

## Определение зависимостей API специализированного программного обеспечения при максимально возможной нагрузке без существенной деградации системы

Л.И. Литвиненко, Л.А. Коробова, И.С. Толстова

Воронежский государственный университет инженерных технологий,  
394036, г. Воронеж, проспект Революции, д. 19, Россия

**Резюме. Цель.** Целью исследования является определение зависимости API специализированного программного обеспечения при максимально возможной нагрузке без существенной деградации системы. **Метод.** Исследование основано на методах регрессионного анализа. Рассматриваются ограничения по объёму оперативной памяти (RAM) и количеству процессорных ядер (CPU), что позволяет исследовать поведение и выявлять узкие места СПО в виртуализированных окружениях с ограниченными ресурсами. **Результат.** Для анализа зависимостей ожидаемых пар параметров предлагается использование моделей - линейной, экспоненциальной, логарифмической и полиномиальной (до 4-й степени). Выбор наилучшей модели предлагается по критерию скорректированного коэффициента детерминации с опциональной фильтрацией выбросов методом межквартильного размаха (IQR) и проверкой статистической значимости уравнения регрессии. Дополнительно рассчитывается выборочная остаточная дисперсия для оценки качества подгонки модели. **Вывод.** Методика представляет системный подход к анализу зависимостей производительности API специализированного программного обеспечения, основанный на принципах прикладной статистики и инженерной интерпретации. Методика может быть использована для принятия обоснованных инфраструктурных решений, а также для рационального сокращения объёма нагрузочного тестирования; позволяет транслировать статистические паттерны в конкретные инженерные действия: оптимизацию инфраструктуры, рефакторинг кода, настройку масштабирования.

**Ключевые слова:** специализированный софт, нагрузочное тестирование, регрессионные модели, коэффициент детерминации, скорректированный коэффициент детерминации; линейная, экспоненциальная, логарифмическая, полиномиальная регрессии; диагностика моделей, парный регрессионный анализ

**Для цитирования:** Л.И. Литвиненко, Л.А. Коробова, И.С. Толстова. Определение зависимостей API специализированного программного обеспечения при максимально возможной нагрузке без существенной деградации системы. Вестник Дагестанского государственного технического университета. Технические науки. 2026;53(1):116-122. DOI:10.21822/2073-6185-2026-53-1-116-122.

## Determination of specialized software API dependencies under maximum possible load without significant system degradation

L.I. Litvinenko, L.A. Korobova, I.S. Tolstova

Voronezh State University of Engineering Technologies,  
19 Revolution Ave., Voronezh 394036, Russia

**Abstract. Objective.** The article considers the topic of determining the dependencies of the API of specialized software under the maximum possible load without significant system degradation. **Method.** The study is based on regression analysis methods. As part of the research, limitations on the amount of random access memory and the number of processor cores are considered, which allows us to study the behavior and identify bottlenecks of SSS in virtualized environments with limited resources. **Result.** To analyze the dependencies of expected pairs of parameters, it is

proposed to use the following models: linear, exponential, logarithmic and polynomial (up to 4th degree). The best model is selected based on the adjusted coefficient of determination, with optional outlier filtering using the interquartile range method and a test of the statistical significance of the regression equation. Additionally, sample variance is calculated to assess the quality of the model fit. **Conclusion.** This methodology presents a systematic approach to analyzing API performance dependencies in specialized software, based on the principles of applied statistics and engineering interpretation. It can be used to make informed decisions about reducing the scope of load testing. The methodology enables the translation of statistical patterns into engineering actions: infrastructure optimization, code refactoring, and scaling.

**Keywords:** specialized software, load testing, regression models, coefficient of determination, adjusted coefficient of determination; linear, exponential, logarithmic, and polynomial regression; model diagnostics, paired regression analysis

**For citation:** L.I. Litvinenko, L.A. Korobova, I.S. Tolstova. Determination of specialized software API dependencies under maximum possible load without significant system degradation. Herald of Daghestan State Technical University. Technical Sciences. 2026;53(1):116-122. (In Russ) DOI:10.21822/2073-6185-2026-53-1-116-122.

**Введение.** Определение зависимостей API специализированного программного обеспечения (СПО) при максимально возможной нагрузке без существенной деградации системы является важной задачей, поскольку позволяет выявить, какие входные параметры системы (ресурсы и метрики производительности) влияют на ключевые показатели работы системы под нагрузкой, а также прогнозировать поведение системы при росте нагрузки, что способствует принятию обоснованных инфраструктурных решений.

**Постановка задачи.** В математической статистике регрессионный анализ является ключевым инструментом в задачах моделирования зависимостей между переменными [1]. Выбор наилучшей модели - линейной, полиномиальной, экспоненциальной или логарифмической, остаётся сложной задачей, особенно при отсутствии априорных предположений о форме зависимости. Выбор модели основывается на коэффициенте детерминации  $R^2$ , однако этот показатель склонен к завышению при увеличении числа параметров (например, степени полинома), что может привести к переобучению. Информационные критерии, такие как AIC (критерий Акаике), вводят штраф за количество параметров, что способствует выбору более простых и интерпретируемых моделей. Однако, они основаны на асимптотических приближениях и не всегда гарантируют наилучшую обобщающую способность на новых данных. Кросс-валидация (CV) позволяет напрямую оценить устойчивость модели к переобучению и её предсказательную способность, но требует значительных вычислительных ресурсов, особенно при большом числе моделей [2].

Цель исследования состоит в разработке методики определения зависимостей API СПО при максимально возможной нагрузке без существенной деградации системы с использованием методов регрессионного анализа с последующей интерпретацией физического смысла зависимости в контексте архитектуры и поведения системы.

**Методы исследования.** Методика включает следующие этапы.

Этап 1. Сбор данных. Требуется провести нагрузочное тестирование приложения в ограниченной среде (контейнер, виртуальная машина) с различными комбинациями выделенных ресурсов (количество ядер CPU и объем RAM). Определяется максимально возможная нагрузка в текущей конфигурации за заданный период времени. Критерии деградации выбрать опционально - это может быть как процент ошибок, так и время отклика (*latency*), превышающее заранее установленный порог, или комбинация нескольких метрик, таких как задержка, ошибка (*rate*) и пропускная способность (*throughput*). Для обеспечения воспроизводимости тестов необходимо стандартизировать тестовые сценарии и исключить внешние факторы, влияющие на результаты (сетевые задержки, фоновые процессы) [3].

Этап 2. Полученные результаты на этапе 1 переносятся в файл формата csv для дальнейшего использования в рамках регрессионного анализа [4] на этапе 3.

Этап 3. Для регрессионного анализа данных предлагается разработать, реализовать и далее использовать приложения [5], в основу которого положен пошаговый алгоритм:

Шаг 1. Загрузка и парсинг CSV-файла с данными. После выбора пользователем предполагаемой пары  $X$ - $Y$  выполняется парсинг данных *cpuCores*, *ramCores*, *qSuccess*, *tAvg*, *tps*, *uCpu*, *uRam*, полученных на шаге 1 и 2, с добавлением проверок на корректность формата и игнорирование строк с ошибками [6]. Возможные типы переменных указаны в табл. 1.

Таблица 1. Парсинг CSV файла с результатами  
 Table 1. Parsing the CSV file with results

Переменная Variable	Возможный тип в реализации Possible type in implementation	Описание Description
cpuCores	Int	Количество выделенных ядер CPU
ramCores	Double	Объём оперативной памяти, обусловлено, что часть
qSuccess	Double	Доля успешных запросов ( <i>QoS — Quality of Service</i> )
tAvg	Double	Среднее время обработки одного запроса ( <i>latency</i> )
Tps	Double	Пропускная способность: транзакций в секунду ( <i>Throughput</i> )
uCpu	Double	Утилизация CPU (%)
uRam	Double	Утилизация RAM (%)

Шаг 2. Выполнить регрессионное моделирование. Если на предыдущих шагах была выбрана фильтрация выбросов, то выполняется фильтрация выбросов по межквартильному размаху (IQR) для  $X$  и  $Y$  одновременно, что исключает аномальные точки, вызванные сетевыми задержками или ошибками измерения [7, 8].

Шаг 3. Провести оценку модели, ранее созданной по каждой комбинации (табл. 2).

Таблица 2. Предлагаемые комбинации параметров для регрессионного моделирования  
 Table 2. Suggested parameter combinations for regression modeling

X	Y	Обоснование выбора комбинации /Justification for the choice of combination
cpuCores	Tps	Количество CPU-ядер напрямую влияет на параллельную обработку запросов. Увеличение ядер позволяет системе обрабатывать больше операций в единицу времени (TPS). Зависимость позволяет оценить масштабируемость системы по CPU и определить точку насыщения, после которой добавление ядер не даёт прироста производительности.
ramCores	tAvg	Объём выделенной RAM влияет на производительность, особенно если приложение интенсивно использует память (кеширование, обработка больших объектов). Недостаток памяти приводит к свопингу, GC-паузам, промахам кэша, что увеличивает среднее время ответа (tAvg). Эта комбинация помогает выявить оптимальное соотношение RAM/CPU.
tps	tAvg	С ростом нагрузки (TPS) система приближается к пределу своей пропускной способности. В соответствии с законом Литтла и теорией массового обслуживания, при увеличении входного потока время ожидания в очереди растёт, что приводит к увеличению tAvg. Эта зависимость критична для определения предела устойчивой работы системы и точки деградации (например, когда tAvg резко возрастает).
uCpu	tAvg	Загрузка CPU (uCpu) - индикатор интенсивности использования процессора. При высокой загрузке (>80–90%) система начинает тратить время на переключение контекстов, обработку прерываний, что замедляет обработку запросов. Связь помогает определить порог CPU-насыщения, при котором начинается деградация времени отклика.
Tps	qSuccess	С ростом нагрузки (TPS) система может не успевать обрабатывать запросы, что приводит к таймаутам, отказам бэкендов, переполнению очередей. Это снижает долю успешных запросов (qSuccess). Эта зависимость позволяет оценить надёжность системы под нагрузкой и определить максимальный безопасный уровень нагрузки без потери качества сервиса.
uCpu	qSuccess	Высокая загрузка CPU может привести к нехватке ресурсов для обработки запросов, особенно если приложение чувствительно к задержкам. Это может вызывать таймауты на уровне приложения или прокси, что снижает qSuccess. Связь помогает понять, насколько стабильность системы зависит от CPU-ограничений и выявить «узкие места» в коде (например, CPU-интенсивные алгоритмы).

В табл. 2 описан пример выбора комбинаций параметров  $X$  и  $Y$  для регрессионного моделирования для оценки производительности API СПО под нагрузкой [7, 8]. Предлагается реализация с возможностью выбора пользователем переменных  $X$  и  $Y$ , после выбора пользователем значений переменных, инициировать обучение нескольких моделей регрессии: линейной, полиномиальной разных степеней от 1 до 4, экспоненциальной и логарифмической.

Шаг 4. Провести диагностику и выбор лучшей модели [3, 9]. Алгоритм на данном шаге включает в себя следующие пункты:

1. Рассчитать коэффициент детерминации  $R^2$ .

$$R^2 = 1 - \frac{e' \cdot e}{y' \cdot y},$$

где  $e' \cdot e$  – скалярное произведение вектора остатков на себя, значение равно сумме квадратов остатков;  $y' \cdot y$  – скалярное произведение вектора  $y$  на себя, значение равно сумме квадратов наблюдаемых значений.

2. Произвести фильтрацию на основании критерия значимости уравнения регрессии [5, 6, 9].

$$F = \frac{R^2 \cdot (n - p - 1)}{(1 - R^2) \cdot p} > F(\alpha, k_1, k_2),$$

где  $R^2$  – коэффициент детерминации;  $n$  – количество наблюдений в выборке;  $p$  – количество независимых переменных (предикатов) в модели;  $F(\alpha, k_1, k_2)$  – значение  $F$ -критерия Фишера-Снедекора, где  $\alpha$  – степень значимости критерия;  $k_1 = p$ ,  $k_2 = n - p - 1$ .

3. Произвести выбор лучшей модели. Среди значимых моделей необходимо выбрать ту, у которой скорректированный коэффициент детерминации имеет максимальное значение. При равенстве – предпочтение модели с меньшим числом параметров. Затем выполнить расчёт скорректированного коэффициента детерминации  $\hat{R}^2$  [5, 6, 9].

$$\hat{R}^2 = 1 - \frac{n - 1}{n - p - 1} \cdot (1 - R^2)$$

4. Выполнить расчет выборочной остаточной дисперсии. Полученные результаты поместить в таблицу (рис. 1). Выборочная остаточная дисперсия [7, 9]:

$$s^2 = \frac{e' \cdot e}{n - p - 1} = \frac{\sum_{i=1}^n e_i^2}{n - p - 1}$$

где  $e$  – вектор остатков;  $i$  – текущий номер наблюдения зависимой переменной.

Вектор остатков – разница между наблюдаемыми значениями зависимой переменной и значениями, предсказанными моделью. На рис. 1 представлен пример возможной реализации табличной части приложения.

Регрессионный анализ параметров нагрузки

Результаты моделей

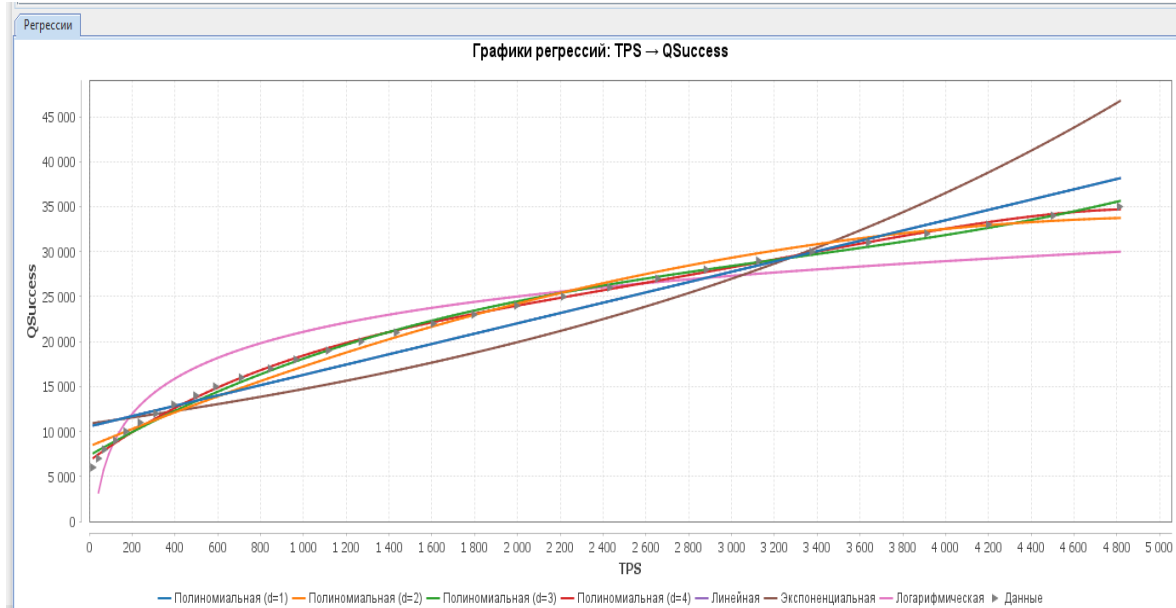
X: TPS Y: QSuccess  Фильтровать выбросы (IQR по X и Y) Анализ парной регрессии

Модель	Коэффициент детерминации	Скорректированный коэффициент детерми...	Несмещенная оценка дисперсии остатков	Уравнение значимо (F)
Полиномиальная (d=4)	0,9986	0,9983	129110,278486	Да
Полиномиальная (d=3)	0,9963	0,9959	320206,131248	Да
Полиномиальная (d=2)	0,9888	0,9880	933435,312882	Да
Полиномиальная (d=1)	0,9477	0,9459	4195959,588554	Да
Линейная	0,9477	0,9459	4195959,588554	Да
Логарифмическая	0,8910	0,8871	8747012,627462	Да
Экспоненциальная	0,7820	0,7742	17495960,971229	Да

Рис. 1 – Пример реализации табличной части приложения (значения в табличной части рассчитаны на синтетических данных, специально созданных для целей отладки)

Fig. 1 – Example of the application's table section implementation (The values in the table section are calculated using synthetic data specifically created for debugging purposes)

На рис. 2 представлен пример возможной реализации графической части приложения.



**Рис. 2 – Пример реализации графической части приложения (графики построены на синтетических данных, специально созданных для целей отладки)**  
**Fig. 2 – An example of the implementation of the graphical part of the application (graphs are built on synthetic data, specially created for debugging purposes)**

Этап 4. Анализ результатов, полученных на предыдущих этапах.

Выбор наилучшей формы регрессионной зависимости позволяет отразить физические закономерности поведения системы под нагрузкой: линейная модель соответствует пропорциональному росту/падению без насыщения; полиномиальная модель (степень  $\geq 2$ ) указывает на наличие нелинейных эффектов насыщения, ускорения или замедления роста; экспоненциальная модель характерна для систем с резким ростом/падением показателей; логарифмическая модель отражает затухающий эффект. Вероятные типы регрессионных моделей и возможная интерпретация представлены в табл. 3.

**Таблица 3. Вероятные типы регрессионных моделей и возможная интерпретация зависимостей в системе под нагрузкой**  
**Table 3. Probable types of regression models and possible interpretation of dependencies in the system under load**

$X, Y$	Вероятная лучшая модель Probable best model	Возможная интерпретация результатов Possible interpretation of results
$X = \text{cpuCores},$ $Y = \text{tps}$	Полиномиальная или логарифмическая Polynomial or logarithmic	Полиномиальная модель — система достигает предела масштабируемости по CPU. Линейная модель - система эффективно использует дополнительные ядра, что указывает на отсутствие блокировок и IO-ограничений. При нелинейности требуется искать узкие места в БД, сети или синхронизации.
$X = \text{ramCores},$ $Y = \text{tAvg}$	Экспоненциальная или полиномиальная Exponential or polynomial	Экспоненциальная модель указывает на сильное влияние нехватки памяти (свопинг, GC). Линейная модель - влияние RAM незначительно. Определить оптимальный порог ramCores, после которого tAvg стабилизируется. При экспоненциальной зависимости: увеличить RAM критично.
$X = \text{tps},$ $Y = \text{tAvg}$	Экспоненциальная или полиномиальная Exponential or polynomial	Экспоненциальная модель подтверждает закон теории массового обслуживания (M/M/1). Низкий $R^2$ может указывать на нестабильность системы. Определить предел устойчивой нагрузки (например, TPS при tAvg < 1000 мс). Установить ограничения на входной поток (rate limiting).
$X = \text{uCpu},$ $Y = \text{tAvg}$	Полиномиальная Polynomial	Нелинейность указывает на пороговое поведение. Линейная модель - система устойчива даже при высокой загрузке. Поддерживать uCpu < 75% в продакшене. Оптимизировать CPU-интенсивные участки кода.

$X, Y$	<b>Вероятная лучшая модель</b> <b>Probable best model</b>	<b>Возможная интерпретация результатов</b> <b>Possible interpretation of results</b>
$X = T_{ps}$ $Y = q_{Success}$	Логарифмическая или полиномиальная Logarithmic or polynomial	Логарифмическая модель - плавная деградация. Полиномиальная - резкое падение (например, отказ бэкенда). Определить максимальный безопасный TPS ( $q_{Success} > 99\%$ ). Реализовать circuit breaker или fallback-логику.
$X = u_{Cpu}$ , $Y = q_{Success}$	Полиномиальная Polynomial	Нелинейное падение указывает на критическую зависимость надёжности от CPU. Внедрить автомасштабирование при высокой загрузке CPU. Оптимизировать алгоритмы (например, кэширование, асинхронная обработка).

**Вывод.** Предложенная методика представляет системный подход к анализу зависимостей производительности API специализированного программного обеспечения (СПО), основанный на принципах прикладной статистики и инженерной интерпретации. Она может быть использована для принятия обоснованных инфраструктурных решений, а также для рационального сокращения объёма нагрузочного тестирования.

Методика опирается на сравнение скорректированного коэффициента детерминации для нескольких регрессионных моделей, при этом предварительно выполняется опциональная фильтрация выбросов по межквартильному размаху (IQR) - одновременно для обеих переменных ( $X$  и  $Y$ ).

В анализ включаются только модели, для которых уравнение регрессии статистически значимо (по F-критерию), а также дополнительно рассчитывается выборочная остаточная дисперсия, для оценки точности подгонки модели к данным.

Подход позволяет обоснованно связывать статистические результаты с архитектурными ограничениями системы, опираясь на инженерные гипотезы о её внутренних механизмах.

Методика позволяет транслировать статистические паттерны в конкретные инженерные действия: оптимизацию инфраструктуры, рефакторинг кода, настройку масштабирования.

Предварительные гипотезы о характере зависимостей позволяют осознанно сокращать объём нагрузочного тестирования.

Например, вместо проведения полного цикла тестов на всех возможных конфигурациях, можно построить статистическую модель и экстраполировать поведение системы, ограничившись проверкой в нескольких контрольных точках для верификации модели. Это существенно снижает затраты на тестирование без потери точности прогнозов.

#### **Библиографический список:**

1. Черняева С.Н., Коробова Л.А., Бугаев Ю.В. Алгоритм коллективного выбора на необозримом для лица принимающего решения множестве альтернатив //Инженерные технологии. – 2024. – № 1(5). – С. 22-27. – EDN JMQXRJ.
2. Гмурман В.Е. Теория вероятностей и математическая статистика: учеб. пособие / В.Е. Гмурман. — 12-е изд., перераб. и доп. — М.: Высшее образование, 2006. — 479 с.
3. Черняева С.Н., Коробова Л.А. Анализ вырожденности матриц экспертного упорядочивания в задачах коллективного выбора//Моделирование энергоинформационных процессов: Сборник статей XII национальная научно-практическая конференция с международным участием.– Воронеж: Воронежский государственный университет инженерных технологий, 2024. – С. 213-224. – EDN IDJAIF.
4. Кремер Н.Ш., Путко Б.А. Эконометрика. — М.: ЮНИТИ-ДАНА, 2002. — 315 с.
5. Ивлиев, М.Н. Финансовая математика. Методы и модели в экономике. Сборник задач: учебное пособие / М.Н. Ивлиев, Л.А. Коробова, К.В. Чекудаев. – Воронеж: Воронежский государственный университет инженерных технологий, 2019. – 91 с. – ISBN 978-5-00032-444-8. – EDN MHQVNZ.
6. Корчуганова М.А. Лабораторный практикум по дисциплине «Эконометрика»: учебное пособие Томск: Изд-во Томского политехнического университета, 2016. — 91 с.
7. Implementation of the Extrapolation Method of Expert Assessments in Selection Problems / S. Chernyaeva, L. Korobova, M. Ivliev [et al.] // High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production, Barnaul, 15–16 мая 2020 года / Vladimir Jordan Nikolay Filimonov Ilya Tarasov Vladimir Faerman (Eds.). Vol. 1304. – Barnaul: Springer, 2020. – P. 147-161. – DOI 10.1007/978-3-030-66895-2\_10. – EDN HINSXG.

8. Хныкин, Д.Е. Методология и результаты применения обратного инжиниринга / Д. Е.Хныкин, Л.А. Коробова // Инженерные технологии. – 2025. – № 1(9). – С. 30-35. – EDN UVTHPV.
9. Бугаев Ю.В., Л.А. Коробова, Шурупова И.Ю. О статистической устойчивости оптимального решения, найденного по уравнению регрессии//Вестник Воронежского государственного университета инженерных технологий. – 2024. – Т.86, № 2(100). – С. 48-55. DOI 10.20914/2310-1202-2024-2-48-55.

#### References:

1. Chernyaeva S.N., L.A. Korobova, Yu.V. Bugaev. Algorithm of collective choice on a set of alternatives that is boundless for the decision maker. *Engineering technologies*. 2024;1(5): 22-27. - EDN JMQXRJ.
2. Gmurman, V.E. Probability theory and mathematical statistics: textbook. - 12th ed., revised and enlarged. - Moscow: Higher education, 2006; 479 p.
3. Chernyaeva S.N., Korobova L.A. Analysis of the degeneracy of expert ordering matrices in collective choice problems. Modeling of energy-information processes: Collection of articles of the XII national scientific and practical conference with international participation. - Voronezh: Voronezh State University of Engineering Technologies, 2024; 213-224. - EDN IDJAIF.
4. Kremer N.Sh., Putko B.A. Econometrics. - Moscow: UNITY-DANA, 2002. - 315 p.
5. Ivliev, M.N. Financial Mathematics. Methods and Models in Economics. Collection of problems: textbook / M.N. Ivliev, L.A. Korobova, K.V. Chekudaev. – Voronezh: Voronezh State University of Engineering Technologies, 2019; 91 p. – ISBN 978-5-00032-444-8. – EDN MHQVNZ.
6. Korchuganova M.A. Laboratory practical training in the discipline "Econometrics": Tomsk: Publishing house of Tomsk Polytechnic University, 2016; 91 p.
7. Implementation of the Extrapolation Method of Expert Assessments in Selection Problems / S. Chernyaeva, L. Korobova, M. Ivliev [et al.] High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production, Barnaul, May 15-16, 2020 / Vladimir Jordan Nikolay Filimonov Ilya Tarasov Vladimir Faerman (Eds.) – Barnaul: Springer, 2020;1304: 47-161. – DOI 10.1007/978-3-030-66895-2\_10. – EDN HINSXG.
8. Khnykin, D.E. Methodology and results of applying reverse engineering / D.E. Khnykin, L.A. Korobova *Engineering technologies*. 2025;1(9):30-35. – EDN UVTHPV.
9. Bugayev Yu.V., Korobova L.A., Shurupova I.Yu. On the statistical stability of the optimal solution found by the regression equation *Bulletin of the Voronezh State University of Engineering Technologies*. 2024;86(2(100)):48-55. – DOI 10.20914/2310-1202-2024-2-48-55. – EDN IYNADI.

#### Сведения об авторах:

Литвиненко Лилия Валериевна, магистрант, кафедра информационных технологий, моделирования и управления; [litvinenko.lilia01@mail.ru](mailto:litvinenko.lilia01@mail.ru)

Коробова Людмила Анатольевна, кандидат технических наук, доцент, доцент, кафедра информационных технологий, моделирования и управления; [Lyudmila\\_korobova@mail.ru](mailto:Lyudmila_korobova@mail.ru)

Толстова Ирина Сергеевна, старший преподаватель, кафедра информационных технологий, моделирования и управления; [irin2102ka@mail.ru](mailto:irin2102ka@mail.ru)

#### Information about authors:

Liliya V. Litvinenko, Master's Student, Department of Information Technology, Modeling, and Management; [litvinenko.lilia01@mail.ru](mailto:litvinenko.lilia01@mail.ru)

Lyudmila A. Korobova, Cand. Sci. (Eng.), Assoc. Prof., Assoc. Prof., Department of Information Technology, Modeling, and Management; [Lyudmila\\_korobova@mail.ru](mailto:Lyudmila_korobova@mail.ru)

Irina Sergeevna Tolstova, Senior Lecturer, Department of Information Technology, Modeling, and Management; [irin2102ka@mail.ru](mailto:irin2102ka@mail.ru)

#### Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/Received 04.11.2025.

Одобрена после рецензирования/Revised 09.12.2025.

Принята в печать/Accepted for publication 17.01.2026.