



*Российская Академия Наук*

# **А Т АВТОМАТИКА И ТЕЛЕМЕХАНИКА**

Журнал основан в 1936 году

Выходит 12 раз в год

**5**

**М А Й**

**Москва**

**2025**

Учредители журнала:

Отделение энергетики, машиностроения, механики и процессов управления РАН,  
Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН),  
Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН)

**Главный редактор:**

Галяев А.А.

**Заместители главного редактора:**

Рубинович Е.Я., Хлебников М.В.

**Ответственный секретарь:**

Самохин А.С.

**Редакционный совет:**

Васильев С.Н., Желтов С.Ю., Каляев И.А., Кузнецов О.П., Кулешов А.П.,  
Куржанский А.Б., Мартынюк А.А. (Украина), Пешехонов В.Г., Попков Ю.С.,  
Федосов Е.А., Черноусько Ф.Л.

**Редакционная коллегия:**

Алескеров Ф.Т., Арутюнов А.В., Бахтадзе Н.Н., Бобцов А.А., Виноградов Д.В.,  
Вишневский В.М., Воронцов К.В., Граничин О.Н., Жилиякова Н.Ю.,  
Каравай М.Ф., Кибзун А.И., Краснова С.А., Крищенко А.П., Кузнецов Н.В.,  
Кушнер А.Г., Лазарев А.А., Ляхов А.И., Маликов А.И., Матасов А.И.,  
Меерков С.М. (США), Мещеряков Р.В., Миллер Б.М., Михальский А.И.,  
Моржин О.В., Мунасыпов Р.А., Назин А.В., Немировский А.С. (США),  
Новиков Д.А., Олейников А.Я., Пакшин П.В., Пальчунов Д.Е.,  
Поляков А.Е. (Франция), Протасов В.Ю., Рапопорт Л.Б., Родионов И.В.,  
Сельвесюк Н.И., Соболевский А.Н., Степанов О.А., Филимонюк Л.Ю.,  
Фрадков А.Л., Хоров Е.М., Цыбаков А.Б. (Франция), Чеботарев П.Ю.,  
Чхартишвили А.Г., Щербаков П.С.

Адрес редакции: 117997, Москва, Профсоюзная ул., 65

Тел./факс: 8 (495) 198-17-20, доб. 1443

Электронная почта: redacsia@ipu.ru

Зав. редакцией *Е.А. Мартехина*

Москва

ФГБУ «Издательство «Наука»

# Стохастические системы

© 2025 г. А.А. НАЗАРОВ, д-р техн. наук (nazarov.tsu@gmail.com),  
Я.Е. ИЗМАЙЛОВА, канд. физ.-мат. наук (evgenevna.92@mail.ru)  
(Национальный исследовательский Томский государственный университет)

## АСИМПТОТИЧЕСКИ-ДИФFUЗИОННЫЙ АНАЛИЗ ПРИОРИТЕТНОЙ RQ-СИСТЕМЫ $M^{(2)}|M^{(2)}|1$

Рассматривается система массового обслуживания с повторными вызовами (RQ-система). Поступает в систему два входящих пуассоновских потока. Первый поток заявок – приоритетный, второй – неприоритетный. Время обслуживания имеет экспоненциальное распределение. Если заявки приоритетного потока обнаруживают прибор занятым заявкой того же класса, то они переходят на орбиту (орбита для приоритетных заявок), где осуществляют экспоненциально распределенную случайную задержку, после которой обращаются к прибору с повторной попыткой захвата. Если заявки приоритетного потока обнаруживают прибор занятым обслуживанием заявки альтернативного потока, то пришедшая заявка вытесняет обслуживаемую и сама встает на прибор. Вытесненная заявка переходит на орбиту для неприоритетных заявок. Если заявка неприоритетного потока обнаруживает прибор занятым, то она переходит на орбиту (орбита для неприоритетных заявок), где осуществляют случайную задержку. Дисциплина обращения заявок с орбит аналогична дисциплине обращения вновь прибывших в систему заявок. После успешного окончания обслуживания заявки покидают систему. Найдены распределения вероятностей числа заявок на неприоритетной и приоритетной орбитах. Число заявок на приоритетной орбите получено методом асимптотически-диффузионного анализа.

*Ключевые слова:* система массового обслуживания, RQ-система, орбита, асимптотически-диффузионный анализ, диффузионный процесс, распределение вероятностей.

DOI: 10.31857/S0005231025050013, EDN: AXYZTN

### 1. Введение

Образование очередей в офисах банка, в больницах, в торговых центрах и т.д. – это то, с чем современный человек сталкивается в повседневной жизни. Иногда, не желая стоять и ждать в очереди, люди уходят и повторяют свою попытку обслужиться позже, когда нагрузка уменьшится. Такое явление и породило специальные системы в теории массового обслуживания, называемые системами с повторными вызовами (RQ-системы). Эти системы

характеризуются тем, что клиент (заявка), обнаружив прибор занятым, переходит на виртуальную орбиту, где находится некоторое случайное время, после которого повторяет свою попытку обслужиться. Большая работа проделана рядом ученых, занимающихся вопросами исследования RQ-систем. Обзор по данной тематике можно найти в [1–5].

Однако на практике традиционные модели RQ-систем трудно широко применять в реальных условиях из-за их недостаточной гибкости. Почти каждая сфера жизни подвержена влиянию приоритетного обслуживания одного клиента над другим. В системах обслуживания, таких как авиалинии, клиенты первого класса имеют приоритет перед эконом-классом. В телекоммуникационной системе голосовые пакеты имеют более высокий приоритет, чем другие пакеты данных, такие как электронная почта и т.д. В [6] проанализированы различные RQ-системы с приоритетными вызовами и многими другими параметрами системы, такими как потери клиентов, обратная связь и т.д.

В последнее время мобильный трафик резко увеличился, что привело к дефициту беспроводных спектров. Когнитивные радиосети являются перспективными технологиями для решения проблемы нехватки спектра. В сетях когнитивного радио есть приоритетные и неприоритетные пользователи. Основные пользователи (приоритетные) предоставляют некоторые полосы спектра вторичным пользователям (неприоритетным). Вторичные пользователи могут когнитивно использовать эти полосы, когда они не используются основными, но когда основной пользователь прибыл для обслуживания, текущий вторичный пользователь должен покинуть прибор и повторить попытку занять канал позже. С математической точки зрения исследование RQ-систем с двумя типами клиентов существенно сложнее, чем с одним типом клиентов. Ряд работ посвящено исследованию систем с двумя потоками. В [7] исследована приоритетная система с нетерпеливыми заявками и неоднородным обслуживанием. В [8] авторы рассмотрели систему, в которой приоритетная заявка, обнаружив прибор занятым, либо вытесняет заявку на приборе, либо с некоторой вероятностью встает в очередь. В [8] авторами учитывается факт возможной поломки прибора. В [9–11] также найдены показатели функционирования RQ-систем различных конфигураций с двумя потоками (приоритетным и неприоритетным) и очередью для приоритетных.

Стоит отметить, что в описанных выше работах приоритетные заявки, обнаружив прибор занятым, встают в очередь. В данной работе предлагается система с повторными вызовами, в которой приоритетные клиенты, также как и неприоритетные, уходят на орбиту, т.е. надо рассмотреть систему с двумя орбитами. В [12] авторами рассмотрена тандемная RQ-система с двумя орбитами. Особенностью исследуемой в этой статье системы является тот факт, что приоритетные заявки могут вытеснять обслуживаемые неприоритетные заявки.

## 2. Математическая модель и постановка задачи

Рассмотрим систему с повторными вызовами (retrial queueing system, RQ-система), рис. 1.

Поступает два простейших потока с параметрами  $\lambda_1$  и  $\lambda_2$ . Первый поток – поток приоритетных заявок, второй – неприоритетных. Когда в момент прихода заявка обнаруживает прибор свободным, то она начинает обслуживаться в течение времени, распределенного по экспоненциальному закону с параметрами  $\mu_1$  и  $\mu_2$  соответственно для заявок приоритетного и неприоритетного потоков. После успешного окончания обслуживания заявка покидает систему. Когда в момент прихода заявка приоритетного потока обнаруживает прибор занятым, то:

– если на приборе обслуживалась приоритетная заявка, то пришедшая переходит на орбиту, где осуществляет случайную задержку, имеющую экспоненциальное распределение с параметром  $\sigma_1$ . После случайной задержки она вновь обращается к прибору с повторной попыткой его захвата;

– если на приборе обслуживалась неприоритетная заявка, то пришедшая заявка вытесняет обслуживаемую и сама начинает обслуживаться, а вытесненная переходит на орбиту для неприоритетных заявок, где осуществляет случайную задержку, экспоненциально распределенную с параметром  $\sigma_2$ , после нее обращается к прибору с повторной попыткой его захвата. Если он свободен, то начинается обслуживание, если занят, то мгновенно возвращается обратно.

Дисциплина обращения к прибору приоритетных заявок с орбиты такая же как и при обращении вновь прибывших в систему приоритетных заявок.

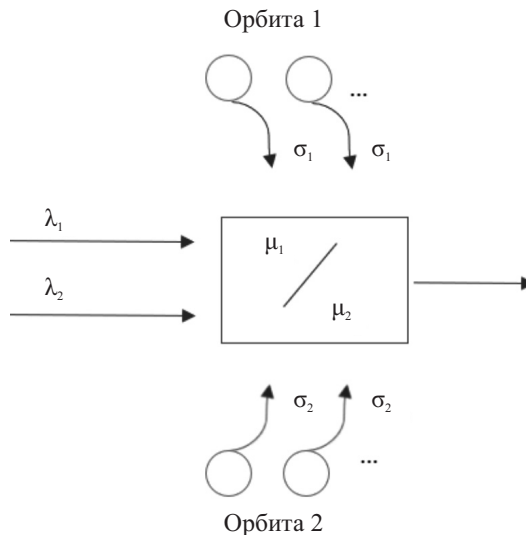


Рис. 1. Математическая модель RQ-системы  $M^{(2)}|M^{(2)}|1$ .

Обозначим:  $i_1(t)$  – число заявок на первой орбите,  $i_2(t)$  – число заявок на второй орбите. Состояние прибора определяется величиной  $k(t)$ :  $k(t) = 0$ , если прибор свободен для обслуживания;  $k(t) = 1$ , если прибор занят обслуживанием приоритетной заявки;  $k(t) = 2$ , если прибор занят обслуживанием непериприоритетной заявки.

Тогда вероятность того, что в момент времени  $t$  прибор находится в состоянии  $k$ , на первой орбите количество заявок равно  $i_1$ , на второй орбите количество заявок равно  $i_2$ , обозначим как

$$(2.1) \quad P\{k(t) = k, i_1(t) = i_1, i_2(t) = i_2\} = P_k(i_1, i_2, t).$$

Для распределения (2.1), используя формулу полной вероятности, составим систему дифференциальных уравнений Колмогорова:

$$(2.2) \quad \begin{aligned} \frac{\partial P_0(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + i_1\sigma_1 + i_2\sigma_2)P_0(i_1, i_2, t) + \\ &\quad + \mu_1 P_1(i_1, i_2, t) + \mu_2 P_2(i_1, i_2, t), \\ \frac{\partial P_1(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_1)P_1(i_1, i_2, t) + \lambda_1 P_1(i_1 - 1, i_2, t) + \\ &\quad + \lambda_2 P_1(i_1, i_2 - 1, t) + \lambda_1 P_0(i_1, i_2, t) + \\ &\quad + (i_1 + 1)\sigma_1 P_0(i_1 + 1, i_2, t) + \lambda_1 P_2(i_1, i_2 - 1, t) + \\ &\quad + (i_1 + 1)\sigma_1 P_2(i_1 + 1, i_2 - 1, t), \\ \frac{\partial P_2(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_2 + i_1\sigma_1)P_2(i_1, i_2, t) + \\ &\quad + \lambda_2 P_2(i_1, i_2 - 1, t) + \lambda_2 P_0(i_1, i_2, t) + \\ &\quad + (i_2 + 1)\sigma_2 P_0(i_1, i_2 + 1, t). \end{aligned}$$

Чтобы найти характеристики данной системы массового обслуживания, необходимо найти распределение вероятностей (2.1). В допредельном случае, т.е. решив систему уравнений (2.2), это сделать затруднительно, так как система является нетривиальной. Поэтому предлагается построить диффузионный процесс с помощью которого можно аппроксимировать распределение вероятностей (2.1), тем самым решив задачу.

*Определение 1. Частичной производяще-характеристической функцией будем называть функцию вида*

$$(2.3) \quad H_k(z, u, t) = \sum_{i_1=0}^{\infty} z^{i_1} \sum_{i_2=0}^{\infty} e^{ju_2 i_2} P_k(i_1, i_2, t).$$

Для дальнейших исследований перейдем от системы уравнений (2.2) к системе уравнений для функций (2.3). Получим:

$$\begin{aligned}
 \frac{\partial H_0(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2)H_0(z, u, t) - \sigma_1 z \frac{\partial H_0(z, u, t)}{\partial z} + \\
 &+ j\sigma_2 \frac{\partial H_0(z, u, t)}{\partial u} + \mu_1 H_1(z, u, t) + \mu_2 H_2(z, u, t), \\
 \frac{\partial H_1(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_1)H_1(z, u, t) + \lambda_1 z H_1(z, u, t) + \\
 (2.4) \quad &+ \lambda_2 e^{ju} H_1(z, u, t) + \lambda_1 H_0(z, u, t) + \lambda_1 e^{ju} H_2(z, u, t) + \\
 &+ \sigma_1 \frac{\partial H_0(z, u, t)}{\partial z} + \sigma_1 e^{ju} \frac{\partial H_2(z, u, t)}{\partial z}, \\
 \frac{\partial H_2(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_2)H_2(z, u, t) + \lambda_2 e^{ju} H_2(z, u, t) + \\
 &+ \lambda_2 H_0(z, u, t) - \sigma_1 z \frac{\partial H_2(z, u, t)}{\partial z} - j\sigma_2 e^{-ju} \frac{\partial H_0(z, u, t)}{\partial u}.
 \end{aligned}$$

Просуммировав уравнения системы (2.4), положив  $z = 1$  и обозначив  $H_0(1, u, t) + H_1(1, u, t) + H_2(1, u, t) = H(1, u, t)$ , получим еще добавочное уравнение:

$$\begin{aligned}
 \frac{\partial H(1, u, t)}{\partial t} &= (e^{ju} - 1) \times \\
 (2.5) \quad &\times \left( \lambda_2 H_1(1, u, t) + (\lambda_1 + \lambda_2) H_2(1, u, t) + \right. \\
 &\left. + j\sigma_2 e^{-ju} \frac{\partial H_0(1, u, t)}{\partial u} + \sigma_1 \frac{\partial H_2(1, u, t)}{\partial z} \right),
 \end{aligned}$$

которое вместе с системой (2.4) будем решать методом асимптотически-диффузионного анализа.

Диффузионный анализ будем проводить по неприоритетной компоненте в предельном условии большой задержки заявок на второй орбите, т.е. при  $\sigma_2 \rightarrow 0$ , в несколько этапов:

1) выполнив асимптотику первого порядка, получим коэффициент переноса некоторого диффузионного процесса, с помощью которого проведем аппроксимацию распределения вероятностей числа заявок на орбите. Также на этом этапе находим выражения для стационарного распределения вероятностей состояний прибора и для частичной производящей функции числа заявок на первой орбите;

2) выполнив асимптотику второго порядка, получим коэффициент диффузии некоторого диффузионного процесса;

3) на третьем этапе построим диффузионный процесс и получим аппроксимацию распределения вероятностей числа заявок на неприоритетной орбите.

### 3. Асимптотически-диффузионный анализ по неприоритетной компоненте

#### 3.1. Асимптотика первого порядка

В системе (2.4) и уравнении (2.5), обозначив  $\sigma_2 = \epsilon$ , сделаем следующие замены:

$$\sigma_2 t = \tau, \quad u = \sigma_2 \epsilon = \epsilon w, \quad H_k(z, u, t) = F_k(z, w, \tau, \epsilon).$$

Тогда (2.4) и (2.5) с учетом замен переписываются в виде:

$$(3.1) \quad \begin{aligned} \epsilon \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2)F_0(z, w, \tau, \epsilon) - \sigma_1 z \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial z} + \\ &+ j \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial w} + \mu_1 F_1(z, w, \tau, \epsilon) + \mu_2 F_2(z, w, \tau, \epsilon), \\ \epsilon \frac{\partial F_1(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2 + \mu_1)F_1(z, w, \tau, \epsilon) + \lambda_1 z F_1(z, w, \tau, \epsilon) + \\ &+ \lambda_2 e^{j\epsilon w} F_1(z, w, \tau, \epsilon) + \lambda_1 F_0(z, w, \tau, \epsilon) + \\ &+ \lambda_1 e^{j\epsilon w} F_2(z, w, \tau, \epsilon) + \sigma_1 \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial z} + \\ &+ \sigma_1 e^{j\epsilon w} \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial z}, \end{aligned}$$

$$\begin{aligned} \epsilon \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2 + \mu_2)F_2(z, w, \tau, \epsilon) + \lambda_2 e^{j\epsilon w} F_2(z, w, \tau, \epsilon) + \\ &+ \lambda_2 F_0(z, w, \tau, \epsilon) - \sigma_1 z \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial z} - \\ &- j e^{-j\epsilon w} \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial w}. \end{aligned}$$

$$(3.2) \quad \begin{aligned} \epsilon \frac{\partial F(1, w, \tau, \epsilon)}{\partial \tau} &= (e^{j\epsilon w} - 1) \left( \lambda_2 F_1(1, w, \tau, \epsilon) + (\lambda_1 + \lambda_2) F_2(1, w, \tau, \epsilon) + \right. \\ &\left. + j e^{-j\epsilon w} \frac{\partial F_0(1, w, \tau, \epsilon)}{\partial w} + \sigma_1 \frac{\partial F_2(1, w, \tau, \epsilon)}{\partial z} \right), \end{aligned}$$

где

$$F(1, w, \tau, \epsilon) = F_0(1, w, \tau, \epsilon) + F_1(1, w, \tau, \epsilon) + F_2(1, w, \tau, \epsilon).$$

Сформулируем следующую теорему.

*Теорема 1. Обозначим решение системы уравнений (3.1):*

$$\lim_{\epsilon \rightarrow 0} F_k(z, w, \tau, \epsilon) = F_k(z, w, \tau), \quad k = \overline{0, 2}.$$

*Тогда справедливо утверждение*

$$(3.3) \quad F_k(z, w, \tau) = G_k(z, x(\tau)) e^{jw x(\tau)}, \quad k = \overline{0, 2}.$$

Для удобства записи аргумент  $\tau$  будем опускать:  $x(\tau) = x$ . Функции  $G_k(z, x)$ ,  $k = \overline{0, 2}$  – частичные производящие функции числа заявок на первой орбите, которые имеют вид

$$\begin{aligned}
 G_0(z, x) &= \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 z)^{\frac{\lambda_1}{\sigma_1}}} - \\
 &\quad - \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy, \\
 (3.4) \quad G_1(z) &= \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} \frac{\lambda_1}{\mu_1}, \\
 G_2(z, x) &= \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy,
 \end{aligned}$$

величина  $x(\tau)$  является решением дифференциального уравнения

$$(3.5) \quad x'(\tau) = -x(\tau)G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2)G_2(1, x) + \sigma_1 \left. \frac{\partial G_2(z, x)}{\partial z} \right|_{z=1}.$$

*Доказательство.*

В системе (3.1) выполним предельный переход при  $\epsilon \rightarrow 0$ , получим

$$\begin{aligned}
 &-(\lambda_1 + \lambda_2)F_0(z, w, \tau) - \sigma_1 z \frac{\partial F_0(z, w, \tau)}{\partial z} + j \frac{\partial F_0(z, w, \tau)}{\partial w} + \\
 &\quad + \mu_1 F_1(z, w, \tau) + \mu_2 F_2(z, w, \tau) = 0, \\
 (3.6) \quad &-(\lambda_1 + \mu_1)F_1(z, w, \tau) + \lambda_1 z F_1(z, w, \tau) + \\
 &+ \lambda_1 F_0(z, w, \tau) + \lambda_1 F_2(z, w, \tau) + \sigma_1 \frac{\partial F_0(z, w, \tau)}{\partial z} + \sigma_1 \frac{\partial F_2(z, w, \tau)}{\partial z} = 0, \\
 &-(\lambda_1 + \mu_2)F_2(z, w, \tau) + \lambda_2 F_0(z, w, \tau) - \sigma_1 z \frac{\partial F_2(z, w, \tau)}{\partial z} - j \frac{\partial F_0(z, w, \tau)}{\partial w} = 0.
 \end{aligned}$$

Решение системы уравнений (3.6) будем искать в виде (3.3). Тогда (3.6) перепишется как

$$\begin{aligned}
 &-(\lambda_1 + \lambda_2 + x(\tau))G_0(z, x) - \sigma_1 z \frac{\partial G_0(z, x)}{\partial z} + \mu_1 G_1(z) + \mu_2 G_2(z, x) = 0, \\
 &\quad -(\lambda_1(1 - z) + \mu_1)G_1(z) + \lambda_1 G_0(z, x) + \\
 (3.7) \quad &+ \lambda_1 G_2(z, x) + \sigma_1 \frac{\partial G_0(z, x)}{\partial z} + \sigma_1 \frac{\partial G_2(z, x)}{\partial z} = 0, \\
 &-(\lambda_1 + \mu_2)G_2(z, x) + (\lambda_2 + x(\tau))G_0(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} = 0.
 \end{aligned}$$

Введем обозначение

$$G_{02}(z, x) = G_0(z, x) + G_2(z, x).$$

Тогда продифференцировав  $G_{02}(z, x)$  по  $z$ , получаем

$$\frac{\partial G_{02}(z, x)}{\partial z} = \frac{\partial G_0(z, x)}{\partial z} + \frac{\partial G_2(z, x)}{\partial z}.$$

Просуммируем первое и третье уравнение (3.7) и составим систему уже из двух уравнений (суммы первого и третьего уравнений и второго уравнения (3.7)). Учитывая введенные обозначения, получим

$$(3.8) \quad \begin{aligned} -\lambda_1 G_{02}(z, x) + \mu_1 G_1(z) - \sigma_1 z \frac{\partial G_{02}(z, x)}{\partial z} &= 0, \\ -(\lambda_1(1-z) + \mu_1) G_1(z) + \lambda_1 G_{02}(z, x) + \sigma_1 \frac{\partial G_{02}(z, x)}{\partial z} &= 0. \end{aligned}$$

Домножив второе уравнение на  $z$  и просуммировав уравнения системы, получаем

$$(3.9) \quad \lambda_1 G_{02}(z, x) - \mu_1 G_1(z) + \lambda_1 z G_1(z) = 0.$$

В (3.9) выразим  $G_1(z)$  через  $G_{02}(z, x)$  и подставим в первое уравнение системы (3.8), получим однородное дифференциальное уравнение относительно функции  $G_{02}(z, x) = G_{02}(z)$ :

$$\frac{\lambda_1^2 z}{\mu_1 - \lambda_1 z} G_{02}(z) - \sigma_1 z \frac{\partial G_{02}(z)}{\partial z} = 0,$$

решение которого имеет вид

$$(3.10) \quad G_{02}(z) = (\mu_1 - \lambda_1 z)^{-\frac{\lambda_1}{\sigma_1}} C.$$

Для функции  $G_1(z)$  аналогично получаем дифференциальное уравнение

$$(\lambda_1 + \sigma_1) G_1(z) + \left( \sigma_1 z - \frac{\sigma_1 \mu_1}{\lambda_1} \right) \frac{\partial G_1(z)}{\partial z} = 0,$$

решение которого имеет вид

$$(3.11) \quad G_1(z) = (\mu_1 - \lambda_1 z)^{-\frac{\lambda_1 + \sigma_1}{\sigma_1}} C.$$

Найдем константу  $C$ . В функциях  $G_k(z, x)$ ,  $k = \overline{0, 2}$ , положим  $z = 1$  и обозначим  $G_0(1, x) = R_0(x)$ ,  $G_1(1) = R_1$ ,  $G_2(1, x) = R_2(x)$ . Величины  $R_0(x)$ ,  $R_1$ ,  $R_2(x)$  удовлетворяют условию нормировки  $R_0(x) + R_1 + R_2(x) = 1$ . Если в уравнении (3.5) перейти к стационарному режиму и решение стационарного

уравнения обозначить  $x = \kappa$ , то получаем стационарное распределение вероятностей состояний прибора  $R_0(\kappa), R_1, R_2(\kappa)$ .

В уравнении (3.9) положим  $z = 1$  и, добавляя условие нормировки, получим систему уравнений

$$\begin{aligned}\lambda_1(R_0(x) + R_2(x)) - \mu_1 R_1 + \lambda_1 R_1 &= 0, \\ R_0(x) + R_1 + R_2(x) &= 1.\end{aligned}$$

Откуда получаем

$$R_1 = \frac{\lambda_1}{\mu_1}, \quad R_0(x) + R_2(x) = \frac{\mu_1 - \lambda_1}{\mu_1}.$$

Тогда, полагая  $z = 1$  в (3.10) и (3.11) и используя найденные  $R_k(x)$ , получаем

$$G_1(z) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} \frac{\lambda_1}{\mu_1}, \quad G_{02}(z) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1}.$$

Рассмотрим третье уравнение системы (3.7) и найденное  $G_{02}(z) = G_{02}(z, x) = G_0(z, x) + G_2(z, x)$ :

$$\begin{aligned}(3.12) \quad & -(\lambda_1 + \mu_2)G_2(z, x) + (\lambda_2 + x(\tau))G_0(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} = 0, \\ & G_0(z, x) + G_2(z, x) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1}.\end{aligned}$$

Выразив из второго уравнения  $G_0(z, x)$  и подставив в первое уравнение системы (3.12), получим неоднородное дифференциальное уравнение относительно функции  $G_2(z, x)$ :

$$(G_2(z, x))'_z + \frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1 z} G_2(z, x) = \frac{\lambda_2 + x}{\sigma_1 z} \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1},$$

решение которого имеет вид

$$G_2(z, x) = \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy.$$

Подставив найденное выражение для функции  $G_2(z, x)$  в систему (3.12), получим решение для функции  $G_0(z, x)$ :

$$\begin{aligned}G_0(z, x) &= \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 z)^{\frac{\lambda_1}{\sigma_1}}} - \\ &- \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy.\end{aligned}$$

Найденные выражения для функций  $G_k(z, x)$ ,  $k = 0, 1, 2$  совпадают с (3.4).

Рассмотрим уравнение (3.2). Разложим экспоненту в ряд Тейлора

$$e^{j\epsilon w} = 1 + j\epsilon w + O(\epsilon^2)$$

и разделим левую и правую части (3.2) на  $j\epsilon w$ . Далее выполним предельный переход при  $\epsilon \rightarrow 0$ , получим

$$\begin{aligned} \frac{\partial F(1, w, \tau)}{jw\partial\tau} &= \lambda_2 F_1(1, w, \tau) + (\lambda_1 + \lambda_2) F_2(1, w, \tau) + \\ &+ j \frac{\partial F_0(1, w, \tau)}{\partial w} + \sigma_1 \frac{\partial F_2(1, w, \tau)}{\partial z}. \end{aligned}$$

Выполним замены

$$\begin{aligned} F_k(1, w, \tau) &= G_k(1, x) e^{jwx} = R_k(x) e^{jwx}, \quad k = 0, 1, 2, \\ F(1, w, \tau) &= G(1, x) e^{jwx} = e^{jwx}. \end{aligned}$$

Тогда получим следующее дифференциальное уравнение для  $x(\tau)$ :

$$x'(\tau) = -x(\tau)G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2)G_2(1, x) + \sigma_1 \left. \frac{\partial G_2(z, x)}{\partial z} \right|_{z=1},$$

которое совпадает с (3.5). Таким образом, теорема 1 доказана.

Обозначим:

$$\begin{aligned} \left. \frac{\partial G_k(z, x)}{\partial z} \right|_{z=1} &= \frac{\partial G_k(1, x)}{\partial z}, \quad k = 0, 1, 2, \\ \left. \frac{\partial G_{02}(z)}{\partial z} \right|_{z=1} &= \frac{\partial G_{02}(1)}{\partial z}. \end{aligned}$$

Обозначим правую часть уравнения (3.5) через  $a(x)$  и упростим ее, используя (3.4); получим

$$\begin{aligned} a(x) &= x'(\tau) = \\ &= -x(\tau)G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2)G_2(1, x) + \sigma_1 \frac{\partial G_2(1, x)}{\partial z} = \\ &= \lambda_2 - \mu_2 R_2(x). \end{aligned}$$

Величина  $a(x)$  имеет смысл коэффициента переноса некоторого диффузионного процесса, с помощью которого получим аппроксимацию распределения вероятностей числа заявок на орбите.

*Следствие 1. Условие эргодичности рассматриваемой RQ-системы имеет вид*

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1.$$

*Доказательство.*

Достаточным условием эргодичности системы будет неравенство

$$\lim_{x \rightarrow \infty} a(x) < 0.$$

Таким образом, получаем

$$\lim_{x \rightarrow \infty} a(x) = \lim_{x \rightarrow \infty} (\lambda_2 - \mu_2 R_2(x)) = \lambda_2 - \mu_2 \left( \frac{\mu_1 - \lambda_1}{\mu_1} \right) < 0.$$

Откуда получаем условие

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1.$$

*Следствие 2. Производящая функция распределения вероятностей числа заявок на приоритетной орбите имеет вид*

$$G(z) = \frac{\lambda_1}{\mu_1} \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} + \frac{\mu_1 - \lambda_1}{\mu_1} \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}}.$$

Таким образом, распределение вероятностей числа заявок на первой орбите имеет вид взвешенной суммы двух отрицательно-биномиальных распределений с весами  $\frac{\lambda_1}{\mu_1}$  и  $\frac{\mu_1 - \lambda_1}{\mu_1}$ .

### 3.2. Асимптотика второго порядка

Рассмотрим характеристическую функцию случайного процесса  $i(t) - \frac{x(\sigma_2 t)}{\sigma_2}$ . Для этого в системе (2.4) и уравнении (2.5) сделаем замену:

$$H_k(z, u, t) = e^{\frac{jux(\sigma_2 t)}{\sigma_2}} H_k^{(2)}(z, u, t), \quad k = 0, 1, 2.$$

Обозначая  $\sigma_2 = \epsilon^2$ , в системе уравнений для  $H_k^{(2)}(z, u, t)$ ,  $k = 0, 1, 2$  выполним замены:

$$\tau = \epsilon^2 t, \quad u = \epsilon w, \quad H_k^{(2)}(z, u, t) = F_k^{(2)}(z, w, \tau, \epsilon),$$

получаем систему для функций  $F_k^{(2)}(z, w, \tau, \epsilon)$ ,  $k = 0, 1, 2$  и добавочное уравнение:

$$\begin{aligned}
 & \epsilon^2 \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_0^{(2)}(z, w, \tau, \epsilon) = \\
 & = -(\lambda_1 + \lambda_2 + x) F_0^{(2)}(z, w, \tau, \epsilon) - \sigma_1 z \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial z} + \\
 & + j\epsilon \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial w} + \mu_1 F_1^{(2)}(z, w, \tau, \epsilon) + \mu_2 F_2^{(2)}(z, w, \tau, \epsilon), \\
 & \epsilon^2 \frac{\partial F_1^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_1^{(2)}(z, w, \tau, \epsilon) = \\
 & = -(\lambda_1 + \lambda_2 + \mu_1) F_1^{(2)}(z, w, \tau, \epsilon) + \lambda_1 z F_1^{(2)}(z, w, \tau, \epsilon) + \\
 (3.13) \quad & + \lambda_2 e^{j\epsilon w} F_1^{(2)}(z, w, \tau, \epsilon) + \lambda_1 F_0^{(2)}(z, w, \tau, \epsilon) + \lambda_1 e^{j\epsilon w} F_2^{(2)}(z, w, \tau, \epsilon) + \\
 & + \sigma_1 \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial z} + \sigma_1 e^{j\epsilon w} \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial z},
 \end{aligned}$$

$$\begin{aligned}
 & \epsilon^2 \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_2^{(2)}(z, w, \tau, \epsilon) = \\
 & = -(\lambda_1 + \lambda_2 + \mu_2) F_2^{(2)}(z, w, \tau, \epsilon) + \lambda_2 e^{j\epsilon w} F_2^{(2)}(z, w, \tau, \epsilon) + \\
 & + \lambda_2 F_0^{(2)}(z, w, \tau, \epsilon) - \sigma_1 z \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial z} - \\
 & - j\epsilon e^{-j\epsilon w} \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial w} + x e^{-j\epsilon w} F_0^{(2)}(z, w, \tau, \epsilon).
 \end{aligned}$$

$$\begin{aligned}
 & \epsilon^2 \frac{\partial F^{(2)}(1, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F^{(2)}(1, w, \tau, \epsilon) = (e^{j\epsilon w} - 1) \times \\
 (3.14) \quad & \times \left( -x e^{-j\epsilon w} F_0^{(2)}(1, w, \tau, \epsilon) + \lambda_2 F_1^{(2)}(1, w, \tau, \epsilon) + \right. \\
 & \left. + (\lambda_1 + \lambda_2) F_2^{(2)}(1, w, \tau, \epsilon) + j\epsilon e^{-j\epsilon w} \frac{\partial F_0^{(2)}(1, w, \tau, \epsilon)}{\partial w} + \sigma_1 \frac{\partial F_2^{(2)}(1, w, \tau, \epsilon)}{\partial z} \right).
 \end{aligned}$$

Обозначим:  $F_k^{(2)}(z, w, \tau) = \lim_{\epsilon \rightarrow 0} F_k^{(2)}(z, w, \tau, \epsilon)$ ,  $k = 0, 1, 2$ . Сформулируем и докажем следующую теорему.

*Теорема 2. Функции  $F_k^{(2)}(z, w, \tau)$ ,  $k = 0, 1, 2$  имеют вид  $F_k^{(2)}(z, w, \tau) = \Phi(w, \tau) G_k(z, x)$ , где функция  $\Phi(w, \tau)$  является характеристической функцией процесса  $y(\tau) = \lim_{\sigma_2 \rightarrow 0} \sqrt{\sigma_2} \left( i \left( \frac{\tau}{\sigma_2} \right) - \frac{x(\tau)}{\sigma_2} \right)$ , и удовлетворяет дифференциальному уравнению*

$$(3.15) \quad \frac{\partial \Phi(w, \tau)}{\partial \tau} = a'(x) w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2} b(x) \Phi(w, \tau),$$

зде

$$(3.16) \quad a(x) = \lambda_2 - \mu_2 R_2(x),$$

$$(3.17) \quad b(x) = a(x) + 2((\lambda_2 - a(x))R_2(x) - \mu_2 h_2(x)).$$

Здесь  $h_2(x) = h_2(1, x)$ ,

$$(3.18) \quad h_2(1, x) = \frac{1}{\sigma_1} \int_0^1 y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \times D(y, x) dy,$$

$$(3.19) \quad D(z, x) = (\lambda_2 + x)h_{02}(z, x) + (\lambda_2 - a(x))G_2(z, x) - xG_0(z, x),$$

$$(3.20) \quad h_{02}(z, x) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - z\lambda_1} \right)^{\frac{\lambda_1}{\sigma_1}} \left( \frac{1}{\mu_1} A(1, x) - \frac{1}{\sigma_1} \int_z^1 \left( \frac{\mu_1 - \lambda_1}{\mu_1 - y\lambda_1} \right)^{\frac{-\lambda_1}{\sigma_1}} B(y, x) dy \right),$$

$$(3.21) \quad A(1, x) = a(x) \frac{\partial G_{02}(1)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(1, x)}{\partial z} + x \frac{\partial G_0(1, x)}{\partial z} - \\ - \lambda_1 R_2(x) + (a(x) - \lambda_2) R_1 + (a(x) - \lambda_2) \frac{\partial G_1(1)}{\partial z} - \sigma_1 \frac{\partial^2 G_2(1, x)}{\partial z^2},$$

$$(3.22) \quad B(z, x) = (a(x) - \lambda_2) G_1(z) - \lambda_1 G_2(z, x) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \\ - \frac{(z-1)\lambda_1 - \mu_1}{(z\lambda_1 - \mu_1)(z-1)} \left( a(x) G_{02}(z) - (\lambda_2 + z\lambda_1) G_2(z, x) + \right. \\ \left. + x G_0(z, x) + z(a(x) - \lambda_2) G_1(z) - z\sigma_1 \frac{\partial G_2(z, x)}{\partial z} \right).$$

*Доказательство.*

Решение системы уравнений (3.13) запишем в следующем виде:

$$F_k(z, w, \tau, \epsilon) = \Phi(w, \tau) (G_k(z, x) + j\epsilon w g_k(z, x)) + O(\epsilon^2), \quad k = 0, 1, 2.$$

Подставляя данные функции в систему уравнений (3.13), сделав несложные преобразования и выполнив предельный переход при  $\epsilon \rightarrow 0$ , получим систему уравнений:

$$(3.23) \quad a(x)G_0(z, x) = -(\lambda_1 + \lambda_2 + x)g_0(z, x) + \mu_1 g_1(z, x) + \mu_2 g_2(z, x) - \\ - \sigma_1 z \frac{\partial g_0(z, x)}{\partial z} + G_0(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \\ a(x)G_1(z) = -(\lambda_1(1-z) + \mu_1)g_1(z, x) + \lambda_1 g_0(z, x) + \lambda_1 g_2(z, x) + \\ + \lambda_2 G_1(z, x) + \lambda_1 G_2(z, x) + \sigma_1 \frac{\partial g_0(z, x)}{\partial z} + \sigma_1 \frac{\partial g_2(z, x)}{\partial z} + \sigma_1 \frac{\partial G_2(z, x)}{\partial z}, \\ a(x)G_2(z, x) = -(\lambda_1 + \mu_2)g_2(z, x) + (\lambda_2 + x)g_0(z, x) + \lambda_2 G_2(z, x) - \\ - x G_0(z, x) - \sigma_1 z \frac{\partial g_2(z, x)}{\partial z} - G_0(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}.$$

Функции  $g_k(z, x)$ ,  $k = 0, 1, 2$  будем искать в виде суммы общего решения однородного и частного решения неоднородного дифференциального уравнения:

$$g_k(z, x) = CG_k(z, x) + h_k(z, x) - \phi_k(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \quad k = 0, 1, 2.$$

Подставляя  $g_k(z, x)$  в (3.23), можно убедиться, что коэффициент перед константой  $C$  равен нулю. Тогда, приравнявая коэффициенты при  $\frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}$ , получаем систему уравнений для нахождения функций  $\phi_k(z, x)$ . Если систему уравнений для функций  $G_k(z, x)$  продифференцировать по переменной  $x$ , то можно убедиться, что она совпадет с системой уравнений для нахождения функций  $\phi_k(z, x)$ . Таким образом, делаем вывод, что

$$\phi_k(z, x) = \frac{\partial G_k(z, x)}{\partial x}, \quad k = 0, 1, 2.$$

Приравнявая оставшиеся слагаемые, можно записать систему уравнений для нахождения функций  $h_k(z, x)$ ,  $k = 0, 1, 2$ :

$$\begin{aligned} & -(\lambda_1 + \lambda_2 + x)h_0(z, x) + \mu_1 h_1(z, x) + \mu_2 h_2(z, x) - \sigma_1 z \frac{\partial h_0(z, x)}{\partial z} = \\ & \quad = a(x)G_0(z, x), \\ (3.24) \quad & \lambda_1 h_0(z, x) - (\lambda_1(1 - z) + \mu_1)h_1(z, x) + \lambda_1 h_2(z, x) + \sigma_1 \frac{\partial h_0(z, x)}{\partial z} + \\ & \quad + \sigma_1 \frac{\partial h_2(z, x)}{\partial z} = (a(x) - \lambda_2)G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x), \\ & \quad (\lambda_2 + x)h_0(z, x) - (\lambda_1 + \mu_2)h_2(z, x) - \sigma_1 z \frac{\partial h_2(z, x)}{\partial z} + \\ & \quad = (a(x) - \lambda_2)G_2(z, x) + xG_0(z, x). \end{aligned}$$

Просуммируем первое и третье уравнение системы (3.24) и добавим к полученному уравнению второе, получим систему:

$$\begin{aligned} & -\lambda_1(h_0(z, x) + h_2(z, x)) + \mu_1 h_1(z, x) - \sigma_1 z \left( \frac{\partial h_0(z, x)}{\partial z} + \frac{\partial h_2(z, x)}{\partial z} \right) = \\ & \quad = a(x)(G_0(z, x) + G_2(z, x)) - \lambda_2 G_2(z, x) + xG_0(z, x), \\ & \lambda_1(h_0(z, x) + h_2(z, x)) - (\lambda_1(1 - z) + \mu_1)h_1(z, x) + \sigma_1 \left( \frac{\partial h_0(z, x)}{\partial z} + \frac{\partial h_2(z, x)}{\partial z} \right) = \\ & \quad = (a(x) - \lambda_2)G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x). \end{aligned}$$

Обозначим:

$$h_{02}(z, x) = h_0(z, x) + h_2(z, x).$$

Тогда система уравнений переписывается в виде:

$$\begin{aligned}
& -\lambda_1 h_{02}(z, x) + \mu_1 h_1(z, x) - \sigma_1 z \frac{\partial h_{02}(z, x)}{\partial z} = \\
& = a(x)(G_0(z, x) + G_2(z, x)) - \lambda_2 G_2(z, x) + x G_0(z, x), \\
(3.25) \quad & \lambda_1 h_{02}(z, x) - (\lambda_1(1-z) + \mu_1) h_1(z, x) + \sigma_1 \frac{\partial h_{02}(z, x)}{\partial z} = \\
& = (a(x) - \lambda_2) G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x).
\end{aligned}$$

Домножим второе уравнение на  $z$  и просуммируем уравнения системы, получим:

$$\begin{aligned}
(3.26) \quad & \lambda_1 h_{02}(z, x) - (\mu_1 - \lambda_1 z) h_1(z, x) = \\
& = \frac{1}{z-1} \left( a(x)(G_0(z, x) + G_2(z, x) + z G_1(z)) + \right. \\
& \left. + x G_0(z, x) - \lambda_2 z G_1(z) - (\lambda_1 z + \lambda_2) G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} \right).
\end{aligned}$$

Обозначим правую часть уравнения через  $A(z, x)$  и найдем предел при  $z \rightarrow 1$ :

$$\begin{aligned}
& \lim_{z \rightarrow 1} \frac{(a(x) + x) G_0(z, x) + z(a(x) - \lambda_2) G_1(z) - (\lambda_1 z + \lambda_2 - a(x)) G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z}}{z-1} = \\
& = \lim_{z \rightarrow 1} \left( a(x) \frac{\partial G_{02}(z)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(z, x)}{\partial z} + x \frac{\partial G_0(z, x)}{\partial z} - \lambda_1 G_2(z, x) + \right. \\
& \quad \left. + (a(x) - \lambda_2) G_1(z) + (a(x) - \lambda_2) \frac{\partial G_1(z)}{\partial z} - \sigma_1 z \frac{\partial^2 G_2(z, x)}{\partial z^2} \right) = \\
& = \lim_{z \rightarrow 1} A(z, x) = a(x) \frac{\partial G_{02}(1)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(1, x)}{\partial z} + x \frac{\partial G_0(1, x)}{\partial z} - \\
& - \lambda_1 R_2(x) + (a(x) - \lambda_2) R_1 + (a(x) - \lambda_2) \frac{\partial G_1(1)}{\partial z} - \sigma_1 \frac{\partial^2 G_2(1, x)}{\partial z^2} = A(1, x),
\end{aligned}$$

что совпадает с (3.21).

Тогда уравнение (3.26) можно записать в виде

$$\lambda_1 h_{02}(z, x) - (\mu_1 - \lambda_1 z) h_1(z, x) = A(1, x).$$

Далее выражаем функцию  $h_1(z, x)$  и подставляем во второе уравнение системы (3.25), получаем неоднородное дифференциальное уравнение относительно функции  $h_{02}(z, x)$ . Обозначим правую часть уравнения через  $B(z, x)$ :

$$\begin{aligned}
B(z, x) = & (a(x) - \lambda_2) G_1(z) - \lambda_1 G_2(z, x) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \frac{(z-1)\lambda_1 - \mu_1}{(z\lambda_1 - \mu_1)(z-1)} \times \\
& \times \left( a(x) G_{02}(z) - (\lambda_2 + z\lambda_1) G_2(z, x) + x G_0(z, x) + \right. \\
& \left. + z(a(x) - \lambda_2) G_1(z) - z\sigma_1 \frac{\partial G_2(z, x)}{\partial z} \right).
\end{aligned}$$

Тогда дифференциальное уравнение запишется в следующем виде:

$$\lambda_1 \left( 1 - \frac{\lambda_1(1-z) + \mu_1}{\mu_1 - \lambda_1 z} \right) h_{02}(z, x) + \sigma_1 \frac{\partial h_{02}(z, x)}{\partial z} = B(z, x).$$

Его решение имеет вид

$$h_{02}(z, x) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - z\lambda_1} \right)^{\frac{\lambda_1}{\sigma_1}} \left( \frac{1}{\mu_1} A(1, x) - \frac{1}{\sigma_1} \int_z^1 \left( \frac{\mu_1 - \lambda_1}{\mu_1 - y\lambda_1} \right)^{\frac{-\lambda_1}{\sigma_1}} B(y, x) dy \right),$$

что совпадает с (3.20).

Так как  $h_0(z, x) = h_{02}(z, x) - h_2(z, x)$ , то подставим это в третье уравнение системы (3.24) и обозначим правую часть как

$$D(z, x) = (\lambda_2 + x)h_{02}(z, x) + (\lambda_2 - a(x))G_2(z, x) - xG_0(z, x),$$

получим дифференциальное уравнение относительно  $h_2(z, x)$ :

$$(\lambda_1 + \lambda_2 + \mu_2 + x)h_2(z, x) + \sigma_1 z \frac{\partial h_2(z, x)}{\partial z} = D(z, x),$$

решение которого имеет вид

$$h_2(z, x) = z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \frac{1}{\sigma_1} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} D(y, x) dy.$$

Полагая здесь  $z = 1$ , получаем (3.18).

Рассмотрим уравнение (3.14). Подставим в него

$$F_k(z, w, \tau, \epsilon) = \Phi(w, \tau) (G_k(z, x) + j\epsilon w g_k(z, x)) + O(\epsilon^2), \quad k = 0, 1, 2,$$

сделав несложные преобразования и выполнив предельный переход при  $\epsilon \rightarrow 0$ , получим

$$\begin{aligned} & -\frac{\partial \Phi(w, \tau)}{w^2 \Phi(w, \tau) \partial \tau} + a(x)g(1, x) = \lambda_2 g_1(1, x) + (\lambda_1 + \lambda_2)g_2(1, x) + \\ & + G_0(1, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w} - xg_0(1, x) + xG_0(1, x) + \sigma_1 \frac{\partial g_2(1, x)}{\partial z} + \frac{1}{2}a(x). \end{aligned}$$

Функции  $g_k(1, x)$  запишем в виде

$$g_k(1, x) = C \times G_k(1, x) + h_k(1, x) - \phi_k(1, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \quad k = 0, 1, 2.$$

С учетом условий  $\sum_{k=0}^2 h_k(z, x)\Big|_{z=1} = 0$ ,  $\sum_{k=0}^2 \phi_k(z, x)\Big|_{z=1} = 0$  получим уравнение

$$\begin{aligned} \frac{\partial \Phi(w, \tau)}{\partial \tau} = & \left( \lambda_2 \phi_1(1, x) + (\lambda_1 + \lambda_2) \phi_2(1, x) - \right. \\ & \left. - G_0(1, x) - x \phi_0(1, x) + \sigma_1 \frac{\partial \phi_2(1, x)}{\partial z} \right) \times \\ & \times w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2} \Phi(w, \tau) \left( a(x) + 2 \left( \lambda_2 h_1(1, x) + (\lambda_1 + \lambda_2) h_2(1, x) - \right. \right. \\ & \left. \left. - x h_0(1, x) + x G_0(1, x) + \sigma_1 \frac{\partial h_2(1, x)}{\partial z} \right) \right). \end{aligned}$$

Преобразуем коэффициенты перед  $w \frac{\partial \Phi(w, \tau)}{\partial w}$  и  $\frac{w^2}{2} \Phi(w, \tau)$ , получаем

$$\begin{aligned} \lambda_2 \phi_1(1, x) + (\lambda_1 + \lambda_2) \phi_2(1, x) - G_0(1, x) - x \phi_0(1, x) + \sigma_1 \frac{\partial \phi_2(1, x)}{\partial z} &= a'(x), \\ a(x) + 2 \left( \lambda_2 h_1(1, x) + (\lambda_1 + \lambda_2) h_2(1, x) - x h_0(1, x) + x G_0(1, x) + \sigma_1 \frac{\partial h_2(1, x)}{\partial z} \right) &= \\ = a(x) + 2((\lambda_2 - a(x))R_2(x) - \mu_2 h_2(1, x)) &= b(x). \end{aligned}$$

Тогда получаем уравнение

$$\frac{\partial \Phi(w, \tau)}{\partial \tau} = a'(x) w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2} b(x) \Phi(w, \tau).$$

Теорема доказана.

Функция  $b(x)$  имеет смысл коэффициента диффузии некоторого диффузионного процесса, с помощью которого проведем аппроксимацию распределения вероятностей числа заявок на орбите.

### 3.3. Диффузионная аппроксимация распределения вероятностей числа заявок на неприоритетной орбите

К уравнению (3.15) применим обратное преобразование Фурье. Получим уравнение Фоккера–Планка для функции плотности распределения вероятностей  $P(y, \tau) = \frac{\partial P\{y(\tau) < y\}}{\partial y}$ :

$$\frac{\partial P(y, \tau)}{\partial \tau} = - \frac{\partial \{y a'(x) P(y, \tau)\}}{\partial y} + \frac{1}{2} \frac{\partial^2 \{b(x) P(y, \tau)\}}{\partial y^2}.$$

Можно сделать вывод, что  $y(\tau) = \lim_{\sigma_2 \rightarrow 0} \sqrt{\sigma_2} \left( i \left( \frac{\tau}{\sigma_2} \right) - \frac{x(\tau)}{\sigma_2} \right)$  является решением стохастического дифференциального уравнения

$$dy(\tau) = a'(x) y d\tau + \sqrt{b(x)} dW(\tau),$$

где  $W(\tau)$  является винеровским случайным процессом,  $a'(x)y$  – коэффициент переноса,  $\sqrt{b(x)}$  – коэффициент диффузии.

Рассмотрим диффузионный процесс  $z(\tau) = x(\tau) + \sqrt{\sigma_2}y(\tau)$ . Заметим, что  $z(\tau) = \lim_{\sigma_2 \rightarrow 0} \sigma_2 i \left( \frac{\tau}{\sigma_2} \right)$ . Обозначим через  $V(z)$  стационарную плотность распределения вероятностей процесса  $z(\tau)$ . Можно показать, что плотность  $V(z)$  имеет вид

$$V(z) = \frac{C}{b(z)} \times \exp \left( \frac{2}{\sigma_2} \int_0^z \frac{a(x)}{b(x)} dx \right).$$

Более подробное описание процедуры построения диффузионной аппроксимации и нахождения вида плотности  $V(z)$  можно найти в [13, 15].

Для построения диффузионной аппроксимации распределения вероятностей числа заявок на неприоритетной орбите будем пользоваться формулой:

$$(3.27) \quad P(i) = \frac{V(i\sigma_2)}{\sum_{n=0}^{\infty} V(n\sigma_2)}.$$

Таким образом, в определении значения константы  $C$  нет необходимости.

#### 4. Алгоритм построения диффузионной аппроксимации в пакете MathCAD

Полученные теоретические результаты были реализованы в пакете MathCAD. Ниже приведен алгоритм численной реализации для нахождения диффузионной аппроксимации числа заявок на неприоритетной орбите.

- Алгоритм 1.* 1) задаем параметры системы:  $\lambda_1, \lambda_2, \mu_1, \mu_2, \sigma_1, \sigma_2$ . Задаем достаточно большое число  $N$ ;
- 2) по формулам (3.4) записываем функции  $G_k(z, x)$ ,  $k = 0, 1, 2$  и  $R_k(x) = G_k(1, x)$ ,  $k = 0, 1, 2$ ;
- 3) по формуле (3.16) записываем функцию  $a(x)$  и находим решение  $\kappa$  стационарного уравнения  $a(x) = 0$ , используя встроенную функцию пакета MathCAD:

$$\kappa := \text{root}(a(x), x, 0, N);$$

- 4) находим значения  $R_0(\kappa)$ ,  $R_1, R_2(\kappa)$ , которые являются стационарными вероятностями состояний прибора;
- 5) записываем функцию

$$G_{02}(z) = \left( \frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1};$$

6) считаем производные по  $z$  от функций  $G_k(z, x)$ ,  $k = 0, 1, 2$  и  $G_{02}(z, x)$ :

$$\begin{aligned}\frac{\partial G_{02}(z)}{\partial z} &= G_{02}(z) \frac{\lambda_1}{\sigma_1} \frac{\lambda_1}{\mu_1 - z\lambda_1}; & \frac{\partial G_1(z)}{\partial z} &= \frac{\lambda_1 G_1(z) + \lambda_1 \frac{\partial G_{02}(z)}{\partial z}}{\mu_1 - z\lambda_1}, \\ \frac{\partial G_0(z, x)}{\partial z} &= \frac{-(\lambda_1 + \lambda_2)G_0(z, x) + \mu_1 G_1(z) + \mu_2 G_2(z, x) - xG_0(z, x)}{\sigma_1 z}, \\ \frac{\partial G_2(z, x)}{\partial z} &= \frac{(\lambda_2 + x)G_{02}(z) - (\lambda_1 + \lambda_2 + \mu_2 + x)G_2(z, x)}{\sigma_1 z}, \\ \frac{\partial^2 G_2(z, x)}{\partial z^2} &= \frac{(\lambda_2 + x)G_{02}(z) - (\lambda_1 + \lambda_2 + \mu_2 + x + \sigma_1) \frac{\partial G_2(z, x)}{\partial z}}{\sigma_1 z};\end{aligned}$$

7) записываем  $A(z, x)$  по формуле

$$\begin{aligned}A(z, x) &= \frac{1}{z-1} \left( a(x) + x \right) G_0(z, x) + z(a(x) - \lambda_2) G_1(z) - \\ &\quad - (\lambda_1 z + \lambda_2 - a(x)) G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} \Big),\end{aligned}$$

$B(z, x)$  по формуле (3.22), для  $h_{02}(z, x)$  – (3.20):

$$h_{02}(z, x) := if \left( z = 1, \frac{1}{\mu_1} A(1, x), h_{02}(z, x) \right);$$

8) находим  $D(z, x)$  (3.19) и  $h_2(1, x)$  (3.18);

9) записываем коэффициент диффузии  $b(x)$  (3.17);

10) строим  $P1(i)$  (3.27):

$$P1(i) = \frac{1}{b(\sigma_2 i)} \exp \left( \frac{2}{\sigma_2} \int_0^{\sigma_2 i} \frac{a(x)}{b(x)} dx \right);$$

11) выполняем нормировку и получаем аппроксимацию дискретного распределения вероятностей числа заявок на неприоритетной орбите

$$P(i) := P1(i) \left( \sum_{i=0}^N P1(i) \right)^{-1}.$$

*Пример 1.* Зададим параметры системы:

$$\lambda_1 = 0,3, \quad \lambda_2 = 0,9, \quad \mu_1 = 1, \quad \mu_2 = 2, \quad \sigma_1 = 1.$$

Получаем распределение вероятностей состояний прибора:

$$R_0 = 0,25, \quad R_1 = 0,3, \quad R_2 = 0,45.$$

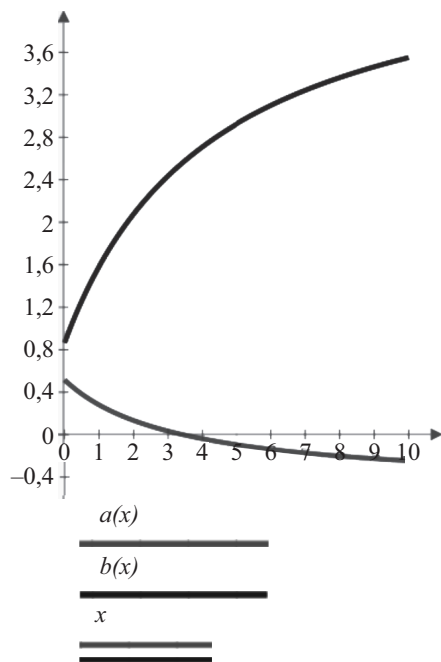


Рис. 2. Графики коэффициента переноса и коэффициента диффузии.

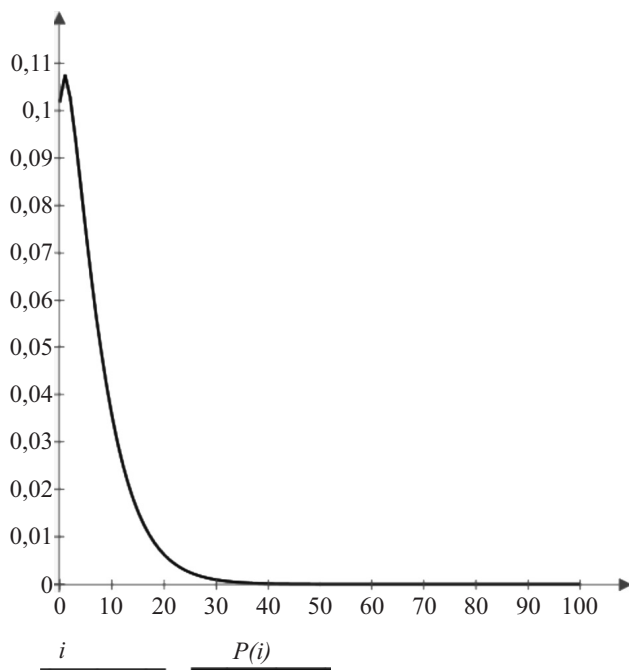


Рис. 3. Диффузионная аппроксимация  $P(i)$  при  $\sigma_2 = 1$ ,  $N = 100$ .

На рис. 2 представлена зависимость коэффициента переноса  $a(x)$  и диффузии  $b(x)$  от числа заявок на неприоритетной орбите. Можно сделать вывод,

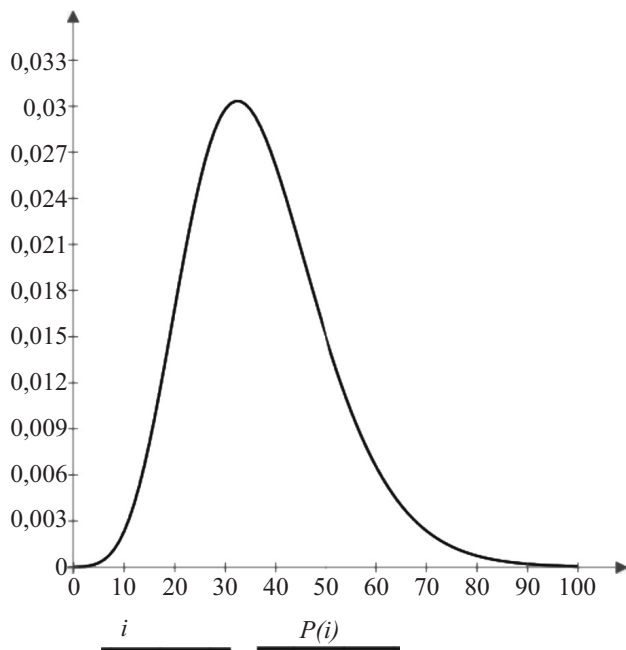


Рис. 4. Диффузионная аппроксимация  $P(i)$  при  $\sigma_2 = 0,1$ ,  $N = 100$ .

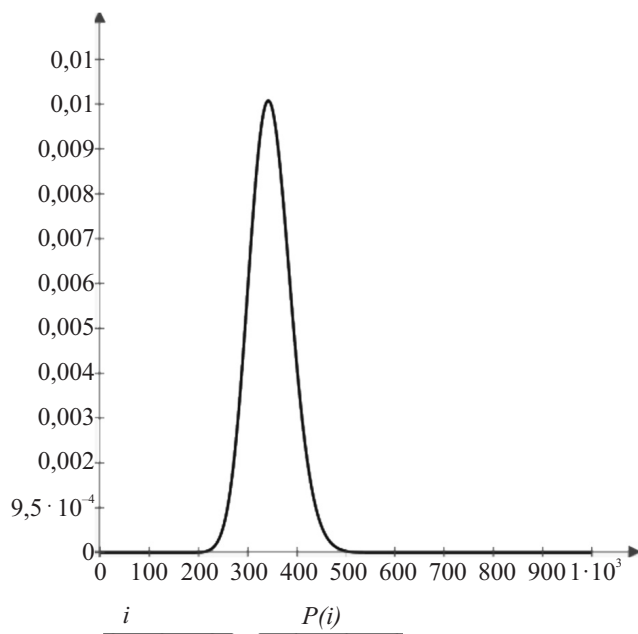


Рис. 5. Диффузионная аппроксимация  $P(i)$  при  $\sigma_2 = 0,01$ ,  $N = 1000$ .

что с увеличением числа заявок разброс относительно среднего увеличивается. На рис. 3–5 изображены графики диффузионной аппроксимации распределения вероятностей числа заявок на неприоритетной орбите.

Численные результаты показывают возможность осуществления предложенного подхода исследования к задачам такого класса. Также по графику распределения можно сделать вывод, что при малом значении  $\sigma_2$  распределение близко к гауссовскому распределению вероятностей. Этот вывод можно подтвердить, обратившись к [14], где исследуется система  $M^{(2)}|B^{(2)}(x)|1$  методом гауссовской аппроксимации.

## 5. Заключение

В работе исследована система массового обслуживания с повторными вызовами и двумя входящими потоками заявок (приоритетный и неприоритетный поток).

Для числа приоритетных заявок на орбите найдена производящая функция в виде взвешенной суммы производящих функций отрицательно-биномиальных распределений.

Для числа неприоритетных заявок на орбите получено распределение вероятностей, которое называем диффузионной аппроксимацией.

Найденные распределения вероятностей позволяют определить все необходимые вероятностно-временные характеристики системы для поступившего числа заявок. Полученные теоретические результаты могут быть использованы для решения ряда практических задач, в которых необходимо разделение заявок по приоритетности: когнитивное радио, передача данных, где необходимо разделение по объему передаваемых пакетов данных. Вызывает интерес исследование задач более общего типа (с произвольной функцией распределения времени обслуживания).

## СПИСОК ЛИТЕРАТУРЫ

1. *Yang T., Templeton J.G.C.* A survey on retrial queues // *Queueing Syst.* 1987. V. 2 (3). P. 201–233.
2. *Artalejo J.R.* A classified bibliography of research on retrial queues // *Progress in* 1990–1999. 1999. V. 7 (2). P. 187–211.
3. *Artalejo J.R.* Accessible bibliography on retrial queues // *Math. Comput. Model.* 1999. V. 30 (3–4). P. 1–6.
4. *Artalejo J.R., Falin G.I.* Standard and retrial queueing systems: a comparative analysis // *Revista Matematica Complutense.* 2002. V. 15. P. 101–129.
5. *Artalejo J.R.* Accessible bibliography on retrial queues // *Math. Comput. Model.* 2010. V. 51. P. 1071–1081.
6. *Choi B.D., Chang Y.* Single server retrial queues with priority calls // *Math. Comput. Model.* 1999. V. 30. P. 7–32.
7. *Yin M., Yan M., Guo Y., Liu M.* Analysis of a preemptive two-priority queueing system with impatient customers and heterogeneous servers // *Mathematics.* 2023. V. 11. P. 3878.
8. *Razumchik R.* Two-priority queueing system with lcs service, probabilistic priority and batch arrivals // *AIP Conference Proceedings.* 2019. V. 2116. P. 090011.

9. *Dimitriou I.* A mixed priority retrial queue with negative arrivals, unreliable server and multiple vacations // *Appl. Math. Model.* 2013. V. 37. P. 1295–1309.
10. *Jain M., Bhagat A., Shekhar C.* Double orbit finite retrial queues with priority customers and service interruptions // *Appl. Math. Comput.* 2015. V. 253. P. 324–344.
11. *Atencia I., Moreno P., Bouza G.* An  $m2/g2/1$  retrial queue with priority customers, 2nd optional service and linear retrial policy // *Invest. Oper.* 2006. V. 27(3). P. 229–248.
12. *Nazarov A., Phung-Duc T., Paul S., Morozova M.* Scaling limits of a tandem queue with two infinite orbits // *Mathematics.* 2023. V. 11(11). P. 2454.
13. *Nazarov A., Phung-Duc T., Izmailova Ya.* Asymptotic-diffusion analysis of multiserver retrial queueing system with priority customers // *Commun. in Comput. and Inform. Sci.* 2021. V. 1391. P. 236–250.
14. *Nazarov A., Phung-Duc T., Paul S., Lizyura O.* Diffusion limit for single-server retrial queues with renewal input and outgoing calls // *Mathematics.* 2022. V. 10(6). P. 948.
15. *Назаров А.А., Измайлова Я.Е.* Исследование  $rq$ -системы  $m(2)|m(2)|1$  с  $r$  настойчивым вытеснением альтернативных заявок // *Вест. Сиб. гос. аэрокос. ун-та им. акад. М.Ф. Решетнева.* 2016. № 17(2). С. 328–334.

*Статья представлена к публикации членом редколлегии А.И. Ляховым.*

Поступила в редакцию 22.04.2024

После доработки 02.03.2025

Принята к публикации 06.03.2025

# Робастное, адаптивное и сетевое управление

© 2025 г. А.А. ГУКОВ (artemiy.gukoff4u@mail.ru)  
(Институт проблем машиноведения РАН, Санкт-Петербург)

## АДАПТИВНОЕ УПРАВЛЕНИЕ ПО ВЫХОДУ С ГАРАНТИЕЙ ЗАДАННОГО КАЧЕСТВА РЕГУЛИРОВАНИЯ<sup>1</sup>

Предложен алгоритм управления по выходу линейными объектами с произвольной относительной степенью в условиях параметрической неопределенности и ограниченных возмущений. В отличие от классических алгоритмов адаптивного управления предложенный алгоритм позволяет гарантировать слежение выхода объекта за эталонным сигналом с нахождением ошибки слежения в заданном разработчиком множестве. Приведен пример, иллюстрирующий эффективность предложенного метода.

*Ключевые слова:* адаптивное управление, динамическая система, замена координат, устойчивость, управление.

DOI: 10.31857/S0005231025050025, EDN: AXWDDU

### 1. Введение

В статье рассматривается задача адаптивного слежения выходного сигнала объекта за эталонным с гарантией заданного качества регулирования в любой момент времени. Рассматриваются линейные системы в условиях параметрической неопределенности и внешних возмущений.

Задача адаптивного управления с эталонной моделью, впервые сформулированная в 50-х годах XX в., является одной из наиболее изученных. Несмотря на ее внушительный возраст, она остается до сих пор актуальной. Возникают новые проблемы, такие как повышение вычислительной эффективности алгоритмов, улучшение качества настройки регуляторов, обобщение методов на более широкие классы систем и т.д.

Первые решения задач адаптивного управления были сопряжены с рядом допущений, например с доступностью измерения вектора состояния, с предположением строгой положительной вещественности передаточной функции системы или с воспроизводимостью задающего сигнала или внешних возмущений с помощью автономного генератора [1–4]. Подходы, использующие аппарат второго метода Ляпунова [5] и теории гиперустойчивости [6], требовали

---

<sup>1</sup> Работа выполнена в ИПМаш РАН при поддержке госзадания № FFNF-2024-0008 (№ 124041100006-1 в ЕГИСУ НИОКТР).

измерения старших производных ошибки слежения. Позже данная проблема была решена с помощью метода расширенной ошибки [7], алгоритмов адаптации высоких порядков [8], методов компенсации возмущений [9, 10] и т.д.

Одна из ключевых проблем в вышеописанных методах состоит в невозможности повлиять на качество переходных процессов [11]. Для частичного решения данной проблемы предлагались различные решения, например схема с ускоренной сходимостью [12, 13], схема с использованием больших коэффициентов в обратной связи [14], выполнение условия неисчезающего возбуждения [15], негладкие законы управления [16] и т.д.

Таким образом, методы [7–10, 12–16] сопряжены с наложением существенных ограничений и решают задачу не в полном объеме, достигая цели управления лишь в асимптотике. При этом полученные оценки характеристик предельного множества достаточно грубые.

В [17–21] предложен метод, гарантирующий нахождение выходного сигнала в заданном множестве. В [22] на базе метода [17–21] предложен адаптивный закон управления с нахождением выходного сигнала в заданном множестве. Однако в [22] предполагается, что относительная степень объекта равна единице. В данной статье с использованием подходов [17–22] и модифицированного алгоритма адаптации [23] предлагается новое решение задачи адаптивного управления с гарантией заданного качества регулирования минимально-фазовыми объектами [7, 8, 12, 14–16] с произвольной относительной степенью.

Статья организована следующим образом. Во втором разделе формулируется задача адаптивного слежения с ограничениями на выходную переменную. В третьем разделе сначала синтезируется закон управления, предполагающий доступность измерению производных выходного сигнала объекта. Затем полученное решение обобщается на случай недоступности измерению данных производных. В четвертом разделе приведено численное моделирование, иллюстрирующее эффективность полученного решения.

## 2. Постановка задачи

Рассмотрим динамическую систему

$$(1) \quad Q(p)y(t) = kR(p)u(t) + f(t),$$

где  $t \geq 0$ ,  $u(t) \in \mathbb{R}$  – сигнал управления,  $y(t) \in \mathbb{R}$  – регулируемый сигнал, доступный измерению,  $f(t) \in \mathbb{R}$  – ограниченное внешнее возмущение,  $Q(s)$  и  $R(s)$  – нормализованные полиномы (т.е. полиномы со старшими коэффициентами, равными 1) с неизвестными вещественными коэффициентами и со степенями, равными  $n$  и  $m$  соответственно,  $\rho = n - m \geq 1$ , полином  $R(s)$  гурвицев,  $p = d/dt$  – оператор дифференцирования, коэффициент  $k > 0$  неизвестен, начальные условия  $y^{(i)}(0)$ ,  $i = \overline{2, n}$  неизвестны, но известно множество начальных условий  $y(0)$ . Всюду в статье  $s$  – комплексная переменная.

Зададим эталонную модель:

$$(2) \quad T(p)y_m(t) = k_m g(t),$$

где  $g(t) \in \mathbb{R}$  – ограниченное и  $(\rho - 1)$  раз дифференцируемое задающее воздействие,  $y_m(t) \in \mathbb{R}$  – выход эталонной модели,  $T(s)$  – известный нормализованный гурвицев полином с вещественными коэффициентами и степени  $\rho$ ,  $k_m > 0$ .

Цель работы состоит в синтезе закона управления, обеспечивающего принадлежность ошибки слежения  $e(t) = y(t) - y_m(t)$  следующему множеству:

$$(3) \quad \mathcal{E} = \{(t, e) \in \mathbb{R}^2 \mid t \geq 0, \underline{g}(t) < e(t) < \bar{g}(t)\},$$

где функции  $\underline{g}(t) < 0$ ,  $\bar{g}(t) > 0$ ,  $\bar{g}(t) - \underline{g}(t) > \delta$ ,  $\delta > 0$  ограничены и имеют ограниченные первые производные для любых  $t \geq 0$ , а также  $e(0) \in \mathcal{E}$ . Ограниченность производных  $\underline{g}(t)$  и  $\bar{g}(t)$  необходима для применения метода [19].

### 3. Основной результат

#### 3.1. Синтез идеального закона управления

Введем вспомогательное управляющее воздействие  $v$  и сначала предположим, что его производные доступны измерению. Рассмотрим закон управления

$$(4) \quad u(t) = \frac{T(p)}{p} v(t).$$

Представим полиномы  $Q(s)$  и  $R(s)$  в виде  $Q(s) = Q_m(s) + \Delta Q(s)$  и  $R(s) = R_m(s) + \Delta R(s)$ , где нормализованные гурвицевы полиномы  $Q_m(s)$  и  $R_m(s)$  имеют степени  $n$  и  $m$  соответственно, и  $Q_m(s)/R_m(s) = T(s)$ . Неизвестные полиномы  $\Delta Q(s)$  и  $\Delta R(s)$  имеют степени  $n - 1$  и  $m - 1$  соответственно. Преобразуем (1), учитывая (4), к виду

$$(5) \quad Q_m(p)y(t) = \frac{kR_m(p)T(p)}{p}v(t) + \frac{k\Delta R(p)T(p)}{p}v(t) - \Delta Q(p)y(t) + f(t).$$

Разделив (5) на  $Q_m(p)$ ,  $kR_m(p)T(p)$  и  $p$ , запишем результат в виде

$$(6) \quad y(t) = \frac{k}{p} \left[ v(t) + \frac{\Delta R(p)}{R_m(p)} v(t) - \frac{p\Delta Q(p)}{kQ_m(p)} y(t) + \frac{p}{kQ_m(p)} f(t) + \epsilon_1(t) \right],$$

где  $\epsilon_1(t)$  – экспоненциально затухающая функция, зависящая от начальных условий (1). Выразим  $y_m(t)$  из (2) в виде

$$(7) \quad y_m(t) = \frac{k}{p} \left[ \frac{k_m}{k} g_r(t) + \epsilon_2(t) \right],$$

где  $g_r(t) = \frac{p}{T(p)}g(t)$ ,  $\epsilon_2(t)$  – экспоненциально затухающая функция, зависящая от начальных условий (2). Учитывая структуру (6) и (7), перепишем  $e(t)$  в виде

$$(8) \quad e(t) = \frac{k}{p} \left[ v(t) - c_{01}y(t) - c_{02}^\top \zeta_y(t) - c_{03}^\top \zeta_v(t) - \frac{k_m}{k} g_r(t) + \frac{p}{kQ_m(p)} f(t) + \epsilon(t) \right],$$

где  $\epsilon(t) = \epsilon_1(t) - \epsilon_2(t)$ , сигналы  $\zeta_y(t)$ ,  $\zeta_v(t)$  и  $g_r(t)$  сформированы с помощью следующих фильтров:

$$(9) \quad \begin{aligned} \dot{\zeta}_y(t) &= F_y \zeta_y(t) + b_y y(t), & \zeta_y(0) &= 0, \\ \dot{\zeta}_v(t) &= F_v \zeta_v(t) + b_v v(t), & \zeta_v(0) &= 0, \\ \dot{\zeta}_g(t) &= F_g \zeta_g(t) + b_g g(t), & \zeta_g(0) &= 0, & g_r(t) &= L_2 \zeta_g(t). \end{aligned}$$

Здесь  $b_i^\top = [0 \dots 0 \ 1]$  – вектор-столбец с единицей на последней позиции и нулями на остальных,  $i \in \{y, v, g, \eta\}$ ,  $L_j = [0 \dots 0 \ 1 \ 0 \dots 0]$  – вектор-строка с единицей на  $j$ -й позиции и нулями на остальных. Здесь и далее матрицы  $b_i$  и  $L_j$  будут иметь ту же структуру, а их размерность будет понятна из контекста. Матрицы  $F_y$ ,  $F_v$  и  $F_g$  фильтров (9) заданы в форме Фробениуса с характеристическими полиномами  $Q_m(s)$ ,  $R_m(s)$  и  $T(s)$  соответственно. Коэффициент  $c_{01}$  получен из следующего выражения:

$$\frac{p\Delta Q(p)}{kQ_m(p)} = c_{01} + \frac{\Delta \tilde{Q}(p)}{kQ_m(p)},$$

где  $\Delta \tilde{Q}(s)$  – неизвестный полином степени  $n - 1$ . Векторы  $c_{02}$  и  $c_{03}$  состоят из коэффициентов многочленов  $\Delta \tilde{Q}(s)/k$  и  $\Delta R(s)/k$  соответственно. Введем вектор постоянных неизвестных параметров  $c_0^\top = [c_{01} \ c_{02}^\top \ c_{03}^\top \ k_m/k]$  и вектор регрессии  $\omega^\top(t) = [y(t) \ \zeta_y^\top(t) \ \zeta_v^\top(t) \ g_r(t)]$ . Перепишем (8) как

$$(10) \quad \dot{e}(t) = k \left( v(t) - c_0^\top \omega(t) + \bar{f}(t) + \epsilon(t) \right),$$

где  $\bar{f}(t) = \frac{p}{kQ_m(p)} f(t)$  – сигнал, ограниченный в силу гурвицевости  $Q_m(s)$ .

Применим метод [19] для задания вспомогательного управляющего воздействия  $v(t)$ , гарантирующего нахождение ошибки  $e(t)$  во множестве (3). Для этого введем в рассмотрение вспомогательный сигнал  $\varepsilon(t)$ , вычисляемый из формулы

$$(11) \quad e(t) = \Phi(\varepsilon(t), t),$$

где функция  $\Phi(\varepsilon(t), t)$  удовлетворяет условиям:

а)  $\underline{g}(t) < \Phi(\varepsilon(t), t) < \bar{g}(t)$  для любых  $t \geq 0$  и  $\varepsilon \in \mathbb{R}$ ;

б) функция  $\Phi(\varepsilon(t), t)$  непрерывно-дифференцируемая по  $\varepsilon(t)$  и  $t$ , причем для любых  $\varepsilon \in \mathcal{E}$  и  $t \geq 0$  выполнено

$$\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \neq 0;$$

в) функция  $\frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$  ограничена для любых  $\varepsilon \in \mathbb{R}$  и  $t \geq 0$ .

Приведем пример преобразования (11) в виде

$$(12) \quad \Phi(\varepsilon(t), t) = \frac{\bar{g}(t) \exp(\varepsilon(t)) + \underline{g}(t)}{\exp(\varepsilon(t)) + 1}.$$

Другие примеры подобных функций (11) можно найти в [19].

Принимая во внимание (11), рассмотрим полную производную по времени от  $e(t)$  в виде

$$\dot{e}(t) = \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \dot{\varepsilon}(t) + \frac{\partial \Phi}{\partial t}(\varepsilon(t), t).$$

Учитывая (10) и свойство (б), выразим  $\dot{e}(t)$  следующим образом:

$$(13) \quad \dot{e}(t) = \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left( k(v(t) - c_0^\top \omega(t) + \bar{f}(t) + \varepsilon(t)) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t) \right).$$

Теперь выберем вспомогательное управляющее воздействие и алгоритм адаптации в виде

$$(14) \quad v(t) = -\text{sign} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \alpha \varepsilon(t) + c^\top(t) \omega(t),$$

$$(15) \quad \dot{c}(t) = - \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t) \omega(t) - \gamma c(t),$$

где  $\alpha > 0$  и  $\gamma > 0$ ,  $c(t)$  – вектор настраиваемых параметров. Преобразование  $\Phi(\varepsilon(t), t)$  выбирается заранее, а знак  $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t)$  неизменен в силу свойства (б), поэтому значение функции  $\text{sign}\{\cdot\}$  в (14) известно. Подставляя (14) в (13), получим следующее выражение:

$$(16) \quad \dot{e}(t) = \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left( -\text{sign} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) + \right. \\ \left. + k(c(t) - c_0)^\top \omega(t) + \Xi(t) \right),$$

где  $\Xi(t) = k(\bar{f}(t) + \varepsilon(t)) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$  – ограниченная функция. Сформулируем теорему, доказательство которой приведено в Приложении.

*Теорема 1.* Пусть функции  $\underline{g}(t)$  и  $\bar{g}(t)$  удовлетворяют наложенным требованиям (см. абзац после (3)) и выполнены свойства (а)–(в) преобразования (11). Тогда для любых  $\alpha > 0$  и  $\gamma > 0$  в замкнутой системе (1), (2), (9), (15), (16) будет достигнута цель управления (3).

*Замечание 1.* В [22] рассмотрена задача с относительной степенью объекта, равной 1, и известным коэффициентом  $k$ . В отличие от представленного в [22] алгоритма предложенная в настоящей работе процедура позволяет исключить слагаемое, содержащее  $e(t)$ , из динамики  $\varepsilon(t)$  и выбрать упрощенный закон управления, а также преодолеть проблему неизвестного  $k$ .

### 3.2. Синтез реализуемого закона управления

Закон управления (4) содержит производные вспомогательного управляющего воздействия  $v$  вплоть до  $(\rho - 1)$  порядка. Введем в рассмотрение оценку  $\tilde{v}$  вспомогательного управляющего воздействия (14). Тогда новый закон управления будет выглядеть как

$$(17) \quad u(t) = \frac{T(p)}{p} \tilde{v}(t),$$

$$(18) \quad \dot{\xi}(t) = G_0 \xi(t) + D_0(\tilde{v}(t) - v(t)), \quad \tilde{v}(t) = \xi_1(t) = L_1 \xi(t).$$

Здесь  $\xi(t) \in \mathbb{R}^\rho$  – вектор оценок сигнала  $v$  и его производных,

$$G_0 = \begin{bmatrix} 0 & I_{\rho-1} \\ 0 & 0 \end{bmatrix} \quad D_0^\top = \begin{bmatrix} -\frac{d_1}{\mu} & -\frac{d_2}{\mu^2} & \dots & -\frac{d_\rho}{\mu^\rho} \end{bmatrix},$$

числа  $d_i$ ,  $i = \overline{1, \rho}$  выбираются так, чтобы матрица  $G = G_0 + DL_1$  была гурвицевой,  $D^\top = [d_1 \ d_2 \ \dots \ d_\rho]$ , а  $\mu > 0$  – достаточно малое число. Введем в рассмотрение следующий вектор:

$$(19) \quad \eta(t) = \Gamma^{-1}(\xi(t) - \theta(t)), \quad \Gamma = \text{diag} \{ \mu^{\rho-1}, \mu^{\rho-2}, \dots, \mu, 1 \},$$

где  $\theta^\top(t) = [v(t) \ \dot{v}(t) \ \dots \ v^{(\rho-1)}(t)]$ . Принимая во внимание (19), запишем:

$$(20) \quad \Delta v(t) = \tilde{v}(t) - v(t) = \mu^{\rho-1} \eta_1(t) = \mu^{\rho-1} L_1 \eta(t), \quad \tilde{v}(t) = v(t) + \mu^{\rho-1} L_1 \eta(t).$$

С учетом (17) и (20) выразим  $\dot{\varepsilon}$  как

$$(21) \quad \dot{\varepsilon}(t) = \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left( -\text{sign} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) + \right. \\ \left. + k(c(t) - c_0)^\top \omega(t) + \mu^{\rho-1} k L_1 \eta(t) + \Xi(t) \right).$$

Учитывая (18) и (19), получим

$$(22) \quad \mu \dot{\eta}(t) = G \eta(t) - \mu b_\eta v^{(\rho)}(t).$$

Сформулируем основной результат статьи.

*Теорема 2.* Пусть выполнены условия теоремы (1). Тогда существует такое число  $\mu_0$ , что при  $\mu \leq \mu_0$  в замкнутой системе (1), (2), (9), (15), (21), (22) будет достигнута цель управления (3).

*Замечание 2.* Уравнение (22) представляет собой сингулярно возмущенную динамическую систему. Из анализа таких систем известно [24], что при определенных условиях на правую часть системы и при достаточно малых  $\mu$  система обладает той же областью диссипативности и той же областью притяжения, что и система при  $\mu = 0$ , что эквивалентно использованию

идеального закона управления. Как было показано в теореме (1), такая система достигает цели (3), поэтому для доказательства теоремы достаточно будет воспользоваться одним из результатов теории сингулярно возмущенных систем.

Теорема 2 показывает существование достаточно малого параметра  $\mu_0$ , в то время как поиск фактического значения является неразрешенной проблемой. Поиск некоторых количественных характеристик, в структуру которых входят наблюдатели высокого порядка, не всегда возможен и зачастую остается в качественном виде [25, 26]. В [27, 28] отмечается, что итеративный поиск величины  $\mu_0$ , при котором достигается устойчивость замкнутой системы, может осуществляться на этапе моделирования.

*Доказательство теоремы 2.* Перепишем (21) и (22) в виде

$$(23) \quad \begin{aligned} \dot{\varepsilon}(t) = & \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left( -\text{sign} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) + \right. \\ & \left. + k(c(t) - c_0)^\top \omega(t) + \mu_2^{\rho-1} k L_1 \eta(t) + \Xi(t) \right), \\ \mu_1 \dot{\eta}(t) = & G \eta(t) - \mu_2 b_\eta v^{(\rho)}(t), \end{aligned}$$

где  $\mu_1 = \mu_2 = \mu$ . Воспользуемся леммой из [29].

*Лемма.* Если система описывается уравнением  $\dot{x} = f(x, \mu_1, \mu_2)$ ,  $x \in \mathbb{R}^{m_0}$ , где  $f(x, \mu_1, \mu_2)$  – непрерывная функция, липшицева по  $x$  и при  $\mu_2 = 0$  имеет ограниченную замкнутую область диссипативности  $D = \{x \mid F(x) < C\}$ , где  $F(x)$  – положительно определенная в  $\mathbb{R}^{m_0}$  (в смысле [30]) непрерывная кусочно-гладкая функция, то существует  $\mu_0 > 0$  такое, что при  $\mu_2 \leq \mu_0$  исходная система имеет ту же область диссипативности  $D$ , если для некоторых  $C_1 > 0$  и  $\bar{\mu}_1 > 0$  при  $\mu_2 = 0$  выполнено соотношение

$$\sup_{|\mu_1| \leq \bar{\mu}_1} \left\{ \left\langle \frac{\partial F}{\partial x}(x), f(x, \mu_1, 0) \right\rangle \Big|_{F(x)=C} \right\} \leq -C_1.$$

При подстановке  $\mu_2 = 0$  в (23) имеем полученную в предыдущем подразделе замкнутую систему с идеальным законом управления. В теореме 1 показано стремление решений замкнутой системы в ограниченное множество, а дополнительное уравнение  $\mu_1 \dot{\eta}(t) = G \eta(t)$  не нарушает данное условие в силу гурвицевости  $G$ . Значит, при достаточно малом  $\mu$  система сохранит область диссипативности. Теорема доказана.

#### 4. Моделирование

*Пример 1.* Рассмотрим объект управления (1) со следующими линейными дифференциальными операторами:

$$Q(p) = p^4 + 6p^3 - 3p^2 - p + 2 \quad \text{и} \quad kR(p) = p + 1.$$

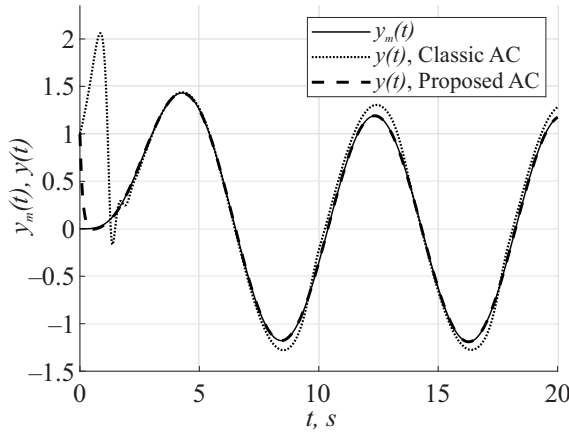


Рис. 1. Выход эталонной модели (сплошная кривая), выход объекта (1) при использовании адаптивного робастного алгоритма высокого порядка (точечная кривая), выход объекта (1) при использовании предложенного алгоритма (пунктирная кривая).

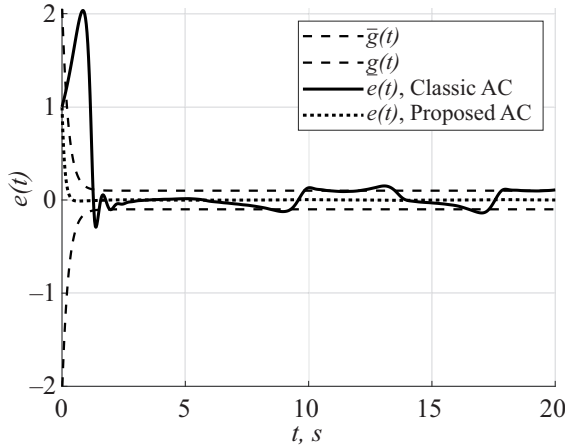


Рис. 2. Ограничения  $\bar{g}(t)$  и  $\underline{g}(t)$  (пунктирные кривые), ошибка регулирования при использовании адаптивного робастного алгоритма высокого порядка (сплошная кривая), ошибка регулирования при использовании предложенного алгоритма (точечная кривая).

Начальные условия  $p^3y(0) = p^2y(0) = py(0) = y(0) = 1$ . Внешнее возмущение  $f(t) = 2 \sin(1,5t) + d(t)$ , где  $d(t) = \text{sat}\{\hat{d}(t)\}$ ,  $\text{sat}\{\cdot\}$  – функция насыщения,  $\hat{d}(t)$  – белый шум, моделируемый в Matlab Simulink с помощью блока “Band-Limited White Noise” с параметром “Noise power”, равным 1, и “Sample time”, равным 0,1. Эталонная модель (2), на которую подается задающее воздействие  $g(t) = 2,5 \sin(0,8t)$ , имеет следующие параметры:

$$T(p) = p^3 + 3p^2 + 3p + 1 \quad \text{и} \quad k_m = 1.$$

Согласно (4) зададим закон управления в виде  $u(t) = \frac{T(p)}{p}v(t)$ . В качестве функции  $\Phi(\varepsilon(t), t)$  возьмем (12). Ограничения для ошибки слежения зададим функциями  $\bar{g}(t) = 2 \exp(-4t) + 0,1$  и  $\underline{g}(t) = -\bar{g}(t)$ . Выберем  $Q_m(p) = p^4 + 4p^3 + 6p^2 + 4p + 1$ ,  $R_m(p) = p + 1$  и зададим в фильтрах (9):  $F_v = -1$ ,  $F_g = \begin{bmatrix} 0 & I_2 \\ -1 & -3 & -3 \end{bmatrix}$  и  $F_y = \begin{bmatrix} 0 & I_3 \\ -1 & -4 & -6 & -4 \end{bmatrix}$ . Выберем в (14), (15) и (18) параметры  $\alpha = 5$ ,  $\gamma = 1$  и  $\mu = 10^{-3}$ . Сравним предложенный алгоритм управления с классическим адаптивным алгоритмом адаптации высокого порядка [31] с параметрами  $\sigma_c = 5$ ,  $\gamma_c = 100$  и  $\mu_c = 200$ .

На рис. 1 и 2 видно, что классический алгоритм не справляется с поставленной задачей. Во-первых, отсутствует возможность задать качество регулирования во время переходных процессов. Во-вторых, предельное множество не определяется заранее и оказывается больше, чем заданное с помощью предложенного алгоритма.

## 5. Заключение

В статье предложен новый алгоритм адаптивного слежения за выходом эталонной модели с гарантией заданного качества регулирования на всем промежутке функционирования системы. Метод объединяет в себе идею адаптивного управления [23] и подход [19], использованный для синтеза нелинейного закона управления в условиях ограничений. Произведено сравнение полученного результата с классическим законом адаптивного управления высокого порядка [31] путем моделирования. Предложенный алгоритм позволил обеспечить нахождение ошибки слежения в заранее заданном множестве в любой момент времени, в то время как методы [23, 31] не позволяют контролировать значения ошибки слежения в переходном режиме. Кроме того, оценки величины предельного целевого множества, полученные в [23, 31], зависят от значений неизвестных параметров и имеют завышенные значения, в связи с чем не могут использоваться для задания точности регулирования в установившемся режиме. Предложенный подход решает обе указанные проблемы, а качество регулирования полностью определяется выбором ограничивающих функций.

## ПРИЛОЖЕНИЕ

### Доказательство теоремы 1.

В силу свойства (б) преобразования (11) функция  $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t)$  является знакоопределенной. Сначала предположим, что  $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) > 0$ . Выберем функцию Ляпунова вида

$$V_1 = 0,5\varepsilon^2(t) + 0,5k(c(t) - c_0)^\top (c(t) - c_0) + \chi \int_t^\infty \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(s), s) \right)^{-1} \varepsilon^2(s) ds,$$

где  $\chi > 0$ . Учитывая (15) и (16), найдем  $\dot{V}_1$  в виде

$$(II.1) \quad \begin{aligned} \dot{V}_1 = & -\alpha k \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon^2(t) + \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t)(\psi(t) + k\varepsilon(t)) - \\ & - \gamma k(c(t) - c_0)^\top c(t) - \chi \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon^2(t), \end{aligned}$$

где  $\psi(t) = k\bar{f}(t) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$ . Воспользуемся следующими соотношениями:

$$(II.2) \quad \begin{aligned} \varepsilon(t)\psi(t) & \leq 0,5(\nu^{-1}\varepsilon^2(t) + \nu\psi^2(t)), \\ \varepsilon(t)\varepsilon(t) & \leq 0,5(\nu^{-1}\varepsilon^2(t) + \nu\varepsilon^2(t)), \\ -(c(t) - c_0)^\top c(t) & = -0,5((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t) - c_0^\top c_0). \end{aligned}$$

С учетом (II.2) перепишем (II.1) как

$$(II.3) \quad \begin{aligned} \dot{V}_1 \leq & - \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\alpha k - 0,5\nu^{-1}(1+k))\varepsilon^2(t) + \\ & + 0,5 \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \nu\psi^2(t) + 0,5\gamma k c_0^\top c_0 - \\ & - 0,5\gamma k ((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t)) - \\ & - \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\chi - 0,5k\nu)\varepsilon^2(t). \end{aligned}$$

Заметим, из свойств (а) и (б) функции (11) следует, что она всюду дифференцируема, строго монотонна и ограничена, а значит,  $\sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} < \infty$ . Кроме того,  $\sup_{t \geq 0} \{\psi(t)\} < \infty$  согласно свойству (в). Тогда из (II.3) следует, что  $\dot{V}_1 < 0$  достигается при выполнении следующих условий:

$$(II.4) \quad \nu > 0,5\alpha^{-1}(k^{-1} + 1), \quad \chi > 0,5k\nu,$$

$$|\varepsilon(t)| > \sqrt{\frac{0,5}{\alpha k - 0,5\nu^{-1}(1+k)} \left( \frac{\nu}{k} \sup_{t \geq 0} \{\psi(t)\}^2 + \sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \gamma k c_0^\top c_0 \right)}.$$

Очевидно, что всегда найдутся  $\nu$  и  $\chi$  такие, что условия (II.4) выполнены. Следовательно,  $\dot{V}_1 < 0$ .

Теперь рассмотрим случай  $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) < 0$ . Выберем функцию Ляпунова вида

$$V_2 = 0,5\varepsilon^2(t) + 0,5k(c(t) - c_0)^\top (c(t) - c_0).$$

Найдем  $\dot{V}_2$  в виде

$$(II.5) \quad \begin{aligned} \dot{V}_2 = & \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} k\alpha\varepsilon^2(t) + \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t)(\psi(t) + k\varepsilon(t)) - \\ & - \gamma k(c(t) - c_0)^\top c(t). \end{aligned}$$

Применяя соотношения (П.2) при  $\nu = 1$ , перепишем (П.5) как

$$\begin{aligned}
 \dot{V}_2 \leq & \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\alpha k + 0,5k + 0,5) \varepsilon^2(t) + \\
 & + 0,5 \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \psi^2(t) + 0,5\gamma k c_0^\top c_0 - \\
 & - 0,5\gamma k ((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t)) + \\
 & + 0,5 \left( \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} k \varepsilon^2(t).
 \end{aligned}
 \tag{П.6}$$

Так как  $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) < 0$ , то все слагаемые, кроме  $0,5\gamma k c_0^\top c_0$ , отрицательны. Ранее показано, что  $\sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} < \infty$ . Тогда из (П.6) следует, что  $\dot{V}_2 < 0$  достигается при выполнении условия

$$|\varepsilon(t)| > \sqrt{-\inf_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \frac{0,5\gamma k c_0^\top c_0}{\alpha k + 0,5k + 0,5}}.
 \tag{П.7}$$

Неравенства (П.4) и (П.7) определяют множества, в которые стремятся траектории (16) в каждом из рассмотренных случаев. Тогда по теореме 3.1 из [19] траектории (10) будут принадлежать некоторому подмножеству  $\mathcal{E}$ , а это означает выполнение цели (3). Теорема доказана.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Landau I.D.* A survey of model reference adaptive techniques—Theory and applications // *Automatica*. 1974. V. 10. No. 4. P. 353–379. ISSN 0005-1098. [https://doi.org/10.1016/0005-1098\(74\)90064-8](https://doi.org/10.1016/0005-1098(74)90064-8)
2. *Chang-chieh Hang, Parks P.C.* Comparative studies of model reference adaptive control systems // *IEEE Trans. Autom. Control*. 1973. V. 18. No. 5. P. 419–428.
3. *Johnson C.D.* Accommodation of external disturbances in linear regulator and servomechanism problems // *IEEE Trans. Autom. Control*. 1971. V. 16. No. 6. P. 635–644.
4. *Davison E.J.* The robust control of a servomechanism problem for linear timeinvariant multivariable systems // *IEEE Trans. Autom. Control*. 1976. V. 21. P. 25–34.
5. *Parks P.* Liapunov redesign of model reference adaptive control systems // *IEEE Trans. Autom. Control*. 1976. V. 11. No. 3. P. 362–367.
6. *Landau I.D., Ioan D.* A hyperstability criterion for model reference adaptive control systems // *IEEE Trans. Autom. Control*. 1969. V. 14. P. 552–555. <https://api.semanticscholar.org/CorpusID:121498902>
7. *Morse A.S.* Global stability of parameter adaptive control systems // *IEEE Trans. Autom. Control*. 1980. V. 25. No. 3. P. 433–439.
8. *Morse A.S.* High-order parameter tuners for the adaptive control of nonlinear systems // *US-Italy Seminar Syst. Model. Feedback Theor. Appl.* 1992. V. 12. P. 339–364. [https://doi.org/10.1007/978-1-4757-2204-8\\_23](https://doi.org/10.1007/978-1-4757-2204-8_23)

9. Андриевский Б.Р., Фуртат И.Б. Наблюдатели возмущений: методы и приложения. Часть 1. Методы // *АиТ*. 2020. № 9. С. 3–61.
10. Андриевский Б.Р., Фуртат И.Б. Наблюдатели возмущений: методы и приложения. Часть 2. Приложения // *АиТ*. 2020. № 10 С. 35–92.
11. Zang Z., Bitmead, R. Transient bounds for adaptive control systems // *IEEE Trans. Autom. Control*. 1991. V. 5. No. 39. P. 2724–2729.  
<http://dx.doi.org/10.1109/CDC.1990.203273>
12. Kreisselmeier G. Adaptive observers with exponential rate of convergence // *IEEE Trans. Autom. Control*. 1977. V. 22. No. 1. P. 2–8.  
<https://doi.org/10.1109/TAC.1977.1101401>
13. Lion P.M. Rapid identification of linear and nonlinear systems // *AIAA J*. 1967. V. 5. P. 1835–1842.
14. Sun J. A modified model reference adaptive control scheme for improved transient performance // *IEEE Trans. Autom. Control*. 1993. V. 38. No. 8. P. 1255–1259.  
<https://doi.org/10.1109/9.233162>
15. Arteaga M., Tang Y. Adaptive control of robots with an improved transient performance // *IEEE Trans. Autom. Control*. 2002. V. 47. P. 1198–1202.  
<http://dx.doi.org/10.1109/TAC.2002.800672>
16. Morse A.S. Supervisory Control of Families of Linear Set-Point Controllers-Part I. Exact Matching // *IEEE Trans. Autom. Control*. 1996. V. 41. No. 10. P. 1413–1431.  
<https://doi.org/10.1109/9.539424>
17. Фуртат И.Б. Дивергентные условия устойчивости динамических систем // *АиТ*. 2020. № 2. С. 62–75.
18. Фуртат И.Б., Гуцин П.А. Управление динамическими объектами с гарантией нахождения регулируемого сигнала в заданном множестве // *АиТ*. 2021. № 4. С. 121–139.
19. Furtat I.B., Gushchin P.A. Nonlinear feedback control providing plant output in given set // *Int. J. Control*. 2022. V. 95. No. 6. P. 1533–1542.  
<https://doi.org/10.1080/00207179.2020.1861336>
20. Фуртат И.Б. Плотностные системы. Анализ и управление // *АиТ*. 2023. № 11. С. 55–76.
21. Фуртат И.Б., Гуцин П.А., Нгуен Ба Хю. Управление динамическими системами при ограничениях на входные и выходные сигналы // *АиТ*. 2023. № 4. С. 45–63.
22. Фуртат И.Б., Гуцин П.А., Нгуен Б., Колесник Н.С. Адаптивное управление с гарантией заданного качества регулирования // *УБС*. 2023. № 102. С. 44–57.  
<https://doi.org/10.25728/ubs.2023.102.3>
23. Цыкунов А.М. Модифицированный адаптивный алгоритм высокого порядка для управления линейным объектом по выходу // *АиТ*. 2006. № 8. С. 143–153.
24. Халил Х.К. Нелинейные системы. М.-Ижевск: Ин-т компьютерных исслед., 2009.
25. Atassi A.N., Khalil H.K. A separation principle for the stabilization of a class of nonlinear systems // *IEEE Transact. Autom. Control*. 1999. V. 44. No. 9. P. 1672–1687. <https://doi.org/10.1109/9.788534>

26. *Borisov O.I., Pyrkin A.A., Isidori A.* Robust Output Regulation of Permanent Magnet Synchronous Motors by Enhanced Extended Observer // IFAC-PapersOnLine. 2020. V. 53. No. 2. P. 4881–4886. ISSN 2405-8963. <https://doi.org/10.1016/j.ifacol.2020.12.1056>
27. *Цыкунов А.М.* Алгоритм робастного управления нестационарным линейным объектом с компенсацией возмущения // Изв. РАН. Теория и системы управления. 2008. № 4. С. 33–40.
28. *Цыкунов А.М.* Алгоритмы робастного управления с компенсацией ограниченных возмущений // АиТ. 2007. № 7. С. 103–115.
29. *Брусин В.А.* Об одном классе сингулярно-возмущенных адаптивных систем. I // АиТ. 1995. № 4. С. 119–129. <https://doi.org/10.25728/ubs.2023.102.3>
30. *Руш Н., Абетс П., Лалуа М.* Прямой метод Ляпунова в теории устойчивости. М.: Мир, 1980.
31. *Мирошник И.В., Никифоров В.О., Фрадков А.Л.* Нелинейное и адаптивное управление сложными динамическими системами. Учебное пособие. СПб.: Наука, 2000.

*Статья представлена к публикации членом редколлегии А.И. Матасовым.*

Поступила в редакцию 21.08.2024

После доработки 10.03.2025

Принята к публикации 16.03.2025

## Управление в технических системах

© 2025 г. Д.В. ЕФАНОВ, д-р техн. наук (TrES-4b@yandex.ru)  
(Санкт-Петербургский политехнический университет Петра Великого;  
Российский университет транспорта, Москва),  
Д.В. ПИВОВАРОВ, канд. техн. наук (pivovarov.d.v.spb@gmail.com)  
(Петербургский государственный университет путей сообщения  
Императора Александра I, Санкт-Петербург)

### СИНТЕЗ САМОПРОВЕРЯЕМЫХ ДИСКРЕТНЫХ УСТРОЙСТВ НА ОСНОВЕ ПОЛИНОМИАЛЬНЫХ КОДОВ С КОНТРОЛЕМ ВЫЧИСЛЕНИЙ ПО НЕСКОЛЬКИМ ДИАГНОСТИЧЕСКИМ ПРИЗНАКАМ

Предложено при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам использовать полиномиальные коды. Разработан алгоритм быстрого получения функций, описывающих проверочные символы полиномиальных кодов в виде логических выражений. Показано, что кодеры полиномиальных кодов могут быть отнесены к устройствам трех видов: 1) на выходах которых реализуются исключительно самодвойственные булевы функции, 2) на выходах которых реализуются исключительно «близкие» к самодвойственным (самоквазидвойственным) булевы функции и 3) на выходах которых реализуются и самодвойственные, и самоквазидвойственные булевы функции. Разработана классификация полиномиальных кодов, учитывающая эту особенность. Описана структура организации контроля вычислений на выходах самодвойственных дискретных устройств по нескольким диагностическим признакам. Предложен алгоритм синтеза полностью самопроверяемого дискретного устройства с контролем вычислений по нескольким диагностическим признакам, отличающийся от известных тем, что учитывает характер возникающих на выходах дискретных устройств ошибок и предварительное покрытие их с помощью тестеров самодвойственных и/или самоквазидвойственных сигналов. Результаты могут быть использованы при разработке автоматизированных средств проектирования дискретных устройств для широкого круга приложений.

*Ключевые слова:* самопроверяемые дискретные устройства, контроль вычислений по нескольким диагностическим признакам, контроль самодвойственности сигналов, контроль самоквазидвойственности сигналов, полиномиальные коды.

DOI: 10.31857/S0005231025050036, EDN: AXPIOQ

## 1. Введение

Рост сложности разрабатываемых и внедряемых технических систем, обеспечивающих протекание ответственных технологических процессов, повышение их производительности, наделение развитым функционалом, в том числе искусственным интеллектом, требуют особого внимания к надежности и безопасности работы. В этой связи важнейшим аспектом является наделение устройств возможностью идентификации неисправностей в процессе выполнения ими своих функций.

При разработке дискретных систем и устройств управления их структуры наделяют различными свойствами, позволяющими с легкостью обеспечивать обнаружение возникающих неисправностей. Существуют методы контролепригодного проектирования устройств [1], а также разработки для них самопроверяемых структур [2]. Это для любых систем критического применения, в том числе для таких, входные данные в которых меняются редко (к ним можно отнести системы управления в атомной промышленности, установки электрической централизации стрелок и сигналов на железнодорожном транспорте, комплексы противовоздушной обороны и т.д.). В таких системах не должно возникать «скрытых» неисправностей и накопления ошибок, поскольку так или иначе это с течением времени приведет к нарушению свойства готовности реализации алгоритма управления [3, 4].

Для своевременного обнаружения неисправностей в дискретных устройствах требуется их самопроверяемая реализация. Это делается различными методами с внесением избыточности в их структуры по определенным алгоритмам, подразумевающим, например, кодирование состояний и использование самопроверяемых схем встроенного контроля (СВК) [5].

Одним из действенных подходов к реализации самопроверяемых СВК является использование так называемого инвертирования данных, когда все входные и выходные сигналы представляются не в виде постоянных значений 0 и 1, а в виде последовательностей 0101...01 и 1010...10 [6]. Такой подход к реализации самопроверяемых дискретных устройств связан с использованием не только аппаратной избыточности, но и временной. Он подразумевает организацию контроля вычислений по определенному диагностическому признаку, учитывающему постоянную смену входных сигналов и подачу наборов значений аргументов парами – рабочего и контрольного наборов. Особенности, которые можно учесть при организации диагностического обеспечения в таком режиме функционирования, обладают самодвойственными и «близкие» к ним булевы функции [7]. Вопросам реализации самодвойственных дискретных устройств посвящено некоторое (небольшое) количество научных работ, например [6, 8–11]. Также известны три монографии, освещающие базовые сведения из теории синтеза самодвойственных дискретных устройств [12–14].

Для контроля вычислений по признаку самодвойственности требуется самодвойственная реализация дискретных устройств, подразумевающая такую организацию их структур, которая позволяет описать все их выходы самодвойственными булевыми функциями. Далеко не все дискретные устройства являются самодвойственными. Можно привести здесь несколько примеров: полный сумматор, мажоритарный элемент, кодеры некоторых разделимых блоковых кодов и др. [6]. Однако учитывая то, что число самодвойственных функций от  $t$  переменных равно  $2^{2^t-1}$ , существует возможность построения  $C_{2^{2^t-1}}^m$  комбинационных устройств с  $m \leq 2^{2^t-1}$  выходами. Например, при  $t = 4$  и  $m = 4$  можно построить 174 792 640 различных самодвойственных комбинационных дискретных устройств. Кроме того, любую структуру комбинационного дискретного устройства можно преобразовать в самодвойственную с использованием только одной переменной на основании известного разложения Шеннона по одной переменной [15]. Для самодвойственной реализации устройств с памятью достаточно комбинационные части сделать самодвойственными [6]. Особенности построения самодвойственных дискретных устройств хорошо изучены, например, в [9].

В настоящей статье сфокусируем внимание читателя на вопросах организации контроля вычислений на выходах самодвойственных дискретных устройств, применяя не только диагностический признак самодвойственности сигналов, но и ряд дополнительных признаков. Такой подход к построению самопроверяемого устройства, как показывают многочисленные эксперименты, позволяет повышать показатели контролепригодности в части наблюдаемости ошибок при использовании аппаратной избыточности меньшей, чем при известном методе дублирования [16]. Это достигается за счет использования свойств некоторых помехозащищенных кодов, обладающих самодвойственными и «близкими» к ним функциями, описывающими их проверочные символы [17]. В качестве таких кодов в статье рассматривается широкий класс полиномиальных, или алгебраических, кодов [18].

## 2. Полиномиальные коды и их особенности

### 2.1. Принципы построения полиномиальных кодов

Полиномиальные коды строятся на основании деления двоичных полиномов, соответствующих информационным сообщениям, с остатком. При делении с остатком различных полиномов степени  $q$  на порождающий полином  $G(y)$  получают  $2^q$  остатков от деления. Каждый проверочный символ полиномиального кода, таким образом, описывается логическим выражением, в которое входит только операция суммирования по модулю  $M = 2(\text{XOR})$  со значениями определенных информационных символов. Далее, для простоты изложения, назовем функции, описывающие проверочные символы кодовых слов полиномиальных кодов, *контрольными функциями*.

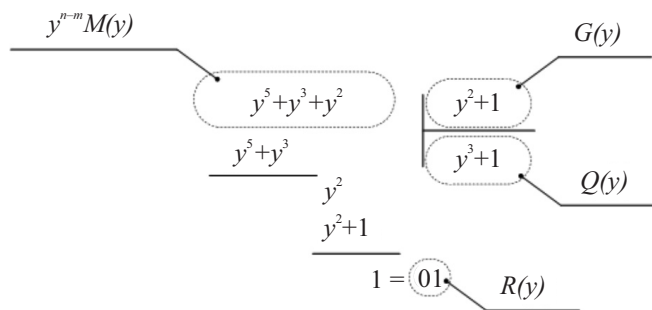


Рис. 1. Пример получения остатка от деления полиномов.

При формировании кодовых слов полиномиальных кодов каждому разряду кодового вектора присваивается переменная  $y$  со степенью  $j$ , которая соответствует расположению разряда: младшему разряду присваивается значение  $j = 0$ , следующему –  $j = 1$  и т.д. Алгебраическая запись кодового вектора получается путем умножения величины  $y^j$  на значение соответствующего разряда. Например, кодовое слово  $\langle 1011 \rangle$  можно представить в виде полинома следующим образом:  $1 \times y^3 + 0 \times y^2 + 1 \times y^1 + 1 \times y^0$ . Удалив нулевые слагаемые, имеем полином вида:  $y^3 + y + 1$ .

Для образования кодового слова  $V(y)$  полиномиального кода определяют остаточный полином  $R(y)$ , получаемый при делении полинома  $M(y)$ , умноженного на величину  $y^{n-m}$ , на порождающий полином [18]:  $y^{n-m}M(y) = G(y)Q(y) + R(y)$ , где  $Q(y)$  – частное, а  $R(y)$  – остаток от деления  $y^{n-m}M(y)$  на  $G(y)$ . В этом случае найденный при делении остаток  $R(y)$  будет представлять контрольный вектор, занимающий  $k$  младших разрядов кодового слова длиной  $n$ . Старшие же  $t$  разрядов будут соответствовать информационному вектору.

Закодируем информационный вектор  $\langle 1011 \rangle$  полиномиальным кодом с порождающим полиномом  $y^2 + 1$ . Степень порождающего полинома равна двум (она всегда должна быть меньше или равна степени информационного полинома). При выполнении операции умножения полином  $y^{n-m}M(y)$  будет иметь следующий вид:  $y^2(y^3 + y + 1) = y^5 + y^3 + y^2$ .

Разделим его на порождающий полином  $y^2 + 1$  (рис. 1).

Полином, соответствующий кодовому слову, получается в виде суммы полиномов  $y^{n-m}M(y)$  и  $R(y)$ :  $y^5 + y^3 + y^2 + 1$ . Таким образом, зашифрованное сообщение примет такой вид:  $\langle 101101 \rangle$  (вместо исходной комбинации  $\langle 1011 \rangle$ ). По сути, информационный вектор был дополнен контрольным  $\langle 01 \rangle$ .

Так как порождающему полиному можно сопоставить двоичное число, будем для сокращения записи использовать десятичный эквивалент  $N$  этого двоичного числа для указания порождающего полинома. Некоторые свойства обнаружения ошибок в СВК на основе полиномиальных кодов с различными значениями  $N$  изучены в [19, 20]. Следует отметить, что порождающий

полином  $G(y)$  выбирается исходя из условия обнаружения максимального количества ошибок. Например, нет смысла в выборе в качестве порождающего полинома  $y^0$ , так как при делении любого полинома с произвольной степенью остаток всегда будет равным нулю (имеется еще ряд полиномов даже с высшей степенью, при делении на которые формируется нулевой остаток, например  $y^2$  или  $y^3$ ). Такие полиномы не позволяют обнаруживать ошибки в кодовом векторе. Далее исключим из рассмотрения порождающие полиномы, не имеющие свободного члена  $y^0$  (числа  $N$ , соответствующие таким полиномам, являются четными). Далее будем рассматривать полиномиальные коды, порождающим полиномам которых соответствуют нечетные числа  $N$ . Только для таких полиномов возможно получение полного множества контрольных векторов с  $k$  разрядами и, соответственно, для них более просто обеспечить полную самопроверяемость кодеров и тестеров данных кодов в СВК [20].

## *2.2. Способ получения логических выражений, описывающих контрольные функции для полиномиальных кодов*

Значения проверочных символов для полиномиальных кодов получают, как правило, с использованием сдвиговых регистров [18], что обуславливается спецификой их применения. Однако существуют некоторые приложения полиномиальных кодов, для которых требуется использование кодеров полиномиальных кодов, являющихся комбинационными схемами. Одним из таких приложений является синтез СВК для дискретных устройств. При построении кодера требуется знать формульную запись, описывающую каждый проверочный символ полиномиального кода, поскольку данное устройство должно быть универсальным и в один такт генерировать по поступающему информационному вектору контрольный вектор.

Чтобы получить формулы, описывающие проверочные символы полиномиального кода, можно использовать тривиальный способ. Он подразумевает для полного множества информационных векторов с числом разрядов  $m$  расчет значений проверочных символов и формирование контрольных векторов, а затем для каждого проверочного символа получение логических выражений. Далее по таким функциям легко синтезируется кодер рассматриваемого полиномиального кода, являющийся комбинационной схемой.

Для получения логических выражений, описывающих контрольные функции, необходимо определить значения всех проверочных символов при подстановке в качестве аргументов булевых векторов, соответствующих информационным векторам кода, из полного множества  $2^m$ . Здесь  $m$  – число аргументов. Однако уже при  $m \geq 6$  это становится достаточно трудоемким. Сократить число информационных векторов при рассмотрении до величины  $m$  можно с использованием следующей теоремы.

*Теорема 1. Для получения функций, описывающих проверочные символы полиномиальных кодов, необходимо и достаточно определить значения*

проверочных символов при подстановке аргументов из информационных векторов с весом  $r = 1$ .

Доказательство теоремы приводится в Приложении.

Из теоремы 1 следует алгоритм определения логических выражений, описывающих контрольные функции полиномиального кода.

**Алгоритм 1.** Правила формирования контрольных функций для полиномиальных кодов:

1. Задаются порождающий полином  $N$  и количество  $m$  разрядов информационного вектора.
2. Определяется число разрядов в контрольных векторах  $k = \lfloor \log_2 N \rfloor$ .
3. Формируется множество  $Q$  информационных векторов с весом  $r = 1$  с мощностью  $|Q| = m$ . Информационные векторы ранжируются по старшинству значащего разряда, и образуется их список.
4. Полагается  $i = 1$ .
5. Рассматривается  $i$ -ый информационный вектор, для которого путем деления определяются значения разрядов контрольного вектора.
6. Составляются формулы для контрольных функций: если значение проверочного символа равно 1, то в формулу включается  $i$ -ый информационный разряд, иначе разряд не включается.
7. Разряды в формулах записываются через знак операции XOR.
8. Полагается  $i := i + 1$ .
9.  $i + 1 > m$ ? Если «да», то переходят к п. 10, если «нет», то переходят к п. 5.
10. Формулы получены.

Пользуясь алгоритмом 1, получим формулы, описывающие контрольные функции рассматриваемого выше полиномиального кода с  $N = 13$  и  $m = 6$ . В табл. 1 приведены все информационные векторы с весом  $r = 1$  для рассматриваемого случая.

По табл. 1 нетрудно для каждой функции  $g_j$  получить формульную запись. Для этого можно поступить следующим образом.

**Таблица 1.** Таблица кодовых слов полиномиального кода с порождающим полиномом, соответствующим числу  $N = 13$  при  $m = 3$ , для информационных векторов с весом  $r = 1$

$m$	$y_6$	$y_5$	$y_4$	$y_3$	$y_2$	$y_1$	$g_3$	$g_2$	$g_1$
6	1	0	0	0	0	0	1	0	1
5	0	1	0	0	0	0	1	1	1
4	0	0	1	0	0	0	1	1	0
3	0	0	0	1	0	0	0	1	1
2	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	1	0	1	0

**Алгоритм 2.** Правила формирования контрольных функций для полиномиальных кодов по таблице кодовых слов для информационных векторов с весом  $r = 1$ :

1. Определяются  $m$  кодовых слов для информационных векторов с весом  $r = 1$ .
2. Полагается  $j = 1$ .
3. Рассматривается  $j$ -ый столбец, соответствующий  $j$ -ой контрольной функции.
4. Записывается формула, в которую включаются те информационные символы  $f_i$ , для которых в  $j$ -ом столбце записана 1.
5. Полагается  $j := j + 1$ .
6.  $j + 1 > k$ ? Если «да», то переходят к п. 7, если «нет», то переходят к п. 3.
7. Формулы получены.

Получаем следующую систему булевых функций:

$$\begin{cases} g_1 = y_3 \oplus y_5 \oplus y_6; \\ g_2 = y_1 \oplus y_3 \oplus y_4 \oplus y_5; \\ g_3 = y_2 \oplus y_4 \oplus y_5 \oplus y_6. \end{cases}$$

Использование алгоритмов 1 и 2 позволяет получить формульные записи для любого полиномиального кода с линейной трудоемкостью. Их можно использовать при синтезе кодеров полиномиальных кодов для применения в составе тестеров в СВК.

### **3. Классификация полиномиальных кодов по виду контрольных функций**

#### *3.1. Особые свойства полиномиальных кодов*

Контрольные функции полиномиальных кодов являются линейными. Хорошо известны следующие особенности линейных булевых функций [17].

*Теорема 2. Линейная булева функция будет самодвойственной только в том случае, если имеет нечетное количество аргументов, от которых она зависит существенно.*

*Теорема 3. Линейная булева функция будет самоквазидвойственной только в том случае, если имеет четное количество аргументов, от которых она зависит существенно.*

Здесь следует напомнить читателю несколько известных определений [7].

**Определение 1.** *Функция принадлежит классу самодвойственных булевых функций, если при инвертировании всех ее аргументов она принимает противоположные значения:*

$$SD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = \overline{f(x_1, x_2, \dots, x_t)}\}.$$

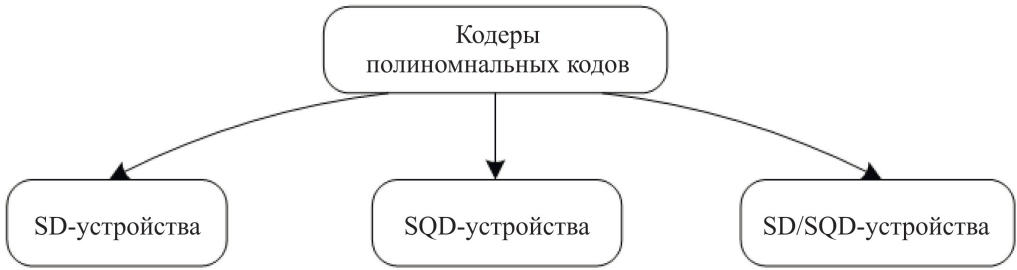


Рис. 2. Классификация кодеров полиномиальных кодов.

*Определение 2. Функция принадлежит классу самоквазидвойственных<sup>1</sup> булевых функций, если при инвертировании всех ее аргументов она принимает такие же значения:*

$$SQD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = f(x_1, x_2, \dots, x_t)\}.$$

Теоремы 2 и 3 позволяют говорить о том, что полиномиальные коды с различными порождающими полиномами  $N$  и с различным количеством  $m$  информационных символов могут принадлежать к трем классам: I – коды, кодеры которых описываются исключительно самодвойственными булевыми функциями, II – коды, кодеры которых описываются исключительно самоквазидвойственными булевыми функциями, III – коды, кодеры которых описываются самодвойственными и самоквазидвойственными булевыми функциями.

*Определение 3. Дискретные устройства, выходы которых описываются исключительно самодвойственными функциями, называются самодвойственными устройствами, или SD-устройствами.*

*Определение 4. Дискретные устройства, выходы которых описываются исключительно самоквазидвойственными функциями, называются самоквазидвойственными устройствами, или SQD-устройствами.*

*Определение 5. Дискретные устройства, часть выходов которых описывается самодвойственными булевыми функциями, а часть – самоквазидвойственными, называются самодвойственно-самоквазидвойственными устройствами, или SD/SQD-устройствами.*

Таким образом, кодеры полиномиальных кодов могут оказаться устройствами только трех видов (рис. 2).

Необходимо отметить следующие важные особенности, присущие кодерам полиномиальных кодов [17].

*Теорема 4. В схемной реализации линейной булевой функции от нечетного числа аргументов, от которых она зависит существенно, при контро-*

<sup>1</sup> Автор [7] использовал для обозначения самоквазидвойственных функций не вполне удачный термин «самоантидвойственные» функции.

ле вычислений по признаку самодвойственности не тестируются неисправности логических элементов, которые связаны путями с четным числом входов.

*Теорема 5. В схемной реализации линейной булевой функции от четного числа аргументов, от которых она зависит существенно, при контроле вычислений по признаку самоквазидвойственности не тестируются неисправности логических элементов, которые связаны путями с четным числом входов.*

Теоремы 4 и 5 говорят о том, что невозможно синтезировать полностью самопроверяемую СВК по признаку самодвойственности и/или самоквазидвойственности контрольных функций, поскольку невозможно обеспечить покрытие всех неисправностей кодеров полиномиальных кодов при контроле только таких признаков. В самом деле, ошибки, вызываемые неисправностями сумматоров по модулю два в линейных схемах, всегда транслируются на их выходы, а при контроле вычислений по признаку самодвойственности или самоквазидвойственности проверяется парафазность пространственного сигнала, а не параллельно поступающего на входы тестера. Поэтому ошибка при соблюдении условий теорем 4 и 5 проявляется при подаче на входы устройства на обоих наборах значений аргументов и не фиксируется тестером самодвойственного или самоквазидвойственного сигнала. Требуется дополнительный контроль по еще одному диагностическому признаку. В качестве такого признака может выступать контроль принадлежности кодовых слов выбранному полиномиальному коду. Это действительно так, потому что на каждом наборе значений аргументов при наличии ошибки будет либо вычислено правильное значение каждого разряда контрольного вектора, либо произойдет его искажение хотя бы в одном разряде. Тестер полиномиального кода зафиксирует это событие. Таким образом, полиномиальные коды можно эффективно использовать в структуре организации контроля вычислений по нескольким диагностическим признакам, о которой речь пойдет далее.

### *3.2. Классы полиномиальных кодов по виду кодера*

Использование свойств линейных функций, а также способа простого получения логических выражений для функций, описывающих проверочные символы полиномиальных кодов, позволило сформировать классификацию кодов по виду их кодеров, используемых в составах тестеров в СВК [21]. Ее фрагмент приведен в табл. 2.

В табл. 2 в первой графе записаны числа  $N$ , соответствующие порождающим полиномам. В таблице приведены все порождающие полиномы для построения полиномиальных кодов с числом проверочных символов  $k = 1, \dots, 6$ .

Для практического использования удобной формой представления различных классов полиномиальных кодов является матричная форма, где отмечается принадлежность кодера конкретного кода с заданным порождающим

**Таблица 2.** Детализированная классификация кодов по видам кодеров

$N$	SD-	SQD-	SD/SQD-
$k = 1$			
3	$m \equiv 1(\text{mod}2)$	$m \equiv 0(\text{mod}2)$	–
$k = 2$			
5	$m \equiv 2(\text{mod}4)$	$m \equiv 0(\text{mod}4)$	$m \equiv 1(\text{mod}2)$
7	$m \equiv 2(\text{mod}3)$	$m \equiv 0(\text{mod}3)$	$m \equiv 1(\text{mod}3)$
$k = 3$			
9	$m \equiv 3(\text{mod}6)$	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 5\} \setminus \{0, 3\}$
11	$m \equiv 2(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 2\}$
13	$m \equiv 6(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 6\}$
15	–	$m \equiv 0(\text{mod}4)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{1, 2, 3\}$
$k = 4$			
17	$m \equiv 4(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 4\}$
19	$m \equiv 3(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 3\}$
21	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, 3, 4, 5\}$
23	$m \equiv 2(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 2\}$
25	$m \equiv 8(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 8\}$
27	$m \equiv 4(\text{mod}12)$	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{0, 1, \dots, 11\} \setminus \{0, 4\}$
29	$m \equiv 6(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 6\}$
31	$m \equiv 4(\text{mod}5)$	$m \equiv 0(\text{mod}5)$	$m \equiv \alpha(\text{mod}5), \alpha \in \{1, 2, 3\}$
$k = 5$			
33	$m \equiv 5(\text{mod}10)$	$m \equiv 0(\text{mod}10)$	$m \equiv \alpha(\text{mod}10), \alpha \in \{0, 1, \dots, 9\} \setminus \{0, 5\}$
35	$m \equiv 2(\text{mod}42)$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 20, 31\}$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 1, \dots, 41\} \setminus \{0, 2, 20, 31\}$
37	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{40, 56\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 38\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 38, 40, 56\}$
39	$m \equiv 1(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 1\}$
41	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{18, 60\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 39, 62\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 18, 39, 60, 62\}$
43	$m \equiv 11(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 11\}$
45	–	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{1, 2, \dots, 11\}$
47	$m \equiv 2(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 2\}$
49	–	$m \equiv 0(\text{mod}21)$	$m \equiv \alpha(\text{mod}21), \alpha \in \{1, 2, \dots, 20\}$
51	$m \equiv 5(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 5\}$
53	$m \equiv 1(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 1\}$
55	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{34, 42\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 10\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 62\} \setminus \{0, 10, 34, 42\}$
57	$m \equiv 17(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 17\}$
59	$m \equiv 18(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 18\}$
61	$m \equiv 22(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 22\}$
63	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, \dots, 5\}$

**Таблица 3.** Матрица полиномиальных кодов с числом проверочных символов  $k = 1, \dots, 4$

$m$	$N$														
	$k = 1$		$k = 2$		$k = 3$				$k = 4$						
	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															

полиномом тому или иному классу при заданном значении числа информационных символов. Такая матрица для кодов с числом проверочных символов  $k \leq 4$  представлена в табл. 3. В ней клетки на пересечении конкретной строки и столбца окрашены в три различных цвета: темно-серый цвет использован для указания на принадлежность кодера полиномиального кода классу SD-устройств, светло-серый цвет говорит о принадлежности кодера полиномиального кода классу SQD-устройств, белый цвет говорит о том, что кодер является SD/SQD-устройством.

#### 4. Использование полиномиальных кодов при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам

##### 4.1. Структура организации схемы встроенного контроля

Рассмотрим особенности использования полиномиальных кодов при синтезе самопроверяемых дискретных устройств с контролем вычислений по нескольким диагностическим признакам. Подход к организации СВК сразу же по нескольким диагностическим признакам использовался ранее в [22] при организации контроля вычислений устройств с применением принципа логической коррекции сигналов и свойств равновесных кодов вида « $r$  из  $2r$ », где  $r$  – вес кодового слова. Также он исследовался в [16] при контроле вычислений по нескольким диагностическим признакам с применением хорошо известных кодов Хэмминга. Структура организации контроля вычислений по нескольким диагностическим признакам изображена на рис. 3.

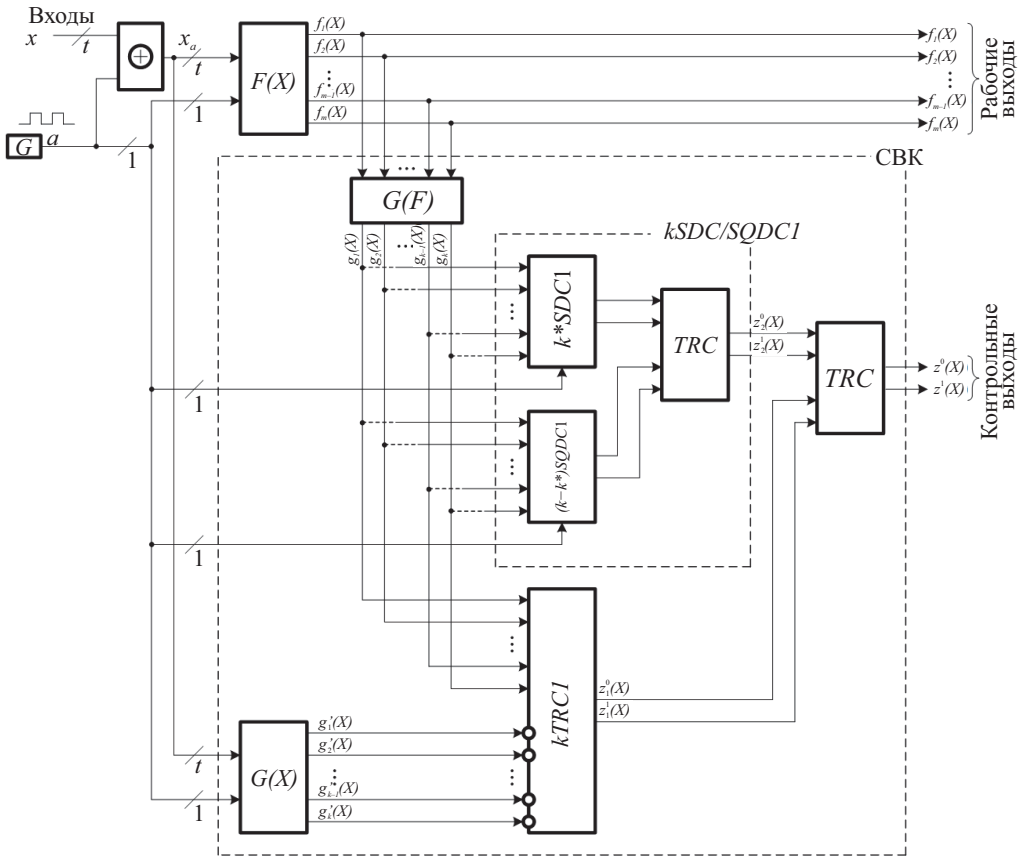


Рис. 3. Структура организации СВК для самодвойственных дискретных устройств по нескольким диагностическим признакам.

В этой структуре исходным дискретным устройством (объектом диагностирования) является блок  $F(X)$ , вычисляющий значения булевых функций  $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$  при подаче на входы наборов значений аргументов  $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$ . Объект диагностирования является SD-устройством. Для контроля вычислений на выходах блока  $F(X)$  используется специализированная СВК по нескольким диагностическим признакам. СВК организуется на основе полиномиальных кодов. Выходы объекта диагностирования отождествляются с информационным вектором с  $m$  разрядами. Блок  $G(F)$  по значениям функций  $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$  формирует значения проверочных символов  $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$  кодовых слов для заданного полиномиального кода, генерируя тем самым конкретный контрольный вектор. С другой стороны, блок  $G(X)$  формирует значения проверочных символов  $g'_1(X), g'_2(X), \dots, g'_{k-1}(X), g'_k(X)$  кодовых слов этого же полиномиального кода при подаче на входы наборов значений аргументов  $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$ . Это позволяет сформировать под схему контроля вычислений по признаку принадлежности формируемых в СВК кодовых слов выбранному полиномиальному коду. Так, одноименные выходы блоков  $G(F)$  и  $G(X)$  подключены ко входам самопроверяемого компаратора  $kTRC1$ , сравнивающего  $k$  пар сигналов на входах и формирующей единственную пару контрольных сигналов  $z_1^0(X)$  и  $z_1^1(X)$ . Самопроверяемый компаратор реализуется в парафазной логике и является схемой сжатия парафазных сигналов, синтезируемой на основе  $k - 1$  элементарных модулей сжатия парафазных сигналов  $TRC$  (*two-rail checker*) [23]. Поэтому сигналы от одного из блоков  $G(F)$  или  $G(X)$  предварительно инвертируются (в структуре рис. 3 инвертированы сигналы от блока  $G(X)$ ). Вообще при синтезе блока  $G(X)$  можно сразу же получать инвертированные сигналы  $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$ , тогда инверторы на входах компаратора не потребуются. Ошибки в вычислениях на выходах блоков  $G(F)$  и  $G(X)$ , а также в структуре самого компаратора приводят к нарушению парафазности сигнала на выходах  $z_1^0(X)$  и  $z_1^1(X)$ , что косвенно свидетельствует о наличии неисправностей в структуре всей системы. Структура организации контроля вычислений именно таким образом – с дополнением информационных символов контрольными – довольно хорошо изучена [24].

Так как выходы блока  $G(F)$  описываются самодвойственными и/или самоквазидвойственными функциями, то для контроля вычислений используется и под схема контроля самодвойственности/самоквазидвойственности сигналов  $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$ . Для контроля самодвойственности сигналов устанавливается блок  $k^*SDC1$ , осуществляющий контроль  $k^*$  самодвойственных сигналов и формирующий одну пару парафазных выходов. Аналогично для контроля самоквазидвойственности сигналов устанавливается блок  $(k - k^*)SQDC1$ , осуществляющий контроль  $k - k^*$  самоквазидвойственных сигналов и формирующий одну пару парафазных выходов. Данные блоки строятся на основе схем сжатия самодвойственных и самоквазидвойственных сигналов [11] и тестеров самодвойственных и самоквазидвойствен-

ных сигналов [25]. Выходы обоих блоков  $k^*SDC1$  и  $(k - k^*)SQDC1$  подключены ко входам элементарного модуля сжатия парафазных сигналов  $TRC$ . Его выходы являются контрольными выходами  $z_2^0(X)$  и  $z_2^1(X)$  подсхемы контроля вычислений по признакам самодвойственности/самоквазидвойственности. Выходы подсхем контроля вычислений по различным диагностическим признакам подключаются ко входам одного модуля  $TRC$ . Его выходы  $z^0(X)$  и  $z^1(X)$  являются контрольными выходами СВК.

Принципы работы и настройки устройств контроля самодвойственных и самоквазидвойственных сигналов описаны авторами настоящей статьи в [25]. Для работы всей структуры используется временная избыточность и режим инвертирования данных [6]. Он подразумевает представление сигналов в виде импульсных последовательностей: логический ноль кодируется последовательностью 0101...01, а логическая единица – 1010...10. Это позволяет реализовать работу устройств при подаче на входы наборов значений аргументов парами: первый набор является рабочим, а второй – контрольным. При этом в паре наборы значений аргументов оказываются ортогональными по всем переменным (инверсными по всем переменным). Данная особенность подхода к организации контроля вычислений требует наличия генератора прямоугольных импульсов  $G$ , формирующего сигнал  $a$  со скважностью 2. Для формирования последовательностей сигналов, соответствующих входным переменным, используют каскад двухвходовых элементов сложения по модулю 2, на первые входы которых подают сигналы от конкретных входов, а на вторые – сигнал  $a$ . Этот же сигнал может подаваться на входы блоков  $F(X)$  и  $G(X)$  при реализации их из несамодвойственных структур с использованием разложения Шеннона по одной переменной. Сигнал  $a$  требуется и для работы тестеров самодвойственности/самоквазидвойственности.

В [16] показано, что структура контроля вычислений по нескольким диагностическим признакам обладает улучшенными показателями контролепригодности в части наблюдаемости ошибок на выходах объектов диагностирования за счет используемого режима работы и возможности разностороннего контроля вычислений.

Отметим следующую важную особенность структуры контроля вычислений по нескольким диагностическим признакам.

*Теорема 6. В структуре СВК с контролем вычислений по нескольким диагностическим признакам обнаруживаются любые одиночные ошибки на выходах подсхемы контроля вычислений по признаку самодвойственности и/или самоквазидвойственности формируемых сигналов.*

Доказательство теоремы приводится в Приложении.

Теорема 6 позволяет на практике упростить задачу построения СВК с покрытием любых ошибок на выходах объекта диагностирования за счет исключения из рассмотрения тех неисправностей объекта диагностирования, которые приводят к возникновению одиночных ошибок на их выходах.

## 4.2. Алгоритм выбора порождающего полинома

Общий алгоритм построения самопроверяемой СВК по нескольким диагностическим признакам с использованием полиномиальных кодов содержит следующие шаги.

*Алгоритм 3.* Правила синтеза схемы встроенного контроля по нескольким диагностическим признакам с использованием полиномиальных кодов:

1. Рассматривается структура комбинационного объекта диагностирования с  $m$  выходами и  $t$  входами.

2. Из полного множества  $\Omega$  неисправностей из выбранной модели, например одиночных константных неисправностей выходов логических элементов, выделяется подмножество  $\Omega' \subset \Omega$  только тех, которые возникают на выходах логических элементов, связанных путями с двумя и более выходами объекта диагностирования.

3. Если  $\Omega' = \emptyset$  то контроль вычислений организуют с помощью полиномиального кода с порождающим полиномом  $N = 3$ , синтезируя СВК по нескольким диагностическим признакам. Если  $\Omega' \neq \emptyset$ , то переходят к рассмотрению возможностей покрытия ошибок с помощью полиномиальных кодов с порождающими полиномами  $N = 3 + 2i$ ,  $i \in \mathbb{N}$ . При этом рассматриваются только такие полиномы, для которых число  $k = \lfloor \log_2 N \rfloor < m$ . Другими словами, максимальное значение числа равно  $N = 2^{m-1} - 1$ , а список содержит числа  $N = 3 + 2i, i = \{1, \dots, 2^{m-2} - 2\}$ .

4. Моделируется работа самодвойственного устройства при подаче на его входы всех пар  $(A_i^j, B_{2^n-1-i}^j)$ , где  $A_i^j$  и  $B_{2^n-1-i}^j$  – значения  $j$ -ой функции, формируемой на  $j$ -ом выходе объекта диагностирования при поступлении входной комбинации, соответствующей десятичному эквиваленту  $i$  и  $2^n - 1 - i$ , наборов значений аргументов и действию одиночных константных неисправностей из множества  $\Omega'$ , а также фиксируются вызываемые ими ошибки на выходах.

5. Образуется список порождающих полиномов  $N = 3 + 2i$ ,  $i = \{1, 2, \dots, 2^{m-2} - 2\}$ , в котором они упорядочиваются по мере возрастания чисел  $N$ . Такое упорядочение необходимо для рассмотрения в первую очередь порождающих полиномов, позволяющих строить коды с наименьшим числом проверочных символов.

6. Выбирается первый член из образованного списка. Он удаляется из списка.

7. На полном множестве наборов значений аргументов определяются значения функций, задающих проверочные символы кодовых слов полиномиального кода с выбранным порождающим полиномом  $N$ .

8. Проверяется покрытие ошибок с помощью контроля вычислений по признаку самодвойственности и/или самоквазидвойственности вычисляемых контрольных функций. Из полного множества  $\Psi$  ошибок, формируемых на выходах схем, исключаются однократные ошибки, а также все ошибки с

кратностями  $d \geq 2$ , покрываемые при контроле вычислений по признаку самодвойственности и/или самоквазидвойственности. Образуется множество  $\Psi' \subset \Psi$  необнаруживаемых по данному признаку ошибок.

9. Если  $\Psi' = \emptyset$ , то контроль вычислений организуют с помощью полиномиального кода с порождающим полиномом, соответствующим числу  $N$ . Если  $\Psi' \neq \emptyset$ , то осуществляют проверку покрытия ошибок из множества  $\Psi'$  с помощью выбранного полиномиального кода.

10. Если ошибки покрываются все, то синтезируют СВК по выбранному полиномиальному коду, если нет, то выбирают следующий порождающий полином из образованного списка, если он не пустой, и переходят к выполнению п. 4 настоящего алгоритма. Если список пустой, то оказывается невозможным синтез полностью самопроверяемой структуры на основе контроля вычислений по нескольким диагностическим признакам.

Рассмотрим пример организации контроля вычислений с использованием представленного алгоритма для самодвойственного дискретного устройства, описываемого следующей системой булевых функций:

$$\begin{aligned}
 f_1 &= \underbrace{x_1 x_2}_1 \vee \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_2 \bar{x}_4}_3; \\
 f_2 &= \underbrace{x_3 x_4}_4 \vee \underbrace{x_1 x_4}_5 \vee \underbrace{x_1 x_3}_6; \\
 f_3 &= \underbrace{x_1 \bar{x}_2 \bar{x}_3}_7 \vee \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_1 x_2 x_3}_{10}; \\
 f_4 &= \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_2 x_3 \bar{x}_4}_{11} \vee \underbrace{x_1 x_3 x_4}_{12} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{13}; \\
 f_5 &= \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_1 \bar{x}_3}_{14} \vee \underbrace{x_1 \bar{x}_2}_{15} \vee \underbrace{\bar{x}_2 \bar{x}_3 \bar{x}_4}_{16}; \\
 f_6 &= \underbrace{x_1 \bar{x}_3 x_4}_{17} \vee \underbrace{x_1 x_3 \bar{x}_4}_{18} \vee \underbrace{x_1 \bar{x}_2 x_3}_{19} \vee \underbrace{\bar{x}_1 x_3 x_4}_{20} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{21}.
 \end{aligned}$$

На первом шаге алгоритма рассмотрим конкретную реализацию устройства по приведенным выше формулам в виде двухуровневой структуры с учетом того, что инверсии аргументов будем переносить на входы элементов первого каскада схемы. При этом пронумеруем все неповторяющиеся конъюнкции в выражениях. Саму схему здесь приводить не будем ввиду ее громоздкости. Она содержит 21 многоходовой логический элемент И на первом каскаде и 6 многоходовых элементов ИЛИ на втором каскаде.

На втором и третьем шагах определим подмножество одиночных константных неисправностей логических элементов, вызывающих кратные ошибки на выходах устройства, а также возможности покрытия ошибок с помощью полиномиального кода с порождающим полиномом  $N = 3$ . Кратные ошибки на выходах устройства могут вызывать только неисправности элементов, реализующих конъюнкции с номерами 2 (могут одновременно искажаться значе-

ния функций  $f_1$  и  $f_5$ ), 8 (могут одновременно искажаться значения функций  $f_3$  и  $f_4$ ) и 9 (могут одновременно искажаться значения функций  $f_3$  и  $f_4$ ). Полиномиальный код с порождающим полиномом  $N = 3$  не подходит для контроля вычислений в рассматриваемом случае.

На четвертом шаге моделируем работу устройства в условиях действия одиночных константных неисправностей и запоминаем полученные результаты, храня их в табличной форме.

Образуем список порождающих полиномов  $N = 3 + 2i$ ,  $i = \{1, \dots, 14\}$ .

Выбираем полином, соответствующий номеру  $N = 5$ . Удаляем его из списка. Проверяем возможности покрытия ошибок на выходах устройства.

Расчеты показали, что полиномиальным кодом с выбранным порождающим полиномом покрываются все кратные ошибки, вызываемые неисправностями элементов, реализующих конъюнкции с номерами 2, 8 и 9, кроме пяти двукратных ошибок, вызываемых неисправностью вида константа 1 элемента, реализующего конъюнкцию с номером 2 (эти пять неисправностей не обнаруживаются ни при контроле самодвойственности функций (см. матрицу в табл. 3), описывающих проверочные символы кодовых слов, ни при контроле по принадлежности контрольного вектора данному полиномиальному коду). Мощность множества  $|\Psi'| \neq 5$ . А это значит, что требуется выбрать следующий полином из сокращенного списка. Этот полином соответствует числу  $N = 7$ .

Выполняем те же действия, что и ранее. Здесь уже, судя по матрице в табл. 3, осуществляется контроль вычислений по признаку принадлежности функций, описывающих проверочные символы кодовых слов, классу самоквазидвойственных, а также контроль принадлежности контрольного вектора кодовым словам выбранного полиномиального кода. Расчеты показали, что все кратные ошибки покрываются с помощью контроля вычислений только по одному из диагностических признаков. Далее синтезируем СВК по двум диагностическим признакам с использованием выбранного полиномиального кода.

Следует отметить, что контроль вычислений по признакам самодвойственности и самоквазидвойственности не приводит к существенному усложнению СВК относительно СВК с контролем вычислений по полиномиальному коду, так как структуры схем сжатия самодвойственных/самоквазидвойственных сигналов, а также тестеры данных сигналов довольно просты [11, 25]. Поэтому удается синтезировать менее избыточные структуры, чем, например, при дублировании, при этом получая улучшенные показатели контролепригодности в части наблюдаемости ошибок на выходах СВК.

В качестве достоинств алгоритма 3 следует выделить то, что за счет выбора порождающего полинома от малых значений  $N = 3$  до  $N = 2^{m-2} - 2$  можно добиться полного покрытия ошибок с помощью кодов с малыми значениями  $k$ , что влияет и на показатели эффективности реализации СВК. Недостатком же является необходимость моделирования неисправностей и

фиксации ошибок на выходах объектов диагностирования, что ограничивает возможности применения метода при достаточно большом числе входов. При этом применимость метода для организации СВК для тех или иных дискретных устройств также определяется и сложностью вычисляемых на их выходах булевых функций [26]. Трудоемкость алгоритма построения самопроверяемого устройства может быть асимптотически оценена величиной  $2^{O(t)}$ , т.е. задача решается за экспоненциальное время с линейной экспонентой. Тем не менее всегда можно использовать методы декомпозиции при синтезе СВК. Кроме того, учет структур объектов диагностирования является одним из действенных методов синтеза СВК, когда не используются стандартные методы, основанные на внесении модульной избыточности.

Также следует отметить, что полиномиальные коды обнаруживают большое количество ошибок при высоких степенях полиномов  $M(y)$  и  $G(y)$ , что широко используется при контроле вычислений и передаче данных. Однако в рассматриваемом приложении полиномиальных кодов степень полинома  $M(y)$  определяется числом выходов объекта диагностирования, а вот степени образующих полиномов подбираются с учетом покрытия ошибок, вызываемых неисправностями в структурах объектов диагностирования. По сути, чем раньше будет найдено число  $N$  по алгоритму 3, тем меньше будет степень у остатка  $R(y)$  и тем меньшую структурную избыточность будет иметь конечное самопроверяемое дискретное устройство.

## 5. Заключение

Предложенный в статье метод синтеза полностью самопроверяемых дискретных устройств позволяет наделять их не только повышенными показателями контролепригодности относительно широко известных методов, но и обеспечивать обнаружение некоторого подмножества ошибок в вычислениях при условии отсутствия изменения входных воздействий. Это связано с представлением сигналов в дискретных устройствах в виде двоичных последовательностей.

Разработанный алгоритм синтеза СВК на основе свойств полиномиальных кодов подразумевает последовательный перебор порождающих полиномов от самого малого, соответствующего числу  $N = 3$ , к соответствующему числу  $N = 2^{m-2} - 2$ . При малых значениях  $m < 6$  количество порождающих полиномов, удовлетворяющих этому условию, будет небольшим, что снизит потенциальные возможности построения полностью самопроверяемого устройства. Поэтому наиболее эффективным рассматриваемый подход должен оказаться для устройств с относительно большим числом выходов  $m \geq 6$ . С ростом значения  $N$  до предельного числа количество проверяющих символов в кодовых словах меняется от величины  $k = 1$  к величине  $k = m - 1$ , что обеспечивает и увеличение мощности множества покрываемых полиномиальными кодами ошибок. Поэтому алгоритм будет эффективным при построении самопроверяемых устройств практически для любых структур объектов диагностиро-

вания с  $m \geq 6$ . Тем не менее для каждого конкретного дискретного устройства потребуется сравнение показателей их самопроверяемой реализации с показателями самопроверяемых реализаций, получаемых по иным методам, например по методу дублирования.

Возможные направления дальнейших исследований могут быть связаны с постановкой разнообразных экспериментов с тестовыми дискретными устройствами и оценкой показателей эффективности их самопроверяемых реализаций, разработкой иных алгоритмов выбора порождающих полиномов с учетом особенностей структур объектов диагностирования, учетом особенностей иных линейных кодов при синтезе СВК на основе нескольких диагностических признаков, а также с изучением особенностей построения подобных устройств на различной, в том числе программируемой, элементной базе.

Использование при построении самопроверяемых дискретных устройств контроля вычислений по нескольким диагностическим признакам представляется авторам перспективным и до конца не исследованным подходом к реализации составляющих высоконадежных систем управления.

## ПРИЛОЖЕНИЕ

### *Доказательство теоремы 1.*

*Необходимость.* Каждая контрольная функция описывается выражением вида  $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$ ,  $j \in \{1, k\}$ ,  $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$ . Для получения выражений, описывающих каждую контрольную функцию, необходимо установить, какие из информационных символов являются значимыми. При рассмотрении всех информационных векторов с весом  $r = 1$  последовательно определяются значения проверочных символов при подстановке единственного значимого информационного символа. Если значение функции  $g_j = 0$ , то соответствующий информационный символ не влияет на конкретную контрольную функцию, иначе, при  $g_j = 1$ , он включается в логическое выражение, описывающее контрольную функцию.

*Достаточность.* При подстановке информационного вектора с весом  $r = 1$  значение функции  $g_j$  будет определяться только значением значащего информационного символа, следовательно, он будет входить в формулу описания соответствующей контрольной функции. При рассмотрении всех информационных векторов с весом  $r = 1$  будут определены все суммируемые в каждой функции  $g_j$  аргументы. Это позволяет определить формулы, описывающие проверочные символы полиномиальных кодов.

### *Доказательство теоремы 6.*

Все функции кодера полиномиального кода являются линейными:  $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$ ,  $j \in \{1, k\}$ ,  $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$ . Рассмотрим одну из них. Положим, происходит одиночная ошибка на выходе  $f_q \in \{1, \dots, m\}$ . Тогда значение функции  $f_p$  искажается, т.е. становится равным  $f_p = 0$  или  $f_p = 1$ . Выражение для контрольной функции выглядело так:  $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_p \oplus \dots \oplus f_{i_q}$ . При искажении оно получило вид:  $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus$

$\oplus(f_p = 0) \oplus \dots \oplus f_{i_q} = \frac{f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}}{\oplus(f_p = 1) \oplus \dots \oplus f_{i_q} = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}}$  или же  $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$ . В первом случае имеем изменение четности числа аргументов исходной функции и зависимости ее от всех переменных, кроме  $f_p$ . Во втором случае четность числа аргументов также меняется, но при этом еще реализуется инверсия функции сложения по модулю 2, сохраняющая все признаки принадлежности к классам самодвойственных и/или самоквазидвойственных функций. Другими словами, функция поменяла свой вид с самоквазидвойственной/самодвойственной на самодвойственную/самоквазидвойственную. Раз вид функции поменялся, то рассматриваемая ошибка обнаружится в подсхеме контроля вычислений по выбранному признаку, так как он изменится при одиночной ошибке.

### СПИСОК ЛИТЕРАТУРЫ

1. *Bennetts R.G.* Design of Testable Logic Circuits. Addison-Wesley Publishing Company, 1984. 164 p.
2. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
3. *Drozd A., Kharchenko V., Antoshchuk S., et. al.* Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions // Proceedings of 9th IEEE East-West Design and Test Symposium (EWDTS'2011), Sevastopol, Ukraine, 2011. P. 411-416. <https://doi.org/10.1109/EWDTS.2011.6116606>
4. *Дрозд А.В., Харченко В.С., Антощук С.Г. и др.* Рабочее диагностирование безопасных информационно-управляющих систем. Под ред. А.В. Дрозда, В.С. Харченко. Харьков: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «ХАИ», 2012. 614 с.
5. *Сапожников В.В., Сапожников Вл.В.* Самопроверяемые дискретные устройства. СПб.: Энергоатомиздат, 1992. 224 с.
6. *Аксёнова Г.П.* Восстановление в дублированных устройствах методом инвертирования данных // АИТ. 1987. № 10. С. 144–153.
7. *Шальто А.А.* Модули, универсальные в классе самодвойственных функций и в «близких» к ним классах // Известия Российской Академии наук наук. Теория и системы управления. 2001. № 5. С. 110–120.
8. *Saposhnikov Vl.V., Dmitriev A., Goessel M., Saposhnikov V.V.* Self-Dual Parity Checking a New Method for on Line Testing // Proceedings of 14th IEEE VLSI Test Symposium, USA, Princeton, 1996. P. 162–168. <https://doi.org/10.1109/VTEST.1996.510852>
9. *Гессель М., Мошанин В.И., Сапожников В.В., Сапожников Вл.В.* Обнаружение неисправностей в самопроверяемых комбинационных схемах с использованием свойств самодвойственных функций // АИТ. 1997. № 12. С. 193–200.
10. *Гессель М., Дмитриев А.В., Сапожников В.В., Сапожников Вл.В.* Самотестируемая структура для функционального обнаружения отказов в комбинационных схемах // АИТ. 1999. № 11. С. 162–174.
11. *Dmitriev A., Saposhnikov V., Saposhnikov V., Goessel M.* New Self-Dual Circuits for Error Detection and Testing // VLSI Design. 2000. Vol. 11. Issue 1. P. 1–21. <https://doi.org/10.1155/2000/84720>

12. Сапожников В.В., Сапожников Вл.В., Гессель М. Самодвойственные дискретные устройства. СПб.: Энергоатомиздат (Санкт-Петербургское отделение), 2001. 331 с.
13. Сапожников В.В., Сапожников Вл.В., Валиев Р.Ш. Синтез самодвойственных дискретных систем. СПб.: Элмор, 2006, 220 с.
14. Goessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
15. Яблонский С.В. Введение в дискретную математику. Под ред. В.А. Садовничева. М.: Высш. шк. 2003. 384 с.
16. Ефанов Д.В., Погодина Т.С. Исследование свойств самодвойственных комбинационных устройств с контролем вычислений на основе кодов Хэмминга // Информатика и автоматизация. 2023. Том 22. № 2. С. 349–392. <https://doi.org/10.15622/ia.22.2.5>
17. Ефанов Д.В. Особенности реализации самопроверяемых структур на основе метода инвертирования данных и линейных кодов // Вест. Том. гос. ун-та. Управление, вычислительная техника и информатика. 2023. № 65. С. 126–138. <https://doi.org/10.17223/19988605/65/13>
18. Sellers F.F., Hsiao M.-Y., Bearson L.W. Error Detecting Logic for Digital Computers. New York: McGraw-Hill, 1968. XXI+295 p.
19. Efanov D., Plotnikov D., Sapozhnikov V., Sapozhnikov Vl., Abdullaev R. Experimental Studies of Polynomial Codes in Concurrent Error Detection Systems of Combinational Logical Circuits // Proceedings of 16th IEEE East-West Design and Test Symposium (EWDTS'2018), Kazan, Russia, September 14–17, 2018. P. 184–190. <https://doi.org/10.1109/EWDTS.2018.8524684>
20. Abdullaev R., Efanov D. Polynomial Codes Properties Application in Concurrent Error-Detection Systems of Combinational Logic Devices // Proceedings of 19th IEEE East-West Design and Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021, pp. 40–46. <https://doi.org/10.1109/EWDTS52692.2021.9580992>
21. Ефанов Д.В., Пивоваров Д.В. Организация контроля вычислений на выходах самодвойственных цифровых устройств с применением циклических избыточных кодов // Матер. XIV Всероссийского совещания по проблемам управления (ВСПУ-2024) под ред. акад. Д.А. Новикова, М., 17–20 июня 2024 г., с. 2395–2399.
22. Efanov D.V., Pivovarov D.V. The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters // Proceedings of 19th IEEE East-West Design and Test Symposium (EWDTS'2021), Batumi, Georgia, September 10–13, 2021, с. 200–206. <https://doi.org/10.1109/EWDTS52692.2021.9581019>
23. Lala P.K. Self-Checking and Fault-Tolerant Digital Design. San Francisco: Morgan Kaufmann Publishers, 2001. 216 p.
24. Sahana A.R., Chiraag V., Suresh G., Thejaswini P., Nandi S. Application of Error Detection and Correction Techniques to Self-Checking VLSI Systems: An Overview // Proceedings of 2023 IEEE Guwahati Subsection Conference (GCON), Guwahati, 2023. <https://doi.org/10.1109/GCON58516.2023.10183449>

25. Ефанов Д.В., Пивоваров Д.В. Тестеры самодвойственных и «близких» к ним сигналов // Известия высших учебных заведений. Приборостроение. 2024. Т. 67. № 1. С. 5–19. <https://doi.org/10.17586/0021-3454-2024-67-1-5-19>
26. Коршунов А.Д. Сложность вычислений булевых функций // Успехи математических наук. 2012. Том 67. № 1 (403). С. 97–168. <https://doi.org/10.4213/rm9459>

*Статья представлена к публикации членом редколлегии Л.Ю. Филимонюком.*

Поступила в редакцию 28.11.2024

После доработки 17.03.2025

Принята к публикации 27.03.2025

# Интеллектуальные системы управления, анализ данных

© 2025 г. А.Н. ВАРНАВСКИЙ, канд. техн. наук (avarnavsky@hse.ru)  
(Национальный исследовательский университет  
“Высшая школа экономики”, Москва)

## РЕГРЕССИОННЫЕ МОДЕЛИ ДЛЯ ИГРОВОГО ЧАТ-БОТА С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ГОЛОВОЛОМОК ТИПА WORDLE

Программирование является одним из важнейших навыков XXI в. Однако для многих учащихся обучение программированию является довольно сложным процессом. В таких случаях важно поддерживать интерес и вовлеченность студентов в процесс обучения. Считается, что цифровые игры могут решить эту проблему. Одним из видов игр, которые хорошо подходят для сферы компьютерных наук, являются головоломки, которые направлены в том числе и на развитие когнитивных способностей. Целью статьи является разработка моделей, алгоритма работы и структуры игрового чат-бота с искусственным интеллектом для обучения программированию с помощью заданий-головоломок по типу словесной игры Wordle. Wordle выбрана по причине ее всемирной популярности и адаптирована в виде игрового чат-бота для использования в процессе обучения программированию. Искусственный интеллект в чат-боте необходим для контроля целесообразности и подходящего времени его использования, а также адаптивного формирования уровня сложности заданий-головоломок. На основе собранных в результате использования неинтеллектуального игрового чат-бота данных были построены регрессионные модели влияния показателей студентов на уровень интереса и сложности предлагаемых игровым чат-ботом заданий-головоломок. Разработанные модели легли в основу алгоритма работы и структуры игрового чат-бота с искусственным интеллектом. При использовании интеллектуального игрового чат-бота есть возможность дообучать модели и корректировать полученные ранее значения коэффициентов.

*Ключевые слова:* игровой чат-бот, обучение программированию, искусственный интеллект.

**DOI:** 10.31857/S0005231025050049, **EDN:** АХРНОК

### 1. Введение

Для многих студентов обучение программированию является довольно сложным процессом. Это связано с тем, что решение задач программиро-

вания требует не только наличия знаний в области языков программирования, но и соответствующих уровней когнитивных способностей (логического, критического, вычислительного мышлений). Невысокие уровни таких способностей зачастую создают трудности в обучении и приводят к снижению вовлеченности и мотивации учащихся.

Использование обучающих компьютерных игр является одним из методов поддержания интереса учащихся к обучению в различных дисциплинах и сферах [1], в том числе в программировании [2]. Когнитивные способности и функции (рабочая память, скорость обработки, исполнительные функции и т.п.) могут значительно улучшиться при использовании игр [3], причем конкретный эффект зависит от видов используемых игр [4]. Так, для обучения компьютерным наукам хорошо подходят логические игры и игры-головоломки [5].

Традиционно для обучения программированию на начальном этапе хорошо подходят головоломки Парсонса (Parsons programming puzzles) [6], которые представляют собой фрагменты кода, перемешанные в случайном порядке, и которые обучаемый должен расположить в правильной последовательности для получения работающей программы. Они эффективны для освоения базовых синтаксических конструкций и логики программирования, но их возможности ограничены и они недостаточно ориентированы на развитие более сложных когнитивных навыков.

В то же время словесные головоломки, которые широко используются в обучении иностранным языкам и медицине, практически не применяются в обучении программированию, несмотря на их огромный потенциал для развития когнитивных способностей [7]: словарного запаса, рабочей памяти, когнитивной обработки, стратегий извлечения информации из рабочей памяти, способностей к рассуждениям и мышлению [8].

Эффективное программирование напрямую зависит от знания синтаксиса языка программирования, словарного запаса, относящегося к специфической терминологии, синтаксических правил [9]. Даже незначительное отклонение от синтаксиса может привести к ошибкам компиляции или выполнения, требующим значительных временных затрат на отладку. Более того, нечеткое и неточное использование терминологии в коде значительно затрудняет его понимание как самим автором, так и другими разработчиками, увеличивая стоимость и время, затрачиваемые на поддержку и модернизацию программного обеспечения [10]. Это особенно критично в крупных проектах, где код пишется и поддерживается множеством людей и неясная лексика может привести к несовместимости кода, потере времени и даже к возникновению ошибок, ведущих к сбою всей системы. Учитывая потенциал словесных головоломок, следует развивать направление их использования в обучении программированию и компьютерных науках. Причем необходим поиск вариантов воплощения словесных головоломок, которые были бы интересными и увлекательными для студентов разного уровня, развивая при этом их навыки.

Одним из подходов к созданию образовательных игр является адаптация известных компьютерных игр (например, адаптация Super Mario в [11]). В последние годы достаточно популярной словесной игрой является Wordle (<https://www.nytimes.com/games/wordle/index.html>). Wordle – это словесная игра-головоломка, в которой игроки за несколько попыток должны угадать слово. Появляются научные работы, исследующие стратегии, процесс решения головоломок, сложности слов, поведения игроков в этой игре [12–14]. Поэтому принцип игры Wordle можно адаптировать для создания игры-головоломки по распознаванию программного кода для использования в процессе обучения программированию. Учитывая рост популярности чат-ботов в различных сферах, в том числе в сфере образования [15], а также текстовый формат игры Wordle, адаптацию можно реализовать в виде чат-бота.

В игровой чат-бот с заданиями-головоломками можно заложить модели, которые сделают его интеллектуальным, например в части формирования заданий с соответствующим уровнем сложности и интереса. Это позволит повысить эффективность его использования, в частности за счет того, что интерес влияет на вовлеченность и мотивацию, а сложность заданий может влиять на интерес. В литературе подобных игровых интеллектуальных чат-ботов найдено не было.

Целью статьи является разработка моделей, алгоритма работы и структуры игрового чат-бота с искусственным интеллектом для обучения программированию с помощью заданий-головоломок по типу Wordle. Для достижения поставленной цели нужно собрать данные использования первой версии неинтеллектуального игрового чат-бота, на основе которых можно построить регрессионные модели. Эти модели лягут в основу алгоритма работы и структуры новой версии игрового чат-бота.

## 2. Обзор литературы

### *2.1. Использование образовательных игр*

Несмотря на то, что в некоторых исследованиях не было замечано положительного эффекта от использования образовательных игр, большинство ученых считают игры очень эффективным и недорогим методом обучения, повышающим интерес к дисциплине, вовлеченность студентов, показатели успеваемости и оценки знаний после использования игр [16, 17]. Игры могут влиять на различные аспекты, включая аффективные и мотивационные элементы, изменения поведения, усвоение знаний, а также развитие когнитивных и социальных навыков [18].

Существуют компьютерные игры для обучения информатике и программированию, причем их количество в последние годы постоянно растет [5, 19–21]. Есть игры для разных ступеней образования, хотя преобладают игры для обучения программированию новичков и школьников. Среди существующих игр отдельное направление отводится головоломкам и логическим играм, поскольку они хорошо подходят для обучения компьютерным наукам

и позволяют развивать мыслительные навыки и логическое мышление [22]. Это проявляется в улучшении таких показателей, как скорость обработки информации, объем рабочей памяти, а также в когнитивном контроле, аргументации и решении проблем.

Словесные головоломки являются довольно популярным типом игр, применяющихся в образовательной практике, особенно в обучении иностранным языкам и медицине. Их интеграция с учебными курсами, в сочетании с другими инструментами обучения, способствует снижению когнитивной нагрузки на учащихся, стимулирует мотивацию, повышение академической успеваемости [23], эффективно развивает когнитивные способности [7, 24] и общие знания [25].

В последние годы одной из самых популярных словесных головоломок является игра Wordle. Эта игра предназначена для развлечения, однако можно отметить появление достаточно большого количества научных работ, исследующих эту игру, процесс решения, варианты перебора и поиска решения и т.п. [12–14]. Поэтому можно рассмотреть направление адаптации этой игры применительно к учебному процессу, в частности для применения при обучении программированию.

## *2.2. Чат-боты с искусственным интеллектом*

Реализовать игры-головоломки можно в различных формах, в том числе в чат-боте. Такой формат может хорошо подойти для некоторых текстовых/словесных головоломок. Базовые чат-боты используют подход, основанный на правилах [26]. В отличие от них, чат-боты с искусственным интеллектом (ИИ) используют модели машинного обучения.

Чат-боты с ИИ являются одним из типов приложений диалогового ИИ. Согласно прогнозам (<https://www.marketsandmarkets.com>), ожидается совокупный годовой темп роста размера мирового рынка диалогового ИИ, равный 21,8%, а рынок к 2026 г. увеличится до 18,4 млрд долларов США. При этом диалоговый ИИ может стать основной технологией практически во всех рыночных вертикалях (<https://www.emarketer.com/content/tech-catches-up-conversational-ai-ambitions>).

В целом преимуществами чат-ботов с ИИ является обеспечение удобного интерфейса на естественном языке, что дает возможность улучшить качество работы пользователей [27]. Использование чат-ботов с ИИ может снизить стресс и помочь пользователям избежать информационной перегрузки [28], позволит получить пользователям уникальный и более персонализированный опыт. Такие чат-боты влияют на эмоции пользователей в отличие от использования обычных корпоративных систем [29].

Диалоговые системы искусственного интеллекта могут проявлять индивидуальность и обладать антропоморфными характеристиками [30], лучше соответствуют пользовательскому и организационному контексту, способны постоянно развиваться за счет обучения на содержании запросов и способах

взаимодействия пользователей с ними. Поэтому постоянное использование систем с ИИ имеет важное значение для их дальнейшего совершенствования, поскольку в их основе, как правило, лежат технологии машинного обучения. Так как постоянное совершенствование чат-бота с ИИ зависит от его использования, важно учитывать поведение пользователей и задействовать модели использования [31].

В настоящее время развитие чат-ботов с ИИ сосредоточено на развитии методов обучения алгоритмов и подходов, связанных с настройкой, антропоморфизмом [32], персонализацией [33], совершенствованием алгоритмов для получения более точных результатов [34] и расширении функциональных возможностей, включая, например, анализ настроений [35]. Есть предложения по созданию нового человеко-машинного интеллекта [36].

В качестве примеров можно привести чат-боты с ИИ, которые находят применение на рабочем месте [37], в бизнес-процессах [38], в организации эффективных взаимодействий с пользователями и клиентами [39], в обучении [40]. Такие чат-боты более эффективны за счет автоматического выявления потребностей пользователей [41, 42] и формирования поведения, ориентированного на пользователя [43]. Например, в обучении чат-боты с ИИ способны повысить интерес пользователей к изучению математики [44] и чтению [45].

### *2.3. Постановка задачи*

В целом игровых образовательных чат-ботов в литературе описано достаточно мало, игровых образовательных чат-ботов с ИИ еще меньше. Для примера можно привести [46].

Каждый описанный чат-бот имеет свою логическую/сценарную или AI/ML модель поведения. С учетом оригинальности задачи такие модели не подойдут для построения интеллектуального игрового чат-бота с головоломками по типу Wordle и их необходимо разработать. Для этого используем неинтеллектуальный игровой чат-бот и проведем с ним эксперимент. Собранные в результате эксперимента данные используем для создания моделей машинного обучения для управления поведением интеллектуального игрового чат-бота. Модели будут управлять уровнем сложности заданий и оценивать целесообразность использования чат-бота на данной стадии изучения языка программирования через прогнозирование интереса к заданию. Модели могут дообучаться на основе новых данных, полученных при использовании интеллектуального игрового чат-бота. На основе полученных моделей создадим алгоритм работы и структуру игрового чат-бота с ИИ для обучения программированию с помощью заданий-головоломок по типу Wordle.

## **3. Игровой образовательный чат-бот с головоломками по типу игры Wordle**

Создан неинтеллектуальный игровой чат-бот «Распознай команду Python» для мессенджера Telegram с текстовыми заданиями-головоломками

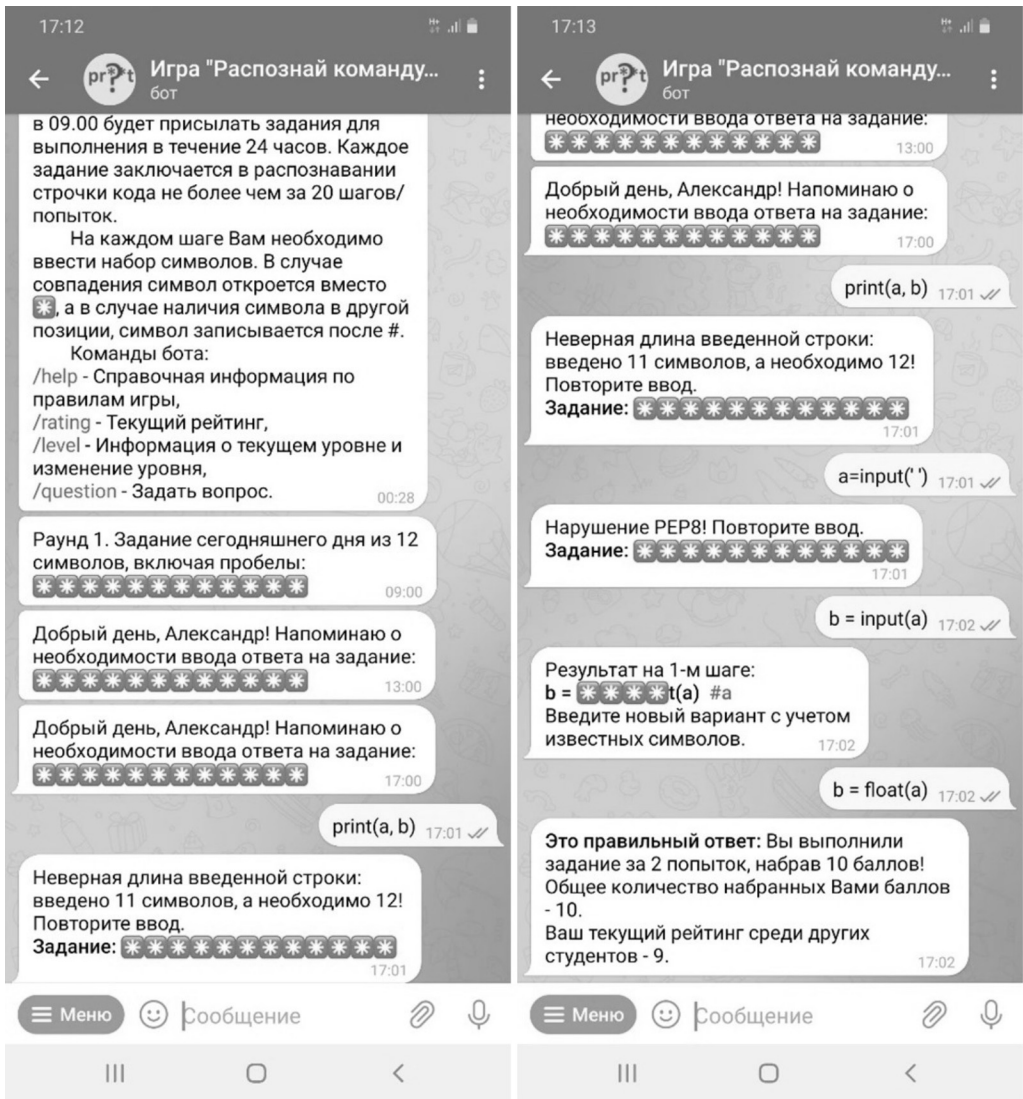


Рис. 1. Работа игрового чат-бота «Распознай команду Python».

по типу игры Wordle [47], который предназначен для обучения текстовому языку программирования, в частности Python (рис. 1), и имеет следующую игровую механику.

1. Цель каждого задания – распознать строчку некоторой программы за несколько шагов-попыток. Изначально взятая строчка программы предъядляется в виде последовательности символов \*, количество которых равно длине строчки (с пробелами).

2. Упрощения при формировании задания осуществляются в отношении названий переменных, значений чисел, строк, списков, словарей и других структур. Так переменные изменяются и задействуются в порядке a, b, c, ...,

числа – в порядке 1, 2, 3, ..., символы и строки – в порядке «a», «b», «c», ..., математические операции – в порядке «+», «-», «\*», списки, кортежи, словари и т.д. упрощаются до одного элемента.

3. Введенный на каждом шаге текст проверяется на соответствие длине искомой строчки программы, синтаксическим правилам языка и правилам PEP8 (в случае Python), а также осуществляется проверка наличия уже открытых и известных символов. В случае наличия нарушений студенту сообщается о несоответствии и некорректности ввода, а в случае отсутствия – выдается строка с открытыми символами, за которой после символа # следует подсказка. Подсказка выдается в виде набора символов, отсортированных в порядке возрастания кода ASCII.

4. Открытые символы и подсказка после каждой попытки формируются на основе результата посимвольного сравнения введенного текста с искомой строчкой программы по тем же правилам, что и в игре Wordle: в случае совпадения символов, т.е. находящихся на правильных местах, вместо \* появляются эти символы, а в случае наличия символов в других позициях, т.е. стоящих не на своих местах, эти символы появляются в подсказке.

5. Максимальное число корректных попыток, которое дается на выполнение задания, равно 20.

6. Используется три уровня заданий разной сложности и разным начислением очков, возможность изменения и перехода между уровнями. При выполнении минимум трех заданий на текущем уровне можно перейти на следующий уровень. С более высокого уровня всегда можно перейти на нижний уровень. При выполнении задания 1-го уровня за 5 и менее шагов начисляется 10 баллов, 2-го уровня – 20 баллов, 3-го уровня – 30 баллов. Если задание выполнено за большее число шагов, то число полученных баллов уменьшается.

7. Число символов в задании зависит от уровня: чем выше уровень, тем больше символов в искомой строчке программы. Часто используемые функции, команды, методы выдаются на 1-ом и 2-ом уровнях, редко используемые – на 3-ем уровне.

8. Ведется рейтинг и после выполнения задания выводится число набранных баллов за текущее задание, общее число баллов и текущая позиция студента в рейтинге.

Геймплей «Распознай команду Python» следующий.

1. Каждый день в 09.00 бот присылает задание. Длительность выполнения задания в течение 24 ч никак не ограничена.

2. Бот отслеживает выполнение задания, введенные попытки и при их отсутствии два раза в день присылает напоминание о необходимости выполнения задания.

3. Ввод текста осуществляется в строке ввода Telegram, какие-либо клавиатуры, помимо стандартной для ввода текста на смартфоне, не используются.

4. Имеется функционал ответа на вопросы студентов.

5. Действия студента фиксируются, осуществляется сбор статистики работы с чат-ботом: определение числа шагов, совершенных при выполнении каждого задания, набранные баллы, изменение уровня.

6. Используется три уровня заданий разной сложности и с разным начислением очков, возможностью изменения и перехода между уровнями. При выполнении минимум трех заданий на текущем уровне можно перейти на следующий уровень. С более высокого уровня всегда можно перейти на нижний уровень. При выполнении задания 1-го уровня за 5 и менее шагов начисляется 10 баллов, 2-го уровня – 20 баллов, 3-го уровня – 30 баллов. Если задание выполнено за большее число шагов, то число полученных баллов уменьшается.

7. Взаимодействие с игровым чат-ботом направлено на поддержку развития *hard skills* (запоминания и вспоминания команд, функций, методов языка программирования), а также когнитивных навыков (рабочей памяти, внимания и логического мышления) [48].

## 4. Разработка моделей

### 4.1. Проведенный эксперимент

В проведенном эксперименте приняли участие 60 студентов первого курса ИТ-направления университета, изучавшие язык программирования Python в рамках двухсеместрового учебного курса. Большая часть студентов на момент начала обучения в университете не имела опыта программирования. Сам эксперимент проходил во втором семестре.

В эксперименте использовался неинтеллектуальный игровой чат-бот «Распознай команду Python» и две анкеты. Оценивание показателей в анкетах осуществлялось по пятибалльной шкале Лайкерта.

Этапы проведения эксперимента.

1. Заполнение первой анкеты и указание самооценки уровня написания программ на Python и сложности учебного курса.

2. Использование чат-бота в течение 10 дней вне аудитории и решение по одной головоломки в день. Автоматический сбор данных о ходе и результатах решения заданий для каждого студента.

3. Заполнение второй анкеты и указание интереса и сложности заданий, задействования рабочей памяти, внимания и логического мышления при решении заданий.

### 4.2. Разработанные модели

Для построения моделей были использованы три источника данных: первая анкета, вторая анкета и статистика работы студентов с игровым чат-ботом.

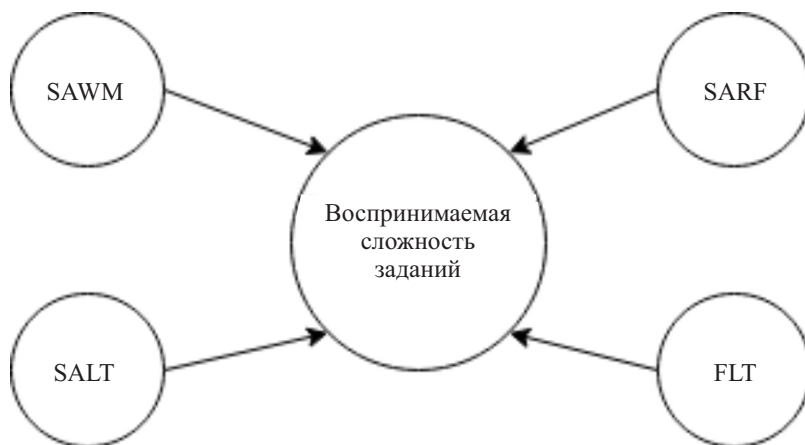


Рис. 2. Гипотеза о влиянии показателей на воспринимаемую сложность заданий.

Из первой анкеты получены показатели:

- Самооценка навыка написания программного кода на языке Python (SAWP);
- Оценка сложности учебного курса по изучению Python (ACD).

Из второй анкеты получены показатели:

- Самооценка задействования рабочей памяти в решении заданий (SAWM);
- Самооценка задействования логического мышления в решении заданий (SALT);
- Самооценка вспоминания функций Python при решении заданий (SARF);
- Воспринимаемая сложность заданий (PDT);
- Интерес к заданиям PИТ.

Из данных статистики работы студентов получены показатели:

- Итоговый уровень заданий в игровом чат-боте (FLT);
- Количество шагов/попыток, затраченных на решение отдельного задания (NS);
- Среднее количество шагов, затраченных на решение заданий (ANS);
- Итоговое количество баллов за выполнение заданий (SRT).

Можно предположить, что на воспринимаемую сложность заданий PDT влияют SAWM, SALT, SARF, FLT (рис. 2). Объяснить такое влияние можно следующим образом. Если студент находился на 2/3-м уровнях, то он сталкивался с более сложными заданиями, по сравнению с теми, кто находился на 1/2-м уровнях и не был на 3-м. Поэтому такие студенты, очевидно, оценят задания более сложными, т.е. FLT влияет на PDT. Однако необходимо учитывать, что если студент не сможет решить задание, находясь на уровне, он, скорее всего, будет переходить на более низкий уровень. Ключевую роль в решении различных задач играет рабочая память и логическое мышле-

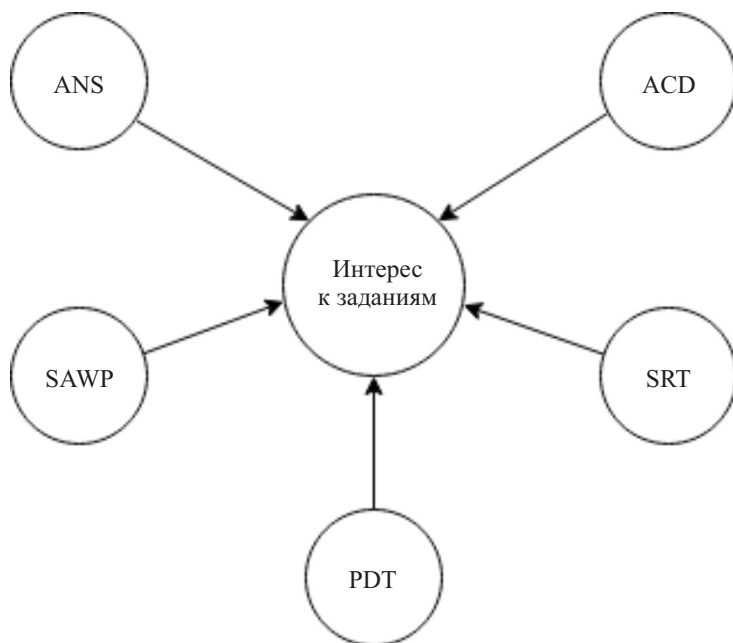


Рис. 3. Гипотеза о влиянии показателей на интерес к заданиям.

ние. Для успешного выполнения заданий требуется вспоминание функций и команд Python и применение логических рассуждений на каждом шаге (попытке) при появлении открытых символов. Рабочая память будет обеспечивать процесс решения заданий, поэтому SAWM будет влиять на PDT. Следует отметить, что простые задачи, в отличие от сложных, не оказывают значительной нагрузки на рабочую память. Помимо рабочей памяти, уровень логического мышления студента и количество функций и команд Python, запомненных студентом в процессе обучения, будут влиять на воспринимаемую сложность заданий. При низком уровне логического мышления и количества запомненных функций студент может испытывать трудности при решении даже задач средней сложности, а сами задания будут восприниматься как сложные. Отсюда и влияние SARF и SALT на PDT [49].

Можно предположить, что на интерес к заданиям PIT влияют ANS, SAWP, PDT, ACD, SRT (рис. 3). Объяснить такое влияние можно следующим образом. Большое количество попыток, затраченных при решении задания, может указывать как на высокую сложность задания, так и на использование перебора. Оба этих варианта негативно сказываются на интересе к выполняемым заданиям. Поэтому ANS влияет на PIT. Для успешного написания кода студенты должны знать и помнить функции языка Python. Студенты, которые высоко оценивают свои навыки программирования, как правило, знают достаточное количество функций и обладают развитым логическим и алгоритмическим мышлением. Поэтому для них решение задач будет более интересным, чем для студентов, которые не уверены в своих способностях

к программированию, не знакомы с достаточным количеством функций и не обладают развитым логическим и алгоритмическим мышлением. Поэтому SAWP влияет на PIT. Если студенты тратят много времени на решение сложных задач, особенно если у них низкий уровень рабочей памяти и логического мышления, это может вызвать некоторый дискомфорт и снизит уровень интереса. Поэтому PDT оказывает влияние на PIT. Использование инструментов геймификации и цифровых игр способно снизить когнитивную нагрузку в сложных учебных курсах. Поэтому ACD влияет на PIT. Чем больше баллов набрал студент в результате работы с игровым чат-ботом, тем больше он был вовлечен и тем интереснее ему было работать с чат-ботом. Таким образом, SRT влияет на PIT [49].

Был проведен регрессионный анализ и построены модели влияния показателей на интерес и воспринимаемую сложность игрового чат-бота. Для оценки важности и вклада каждого параметра в значение зависимой переменной использовалась стандартизация с математическим ожиданием, равным 0, и стандартным отклонением, равным 1. Зависимые переменные не стандартизировались.

Для оценки воспринимаемой сложности заданий игрового чат-бота получена линейная регрессия с уровнем значимости  $p < 0,01$  вида:

$$(1) \quad \begin{aligned} \text{PDT} = & a_{\text{SAWM}} \cdot \text{SAWM} + a_{\text{SARF}} \cdot \text{SARF} + \\ & + a_{\text{SALT}} \cdot \text{SALT} + a_{\text{FLT}} \cdot \text{FLT} + a_0^{\text{PDT}}, \end{aligned}$$

где весовые коэффициенты принимают следующие значения:

$$\begin{aligned} a_{\text{SAWM}} = 0,63^{***}, \quad a_{\text{SARF}} = -0,52^{**}, \quad a_{\text{SALT}} = -0,41^{**}, \quad a_{\text{FLT}} = 0,23^*, \\ a_0^{\text{PDT}} = 3,07^{***} \quad (***) - p < 0,001, \quad (** - p < 0,01, \quad (* - p < 0,1). \end{aligned}$$

Коэффициент детерминации модели составил 0,82 (скорректированный – 0,74).

Для оценки интереса к заданиям игрового чат-бота получена линейная регрессия с уровнем значимости  $p < 0,001$  вида:

$$(2) \quad \begin{aligned} \text{PIT} = & a_{\text{ANS}} \cdot \text{ANS} + a_{\text{SAWP}} \cdot \text{SAWP} + a_{\text{PDT}} \cdot \text{PDT} + \\ & + a_{\text{ACD}} \cdot \text{ACD} + a_{\text{SRT}} \cdot \text{SRT} + a_0^{\text{PIT}}, \end{aligned}$$

где весовые коэффициенты принимают следующие значения:

$$\begin{aligned} a_{\text{ANS}} = -0,68^{**}, \quad a_{\text{SAWP}} = 0,64^{**}, \quad a_{\text{PDT}} = -0,47^*, \\ a_{\text{ACD}} = 0,44^*, \quad a_{\text{SRT}} = 0,27^*, \quad a_0^{\text{PIT}} = 3,73^{***}. \end{aligned}$$

Коэффициент детерминации модели составил 0,86 (скорректированный – 0,78).

В модели (1) показатели SAWM, SALT, SARF на небольшом промежутке времени мало меняются. Показатель FLT характеризует итоговый уровень

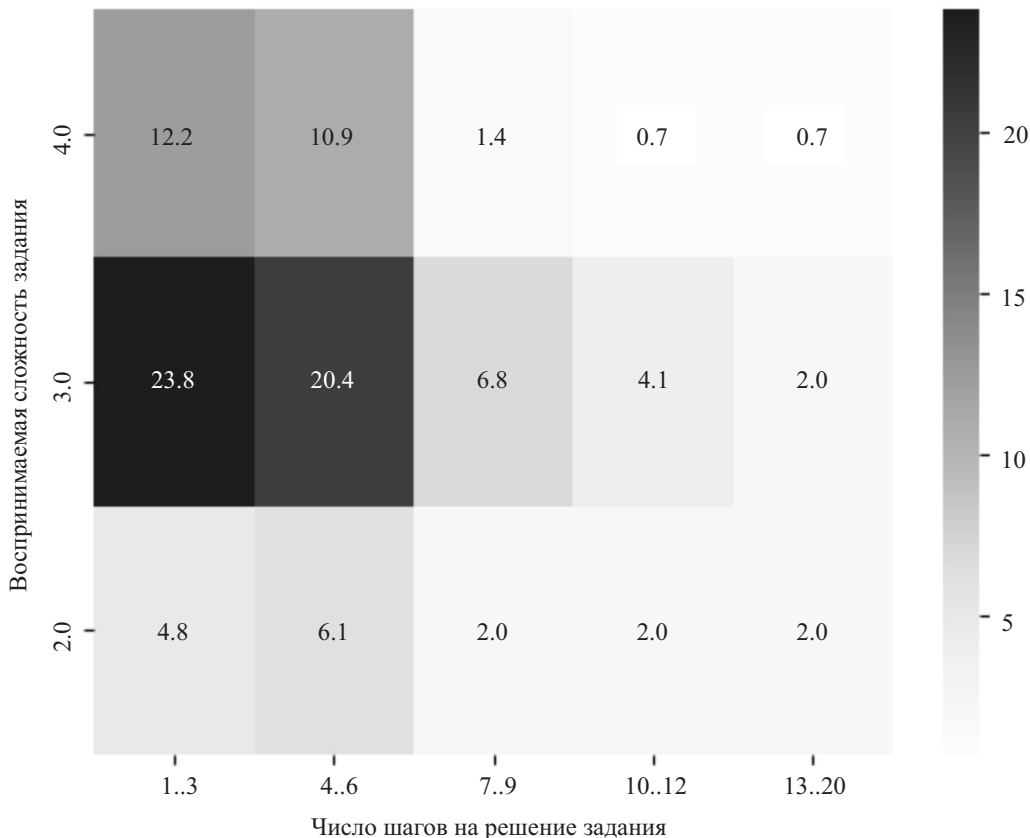


Рис. 4. Тепловая карта частоты встречаемости комбинаций воспринимаемой сложности заданий и числа шагов на решение заданий.

заданий в чат-боте. При разных значениях FLT наблюдаются разные значения числа шагов на решение заданий. Так, при  $FLT = 2$  среднее значение  $ANS = 5,4$ , а при  $FLT = 3$  среднее значение  $ANS = 5,0$ .

В модели (2) показатели SAWP, ACD на небольшом промежутке времени мало меняются. Показатель SRT характеризует итоговое количество набранных баллов за выполнение заданий в чат-боте. Данный показатель коррелируется с ANS, а коэффициент корреляции Пирсона равен  $-0,43$ . Показатель PDT, используемый в модели (2), является результатом модели (1).

Построим тепловую карту, которая покажет частоту встречаемости различных комбинаций воспринимаемой сложности заданий и числа шагов, использованных для решения заданий. На тепловую карту нанесем частоты, выраженные в процентах от общего числа решенных студентами заданий (рис. 4).

Тепловая карта показала, что наибольшие частоты комбинаций значений PDT и числа шагов на решение заданий наблюдаются у студентов, которые решали задания за 6 и менее шагов и у которых PDT является средней или

высокой. Увеличение числа шагов наблюдается у студентов, для которых воспринимаемая сложность заданий игрового чат-бота была средней.

Поэтому можно предложить в алгоритме управления выдачей заданий использовать модели (1) и (2) на основе учета значения ANS.

#### *4.3. Ограничения разработанных моделей*

Несмотря на статистическую значимость полученных результатов, необходимо учитывать ряд ограничений, связанных с разработанными моделями.

– Ограниченная выборка студентов. Эксперимент проводился на ограниченной выборке студентов (60 человек), которая может не отражать многообразие образовательных контекстов и разных групп учащихся. Студенты с разным уровнем подготовки, мотивации и предыдущего опыта взаимодействия с чат-ботами, программирования на Python, с использованием традиционных головоломок могут демонстрировать разную реакцию на задания, что ограничивает обобщаемость результатов.

– Ограниченный набор факторов. Модели не учитывают демографические факторы студентов, влияние внешних факторов. Эти факторы могут влиять на восприятие сложности и интерес к задачам, но они не были включены в анализ. Кроме того, не использовались временные параметры выполнения заданий в чат-боте, например длительности или частота ввода попыток.

– Субъективность оценок. Восприятие трудности и интереса к заданиям субъективны и могут варьироваться у разных учащихся. Модели основаны на самооценках участников, которые могут вносить искажения, связанные с индивидуальными предпочтениями.

– Линейные зависимости. Предположение о линейной зависимости между параметрами и воспринимаемой сложностью и интересом к заданиям может быть чрезмерным упрощением. Реальные отношения могут быть более сложными и нелинейными, что требует дальнейших исследований и более сложных моделей.

– Долгосрочные последствия. В эксперименте не учитываются долгосрочные эффекты от использования игрового чат-бота, такие как влияние на успеваемость, мотивацию и устойчивость интереса к процессу обучения с течением времени. Это ограничивает возможности делать выводы о долгосрочных последствиях внедрения таких технологий в образовательный процесс.

### **5. Структура чат-бота с искусственным интеллектом**

Эффективность использования игрового образовательного чат-бота в обучении программированию зависит от ряда факторов, связанных с характеристиками студентов. К таким факторам относятся:

– Уровень владения навыками программирования: студенты с базовыми знаниями и опытом работы с языком программирования (в данном случае Python) получают больше пользы от использования чат-бота.

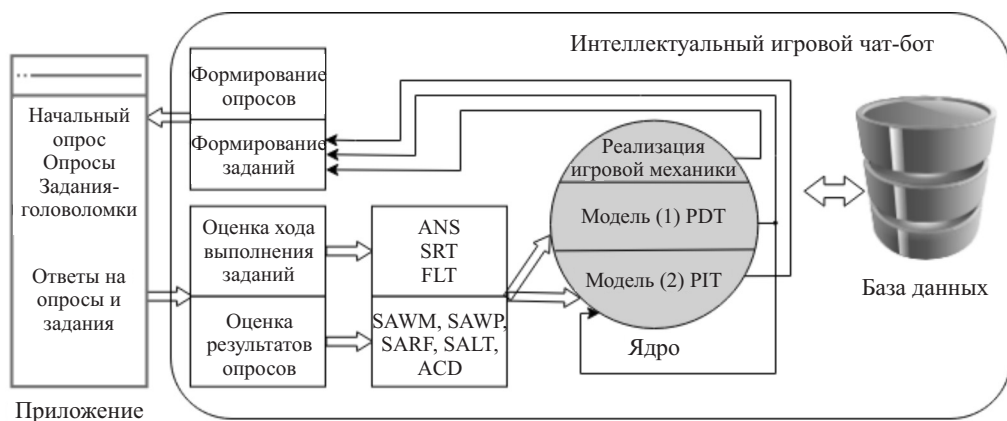


Рис. 5. Структура игрового чат-бота с искусственным интеллектом.

– Стратегии и траектории решения задач: способность студентов эффективно анализировать задачи, планировать шаги решения и использовать адекватные алгоритмы напрямую связана с их успехом в решении заданий.

Восприятие сложности заданий является ключевым фактором, влияющим на интерес к чат-боту. На восприятие сложности оказывают влияние:

– Когнитивные способности: объем рабочей памяти и уровень логического мышления играют важную роль в способности студентов справляться с заданиями.

– Объективная сложность заданий: сложность задач должна быть адекватна уровню подготовки студентов.

– Траектория решения: количество шагов, необходимых для решения задачи, может повлиять на восприятие ее сложности.

– Знание функций и команд языка программирования.

Исходя из этого, можно сделать вывод, что использование игрового чат-бота новичками в программировании сопряжено с определенным риском. И необходим некоторый «пороговый» уровень подготовки студентов, включающий:

– Знание основных функций и команд Python.

– Наличие навыков чтения и написания программного кода.

Студенты с низким уровнем рабочей памяти, логического мышления или недостаточным знанием команд Python могут столкнуться с повышенной сложностью заданий, что негативно скажется на их интересе к чат-боту. Поэтому при внедрении игрового чат-бота в образовательный процесс важно учитывать индивидуальные особенности студентов и обеспечить «пороговый» уровень подготовки для его эффективного использования.

Можно предложить структуру игрового чат-бота с моделями по управлению заданиями-головоломками по типу Wordle для обучения программированию. Здесь по результатам опроса и передачи данных в чат-бот может

быть проведена оценка уровня сложности и интереса заданий, показывающая целесообразность использования чат-бота в текущий момент времени. Соответственно чат-бот в зависимости от полученных значений будет следить за целесообразностью использования, а в случае целесообразности в зависимости от используемой стратегии и выполнения заданий будет подбирать головоломки соответствующего уровня (рис. 5).

В виде псевдокода представим алгоритм работы интеллектуального игрового чат-бота.

*Алгоритм 1. Работа интеллектуального чат-бота*

**Данные:** Модели (1) и (2), результаты опроса и выполнения заданий

**Результат:** Рекомендация по использованию чат-бота, задание-головоломка

1. **if** *Первое задание-головоломка не было сформировано* **then**
2.     Задание пороговых уровней  $PDT_i$  и  $PIT_i$ ;
3.     Определение значений  $SAWM$ ,  $SARF$ ,  $SALT$ ,  $SAWP$ ,  $ACD$  по результатам тестирования и опроса;
4.     Использование моделей (1) и (2) и определение начальных значений  $PDT_0$  и  $PIT_0$ ;
5.     **if**  $PDT_0 > PDT_i$  или  $PIT_0 < PIT_i$  **then**
6.         **return** *Рекомендация о нежелательном и преждевременном использовании игрового чат-бота в данный момент времени;*
7.     **else**
8.         **return** *Первое задание-головоломка;*
9.     **end**
10. **end**
11. Контроль процесса решения задания-головоломки;
12. **if** *Задание-головоломка решено или израсходованы все попытки* **then**
13.     Определение  $NS$ ;
14.     Определение  $ANS$ ;
15.     Определение  $PDT$ ,  $PIT$  на основе моделей (1) и (2);
16.     Определение параметров следующего задания-головоломки с учетом  $PDT$  и  $PIT$ ;
17.     **return** *Новое задание-головоломка;*
18. **end**
19. Контроль длительности выполнения всего задания.

## 6. Заключение

Головоломки по типу Wordle можно использовать на этапе обучения программированию. В целом они являются достаточно интересным образова-

тельным инструментом для студентов. Однако их следует использовать, когда уровень компетенций студентов не является низким. В этом случае решение головоломок будет увлекательным и полезным. Также этот тип головоломок может быть не очень интересен студентам с низким уровнем оперативной памяти и логического мышления.

Поэтому внедрение в игровой чат-бот моделей машинного обучения, учитывающих показатели студента и ход выполнения задания, позволит чат-боту предсказывать воспринимаемую сложность заданий и интерес к ним. За счет этого можно адаптировать поведение чат-бота к уровню студента, чтобы сделать сложность заданий оптимальной, а интерес – максимальным.

Разработанные модели машинного обучения легли в основу алгоритма работы и структуры игрового чат-бота с искусственным интеллектом. На основе разработанной структуры была реализована новая версия игрового чат-бота. Апробация полученного чат-бота показала большой интерес к его использованию среди студентов, обучающихся программированию.

Некоторые параметры моделей (FLT, ANS, SRT, PDT) чат-бот может определять по результатам выполнения задания и использования модели 1. Однако для некоторых параметров (SAWM, SARF, SALT, SAWP, ACD) необходимо делать опросы хотя бы на начальном этапе подключения и запуска чат-бота и, желательно, корректировать их по мере использования чат-бота путем более частых опросов. В связи с этим представляет интерес замена субъективных показателей на объективные и поиск моделей, не содержащих субъективных показателей или в которых эти показатели выражены через параметры выполнения задания. Это будет основным направлением дальнейшей работы.

В дальнейшем планируется расширить выборку и учесть в моделях другие показатели, в частности демографические данные студентов и их опыт работы с чат-ботами, программирования на Python и решения традиционных головоломок. Планируем рассмотреть нелинейные модели влияния показателей на воспринимаемую сложность и интерес к заданиям.

## СПИСОК ЛИТЕРАТУРЫ

1. Mayer R.E. Computer Games in Education // *Ann. Rev. Psychol.* 2019. V. 70. P. 531–549. <https://doi.org/10.1146/ANNUREV-PSYCH-010418-102744>
2. Zhan Z., He L., Tong Y., et. al. The effectiveness of gamification in programming education: Evidence from a meta-analysis // *Comput. Educat.: Artific. Intelligence.* 2022. V. 3. 100096 p. <https://doi.org/10.1016/J.CAEAI.2022.100096>
3. Nouchi R., Taki Y., Takeuchi H., et. al. Brain Training Game Boosts Executive Functions, Working Memory and Processing Speed in the Young Adults: A Randomized Controlled Trial // *PLoS One.* 2013. V. 8. e55518 p. <https://doi.org/10.1371/JOURNAL.PONE.0055518>
4. Huang V., Young M., Fiocco A.J. The Association Between Video Game Play and Cognitive Function: Does Gaming Platform Matter? // *Cyberpsychol. Behav. Soc. Netw.* 2017. V. 20. P. 689–694. <https://doi.org/10.1089/CYBER.2017.0241>

5. *Selamat S.S., Nasir M.K., Adnan N.H.* Investigation of Computational Thinking Skills through Instructional Techniques, Games and Programming Tools // *Int. J. Learn. Teach. Educ. Res.* 2024. V. 23. No. 10. P. 435–452.  
<https://doi.org/10.26803/ijlter.23.10.21>
6. *Parsons D., Haden P.* Parson’s Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses // *Proceedings of the 8th Australasian Conference on Computing Education.* Australian Computer Society, Inc. 2006. V. 52. P. 157–163.
7. *Murphy M., Spillane K., Cully J., et. al.* Can Word Puzzles be Tailored to Improve Different Dimensions of Verbal Fluency? A Report of an Intervention Study // *J. Psych. Elsevier Publ.* 2016. V. 150. P. 743–754.  
<https://doi.org/10.1080/00223980.2016.1182887>
8. *Friedlander K.J., Fine P.A.* Fluid Intelligence is Key to Successful Cryptic Crossword Solving // *J. Expert.* 2020. V. 3. P. 101–132.
9. *Juricic V., Radošević M.* Puzzle-like programming languages in teaching programming // *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 – Proceedings.* Institute of Electrical and Electronics Engineers Inc. 2019. P. 542–546.  
<https://doi.org/10.23919/MIPRO.2019.8757192>
10. *Fakhoury S., Ma Y., Arnaoudova V., Adesope O.* The Effect of Poor Source Code Lexicon and Readability on Developers’ Cognitive Load // *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC).* Gothenburg, Sweden. 2018. P. 286–28610.
11. *Wassila D., Tahar B.* Using serious game to simplify algorithm learning // *International Conference on Education and e-Learning Innovations, ICEELI 2012.* 2012. P. 1–5. <https://doi.org/10.1109/ICEELI.2012.6360569>
12. *Liu Y., Tan J., Yang Z., et. al.* Application Study of Collaborative Multi-Model in Wordle Game // *5th International Conference on Machine Learning and Computer Application (ICMLCA).* 2024. P. 591–596.
13. *Zhang Z., Wang S., Sun D.* A Study of Wordle Games Based on Mathematical Models // *IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT).* 2024. P. 813–817.
14. *Wang S., Zhou G., Zhang Z., et. al.* Exploring wordle: a data-driven analysis of gameplay dynamics, mathematical modeling, and educational implications // *Proc. SPIE 13219, Fourth International Conference on Applied Mathematics, Modelling, and Intelligent Computing (CAMMIC 2024).* 2024. 132190F.  
<https://doi.org/10.1117/12.3036932>
15. *Cunningham-Nelson S., Boles W., Trouton L., et. al.* A review of chatbots in education: Practical steps forward // *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate.* Engineers Australia, Brisbane, Queensland. 2019. P. 299–306.
16. *Setiawan H., Phillipson S.* The effectiveness of Game-Based Science Learning (GBSL) to improve students’ academic achievement: A meta-analysis of current research from 2010 to 2017 // *REID (Research and Evaluation in Education).* 2019. V. 5. P. 152–168. <https://doi.org/10.21831/REID.V5I2.28073>

17. *Karakoç B., Eryılmaz K., Turan Özpolat E., et. al.* The Effect of Game-Based Learning on Student Achievement: A Meta-Analysis Study // *Technology, Knowledge and Learning*. 2020. V. 27. P. 207–222.  
<https://doi.org/10.1007/S10758-020-09471-5>
18. *Connolly T.M., Boyle E.A., MacArthur E., et. al.* A systematic literature review of empirical evidence on computer games and serious games // *Comput. Educ.* 2012. V. 59. P. 661–686. <https://doi.org/10.1016/J.COMPEDU.2012.03.004>
19. *Vahldick A., Mendes A.J., Marcelino M.J.* A review of games designed to improve introductory computer programming competencies // *Proceedings – Frontiers in Education Conference, FIE*. 2014. P. 1–7. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/FIE.2014.7044114>
20. *Miljanovic M.A., Bradbury J.S.* A review of serious games for programming // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018. V. 11243 LNCS. P. 204–216. [https://doi.org/10.1007/978-3-030-02762-9\\_21/COVER](https://doi.org/10.1007/978-3-030-02762-9_21/COVER)
21. *Giannakoulas A., Xinogalos S.* A Review of Educational Games for Teaching Programming to Primary School Students // *Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education*. 2020. P. 1–30. <https://doi.org/10.4018/978-1-7998-4576-8.ch001>
22. *Falkner N., Sooriamurthi R., Michalewicz Z.* Puzzle-based learning for engineering and computer science // *Computer (Long Beach Calif)*. 2010. V. 43. P. 20–28. <https://doi.org/10.1109/MC.2010.113>
23. *Khamparia A., Pandey B.* Effects of visual map embedded approach on students learning performance using Briggs-Myers learning style in word puzzle gaming course // *Comput. Elect. Engin.* 2018. V. 66. P. 531–540. <https://doi.org/10.1016/J.COMPELECENG.2017.12.041>
24. *Toma M., Halpern D.F., Berger D.E.* Cognitive Abilities of Elite Nationally Ranked SCRABBLE and Crossword Experts // *Appl. Cogn. Psychol.* 2014. V. 28. P. 727–737. <https://doi.org/10.1002/ACP.3059>
25. *Hambrick D.Z., Salthouse T.A., Meinz E.J.* Predictors of crossword puzzle proficiency and moderators of age-cognition relations // *J. Exp. Psychol. Gen.* 1999. V. 128. P. 131–164. <https://doi.org/10.1037/0096-3445.128.2.131>
26. *Makasi T., Nili A., Desouza K.C., et. al.* A Typology of Chatbots in Public Service Delivery // *IEEE Softw.* 2022. V. 39. P. 58–66. <https://doi.org/10.1109/MS.2021.3073674>
27. *Marikyan D., Papagiannidis S., Rana O.F., et. al.* “Alexa, let’s talk about my productivity”: The impact of digital assistants on work productivity // *J. Bus. Res.* 2022. V. 142. P. 572–584. <https://doi.org/10.1016/J.JBUSRES.2022.01.015>
28. *Kimani E., Rowan K., McDuff D., et. al.* A Conversational Agent in Support of Productivity and Wellbeing at Work // *8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019*. 2019. P. 332–338. <https://doi.org/10.1109/ACII.2019.8925488>
29. *Gkinko L., Elbanna A.* Hope, tolerance and empathy: employees’ emotions when using an AI-enabled chatbot in a digitalised workplace // *Information Technology and People*. 2022. V. 35. P. 1714–1743. <https://doi.org/10.1108/ITP-04-2021-0328/FULL/PDF>

30. *Bavaresco R., Silveira D., Reis E., et. al.* Conversational agents in business: A systematic literature review and future research directions // *Comput. Sci. Rev.* 2020. V. 36. 100239 p. <https://doi.org/10.1016/J.COSREV.2020.100239>
31. *Brandtzaeg P.B., Følstad A.* Why people use chatbots // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2017. V. 10673 LNCS. P. 377–392. [https://doi.org/10.1007/978-3-319-70284-1\\_30](https://doi.org/10.1007/978-3-319-70284-1_30)
32. *Fotheringham D., Wiles M.A.* The effect of implementing chatbot customer service on stock returns: an event study analysis // *J. Acad. Mark. Sci.* 2023. V. 51. P. 802–822. <https://doi.org/10.1007/S11747-022-00841-2/TABLES/6>
33. *Ciechanowski L., Przegalinska A., Magnuski M., et. al.* In the shades of the uncanny valley: An experimental study of human-chatbot interaction // *Futur. Generat. Comput. Syst.* 2019. V. 92. P. 539–548. <https://doi.org/10.1016/J.FUTURE.2018.01.055>
34. *Thakkar J., Raut P., Doshi Y., et. al.* Erasmus AI Chatbot // *Int. J. Comput. Sci. Eng.* 2018. V. 6. P. 498–502. <https://doi.org/10.26438/IJCSE/V6I10.498502>
35. *Sidaoui K., Jaakkola M., Burton J.* AI feel you: customer experience assessment via chatbot interviews // *J. Servic. Management.* 2020. V. 31. P. 745–766. <https://doi.org/10.1108/JOSM-11-2019-0341/FULL/PDF>
36. *Zhou L., Paul S., Demirkan H., et. al.* Intelligence Augmentation: Towards Building Human-Machine Symbiotic Relationship // *AIS Transactions on Human-Computer Interaction.* 2021. V. 13. P. 243–264. <https://doi.org/10.17705/1thci.00149>
37. *Gkinko L., Elbanna A.* The appropriation of conversational AI in the workplace: A taxonomy of AI chatbot users // *Int. J. Inf. Management.* 2023. V. 69. 102568 p. <https://doi.org/10.1016/J.IJINFOMGT.2022.102568>
38. *Rizomyliotis I., Kastanakis M.N., Giovanis A., et. al.* “How mAy I help you today?” The use of AI chatbots in small family businesses and the moderating role of customer affective commitment // *J. Bus. Res.* 2022. V. 153. P. 329–340. <https://doi.org/10.1016/J.JBUSRES.2022.08.035>
39. *Chen Q., Gong Y., Lu Y., et. al.* Classifying and measuring the service quality of AI chatbot in frontline service // *J. Bus. Res.* 2022. V. 145. P. 552–568. <https://doi.org/10.1016/J.JBUSRES.2022.02.088>
40. *Tam W., Huynh T., Tang A., et. al.* Nursing education in the age of artificial intelligence powered Chatbots (AI-Chatbots): Are we ready yet? // *Nurse Educ. Today.* 2023. V. 129. 105917 p. <https://doi.org/10.1016/J.NEDT.2023.105917>
41. *Zhu Y., Zhang J., Wu J., et. al.* AI is better when I’m sure: The influence of certainty of needs on consumers’ acceptance of AI chatbots // *J. Bus. Res.* 2022. V. 150. P. 642–652. <https://doi.org/10.1016/J.JBUSRES.2022.06.044>
42. *Ruan Y., Mezei J.* When do AI chatbots lead to higher customer satisfaction than human frontline employees in online shopping assistance? Considering product attribute type // *J. Retailing Consum. Ser.* 2022. V. 68. 103059 p. <https://doi.org/10.1016/J.JRETCONSER.2022.103059>
43. *Janssen A., Rodríguez Cardona D., Passlick J., et. al.* How to Make chatbots productive – A user-oriented implementation framework // *Int. J. Hum. Comput. Stud.* 2022. V. 168. 102921 p. <https://doi.org/10.1016/J.IJHCS.2022.102921>

44. *Lee D., Yeo S.* Developing an AI-based chatbot for practicing responsive teaching in mathematics // *Comput. Educ.* 2022. V. 191. 104646 p. <https://doi.org/10.1016/J.COMPEDU.2022.104646>
45. *Liu C.C., Liao M.G., Chang C.H., et. al.* An analysis of children's interaction with an AI chatbot and its impact on their interest in reading // *Comput. Educ.* 2022. V. 189. 104576 p. <https://doi.org/10.1016/J.COMPEDU.2022.104576>
46. *Johnson C., Urazov M., Zanolli E.* Escapeling: A Gamified, AI-Supported Chatbot for Collaborative Language Practice // *Lecture Notes in Networks and Systems.* 2022. V. 349. P. 141–148. [https://doi.org/10.1007/978-3-030-90677-1\\_14/COVER](https://doi.org/10.1007/978-3-030-90677-1_14/COVER)
47. *Варнавский А.Н.* Сравнение показателей восприятия игрового чат-бота с заданиями-головоломками для обучения программированию с показателями восприятия других образовательных инструментов // *Информатика и образование.* 2024. Т. 39. № 4. С. 41–50. <https://doi.org/10.32517/0234-0453-2024-39-4-41-50>
48. *Варнавский А.Н.* Использование игрового чат-бота для развития навыков алгоритмического и логического мышления // *Информатика и образование.* 2024. Т. 39. № 2. С. 48–58. <https://doi.org/10.32517/0234-0453-2024-39-2-48-58>
49. *Варнавский А.Н.* Модели интереса, сложности и воспринимаемой полезности игрового чат-бота с головоломками по типу Wordle для обучения программированию // *Информационные технологии.* 2024. Т. 30. № 7. С. 372–381. <https://doi.org/10.17587/it.30.372-381>

*Статья представлена к публикации членом редколлегии А.А. Галяевым.*

Поступила в редакцию 29.11.2024

После доработки 09.01.2025

Принята к публикации 14.01.2025

# Оптимизация, системный анализ и исследование операций

© 2025 г. Л.Г. АФРАЙМОВИЧ, д-р физ.-мат. наук (levafraimovich@gmail.com),  
М.Д. ЕМЕЛИН (maksim888e@mail.ru)  
(Нижегородский государственный университет)

## ПОСТРОЕНИЕ ОБЛАСТИ ПАРЕТО ПРИ КОМБИНИРОВАНИИ ДОПУСТИМЫХ РЕШЕНИЙ МНОГОКРИТЕРИАЛЬНОЙ АКСИАЛЬНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Рассматривается двухкритериальная трехиндексная аксиальная задача о назначениях, которая уже в однокритериальном случае является одной из классических NP-трудных задач. В рамках данной постановки ставится задача комбинирования допустимых решений, представляющая собой задачу о назначениях на множестве решений, которые содержат только компоненты выбранных допустимых решений. Предлагается полиномиальный алгоритм нахождения Парето-оптимальных решений в задаче комбинирования двух допустимых решений. На его основе строится эвристический подход оценки области Парето многокритериальной аксиальной задачи о назначениях.

*Ключевые слова:* аксиальная задача о назначениях, многоиндексные задачи, многокритериальные задачи, Парето-оптимальность, комбинирование решений, полиномиальная разрешимость, NP-трудность.

**DOI:** 10.31857/S0005231025050056, **EDN:** AXKFNW

### 1. Введение

Существует широкий класс прикладных задач, формализуемых в виде многоиндексных аксиальных задач о назначениях, в качестве примера можно выделить задачи распределения ресурсов, планирования, трекинга объектов, роботизированной логистики [1–5] и др. В общем случае класс многоиндексных аксиальных задач о назначениях является NP-трудным уже в трехиндексном случае [6]. Известны частные полиномиально разрешимые подклассы [1, 7] и подклассы, для которых существуют полиномиальные приближенные алгоритмы [1, 8, 9]. Аксиальная трехиндексная задача о назначениях исследуется в [10–16], где обсуждаются вопросы построения приближенных подходов и реализации точных алгоритмов. Многокритериальные постановки многоиндексных задач о назначениях обсуждаются в [17–20].

В данной работе рассматривается вопрос построения области Парето двухкритериальной трехиндексной аксиальной задачи о назначениях. Сформулирована задача комбинирования допустимых решений, представляющая собой многокритериальную задачу о назначениях на множестве решений, получаемых комбинированием компонент заданных допустимых решений. Предложен полиномиальный алгоритм построения подмножества Парето-оптимальных решений на множестве комбинаций двух допустимых решений. Данный алгоритм применен при построении эвристического алгоритма аппроксимации области Парето двухкритериальной трехиндексной задачи о назначениях. Проведен вычислительный эксперимент, иллюстрирующий данный подход. Работа является продолжением серии статей [21–24], в которых обсуждаются вопросы комбинирования решений аксиальной задачи о назначениях.

Далее статья построена следующим образом. В разделе 2 приводится постановка двухкритериальной трехиндексной аксиальной задачи о назначениях и соответствующей задачи комбинирования допустимых решений. В разделе 3 описывается алгоритм нахождения подмножества Парето-оптимальных решений задачи комбинирования. В разделе 4 приводятся результаты вычислительных экспериментов.

## 2. Постановка задачи

Пусть  $I, J, K$  – непересекающиеся множества индексов,  $I \cap J = \emptyset$ ,  $I \cap K = \emptyset$ ,  $J \cap K = \emptyset$  и  $|I| = |J| = |K| = n$ ;  $c_{ijk}, d_{ijk}, i \in I, j \in J, k \in K$  – трехиндексные матрицы стоимостей;  $x_{ijk}, i \in I, j \in J, k \in K$  – трехиндексная матрица неизвестных. Тогда двухкритериальная трехиндексная аксиальная задача о назначениях ставится как следующая задача целочисленного линейного программирования:

$$(1) \quad \sum_{i \in I} \sum_{j \in J} x_{ijk} = 1, \quad k \in K,$$

$$(2) \quad \sum_{i \in I} \sum_{k \in K} x_{ijk} = 1, \quad j \in J,$$

$$(3) \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad i \in I,$$

$$(4) \quad x_{ijk} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad k \in K,$$

$$(5) \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \rightarrow \min,$$

$$(6) \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk} \rightarrow \min.$$

Для удобства двухкритериальную задачу (1)–(6) обозначим через  $Z_2$ . Обозначим:

$$C(x) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk},$$

$$D(x) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk}.$$

Тогда  $x^*$  называется Парето-оптимальным решением задачи  $Z_2$ , если  $x^*$  удовлетворяет системе ограничений (1)–(4); не существует  $x'$ , удовлетворяющего системе ограничений (1)–(4), такого, что при этом  $C(x^*) > C(x')$  и  $D(x^*) \geq D(x')$  или  $C(x^*) \geq C(x')$  и  $D(x^*) > D(x')$ .

Как известно, (однокритериальная) аксиальная задача о назначениях (1)–(5) является NP-трудной [6]. Отсюда несложно увидеть, что задача построения Парето-оптимального решения двухкритериальной аксиальной задачи о назначениях  $Z_2$  также является NP-трудной. Для доказательства необходимо в соответствующей задаче  $Z_2$  определить  $c_{ijk} = d_{ijk}$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$ .

*Утверждение 1. Задача построения Парето-оптимального решения задачи  $Z_2$  является NP-трудной.*

Таким образом, актуальным является вопрос построения эффективных эвристических подходов при оценке Парето-области задачи  $Z_2$ . Предлагаемый в данной работе эвристический подход основан на идее комбинирования допустимых решений задачи  $Z_2$ . Под комбинированием здесь понимается решение задачи на множестве решений, которые содержат только компоненты заданных допустимых решений.

Приведем постановку задачи комбинирования. Пусть задано множество  $W \subset I \times J \times K$ , которое определяет подмножество разрешенных назначений. Введем вспомогательное ограничение

$$(7) \quad x_{ijk} = 0, \quad (i, j, k) \notin W.$$

Двухкритериальную задачу (1)–(4), (7), (5), (6) для заданного множества  $W$  будем обозначать через  $Z_2(W)$ . Очевидно задача (1)–(6) соответствует задаче  $Z_2(I \times J \times K)$ .

Тогда  $x^*$  называется Парето-оптимальным решением задачи  $Z_2(W)$ , если  $x^*$  удовлетворяет системе ограничений (1)–(4), (7); не существует  $x'$ , удовлетворяющего системе ограничений (1)–(4), (7), такого, что при этом  $C(x^*) > C(x')$  и  $D(x^*) \geq D(x')$  или  $C(x^*) \geq C(x')$  и  $D(x^*) > D(x')$ .

Проблема проверки совместности системы ограничений задачи  $Z_2(W)$ , т.е. системы (1)–(4), (7), для произвольного множества  $W$  является NP-полной [1]. При решении задачи комбинирования будем рассматривать такие множества  $W$ , которые соответствуют назначениям заданного подмножества допустимых решений.

Пусть  $x_{ijk}$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$  – допустимое решение системы ограничений (1)–(4). Тогда через  $W(x)$  обозначим следующее множество разрешенных назначений:

$$W(x) = \{(i, j, k) | x_{ijk} = 1, i \in I, j \in J, k \in K\}.$$

Рассмотрим  $x_{ijk}^1, x_{ijk}^2, \dots, x_{ijk}^r$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$  – произвольные  $r$  допустимых решений системы ограничений (1)–(4). Тогда

$$W(x^1, x^2, \dots, x^r) = W(x^1) \cup W(x^2) \cup \dots \cup W(x^r).$$

Таким образом, далее будем рассматривать задачу  $Z_2(W(x^1, x^2, \dots, x^r))$ .

### 3. Построение подмножества Парето-оптимальных решений

Пусть  $r = 2$  и  $x^1, x^2$  – заданные допустимые решения системы ограничений (1)–(4). Тогда рассмотрим задачу  $Z_2(W(x^1, x^2))$  и вопрос построения ее Парето-оптимальных решений.

*Алгоритм 1.* Построение подмножества Парето-оптимальных решений задачи  $Z_2(W(x^1, x^2))$ .

Шаг 1. Построить граф  $G = (V, A)$ , где

$$V = \{I \cup J \cup K\}, A = \{(i, j), (i, k), (j, k) | (i, j, k) \in W(x^1, x^2)\}.$$

Шаг 2. Найти компоненты связности  $V_l$ ,  $l = \overline{1, q}$  графа  $G$  и построить подграфы  $G_l = (V_l, A_l)$ ,  $l = \overline{1, q}$ , порожденные соответствующими компонентами связности.

Шаг 3. Построить следующие множества:

$$D_l^1 = \{(i, j, k) | (i, j, k) \in W(x^1), (i, j), (i, k), (j, k) \in A_l\}, \quad l = \overline{1, q},$$

$$D_l^2 = \{(i, j, k) | (i, j, k) \in W(x^2), (i, j), (i, k), (j, k) \in A_l\}, \quad l = \overline{1, q}.$$

Шаг 4. Пусть

$$P_l = \left\{ p \left| \sum_{(i,j,k) \in D_l^p} c_{ijk} = \min_{p' \in \{1,2\}} \sum_{(i,j,k) \in D_l^{p'}} c_{ijk}, p \in \{1,2\} \right. \right\},$$

$$p_l = \operatorname{argmin}_{p \in P_l} \sum_{(i,j,k) \in D_l^p} d_{ijk}, \quad l = \overline{1, q}.$$

Шаг 5. Построим Парето-оптимальное решение  $x^{*0}$  по следующему алгоритму. Пусть  $x_{ijk}^{*0} := 0$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$ . Далее для каждого  $l = \overline{1, q}$  выполнить  $x_{ijk}^{*0} := 1$ ,  $(i, j, k) \in D_l^{p_l}$ .

Шаг 6. Построим следующее множество индексов компонент связности:

$$L = \left\{ l \mid \sum_{(i,j,k) \in D_l^{\bar{p}l}} c_{ijk} > \sum_{(i,j,k) \in D_l^{pl}} c_{ijk} \text{ и} \right. \\ \left. \sum_{(i,j,k) \in D_l^{\bar{p}l}} d_{ijk} < \sum_{(i,j,k) \in D_l^{pl}} d_{ijk}, l \in \{1, \dots, q\} \right\},$$

здесь для удобства обозначим  $\bar{p} = 3 - p$  при  $p \in \{1, 2\}$ .

Шаг 7. Упорядочим это множество в порядке невозрастания величины  $tg(l)$ , где

$$tg(l) = \frac{\sum_{(i,j,k) \in D_l^{pl}} d_{ijk} - \sum_{(i,j,k) \in D_l^{\bar{p}l}} d_{ijk}}{\sum_{(i,j,k) \in D_l^{\bar{p}l}} c_{ijk} - \sum_{(i,j,k) \in D_l^{pl}} c_{ijk}},$$

т.е. пусть  $L = \{l_1, \dots, l_{|L|}\}$  и  $tg(l_s) \geq tg(l_{s+1})$ ,  $s = \overline{1, |L| - 1}$ .

Шаг 8. Построим  $|L|$  Парето-оптимальных решений  $x^{*s}$ ,  $s = \overline{1, |L|}$  следующим способом.

Пусть  $x_{ijk}^{*s} := 0$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$ , далее для каждого  $t = \overline{1, |q|}$  выполнить:

если  $t \in \{l_1, \dots, l_s\}$ , то  $x_{ijk}^{*s} := 1$ ,  $(i, j, k) \in D_t^{\bar{p}t}$ ,

иначе  $x_{ijk}^{*s} := 1$ ,  $(i, j, k) \in D_t^{pt}$ .

Приведем численный пример, иллюстрирующий работу алгоритма 1.

*Пример 1.* Пример работы алгоритма 1.

Пусть  $n = 6$ , матрицы  $c_{ijk}$ ,  $d_{ijk}$  заданы следующим образом:

$$c_{ijk} = \begin{cases} 0, (i, j, k) \in \{(1, 1, 1), (1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (3, 3, 4), \\ \quad (5, 5, 5), (5, 5, 6), (6, 6, 5)\}, \\ 1, (i, j, k) \in \{(4, 4, 3)\}, \\ 2, (i, j, k) \in \{(6, 6, 6)\}, \\ 3, (i, j, k) \in \{(2, 2, 2)\}, \\ 10 \text{ иначе.} \end{cases}$$

$$d_{ijk} = \begin{cases} 0, (i, j, k) \in \{(1, 1, 1), (1, 1, 2), (2, 2, 2), (3, 3, 3), (4, 4, 3), (3, 3, 4), \\ \quad (5, 5, 5), (5, 5, 6), (6, 6, 6)\}, \\ 1, (i, j, k) \in \{(2, 2, 1), (4, 4, 4)\}, \\ 4, (i, j, k) \in \{(6, 6, 5)\}, \\ 10 \text{ иначе.} \end{cases}$$

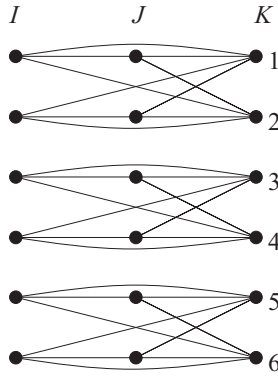


Рис. 1. Граф  $G$  шага 1.

Рассмотрим два допустимых решения  $x^1, x^2$ :

$$x_{ijk}^1 = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{иначе.} \end{cases}$$

$$x_{ijk}^2 = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3), (5, 5, 6), (6, 6, 5)\}, \\ 0 & \text{иначе.} \end{cases}$$

Опишем работу алгоритма 1 на данном численном примере.

Граф  $G$ , полученный на шаге 1, представлен на рис. 1.

На шаге 2 строятся компоненты связности графа  $G$ . Как видим, граф  $G$  имеет три компоненты связности одинаковой структуры. Соответственно множества, получаемые на шаге 3, имеют следующий вид:

$$D_1^1 = \{(1, 1, 1), (2, 2, 2)\}, \quad D_2^1 = \{(3, 3, 3), (4, 4, 4)\}, \quad D_3^1 = \{(5, 5, 5), (6, 6, 6)\}, \\ D_1^2 = \{(1, 1, 2), (2, 2, 1)\}, \quad D_2^2 = \{(3, 3, 4), (4, 4, 3)\}, \quad D_3^2 = \{(5, 5, 6), (6, 6, 5)\}.$$

На шаге 4 получим:

$$P_1 = \{2\}, \quad p_1 = 2, \\ P_2 = \{1\}, \quad p_2 = 1, \\ P_3 = \{2\}, \quad p_3 = 2.$$

На шаге 5 первое полученное Парето-оптимальное решение  $x^{*0}$  будет построено следующим способом:

$$x_{ijk}^{*0} = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (5, 5, 6), (6, 6, 5)\}, \\ 0 & \text{иначе.} \end{cases}$$

Значения критериев, достигаемые этим решением, равны  $C(x^{*0}) = 0$ ,  $D(x^{*0}) = 6$ .

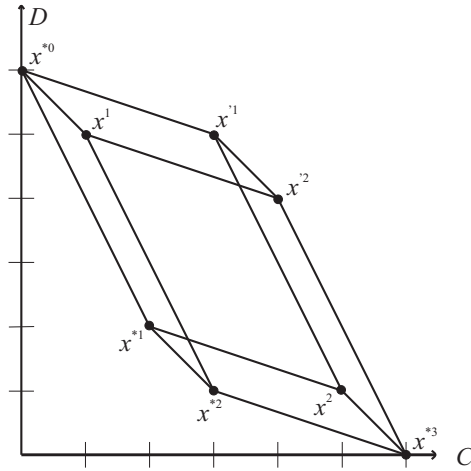


Рис. 2. Найденные решения на плоскости  $CD$ .

На шаге 6 множество  $L$  будет содержать все три компоненты связности и будет упорядочено на шаге 7 следующим образом:  $l_1 = 3$ ,  $l_2 = 2$ ,  $l_3 = 1$ . Следующие три Парето-оптимальные решения, построенные на шаге 8, имеют вид:

$$x_{ijk}^{*1} = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{иначе.} \end{cases}$$

$$x_{ijk}^{*2} = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{иначе.} \end{cases}$$

$$x_{ijk}^{*3} = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 4), (4, 4, 3), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{иначе.} \end{cases}$$

И соответствующие им значения критериев:

$$\begin{aligned} C(x^{*1}) &= 2, & D(x^{*1}) &= 2, \\ C(x^{*2}) &= 3, & D(x^{*2}) &= 1, \\ C(x^{*3}) &= 6, & D(x^{*3}) &= 0. \end{aligned}$$

Полученные решения отображены на плоскости  $CD$ , согласно значению их критерия (см. рис. 2). Здесь  $x^1, x^2$  – исходные решения,  $x^{*0}, x^{*1}, x^{*2}, x^{*3}$  – найденные алгоритмом 1 Парето-оптимальные решения задачи комбинирования,  $x^1, x'^2$  – возможные решения задачи комбинирования. Как можно увидеть на данном примере, кроме найденных алгоритмом решений, существует еще одно Парето-оптимальное решение  $x^1$ .

Далее докажем корректность алгоритма 1.

Рассмотрим многокритериальную задачу. Пусть  $X \subseteq R^n$  – множество допустимых решений задачи,  $f(x) \in R^m$  – целевая функция задачи,  $f_i(x) \rightarrow \min$ ,  $i = \overline{1, m}$ .

*Утверждение 2.* Пусть множество  $X \subseteq R^n$  конечно. Если  $x \in X$  не является Парето-оптимальным решением, то существует Парето-оптимальное решение  $y \in X$ , доминирующее над  $x$ , т.е.  $f_i(y) \leq f_i(x)$ ,  $i = \overline{1, m}$ , и существует  $j \in \{1, \dots, m\}$  такой, что  $f_j(y) < f_j(x)$ .

*Доказательство.* Пусть выполняются условия утверждения и  $x \in X$  не является Парето-оптимальным решением. Тогда построим алгоритм нахождения такого решения  $y$ .

Шаг 1. Найти решение  $x'$ , доминирующее над решением  $x$ . Если бы такого решения не нашлось, то решение  $x$  было бы Парето-оптимальным, что привело бы к противоречию.

Шаг 2.  $x := x'$ , если  $x$  Парето-оптимальное, вернуть  $x$ , иначе перейти на шаг 1.

Из определения Парето-оптимальности следует, что количество шагов 2 алгоритма не превысит мощности множества решений. Так как множество конечно, то алгоритм найдет Парето-оптимальное решение за конечное число шагов. Утверждение доказано.

*Утверждение 3.* Любое допустимое решение  $x$ , удовлетворяющее системе ограничений задачи  $Z_2(W(x^1, x^2))$ , может быть построено путем выбора троек только из первого или только из второго решения для каждой компоненты связности независимо.

*Доказательство.* Данное утверждение было показано в доказательстве теоремы 1 из [21].

Тогда введем:

$$p_l(x) = \begin{cases} 1, & \text{если } x_{ijk} = 1, \text{ для всех } (i, j, k) \in D_l^1, \\ 2, & \text{если } x_{ijk} = 1, \text{ для всех } (i, j, k) \in D_l^2. \end{cases}$$

*Утверждение 4.* Для любого Парето-оптимального решения  $x'$  задачи  $Z_2(W(x^1, x^2))$  выполняется:

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk} \quad \text{и} \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}, \quad l \notin L.$$

*Доказательство.* Доказательство проведем от противного. Предположим, что существует Парето-оптимальное решение  $x'$ , для которого найдется  $l \notin L$ , такое что

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk} \quad \text{или} \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}.$$

Рассмотрим два случая:

1.  $\sum_{(i,j,k) \in D_l^{p_l(x')}$   $c_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $c_{ijk}$ . Согласно шагу 5 алгоритма 1 выполняется

$$\sum_{(i,j,k) \in D_l^{p_l(x')}$$
  $c_{ijk} > \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $c_{ijk}$ .

Тогда

а) если  $\sum_{(i,j,k) \in D_l^{p_l(x')}$   $d_{ijk} < \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $d_{ijk}$ , то  $l \in L$ , получаем противоречие,

б) иначе выполняется  $\sum_{(i,j,k) \in D_l^{p_l(x')}$   $d_{ijk} \geq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $d_{ijk}$ , тогда в реше-

нии  $x'$  тройки, соответствующие компоненте связности  $l$ , можно заменить следующим образом:

$$\begin{aligned} x'_{ijk} &:= 0, & (i, j, k) \in D_l^{p_l(x')}, \\ x'_{ijk} &:= 1, & (i, j, k) \in D_l^{p_l(x^{*0})}, \end{aligned}$$

уменьшив при этом значение обоих критериев, следовательно,  $x'$  – не Парето-оптимальное решение, получаем противоречие.

2.  $\sum_{(i,j,k) \in D_l^{p_l(x')}$   $c_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $c_{ijk}$ ,  $\sum_{(i,j,k) \in D_l^{p_l(x')}$   $d_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $d_{ijk}$ .

Согласно шагу 5 алгоритма 1 выполняется

$$\sum_{(i,j,k) \in D_l^{p_l(x')}$$
  $d_{ijk} > \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}$   $d_{ijk}$ ,

тогда в решении  $x'$  тройки, соответствующие компоненте связности  $l$ , можно заменить следующим образом:

$$\begin{aligned} x'_{ijk} &:= 0, & (i, j, k) \in D_l^{p_l(x')}, \\ x'_{ijk} &:= 1, & (i, j, k) \in D_l^{p_l(x^{*0})}, \end{aligned}$$

уменьшив при этом значение обоих критериев, следовательно,  $x'$  – не Парето-оптимальное решение, получаем противоречие. Утверждение доказано.

Обозначим прямую, соединяющую точки  $x^{*t}$ ,  $x^{*t+1}$  на плоскости  $CD$ , как  $P_t$ ,  $t = \overline{1, |L|}$ . Запишем уравнение прямой  $P_t$ :

$$\frac{D - D(x^{*t})}{D(x^{*t-1}) - D(x^{*t})} = \frac{C - C(x^{*t})}{C(x^{*t-1}) - C(x^{*t})}.$$

Для удобства приведем уравнение к виду

$$D = D(x^{*t}) + \frac{D(x^{*t}) - D(x^{*t-1})}{C(x^{*t-1}) - C(x^{*t})} (C(x^{*t}) - C) = D(x^{*t}) + tg(l_t)(C(x^{*t}) - C).$$

Уравнение прямой также можно записать в виде

$$D = D(x^{*t-1}) + tg(l_t)(C(x^{*t-1}) - C).$$

Будем говорить, что допустимое решение  $x$  задачи  $Z_2(W(x^1, x^2))$  лежит не ниже прямой  $P_t$ , если

$$D(x) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x)).$$

Несложно увидеть, что если решение  $x$  лежит не ниже прямой  $P_t$ , то решение  $x$  не будет доминировать ни над одним из решений  $x^{*t-1}, x^{*t}$ .

*Утверждение 5.* Если для некоторого  $e \in \{1, \dots, |L|\}$  решение  $x$  лежит не ниже прямой  $P_e$  и  $C(x) \leq C(x^{*e})$ , то оно лежит не ниже любой прямой  $P_t$ ,  $t = e, |L|$ .

*Доказательство.* Так как  $x$  лежит не ниже прямой  $P_e$ , то

$$D(x) \geq D(x^{*e}) + tg(l_e)(C(x^{*e}) - C(x)).$$

По построению  $tg(l_t) \geq tg(l_{t+1}) \geq 0$ ,  $t = \overline{1, |L| - 1}$ . Из условия утверждения  $(C(x^{*e}) - C(x)) \geq 0$ . Отсюда

$$tg(l_e)(C(x^{*e}) - C(x)) \geq tg(l_{e+1})(C(x^{*e}) - C(x))$$

и, следовательно,

$$D(x) \geq D(x^{*e}) + tg(l_{e+1})(C(x^{*e}) - C(x)),$$

т.е. решение  $x$  лежит не ниже прямой  $P_{e+1}$ . Отсюда по индукции получили, что  $x$  лежит не ниже прямой  $P_t$ ,  $t \geq e$ . Утверждение доказано.

*Теорема 1.* Не существует Парето-оптимального решения  $x'$  задачи  $Z_2(W(x^1, x^2))$ , доминирующего над каким-либо из решений, построенных алгоритмом 1.

*Доказательство.* Доказательство от противного. Предположим, что существует Парето-оптимальное решение  $x'$ , доминирующее хотя бы над одним из решений  $x^{*0}, \dots, x^{*|L|}$ , построенных алгоритмом. Из утверждения 4 следует, что такое решение может отличаться от  $x^{*0}$  только в тройках компонент  $l \in L$ .

Построим  $|L| + 1$  решение  $x''^s$ ,  $s = \overline{0, |L|}$  следующим способом:

Шаг 1. Пусть  $x''_{ijk} := 0$ ,  $i \in I$ ,  $j \in J$ ,  $k \in K$ .

Шаг 2. Далее для каждого  $t = \overline{1, q}$  выполнить:

– если  $t \in \{l_1, \dots, l_s\}$  и  $p_t(x') \neq p_t(x^{*0})$ , то  $x''_{ijk} := 1$ ,  $(i, j, k) \in D_t^{\overline{p_t}}$ ,

– иначе  $x''_{ijk} := 1$ ,  $(i, j, k) \in D_t^{p_t}$ .

По построению  $x''^0 = x^{*0}$ , отсюда решение  $x''^0$  лежит не ниже прямой  $P_1$ .

Покажем, что если решение  $x''^s$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ , то и решение  $x''^{s+1}$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ .

Для каждой из прямых  $P_t$ ,  $t = \overline{1, r + 1}$  возможны два случая:

1.  $p_{l_{s+1}}(x') = p_{l_{s+1}}(x^{*0})$ , тогда решение  $x''^{s+1}$  совпадает с решением  $x''^s$ , а значит, оно не ниже прямой  $P_t$ ,

2.  $p_{l_{s+1}}(x') \neq p_{l_{s+1}}(x^{*0})$ , тогда из условия, что решение  $x''^s$  лежит не ниже прямой  $P_t$ , следует, что

$$(8) \quad D(x''^s) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^s)).$$

Из построения:

$$D(x''^{s+1}) = D(x''^s) + tg(l_{s+1})(C(x''^s) - C(x''^{s+1})).$$

Применим неравенство (8):

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^s)) + tg(l_{s+1})(C(x''^s) - C(x''^{s+1})).$$

Отсюда:

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^{s+1})) + (tg(l_t) - tg(l_{s+1}))(C(x''^{s+1}) - C(x''^s)).$$

По построению  $C(x''^{s+1}) - C(x''^s) \geq 0$  и  $tg(l_t) - tg(l_{s+1}) \geq 0$  при  $s + 1 \geq t$ . Тогда:

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^{s+1})),$$

т.е. решение  $x''^{s+1}$  не ниже прямой  $P_t$ .

Отсюда: если решение  $x''^s$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ , то и решение  $x''^{s+1}$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, s + 1}$ .

По построению  $C(x''^{s+1}) \leq C(x^{*s+1})$ . Тогда по утверждению 5 решение  $x''^{s+1}$  лежит не ниже прямых  $P_t$ ,  $t = s + 1, |L|$  (здесь  $e = s + 1$ ). Следовательно, если решение  $x''^s$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ , то решение  $x''^{s+1}$  не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ . Отсюда по индукции получаем, что  $x''^{|L|}$  не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ .

По построению  $x''^{|L|} = x'$ , т.е.  $x'$  лежит не ниже прямых  $P_t$ ,  $t = \overline{1, |L|}$ , а следовательно, решение  $x'$  не доминирует ни над одним из решений  $x^{*0}, \dots, x^{*|L|}$ , построенных алгоритмом 1. Получаем противоречие, предположение неверно. Теорема доказана.

*Теорема 2. Решения  $x^{*0}, \dots, x^{*|L|}$ , найденные алгоритмом 1, являются Парето-оптимальными решениями задачи  $Z_2(W(x^1, x^2))$ .*

*Доказательство.* Доказательство проведем от противного. Предположим, что это не так, тогда существует решение  $x^{*q'}$ ,  $q' \in \{0, \dots, L\}$ , не являющееся Парето-оптимальным решением задачи  $Z_2(W(x^1, x^2))$ . Тогда согласно утверждению 2 существует Парето-оптимальное решение  $x'$  задачи  $Z_2(W(x^1, x^2))$ , доминирующее над  $x^{*q'}$ :

$$C(x') < C(x^{*q'}) \quad \text{и} \quad D(x') < D(x^{*q'}).$$

Но согласно теореме 1 такого решения  $x'$  не существует. Получаем противоречие. Предположение неверно. Теорема доказана.

*Утверждение 6. Алгоритм 1 не гарантирует нахождение всего множества Парето-оптимальных решений задачи  $Z_2(W(x^1, x^2))$ .*

*Доказательство.* Построим численный пример. Пусть  $n = 4$  и

$$x_{ijk}^1 = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4)\}, \\ 0 & \text{иначе,} \end{cases}$$

$$x_{ijk}^2 = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\}, \\ 0 & \text{иначе.} \end{cases}$$

Пусть

$$c_{ijk} = \begin{cases} 0, & \text{если } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (2, 2, 1), (4, 4, 3)\}, \\ 1, & \text{если } (i, j, k) = (1, 1, 2), \\ 2, & \text{если } (i, j, k) = (3, 3, 4), \\ 5 & \text{иначе,} \end{cases}$$

$$d_{ijk} = \begin{cases} 0, & \text{если } (i, j, k) \in \{(2, 2, 2), (4, 4, 4), (1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\}, \\ 1, & \text{если } (i, j, k) = (1, 1, 1), \\ 2, & \text{если } (i, j, k) = (3, 3, 3), \\ 5 & \text{иначе.} \end{cases}$$

Тогда существует четыре допустимых решения задачи комбинирования:

$$x_{ijk}^{*1} = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4)\}, \\ 0 & \text{иначе,} \end{cases}$$

здесь  $C(x^{*1}) = 0$ ,  $D(x^{*1}) = 3$ ;

$$x_{ijk}^{*2} = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\}, \\ 0 & \text{иначе,} \end{cases}$$

здесь  $C(x^{*2}) = 3$ ,  $D(x^{*2}) = 0$ ;

$$x_{ijk}^{*3} = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4)\}, \\ 0 & \text{иначе,} \end{cases}$$

здесь  $C(x^{*3}) = 1$ ,  $D(x^{*3}) = 2$ ;

$$x_{ijk}^{*4} = \begin{cases} 1, & \text{если } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 4), (4, 4, 3)\}, \\ 0 & \text{иначе,} \end{cases}$$

здесь  $C(x^{*4}) = 2$ ,  $D(x^{*4}) = 1$ .

Каждое из построенных четырех решений является Парето-оптимальным решением задачи  $Z_2(W(x^1, x^2))$ . При этом  $q = 2$ ,  $|L| = 2$  и, следовательно, алгоритм 1 найдет только три решения. Утверждение доказано.

Таким образом, предложенный алгоритм 1 гарантирует построение Парето-оптимальных решений задачи  $Z_2(W(x^1, x^2))$ , однако не гарантирует построение всей области Парето.

*Теорема 3. Алгоритм 1 требует  $O(n^2)$  вычислительных операций.*

### Доказательство.

Пусть входными данными алгоритма 1 являются допустимые решения  $x^1, x^2$ , представленные в виде коллекции троек  $(i, j, k)$  и стоимостей  $c_{ijk}$ , для которых соответствующие переменные принимают значение 1. Согласно (1)–(4) для каждого из допустимых решений количество таких троек составляет  $n$ . Тогда на шаге 1 алгоритма строится граф  $G = (V, A)$ , где  $|V| = O(n)$ ,  $|A| = O(n)$ . На шаге 2 граф  $G$  разбивается на компоненты связности. Это осуществляется при помощи обхода графа в ширину, который требует  $O(|V| + |A|) = O(n)$  вычислительных операций. На шаге 3 для входных троек алгоритма определяются соответствующие компоненты связности, этот шаг требует  $O(n)$  вычислительных операций. На шаге 4 для каждой компоненты связности определяется, из какого решения будут взяты тройки в первое найденное Парето-оптимальное решение, данный шаг требует  $O(n)$  вычислительных операций. На шаге 5 строится первое Парето-оптимальное решение; так как можно представить решение в виде коллекции из  $n$  троек  $(i, j, k)$ , то этот шаг требует  $O(n)$  вычислительных операций. На шаге 6 из всех компонент связности выбираются только те, которые соответствуют критерию, что требует  $O(n)$  вычислительных операций. На шаге 7 подмножество компонент связности упорядочивается, что требует  $O(n \log(n))$  вычислительных операций. И, наконец, на шаге 8 строится  $O(n)$  решений, каждое из которых строится за  $O(n)$  вычислительных операций. Таким образом, алгоритм 1 требует  $O(n^2)$  вычислительных операций. Теорема доказана.

## 4. Вычислительный эксперимент

Предложенный алгоритм 1 гарантирует построение подмножества области Парето задачи  $Z_2(W(x^1, x^2))$ . Далее применим данный алгоритм при построении эвристических методов аппроксимации Парето области исходной задачи  $Z_2$ . Эффективность работы эвристического алгоритма будем оценивать вычислительным экспериментом.

Эксперимент 1. Построим матрицы  $c_{ijk}$ ,  $d_{ijk}$  следующим способом: для каждого индекса  $i \in I \cup J \cup K$  сгенерируем случайную точку  $p$  на плоскости  $XY$  так, чтобы  $p_x, p_y$  были целочисленными и равномерно распределенными на отрезке  $[0, 2^{32} - 1]$ , тогда  $c_{ijk} = \text{dist}(i, j) + \text{dist}(j, k) + \text{dist}(i, k)$ , где  $\text{dist}(a, b)$  – Манхэттенское расстояние между точками  $a$  и  $b$ . Аналогичным образом определим  $d_{ijk}$ .

Определим процедуру локальной оптимизации для допустимого решения  $x$  задачи  $Z_2$ :

Шаг 1. Равновероятно выберем случайное число  $a \in [0, 1]$ .

Шаг 2. Построим трехиндексную матрицу стоимостей  $e_{ijk} = ac_{ijk} + (1 - a)d_{ijk}$ .

Шаг 3. Применим процедуру локальной оптимизации, предложенную в [15] к решению  $x$  исходной задачи, но с критерием  $\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} e_{ijk} x_{ijk} \rightarrow \min$ .

Построим  $n$  случайных допустимых решений  $x_1, \dots, x_n$  задачи  $Z_2$ , к каждому из которых будем применять предложенную процедуру локальной оптимизации до тех пор, пока решение не перестанет меняться. Полученные решения обозначим как  $x'_1, \dots, x'_n$ . Из полученных решений выберем недоминируемые. Обозначим построенную из этих решений аппроксимацию Парето-кривой как  $R$ .

Применим алгоритм 1 к парам решений, полученным после локальной оптимизации, для решения следующих задач:

$$Z_2(W(x'_1, x'_2)), \dots, Z_2(W(x'_{n-1}, x'_n)).$$

Из всех полученных решений выберем недоминируемые. Обозначим построенную из этих решений аппроксимацию Парето-кривой как  $Q$ .

Будем сравнивать количество точек, которые не доминируются соответствующей аппроксимацией и не совпадают по значению критерия с какой-либо точкой этой аппроксимации. Введем обозначение количества точек, которые не доминируются аппроксимацией  $A$ , следующим образом:  $B(A) = |\{x|x - \text{допустимое решение задачи } Z_2, \nexists x' \in A \text{ такого, что } C(x') \leq C(x) \text{ и } D(x') \leq D(x)\}|$ .  $B(R), B(Q)$  считались для каждого теста путем проверки (полным перебором) всего множества допустимых решений на соответствие условию несуществования доминирующего или равного по значению критериев решения в соответствующей аппроксимации. Тесты проводились на размерностях  $n = 6, 7, 8$ , для каждой размерности было проведено 50 тестов.

В табл. 1 приведены результаты первого вычислительного эксперимента. Согласно им применение алгоритма 1 позволило в среднем уменьшить количество недоминируемых точек на 6,57%.

**Таблица 1**

Размерность задачи	$\frac{B(R)-B(Q)}{B(R)} 100\%$
6	2,14%
7	8,03%
8	9,54%

Эксперимент 2. Построим матрицы  $c_{ijk}, d_{ijk}$  следующим способом: для каждого индекса  $i \in I \cup J \cup K$  сгенерируем случайную точку  $p$  на плоскости  $XY$  так, чтобы  $p_x, p_y$  были целочисленными и равномерно распределенными на отрезке  $[0, 2^{32} - 1]$ , тогда  $c_{ijk} = \text{dist}(i, j) + \text{dist}(j, k) + \text{dist}(i, k)$ ,  $d_{ijk} = \max(\text{dist}(i, j), \text{dist}(j, k), \text{dist}(i, k))$ , где  $\text{dist}(a, b)$  – Манхэттенское расстояние между точками  $a$  и  $b$ .

Построим  $n^3$  случайных решений  $x_1, \dots, x_{n^3}$  задачи  $Z_2$ , к каждому из которых применим один шаг предложенной выше процедуры локальной оптимизации. Полученные решения обозначим как  $x'_1, x'_2, \dots, x'_{n^3}$ . Из полученных решений выберем недоминируемые. Обозначим построенную из этих решений аппроксимацию Парето-кривой как  $R$ .

Применим алгоритм 1 к парам решений, полученным после локальной оптимизации, для решения следующих задач:

$$Z_2(W(x'_1, x'_2)), \dots, Z_2(W(x'_{n^3-1}, x'_{n^3})).$$

Из всех полученных решений выберем недоминируемые. Обозначим построенную из этих решений аппроксимацию Парето-кривой как  $Q$ .

Будем сравнивать количество точек, которые не доминируются соответствующей аппроксимацией и не совпадают по значению критерия с какой-либо точкой этой аппроксимации.

Тесты проводились на размерностях  $n = 6, 7, 8$ , было проведено 50 тестов.

**Таблица 2**

Размерность задачи	$\frac{B(R)-B(Q)}{B(R)} 100\%$
6	11,44%
7	18,08%
8	11,71%

В табл. 2 приведены результаты второго вычислительного эксперимента. Согласно им применение алгоритма 1 позволило в среднем уменьшить количество недоминируемых точек на 13,74%.

## 5. Заключение

В работе предложен алгоритм построения Парето-оптимальных решений задачи комбинирования двух допустимых решений двухкритериальной трехиндексной задачи о назначениях. Доказана корректность построенного алгоритма и получена квадратичная оценка его сложности. Данный алгоритм может быть применен в качестве дополнительного шага в эвристических подходах оценки области Парето двухкритериальной трехиндексной задачи о назначениях. Приведены результаты вычислительных экспериментов, демонстрирующие повышение качества приближенных решений, полученных в результате постобработки предложенным алгоритмом оптимального комбинирования.

## СПИСОК ЛИТЕРАТУРЫ

1. *Spieksma F.C.R.* Multi Index Assignment Problems. Complexity, Approximation, Applications. P.M. Pardalos, L.S. Pitsoulis (Eds.) / Nonlinear Assignment Problems: Algorithms and Applications. Dordrecht: Kluwer Acad. Publishers, 2000. P. 1–11.
2. *Burkard R., Dell'Amico M., Martello S.* Assignment problems: revised reprint. PA: SIAM, 2012.
3. *Kuroki Y., Matsui T.* An approximation algorithm for multidimensional assignment problems minimizing the sum of squared errors // Discret. Appl. Math. 2009. V. 157. No. 9. P. 2124–2135.

4. *Poore A.B.* Multidimensional Assignment Problems Arising in Multitarget and Multisensor Tracking. P.M. Pardalos, L.S. Pitsoulis (Eds.) / *Nonlinear Assignment Problems: Algorithms and Applications*. Dordrecht: Kluwer Acad. Publishers, 2000. P. 13–38.
5. *Zhuang Y., Zhou Y., Hassini E., Yuan Y., Hu X.* BRack retrieval and repositioning optimization problem in robotic mobile fulfillment systems // *Transportation Research Part E: Logistics and Transportation Review*, 2022. V. 167. P. 102920.
6. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
7. *Афраймович Л.Г.* Многоиндексные транспортные задачи с 2-вложенной структурой // *АиТ*. 2013. № 1. С. 116–134.  
*Afraimovich L.G.* Multiindex Transportation Problems with 2-embedded Structure // *Autom. Remote Control*. 2013. V. 74. No. 1. P. 90–104.
8. *Bandelt H.J., Crama Y., Spieksma F.C.R.* Approximation algorithms for multidimensional assignment problems with decomposable costs // *Discret. Appl. Math.* 1994. V. 49. P. 25–50.
9. *Crama Y., Spieksma F.C.R.* Approximation Algorithms for Three-Dimensional Assignment Problems with Triangle Inequalities // *Eur. J. Oper. Res.* 1992. V. 60. P. 273–279.
10. *Burkard R.E., Rudolf R., Woeginger G.J.* Three-dimensional axial assignment problems with decomposable cost coefficients // *Discret Appl. Math.* 1996. V. 65. P. 123–139.
11. *Spieksma F., Woeginger G.* Geometric three-dimensional assignment problems // *Eur. J. Oper. Res.* 1996. V. 91. P. 611–618.
12. *Ćustić A., Klinz B., Woeginger G.J.* Geometric versions of the three-dimensional assignment problem under general norms // *Discret. Optim.* 2015. V. 18. P. 38–55.
13. *Balas E., Saltzman M.J.* An Algorithm for the Three-Index Assignment Problem // *Oper. Res.* 1991. V. 39. No. 1. P. 150–161.
14. *Natu S., Date K., Nagi R.* GPU-accelerated Lagrangian heuristic for multidimensional assignment problems with decomposable costs // *Parallel Comput.* 2020. V. 97. P. 102666.
15. *Huang G., Lim A.* A hybrid genetic algorithm for the Three-Index Assignment Problem // *Eur. J. Oper. Res.* 2006. V. 172. P. 249–257.
16. *Kim B.J., Hightower W.L., Hahn P.M., Zhu Y.R., Sun L.* Lower bounds for the axial three-index assignment problem // *Eur. J. Oper.* 2010. V. 202. P. 654–668.
17. *Дичковская С.А., Кравцов М.К.* Исследование полиномиальных алгоритмов решения многокритериальной трехиндексной планарной задачи о назначениях // *Журн. вычислит. мат. и мат. физики*. 2007. Т. 47. 6. С. 1077–1086.
18. *Емеличев В.А., Перепелица В.А.* Сложность дискретных многокритериальных задач // *Дискретная математика*. 1994. Т. 6. Вып. 1. С. 3–33.
19. *Прилуцкий М.Х.* Многокритериальные многоиндексные задачи объемнокалендарного планирования // *Известия РАН. Теория и системы управления*. 2007. 1. С. 78–82.
20. *Прилуцкий М.Х.* Многокритериальное распределение однородного ресурса в иерархических системах // *АиТ*. 1996. № 2. С. 24–29.

21. *Афраймович Л.Г., Емелин М.Д.* Комбинирование решений аксиальной задачи о назначениях // *АиТ.* 2021. No. 8. С. 159–168.  
*Afraimovich L.G., Emelin M.D.* Combining solutions of the axial assignment problem // *Autom. Remote Control*, 2021. V. 82. No. 8. P. 1418–1425.
22. *Афраймович Л.Г., Емелин М.Д.* Эвристические стратегии комбинирования решений трехиндексной аксиальной задачи о назначениях // *АиТ.* 2021. № 10. С. 6–12.  
*Afraimovich L.G., Emelin M.D.* Heuristic Strategies for Combining Solutions of the Three-Index Axial Assignment Problem // *Autom. Remote Control*, 2021. V. 82. No. 10. 1635–1640.
23. *Afraimovich L.G., Emelin M.D.* Complexity of Solutions Combination for the Three-Index Axial Assignment Problem // *Mathematics* 2022. V. 10. No. 7. 1062.
24. *Афраймович Л.Г., Емелин М.Д.* Свертки критериев при комбинировании решений многокритериальной аксиальной задачи о назначениях // *АиТ.* 2024. № 8. С. 86–98.  
*Afraimovich L.G., Emelin M.D.* Convolution of criteria of a multicriterial axial assignment problem // *Autom. Remote Control*, 2024. V. 85. No. 8. P. 809–818.

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

Поступила в редакцию 10.11.2024

После доработки 12.02.2025

Принята к публикации 26.02.2025

# Оптимизация, системный анализ и исследование операций

© 2025 г. Ю.Ф. ЛЕОНОВА (yuliya.igosheva@gmail.com),  
А.В. ПАНЮКОВ, д-р физ.-мат. наук (paniukovav@susu.ru)  
(Южно-Уральский государственный университет, Челябинск)

## СТАТИСТИЧЕСКОЕ ИССЛЕДОВАНИЕ КАЧЕСТВА АЛГОРИТМА СОЕДИНЕНИЯ ЦИКЛОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА НА МИНИМУМ

Задача коммивояжера является одной из наиболее изученных в комбинаторной оптимизации, однако исследование новых подходов и улучшение существующих методов остается актуальной задачей. В данной работе проведен анализ качества алгоритма соединения циклов для задачи коммивояжера на минимум. Представлены результаты вычислительного эксперимента на пяти семействах задач, проанализированы точность и временная сложность алгоритма. Для симметричных экземпляров задачи построена регрессионная модель, описывающая зависимость оценки относительной погрешности от числа вершин. Показано, что полиномиальная модель наилучшим образом аппроксимирует полученные данные и удовлетворяет основным статистическим предположкам. Полученные результаты позволяют оценить характер роста ошибки и обосновать применимость алгоритма к экземплярам задачи коммивояжера большой размерности.

*Ключевые слова:* задача коммивояжера, эвристический алгоритм, асимптотическая точность алгоритма, статистическое исследование.

**DOI:** 10.31857/S0005231025050065, **EDN:** AXACAX

### 1. Введение

Задача коммивояжера является классической задачей комбинаторной оптимизации и при этом остается одной из наиболее важных и актуальных в области прикладной математики и информатики. Ее значимость проявляется во множестве практических сфер, таких как логистика, транспорт, производство, планирование маршрутов и др. Задача коммивояжера на минимум состоит в поиске гамильтонова цикла с минимальной суммой весов ребер в полном графе.

В последние десятилетия широко исследовались различные методы решения задачи коммивояжера, включая точные алгоритмы, аппроксимационные и эвристические методы [1, 2]. Однако с появлением новых вычислительных технологий и постоянным увеличением объема данных возникает необходи-

мость в разработке более эффективных и масштабируемых алгоритмов для решения этой проблемы.

Задача коммивояжера является NP-трудной [3]. Поэтому для ее решения актуальной является разработка полиномиальных приближенных алгоритмов.

В данной работе представлено эмпирическое исследование и статистический анализ точности и временной сложности эвристического алгоритма для решения задачи коммивояжера – алгоритма соединения циклов. В разделе 2 приведены характеристики оценок точности и времени работы популярных эффективных алгоритмов для решения задачи коммивояжера на минимум. В разделе 3 дается краткое описание исследуемого алгоритма. В разделе 4 описана генерация задач и проведение вычислительного эксперимента. Результаты вычислительного эксперимента на случайных входных данных представлены в разделе 5: приведены данные по точности и времени решения задач. В разделе 6 приводится статистический анализ точности алгоритма соединения циклов. В разделе 7 сформулированы основные выводы по результатам экспериментов.

## 2. Обзор состояния проблемы

Эвристические и приближенные алгоритмы, такие как алгоритм ближайшего соседа, жадный алгоритм, ближайшая вставка и другие, широко используются в практике благодаря их простоте и эффективности. Они часто

**Таблица 1.** Верхняя граница оценки точности и время работы алгоритмов

Алгоритм	Верхняя граница	Время работы
Ближайший сосед [5]	0, 5 $[\log_2 N + 1]$	$O(N^2)$
Ближайший сосед с двойным концом [6]		$O(N^2) \log N$
Жадный [5]		
Ближайшее сложение [6]	$2 - 2/N$	$O(N^2)$
Ближайшая вставка [6, 7]		$O(N^2) \log N$
Самая дешевая вставка [6, 7]		$O(N^2)$
Самая дальняя вставка [6, 7]		
Произвольная вставка [7]		
Вставка ближайшего сегмента		
Двойное минимальное пространственное дерево [8]		
Двойное минимальное пространственное дерево модифицированное		
Кристофидес [9]	$3/2 - 1/N$	$O(N^3)$
Кривая Мура [10]	$\log N$	$O(N \log N)$
Кривая Серпинского [11]		
2-Opt [12]	$\approx 2$	$O(N^2)$

демонстрируют хорошие результаты для небольших наборов данных, но могут не обеспечивать оптимальное решение для задач большой размерности.

В табл. 1 представлены априорные оценки точности и времени некоторых эвристических алгоритмов [4] для задачи коммивояжера на минимум. Верхние границы для алгоритмов рассчитаны как отношение  $f(s)/f(s_0)$ , где  $f(s)$  – полученная длина тура, а  $f(s_0)$  – оптимальная длина тура.

Приведенные алгоритмы для решения задачи коммивояжера имеют различные преимущества и недостатки в зависимости от характеристик входных данных и требуемой точности результата.

В последующих разделах будет представлен анализ точности и временной сложности алгоритма соединения циклов.

### 3. Алгоритм соединения циклов для решения задачи коммивояжера

Авторами статьи [13] был исследован эвристический алгоритм для решения задачи коммивояжера – алгоритм соединения циклов (Cycles Merging Algorithm – CMA). Приведем краткое описание алгоритма.

Первый шаг алгоритма состоит в нахождении в заданном графе 2-регулярного суграфа экстремального веса, т.е. покрытия этого графа циклами. Данную конструкцию принято называть 2-фактором минимального веса. Задача поиска 2-фактора минимального веса может быть сведена к задаче о назначении, для которой известны точные полиномиальные алгоритмы.

На втором шаге проверяется условие единственности цикла полученного решения. Если 2-фактор представлен единственным циклом, то этот цикл является решением задачи. Если 2-фактор представлен несколькими циклами, то перебираются попарно различные циклы  $r$  и  $t$ , в каждом цикле выбранной пары перебираются по одному ребру.

Пусть выбраны  $e_{\{r \oplus t\}}^r = [v_1, v_2]$  и  $e_{\{r \oplus t\}}^t = [u_1, u_2]$ , для них находятся две пары сопряженных ребер ( $f = [v_1, u_1], g = [v_2, u_2]$ ) и ( $f = [v_1, u_2], g = [v_2, u_1]$ ), используемых для соединения циклов (рис. 1). Выполняется поиск набора

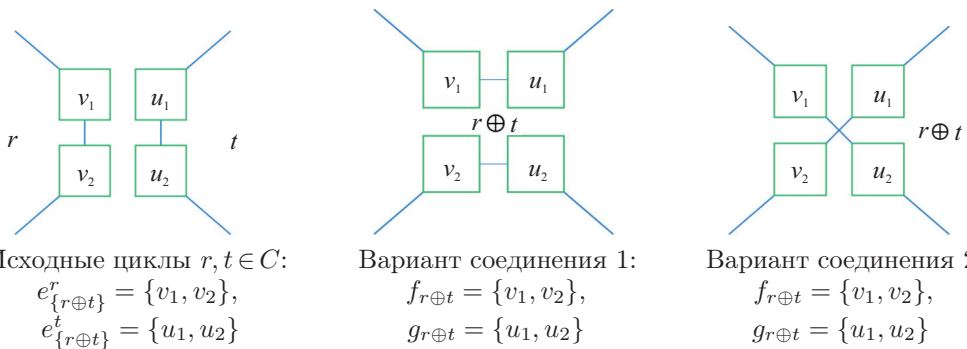


Рис. 1. Варианты соединения циклов.

$r, t, e_{\{r \oplus t\}}^r, e_{\{r \oplus t\}}^t, f, g$  из перечисленных элементов такой, что соединенный цикл имеет минимальный вес.

Затем найденная пара циклов заменяется объединенным циклом с минимальной стоимостью. Алгоритм завершает работу, когда текущий 2-фактор содержит один цикл.

Для описанного алгоритма сформулирована и доказана теорема о вычислительной сложности алгоритма.

*Теорема 1. Вычислительная сложность алгоритма соединения циклов не превосходит величины  $O(|V|^3)$ .*

Более подробное описание алгоритма соединения циклов и доказательство теоремы о вычислительной сложности приведены в [13].

Для метрической задачи коммивояжера на максимум для данного алгоритма ранее была установлена теоретическая граница точности [13], что позволяет строго оценивать его качество в этом классе задач. Однако для задачи коммивояжера на минимум такие теоретические границы отсутствуют, поэтому представляет интерес проведение эмпирического исследования точности алгоритма.

#### 4. Описание тестовых задач

Для оценки качества алгоритма соединения циклов для задачи коммивояжера на минимум вычислительный эксперимент проводился на наборах задач из библиотеки TSPLIB [14], а также на случайно сгенерированных экземплярах с различными характеристиками матрицы стоимостей.

Оценка эффективности алгоритма проводилась на следующих семействах задачи коммивояжера:

1. Асимметричные экземпляры задачи коммивояжера из TSPLIB. TSPLIB содержит реальные и искусственно сгенерированные задачи, широко используемые в тестировании эвристических и точных алгоритмов. Асимметричность стоимости переездов между вершинами делает решение более сложным, так как нарушается свойство симметрии, присущее многим классическим методам.

2. Евклидовы экземпляры задачи коммивояжера из TSPLIB ( $n \leq 3000$ ). Этот тип данных моделирует геометрические версии задачи, где вершины соответствуют точкам на плоскости, а стоимости переходов между ними определяются евклидовым расстоянием. Такие задачи широко встречаются в логистике, навигации и сетевом планировании.

3. Асимметричные матрицы стоимости  $W = [w(i, j)]$ , где  $w(i, j)$  – случайные числа из  $\{0, 1, 2, \dots, 10^5\}$ . Равномерное распределение значений в фиксированном интервале моделирует случайные системы с независимыми весами переходов. Это позволяет протестировать алгоритм в условиях полной неопределенности структуры стоимости.

4. Асимметричные матрицы стоимости, где  $w(i, j)$  – случайные числа из  $\{0, 1, 2, \dots, i \times j\}$ . В данном семействе диапазон возможных стоимостей увеличивается по мере роста индексов вершин, что моделирует задачи, где связь между вершинами становится дороже или сложнее по мере увеличения их номеров. Такой вид матрицы позволяет проанализировать поведение алгоритма на задачах с неравномерной сложностью связей.

5. Симметричные матрицы стоимости  $W$ , где  $w(i, j)$  – случайные числа из  $\{0, 1, 2, \dots, 10^5\}$  для  $i < j$ . В отличие от асимметричных случаев, симметричная структура ограничивает множество возможных решений, но при этом приводит к увеличению числа коротких циклов в исходном покрытии. Это затрудняет процесс объединения циклов в один маршрут и может негативно влиять на точность эвристического алгоритма.

6. Симметричные матрицы, где  $w(i, j)$  – случайные числа из  $\{0, 1, 2, \dots, i \times j\}$  для  $i < j$ . Аналогично асимметричному случаю, но с учетом симметрии весов. Такой тип задачи моделирует случаи, где удаленные вершины связаны с более высокой стоимостью переходов. Однако наличие множества коротких циклов на ранних этапах решения делает задачу вычислительно сложной и может ухудшать точность приближенного решения.

7. Экземпляры задачи на наклонной плоскости, где вес дуги определяется как

$$w(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \max(0, y_i - y_j) + 2 \max(0, y_j - y_i),$$

причем координаты вершин генерировались независимо и равномерно на интервале  $\{0, 1, 2, \dots, 10^5\}$ . Эта модель имитирует ситуацию, когда перемещение по вертикали сопряжено с дополнительными затратами, например при передвижении по горным дорогам. Разница в стоимости движения вверх и вниз делает задачу асимметричной и усложняет ее решение.

Для семейств 3–6 количество вершин  $n$  варьировалось от 100 до 3000 с шагом 100. Все результаты являются средними по 100 испытаниям каждое.

В эксперименте используются экземпляры, созданные как случайным, так и детерминированным образом. Количество и разнообразие используемых семейств позволяет проверить надежность тестируемого алгоритма.

## **5. Результаты вычислительного эксперимента для задачи коммивояжера на минимум**

Для первого и второго семейства экземпляров задачи (асимметричные и евклидовы задачи из TSPLIB) результаты приведены в [15]. Анализ точности и вычислительной эффективности алгоритма соединения циклов для остальных семейств рассматриваются в рамках данной работы.

### 5.1. Анализ оценки относительной погрешности

На рис. 2 приведены оценки относительной погрешности, полученные в ходе вычислительного эксперимента для различных семейств задач коммивояжера. Элементы матрицы стоимостей этих задач были сгенерированы случайным образом в соответствии с правилами генерации, описанными в разделе 4.

Анализ результатов показывает различия в поведении оценки относительной погрешности в зависимости от семейства рассматриваемых экземпляров.

Для асимметричных экземпляров задачи коммивояжера с равномерно распределенными элементами матрицы весов из множества  $\{0, 1, 2, \dots, 10^5\}$  (семейство 3) относительная погрешность уменьшается с ростом числа вершин. При  $n = 100$  ее среднее значение составляет 9,15%, а при  $n = 3000$  снижается до 1,97%. Аналогичная тенденция наблюдается и для семейства 4, однако величина погрешности в среднем остается ниже, уменьшаясь с 5,12% при  $n = 100$  до 1,06% при  $n = 3000$ .

Для симметричных экземпляров задачи коммивояжера, представленных семействами 5 и 6, наблюдается иное поведение. Относительная погрешность в обоих случаях возрастает с увеличением числа вершин. Так, для семейства 5 она увеличивается с 103% при  $n = 100$  до 541% при  $n = 3000$ . Аналогичный рост характерен и для семейства 6, с увеличением погрешности с 65 до 175% при тех же значениях  $n$ .

Для экземпляров наклонных плоскостей (семейство 7) оценка относительной погрешности остается относительно стабильной на всем интервале значений  $n$ , колеблясь в пределах 36,6–37,7%. Данный результат свидетельствует

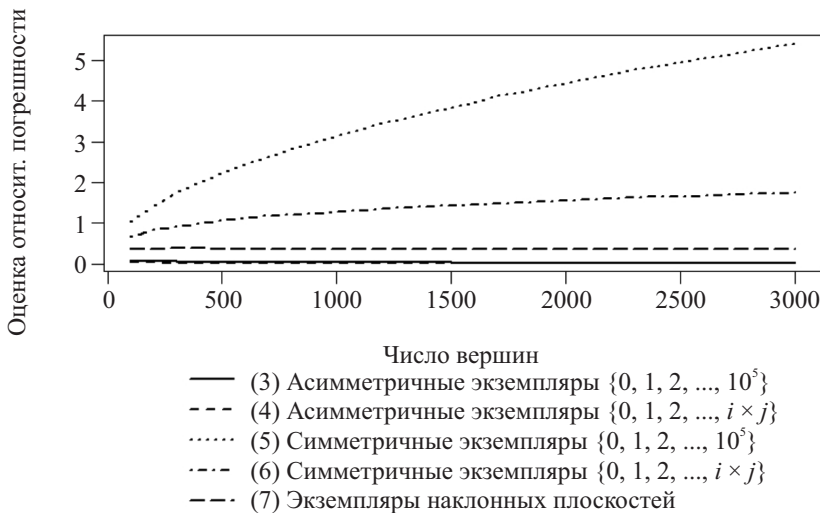


Рис. 2. Оценка относительной погрешности алгоритма соединения циклов для различных семейств задачи коммивояжера.

о меньшей зависимости точности алгоритма от размерности задачи в данном семействе экземпляров.

Таким образом, анализ показывает, что:

- для асимметричных экземпляров задачи коммивояжера оценка относительной погрешности уменьшается с увеличением числа вершин;
- для симметричных экземпляров задачи коммивояжера наблюдается обратная тенденция – погрешность увеличивается с ростом размерности задачи;
- семейство наклонных плоскостей демонстрирует стабильные значения относительной погрешности, незначительно изменяющиеся с увеличением числа вершин.

### 5.2. Анализ времени решения

На рис. 3 представлено среднее время работы алгоритма соединения циклов при решении одной задачи для различных семейств экземпляров задачи коммивояжера.

Анализ временных характеристик показывает различное поведение роста времени решения в зависимости от семейства рассматриваемых экземпляров.

Для асимметричных экземпляров задачи коммивояжера с равномерно распределенными элементами матрицы весов из множества  $\{0, 1, 2, \dots, 10^5\}$  (семейство 3) время решения увеличивается от 0,105 с при  $n = 100$  до 259,88 с при  $n = 3000$ . Для семейства 4 ( $W = [w(i, j)]$ , где  $w(i, j)$  – равномерно распределенные случайные числа из  $\{0, 1, 2, \dots, i \times j\}$ ) время решения растет быстрее, начиная с 0,14 с при  $n = 100$  и достигая 3057,04 с при  $n = 3000$ . Это

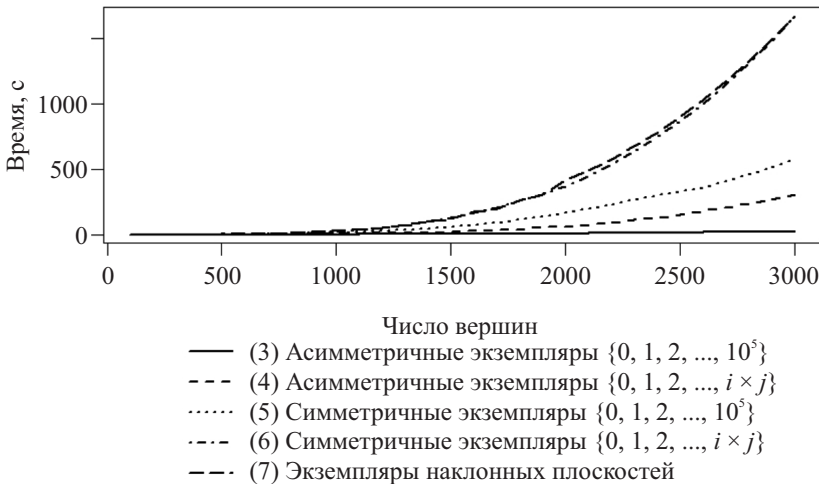


Рис. 3. Время решения одного экземпляра задачи коммивояжера алгоритмом соединения циклов для различных семейств задач.

свидетельствует о значительном влиянии структуры матрицы стоимостей на вычислительную сложность алгоритма.

Для симметричных экземпляров задачи коммивояжера наблюдается более быстрый рост времени решения. В частности, для семейства 5 ( $W = [w(i, j)]$ , где  $w(i, j)$  – равномерно распределенные случайные числа из  $\{0, 1, 2, \dots, 10^5\}$ ) оно возрастает с 0,114 с при  $n = 100$  до 5760,3 с при  $n = 3000$ . Для семейства 6 ( $W = [w(i, j)]$ , где  $w(i, j)$  – равномерно распределенные случайные числа из  $\{0, 1, 2, \dots, i \times j\}$ ) этот рост еще более выражен: от 0,27 с до 16 669,63 с на том же интервале значений  $n$ . Таким образом, специфика структуры весов оказывает влияние не только на точность, но и на вычислительную сложность алгоритма.

Для экземпляров наклонных плоскостей (семейство 7) также наблюдается значительный рост времени решения. При  $n = 100$  среднее время составляет 0,279 с, а при  $n = 3000$  оно достигает 16 694,93 с, что аналогично показателям для шестого семейства задач и существенно превышает данные по всем остальным семействам задач. Это указывает на сложность обработки таких экземпляров алгоритмом соединения циклов.

В целом анализ временных характеристик показывает, что:

- асимметричные экземпляры задачи коммивояжера решаются быстрее, чем симметричные, однако структура матрицы стоимостей существенно влияет на рост времени решения;
- для симметричных экземпляров наблюдается ускоренный рост времени работы алгоритма, особенно для семейства 6;
- экземпляры наклонных плоскостей демонстрируют наибольшие значения времени решения, что может быть связано с особенностями геометрической структуры задач.

Сложность симметричной неевклидовой задачи коммивояжера для алгоритмов построения тура, начинающихся с решения задачи о назначении, обусловлена низкой точностью нижней границы, полученной на основе этого метода. В частности, для подавляющего большинства симметричных экземпляров задача о назначении приводит к цикловому покрытию, содержащему большое количество коротких циклов (табл. 2). Это значительно затрудняет процесс их объединения в единый маршрут, увеличивая как вычислительные затраты, так и итоговую относительную погрешность.

Как видно из таблицы, для асимметричных экземпляров отношение числа начальных циклов к числу вершин существенно уменьшается с увеличением размерности задачи, снижаясь с 0,0382 (100 вершин) до 0,0070 (1000 вершин). Это указывает на тенденцию к формированию более длинных циклов, что способствует более эффективному соединению маршрутов на последующих этапах работы алгоритма.

В отличие от этого для симметричных экземпляров данное отношение остается практически неизменным, колеблясь в диапазоне 0,4768–0,4920. Это свидетельствует о том, что при формировании 2-фактора для данного се-

**Таблица 2.** Отношение числа начальных циклов к числу вершин задачи

Число вершин	Асимметричные экземпляры	Симметричные экземпляры
100	0,0382	0,4920
200	0,0225	0,4768
300	0,0162	0,4771
400	0,0139	0,4892
500	0,0118	0,4848
600	0,0101	0,4867
700	0,0086	0,4896
800	0,0076	0,4902
900	0,0071	0,4890
1000	0,0070	0,4878

мейства задач образуется множество коротких подциклов, что затрудняет их объединение в итоговый маршрут.

Рост времени решения для симметричных экземпляров объясняется тем, что процесс соединения большого количества коротких циклов требует значительно большего количества итераций, что приводит к усложнению структуры алгоритма и росту его вычислительной сложности. Как показано на рис. 3, время работы алгоритма соединения циклов для симметричных экземпляров существенно превышает аналогичные значения для асимметричных задач.

Таким образом, для повышения точности и сокращения времени решения симметричной неевклидовой задачи коммивояжера требуется дальнейшее изучение специализированных эвристик и приближенных алгоритмов, адаптированных для работы с данным семейством задач.

## **6. Статистический анализ точности алгоритма соединения циклов для симметричной задачи коммивояжера**

В рамках исследования точности алгоритма соединения циклов для симметричной задачи коммивояжера на минимум был проведен статистический анализ эмпирических данных, полученных для экземпляров задачи с матрицей весов, элементы которой генерируются равномерно из множества  $\{0, 1, 2, \dots, i \times j\}$  (пятое семейство экземпляров задач из раздела 4). Поскольку для симметричной задачи коммивояжера на минимум отсутствуют теоретические границы точности алгоритма, для прогноза зависимости относительной погрешности от числа вершин были использованы методы регрессионного анализа.

Рассматривались три возможные регрессионные модели: степенная, полиномиальная второго порядка и логарифмическая. Эти модели были выбраны на основе наблюдений о характере роста относительной погрешности с увеличением числа вершин.

### 6.1. Выбор модели регрессии

Исследовательский анализ является необходимым этапом для проверки допущений [16], связанных с регрессионными моделями, и выбора наиболее подходящей модели для аппроксимации данных. В рамках этого этапа анализа были проведены визуальные проверки и вычислены статистические характеристики для каждой модели, что позволило сравнить их эффективность и точность.

Визуальные методы, такие как точечная диаграмма рассеяния и матрица точечной диаграммы, использовались для анализа взаимосвязи между числом вершин и относительной погрешностью. Также были использованы гистограммы и тесты на нормальность распределения остатков. Статистические тесты, такие как коэффициент детерминации  $R^2$ , скорректированный  $R^2$ , а также тесты Дарбина–Уотсона и Бреуша–Пагана, позволили оценить соответствие модели статистическим предположениям.

#### 6.1.1. Степенная регрессия

Степенная модель имеет вид

$$(1) \quad y = ax^b.$$

После логарифмирования уравнение принимает линейный вид:

$$(2) \quad \log y = \log a + b \times \log x.$$

Для этой модели были получены следующие результаты:

- Коэффициент детерминации  $R^2 = 0,9676$ , скорректированный  $R^2 = 0,9665$ , что свидетельствует о высоком качестве аппроксимации данных.
- Значимость коэффициентов модели подтверждается  $p$ -значениями, которые меньше  $2 \times 10^{-16}$ .
- Тест Дарбина–Уотсона показал отсутствие значимой автокорреляции остатков ( $DW = 1,9527$ ,  $p = 0,3711$ ).
- Тест Бреуша–Пагана выявил наличие гетероскедастичности ( $p = 1,24 \times 10^{-6}$ ), что указывает на нестабильность дисперсии остатков.
- Глобальный тест на соответствие предположениям линейной регрессии показал нарушение предположений ( $p = 2,81 \times 10^{-7}$ ).

Таким образом, хотя степенная модель имеет высокий коэффициент детерминации и предоставляет хорошую аппроксимацию, выявленные проблемы с гетероскедастичностью и нарушением линейных предположений указывают на ее ограниченность.

#### 6.1.2. Полиномиальная регрессия

Полиномиальная модель второго порядка имеет вид

$$(3) \quad \log y = a + b \times \log x + c \times \log^2 x.$$

Статистические характеристики этой модели:

- $R^2 = 0,9999$ , скорректированный  $R^2 = 0,9999$ , что подтверждает почти идеальное соответствие данных модели.
- Все коэффициенты модели статистически значимы ( $p < 0,001$ ).
- Тест Дарбина–Уотсона показал отсутствие автокорреляции ( $DW = 2,2082$ ,  $p = 0,5771$ ).
- Тест Бреуша–Пагана не выявил гетероскедастичности ( $p = 0,1035$ ).
- Глобальный тест показал небольшие отклонения от предпосылок ( $p = 0,002$ ).

Полиномиальная модель второго порядка показала значительно лучшее соответствие данным по сравнению со степенной моделью, устранив проблемы гетероскедастичности. Однако тест на глобальное соответствие предпосылкам линейности указывает на возможные отклонения, которые тем не менее не имеют значительного влияния на точность модели.

### 6.1.3. Логарифмическая регрессия

Логарифмическая модель описывается уравнением

$$(4) \quad y = a + b \times \ln x.$$

Статистические результаты:

- $R^2$  немного ниже, чем у полиномиальной модели, но остается высоким.
- Остатки соответствуют требованиям нормальности и гомоскедастичности.
- Глобальный тест подтверждает удовлетворительное соответствие предпосылкам.

Логарифмическая модель, как и полиномиальная модель, показала хорошее соответствие данным и предпосылкам регрессии с легко интерпретируемыми коэффициентами.

### 6.1.4. Выбор модели

На рис. 4 представлены графики эмпирических и модельных значений для различных регрессионных моделей.

Анализ показал, что степенная модель имеет высокий коэффициент детерминации, но нарушает предпосылки регрессии. Полиномиальная модель второго порядка значительно улучшает качество аппроксимации, устраняя гетероскедастичность. Добавление третьего порядка улучшает качество предсказаний, но может привести к избыточному усложнению. Логарифмическая модель хорошо описывает замедляющийся рост, но имеет ограничения при экстраполяции.

С учетом результатов тестов и интерпретации данных наиболее подходящими являются полиномиальная модель второго порядка и логарифмическая регрессия. Полиномиальная модель обеспечила наилучшее соответствие данным, устранив основные проблемы степенной регрессии. Логарифмическая модель хорошо описывает замедляющийся рост, но при экстраполяции

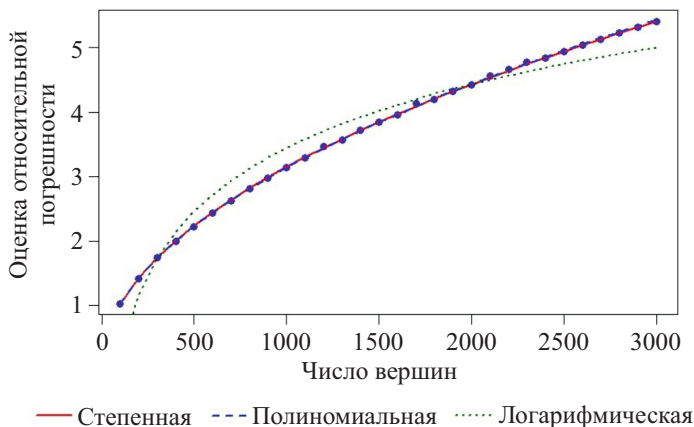


Рис. 4. Сравнение моделей регрессии для симметричных экземпляров из множества  $\{0, 1, 2, \dots, 10^5\}$ .

на значения, выходящие за пределы обучающего набора, она может давать смещение, так как логарифмическая зависимость предполагает бесконечный рост при больших  $x$ , что не всегда соответствует реальным данным.

В дальнейшем анализе будет использована полиномиальная модель второго порядка, как наиболее точная и удовлетворяющая ключевым статистическим требованиям (табл. 3).

**Таблица 3.** Глобальная проверка предпосылок полиномиальной регрессии

Тест	Значение	$p$ -значение
Глобальная статистика	16,843	0,0021 (нарушены)
Асимметрия	3,189	0,0742 (допустимо)
Экссесс	2,321	0,1277 (допустимо)
Функция связи	9,898	0,0017 (нарушены)
Гетероскедастичность	1,436	0,2308 (допустимо)

Полиномиальная модель в логарифмической шкале демонстрирует хорошее соответствие данным, однако небольшие отклонения от линейности могут указывать на ее ограничения при экстраполяции. В целом результаты подтверждают ее применимость, но при прогнозировании для больших значений необходимо учитывать возможное смещение.

### 6.2. Оценка и интерпретация модели

Построенная полиномиальная регрессионная модель описывает зависимость средней относительной погрешности от числа вершин. Включение квадратичного члена позволило учесть нелинейный характер роста погрешности. Коэффициент  $b_1 = 0,4465$  при  $\log n$  указывает на ускоряющийся рост

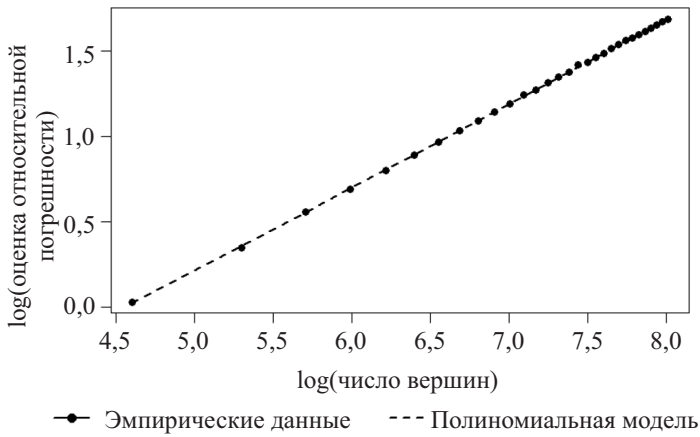


Рис. 5. Эмпирические и модельные значения логарифмически преобразованных данных для полиномиальной регрессии для симметричных экземпляров из множества  $\{0, 1, 2, \dots, 10^5\}$ .

ошибки при увеличении размерности задачи, а положительное значение  $b_2 = 0,0035$  при  $(\log n)^2$  подтверждает небольшое ускорение роста при малых значениях  $n$ . Однако данный эффект сглаживается при больших размерностях задачи, что свидетельствует о тенденции к стабилизации погрешности.

Ключевые статистические показатели модели:

- $R^2 = 0,9999$  (практически полное объяснение дисперсии).
- $F$ -статистика: 129 100 ( $p < 2 \times 10^{-16}$ ), что подтверждает значимость модели и высокую степень соответствия данным.
- Доверительные интервалы показывают, что все коэффициенты модели статистически значимы ( $p < 0,01$ ).

На рис. 5 представлено графическое сравнение эмпирических данных и предсказанных значений модели.

Анализ глобального теста предпосылок регрессии показал нарушение функции связи ( $p = 0,0017$ ). Однако отсутствие гетероскедастичности и автокорреляции остатков позволяет считать полиномиальную модель подходящей для описания зависимости, несмотря на небольшие отклонения от предположений линейности.

### 6.3. Прогнозирование для задач большой размерности

На основании построенной полиномиальной модели выполнена экстраполяция для прогнозирования относительной погрешности для задач размером до 10 000 вершин (рис. 6). Результаты показывают, что при больших  $n$  тенденция к росту ошибки сохраняется, однако эффект ускорения, обусловленный квадратичным членом, становится менее выраженным. Это подтверждает, что относительная погрешность стремится к стабилизации, а ее увеличение замедляется.

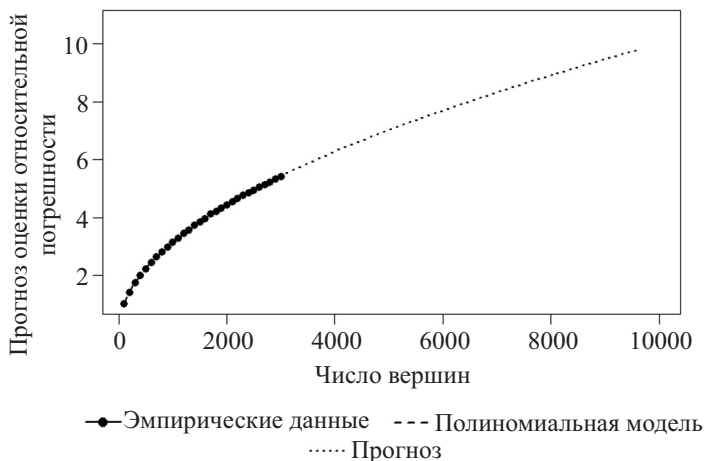


Рис. 6. Прогнозирование оценки относительной погрешности для задач большой размерности для симметричных экземпляров из множества  $\{0, 1, 2, \dots, 10^5\}$ .

Статистический анализ демонстрирует, что предложенная полиномиальная модель хорошо описывает зависимость относительной погрешности от размерности задачи и позволяет выполнять предсказания для больших  $n$ . Хотя модель сохраняет высокую точность аппроксимации, возможные отклонения при экстраполяции указывают на необходимость дальнейшего исследования поведения ошибки для задач экстремально больших размеров. Тем не менее полученные результаты подтверждают, что алгоритм соединения циклов демонстрирует устойчивый характер изменения погрешности и может быть применим к решению симметричной задачи коммивояжера в случае большой размерности.

## 7. Заключение

В статье приведены результаты эмпирического исследования точности и временной сложности алгоритма соединения циклов, а также статистического анализа точности для одного из рассматриваемых семейств симметричной задачи коммивояжера на минимум.

Эмпирический анализ выполнен на семи семействах задач, из которых в статье представлены результаты для пяти семейств случайно сгенерированных экземпляров. Исследование показало, что качество решения и вычислительная сложность алгоритма зависят от структуры матрицы стоимостей. Для асимметричных экземпляров оценка относительной погрешности снижается до 1–2% при 3000 вершинах, тогда как для симметричных экземпляров и экземпляров наклонных плоскостей наблюдается постепенный рост относительной погрешности. Время решения для всех семейств увеличивается полиномиально, при этом для симметричных экземпляров и экземпляров

наклонных плоскостей оно оказывается значительно выше, чем для асимметричных.

Проведенный анализ показывает, что относительная погрешность алгоритма соединения циклов для симметричной задачи коммивояжера демонстрирует полиномиальный рост с увеличением размерности задачи. Это подтверждается статистической обработкой экспериментальных данных, где полиномиальная регрессионная модель обеспечивает наилучшее соответствие эмпирическим наблюдениям.

Несмотря на рост погрешности, предложенный алгоритм сохраняет приемлемую точность и полиномиальное время работы, что делает его перспективным инструментом для решения экземпляров задачи коммивояжера большой размерности. Для дальнейшего повышения качества решений целесообразно исследовать гибридные подходы, сочетающие соединение циклов с дополнительными эвристиками и локальными улучшениями.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Zhang C., Sun P.* Heuristic Methods for Solving the Traveling Salesman Problem (TSP): A Comparative Study // 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). 2023. P. 1–6. <https://doi.org/10.1109/PIMRC56721.2023.10293957>
2. *Toaza B., Esztergár-Kiss D.* A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems // *Appl. Soft Comput.* 2023. V. 148. 110908 p. <https://doi.org/10.1016/j.asoc.2023.110908>
3. *Reingold E.M., Nievergelt J., Deo N.* Combinatorial algorithms: theory and practice. Prentice Hall College Div, 1977. <https://doi.org/10.2307/2987917>
4. *Авдошин С.М., Береснева Е.Н.* Метрическая задача коммивояжера: экспериментальное исследование Парето-оптимальных алгоритмов // Труды Института системного программирования РАН. 2017. 29(4). С. 123–138. [https://doi.org/10.15514/ISPRAS-2017-29\(4\)-8](https://doi.org/10.15514/ISPRAS-2017-29(4)-8)
5. *Flood M.M.* The traveling-salesman problem // *Oper. res.* 1956. V. 4. P. 61–75. <https://doi.org/10.1287/opre.4.1.61>
6. *Rosenkrantz D., Stearns R., Lewis II P.* An analysis of several heuristics for the traveling salesman problem // *SIAM J. Comput.* 1977. V. 6. P. 563–581. <https://doi.org/10.1137/0206041>
7. *Hahsler M., Hornik K.* TSP – Infrastructure for the traveling salesperson problem // *J. Statist. Softwar.* 2007. V. 23. No. 2. <https://doi.org/10.18637/jss.v023.i02>
8. *Kay E., Christofides N.* Graph Theory: An Algorithmic Approach // *Oper. Res. Quarterly.* 1976. V. 27. No. 4. P. 1027. <https://doi.org/10.2307/3009122>
9. *Christofides N.* Worst-case analysis of a new heuristic for the travelling salesman problem // *Graduat. School Industr. Administrat. CMU.* 1976. <https://doi.org/10.1007/s43069-021-00101-z>
10. *Buchin K.* Space-filling curves. Organizing points sets: space-filling curves, delaunay tessellations of random point sets, and flow complexes. Berlin, Free University of Berlin, 2007. P. 5–30.

11. *Bartholdi J., Platzman L., Collins R., Warden W.* A minimal technology routing system for meals on wheels // *Interfaces*. 1983. V. 13. No. 3. P. 1–8.  
<https://doi.org/10.1287/inte.13.3.1>
12. *Aarts E., Lenstra J.* Local search in combinatorial optimization, Princeton, New Jersey: Princeton University Press, 2003. <https://doi.org/10.1515/9780691187563>
13. *Панюков А.В., Леонова Ю.Ф.* Алгоритм соединения циклов для метрической задачи коммивояжера на максимум // *Вест. Южно-Урал. гос. ун-та. Сер. Вычисл. мат. и информатика*. 2021. Т. 10. No. 4. С. 26–36.  
<https://doi.org/10.14529/cmse210402>
14. *Reinelt G.* TSPLIB—A traveling salesman problem library // *ORSA J. Comput.* 1991. V. 3. P. 376–384. <https://doi.org/10.1287/ijoc.3.4.376>.
15. *Леонова Ю.Ф.* Исследование качества алгоритма соединения циклов на примерах из библиотеки TSPLIB / Ю.Ф. Леонова // *Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 13–15 декабря 2021 г. Воронеж, 2022. С. 573–579.*
16. *Heins J., Bossek J., Pohl J., et. al.* A study on the effects of normalized TSP features for automated algorithm selection // *Theoretical Computer Science*. 2023. V. 940. P. 123–145. <https://doi.org/10.1016/j.tcs.2023.01.045>

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

Поступила в редакцию 11.03.2024

После доработки 28.02.2025

Принята к публикации 14.03.2025

# Оптимизация, системный анализ и исследование операций

© 2025 г. М.А. ТУЖИЛИН, канд. физ.-мат. наук (mtu93@mail.ru)  
(Национальный исследовательский университет  
«Высшая школа экономики», Москва;  
Московский государственный университет им. М.В. Ломоносова)

## СООТНОШЕНИЯ МЕЖДУ ДЛИНОЙ СРЕДНЕГО КРАТЧАЙШЕГО ПУТИ И ДРУГИМИ ЦЕНТРАЛЬНОСТЯМИ В ГРАФАХ

Доказываются теоремы о связи между длиной среднего кратчайшего пути в простых связных графах и радиальной центральностью, центральностью по близости, центральностью напряжения и средним кластерным коэффициентом.

*Ключевые слова:* сети, центральности, локальные и глобальные характеристики графов, средний кратчайший путь, средний кластерный коэффициент.

DOI: 10.31857/S0005231025050072, EDN: AWYRUY

### 1. Введение

Длина среднего кратчайшего пути является одной из самых важных характеристик графов реального мира (или сетей). Первые попытки вычисления этой характеристики были предприняты в 1969 г. американскими социальными психологами Стэнли Милгрэмом и Джеффри Трэверсом для графа знакомых [1]. Они обнаружили, что каждый человек опосредованно знаком с любым другим человеком через цепочку из шести знакомств. Эта теория в последствии была названа теорией шести рукопожатий. Далее эта теория была доказана Дунканом Уоттсом и Стивеном Строгацем в статье [2]. Оказалось, что многие графы, получающиеся в реальных прикладных задачах, имеют похожее свойство: имеют небольшую длину среднего кратчайшего расстояния.

Другой характеристикой графов в прикладных задачах является наличие большого среднего кластерного коэффициента или кластерного коэффициента Уоттса и Строгаца [2]. Графы с такими характеристиками (небольшой длиной среднего кратчайшего расстояния и большим средним кластерным коэффициентом) называются сетями малого мира. Кроме длины среднего кратчайшего расстояния и среднего кластерного коэффициента, в теории социальных сетей также выделяют ряд других важных характеристик, которые называются центральностями (например, центральность напряжения, близости, посредничества, радиальная центральность и пр.). Для того чтобы понять, какими внутренними характеристиками обладает граф, полученный в приложениях, анализируются распределения этих центральностей

(см., например, [3–7]), однако наличие зависимостей между центральностями является мало изученным вопросом. В данной статье доказываются теоремы о зависимости между длиной среднего кратчайшего пути и радиальной центральностью, центральностью по близости, центральностью напряжения, а также было получено соотношение между длиной среднего кратчайшего пути и средним кластерным коэффициентом для геодезических графов.

## 2. Основные определения

Все последующие определения даются для простого связного неориентированного графа  $G$ . Введем необходимые обозначения. Обозначим:

- $V(G)$  – множество вершин графа,  $E(G)$  – множество ребер графа,  $A = \{a_{ij}\}$  – матрица смежности графа  $G$ ,
- $N(v)$  – индуцированный подграф в графе  $G$  на вершинах, смежных с вершиной  $v$ ,
- $N'(v)$  – индуцированный подграф в графе  $G$  на вершинах  $V(N(v)) \cup \{v\}$ ,
- $\bar{f}(x_1, \dots, x_k)$  для любой функции  $f : V \times V \times \dots \times V \rightarrow \mathbb{R}$  – ограничение этой функции на подграф  $N'(v)$  (например  $\bar{L}(x, y)$  – среднее кратчайшее расстояние между вершинами  $x$  и  $y$  в подграфе  $N'(v)$ ),
- $d_i = \deg(v_i)$ ,
- $\text{dist}(s, t)$  – длина кратчайшего расстояния между  $s$  и  $t$ ,
- $n = |V(G)|$ ,  $m = |E(G)|$ ,
- $X(i) = X(v_i)$  для любого  $X$  – функции или множества, соответствующего вершине  $v_i$ .

Дадим определения центральностей.

1. Диаметр графа  $\text{diam}(G) = \max_{s, t \in V(G)} \text{dist}(s, t)$ .

2. Длина среднего кратчайшего пути в графе

$$L(G) = \frac{1}{n(n-1)} \sum_{s, t \in V(G), s \neq t} \text{dist}(s, t).$$

3. Локальный кластерный коэффициент

$$c_i = c(i) = \frac{\text{число ребер в подграфе } N(i)}{\text{максимально возможное число ребер в подграфе } N(i)} = \frac{2|E(N(i))|}{d_i(d_i-1)}.$$

4. Средний кластерный коэффициент графа

$$C_{WS}(G) = \frac{1}{n} \sum_{i \in V(G)} c_i = \frac{1}{n} \sum_{i \in V(G)} \frac{2|E(N(i))|}{d_i(d_i-1)} = \frac{1}{n} \sum_{i \in V(G)} \frac{\sum_{j, k \in V(G)} a_{ij} a_{jk} a_{ki}}{d_i(d_i-1)}.$$

5. Центральность по близости  $\text{Clo}(v) = \frac{n-1}{\sum_{t \in V(G)} \text{dist}(v, t)}$ .

6. Радиальная центральность  $\text{Rad}(v) = \frac{\sum_{t \in V(G), t \neq v} (\text{diam}(G)+1-\text{dist}(v, t))}{n-1}$ .

7. Центральность напряжения  $\text{Str}(i) = \sum_{s, t \in V(G), s \neq t \neq i} \sigma_{st}(i)$ , где  $\sigma_{st}(i)$  – число кратчайших путей из  $s$  в  $t$  через вершину  $i$ .

Заметим, что все центральности неотрицательные, а также некоторые из них меньше либо равны 1, а именно:  $c_i, C_{WS}, \text{Clo}(v)$ .

Далее потребуется

*Определение 1.* Граф называется **геодезическим**, если между любыми двумя вершинами существует единственный кратчайший путь.

Это определение эквивалентно условию, когда в графе могут быть только циклы нечетной длины.

### 3. Основные результаты

В общем случае индуцированный подграф  $G' \subset G$  не обязан быть связным. В таком случае можно определить длину среднего кратчайшего расстояния между вершинами подграфа  $G'$  по отношению к расстоянию  $\text{dist}$  в объемлющем графе  $G$ . Назовем  $L(N(i))$  локальной длиной среднего кратчайшего расстояния для вершины  $i$ .

Начнем с доказательства некоторых несложных соотношений: докажем лемму о связи между локальной длиной среднего кратчайшего расстояния и локальным кластерным коэффициентом этой вершины.

*Лемма 1.*

$$L(N(i)) = 2 - c_i.$$

*Доказательство.*

$$\begin{aligned} L(N(i)) &= \frac{1}{d_i(d_i - 1)} \sum_{s,t \in N(i), s \neq t} \text{dist}(s, t) = \\ &= \frac{1}{d_i(d_i - 1)} \sum_{(s,t) \in E(N(i))} \text{dist}(s, t) + \sum_{s,t \in N(i), (s,t) \notin E(N(i))} \text{dist}(s, t) = \\ &= \frac{1}{d_i(d_i - 1)} \left( 2|E(N(i))| + \sum_{(s,i), (i,t) \in E(G), (s,t) \notin E(G)} \text{dist}(s, t) \right) = \\ &= \frac{1}{d_i(d_i - 1)} \left( 2|E(N(i))| + 2(d_i(d_i - 1) - 2|E(N(i))|) \right) = 2 - c_i. \end{aligned}$$

Отметим, что кратчайшее расстояние для вершин из  $N(i)$  считается по всему графу  $G$ .

Усреднением по всем вершинами получаем тривиальное следствие о связи между средним кластерным коэффициентом и средней локальной длиной среднего кратчайшего расстояния.

*Следствие 1.*

$$C_{WS}(G) = 2 - \frac{1}{n} \sum_{i \in V(G)} L(N(i)).$$

Докажем лемму о связи длины среднего кратчайшего расстояния в подграфе  $N'(i)$  с длиной среднего кратчайшего расстояния в  $N(i)$ .

*Лемма 2.*

$$L(N'(i)) = \frac{(d_i - 1)L(N(i)) + 2}{d_i + 1}.$$

*Доказательство.* Распишем определение

$$L(N'(i)) = \frac{1}{(d_i + 1)d_i} \sum_{s,t \in V(N'(i)), s \neq t} \text{dist}(s, t) = \frac{d_i - 1}{d_i + 1} L(N(i)) + \frac{2}{d_i + 1}.$$

Докажем теорему о связи длины среднего кратчайшего расстояния в индуцированном подграфе с длиной среднего кратчайшего расстояния в объемлющем графе, если индуцированный подграф получен из объемлющего графа удалением одной вершины.

*Теорема 1.* Пусть граф  $G'$  получен из простого связного графа  $G'$  удалением одной вершины и  $|V(G)| = n$ . Тогда

$$L(G') \geq \frac{n}{n + 1} L(G),$$

где длина среднего кратчайшего расстояния  $L(G)$  определена в объемлющем графе  $G'$ , если  $G$  не связный.

*Доказательство.* Обозначим удаленную вершину через  $v$ . По неравенству треугольника имеем  $\forall s, t \in V(G) : \text{dist}(s, v) + \text{dist}(v, t) \geq \text{dist}(s, t)$ , где равенство достигается, когда не существует пути из  $s$  в  $t$  в графе  $G$ . Следовательно,

$$\begin{aligned} \sum_{s,t \in V(G), s \neq t} (\text{dist}(s, v) + \text{dist}(v, t)) &\geq \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t), \\ \frac{2(n-1)}{n(n-1)} \sum_{t \in V(G)} \text{dist}(v, t) &\geq \frac{1}{n(n-1)} \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t), \\ \frac{2}{n} \sum_{t \in V(G)} \text{dist}(v, t) &\geq L(G). \end{aligned}$$

Тогда,

$$\begin{aligned} L(G') &= \frac{1}{(n+1)n} \sum_{s,t \in V(G'), s \neq t} \text{dist}(s, t) = \\ &= \frac{1}{(n+1)n} \left( 2 \sum_{t \in V(G)} \text{dist}(v, t) + \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t) \right) = \\ &= \frac{1}{n+1} \frac{2}{n} \sum_{t \in V(G)} \text{dist}(v, t) + \frac{n-1}{n+1} L(G) \geq \frac{n}{n+1} L(G). \end{aligned}$$

Заметим, что если  $G$  состоит из  $n$  изолированных вершин, то неравенство становится равенством.

Получаем следствия.

*Следствие 2.* Пусть граф  $G$  получен из графа  $G' \subset H$  удалением одной вершины и  $|V(G)| = n$ , граф  $H$  связный и простой. Тогда

$$L(G') \geq \frac{n}{n+1}L(G),$$

где длины средних кратчайших расстояний  $L(G)$  и  $L(G')$  определены в объемлющем графе  $H$ , если соответствующие графы не связные.

*Доказательство.* Доказательство аналогичное, как в предыдущей теореме, и равенство также достигается в случае, когда  $G$  состоит из изолированных вершин.

*Следствие 3.*

$$L(N'(i)) \geq \frac{d_i}{d_i+1}L(N(i)).$$

Теперь докажем теорему о связи длины среднего кратчайшего расстояния в индуцированном подграфе с длиной среднего кратчайшего расстояния в объемлющем графе.

*Теорема 2.* Рассмотрим индуцированный подграф  $G' \subset G$ . Пусть  $|V(G)| = n$ ,  $|V(G')| = n+k$ . Тогда

$$L(G') \geq \frac{n}{n+k}L(G),$$

где длина среднего кратчайшего расстояния  $L(G)$  определена в объемлющем графе  $G'$ , если  $G$  не связный.

*Доказательство.* Рассмотрим процесс получения графа  $G'$  из графа  $G$  последовательным добавлением  $k$  вершин и соответствующих ребер. Сначала будем последовательно добавлять вершины, смежные с вершинами графа  $G$ . Добавляя по одной вершине, получим последовательность графов  $G_1, G_2, \dots, G_p$ , где  $|V(G_i)| = n+i$ . Далее будем последовательно добавлять вершины, смежные с вершинами графа  $G_p$ , и получим последовательность  $G_{p+1}, G_{p+2}, \dots$ . В конечном итоге получим граф  $G'$ . По предыдущему следствию имеем

$$\begin{aligned} L(G') &\geq \frac{n+k-1}{n+k}L(G_{k-1}) \geq \frac{n+k-1}{n+k} \frac{n-k-2}{n-k-1}L(G_{k-2}) = \\ &= \frac{n-k-2}{n+k}L(G_{k-2}) \geq \dots \geq \frac{n}{n+k}L(G). \end{aligned}$$

Докажем лемму о связи средней центральности по близости и среднего кратчайшего расстояния в графе.

*Лемма 3.*

$$L(G) \geq \frac{n}{\sum_{v \in V(G)} \text{Clo}(v)}.$$

*Доказательство.* Воспользуемся неравенством о среднем гармоническом и арифметическом:

$$\frac{1}{n} \sum_{v \in V(G)} \text{Clo}(v) = \frac{1}{n} \sum_{v \in V(G)} \frac{n-1}{\sum_{t \in V(G)} \text{dist}(v, t)} \geq \frac{n(n-1)}{\sum_{v, t \in V(G)} \text{dist}(v, t)} = \frac{1}{L(G)}.$$

Заметим, что равенство выполнено, когда все средние кратчайшие расстояния от каждой вершины до всех других равны.

Докажем лемму о связи длины среднего кратчайшего расстояния в графе со средней радиальной центральностью.

*Лемма 4.*

$$L(G) = \text{diam}(G) + 1 - \frac{1}{n} \sum_{v \in V(G)} \text{Rad}(v).$$

*Доказательство.* Доказательство следует напрямую из определения:

$$\begin{aligned} \frac{1}{n} \sum_{v \in V(G)} \text{Rad}(v) &= \frac{1}{n} \sum_{v \in V(G)} \frac{(n-1)(\text{diam}(G) + 1) - \sum_{t \in V(G), t \neq v} \text{dist}(v, t)}{n-1} = \\ &= \text{diam}(G) + 1 - L(G). \end{aligned}$$

Докажем теорему о связи между длиной среднего кратчайшего расстояния в графе и средней центральностью напряжения для геодезических графов.

*Теорема 3.* Если  $G$  – геодезический граф, то

$$L(G) = 1 + \frac{1}{n(n-1)} \sum_{i \in V(G)} \text{Str}(i).$$

*Доказательство.* Определим функцию

$$\chi_{st}(i) = \begin{cases} 1, & \text{если } i \neq s \neq t - \text{вершина кратчайшего пути между } s \text{ и } t, \\ 0, & \text{в остальном.} \end{cases}$$

Следовательно, для центральности напряжения  $\text{Str}(i) = \sum_{s, t \in V(G)} \chi_{st}(i)$ .

Если в графе  $G$  между любыми двумя вершинами существует единственный кратчайший путь, то  $\text{dist}(s, t) = \sum_{i \in V(G)} \chi_{st}(i) + 1$  (в противном случае

$\text{dist}(s, t) \leq \sum_{i \in V(G)} \chi_{st}(i) + 1$ ). Следовательно, для любого  $i$

$$\begin{aligned} \sum_{s, t \in V(G)} \text{dist}(s, t) &= 2|E| + \sum_{s, t \in V(G), \text{dist}(s, t) \geq 2} \text{dist}(s, t) = \\ &= 2|E| + \sum_{s, t \in V(G), \text{dist}(s, t) \geq 2} \left( \sum_{i \in V(G)} \chi_{st}(i) + 1 \right) = \\ &= 2|E| + \sum_{i \in V(G)} \text{Str}(i) + n(n-1) - 2|E| = \sum_{i \in V(G)} \text{Str}(i) + n(n-1). \end{aligned}$$

Получаем

*Следствие 4. Для любого просто связного графа  $G$*

$$L(G) \leq 1 + \frac{1}{n(n-1)} \sum_{i \in V(G)} \text{Str}(i).$$

Для получения соотношения между длиной среднего кратчайшего расстояния и средним кластерным коэффициентом сначала докажем теорему о связи среднего кластерного коэффициента с центральностью напряжения.

*Теорема 4. Для простого связного графа  $G$  без висячих вершин*

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left( 1 - \frac{\text{Str}(i)}{d_i(d_i-1)} \right).$$

*Доказательство.* Заметим, что  $\forall j, k \in N(i) : (j, k) \notin E(N(i))$  кратчайшее расстояние между  $j$  и  $k$  – это  $j \rightarrow i \rightarrow k$ . Тогда

$$\begin{aligned} \text{Str}(i) &\geq 2 \left( \frac{d_i(d_i-1)}{2} - |E(N(i))| \right), \\ \frac{1}{d_i(d_i-1)} \text{Str}(i) &\geq 1 - c_i. \end{aligned}$$

Усреднением по  $i$  получаем:

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left( 1 - \frac{\text{Str}(i)}{d_i(d_i-1)} \right).$$

Заметим, что равенство достигается, если  $\text{diam}(G) = 2$ .

*Следствие 5. Для любого простого связного графа  $G$*

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left( 1 - \frac{\text{Str}(i)}{d_i(d_i-1)} \right) - \frac{\text{число висячих вершин}}{n}.$$

*Доказательство.* Для каждой висячей вершины положим  $\frac{\text{Str}(i)}{d_i(d_i-1)} = 0$ , тогда в правой части неравенства  $\frac{1}{d_i(d_i-1)}\text{Str}(i) \geq 1 - c_i$  будет 1, а в левой 0. Если добавим в левую часть 1 для каждой висячей вершины, то получим верное равенство.

Теперь докажем теорему о связи между длиной среднего кратчайшего расстояния и средним кластерным коэффициентом.

*Теорема 5.* Если  $G$  – геодезический граф и  $\forall i, j \in V(G)$  выполнено, если  $d_i \leq d_j$ , то  $\text{Str}(i) \leq \text{Str}(j)$ , тогда

$$1 - C_{WS}(G) \leq \frac{1}{n} \sum_{i \in V(G)} \frac{(L(G) - 1)(n - 1)}{d_i(d_i - 1)} + \frac{\text{число висячих вершин}}{n}.$$

*Доказательство.* Перенумеруем вершины так, что  $\forall i \leq j : d_i \leq d_j$ . Тогда для  $i \leq j$  выполнено  $\text{Str}(i) \leq \text{Str}(j)$  и  $d_i(d_i - 1) \leq d_j(d_j - 1)$ . По теореме 3 и следствию 5, если в графе нет висячих вершин, получаем

$$\begin{aligned} 1 - C_{WS}(G) &\leq \frac{1}{n} \sum_{i \in V(G)} \frac{\text{Str}(i)}{d_i(d_i - 1)} \stackrel{\text{неравенство Чебышева}}{\leq} \\ &\stackrel{\text{неравенство Чебышева}}{\leq} \left( \frac{1}{n} \sum_{i \in V(G)} \text{Str}(i) \right) \left( \frac{1}{n} \sum_{i \in V(G)} \frac{1}{d_i(d_i - 1)} \right) = \\ &= \frac{1}{n} \sum_{i \in V(G)} \frac{(L(G) - 1)(n - 1)}{d_i(d_i - 1)}. \end{aligned}$$

В случае висячих вершин справа добавится слагаемое  $\frac{\text{число висячих вершин}}{n}$ . Отметим, что если существуют две вершины  $i, j \in V(G)$  такие, что  $d_i < d_j$  и  $\text{Str}(i) < \text{Str}(j)$ , то неравенство в этой теореме будет строгое.

*Пример 1.* Рассмотрим граф звезды с  $|V(G)| = n + 1$  вершиной. Это геодезический граф. Центральная вершина имеет степень  $n$ , локальный кластерный коэффициент  $c_i = 0$  и центральность стресса  $\text{Str}(i) = n(n - 1)$ . Все остальные вершины висячие ( $d_i = 1$ ,  $c_i = 0$ ,  $\text{Str}(i) = 0$ ), поэтому для этого графа выполнено условие теоремы 5.

$$\begin{aligned} L(G) &= \frac{n(2n - 1) + n}{(n + 1)n} = \frac{2n}{n + 1}, \quad C_{WS} = 0. \\ 1 - C_{WS}(G) &= 1 = \frac{\frac{n-1}{n+1}n}{n(n-1)} + \frac{n}{n+1} = \\ &= \frac{1}{n+1} \sum_{i \in V(G)} \frac{(L(G) - 1)n}{d_i(d_i - 1)} + \frac{\text{число висячих вершин}}{n+1}. \end{aligned}$$

Следовательно, для этого графа выполнено равенство.

## СПИСОК ЛИТЕРАТУРЫ

1. *Travers J., Milgram S.* An experimental study of the small world problem / Social networks. Academic Press. 1977. P. 179–197.
2. *Watts D.J., Strogatz S.H.* Collective dynamics of ‘small-world’ networks // Nature. 1998. V. 393. No. 6684. P. 440–442.
3. *Borgatti S.P., Everett M.G.* A graph-theoretic perspective on centrality // Social networks. 2006. V. 28. No. 4. P. 466–484.
4. *Kiss C., Bichler M.* Identification of influencers-measuring influence in customer networks // Decision Support Systems. 2008. V. 46. No. 1. P. 233–253.
5. *Lee S.H.M., Cotte J., Noseworthy T.J.* The role of network centrality in the flow of consumer influence // J. Consum. Psychol. 2010. V. 20. No. 1. P. 66–77.
6. *You J., Gomes-Selman J.M., Ying R., et al.* Identity-aware graph neural networks // Proceedings of the AAAI conference on artificial intelligence. 2021. V. 35. No. 12. P. 10737–10745.
7. *Yuan M.M., Guo X., Wu L., et al.* Climate warming enhances microbial network complexity and stability // Nature Climate Change. 2021. V. 11. No. 4. P. 343–348.

*Статья представлена к публикации членом редколлегии А.Г. Кушнером.*

Поступила в редакцию 16.12.2024

После доработки 30.12.2024

Принята к публикации 09.01.2025

---

---

## СОДЕРЖАНИЕ

### Стохастические системы

- Назаров А.А., Измайлова Я.Е.** Асимптотически-диффузионный анализ приоритетной RQ-системы  $M^{(2)}|M^{(2)}|1$ .....3

### Робастное, адаптивное и сетевое управление

- Гуков А.А.** Адаптивное управление по выходу с гарантией заданного качества регулирования ..... 26

### Управление в технических системах

- Ефанов Д.В., Пивоваров Д.В.** Синтез самопроверяемых дискретных устройств на основе полиномиальных кодов с контролем вычислений по нескольким диагностическим признакам ..... 39

### Интеллектуальные системы управления, анализ данных

- Варнавский А.Н.** Регрессионные модели для игрового чат-бота с искусственным интеллектом для обучения программированию на основе головоломок типа Wordle ..... 61

### Оптимизация, системный анализ и исследование операций

- Афраймович Л.Г., Емелин М.Д.** Построение области Парето при комбинировании допустимых решений многокритериальной аксиальной задачи о назначениях..... 81
- Леонова Ю.Ф., Панюков А.В.** Статистическое исследование качества алгоритма соединения циклов для решения задачи коммивояжера на минимум... 98
- Тужилин М.А.** Соотношения между длиной среднего кратчайшего пути и другими центральностями в графах..... 114

## C O N T E N T S

### Stochastic Systems

<b>Nazarov A.A., Izmailova Ya.E.</b> Asymptotic-Diffusion Analysis of the Retrieval Queueing System $M^{(2)} M^{(2)} 1$ with Priority Customers for a Non-Priority Component .....	3
--	---

### Robust, Adaptive and Network Control

<b>Gukov A.A.</b> Adaptive Output Control with Guaranteed Control Quality .....	26
---	----

### Control in Technical Systems

<b>Efanov D.V., Pivovarov D.V.</b> Synthesis of Self-Checking Discrete Devices Based on Polynomial Codes with Calculation Testing According to Several Diagnostic Signs .....	39
---	----

### Intellectual Control Systems, Data Analysis

<b>Varnavsky A.N.</b> Regression Models for the AI Gaming Chatbot for Learning Programming Based on Wordle-Type Puzzles .....	61
---	----

### Optimization, System Analysis, and Operations Research

<b>Afraimovich L.G., Emelin M.D.</b> Construction of the Pareto Front When Combining Feasible Solutions of a Multi-Criteria Axial Assignment Problem .....	81
<b>Leonova Yu.F., Panyukov A.V.</b> Statistical Study of the Quality of the Cycle Merging Algorithm for Solving the Traveling Salesman Problem at Minimum .....	98
<b>Tuzhilin M.A.</b> Relations between Average Shortest Path Length and Another Centralities in Graphs .....	114