляемых к надежности и отказоустойчивости современных сложных систем ответственного назначения, требует осуществления оперативного диагностирования (обнаружения и поиска) дефектов, возникающих в процессе функционирования систем, в целях последующего парирования или устранения обнаруженных дефектов. В настоящей статье рассматривается задача функционального диагностирования дефектов в так называемых дискретно-событийных системах (ДСС). К ДСС относятся не только системы, естественным образом попадающие в этот класс (например, цифровые системы обработки информации и управления).

Многие системы, традиционно относящиеся к непрерывным (как то: физические, технические (в том числе человеко-машинные) и социально-экономические), могут рассматриваться как ДСС на верхнем уровне своей иерархии. Отличительные особенности ДСС состоят в следующем [1]:

- Дискретно-событийные системы являются дискретными по времени и значениям переменных состояния.
- Пространство состояний ДСС конечно; состояния могут иметь следующий смысл: ДСС простаивает, ДСС работает в определенном режиме, ДСС неисправна, ДСС восстанавливается и т. п.
- Функционирование ДСС определяется событиями, которые могут быть следствиями выполнения различных команд, например: «приступить к работе», «изменить режим работы», «осуществить диагностику ДСС», «приступить к восстановлению ДСС», «завершить восстановление ДСС», «завершить работу ДСС» и т. п.

[#] Работа выполнена при финансовой поддержке Минобрнауки России (проект FZNS-2023-0011).

• Дискретно-событийная система, как правило, ведет себя случайным образом, что связано с возможностью реализации разных переходов из одного состояния в различные состояния, инициируемых одним и тем же событием.

Обобщенная схема диагностирования и отказоустойчивого управления ДСС представлена на рис. 1. В соответствии с этой схемой контроллер генерирует внешние (по отношению к ДСС) события (входы ДСС или команды). В качестве ответа ДСС формирует внутренние (по отношению к ДСС) события, рассматриваемые как выходы ДСС. Контроллер отслеживает выходы ДСС, а также диагноз, определяемый системой диагностирования (СД), с целью формирования нового внешнего события (входа ДСС). В свою очередь, СД отслеживает как внешние, так и внутренние события для определения диагноза. Последний дает ответ на вопрос: исправна ли ДСС? Если ответ отрицательный, то делается дополнительное заключение о виде дефекта. Если дефект обнаружен и классифицирован, контроллер сообщает о новом событии, реакция на которое предполагает парирование дефекта (формирование последовательности команд, позволяющей исключить влияние дефекта на достижение цели) или устранение дефекта (ремонт ДСС).

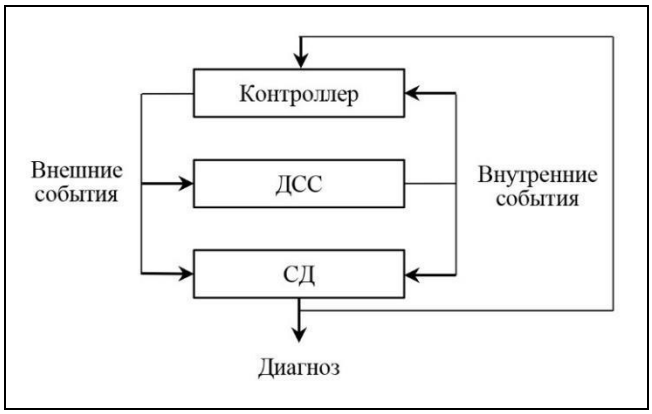

Рис. 1. Схема диагностирования и отказоустойчивого управления ДСС

Достаточно полный обзор существующих методов диагностирования ДСС можно найти в статьях [2, 3] и монографии [4]. В обзоре [3] отмечается, что для описания ДСС наиболее часто привлекаются следующие математические модели:

– детерминированные конечные автоматы,

– вероятностные конечные автоматы и марковские цепи,

– сети Петри.

Настоящая статья фокусируется на применении исключительно моделей конечного автомата (КА). Использование подобных моделей для решения диагностических задач применительно к ДСС предполагает, что входы и выходы КА формируются как наблюдаемые события (внешние и внутренне соответственно), в то время как возникновение дефекта в ДСС рассматривается как непосредственно ненаблюдаемое внутреннее событие.

В тех случаях, когда детерминированная конечноавтоматная модель ДСС отсутствует, помимо модели в виде вероятностного КА также могут использоваться модели недетерминированного и нечеткого КА (НДКА и НЧКА соответственно). Применение недетерминированных и нечетких моделей (по сравнению с вероятностными) позволяет значительно сократить объем необходимых вычислений, что приводит к увеличению быстродействия процесса диагностирования. Модели НДКА и НЧКА учитывают ситуацию, когда для некоторого состояния и некоторого входа автомата могут быть реализованы переходы в различные последующие состояния. При этом, если в случае НДКА нельзя отдать приоритет возможности реализации того или иного перехода, то в случае НЧКА, пользуясь дополнительной (например, статистической) информацией, экспертными оценками, результатами обучения и т. п., можно говорить о степени уверенности в осуществлении каждого из возможных переходов.

Для диагностирования ДСС, описываемых моделью НДКА, в статьях [5] и [6] были представлены разработанные авторами методы, основанные на использовании соответственно парной алгебры разбиений и парной алгебры покрытий [7].

Целью настоящей статьи является разработка нового метода диагностирования на основе модели НЧКА. Для этого привлекаются математические конструкции нечеткой логики [8], а также понятия нечеткого автомата [9] и его детерминизатора [10].

Иллюстрация особенностей применения разработанного метода осуществляется на основе модели процесса управления изменениями в IT-системе, ранее рассмотренной в работах [5] и [6] применительно к задаче диагностирования НДКА.

1. РЕШАЕМЫЕ ЗАДАЧИ И СТРУКТУРА СТАТЬИ

1.1. Используемые модели

Пусть при исправной работе ДСС описывается НЧКА-моделью вида

$$A=\left(U, X, Y, \delta, \lambda, x(0)\right), \qquad (1)$$

где $U=\{u_1, u_2, \ldots, u_m\}$, $X=\{x_1, x_2, \ldots, x_n\}$ и $Y=\{y_1, y_2, \ldots, y_l\}$ – конечные множества входов,

состояний и выходов соответственно; $x(0) \in X$ – известное начальное состояние; $\delta : X \times U \to \mu(X)$ – нечеткая функция переходов; $\mu(X) \in \{\mu(x_i) \in [0,1],\ 1 \le i \le n\}$ – нечеткое множество; $\lambda : X \to Y$ – функция выхода.

Для нечеткой функции переходов δ будем использовать матричное задание, при котором переходы, осуществляемые под воздействием входа u_k, описываются матрицей M^k размера $n \times n$, элемент которой $M^k_{i,j} \in [0,1]$ характеризует степень уверенности в том, что автомат A из состояния $x_i \in X$ под воздействием входа $u_k \in U$ перейдет в состояние $x_j \in X$. Обозначим через $S(U)$ множество всех матриц $M^k, 1 \le k \le m$.

Если обоснованное задание степени уверенности в возможности осуществления переходов невозможно, предлагается поступать следующим образом: элемент матрицы $M^k_{i,j}$, соответствующий допустимому переходу, задается равным единице, в то время как элемент этой матрицы, соответствующий недопустимому переходу, принимается равным нулю. Тем самым осуществляется переход от НЧКА-модели ДСС к ее НДКА-модели.

Для задания функции выхода λ будем использовать матрицу L размера $l \times n$, элемент которой $L_{i,j} = 1$, если выход $y_i \in Y$ формируется автоматом A в состоянии $x_j \in X$, в противном случае $L_{i,j} = 0$. Обозначим через $S(Y)$ множество всех строк $\{L_i,\ 1 \le i \le l\}$ матрицы L.

Пусть $X_{i,k}$, $X_{i,k} \subseteq X$, – множество всех состояний, достижимых из состояния $x_i \in X$ под воздействием входа $u_k \in U$. Предположим, что дефект $f_s, 1 \le s \le N$, в ДСС на модели (1) может быть представлен искажением функции перехода δ таким, что вместо допустимого перехода из состояния x_i в состояние $x_j \in X_{i,k}$ под воздействием входа $u_k \in U$ реализуется недопустимый переход в состояние $x_t \notin X_{i,k}$. Этот факт будем отражать следующим образом: $f_s : (x_j \to x_t)_{i,k}$. При этом матрицу M^k ДСС с дефектом получим изменением значения элемента $M^k_{i,t}$ матрицы M^k исправной ДСС с нуля на единицу. Для упрощения дальнейшего изложения будем придерживаться гипотезы об однократных дефектах из заранее заданного списка $F = \{f_1, f_2 \ldots, f_N\}$. Обозначим через A_s вспомогательный автомат, функция перехода δ_s которого получена изменением (способом, как было указано выше) функции перехода δ автомата A на переход, обусловливаемый дефектом f_s:

$$A_s = (U, X, Y, \delta_s, \lambda, x(0)). \qquad (2)$$

Модель (2) в силу особенностей построения матрицы M^k учитывает возможность «правильного» перехода при исправной работе ДСС, но также допускает и возможность «неправильного» перехода в состояние, обусловленное дефектом f_s в ДСС. Как следствие, можно говорить об определенном соответствии между поведением модели (2) и поведением как исправной ДСС, так и ДСС с дефектом. Способ оценки этого соответствия в терминах возможности (уверенности) предлагается в п. 3.2 и лежит в основе формирования структурированного вектора невязок.

1.2. Схема диагностирования

Для обнаружения и поиска дефектов предлагается использовать схему диагностирования, приведенную на рис. 2. Эта схема содержит N каналов (по числу возможных дефектов в ДСС), где каждый из каналов включает детерминированный КА A^d_s и блок принятия решения БПР_s. Синтез схемы диагностирования предполагает нахождение составляющих каждого канала. Необходимость решения этих задач определяет следующую структуру статьи.

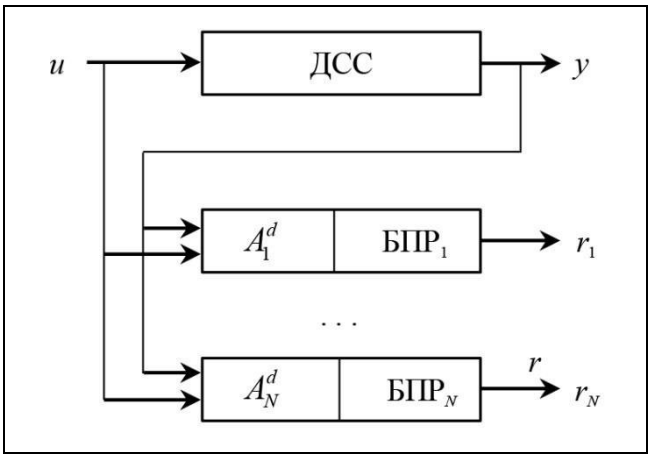

Рис. 2. Схема диагностирования ДСС

1.3. Структура статьи

Статья структурирована следующим образом. В § 2 приводятся те конструкции математического аппарата нечеткой логики, которые потребуются для получения основных результатов статьи. В § 3 решение задачи синтеза ДКА A_s^d, называемого наблюдателем НДКА A_s, сводится к модифицированной задаче нахождения детерминизатора [10] этого автомата. Далее в этом же параграфе предлагается правило функционирования БПР$_s$, $1 \le s \le N$, обеспечивающее свойство структурированности вектора невязки. Суть этого свойства состоит в следующем. Равенство нулю всех компонент вектора невязки означает отсутствие дефектов в ДСС. Равенство нулю только одной из компонент вектора невязки, в то время как остальные компоненты равны единице, соответствует случаю возникновения дефекта в ДСС, номер которого совпадает с номером нулевой компоненты вектора невязки. В § 4 рассматривается пример и приводятся результаты моделирования, иллюстрирующие особенности предложенного метода. В § 5 содержатся основные выводы по статье.

2. МАТЕМАТИЧЕСКИЕ КОНСТРУКЦИИ

2.1. Операции нечеткой логики

Напомним ряд операций нечеткой логики [8], играющих важную роль для дальнейшего изложения. Нечеткая матрица $B = \{B_{ij}\}$ – это матрица с элементами $B_{ij} \in [0,1]$. Следовательно, введенные выше матрицы M^k, $1 \le k \le m$, – суть нечеткие матрицы. Пусть теперь B и C – нечеткие матрицы размера $a \times b$ и $b \times c$ соответственно. Произведение нечетких матриц определяется формулой [10]

$$(BC)_{ij} = \max_{i,j} \min \left(B_{ih} \; C_{hj} \right),$$

где B_{ih} и C_{hj} – соответствующие элементы матриц B и C, $1 \le i \le a$, $1 \le h \le b$, $1 \le j \le c$. Эта формула является обобщением известной формулы умножения матриц [11, *с. 24*]; она получается заменой произведения элементов матриц операцией нахождения минимума и суммы элементов – операцией нахождения максимума. Рассмотрим простой пример.

Пример 1. Пусть матрицы B и C имеют вид:

$$B = \begin{pmatrix} 0,1 & 0,9 \\ 0,7 & 0,2 \end{pmatrix}, \; C = \begin{pmatrix} 0,6 & 0,3 \\ 0,4 & 0,8 \end{pmatrix}.$$

После проведенных вычислений получим: $(BC)_{11} = \max\left(\min(0,1 \;\; 0,6), \min(0,9 \;\; 0,4)\right) = 0,4$, $(BC)_{12} = \max\left(\min(0,1 \;\; 0,3), \min(0,9 \;\; 0,8)\right) = 0,8$, $(BC)_{21} = \max\left(\min(0,7 \;\; 0,6), \min(0,2 \;\; 0,4)\right) = 0,6$, $(BC)_{22} = \max\left(\min(0,7 \;\; 0,3), \min(0,2 \;\; 0,8)\right) = 0,3$. В результате произведение нечетких матриц B и C принимает вид:

$$BC = \begin{pmatrix} 0,4 & 0,8 \\ 0,6 & 0,3 \end{pmatrix}. \blacklozenge$$

2.2. Понятие детерминизатора нечеткого конечного автомата

Определение детерминизатора НЧКА [10] вводится следующим образом. Пусть E – некоторое множество нечетких векторов размера $1 \times n$, компоненты которых могут принимать значения на интервале $[0,1]$. Пусть также E^0, $E^0 \subseteq E$, – множество всех единичных векторов размера $1 \times n$, называемых порождающими состояниями детерминизатора $D(A')$. Замыкание $[E]_\Sigma$ множества E относительно сигнатуры (множества допустимых операций) Σ есть множество всех векторов (включающее сами векторы множества E), которые можно получить, применяя операции из сигнатуры Σ к векторам из множества E. Для построения замыкания предлагается использовать следующую процедуру:

1. Положить $i = 0$.
2. Найти множество векторов $E^{i+1} = E^i \cup \left[E^i \right]_\Sigma$.
3. Если $E^{i+1} = E^i$, положить $[E]_\Sigma = E^i$. Конец.
4. Иначе положить $i = i + 1$ и перейти к п. 2.

Пусть $A' = (U, X, \delta)$ – нечеткий полуавтомат (т. е. автомат A без функции выхода), $S(U) = \{M^k, 1 \le k \le m\}$ – множество нечетких матриц перехода автомата A и сигнатура Σ включает все операции, предполагающие умножение вектора размерности $1 \times n$ справа на матрицу из множества $S(U)$. Детерминизатором нечеткого полуавтомата A' называется ДКА, описываемый тройкой

$$D(A') = (S(X), U, \Delta),$$

где $S(X) = [E]_{S(U)}$, а $\Delta : S(X) \times S(U) \to S(X)$ – функция перехода детерминизатора, определяемая соотношением

$$\Delta(\mu_i, u_k) = \mu_i \cdot M^k, \; \mu_i \in S(X), \; M^k \in S(U). \quad (3)$$

Для пояснения формулы (3) напомним, что матрица M^k описывает переход, активируемый входом u_k.

Пример 2. Пусть полуавтомат A' задается нечеткими матрицами перехода

$$M^1 = \begin{pmatrix} 0,4 & 0,6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \; M^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Зададим порождающие состояния детерминизатора: $\mu_1 = (1 \;\; 0 \;\; 0)$, $\mu_2 = (0 \;\; 1 \;\; 0)$, $\mu_3 = (0 \;\; 0 \;\; 1)$. Осуществляя необходимые вычисления согласно формуле (3), найдем векторы μ_4, μ_5 и μ_6 замыкания $[E]_{S(U)}$: $\mu_4 = \mu_1 \cdot M^1 = (0,4 \;\; 0,6 \;\; 0)$, $\mu_5 = \mu_4 \cdot M^1 = (0,4 \;\; 0,4 \;\; 0,6)$, $\mu_6 = \mu_4 \cdot M^2 = (0,6 \;\; 0 \;\; 0,4)$. Рассуждая аналогичным образом, получим: $\mu_7 = (0,4 \;\; 0,6 \;\; 0,4)$, $\mu_8 = (0,6 \;\; 0,4 \;\; 0,4)$, $\mu_9 = (0 \;\; 0,4 \;\; 0,6)$, $\mu_{10} = (0 \;\; 0 \;\; 0,4)$, $\mu_{11} = (0 \;\; 0,4 \;\; 0)$, $\mu_{12} = (0,4 \;\; 0 \;\; 0)$, $\mu_{13} = (0,4 \;\; 0,4 \;\; 0)$, $\mu_{14} = (0,4 \;\; 0,4 \;\; 0,4)$, $\mu_{15} = (0,4 \;\; 0 \;\; 0,4)$, $\mu_{16} = (0 \;\; 0,4 \;\; 0,4)$. Окончательно, принимая $M(X) = \{\mu_i, 1 \le i \le 16\}$, построим таблицу переходов детерминизатора $D(A')$ (табл. 1). В этой таблице, например, указаны переходы из состояния μ_9 в состояние μ_{10} под воздействием входа u_1 и из состояния μ_9 в состояние μ_4 под воздействием входа u_2, поскольку $\mu_9 \cdot M^1 = \mu_{10}$ и $\mu_9 \cdot M^2 = \mu_4$.

Заметим, что число состояний детерминизатора $D(A')$ существенно превосходит число состояний исходного полуавтомата A'. Это связано с тем, что при построении детерминизатора не происходит потери информации о степени уверенности в осуществлении каждого из возможных переходов, поскольку каждое состояние детерминизатора – это вектор, компоненты которого равны возможностям перехода ДСС в соответствующие свои состояния в определенный момент времени. ♦

Для получения верхней граничной оценки числа состояний детерминизатора учтем, что компоненты векторов из множества $S(X)$ могут принимать только те значения, которые содержатся в матрицах из множества $S(U)$. Исключая из рассмотрения нулевой вектор, получим формулу:

$$\#S(X) \le q^n - 1, \qquad (4)$$

где символ # обозначает мощность множества; q – число различных значений элементов нечетких матриц из множества $S(U)$; n, как и ранее, – число состояний НЧКА. Так, для системы (3) имеем: $q = 4$ (множество различных значений элементов матриц M^1 и M^2 включает числа $\{0;\; 0,4;\; 0,6;\; 1\}$), $n = 3$ и, как следствие, $\#S(X) \le 4^3 - 1 = 63$.

Пример 3. Описанный выше подход также может быть применен и для детерминизации НДКА. Проиллюстрируем это на примере модели НДКА, полученной из НЧКА-модели (3) посредством задания единичных значений степени уверенности для всех допустимых переходов:

$$M^1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \; M^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \qquad (5)$$

Отметим, что рассматриваемый подход и в случае НДКА не гарантирует получение детерминизатора, размерность которого меньше размерности исходного автомата. Действительно, так как для НДКА $q = 2$, то согласно выражению (4) число состояний детерминизатора будет определяться формулой

$$\#S(X) \le 2^n - 1.$$

Таблица 1

Таблица переходов детерминизатора $D(A)$

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{16}
u_1	μ_4	μ_3	μ_3	μ_5	μ_5	μ_7	μ_5	μ_7	μ_{10}	μ_{10}	μ_{10}	μ_{13}	μ_{14}	μ_{14}	μ_{14}	μ_{10}
u_2	μ_3	μ_1	μ_2	μ_6	μ_7	μ_9	μ_8	μ_5	μ_4	μ_{11}	μ_{12}	μ_{10}	μ_{15}	μ_{14}	μ_{16}	μ_{13}

В частности, для НДКА (5) имеем $\#S(X) \le 2^3 - 1 = 7$. Опуская промежуточные выкладки, приведем таблицу переходов детерминизатора НДКА (5).

Таблица 2

Таблица переходов детерминизатора НДКА (5)

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
u_1	μ_4	μ_3	μ_3	μ_5	μ_5	μ_5	μ_3
u_2	μ_3	μ_1	μ_2	μ_6	μ_5	μ_7	μ_4

В табл. 2 состояния детерминизатора таковы: $\mu_1 = (1\ 0\ 0)$, $\mu_2 = (0\ 1\ 0)$, $\mu_3 = (0\ 0\ 1)$, $\mu_4 = (1\ 1\ 0)$, $\mu_5 = (1\ 1\ 1)$, $\mu_6 = (1\ 0\ 1)$, $\mu_7 = (0\ 1\ 1)$. ♦

Заметим, что алгебраический подход к детерминизации НДКА (алгебра разбиений и алгебра покрытий), описанный соответственно в работах [5] и [6], гарантирует нахождение ДКА, размерность которого не превышает размерности исходного автомата. Это обусловливается тем, что в рамках алгебраического подхода не ставится задача сохранения информации о степени уверенности в осуществлении каждого из возможных переходов. Рассматриваемый нечеткий подход позволяет использовать при диагностировании дополнительную полезную информацию о степени уверенности в осуществлении каждого из возможных переходов, что предоставляет потенциальную возможность для увеличения (при необходимости) глубины поиска дефектов. Плата за это – значительная размерность таблицы переходов детерминизатора.

Основное отличие описания ДКА $A_s^d, 1 \le s \le N$, из схемы диагностирования, приведенной на рис. 2, от описания детерминизатора $D(A_s')$ соответствующего полуавтомата A_s' состоит в том, что функция переходов автомата A_s^d дополнительно зависит от выходов исходного автомата. Это позволяет использовать дополнительную информацию для корректировки поведения и снижения размерности автомата A_s^d. Проводя аналогию с наблюдателями непрерывной динамической системы, ДКА A_s^d назовем наблюдателем НЧКА A_s.

3. СИНТЕЗ КАНАЛА ДИАГНОСТИРОВАНИЯ

3.1. Синтез наблюдателя A_S^d

Дальнейшее изложение привязано к каналу схемы диагностирования, ориентированному на поиск дефекта f_s. Рассмотрим сначала способ нахождения функции перехода δ_s^d наблюдателя A_s^d, основанный на небольшой модификации соотношения (3). Для векторов $\mu_i \in S(X)$ и $L_j \in S(Y)$ введем следующее обозначение:

$$\langle \mu_i, L_j \rangle = \big(\min(\mu_{i,1}, L_{j,1}),\ \min(\mu_{i,2}, L_{j,2}), \ldots, \min(\mu_{i,n}, L_{j,n})\big). \quad (6)$$

Функцию перехода $\Delta: S(X) \times S(Y) \times S(U) \to S(X)$ ДКА A_s^d зададим следующим образом:

$$\Delta(\mu_i, y_j, u_k) = \langle \mu_i, L_j \rangle \cdot M^k, \quad \mu_i \in S(X),\ L_j \in S(Y),\ M^k \in S(U). \quad (7)$$

Использование $\langle \mu_i, L_j \rangle$ в формуле (7) вместо вектора μ_i в выражении для функции перехода (3) детерминизатора позволяет уточнить значения функций принадлежности компонент вектора $x \in X$ соответствующим нечетким множествам до осуществления перехода.

Пример 4. Пусть НЧКА при отсутствии дефектов описывается матрицами M^1 и M^2 из примера 2. Пусть также функция выхода этого автомата задается матрицей

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Ошибки переходов автомата являются следствием двух дефектов: $f_1 : (x_3 \to x_1)_{1,2}$ и $f_2 : (x_3 \to x_2)_{1,2}$. Матрицы M_1^2 и M_2^2 автомата, включающие дополнительный ошибочный переход, обусловленный дефектом (нижний индекс при матрице указывает на номер дефекта), имеют вид

$$M_1^2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},\ M_2^2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Матрица M^1 не подвержена воздействию дефекта и сохраняет свой вид.

Проиллюстрируем синтез наблюдателя A_1^d (в этом случае используется только матрица M_1^2). Как и в предыдущем примере, примем порождающие состояния $\mu_1 = (1\ \ 0\ \ 0)$, $\mu_2 = (0\ \ 1\ \ 0)$, $\mu_3 = (0\ \ 0\ \ 1)$. Осуществляя вычисления согласно правой части соотношения (7), получим:

$$\langle \mu_1, L_1 \rangle \cdot M^1 = (\min(1,1)\ \ \min(0,0)\ \ \min(0,0)) \times$$

$$\times \begin{pmatrix} 0{,}4 & 0{,}6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (0{,}4\ \ 0{,}6\ \ 0) = \mu_4.$$

Аналогично можно найти $\mu_5 = (1\ \ 0\ \ 1)$, $\mu_6 = (0{,}4\ \ 0{,}4\ \ 0)$, $\mu_7 = (0\ \ 0\ \ 0{,}6)$, $\mu_8 = (0{,}4\ \ 0\ \ 0{,}4)$, $\mu_9 = (0{,}6\ \ 0\ \ 0)$, $\mu_{10} = (0\ \ 0\ \ 0{,}4)$, $\mu_{11} = (0{,}4\ \ 0\ \ 0)$, $\mu_{12} = (0\ \ 0{,}6\ \ 0)$, $\mu_{13} = (0\ \ 0{,}4\ \ 0)$, $\mu_{14} = (0{,}6\ \ 0\ \ 0{,}6)$ и построить таблицу переходов (табл. 3) наблюдателя A_1^d.

В табл. 3 прочерком обозначены переходы, соответствующие несовместимым значениям состояния наблюдателя и выхода ДДС. Например, состояние наблюдателя μ_2 допускает только выход y_2 ДСС, что непосредственно следует из записи вектора μ_2 и матрицы L. Следовательно, сочетание μ_2 и y_1 невозможно при исправной работе ДСС.

Сравнивая размерность детерминизатора $D(A')$ (см. табл. 1) и размерность наблюдателя A_1^d (см. табл. 3), несложно видеть, что размерность наблюдателя не превышает размерность детерминизатора. Однако при этом размерность наблюдателя на практике может превышать размерность исходного НЧКА. ♦

Казалось бы, последний факт должен накладывать ограничения на возможность практической реализации процесса диагностирования ДСС на основе модели НЧКА, обусловливаемые значительной размерностью реальных систем. Эти ограничения представляются менее выраженными при использовании алгебраического подхода [5, 6]. Однако в действительности это не так. Дело в том, что применение алгебраического подхода ориентировано на нахождение табличного описания функции переходов наблюдателей (ДКА A_i^d, $1 \le i \le N$) на основе табличного описания исходной конечноавтоматной модели ДСС. Предлагаемый же подход ориентирован на задание функции переходов в компактном аналитическом виде, что делает его малочувствительным к росту размерности исходной модели ДСС. Более того, в силу отмеченной особенности предлагаемый подход представляется предпочтительным и в случае синтеза схем диагностирования ДСС на основе ДКА- и НДКА-моделей. При его применении преодолевается так называемое «проклятие размерности», характерное для всех методов, ориентированных на табличное (или в виде графов) описание конечноавтоматной модели ДСС.

3.2. Построение блока принятия решения канала диагностирования

Предположим, что при диагностировании ДСС одновременно генерируются выход ДСС y_j и состояние μ_i наблюдателя A_s^d. Обозначим через σ_s значение максимальной компоненты вектора $\langle \mu_i, L_j \rangle$. Это значение будем рассматривать как степень уверенности в том, что поведение ДСС соответствует поведению наблюдателя A_s^d. Сказанное является основой для задания отношения Ψ_s:

$$(\mu_i, y_j) \in \Psi_s \Leftrightarrow \sigma_s \ne 0. \qquad (8)$$

Пусть теперь отношение Ψ_s построено согласно выражению (8) и его проверка для конкретной пары y_j и μ_i показывает, что оно не выполняется (т. е. имеет место равенство $\sigma_s = 0$). Это будет свидетельствовать о том, что в ДСС произошла ошибка (дефект) f_k, $k \ne s$. Формируемое при этом значение невязки $r_s = 1$. В противном случае $(\sigma_s \ne 0)$ формируется $r_s = 0$. Для того, чтобы сде-

Таблица 3

Таблица переходов наблюдателя A_1^d

		μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}	μ_{13}	μ_{14}
u_1	y_1	μ_4	–	–	μ_6	μ_4	μ_6	–	μ_6	μ_4	–	μ_6	–	–	μ_4
	y_2	–	μ_3	μ_3	μ_7	μ_3	μ_{10}	μ_7	μ_{10}	–	μ_{10}	–	μ_7	μ_{10}	μ_7
u_2	y_1	μ_5	–	–	μ_8	μ_5	μ_8	–	μ_8	μ_{14}	–	μ_8	–	–	μ_{14}
	y_2	–	μ_1	μ_2	μ_{12}	μ_2	μ_{11}	μ_{12}	μ_{13}	–	μ_{13}	–	μ_{11}	μ_{13}	μ_{12}

лать однозначное заключение о виде дефекта в ДСС на основе структурированного вектора невязок, только одна его компонента, номер которой соответствует номеру дефекта, должна сохранять нулевое значение. Если это не так (т. е. сформировано несколько нулевых невязок), то требуются дополнительные проверки, например, с использованием дополнительных измерений и (или) специальных тестов [12]. При этом порядок проведения проверок целесообразно осуществлять в направлении дефектов, номера которых следуют в порядке уменьшения степени уверенности из списка $\{\sigma_1, \sigma_2, ..., \sigma_N\}$.

4. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

В работах [5, 6] приведено подробное описание процесса управления изменениями IT-системы ([5], *рис. 1*) и рассмотрен способ получения его ДКА- и НДКА-моделей. Отметим, что процесс управления изменениями является одним из важнейших для IT-систем: он отвечает за управление жизненным циклом всех изменений и способствует реализации полезных изменений с минимальным прерыванием предоставления IT-услуг. Субъектами процесса являются:

– инициатор – представитель департамента IT, осуществляющий первичную обработку, назначение и контроль над ходом выполнения изменений;

– исполнитель – инженер, производящий изменения в элементах конфигурации или координирующий работы подрядчика по этим изменениям;

– консультативный комитет по изменениям (КИ) – консультативный орган, регулярно собирающийся для оценки и планирования изменений;

– менеджер процесса – представитель департамента IT, осуществляющий контроль процесса управления изменениями и формирующий предложения по его улучшению.

В этих же работах приведены таблицы переходов и выходов ДКА- и НДКА-моделей процесса. Модель ДКА описывает действия субъектов процесса в полном соответствии с предписанным регламентом. Необходимость использования исходной модели в виде НДКА обусловливается тем, что на практике допустимы некоторые (некритичные) отклонения действия субъектов от заданного регламента, направленные на уточнение самого регламента либо уточнение их знаний о регламенте. В частности, рассматривались следующие ситуации:

– менеджер процесса управления изменениями отправляет результат проверки выполненного задания на повторное согласование в комитет КИ;

– менеджер процесса управления изменениями отправляет полученное согласованное задание обратно инициатору;

– исполнитель отправляет полученное согласованное задание обратно инициатору;

– исполнитель отправляет полученное и согласованное задание обратно на согласование в комитет КИ.

В качестве входов модели рассматриваются следующие внешние события: u_1 – завершение формирования плана работ; u_2 – утверждение плана КИ; u_3 – передача не утвержденного КИ плана на доработку; u_4 – передача задания и плана работ исполнителю; u_5 – закрытие работ; u_6 – занесение внесенных изменений в библиотеку состояния IT-системы. В качестве состояний модели рассматриваются следующие этапы регламента: x_1 – формирование задания и плана реализации; x_2 – согласование задания и плана работ КИ; x_3 – согласование задания и плана работ менеджером процесса; x_4 – выполнение работ исполнителем; x_5 – проверка менеджером процесса выполненных работ; x_6 – работы завершены. В качестве выходов используется доступная информация о некоторых этапах выполнения регламента (функция выхода приводится ниже).

Описывающая безошибочную работу участников процесса (инициатор, исполнитель и менеджер) НЧКА-модель процесса управления изменениями IT-системы может быть получена на основе НДКА-модели ([6], *табл. 3*) и имеющейся статистической информации о возможных действиях участников при выполнении регламента:

$$M^1 = \begin{pmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^2 = \begin{pmatrix} 0 & 0 & & 0 & 0 & & 0 \\ 0 & 0 & & 1 & 0 & \cdots & 0 \\ 0 & 0 & \vdots & 0 & 0 & & 0 \\ 0 & 0 & & 0 & 0 & \ddots & \vdots \\ 0 & 0 & & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^3 = \begin{pmatrix} 0 & 0 & 0 & & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0{,}1 & 0{,}1 & 0 & 0{,}8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0,2 & 0,1 & 0 & 0 & 0,7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M^6 = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & & 0 & 0 \end{pmatrix}.$$

Все вышеприведенные матрицы имеют размерность 6×6. Функция выхода описывается матрицей

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Как и в работе [6], будем рассматривать ошибки инициатора, исполнителя и менеджера процесса соответственно: $f_1 : (x_2 \to x_3)_{(x_1,u_1)}$, $f_2 : (x_5 \to x_6)_{(x_4,u_5)}$, $f_3 : (x_3 \to x_4)_{(x_2,u_2)}$. Этим ошибкам соответствуют матрицы

$$M_1^1 = \begin{pmatrix} 0 & 1 & 1 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M_2^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0,2 & 0,1 & 0 & 0 & 0,7 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M_3^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Все остальные матрицы свои значения сохраняют.

В приведенных ниже таблицах представлены результаты моделирования, полученные для различных допустимых результатов работы субъектов процесса управления изменениями, а также ошибок инициатора, исполнителя и менеджера процесса. Генерация невязок и вспомогательных переменных осуществлялась на основе соотношений (6)–(8) с использованием приведенных выше матриц. Для первого, второго и третьего каналов используются матрицы M_1^1, M^2, M^3, M^4, M^5, M^6; M^1, M^2, M^3, M^4, M_2^5, M^6 и M^1, M_3^2, M^4, M^5, M^6 соответственно. При моделировании авторы ограничились следующими сценариями.

Сценарий 1. Регламент процесса реализуется без каких-либо отклонений (табл. 4).

Сценарий 2. Регламент процесса выполняется с допустимым отклонением, обусловленным передачей менеджером задачи формирования задания и плана работ для повторного согласования в КИ (табл. 5).

Сценарий 3. Регламент процесса выполняется с ошибками, допускаемыми соответственно инициатором, исполнителем и менеджером процесса (табл. 6).

Таблица 4

Сценарий 1

Характеристики		Начальное значение	Вход				
			u_1	u_2	u_4	u_5	u_6
Система	Состояние x	x_1	x_2	x_3	x_4	x_5	x_6
	Выход y	y_1	y_2	y_3	y_4	y_2	y_5
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{16}
	Уверенность σ_1	1	1	1	0,8	0,7	0,7
	Невязка r_1	0	0	0	0	0	0
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{26}
	Уверенность σ_2	1	1	1	0,8	0,7	0,7
	Невязка r_2	0	0	0	0	0	0
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{36}
	Уверенность σ_3	1	1	1	0,8	0,7	0,7
	Невязка r_3	0	0	0	0	0	0

Таблица 5

Сценарий 2

Характеристики		Начальное значение	Вход						
			u_1	u_2	u_4	u_2	u_4	u_5	u_6
Система	Состояние x	x_1	x_2	x_3	x_2	x_3	x_4	x_5	x_6
	Выход y	y_1	y_2	y_3	y_2	y_3	y_4	y_2	y_5
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ^*_{12}	μ^*_{14}	μ^*_{15}	μ^*_{16}
	Уверенность σ_1	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_1	0	0	0	0	0	0	0	0
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ^*_{22}	μ^*_{24}	μ^*_{25}	μ^*_{26}
	Уверенность σ_2	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_2	0	0	0	0	0	0	0	0
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ^*_{32}	μ^*_{34}	μ^*_{35}	μ^*_{36}
	Уверенность σ_3	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_3	0	0	0	0	0	0	0	0

Таблица 6

Сценарий 3

Характеристики		Начальное значение	f_1	f_2				f_3	
			Вход						
			u_1	u_1	u_2	u_4	u_5	u_1	u_2
Система	Состояние x	x_1	x_3	x_2	x_3	x_4	x_6	x_2	x_4
	Выход y	y_1	y_3	y_2	y_3	y_4	y_5	y_2	y_4
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{11}	μ_{12}
	Уверенность σ_1	1	1	1	1	0,8	0	1	0
	Невязка r_1	0	0	0	0	0	1	0	1
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{21}	μ_{22}
	Уверенность σ_2	1	0	1	1	0,8	0,8	1	0
	Невязка r_2	0	1	0	0	0	0	0	1
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{31}	μ_{32}
	Уверенность σ_3	1	0	1	1	0,8	0	1	1
	Невязка r_3	0	1	0	0	0	1	0	0

Отметим, что реализация различных сценариев требует выполнения каждого из этапов регламента разное число раз, что нашло своё отражение в таблицах. Состояния наблюдателей каналов диагностирования, фигурирующие в таблицах, имеют следующие значения: $\mu_{10} = \mu_{20} = \mu_{30} = (1\ 0\ 0\ 0\ 0\ 0)$, $\mu_{11} = (0\ 1\ 1\ 0\ 0\ 0)$, $\mu_{21} = \mu_{31} = (0\ 1\ 0\ 0\ 0\ 0)$, $\mu_{12} = \mu_{22} = (0\ 0\ 1\ 0\ 0\ 0)$, $\mu_{32} = (0\ 0\ 1\ 1\ 0\ 0)$, $\mu_{14} = \mu_{24} = \mu_{34} = (0{,}1\ 0{,}1\ 0\ 0{,}8\ 0\ 0)$, $\mu_{15} = \mu_{35} = (0{,}2\ 0{,}1\ 0\ 0\ 0{,}7\ 0)$, $\mu_{25} = (0{,}2\ 0{,}1\ 0\ 0\ 0{,}7\ 0{,}8)$, $\mu_{16} = \mu_{26} = \mu_{36} = (0\ 0\ 0\ 0\ 0\ 0{,}7)$. Состояния наблюдателей каналов, фигурирующие в табл. 5, имеют следующие значения: $\mu^*_{12} = \mu^*_{22} = (0\ 0\ 0{,}1\ 0\ 0\ 0)$, $\mu^*_{32} = (0\ 0\ 0{,}1\ 0{,}1\ 0\ 0)$, $\mu^*_{14} = \mu^*_{24} = \mu^*_{34} = (0{,}1\ 0{,}1\ 0\ 0{,}1\ 0\ 0)$, $\mu^*_{15} = \mu^*_{35} = (0{,}1\ 0{,}1\ 0\ 0\ 0{,}1\ 0)$, $\mu^*_{25} = (0{,}1\ 0{,}1\ 0\ 0\ 0{,}1\ 0{,}1)$, $\mu^*_{16} = \mu^*_{26} = \mu^*_{36} = (0\ 0\ 0\ 0\ 0\ 0{,}1)$. Нижние индексы при состояниях имеют следующий смысл: первый индекс соответствует номеру канала, второй индекс соответ-

ствует номеру входа, под воздействием которого осуществляется переход в это состояние.

Верхний индекс * вводится для удобства обозначения вновь появляющихся состояний при реализации сценария 2, генерируемых при тех же входах, что и в случае реализации сценариев 1 и 3; это связано с тем, что реализация сценария 2 требует выполнения большего числа этапов регламента.

Из табл. 4 и 5 следует, что как при полном соблюдении регламента, так и при допустимом отклонении от регламента, обусловливаемым действием менеджера процесса, на выходах каналов формируются нулевые значения невязок. В тоже время табл. 6 показывает, что при возникновении ошибок в действиях инициатора, исполнителя и менеджера процесса формируется структурированный вектор невязок, позволяющий сделать однозначный вывод о виде ошибки в момент ее проявления.

ЗАКЛЮЧЕНИЕ

В статье предложен метод диагностирования ДСС, описываемых моделью нечеткого конечного автомата. Использование подобной модели (по сравнению с недетерминированной моделью) может положительно сказаться на возможности достижения требуемой глубины диагностирования. Действительно, пусть сформированный вектор невязок не позволяет сделать однозначное заключение о виде дефекта (вектор содержит несколько нулевых компонент). В этом случае для локализации дефекта может потребоваться проведение ряда дополнительных проверок. С целью сокращения числа таких проверок их следует осуществлять в порядке, соответствующем уменьшению степени уверенности σ_s, $1 \le s \le N$.

Очевидным образом метод может быть распространен на случаи использования более простых моделей в виде детерминированных и недетерминированных конечных автоматов. Отличительной особенностью метода является то, что для его реализации не требуется предварительное нахождение табличного описания средств диагностирования; все вычисления осуществляются непосредственно в процессе диагностирования с использованием компактных аналитических соотношений. Тем самым преодолевается проблема «проклятия размерности», характерная для методов, ориентированных на табличное (или в виде графов) описание конечноавтоматной модели ДСС. Последнее позволяет практически снять ранее существовавшее ограничение на допустимую размерность модели диагностируемой ДСС. Дальнейшее развитие предложенного метода предполагается осуществить на случай задачи анализа диагностируемости и верификации моделей ДСС с большим числом состояний (см. работу [13]).

ЛИТЕРАТУРА

1. *Wonham*, *W.M.*, *Kai C.*, *Rudle*, *K.* Supervisory Control of Discrete-Event Systems: A Brief History 1980-2015 // Proc. of 20th IFAC Congress. – Toulouse, France, 2017. – P. 1827–1833.
2. *Sampath*, *M.*, *Sengupta*, *R.*, *Lafortune.*, *S.*, et al. Failure Diagnosis Using Discrete-Event Models // IEEE Transactions on Control Systems Technology. – 1996. – Vol. 4, no. 2. – P. 105–124.
3. *Zaytoon*, *J.*, *Lafortune*, *S.* Overview of Fault Diagnosis Methods for Discrete Event Systems // Annual Reviews in Control. – 2013. – Vol. 37, no. 2. – P. 308–320.
4. *Cassandras*, *Ch.*, *Lafortune*, *S.* Introduction to Discrete Event Systems. Second edition. – New York: Springer Science+Business Media, LLC, 2008. – 770 p.
5. *Жирабок А. Н.*, *Калинина Н. А.*, *Шумский А. Е.* Метод мониторинга поведения человека-оператора в человеко-машинных системах // Известия РАН. Теория и системы управления. – 2018. – № 3. – С. 98–107. [*Zhirabok*, *A.N.*, *Kalinina*, *N.A.*, *Shumskii*, *A.E.* Technique of Monitoring a Human Operator's Behavior in Man-Machine Systems // Journal of Computer and Systems Sciences International. – 2018. – Vol. 57, no. 3. – P. 443–452.]
6. *Жирабок А. Н.*, *Калинина Н. А.*, *Шумский А. Е.* Метод функционального диагностирования недетерминированных конечных автоматов // Известия РАН. Теория и системы управления. – 2020. – № 4. – С. 62–72. [*Zhirabok*, *A.N.*, *Kalinina*, *N.A.*, *Shumskii*, *A.E.* Method for the Functional Diagnosis of Nondeterministic Finite State Machines //Journal of Computer and Systems Sciences International. – 2020. – Vol. 59, no. 4. – P. 565–574.]
7. *Hartmanis*, *J.*, *Stearns*, *R.* The Algebraic Structure Theory of Sequential Machines. – New York: Prentice-Hall Inc., 1966. – 211 p.
8. *Заде Л.* Понятие лингвистической переменной и ее применение к принятию приближенных решений. – М: Мир, 1976. – 165 с. [*Zadeh, L.A., Fu, K.S., Tanaka, K., and Shimura, M.* Fuzzy Sets and Their Applications to Cognitive and Decision Processes. – New York: Academic Press, 1975.]
9. *Wee*, *W.G.*, *Fu*, *K.S.* A Formulation of Fuzzy Automata and Its Applications as a Model of Learning Systems // IEEE Trans. Syst. Science and Cybernetics. – 1969. – Vol. 5, no. 3. – P. 215–223.
10. *Ульзутуев И.Е.*, *Максимов А.А.* О свойствах решёток подавтоматов нечётких полуавтоматов и их детерминизаторов // Вестник СГТУ. – 2015. – № 2 (79). – С. 117–126. [*Ulzutuev*, *I.E.*, *Maximov*, *A.A.* The Properties of Subautomata Lattices of Fuzzy Semiautomata and Their Determinizators // *Vestnik of Saratov State Technical University*. – 2015. – No. 2 (79). – P. 117–126. (In Russian)]
11. *Воеводин В.В.*, *Кузнецов Ю.А.* Матрицы и вычисления. – М.: Наука, 1984. – 320 с. [*Voevodin*, *V.V.*, *Kuznetsov*, *U.A.* Matrices and computations. – Moscow: Nauka, 1984. – 320 p. (In Russian)]
12. *Грузликов А.М., Колесов Н.В.* Дискретно-событийная диагностическая модель распределенной вычислительной си-

стемы. Независимые цепи // Автоматика и телемеханика. – 2016. – № 10. – С. 139–154. [*Gruzlikov, A.M., Kolesov, N.V.* Discrete-Event Diagnostic Model for a Distributed Computational System. Independent Chains // Automation and Remote Control. – 2017. – Vol. 77, no. 10. – P. 1805–1817.]

13. *Tuxi, T.M.*, *Carvalho, L.K.*, *Nunes, E.V.L.*, *Cunha, A.E.* Diagnosability Verification Using LTL Model Checking // Discrete Event Dynamic Systems. – 2022 – Vol. 32, no. 3. – P. 399–433.

Статья представлена к публикации членом редколлегии С.А. Красновой.

Поступила в редакцию 04.03.2025,
после доработки 17.04.2025.
Принята к публикации 29.04.2025.

Шумский Алексей Евгеньевич – д-р техн. наук,
ORCID iD: https://orcid.org/0000-0002-5429-7482
✉ a.e.shumsky@yandex.com

Жирабок Алексей Нилович – д-р техн. наук,
ORCID iD: https://orcid.org/0000-0001-5927-7117
✉ zhirabok@mail.ru

Дальневосточный федеральный университет, г. Владивосток

A FAULT DIAGNOSIS METHOD FOR DISCRETE-EVENT SYSTEMS BASED ON THE FUZZY FINITE STATE AUTOMATON MODEL

A. E. Shumsky* and A. N. Zhirabok**

*,**Far Eastern Federal University, Vladivostok, Russia

*✉ a.e.shumsky@yandex.com, **✉ zhirabok@mail.ru

Abstract. This paper considers the problem of fault diagnosis in critical-purpose discrete-event systems described by the fuzzy finite state automaton (FSA) model. A solution method involving the mathematical apparatus of fuzzy logic is proposed. Fuzzy logic operations are described, and the concept of the determinizer of a fuzzy FSA is introduced. A diagnosis scheme that forms a structured residual vector is given. This scheme contains several channels (according to the number of possible faults in the system). Each channel is based on an observer, i.e., a determinizer of a special fuzzy FSA that simultaneously considers the possibility of both correct and incorrect transitions of the automaton (the normal operation of the system and the occurrence of a system fault, respectively). Another part of the channel is the decision block. Some ways to design the observer and the decision block are proposed. The features of the solution method are illustrated on the example of error monitoring for human operators in IT systems.

Keywords: discrete-event systems, fuzzy logic, fuzzy finite state automata, determinizer, fault diagnosis, IT systems, monitoring.

Acknowledgments. This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. FZNS-2023-0011.

УДК 330.356.7:519.21

ПОСТРОЕНИЕ ПРОИЗВОДСТВЕННОЙ CES-ФУНКЦИИ НА ОСНОВЕ ДИСКРЕТНОГО РАСПРЕДЕЛЕНИЯ ВЕЙБУЛЛА

В. В. Коков*, В. В. Соколянский**

*,**Московский государственный технический университет им. Н.Э. Баумана, г. Москва

*✉ kokovvsevo@gmail.com, **✉ sokolyansky63@mail.ru

Аннотация. Рассмотрен один из вероятностных подходов к получению производственной CES-функции. Подход заключается в нахождении математического ожидания и медианы функции Леонтьева (количества выпускаемой продукции) как случайной величины, зависимой от мощностей факторов производства – отношений факторов к их удельным значениям. Обоснован вид функции распределения минимума из набора независимых случайных величин. Показано выражение математического ожидания и медианы количества выпускаемой продукции в виде CES-функций в случае, когда мощности факторов производства имеют непрерывные распределения Вейбулла. Предложено рассмотреть дискретно распределённые факторы производства на примере геометрического закона. Выполнена попытка получения CES-функции в случае, когда мощности факторов имеют дискретные распределения Вейбулла. Описаны трудности, возникающие при аналитическом использовании математического ожидания функции Леонтьева.

Ключевые слова: производственная функция, производственная CES-функция, вероятностный подход, распределение Вейбулла, дискретное распределение Вейбулла, геометрическое распределение, математическое ожидание, медиана.

ВВЕДЕНИЕ

Традиционно производственные функции, устанавливающие связь между факторами производства $X_1,...,X_n$ и количеством Q выпускаемой предприятием (или страной) продукции, описывают в терминах, связанных с предельной нормой замещения S_{ij} и эластичностью замещения σ_{ij} фактора X_i фактором X_j; при этом предполагают, что факторы принимают детерминированные значения [1]. В частности, для двух факторов X_1 и X_2 свойством $\sigma_{12}=\text{const}$ обладает функция вида CES.

Однако начиная со второй половины XX в. выделился вероятностный подход к описанию производственных функций, получивший особое развитие в 1990–2015 гг. [2–5]. Наиболее значимым достижением здесь явилась разработка теоретического аппарата на основе следующих понятий: технология (идея), локальная производственная функция, технологическое меню, глобальная (агрегативная) производственная функция [2]. Коротко поясним суть на следующем примере [3].

Пусть в производстве задействованы два фактора X_1 и X_2, и пусть x_1 и x_2 — некоторые технологические параметры, соответствующие данным факторам. Рассмотрим производственную функцию вида CES

$$Y = A\left(\psi\left(\frac{X_1}{x_1}\right)^{\theta} + (1-\psi)\left(\frac{X_2}{x_2}\right)^{\theta}\right)^{\frac{1}{\theta}},$$

где $A>0$, $\psi\in(0;1)$, $\theta\in(-\infty;0)\cup(0;1)$ – константы.

Пару значений параметров (x_1,x_2) называют *технологией*, или *идеей*. Функцию Y с фиксированными значениями x_1 и x_2 называют *локальной производственной функцией, соответствующей технологии* (x_1,x_2).

Пусть на значения параметров x_1 и x_2 наложена связь – *технологическое меню*. Рассмотрим связь вида

$$T_1(x_1)T_2(x_2) = N,$$

где $T_1(x_1)$ и $T_2(x_2)$ – некоторые (неизвестные) функции одной переменной, N – константа.

В рамках указанного примера возникает задача: при выбранных значениях факторов X_1 и X_2 найти такой вид функций $T_1(x_1)$ и $T_2(x_2)$, что при наложенной связи функция Y будет достигать наибольших значений (при этом Y будет называться *глобальной производственной функцией*):

$$\begin{cases} Y = A\left(\psi\left(\frac{X_1}{x_1}\right)^{\theta} + (1-\psi)\left(\frac{X_2}{x_2}\right)^{\theta}\right)^{\frac{1}{\theta}} \to \max; \\ T_1(x_1)T_2(x_2) = N. \end{cases}$$

Воспользовавшись методом множителей Лагранжа и методом разделения переменных, можно показать, что функции $F_1(x_1) = 1 - T_1(x_1)$ и $F_2(x_2) = 1 - T_2(x_2)$ будут представлять собой функции распределения Вейбулла. Специалистами исследована и обратная задача [4]: по распределённым по закону Вейбулла параметрам x_1 и x_2 восстановить глобальную производственную функцию как функцию вида CES.

Иной подход позднее был предложен А. В. Михеевым [6]. Параметром x_i было предложено обозначать *удельное значение фактора* X_i – количество этого фактора, необходимое для изготовления одной единицы продукции. Было введено понятие *мощности* Q_i *фактора* X_i как отношения количества фактора X_i к его удельному значению x_i:

$$Q_i = \frac{X_i}{x_i}.$$

Предложено рассматривать мощности факторов как случайные величины. Рассмотрена двухфакторная производственная функция Леонтьева $Q = \min\{Q_1, Q_2\}$ и показано, что математическое ожидание этой функции можно найти через двойной интеграл [6]

$$EQ = \int_0^{+\infty} q_1 \left(\int_{q_1}^{+\infty} (p_{Q_1Q_2}(q_1, q_2) + p_{Q_1Q_2}(q_2, q_1)) dq_2 \right) dq_1, \quad (1)$$

где $p_{Q_1Q_2}(q_1, q_2)$ – совместная плотность распределения случайных величин Q_1 и Q_2. С помощью формулы (1) было показано, что если Q_1 и Q_2 независимы и имеют распределения Вейбулла с одинаковым коэффициентом формы $\beta > 0$, то EQ выражается через математические ожидания EQ_1 и EQ_2 величин Q_1 и Q_2 в виде CES-функции:

$$EQ = \left((EQ_1)^{-\beta} + (EQ_2)^{-\beta}\right)^{-\frac{1}{\beta}}.$$

При использовании формулы (1) имеют место достаточно объёмные выкладки. Однако возможен иной путь: найти закон (функцию или плотность) распределения случайной величины $Q = \min\{Q_1, Q_2\}$ и выразить математическое ожидание EQ по определению. Если бы случайные величины Q_1 и Q_2 были независимы и подчинялись одинаковому закону распределения, то эта задача превратилась бы в известную из математической статистики задачу о законе распределения минимальной реализации из случайной выборки с заданной генеральной совокупностью [7]. Достоинство указанной задачи в том, что она позволяет работать со случайной выборкой любого объёма n, т. е. возможно рассмотрение n мощностей $Q_1, \ldots, Q_n$. Именно некоторая модификация этой задачи и будет представлять интерес для дальнейших изысканий.

Следует отметить, что в работах [2–6] рассматривались лишь непрерывные модели, в то время как факторы производства или их мощности в реальности могут представлять собой дискретные величины. Здесь интерес представляет рассмотрение мощностей факторов производства как дискретно распределённых случайных величин и попытка построения на их основе производственных функций. Особенно важно рассмотреть дискретный аналог распределения Вейбулла и аналитическими методами попытаться воссоздать на его основе CES-функцию.

Исходя из вышесказанного, выделим ряд задач:

- Предложить эффективный метод нахождения закона распределения функции Леонтьева от мощностей факторов производства как независимых случайных величин.
- Показать возможность получения CES-функции от математических ожиданий и медиан мощностей n факторов производства в случае, когда мощности как независимые случайные величины имеют непрерывные распределения Вейбулла с одинаковым коэффициентом формы.
- Предложить рассмотрение дискретно распределённых мощностей факторов производства на примере геометрического закона.
- Аналитическими методами провести попытку построения CES-функции в случае, когда мощности факторов как независимые случайные величи-

ны имеют дискретные распределения Вейбулла с одинаковым коэффициентом формы.

1. ОБЩЕЕ УТВЕРЖДЕНИЕ

Пусть в производстве задействовано n невзаимозаменяемых факторов, мощности которых обозначим $Q_1,\ldots,Q_n$. Для невзаимозаменяемых факторов можем применить принцип производства Леонтьева, согласно которому количество выпускаемой продукции равно наименьшей из мощностей используемых факторов производства. Кроме того, будем рассматривать мощности $Q_1,\ldots,Q_n$ как независимые случайные величины.

Будем основываться на следующем утверждении.

Утверждение 1. *Пусть $Q_1,\ldots,Q_n$ – независимые случайные величины, каждая из которых имеет функцию распределения вида*

$$F_{Q_i}(q)=\begin{cases} f_i(q), & q\ge b_i;\\ 0, & q<b_i, \end{cases}$$

где b_i – некоторые числа. Тогда функция распределения случайной величины

$$Q=\min\{Q_1,\ldots,Q_n\}$$

имеет вид

$$F_Q(q)=1-\prod_{i=1}^{n}(1-F_{Q_i}(q)). \qquad (2)$$

Доказательство.

$$F_Q(q)=P\{Q<q\}=1-P\{Q\ge q\}=$$

$$=1-P\{\min\{Q_1,...,Q_n\}\ge q\}=1-P\{Q_1\ge q,...,Q_n\ge q\}.$$

Но $P\{Q_1\ge q,...,Q_n\ge q\}=P\{Q_1\ge q\}\cdot...\cdot P\{Q_n\ge q\}$, так как $Q_1,...,Q_n$ независимы. Таким образом,

$$F_Q(q)=1-P\{Q_1\ge q\}\cdot...\cdot P\{Q_n\ge q\},$$

откуда получаем формулу (2). ♦

Замечание. Далее в работе встретятся дискретно распределённые случайные величины $Q_1,\ldots,Q_n$ с целочисленными областями возможных значений 1, 2, 3, … или 0, 1, 2, …; для них утверждение 1 остаётся в силе (для простоты можно представить, что b_i – целые числа и что из множества $q\ge b_i$ отбираются целые числа).

2. СЛУЧАЙ НЕПРЕРЫВНЫХ ВЕЛИЧИН

Представим Q_1 и Q_2 как мощности некоторых факторов производства. Предположим, что это две независимые непрерывные случайные величины, распределённые по законам Вейбулла [8] с одинаковым коэффициентом формы $\beta>0$ и коэффициентами $\alpha_1>0$ и $\alpha_2>0$:

$$F_{Q_1}(q)=\begin{cases} 1-e^{-\alpha_1 q^{\beta}}, & q\ge 0;\\ 0, & q<0, \end{cases}$$

$$F_{Q_2}(q)=\begin{cases} 1-e^{-\alpha_2 q^{\beta}}, & q\ge 0;\\ 0, & q<0. \end{cases}$$

Пусть Q – количество выпускаемой продукции – зависит от мощностей Q_1 и Q_2 факторов согласно принципу производства Леонтьева:

$$Q=\min\{Q_1,Q_2\}.$$

Применяя утверждение 1, получаем

$$F_Q(q)=\begin{cases} 1-e^{-\alpha q^{\beta}}, & q\ge 0;\\ 0, & q<0, \end{cases}$$

где $\alpha=\alpha_1+\alpha_2$. Так, случайная величина Q имеет распределение Вейбулла. Запишем её математическое ожидание [8]:

$$EQ=\alpha^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right)=\left(\alpha_1+\alpha_2\right)^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right), \qquad (3)$$

где Γ – гамма-функция.

Из формул для математических ожиданий EQ_1 и EQ_2 случайных величин Q_1 и Q_2

$$EQ_1=\alpha_1^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right),$$

$$EQ_2=\alpha_2^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right)$$

выражаем коэффициенты α_1 и α_2 и, подставляя их в формулу (3), получаем:

$$EQ=\left((EQ_1)^{-\beta}+(EQ_2)^{-\beta}\right)^{-\frac{1}{\beta}}, \qquad (4)$$

что определяет производственную CES-функцию.

В дальнейшем интерес также представляет и частный случай при $\beta=1$. При этом распределение Вейбулла (для Q_1,Q_2,Q) превращается в экспоненциальное, а выражение (4) примет вид

$$EQ=\frac{EQ_1EQ_2}{EQ_1+EQ_2}. \qquad (5)$$

Проведённые выкладки можно обобщить на случай n факторов. Кроме того, вместо математи-

ческих ожиданий можно рассмотреть и другие характеристики случайных величин, например, медианы.

Утверждение 2. *Пусть* Q_i $(i=1,...,n)$ *– мощности факторов производства, представимые как независимые непрерывные случайные величины, распределённые по законам Вейбулла с одинаковым коэффициентом формы* $\beta>0$ *и коэффициентами* $\alpha_1>0,\ldots,\alpha_n>0$:

$$F_{Q_i}(q)=\begin{cases}1-e^{-\alpha_i q^{\beta}}, & q\ge 0;\\ 0, & q<0,\end{cases}$$

и пусть количество выпускаемой продукции Q *определяется принципом Леонтьева:*

$$Q=\min\{Q_1,...,Q_n\}.$$

Тогда математическое ожидание EQ *и медиана* M *величины* Q *выражаются через соответственно математические ожидания* EQ_i *и медианы* M_i *величин* Q_i *как производственные CES-функции:*

$$EQ=\left((EQ_1)^{-\beta}+...+(EQ_n)^{-\beta}\right)^{-\frac{1}{\beta}}, \qquad (6)$$

$$M=\left(M_1^{-\beta}+...+M_n^{-\beta}\right)^{-\frac{1}{\beta}}. \qquad (7)$$

Доказательство. Применяя утверждение 1, получаем функцию распределения Вейбулла

$$F_Q(q)=\begin{cases}1-e^{-\alpha q^{\beta}}, & q\ge 0;\\ 0, & q<0,\end{cases}$$

где $\alpha=\alpha_1+\ldots+\alpha_n$. Запишем выражения для математического ожидания EQ и медианы M случайной величины Q:

$$EQ=\alpha^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right)=\left(\alpha_1+...+\alpha_n\right)^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right),$$

$$M=\alpha^{-\frac{1}{\beta}}(\ln 2)^{\frac{1}{\beta}}=\left(\alpha_1+...+\alpha_n\right)^{-\frac{1}{\beta}}(\ln 2)^{\frac{1}{\beta}}.$$

Записывая формулы для математических ожиданий EQ_i и медиан M_i случайных величин Q_i

$$EQ_i=\alpha_i^{-\frac{1}{\beta}}\Gamma\left(\frac{1}{\beta}+1\right),$$

$$M_i=\alpha_i^{-\frac{1}{\beta}}(\ln 2)^{\frac{1}{\beta}},$$

выражая из них коэффициенты α_i и подставляя их в выражения для EQ и M, получаем формулы (6) и (7). ♦

3. СЛУЧАЙ ДИСКРЕТНЫХ ВЕЛИЧИН

Попробуем теперь представить мощности факторов производства как случайные величины, имеющие дискретные распределения.

3.1. Пример: геометрическое распределение

Пусть Q_1 и Q_2 – мощности факторов производства, представимые как две независимые случайные величины, имеющие геометрические распределения с параметрами (вероятностями успеха однократного испытания) p_1 и p_2 соответственно. Здесь распределение будем понимать в следующем смысле: случайная величина – номер испытания, давшего первый успех (возможные значения $j=1,2,...$).

Тогда вероятности неудач однократных испытаний равны $q_1=1-p_1$ и $q_2=1-p_2$ соответственно.

Функции распределения случайных величин Q_1 и Q_2 имеют вид

$$F_{Q_1}(j)=P\{Q_1<j\}=1-q_1^{j-1},$$

$$F_{Q_2}(j)=P\{Q_2<j\}=1-q_2^{j-1}.$$

Пусть количество выпускаемой продукции Q определяется принципом Леонтьева:

$$Q=\min\{Q_1,Q_2\}.$$

Применяя утверждение 1, получаем

$$F_Q(j)=1-q_0^{j-1},$$

где $q_0=q_1q_2$. Видим, что случайная величина Q имеет геометрическое распределение с параметром $p_0=1-q_0$. Выразим p_0 через параметры p_1 и p_2:

$$p_0=1-q_1q_2=1-(1-p_1)(1-p_2)=p_1+p_2-p_1p_2.$$

Известно, что

$$EQ_i=\frac{1}{p_i},\ i=1,2, \qquad (8)$$

$$EQ=\frac{1}{p_0}=\frac{1}{p_1+p_2-p_1p_2}. \qquad (9)$$

Выражая p_1 и p_2 из формулы (8) и подставляя их в формулу (9), получаем

$$EQ=\frac{EQ_1EQ_2}{EQ_1+EQ_2-1}. \qquad (10)$$

Укажем, что если для геометрического распределения использовать другое определение (случайная величина – число неудач до первого успеха), то

$$EQ=\frac{EQ_1EQ_2}{EQ_1+EQ_2+1}. \qquad (11)$$

Приведём достаточно примитивный пример ситуации, когда мощности факторов производства Q_1 и Q_2 представимы геометрически распределёнными случайными величинами. Предположим, что для попытки выпуска одного неделимого изделия требуется затратить либо единичную мощность фактора 1, либо единичную мощность фактора 2. Одна попытка может закончиться успехом (изделие прошло контроль и испытания) или неудачей (не прошло). При этом если используется фактор 1, то вероятность успеха обозначим p_1. Если же используется фактор 2, то вероятность успеха обозначим p_2. Пусть номер попытки не влияет на вероятность её успеха.

Номер первой успешной попытки выпуска изделия при использовании выбранного фактора есть реализация мощности выбранного фактора как случайной величины, и эта величина по определению имеет геометрическое распределение. Очевидно, что выгодно задействовать именно тот из факторов, при котором номер первого успеха будет минимален (отсылка к принципу Леонтьева).

Интерес представляет некоторая схожесть формул (10), (11) с формулой (5), полученной для экспоненциально распределённых факторов.

Известно, что геометрическое распределение является дискретным аналогом экспоненциального. Пусть некоторая случайная величина Q имеет экспоненциальное распределение с плотностью

$$p_Q(q)=\begin{cases}\lambda e^{-\lambda q}, & q\ge 0;\\ 0, & q<0,\end{cases}\qquad \lambda>0.$$

Рассмотрим случайную величину $Y=\lceil Q\rceil$ (потолок величины Q). Для натуральных $j=1,2,...$

$$P\{Y=j\}=P\{j-1<Q\le j\}=\int_{j-1}^{j}p_Q(q)dq,$$

откуда

$$P\{Y=j\}=(1-e^{-\lambda})e^{-\lambda(j-1)}.$$

Обозначим $q_0=e^{-\lambda}=1-p_0$; тогда

$$P\{Y=j\}=(1-q_0)q_0^{j-1}=p_0(1-p_0)^{j-1}.$$

Таким образом, величина Y имеет геометрическое распределение с параметром $p_0=1-e^{-\lambda}$ (в смысле «номер испытания, давшего первый успех»).

Если же рассмотреть целую часть $\lfloor Q\rfloor$, то она также подчинена геометрическому закону с параметром $p_0=1-e^{-\lambda}$, но в смысле «число неудач до первого успеха».

3.2. Дискретизация распределения Вейбулла и попытка построения CES-функции

Проведём аналогичным образом «дискретизацию» распределения Вейбулла. В данном случае плотность распределения случайной величины Q имеет вид

$$p_Q(q)=\begin{cases}\alpha\beta q^{\beta-1}e^{-\alpha q^{\beta}}, & q\ge 0;\\ 0, & q<0,\end{cases}\qquad \beta>0,\ \alpha>0.$$

Рассмотрим случайную величину $Y=\lfloor Q\rfloor$. Для $j=0,1,2,...$ справедливо

$$P\{Y=j\}=P\{j\le Q<j+1\}=\int_{j}^{j+1}p_Q(q)dq,$$

после преобразований получаем

$$P\{Y=j\}=e^{-\alpha j^{\beta}}-e^{-\alpha(j+1)^{\beta}}. \qquad (12)$$

Тогда функция распределения случайной величины Y будет иметь вид

$$F_Y(j)=P\{Y<j\}=\sum_{k=0}^{j-1}P\{Y=k\}=1-e^{-\alpha j^{\beta}} \qquad (13)$$

– искомое дискретное распределение Вейбулла (типа 1 [9, 10]).

Найдём медиану M случайной величины Y. Для этого введём в рассмотрение квантиль Q_γ уровня γ; его неокруглённое значение является решением уравнения

$$F_Y(Q_\gamma)=\gamma,$$

т. е., с учётом формулы (13), решением уравнения

$$1-e^{-\alpha Q_\gamma^{\beta}}=\gamma.$$

После преобразований получаем

$$Q_\gamma=\left(-\frac{\ln(1-\gamma)}{\alpha}\right)^{\frac{1}{\beta}},$$

откуда неокруглённое значение медианы равно

$$M=Q_{1/2}=\left(\frac{\alpha}{\ln 2}\right)^{-\frac{1}{\beta}}. \qquad (14)$$

Рассмотрим теперь математическое ожидание случайной величины Y (с учётом формулы (12)):

$$EY=\sum_{j=0}^{\infty}jP\{Y=j\}=\sum_{j=0}^{\infty}j\left(e^{-\alpha j^{\beta}}-e^{-\alpha(j+1)^{\beta}}\right). \qquad (15)$$

Суммирование этого ряда проводят численно [9]; в данном же случае важна попытка его аналитического выражения.

В предположении о сходимости ряда (15) можно, раскрыв скобки в разложении и приведя соседние подобные члены, перейти к выражению

$$EY = \sum_{j=1}^{\infty} (e^{-\alpha})^{j^{\beta}}. \qquad (16)$$

Здесь рассмотрим случай $\beta > 1$. Каждый член ряда (16) при этом меньше соответствующего члена сходящегося ряда геометрической прогрессии $\sum_{j=1}^{\infty} e^{-\alpha j}$, а значит ряд (16) будет сходиться.

Теоретический интерес представляет случай $\beta = 2$ (дискретный вариант распределения Рэлея); далее будем рассматривать только его. Ряд (16) принимает следующий вид (обозначим его $u(\alpha)$):

$$EY = u(\alpha) = \sum_{j=1}^{\infty} e^{-\alpha j^2} = \\ = (e^{-\alpha})^{1^2} + (e^{-\alpha})^{2^2} + (e^{-\alpha})^{3^2} + \ldots \qquad (17)$$

Для рассмотрения ряда (17) целесообразно ввести тета-функцию

$$\theta(s) = \sum_{j=-\infty}^{\infty} (e^{-\pi s})^{j^2}$$

и функцию вида [11]

$$w(s) = \sum_{j=1}^{\infty} (e^{-\pi s})^{j^2} = \frac{1}{2}(\theta(s) - 1).$$

Можно показать [11], что справедливо функциональное уравнение

$$w(s) = \frac{1}{\sqrt{s}} w\left(\frac{1}{s}\right) + \frac{1}{2}\left(\frac{1}{\sqrt{s}} - 1\right).$$

Пусть $\pi s = \alpha$. Функции $u(\alpha)$ и $w(s)$ свяжутся так:

$$u(\alpha) = w(s) = w\left(\frac{\alpha}{\pi}\right),$$

и можно записать

$$u(\alpha) = \sqrt{\frac{\pi}{\alpha}} \cdot u\left(\frac{\pi^2}{\alpha}\right) + \frac{1}{2}\left(\sqrt{\frac{\pi}{\alpha}} - 1\right). \qquad (18)$$

При значениях коэффициента масштаба $0 < \alpha \lesssim 2$ первым слагаемым в правой части выражения (18) можно пренебречь, и в таком случае можно записать приближённое равенство

$$EY \approx \frac{1}{2}\left(\sqrt{\frac{\pi}{\alpha}} - 1\right), \quad 0 < \alpha \lesssim 2,$$

которое представим в виде

$$EY + \frac{1}{2} \approx \frac{\sqrt{\pi}}{2} \alpha^{-\frac{1}{2}}, \quad 0 < \alpha \lesssim 2. \qquad (19)$$

При $\alpha \gtrsim 2$ в разложении (17) можно ограничиться первым членом; можно записать приближённое равенство

$$EY \approx e^{-\alpha}, \quad \alpha \gtrsim 2.$$

На основании утверждения 1 и проведённых для дискретного распределения Вейбулла выкладок можно сформулировать следующее

Утверждение 3. *Пусть* Q_i $(i = 1, \ldots, n)$ *– мощности факторов производства, представимые как независимые случайные величины, распределённые по дискретным законам Вейбулла с одинаковым коэффициентом формы* β:

$$F_{Q_i}(j) = 1 - e^{-\alpha_i j^{\beta}}, \quad j = 0, 1, 2, \ldots$$

и пусть количество выпускаемой продукции Q *определяется принципом Леонтьева:*

$$Q = \min\{Q_1, \ldots, Q_n\}.$$

Тогда неокруглённая медиана (14) M *величины* Q *связана с неокруглёнными медианами* M_i *величин* Q_i *через CES-функцию:*

$$M = \left(M_1^{-\beta} + \ldots + M_n^{-\beta}\right)^{-\frac{1}{\beta}}.$$

Кроме того, если $\beta = 2$ *и* $0 < \alpha \lesssim 2$, *где* $\alpha = \alpha_1 + \ldots + \alpha_n$, *то математическое ожидание* EQ *величины* Q *можно приближённо связать с математическими ожиданиями* EQ_i *величин* Q_i *через CES-функцию:*

$$EQ + \frac{1}{2} \approx \left(\left(EQ_1 + \frac{1}{2}\right)^{-2} + \ldots + \left(EQ_n + \frac{1}{2}\right)^{-2}\right)^{-\frac{1}{2}}.$$

Замечание к утверждению 3. При выполнении ограничения $0 < \alpha = \alpha_1 + \ldots + \alpha_n \lesssim 2$ сразу же будут выполняться неравенства $0 < \alpha_i \lesssim 2$ для $i = 1, \ldots, n$, а значит математические ожидания EQ_i могут быть выражены в нужной для доказательства приближённой форме (19).

ЗАКЛЮЧЕНИЕ

Получены CES-функции от математических ожиданий и медиан мощностей *n* факторов производства в случае, когда мощности представлены как независимые случайные величины, подчинён-

ные непрерывным распределениям Вейбулла с одинаковым коэффициентом формы.

Предложено рассмотрение дискретно распределённых мощностей факторов на примере геометрического закона. При этом, согласно принципу Леонтьева, выгодно задействовать именно тот из факторов, при котором номер первого успеха изготовления изделия будет минимален.

Проведена попытка построения CES-функции в случае, когда мощности факторов как независимые случайные величины имеют дискретные распределения Вейбулла с одинаковым коэффициентом формы. С успехом удалось связать неокруглённые значения (14) медиан мощностей факторов и медиану количества выпускаемой продукции. Трудности возникли при поиске связи между математическими ожиданиями этих величин.

Главной трудностью в исследовании явилось аналитическое выражение математического ожидания случайной величины, распределённой по дискретному закону Вейбулла. В частном случае при коэффициенте формы, равном двум ($\beta = 2$), возможно введение тета-функции и составление функционального уравнения. При некоторых ограничениях на значения коэффициента масштаба распределения удаётся пренебречь частью функционального уравнения, тем самым приближённо выразить искомое математическое ожидание (см. формулу (19)) и составить CES-функцию.

Остаётся открытым вопрос о возможности связывания *в общем случае* (т.е. без ограничений на коэффициенты распределений) в CES-функцию математических ожиданий мощностей факторов, если они как случайные величины подчинены дискретным распределениям Вейбулла с одинаковым коэффициентом формы.

ЛИТЕРАТУРА

1. *Горбунов В.К.* Производственные функции: теория и построение: учебное пособие. – Ульяновск: УлГУ, 2013. – 84 с. – URL: https://ulsu.ru/media/documents/13Горбунов_ПрФунк.pdf [*Gorbunov, V.K.* Proizvodstvennye funktsii: teoriya i postroenie: uchebnoe posobie. – Ulyanovsk: ULGU, 2013. – 84 s. – URL: https://ulsu.ru/media/documents/13Горбунов_ПрФунк.pdf (In Russian)]
2. *Jones, C.I.* The Shape of Aggregate Production Functions and the Direction of Technical Change // Quarterly Journal of Economics. – 2005. – Vol. 120, no. 2. – P. 517–549.
3. *Growiec, J.* Production Functions and Distributions of Unit Factor Productivities: Uncovering the Link // Economics Letters. – 2008. – Vol. 101, no. 1. – P. 87–90.
4. *Growiec, J.* A Microfoundation for Normalized CES Production Functions with Factor-augmenting Technical Change // Journal of Economic Dynamics and Control. – 2013. – Vol. 37, no. 11. – P. 2336–2350.
5. *Матвеенко В.Д.* «Анатомия» производственной функции: технологическое меню и выбор наилучшей технологии // Экономика и математические методы. – 2009. – Т. 45, № 2. – С. 85–95. [*Matveenko, V.D.* “Anatomy” of the Production Function: Technological Menu and Selection of the Best Technology // Economics and Mathematical Methods. – 2009. – Vol. 45, no. 2. – P. 85–95. (In Russian)]
6. *Михеев А.В.* Вероятностный подход к определению производственных функций // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2021. – № 4. – С. 82–94. [*Mikheev, A.V.* Probabilistic Approach to Determining Production Functions // Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics. – 2021. – No. 4. – P. 82–94. (In Russian)]
7. *Математическая* статистика: Учебник для вузов / В.Б. Горяинов, И.В. Павлов, Г.М. Цветкова и др.; под ред. В.С. Зарубина, А.П. Крищенко. – 3-е изд., исправл. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 424 с. [*Matematicheskaya* statistika: Uchebnik dlya vuzov / V.B. Goryainov, I.V. Pavlov, G.M. Tsvetkova i dr.; pod red. V.S. Zarubina, A.P. Krishchenko. – 3-e izd., ispravl. – M.: Izd-vo MGTU im. N.Eh. Baumana, 2008. – 424 s. (In Russian)]
8. *Теория* вероятностей: Учебник для вузов / В.А. Печинкин, О.И. Тескин, Г.М. Цветкова и др.; под ред. В.С. Зарубина, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1998. – 456 с. [*Teoriya* veroyatnostei: Uchebnik dlya vuzov / V.A. Pechinkin, O.I. Teskin, G.M. Tsvetkova i dr.; pod red. V.S. Zarubina, A.P. Krishchenko. – M.: Izd-vo MGTU im. N.E. Baumana, 1998. – 456 s. (In Russian)]
9. *Rinne, H.* The Weibull Distribution: A Handbook. – New York: Chapman and Hall/CRC, 2008. – 808 p.
10. *Barbiero, A.* Discrete Weibull Distributions (Type 1 and 3) // The Comprehensive R Archive Network. – 2025. – URL: https://cran.r-project.org/web/packages/DiscreteWeibull/DiscreteWeibull.pdf.
11. *Woit, P.* Fourier Analysis Notes, Spring 2020. – New York: Department of Mathematics, Columbia University, 2020. – 79 p. – URL: https://www.math.columbia.edu/~woit/fourier-analysis/fouriernotes.pdf.

Статья представлена к публикации руководителем РРС М. И. Гераськиным.

Поступила в редакцию 21.02.2025,
после доработки 27.04.2025.
Принята к публикации 29.04.2025.

Коков Всеволод Вячеславович – студент,
✉ kokovvsevo@gmail.com,
ORCID iD: https://orcid.org/0009-0008-4331-5148

Соколянский Василий Васильевич – доцент,
✉ sokolyansky63@mail.ru,
ORCID iD: https://orcid.org/0000-0002-6636-4638

Московский государственный технический университет им. Н. Э. Баумана, г. Москва.

CONSTRUCTING THE CES PRODUCTION FUNCTION BASED ON THE DISCRETE WEIBULL DISTRIBUTION

V. V. Kokov* and V. V. Sokolyanskiy**

Bauman Moscow State Technical University, Moscow, Russia

*✉ kokovvsevo@gmail.com, **✉ sokolyansky63@mail.ru

Abstract. This paper considers a probabilistic approach to obtaining the CES production function. It consists in calculating the mean and median of the Leontief function (the quantity of output) as a random variable depending on the capacities of production factors, i.e., the ratios of the factors to their per-unit values. The type of the cumulative distribution function of the minimum from a set of independent random variables is substantiated. Explicit expressions are derived for the mean and median of the quantity of output as CES functions when the factor capacities have (continuous) Weibull distributions. Discretely distributed production factors are considered using the example of a geometric law. An attempt is made to derive the CES function when the factor capacities have discrete Weibull distributions. The difficulties arising in the analytical use of the mean of the Leontief function are described.

Keywords: production function, CES production function, probabilistic approach, Weibull distribution, discrete Weibull distribution, geometric distribution, mean, median.

УДК 004.056

МЕТОДИКА ОЦЕНКИ НЕОБХОДИМОСТИ ПРИМЕНЕНИЯ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В ПРОМЫШЛЕННЫХ СИСТЕМАХ

К. В. Семенков*, В. Г. Промыслов**

*,**Институт проблем управления им. В. А. Трапезникова РАН, г. Москва

*✉ semenkov@ipu.ru, **✉ vp@ipu.ru

Аннотация. Рассматривается проблема применения обновлений кибербезопасности (патчей) для программного обеспечения (ПО) автоматизированных систем управления технологическими процессами (АСУ ТП) с длительным жизненным циклом. Проблема исследуется в рамках этапа эксплуатации системы. Основное внимание уделяется: большому числу уязвимостей, обнаруживаемых в ПО; сложности анализа влияния уязвимости на безопасность системы и функции, выполняемые системой; требованиям к тестированию совместимости обновлений и сертификации ПО после внесения изменений. На основе метода анализа отказов (англ. *Failure Mode and Effects Analysis*, FMEA) предложена методика, позволяющая упростить анализ влияния уязвимости на кибербезопасность. Она предполагает рассмотрение не каждой отдельной уязвимости, а меньшего по мощности множества сценариев атак. При анализе сценариев атаки также учитывается действие мер защиты. Методика включает в себя простые критерии применения обновлений безопасности по результатам анализа. Приведен пример анализа уязвимости по предлагаемой методике.

Ключевые слова: уязвимость, обновление, оценка риска, АСУ ТП, кибербезопасность, критерий.

ВВЕДЕНИЕ

Одним из основных путей проведения кибератак является использование уязвимостей в программном обеспечении (ПО). Несмотря на развитие технологий разработки и тестирования ПО, сложность программ не позволяет гарантировать отсутствие уязвимостей в программном обеспечении. Злоумышленники концентрируют свои ресурсы на поиске и использовании уязвимостей, а разработчики и пользователи ПО заинтересованы в поиске, оперативном устранении этих уязвимостей, выпуске и установке соответствующих обновлений безопасности ПО (патчей). В целом под патчем часто понимается весьма широкий спектр изменений ПО [1–4], которые либо характеризуются подходом к патчу как к процессу или объекту, либо связаны с объемом или характером изменений ПО. В неформальном общении специалистов также можно встретить другие схожие термины, например, update, bugfix, hotfix. Для целей настоящей работы примем следующее

Определение. Обновление безопасности (патч) – это модификация установленного программного обеспечения, призванная устранить уязвимости программного обеспечения без изменения других функциональных характеристик программного обеспечения. ♦

В части накопления информации об уязвимостях в мире проделана огромная работа, имеются общедоступные и постоянно пополняемые базы: американская база CVE [5], российский банк данных угроз безопасности информации БДУ ФСТЭК [6], французская CERT-FR [7] и др. В базах для каждой уязвимости из списка обязательно содержатся ее описание, оценка влияния и рекомендации по устранению уязвимости или ослаблению ее негативного влияния. Производители программного обеспечения выпускают обновления безопасности, и установка обновлений зачастую может выполняться в автоматическом режиме.

Наработанный в мире опыт систематизирован в международных и национальных стандартах и методических документах по применению патчей

(см., например, [3, 8, 9]). Указания и рекомендации этих документов в целом сводятся к следующим шагам:

1) постоянно вести мониторинг уязвимостей в применяемом ПО;

2) выполнять анализ вновь открытых уязвимостей и оценку риска для кибербезопасности;

3) в зависимости от результатов оценки риска определить дальнейшие действия: принять риск, устранить риск – применить патч и т. д.;

4) в случае положительного решения по установке обновления:

а) разработать план применения обновления;

б) проверить целостность и достоверность обновления;

в) протестировать обновление;

г) установить обновление;

д) проверить состояние ПО и конфигурацию после установки обновления.

Однако вопросы практического применения имеющихся данных, прежде всего связанные с большим объемом анализируемой информации и ее достоверностью, по мнению авторов, пока до конца не решены.

В данной работе на примере автоматизированной системы управления технологическими процессами (АСУ ТП) будет рассмотрена проблема применения обновлений безопасности [100, 11] в системах с длительным жизненным циклом и предложен новый метод ее решения. Под системой с длительным жизненным циклом в работе понимается такая система, стадия эксплуатации и поддержки которой продолжается в течение нескольких лет или десятилетий.

Согласно указаниям нормативных и методических документов [10–13] идентификация, анализ и оценка уязвимостей должны проводиться в течение всего жизненного цикла защищаемой системы. Так как ресурсы для обеспечения кибербезопасности и контекст, в котором рассматривается система, существенно отличаются для различных этапов жизненного цикла, то различаются и проблемы устранения уязвимостей, и методы их решения. Данная работа посвящена вопросам оценки уязвимостей на этапе эксплуатации в предположении, что на момент окончания этапа разработки разработчик закрыл известные ему уязвимости и применил адекватные меры защиты для снижения риска до приемлемого уровня. Для этапа разработки имеется разнообразный набор рекомендаций по разработке безопасного ПО [14].

Внимание к этапу эксплуатации определяется тем, что, как показывает опыт авторов, проблема управления уязвимостями наиболее полно проявляется на этапе эксплуатации и усложняется со временем. Это связано прежде всего с интегральным эффектом нескольких факторов: накопления обнаруженных уязвимостей в используемых компонентах, окончания срока поддержки некоторых их них разработчиком, устаревания технологий обеспечения информационной защиты, заложенных в проекте системы.

В качестве объекта исследования выбраны сложные программные системы, представляющие собой комплексы программ [15], в которых используются как специально разработанные для системы компоненты, так и сторонние компоненты общего применения. Предположим также, что число компонентов в системе достаточно велико, так как для простых систем проблема патчей представляется не слишком серьезной, потому что количество уязвимостей в ней вряд ли будет большим, и они могут оперативно отслеживаться и устраняться в процессе эксплуатации. Практика показывает, что для АСУ ТП «простыми» являются системы, не выходящие за рамки одного компьютера; тогда число активов и связей между ними позволяет описать возникающие отношения безопасности дискреционной моделью, которую можно использовать при оценке риска для активов, связанных с обнаруживаемыми уязвимостями.

Проблемы управления рисками для промышленных объектов и установка обновления для цифровых систем безопасности имеют большое научное и практическое значение. Обзор проблемы оценки риска для АСУ ТП АЭС приведен в работе [16], а подробный обзор публикаций по управлению обновлениями безопасности, вышедших в 2002–2020 гг., приведен в работе [17]. Новизна настоящей работы заключается в том, что в ней детализируются методики верхнего уровня (как, например, [1, 10]), предлагается конкретный, ориентированный на промышленные системы управления набор действий для принятия решений об установке обновлений.

Проблема управления обновлениями безопасности будет рассмотрена в разрезе функций, выполняемых системой, а не в разрезе характера уязвимости отдельного компонента, для которого имеется патч. Так, основная функция АСУ ТП – это управление промышленным объектом. Тогда цель установки патча для операционной системы (ОС) компьютера в составе АСУ ТП – это не защита ОС как таковой, а снижение риска, связанного с невозможностью управлять объектом в случае использования уязвимости. Для решения предложен риск-ориентированный метод на основе метода анализа отказов и их последствий (англ. *Failure Mode and Effects Analysis*, FMEA) [18], который

дает критерии применения обновлений. В дополнение к управлению уязвимостями на основе их типов предлагается явно учитывать влияние мер защиты на реализуемость атак нарушителя, обладающего определенными возможностями. Применение описанного в работе подхода позволяет сопоставить вновь открываемые уязвимости с известными типами уязвимостей некоторого классификатора (см., например, общий перечень недостатков безопасности ПО (англ. *Common Weakness Enumeration*, CWE) [19]), и ответить на вопрос, имеется ли у системы «иммунитет» в виде барьера из совокупности мер защиты от новой уязвимости. Под барьером здесь и далее будем понимать определенный набор мер защиты, который обеспечивает гарантированную защиту от определенного сценария атак.

1. ОСОБЕННОСТИ ПРИМЕНЕНИЯ МОДЕЛИ УГРОЗ ПРИ УСТАНОВКЕ ПАТЧЕЙ

Большинство риск-ориентированных подходов, применяемых в задаче управления патчами, базируется на подходах к оценке риска, сформулированных в стандарте ИСО 27005 [20]. В соответствии с ними анализ модели угроз, включающей уязвимости, угрозы и нарушителя, в основном влияет на решение о приемлемости или неприемлемости риска и, следовательно, на решение об установке патча. Существует много методов описания элементов модели угроз и составления их таксономии; ниже будут рассмотрены те из них, которые авторы считают наиболее подходящими для оценки риска в сложных промышленных системах на этапе эксплуатации. Для этого сначала рассмотрим отдельные составляющие модели угроз.

1.1. Анализ уязвимостей

Будем придерживаться определения уязвимости из стандартов ИСО/МЭК 27000 и методических документов ФСТЭК [12]: уязвимость – это «… слабость актива (программы), которая может быть использована угрозой».

Уязвимости могут иметь различную природу. Они могут быть связаны со свойствами системы, заложенными при разработке (недостатки в архитектуре глубоко эшелонированной защиты или междоменного взаимодействия, ошибки реализации) или же могут появиться из-за неправильного применения мер защиты (например, из-за неправильного применения мер парольной защиты). Анализ уязвимостей призван установить, насколько уязвимости могут повлиять на защищенность системы и на оценку доверия к реализуемым мерам защиты [21]. Патчи должны применяться к системе в случае, если анализ применимости уязвимостей выявит недопустимый уровень риска информационной безопасности для системы (см. руководство [9], *рис. 3.1*). Попробуем показать, какие проблемы возникают при анализе уязвимостей, для чего рассмотрим использование нормативных методических рекомендаций по анализу и применению патчей более подробно.

Первая проблема, которую необходимо выделить, – это масштаб рассматриваемой системы. Как было указано выше, сложная программная система включает большое количество разнородных компонентов и сторонних приложений, и для обеспечения кибербезопасности необходимо вести мониторинг большого числа уязвимостей, связанных как с компонентами, разработанными специально для системы, так и со сторонними продуктами (например, уязвимостей операционной системы, систем управления базами данных, веб-серверов, интерпретаторов и т. п.).

Количество вновь открываемых уязвимостей с каждым годом увеличивается. Для примера приведем количество новых уязвимостей, внесенных в базы CVE и ФСТЭК в 2021–2023 гг. (табл. 1).

Таблица 1

Количество уязвимостей, добавленное в CVE и БДУ ФСТЭК, тыс.

База данных	2021 г.	2022 г.	2023 г.
БДУ ФСТЕК	6,4	7,5	9,1
CVE [22]	20,2	25,0	29,0

Для сложной системы поток уязвимостей может составить десятки и сотни уязвимостей в сутки, и даже первичный анализ новых уязвимостей может потребовать от организации существенных ресурсов и затрат.

Следующая проблема, на которую необходимо обратить внимание, состоит в том, что описание уязвимости во всех базах дается в относительно свободной форме, какого-то единого стандарта описаний не существует. Описания могут быть как весьма краткими, так и излишне детальными, что существенно усложняет их анализ. Приведем несколько примеров неудачных описаний (табл. 2). Описание уязвимости CVE-2018-19932 содержит много технических подробностей, которые могут заинтересовать лишь разработчиков данного ПО, описание уязвимости CVE-2023-36762 слишком общее, а описание уязвимости CVE-2021-30618 не содержит почти никакой информации.

Таблица 2

Примеры неудачных описаний уязвимостей

Уязвимость	Описание
CVE-2018-19932	Проблема была обнаружена в библиотеке дескрипторов двоичных файлов (BFD) (также известной как libbfd), распространявшейся в GNU Binutils до версии 2.31. Макрос IS_CONTAINED_BY_LMA в elf.c вызывает целочисленное переполнение и бесконечный цикл. (англ. An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils through 2.31. There is an integer overflow and infinite loop caused by the IS_CONTAINED_BY_LMA macro in elf.c.)
CVE-2023-36762	Уязвимость удаленного выполнения кода Microsoft Word. (англ. Microsoft Word remote code execution vulnerability.)
CVE-2021-30618	Неподходящая реализация в DevTools. (англ. Inappropriate implementation in DevTools.)

Имеются работы, направленные на автоматизацию анализа описания уязвимостей (см., например, [23, 24]), в том числе с использованием алгоритмов машинного обучения. Однако пригодные к практическому применению инструменты, полностью автоматизирующие анализ реальных систем, авторам неизвестны.

Проблема описания усугубляется наличием языкового барьера, имеющего комплексный характер. Во-первых, описания большинства уязвимостей в зарубежных базах составляются на английском языке, которым хорошо владеют далеко не все авторы описания уязвимостей. То есть уже на этапе первоначального описания уязвимости ее суть может быть искажена или неполно изложена. Во-вторых, большой объем информации об уязвимостях переносится из международных открытых баз данных в национальные, где описания обычно модерируются и переводятся на национальный язык. Таким образом, в национальных базах уязвимостей при переводе на официальные языки стран могут как добавиться ошибки и неточности, затрудняющие анализ уязвимости, так и наоборот, описания могут расширяться и уточняться. Однако в последнем случае часто теряется обратная связь с разработчиком компонента, в котором открыта уязвимость, так как сделанные уточнения доступны только на национальном языке.

Можно заключить, что для самостоятельного мониторинга и анализа такого объема уязвимостей организации необходим достаточно большой штат экспертов, которые могут квалифицированно выполнить анализ этой слабоструктурированной информации и оценить риск для защищаемой системы, ассоциированный с уязвимостью.

Возможный способ уменьшить объем анализируемой информации – это приоритизация уязвимостей, которая позволяет сконцентрировать усилия на наиболее критичных из них. Одной из самых известных и распространенных шкал оценки критичности является CVSS (*Common Vulnerability Scoring System*) [25–27]. В ней оценка ведется по трем группам: базовых, временны́х и контекстных метрик. Далее будет использована только базовая метрика, так как две последние, по опыту авторов, или не содержат информации, или информация является специфичной для конкретного сценария применения программы. К базовым метрикам CVSS 3.0 [26] относятся:

- вектор атаки, например, сетевая, локальная, физический доступ, атака на смежный сетевой протокол;
- сложность атаки;
- необходимые права доступа для эксплуатации уязвимости;
- участие «обычного» пользователя для эксплуатации уязвимости;
- возможность выхода последствий атаки за пределы исследуемой системы;
- влияние на свойства доступности, конфиденциальности, целостности информационных ресурсов, контролируемых программой, в которой обнаружена уязвимость.

Базовые метрики главным образом отражают свойства самого программного компонента, факторы среды функционирования учитываются весьма условно. Их значения, рассчитанные экспертным способом, приводятся в базах уязвимостей.

Из описания базовых метрик видно, что они главным образом предназначены для разработчиков и пользователей отдельного программного компонента и что роль компонента в информационной системе в базовых метриках не учитывается. Чтобы провести полный анализ уязвимости по системе CVSS для дальнейшей оценки риска, нужно рассчитывать оставшиеся группы метрик или использовать другие метрики и базы данных уязви-

мостей, учитывающие данные аспекты [2, 28]. Может также возникнуть необходимость пересчета значений базовых метрик с учетом особенностей организации или исследуемой системы.

Переход от отдельных уязвимостей к их классам мог бы снизить трудозатраты на анализ и оценку риска. Классификация уязвимостей может быть проведена по-разному в зависимости от предметной области и уровня детальности системы, но в целом выделяют следующие типы уязвимостей, отражающие природу активов [13]:

– аппаратные уязвимости,

– уязвимости программного обеспечения,

– сетевые уязвимости.

В работе будут рассматриваться только уязвимости программного обеспечения, потому что обновления безопасности в основном сосредоточены на них.

Для перехода от отдельных реализаций уязвимостей, которые в большинстве случаев уникальны, к типам применим подход [14], связывающий уязвимости с недостатками программы. В качестве классификатора недостатков по типам возьмем один из наиболее известных – CWE [19].

Этот перечень имеет вид иерархической свободно дополняемой таксономии дефектов программного и аппаратного обеспечения, которая может использоваться в инструментах анализа безопасности.

Классификатор представляет собой древовидную многоуровневую структуру, в которой недостатки распределены по четырем уровням: корневой, базовый, уровень класса, уровень вариантов. Для облегчения работы в CWE выделены так называемые представления – набор записей CWE, предназначенный для решения отдельных типов задач (например, разработки ПО, разработки оборудования, исследований).

Отнесение уязвимостей в CWE производится экспертами вручную, и как показывает практика, во многих случаях мнения экспертов расходятся [29]. Ведутся также исследования относительно полностью автоматической классификации, но однозначных результатов пока нет (см., например, работу [23]).

Каталог CWE можно использовать для создания системы безопасной разработки, но, по мнению авторов, для классификации уязвимости в целях создания системы защиты он малопригоден, так как связь класса CWE с методами атак (например, атаками из классификатора CAPEC [30]) и моделью нарушителя неоднозначна, потому что один и тот же недостаток ПО может использоваться в различных сценариях атак разными нарушителями и приводить к разным последствиям.

1.2. Анализ угроз и характеристик нарушителей

Угроза в рамках кибербезопасности может быть определена как совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации [12].

Виды угрозы обычно соотносятся со способом использования уязвимостей, каждый из которых имеет разные последствия и предпосылки; как правило, виды угрозы можно сопоставить с нарушением свойств кибербезопасности в одной из эталонных моделей (например, в эталонной модели КЦД – конфиденциальность, целостность, доступность).

Отображение свойств кибербезопасности на свойства самой системы индивидуально для каждой конкретной системы, и в зависимости от анализируемой системы каждый вид угрозы может повлиять на любое из свойств системы, как то: надежность, доступность, ремонтопригодность, безопасность. Есть несколько распространенных классификаторов видов угроз:

– банк данных угроз безопасности информации ФСТЭК России [6];

– MITRE ATT&CK [31];

– классификатор угроз STRIDE компании Microsoft [32].

Нарушитель (агент угрозы, злоумышленник) – это активный элемент, субъект в системе, который пытается использовать уязвимость. Примерами возможных нарушителей являются хакеры, компьютерные преступники, террористы, промышленные шпионы, инсайдеры [33]. Подробная классификация злоумышленников содержится в каталогах ФСТЭК, CAPEC, MITRE ATT&CK и др. [12, 30, 31]. Каждый из классификаторов содержит набор атрибутов, по которым могут быть распределены нарушители, например, по типу нарушителей, уровню их компетентности и оснащенности, целям атаки. Нарушители в рамках модели ФСТЭК дополнительно подразделяются на внешних и внутренних. Подходы ИСО/МЭК 27000 и MITRE ATT&CK атрибутируют нарушителя по целям атаки: получение денег, подрыв репутации, получение конкурентного преимущества и т. п.

В качестве примера атрибутирования по документам ФСТЭК приведем выделенные в них типы нарушителей и уровни их компетенции.

Типы нарушителей по ФСТЭК:

- специальные службы иностранных государств;
- террористические, экстремистские группировки;

• преступные группы (криминальные структуры);

• отдельные физические лица (хакеры);

• конкурирующие организации:

• разработчики программных, программно-аппаратных средств;

• лица, обеспечивающие поставку программных, программно-аппаратных средств, обеспечивающих систем;

• поставщики услуг связи, вычислительных услуг;

• лица, привлекаемые для установки, настройки, испытаний, пусконаладочных и иных видов работ;

• лица, обеспечивающие функционирование систем и сетей или обеспечивающих систем оператора (администрация, охрана, уборщики и др.);

• авторизованные пользователи систем и сетей.

Уровни компетентности нарушителя по ФСТЭК:

• базовые возможности по реализации угроз безопасности информации (Н1);

• повышенные возможности по реализации угроз безопасности информации (Н2);

• средние возможности по реализации угроз безопасности информации (Н3);

• высокие возможности по реализации угроз безопасности информации (Н4).

Указанный перечень видов нарушителей может быть дополнен иными нарушителями с учетом особенностей области деятельности, в которой функционируют системы, и связи анализируемой системы с внешним миром.

1.3. Выводы по анализу составляющих модели угроз

Из вышесказанного становится ясно, что работа по анализу отдельных компонентов модели угроз и оценке риска от использования уязвимостей злоумышленниками требует привлечения на постоянной основе специалистов, владеющих широким диапазоном знаний и навыков программирования, оценки риска информационной безопасности, а также обладающих экспертными знаниями как на системном уровне, так и на уровне отдельных компонентов.

Эта работа весьма трудоемка и не ограничивается функциями, связанными с анализом риска для уязвимости, ведь если выявлена необходимость устранения уязвимости с применением патча, то это приводит к дополнительным работам и проблемам для собственника системы.

2. ПРОБЛЕМЫ УПРАВЛЕНИЯ ПАТЧАМИ

В § 1 были приведены основные шаги, предшествующие решению об установке патча, и связанные с ними проблемы. Однако трудности этим не исчерпываются, и в случае положительного решения об установке патча собственник системы сталкивается с целым рядом новых проблем, обусловленных сложностью рассматриваемого класса систем:

• Для компонентов в составе системы обновление безопасности программного компонента часто не выпускается отдельно, а включается в состав новой версии этого компонента, тогда функциональность новой версии компонента может потребовать дополнительного тестирования компонента в составе системы. Замена существующей версии программного компонента может привести к отказу некоторых функций АСУ ТП и к необходимости доработки ПО АСУ ТП.

• Программные компоненты в составе комплекса программ связаны друг с другом цепочкой зависимостей. Зависимость может иметь как горизонтальный характер, например, на уровне прикладных компонентов программы, так и вертикальный – на уровне компонентов операционной системы. Замена любого из ключевых компонентов может повлечь за собой замену остальных и в худшем случае – замену всех компонентов в цепочке зависимостей. Для системы с длительным жизненным циклом может возникнуть такая ситуация, что часть компонентов, входящих в состав цепочки зависимостей, более не поддерживается разработчиком, для них не существует новых версий, и обновление в этом случае нельзя провести простым переходом на новую версию.

• Для АСУ ТП характерны длительный жизненный цикл (период эксплуатации может составлять десятки лет) и жесткие требования к порядку их разработки и тестирования. Учитывая высокую скорость обнаружения новых уязвимостей, можно предположить, что за время между выпуском версии ПО АСУ ТП и запуском системы в эксплуатацию после пусконаладки в программном окружении АСУ ТП найдут новые уязвимости. Необходимо также учитывать, что производители сторонних компонентов могут прекратить поддержку устаревших версий своих продуктов, и для новых уязвимостей будут отсутствовать обновления безопасности.

• Программное обеспечение АСУ ТП тестируется и сертифицируется для работы на определен-

ных технических средствах в заданном программном окружении, и применение патча в зависимости от условий действия сертификата на систему может привести к необходимости дорогостоящей и длительной процедуры повторной сертификации системы.

Немаловажным фактором является вопрос доверия к источнику обновлений и к разработчикам программного обеспечения. Как показывает практика, риск того, что полученное обновление может содержать намеренно встроенные уязвимости, существует [2, 34, 35]. Причем доверие к источнику обновлений может меняться с течением времени от широкого применения программ до их запрета только на основании оценок риска, связанного с социальной и политической ситуацией [36, 37].

Коснемся вопроса тестирования обновлений. Установка и тестирование обновлений могут потребовать испытаний работы всей АСУ ТП. Одним из решений, позволяющих в большинстве случаев провести полноценное тестирование, сравнимое с тестированием на реальном объекте, является использование гибридного цифрового двойника, в котором совместно с чисто цифровыми компонентами используются элементы реального оборудования АСУ ТП [38].

Таким образом, практика последовательного анализа отдельных уязвимостей в компонентах и установки патчей этих компонентов может использоваться только на простых системах. Если же речь идет о больших, сложных системах с длительным периодом эксплуатации, то необходимо искать другие решения проблемы обеспечения кибербезопасности.

Отметим, что в обзоре публикаций [17] составлен перечень описанных в литературе проблем, и, по мнению авторов, наиболее важными являются следующие из них:

– задачи применения патчей для обеспечения информационной безопасности и текущие задачи организации (например, непрерывности бизнеса или технологических процессов) могут быть несовместимы ([17], *п. 2 табл. 5*);

– процесс применения патчей требует дополнительных ресурсов и экспертных знаний, которых нет в компаниях, эксплуатирующих ПО ([17], *п. 5, п. 6 табл. 5*);

– автоматическое тестирование патчей крайне затруднительно, большинство патчей тестируется вручную и качество тестирования зачастую неудовлетворительно ([17], *п. 11, п. 12 табл. 5*);

– проверка корректности применения патча и устранения последствий ошибок развертывания – довольно трудная задача ([17], *п. 13, п. 14 табл. 5*).

Ниже опишем подход, который позволяет, оставаясь в рамках риск-ориентированного подхода, существенно упростить анализ уязвимостей.

3. РИСК-ОРИЕНТИРОВАННЫЙ ПОДХОД К АНАЛИЗУ УЯЗВИМОСТЕЙ С УЧЕТОМ МЕР ЗАЩИТЫ

Предлагаемый подход разработан с учетом его применимости к системам, обладающим следующими характеристиками:

– программная среда системы рассматривается как набор отдельных компонентов, представленных в виде черного ящика и взаимодействующих друг с другом посредством известного набора интерфейсов;

– система создавалась с применением архитектурных принципов инкапсуляции и разделения на домены [39].

Будем рассматривать систему с точки зрения выполнения определенных функций и учитывать вклад каждого компонента в функцию системы.

Основная идея предлагаемого подхода заключается в том, что анализ уязвимостей ПО, составляющего систему, должен дать ответ на вопрос, насколько уязвимость опасна для той или иной функции системы, а не насколько она критична для того или иного программного компонента.

Предлагаемый в данной работе метод анализа уязвимостей основывается на методе анализа видов и последствий отказов (FMEA), который предназначен для выявления слабых мест в архитектуре системы, чтобы повысить таким образом ее надежность и безопасность [18]. Метод FMEA был разработан в 1940-х гг. [40] и первоначально был нацелен исключительно на анализ надежности или безопасности технических систем и оборудования. Позднее были предложены модификации метода для анализа задач кибербезопасности систем с цифровыми компонентами (SFMEA) [41, 42]. В настоящей работе приводятся общие сведения о методе FMEA, подробности читатель может узнать из обширной специальной литературы. В рамках FMEA система должна обладать следующими свойствами:

– иметь определенные цели, представленные в виде требований к ее функциям;

– иметь установленные условия функционирования;

– иметь определенные границы;

– иметь иерархическую структуру.

Вместе с блок-схемой иерархической структуры элементов в FMEA используются блок-схемы, отражающие иерархию выполняемых элементами

системы функций и функциональные взаимосвязи между элементами, что обеспечивает прослеживаемость функциональных отказов системы.

Для каждого компонента системы анализируются (рис. 1):

– причины, которые могут привести данному отказу;

– функция, отказ, который может произойти (вид отказа);

– особенности последствий в случае отказа;

– тяжесть отказа (является ли отказ безвредным или наносит ущерб);

– критичность отказа (как и когда отказ можно обнаружить).

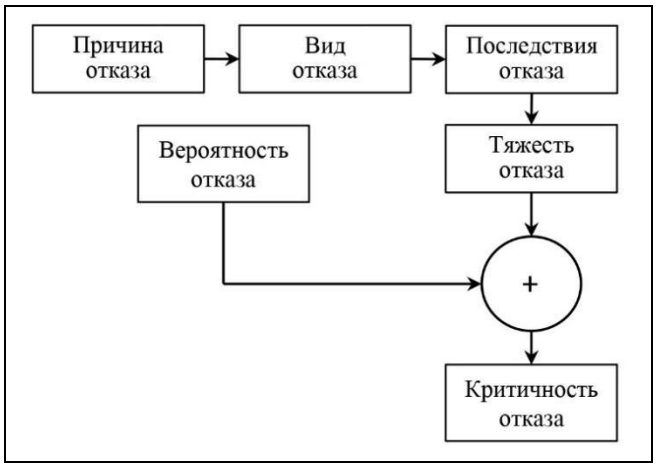

Рис. 1. Одна из возможных последовательностей шагов при оценке критичности отказа согласно стандарту [18]

Применяя подходы FMEA для оценки кибербезопасности, необходимо описать этапы FMEA в терминах кибербезопасности и связать причину отказа с уязвимостью и способностью нарушителя ее использовать.

Предлагается адаптировать методику FMEA для оценки влияния уязвимости и уменьшить таким образом объем анализируемой информации (рис. 2).

Методика учитывает, что уязвимость сама по себе не является причиной отказа: для того, чтобы уязвимость привела к отказу, необходимо, чтобы нарушитель, обладающей достаточной компетенцией, использовал ее для реализации определенной угрозы (см. блок 1 на рис. 2). Нарушитель и его компетенция могут быть типизированы в соответствии с принятой шкалой: например, с использованием классификатора ФСТЭК [12]. Таким образом, отказ – это результат успешной атаки на систему со стороны нарушителя, обладающего достаточной компетенцией и возможностями для эксплуатации уязвимости.

В предлагаемой методике для уменьшения пространства анализа постулируется независимость уязвимостей, что аналогично предположению о независимости отказов в методике FMEA ([18], *п. 4.1*).

Предлагаемая схема анализа, в отличие от типичного рассмотрения отдельных технологий, лежащих в основе уязвимости (как, например, в каталоге CWE), акцентируется на атрибутах CVSS для соотнесения уязвимости и ассоциированной с ней угрозы. Дополнительно выполняется переход от анализа отдельных угроз и нарушителей к анализу типовых сценариев атак, связанных с классами угроз и типовыми возможностями нарушителя, что снижает количество анализируемых угроз и сценариев атак. Примеры типовых сценариев атак для АСУ ТП АЭС описаны в работе [43].

Отказ функции в этом случае связан не с конкретной уязвимостью (см. рис. 2), а с совокупностью условий и факторов, которые привели к отказу компонента и преодолению нарушителем барьера защиты. Необходимо отметить, что отказ может быть непосредственно не связан с (кибер)атакой и с использованием уязвимости, а, например, являться результатом исчерпания ресурса. Однако режимы работы АСУ ТП критических объектов проектируются таким образом, чтобы исключить исчерпание ресурсов.

Вторым важным дополнением к применению подходов FMEA к задачам кибербезопасности является учет наличия барьера, который отражает действие уже реализованного набора мер защиты. Барьер может блокировать влияние отказа компонента на функции всей системы и, следовательно, свести к нулю риски, связанные с атакой на систему. Предполагается, что барьер обладает высокой степенью доверия и способен противодействовать эксплуатации одного или нескольких типов уязвимостей. Таким образом, при анализе сценариев атаки барьер считается абсолютным. Это модельное предположение позволяет существенно уменьшить объем анализа.

Наличие нарушителя, у которого есть мотивация и цель, означает, что отказ в кибербезопасности в общем случае не является случайным событием. Поэтому применение статистических подходов для анализа отказов, как правило, невозможно,

но возможно применение риск-ориентированного подхода и логических правил.

Если допускается вероятностный характер воздействия нарушителя на систему, то влияние отказа на всю систему и связанные с этим риски (см. блок 2 рис. 2) оцениваются с применением методов анализа функциональной зависимости FMEA и разработанных теоретико-вероятностных подходов [18, 44].

Описанный подход можно свести в метод интегрированной защиты и оценки уязвимостей, названный авторами Vulnerability Inspection Control Strategy (VICS), схематически он приведен на рис. 3.

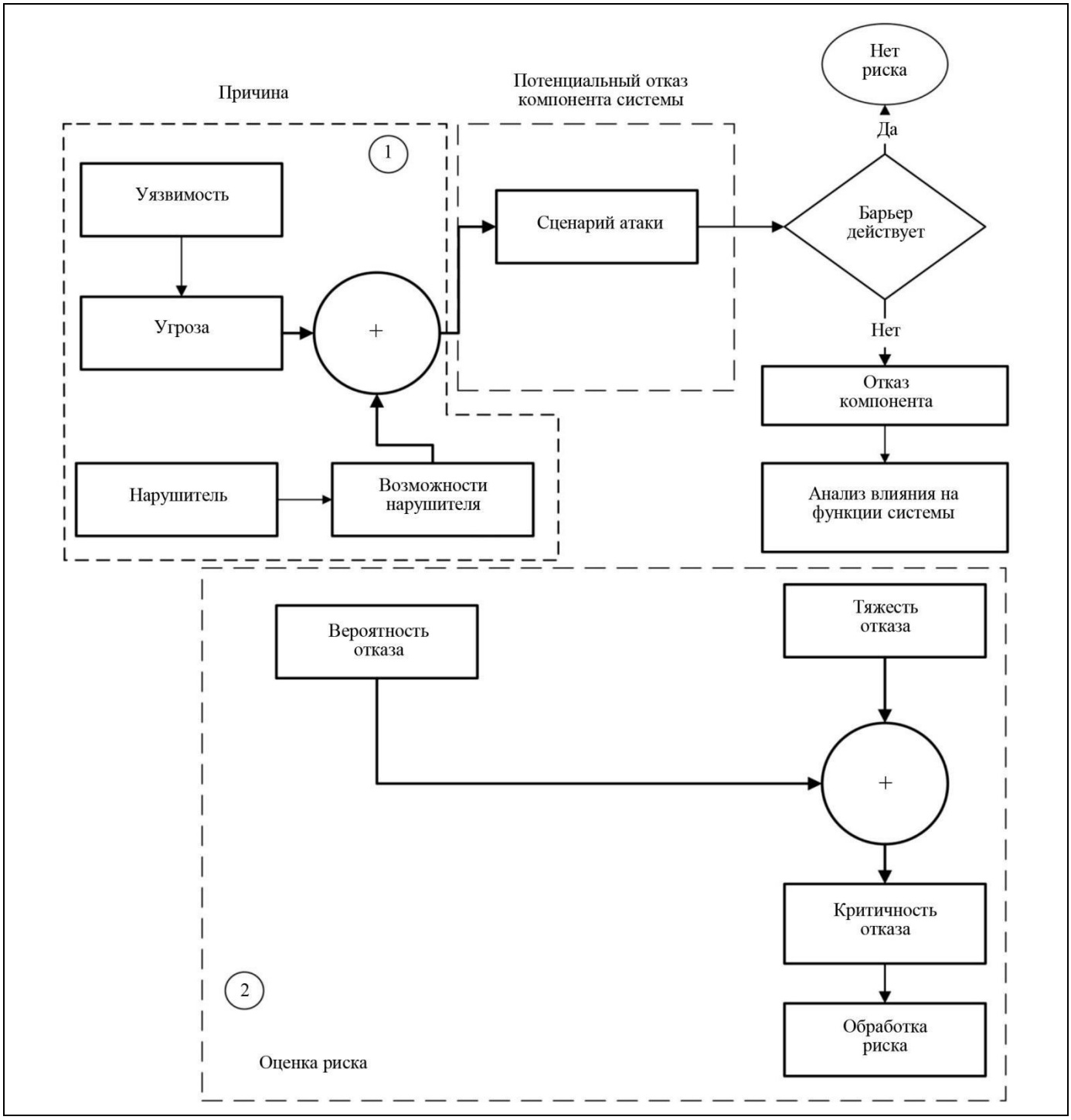

Рис. 2. Комбинированная последовательность шагов при оценке критичности отказа с учетом барьеров защит для задач надежности и кибербезопасности

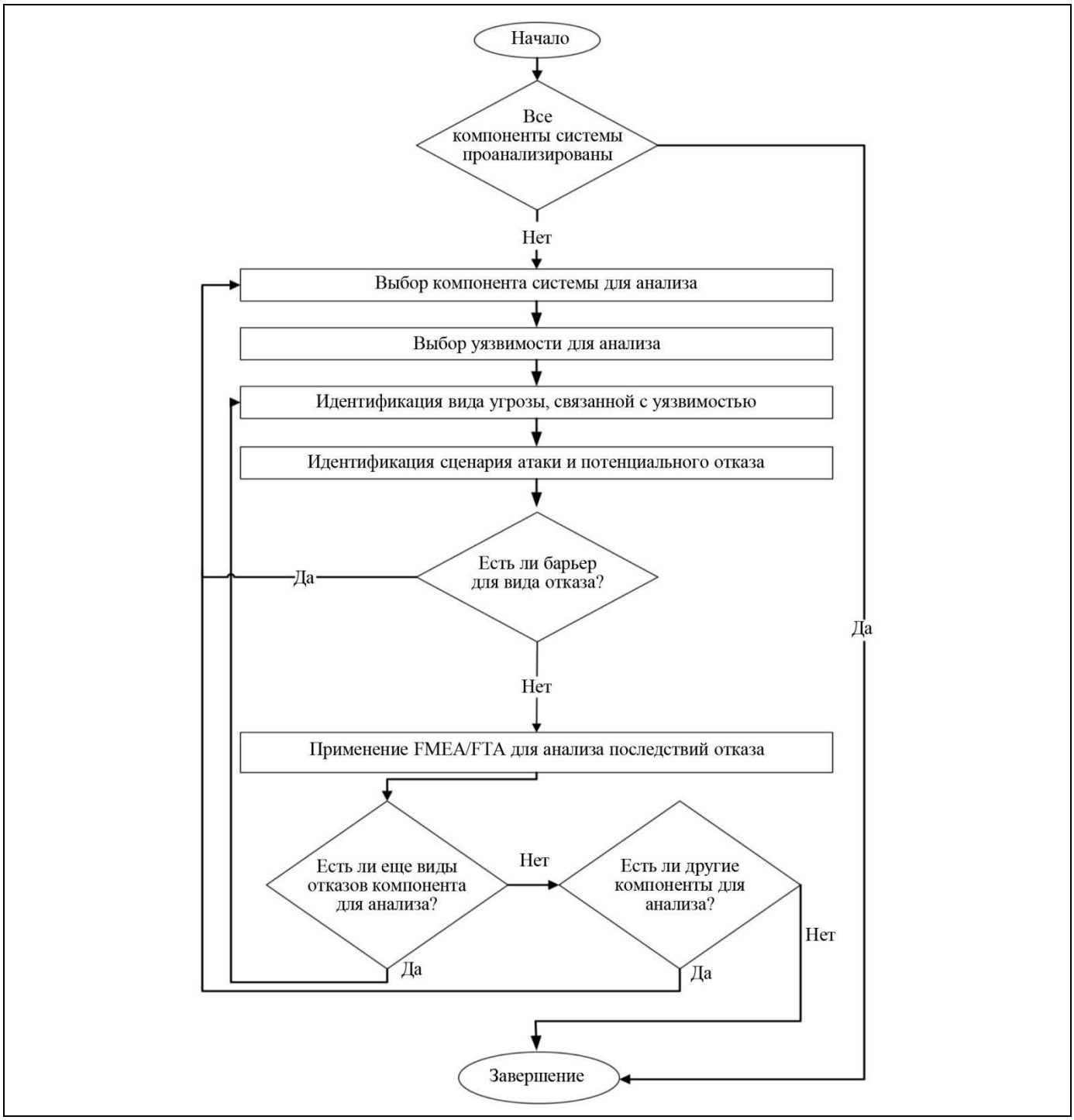

Рис. 3. Комбинированная блок-схема анализа методом интегрированной защиты и оценки уязвимостей VICS

Рассмотрим последовательность действий при анализе уязвимостей и необходимости применения патчей для современной АСУ ТП [45] в рамках VICS. Исходными данными для анализа являются:

– список анализируемых уязвимостей;

– принятая система классификации угроз, действующих на систему (например, модель КЦД или классификатор ФСТЭК);

– принятая модель нарушителя;

– принятые типовые сценарии атак, позволяющие определить для пары «класс угрозы – компетенция нарушителя» потенциальный вид отказа компонента;

– список функций системы, в котором для каждой из функций определены негативные последствия ее нарушения;

– таблица соответствия «отказ функции – барьер», описывающая биграф, в котором для каждого вида отказа определяется барьер в соответствии с реализуемыми мерами защиты;

– структурная схема системы (с ее помощью можно связать уязвимость с одним или несколькими компонентами системы);

– дерево отказов (англ. *Fault Tree Analysis*, FTA), позволяющее проследить влияние отказа скомпрометированных компонентов на функции системы и сгруппировать их по типам.

Анализ сводится к выполнению следующих действий:

1. Для всех уязвимостей определенным способом сопоставить уязвимость с классом угрозы, типовым сценарием атаки, видом отказа компонента с учетом квалификации и мотивации нарушителя.

2. Проанализировать наличие барьера защиты. Если барьер присутствует, то можно прекратить анализ для данной уязвимости.

3. Если барьер отсутствует или недостаточен для данного типа атаки, то выбрать защитную меру или применить патч, которые нейтрализуют выявленные угрозы безопасности или сводят риск их использования к приемлемому, руководствуясь принятой методикой управления патчами (см., например, руководство [9]). Если принято решение реализовать новую меру защиты, то необходимо внести данный сценарий атаки и барьер, ассоциированный с ним, в таблицу, описывающую биграф.

Таким образом, критериями применения обновления безопасности в качестве меры снижения риска, связанного с уязвимостью, являются:

– отсутствие барьера против данного типа атаки,

– недостаточность барьера против данного типа атаки,

– отсутствие иных более эффективных, чем установка обновления, мер защиты.

Предлагается следующая последовательность действий с уязвимостями информационных систем.

1. При разработке системы защиты проводится анализ угроз и создается каталог барьеров защиты и каталог типовых сценариев атак.

2. На основе двух каталогов формируется биграф, описывающий парирование атак барьерами защиты.

3. Каждая новая уязвимость классифицируется в соответствии с классами угроз, нарушителя и сценариями атаки (см. § 4). Если уязвимость попадает в класс, для которого уже установлен барьер в виде набора мер защиты, то установка патча не требуется.

4. В случае если уязвимость приводит к новому сценарию атаки, то необходимо либо установить патч, либо принять другие меры понижения риска. Они, в частности, могут включать введение новых барьеров защиты, сводящих к минимуму последствия данного сценария атаки.

5. Периодически проводится анализ корректности мер защиты, формирующих барьеры. В случае если защитные меры по результатам анализа будут сочтены недостаточными, должны быть разработаны и внедрены другие или дополнительные (включая установку патчей).

Отметим, что стратегия VICS не покрывает уязвимости в программных барьерах защиты и корректность самой среды функционирования. Однако, так как рассматриваемые системы обычно строятся с учетом высокого доверия к корректности реализованных мер защиты и качеству эксплуатации системы, то число уязвимостей в барьерах, как правило, много меньше количества всех уязвимостей системы, и, следовательно, предлагаемый метод покрывает подавляющее большинство уязвимостей. Для управления уязвимостями в барьерах защиты можно использовать существующие нормативные методики (см., например, [3, 8, 9]).

Далее на примере построения системы защиты для системы верхнего уровня АСУ ТП будет продемонстрирована применимость метода VICS на практике.

4. ПРИМЕНЕНИЕ МЕТОДА НА ПРАКТИКЕ

Принципиальную применимость метода VICS продемонстрируем на примере системы верхнего уровня (СВУ) АСУ ТП. Вкратце опишем ее основные функции и свойства (с более подробным рассмотрением систем такого класса можно ознакомиться, например, в работе [45]).

Система верхнего уровня АСУ ТП предназначена:

– для реализации информационных, управляющих и вспомогательных функций;

– для передачи команд оператора по управлению технологическими процессами и технологическим оборудованием;

– для контроля состояния АСУ ТП;

– для интеграции информации от систем АСУ ТП.

Рассматриваемая СВУ АСУ ТП состоит из серверов, которые занимаются сбором и архивированием информации от смежных систем и команд оператора, и рабочих станций, где отображается

информация о состоянии АСУ ТП и вводятся команды управления. Все элементы СВУ резервированы и связаны резервированной сетью. Доступа в Интернет АСУ ТП не имеет. Допускается передача однонаправленного потока данных из АСУ ТП наружу (например, через диод данных).

В качестве модели угроз примем модель угроз КЦД, с которой свяжем основные типы угрозы нарушения конфиденциальности, целостности и доступности. Такой подход, несомненно, является упрощением, но он следует практике, отраженной во многих широко используемых классификаторах безопасности (например, CWE, CVSS). В качестве модели нарушителя примем модель нарушителя с низким или средним уровнем привилегий (права администратора у такого нарушителя отсутствуют) и средними возможностями по реализации угроз безопасности информации (уровень компетентности нарушителя по ФСТЭК Н3). Предположим, что с помощью локальной атаки такой нарушитель хочет нарушить целостность программного обеспечения или данных (сюда входит неавторизованный запуск команд) и выполнить таким способом команду управления, которая приведет к физическому ущербу для объекта управления.

Предположим, что при проектировании АСУ ТП был заложен набор мер защиты, который формирует барьер, препятствующий доступу непривилегированного пользователя со средним уровнем компетенции к системному ПО. Для АСУ ТП меры защиты могут быть выбраны из перечня [46]. Также будем считать, что на момент начала атаки система не является скомпрометированной. Тогда фрагмент биграфа для данного сценария атаки и барьера представлен на рис. 4.

Перейдем теперь непосредственно к анализу уязвимостей. Из базы данных уязвимостей необходимо выгрузить описания уязвимостей, относящихся к используемому в СВУ АСУ ТП программному обеспечению с учетом версий компонентов. Для облегчения работы можно воспользоваться каким-либо сканером уязвимостей.

Пусть для определенности СВУ АСУ ТП реализовано на базе ОС Linux. Рассмотрим для примера уязвимость в базовом компоненте glibc с номером CVE-2020-1752, которая по классификатору CVSS 3.0 имеет высокую степень опасности. Эта уязвимость позволяет локальному нарушителю, передав программе специальным образом сформированный путь к файлу, выполнить произвольный код и нарушить таким образом целостность программного обеспечения. Все эти свойства уязвимости отражены в векторе CVSS 3.0.

В принятой модели безопасности такой уязвимости соответствует сценарий атаки из таблицы, позволяющий выполнить произвольный код без повышения привилегий; против такой атаки действует барьер, препятствующий эксплуатации уязвимости нарушителем в рассматриваемом сценарии атаки.

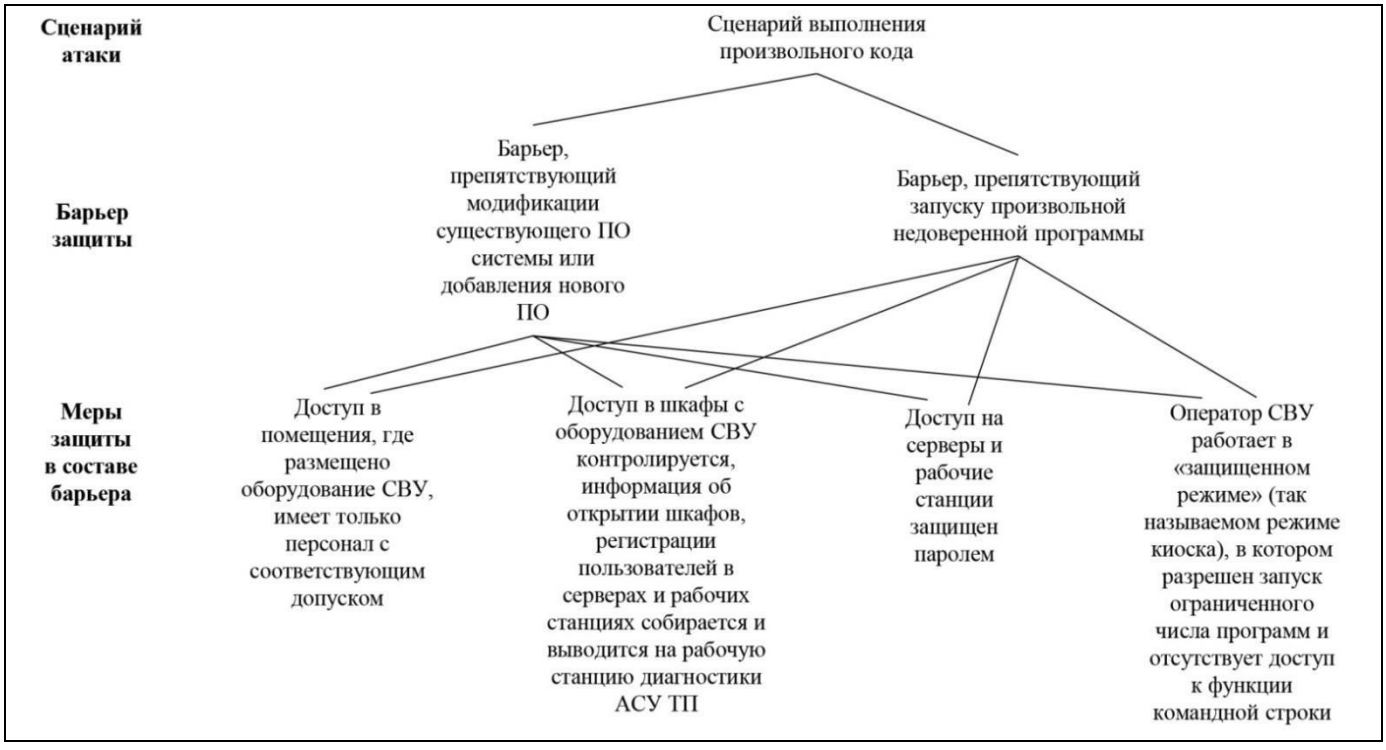

Рис. 4. Биграф, описывающий отношение «сценарий атаки – барьер защиты»

Таким образом, с учетом высокого доверия к способности барьера противостоять неконтролируемому доступу к системному ПО, можно считать, что риск уязвимости минимален и для уязвимости CVE-2020-1752 патчи устанавливать не нужно.

Очевидно, что в отношении уязвимостей, которые могут приводить к сходному сценарию атаки, метод VICS даст такие же результаты.

ЗАКЛЮЧЕНИЕ

Проблемы установки обновлений безопасности стоят довольно остро, что связано с повышением внимания к кибербезопасности во всех областях применения компьютеров. Несмотря на то, что на сегодня имеются выработанные годами рекомендации по использованию патчей [3, 8, 9], применение этих рекомендаций вызывает трудности.

Первая трудность, которая касается информационных систем любого назначения, – это эффект масштаба. На данный момент каждый год открываются тысячи уязвимостей, что делает анализ каждой уязвимости «вручную» неэффективным. Трудности анализа каждой уязвимости по отдельности могут привести к стратегии «патчить все». Однако эта стратегия чревата проблемами с валидацией совместимости компонентов программного обеспечения с оборудованием и друг с другом и нарушением непрерывности защищаемых бизнес-процессов.

Если говорить об АСУ ТП, то их дополнительно отличают долгий жизненный цикл (десятки лет), малая изменчивость аппаратного обеспечения в ходе эксплуатации, достаточно длительный этап разработки и зачастую необходимость сертификации ПО. Отсюда следует, что даже если в принятом к эксплуатации ПО АСУ ТП будут устранены все известные на момент приемки уязвимости, то к моменту завершения пусконаладки на объекте ПО наверняка будет содержать вновь открытые уязвимости, которые невозможно оперативно устранить. Неизменность аппаратного обеспечения приведет к тому, что через несколько лет эксплуатации многие компоненты ПО будет невозможно обновить, не нарушив программно-аппаратную совместимость системы, так как новые версии ПО от сторонних производителей могут перестать поддерживать устаревшее оборудование. Отраслевые правила могут потребовать повторной сертификации после установки обновлений, что дополнительно приведет к затратам времени и средств. Необходимость ресертификации после каждого изменения ПО является существенной и осознается не только разработчиками, но и регулирующими органами. В частности, на международном уровне рассматривается возможность классификации изменений (патчей) по степени их влияния на функциональность ПО и, в зависимости от этого, изменения требований к ресертификации.

Очевидно, что нужен какой-то метод, позволяющий для конкретной системы анализировать большое количество уязвимостей и выдавать рекомендации о необходимости установки патчей, при этом не требующий полного и детального анализа каждой уязвимости.

Предложенный в настоящей работе метод основан на методе анализа отказов и последствий FMEA, в котором рассматривается не уязвимость сама по себе, а влияние уязвимости как части сценария атаки, с учетом принятой модели угроз и нарушителя, на функции всей системы и с учетом существующих барьеров (наборов мер защиты, эффективных в отношении определенного сценария атаки).

В соответствии с принципами безопасного проектирования в систему встраиваются определенные наборы мер защиты, которые формируют гарантированные барьеры для определенных классов уязвимостей, и на этапе эксплуатации именно эти барьеры задействуются.

Отношения «сценарий атаки – барьер» составляют бираф, а анализ рисков при появлении новых уязвимостей сводится к его анализу.

Если в результате обнаружения новой уязвимости появится сценарий атаки, для которого барьер защиты не установлен или неэффективен, то это означает, что новый источник риска требует принятия дополнительных мер в виде установки обновления безопасности или применения иных компенсирующих мер. Это и есть критерий принятия решений об установке обновлений безопасности.

Так как барьеры рассматриваются в контексте защиты от классов уязвимостей, а не от отдельной уязвимости, то данный подход позволяет бороться не только с открытыми, известными уязвимостями, но и предлагает защиту от еще не открытых уязвимостей, которые всегда присутствуют в сложных программных продуктах.

В заключение отметим, что в сложной системе ручной анализ уязвимостей даже при небольшом количестве сценариев атаки, видов нарушителей и угроз является крайне затруднительным и сводится к задаче, относящейся к массовой проблеме. Авторы полагают, что автоматизация анализа уязвимостей с применением формальных проблемно-ориентированных языков описания уязвимости и

сценариев атаки дает возможность применить разработанный метод к реальным системам управления, и ведут работу в этом направлении.

ЛИТЕРАТУРА

1. *IEC* TR 62443-2-3. Technical report. Security for industrial automation and control systems – Part 2-3: Patch management in the IACS environment. – Geneva: International Electrotechnical Commission, 2015. – 61 p.
2. *Методика* тестирования обновлений безопасности программных, программно-аппаратных средств: утв. ФСТЭК России 28 октября 2022 г. [*Metodika* testirovaniya obnovlenii bezopasnosti programmnykh, programmno-apparatnykh sredstv: appr. by FSTEC of Russia on October 28, 2022. (In Russian)]
3. S*ouppaya, M., Scarfone, K.* Guide to enterprise patch management planning: preventive maintenance for technology. Special Publication (NIST SP) – 800-40r4. – Gaithersburg, MD: National Institute of Standards and Technology, 2022. – DOI: https://doi.org/10.6028/NIST.SP.800-40r4
4. *National* Information Assurance (IA) Glossary. CNSS Instruction No. 4009. – Fort Meade: Committee on National Security Systems Instruction, 2015. – 103 p.
5. *CVE*. – URL: https://cve.mitre.org (дата обращения: 02.10.2024). [Accessed October 2, 2024].
6. *Банк* данных угроз безопасности информации. – URL: https://bdu.fstec.ru/vul (дата обращения: 02.10.2024). [*Bank* dannykh ugroz bezopasnosti informatsii. – URL: https://bdu.fstec.ru/vul (accessed October 2, 2024). (In Russian)]
7. *CERT-FR* avis. – URL: https://www.cert.ssi.gouv.fr/avis/ (дата обращения: 02.02.2024). [Accessed February 2, 2024].
8. *ISO/IEC* TS 9569:2023. Technical specification. Information security, cybersecurity and privacy protection – Evaluation criteria for IT security – Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045. – Geneva: International Standard Organization/IEC, International Electrotechnical Commission, 2023. – 36 p.
9. Руководство по организации процесса управления уязвимостями в органе (организации): утв. ФСТЭК России 17 мая 2023 г. [*Rukovodstvo* po organizatsii protsessa upravleniya uyazvimostyami v organe (organizatsii): appr. by FSTEC of Russia on May 17, 2023. (In Russian)]
10. Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств: утв. ФСТЭК России 28 октября 2022 г. [*FSTEHK* Rossii. Metodika otsenki urovney kritichnosti uyazvimostey programmnykh, programmno-apparatnykh sredstv: appr. by FSTEC of Russia on October 28, 2022. (In Russian)]
11. *Scarfone, K., Souppaya, M., Dodson, D.* Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. Special Publication (NIST SP) - 800-218. – Gaithersburg, MD: National Institute of Standards and Technology, 2022. – DOI: https://doi.org/10.6028/NIST.SP.800-218
12. Методика оценки угроз безопасности информации: утв. ФСТЭК России 5 февраля 2021 г. [*Metodika* otsenki ugroz bezopasnosti informatsii: appr. by FSTEC of Russia on February 5, 2021. (In Russian)]
13. *ГОСТ* Р 56205-2014 IEC/TS 62443-1-1:2009. Сети коммуникационные промышленные. Защищенность (кибербезопасность) сети и системы. Часть 1-1. Терминология, концептуальные положения и модели: стандарт. – М.: Стандартинформ, 2020. – 72 р. [*IEC/TS* 62443-1-1:2009 Industrial communication networks. – Network and system security. – Part 1-1: Terminology, concepts and models (IDT). – Geneva: International Electrotechnical Commission, 2009. – 81 p.]
14. *ГОСТ* Р 56939-2016. Защита информации. Разработка безопасного программного обеспечения. Общие требования. – М.: Стандартинформ, 2016. – 24 с. [*GOST* R 56939-2016. Zashchita informatsii. Razrabotka bezopasnogo programmnogo obespecheniya. Obshchie trebovaniya. – Moscow: Standartinform, 2016. – 24 s. (In Russian)]
15. *ГОСТ* 19.101-77. Единая система программной документации. Виды программ и программных документов. – М.: Стандартинформ, 2010. [*GOST 19.101-77.* Edinaya sistema programmnoi dokumentatsii. Vidy programm i programmnykh dokumentov. – Moscow: Standartinform, 2010. (In Russian)]
16. *Промыслов В.Г., Жарко Е.Ф.* Подходы к оценке риска кибербезопасности АСУТП АЭС // Автоматизация в промышленности. – 2022. – № 11. – С. 28–33. [*Promyslov, V.G., Zharko, E.F.* Approaches to risk assessment in cybersecurity of A-plant process control systems // Automation in Industry. – 2022. – No. 11. – P. 28–33. (In Russian)]
17. *Dissanayake, N., Jayatilaka, A., Zahedi, M., AliBabar, M.* Software security patch management - A systematic literature review of challenges, approaches, tools and practices // Information and Software Technology. – 2022. – Vol. 144. – Art. no. 106 771.
18. *ГОСТ* Р 51901.12-2007 (МЭК 60812:2006) Менеджмент риска. Анализ видов и последствий отказов.: стандарт. – М.: Стандартинформ, 2008. – 35 р. [*IEC* 60812:2006 Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA). – Geneva: International Electrotechnical Commission, 2006. – 93 p.]
19. *About* CWE. – URL: https://cwe.mitre.org/about/index.html (дата обращения: 22.04.2024). [Accessed April 22, 2024].
20. *ГОСТ* Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности. – М.: Стандартинформ, 2011. – 51 с. [*GOST* R ISO/MEK 27005-2010. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Menedzhment riska informatsionnoi bezopasnosti. – M.: Standartinform, 2011. – 51 s. (In Russian)]
21. *ГОСТ* Р ИСО/МЭК 15408-3-2008. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности. – М.: Стандартинформ, 2008. – 118 с. [*GOST* R ISO/MEK 15408-3-2008. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologii. Chast' 3. Komponenty doveriya k bezopasnosti. – M.: Standartinform, 2008. – 118 s. (In Russian)]
22. *CVE* Metrics. – URL: https://www.cve.org/About/Metrics (дата обращения: 22.04.2024). [Accessed April 22, 2024].
23. *Haddad, A., Aaraj, N., Nakov P.,* et al. Automated Mapping of CVE Vulnerability Records to MITRE CWE Weaknesses. – arXiv:2304.11130, 2023. – DOI: 10.48550/arXiv.2304.11130
24. *Lin, Y.-Z., Mamun, M., Chowdhury, V.A.,* et al. HW-V2W-Map: Hardware Vulnerability to Weakness Mapping Framework for Root Cause Analysis with GPT-assisted Mitigation Suggestion. – arXiv:2312.13530, 2023. – DOI: https://doi.org/10.48550/arXiv.2312.13530
25. *Mell, P., Scarfone, K., Romanosky, S*. A Complete Guide to the Common Vulnerability Scoring System Version 2.0. – 2007. – 23 p. – URL: https://www.first.org/cvss/v2/guide (дата обращения: 01.04.2024). [Accessed April 1, 2024].
26. *Common* Vulnerability Scoring System v3.0: Specification Document. – URL: https://www.first.org/cvss/v3.0/specificati

on-document (дата обращения: 22.09.2024). [Accessed September 22, 2024].
27. *Калькулятор* CVSS V3. – URL: https://bdu.fstec.ru/calc3 (дата обращения: 22.09.2024). [*Calculiator* besjpasonosti. – URL: https://bdu.fstec.ru/calc3 (accessed September 22, 2024). (In Russian)].
28. *Kekül, H., Ergen, B., and Arslan, H*. Comparison and Analysis of Software Vulnerability Databases // Int. J. Eng. Manuf.. – 2022. – Vol. 12, no. 4. – P. 1–14.
29. *How* We Assess Acceptance Levels. – URL: https://nvd.nist.gov/vuln/cvmap/How-We-Assess-Acceptance-Levels (дата обращения: 26.04.2024). [Accessed April 26, 2024].
30. *CAPEC* List. – URL: https://capec.mitre.org/data/index.html (дата обращения: 31.05.2024). [Accessed April 22, 2024].
31. *MITRE* ATT&CK. – URL: https://attack.mitre.org/ (дата обращения: 22.04.2024). [Accessed April 22, 2024].
32. *Microsoft* Threat Modeling Tool. – URL: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats (дата обращения: 31.05.2024). [Accessed May 31, 2024].
33. *Pietre-Cambacedes, L., Chaudet, C.* Disentangling the relations between safety and security // Proceedings of the 9th WSEAS International Conference on Applied Informatics and Communications. – Stevens Point, Wisconsin, 2009. – P. 156–161.
34. *Boehs, E.* Everything I Know About the XZ Backdoor. – URL: https://boehs.org/node/everything-i-know-about-the-xz-backdoor (дата обращения: 10.04.2024). [Accessed April 10, 2024].
35. *CVE-2022-23812*. – URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-23812 (дата обращения: 10.04.2024). [Accessed April 10, 2024].
36. *Капранов О., Гуреева Ю.* Сотрудникам Минпросвещения запретили пользоваться техникой Apple. – Текст: электронный // Российская газета. – 19.07.2023. – URL: https://rg.ru/2023/07/19/sotrudnikam-minprosveshcheniia-zapretili-polzovatsia-tehnikoj-apple.html (дата обращения: 22.10.2024). [*Kapranov O., Gureeva Yu.* Sotrudnikam Minprosveshcheniya zapretili pol'zovat'sya tekhnikoi Apple. – Tekst: elektron-nyi // Rossiiskaya gazeta. – July 7, 2023. – URL: https://rg.ru/2023/07/19/sotrudnikam-minprosveshcheniia-zapretili-polzovatsia-tehnikoj-apple.html (accessed October 22, 2024). (In Russian)]
37. *Commerce* Department Prohibits Russian Kaspersky Software for U.S. Customers // Bureau of Industry and Security. – June 20, 2024. – URL: https://www.bis.gov/press-release/commerce-department-prohibits-russian-kaspersky-software-us-customers (дата обращения: 20.12.2024). [Accessed December 20, 2024].
38. *Semenkov, K., Promyslov, V., Poletykin, A.,* et al. Validation of Complex Control Systems with Heterogeneous Digital Models in Industry 4.0 Framework // Machines. – 2021. – Vol. 9. – No. 3. – Art. no. 62.
39. *ГОСТ* ISO/IEC TS 19249-2021. Информационные технологии. Методы и средства обеспечения безопасности. Каталог принципов построения архитектуры и проектирования безопасных продуктов, систем и приложений. – М.: Стандартинформ, 2021. – 32 с. [*GOST* ISO/IEC TS 19249-2021. Informatsionnye tekhnologii. Metody i sredstva obespecheniya bezopasnosti. Katalog printsipov postroeniya arkhitektury i proektirovaniya bezopasnykh produktov, sistem i prilozhenii. – M.: Standartinform, 2021. – 32 s. (In Russian)]
40. *MIL-P* 1629. USA Military Standard, Procedure for Performing a Failure Mode, Effects and Criticality Analysis. – Washington, DC: Department of Defense, 1980. – 54 p.
41. *Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E*. Security Application of Failure Mode and Effect Analysis (FMEA) // In: Lecture Notes in Computer Science. – 2014. – Vol. 8666. – P. 310–325. – DOI: https://doi.org/10.1007/978-3-319-10506-2_21
42. *Talwar, P.* Software Failure Mode and Effects Analysis // Advances in Intelligent Systems and Computing. – 2020. – Vol. 1131. – P. 86–91.
43. *Busquim e Silva, R.A., Piqueira, J.R.C., Cruz, J.J., Marques R.P.* Cybersecurity Assessment Framework for Digital Interface Between Safety and Security at Nuclear Power Plants // International Journal of Critical Infrastructure Protection. – 2021. – Vol. 34. – Art. no. 100453. – DOI: https://doi.org/10.1016/j.ijcip.2021.100453.
44. *Бугайский К.А., Калашников А.О., Бирин Д.С.* и др. Применение логико-вероятностного подхода в информационной безопасности (Часть 1) // Cybersecurity Issues. – 2023. – No. 4 (56). – P. 23–32. [*Kalashnikov, A.O., Bugajskij, K.A., Birin D.S.,* et al. Application of the logical-probabilistic method in information security (part 1) // Cybersecurity Issues. 2023. – No. 4 (56). – P. 23–32. (In Russian)]
45. *Менгазетдинов Н.Э., Полетыкин А.Г., Промыслов В.Г.* и др. Комплекс работ по созданию первой управляющей системы верхнего блочного уровня АСУ ТП для АЭС «Бушер» на основе отечественных информационных технологий. – М.: ИПУ РАН. – 2013. – 95 с. [*Mengazetdinov, N.E., Poletykin, A.G., Promyslov, V.G.,* et al. Kompleks rabot po sozdaniyu pervoi upravlyayushchei sistemy verkhnego blochnogo urovnya ASU TP dlya AES “Busher” na osnove otechestvennykh informatsionnykh tekhnologii. – Moscow: IPU RAN. – 2013. – 95 p. (In Russian)]
46. Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации: утв. приказом ФСТЭК России от 25 декабря 2017 г. № 239. [Trebovania po obespecheniyu bezopasnosti znachimyh ob’ektov kriticheskoi informatsionnoi infrastruktury Rossijskoj Federatsii: appr. by FSTEC of Russia on December 25, 2017. (In Russian)]

Статья представлена к публикации членом редколлегии Р.В. Мещеряковым.

Поступила в редакцию 06.06.2024,
после доработки 18.03.2025.
Принята к публикации 28.04.2025.

Семенков Кирилл Валерьевич – канд. физ.-мат. наук,
✉ semenkov@ipu.ru
ORCID iD: https://orcid.org/0000-0003-0865-9072

Промыслов Виталий Георгиевич – канд. физ.-мат. наук,
✉ vp@ipu.ru
ORCID iD: https://orcid.org/0000-0003-1919-8718

Институт проблем управления им. В. А. Трапезникова РАН, г. Москва

A PROCEDURE FOR ASSESSING SECURITY UPDATES IN INDUSTRIAL SYSTEMS

K. V. Semenkov* and V. G. Promyslov**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ semenkov@ipu.ru, **✉ vp@ipu.ru

Abstract. This paper is devoted to the problem of applying cybersecurity updates (patches) for the software of instrumentation and control systems (ICS) with a long lifecycle. The problem is considered for the system operation stage. The main focus is on the large number of vulnerabilities found in software, the complexity of analyzing the impact of a vulnerability on system security, and the requirements for testing the compatibility of updates and software certification after changes have been made. Based on the Failure Mode and Effects Analysis (FMEA), a procedure is proposed to simplify the analysis of the impact of a vulnerability on cybersecurity. This procedure considers a smaller set of attack scenarios rather than each vulnerability separately. The analysis of attack scenarios also covers the effect of security measures. The procedure includes simple criteria for applying security updates based on the analysis results. An example of vulnerability analysis using this procedure is provided.

Keywords: vulnerability, patch, risk assessment, instrumentation and control system (ICS), cybersecurity, criterion.