

Управление загрузкой станции технического обслуживания автомобилей с использованием теории расписаний

Е. С. Козин, А. В. Мальшаков

Тюменский индустриальный университет

Тюмень, ул. Володарского, 38, 625000, Российская Федерация

albert_nord@mail.ru



Аннотация. Эффективное управление загрузкой производственной зоны станции технического обслуживания является ключевым фактором повышения качества клиентского сервиса и операционной эффективности. Предлагаемый подход к оптимизации распределения транспортных средств по постам технического обслуживания основан на методе теории расписаний и учитывает специализацию постов, приоритеты клиентов и минимизацию времени ожидания. Разработан алгоритм динамического назначения задач, реализованный на языке Python с использованием жадной стратегии, ориентированной на раннее освобождение ресурсов. Для практического применения создано программное приложение с веб-интерфейсом и модулем визуализации расписания. Разработанный программный продукт может быть интересен предприятиям технического сервиса автомобилей и специальной техники для управления загрузкой сервисной зоны. Предложенный подход обеспечивает баланс между вычислительной эффективностью и качеством расписания, что делает его пригодным для реального применения в условиях городских СТО.

Ключевые слова: теория расписаний, управление станциями технического обслуживания, оптимизация загрузки СТО, управление загрузкой производственной зоны

Для цитирования: Козин Е. С., Мальшаков А. В. Управление загрузкой станции технического обслуживания автомобилей с использованием теории расписаний. *Архитектура, строительство, транспорт*. 2025;5(4):100–109. <https://doi.org/10.31660/2782-232X-2025-4-100-109> EDN: ANFCXO

Managing the loading of a car service station using scheduling theory

Evgeniy S. Kozin, Albert V. Malshakov

Industrial University of Tyumen

Tyumen, 38 Volodarskogo St., 625000, Russian Federation

albert_nord@mail.ru



Abstract. Effective management of the service station's production area load is a key factor in improving customer service quality and operational efficiency. The proposed approach to optimizing vehicle distribution among service stations is based on scheduling theory and takes into account station specialization, customer priorities, and waiting time minimization. A dynamic task assignment algorithm, implemented in Python, is developed using a greedy strategy focused on early resource release. A software application with a web interface and a schedule visualization module has been created for practical application. This software product may be of interest to automotive and specialized equipment service companies for managing service station loads. The proposed approach provides a balance between computational efficiency and schedule quality, making it suitable for real-world application in urban service stations.

Keywords: scheduling theory, service station managing, service station load optimization, production area load management

For citation: Kozin E. S., Malshakov A. V. Managing the loading of a car service station using scheduling theory. *Architecture, Construction, Transport.* 2025;5(4):100–109. (In Russ.) <https://doi.org/10.31660/2782-232X-2025-4-100-109>



1. Введение / Introduction

Современные станции технического обслуживания (СТО) автомобилей сталкиваются с необходимостью обработки неоднородного потока заявок при ограниченных производственных ресурсах. Отсутствие системного подхода к распределению автомобилей по постам приводит к несбалансированной загрузке, увеличению времени ожидания и снижению удовлетворенности клиентов. Традиционные методы, основанные на ручном планировании, или простых правилах FCFS (First Come, First Served (первым пришел – первым обслужен)), не учитывают такие факторы, как специализация оборудования, квалификация персонала и приоритетные категории клиентов.

Теория расписаний (Scheduling Theory) предлагает математический аппарат для решения подобных задач, однако ее применение в сфере автосервиса остается недостаточно исследованным. Особенno актуален вопрос разработки практически реализуемых алгоритмов, сочетающих адекватную точность с низкими вычислительными затратами, что критично для малых и средних предприятий. В данной работе представлен подход к управлению производственной зоной СТО, основанный на модифицированной задаче о параллельных машинах с ограничениями совместимости и приоритетами. Описана программная реализация на языке Python и указаны направления оценки эффективности pilotного внедрения программного продукта управления загрузкой в условиях реального автосервиса.

Несмотря на развитость общей теории, вопрос ее применения именно к станциям технического обслуживания остается недостаточно освещенным в научной литературе [1, 2]. Большинство публикаций фокусируется на моделях массового обслуживания (Queueing Theory), информационных системах и CRM, эвристической оценке направлений оптимизации производственных процессов, а также на оптимизации на основе программирования с ограничениями (Constraint Programming). Например, в работе [3] СТО смоделирована как система с очередями, дана оценка среднего времени ожидания и вероятности отказа. Однако такие модели не решают задачу конкретного назначения автомобиля на пост и не учитывают разнородность операций и ресурсов. В [4] описана архитектура цифровых платформ для управления СТО, включая модули записи и учета. Однако алгоритмическая часть планирования часто сводится к простым правилам очередности обработки заявок без оптимизации расписания. Отдельные исследователи применяют генетические алгоритмы или имитацию отжига для расчета расписаний на СТО. Хотя такие подходы могут давать качественные решения, они требуют значительных вычислительных ресурсов и сложны в отладке, что снижает их практическую применимость в малом бизнесе. Платформы вроде Google OR-Tools предлагают готовые модели для задач управления производственными заданиями (обычных и гибких задач календарного планирования – Job Shop и Flexible Job Shop)^{1,2}. Однако их применение в деятельности СТО требует адаптации к динамике поступления заявок, приоритетам клиентов, непрерывности смен.

¹ Google OR-Tools. Scheduling Overview. URL: <https://developers.google.com/optimization/scheduling> (дата обращения: 08.09.2025).

² Streamlit Documentation. Building Data Apps in Python. URL: <https://docs.streamlit.io> (дата обращения: 08.09.2025).

2. Материалы и методы / Materials and methods

Рассмотрим СТО со множеством постов $B = \{b_1, b_2, \dots, b_m\}$. Каждый пост b_i характеризуется множеством допустимых типов работ $C_i \subseteq \{1, 2, 3\}$, где 1 – техническое обслуживание (ТО); 2 – диагностика и текущий ремонт; 3 – ремонт с большой трудоемкостью (капитальный ремонт).

Поступает множество клиентских заявок $J = \{j_1, j_2, \dots, j_n\}$, где каждая заявка j_k задается тройкой показателей:

- a_k – время прибытия (arrival time),
- $\tau_k \in \{1, 2, 3\}$ – тип работы,
- $\rho_k \in \{0, 1\}$ – приоритет (0 – VIP, 1 – стандартный).

Время выполнения работы типа τ фиксировано и равно t_τ (например, $t_1 = 30, t_2 = 60, t_3 = 120$ минут).

Требуется назначить каждую заявку на один из постов b_i , для которого $\tau_k \in C_i$, так, чтобы:

- минимизировать суммарное (или максимальное) время ожидания:

$$W_k = s_k - a_k, \quad (1)$$

где s_k – время начала обслуживания;

- максимизировать коэффициент загрузки постов ρ :

$$\rho = \frac{\sum_{k=1}^n t_{\tau_k}}{\sum_{i=1}^m T_{\text{смена}}}. \quad (2)$$

Задача относится к классу NP-трудных, однако в условиях динамического поступления заявок (Online Scheduling) точные методы (MILP, CP) малоприменимы, поэтому предлагается эффективный эвристический алгоритм.

Алгоритм основан на стратегии самого раннего доступного совместимого бокса (Earliest Available Compatible Bay – EACB) с учетом приоритетов [5]:

1. Все заявки сортируются по невозрастанию приоритета, затем по времени прибытия:

$$(\rho_k, a_k) \rightarrow \text{asc.} \quad (3)$$

2. Для каждой заявки:

формируется подмножество совместимых постов:

$$B' = \{b_i \in B \mid \tau_k \in C_i\}; \quad (4)$$

выбирается пост:

$$b^* = \operatorname{argmin}_{b_i \in B'} \operatorname{available_time}(b_i); \quad (5)$$

заявка назначается на b^* с началом

$$s_k = \max(a_k, \operatorname{available_time}(b^*)); \quad (6)$$

время освобождения поста обновляется:

$$\operatorname{available_time}(b^*) \leftarrow s_k + t_{\tau_k}. \quad (7)$$

В качестве целевой функции может быть использовано следующее выражение:

$$\min \left(\alpha \cdot \frac{1}{n} \sum_{k=1}^n (s_k - a_k) + \beta \cdot (1 - \rho) \right), \quad (8)$$

где s_k – время начала обслуживания заявки j_k ,

ρ – коэффициент загрузки постов,

α, β – весовые коэффициенты, определяемые эмпирическим путем на основе эксперимента [6].

3. Результаты и обсуждение / Results and discussion

Алгоритм решения в обобщенном виде представлен на рис. 1.

Алгоритм работает в пакетном режиме (все заявки на расчетный период известны заранее), что типично для внутриоперационного планирования смены на СТО, и состоит из следующих основных шагов.

Шаг 1. Ввод исходных данных.

Для каждого поста задается множество типов работ, которые он может выполнять (например, пост 1 – только ТО и текущий ремонт, пост 2 – текущий и капитальный ремонт).

Для каждого автомобиля-клиента указываются:

- время прибытия на СТО,
- тип требуемых технических воздействий (1 – ТО, 2 – диагностика/текущий ремонт, 3 – капитальный ремонт),
- приоритет (VIP или стандартный).

Известно фиксированное нормативное время выполнения для каждого типа работ (например, 30, 60 и 120 минут соответственно).

Шаг 2. Сортировка заявок.

Все клиентские заявки сортируются по следующему правилу. В первую очередь реализуется сортировка по возрастанию приоритета (VIP-клиенты идут раньше). При равенстве приоритета реализуется сортировка по возрастанию времени прибытия. Это обеспечивает соблюдение принципа приоритетного обслуживания без ущерба для порядка поступления.

Шаг 3. Последовательное назначение заявок на посты.

Для каждой заявки в отсортированном списке выполняются следующие этапы.

1. Фильтрация совместимых постов. Из общего списка постов выбираются только те, которые могут выполнить данный тип работы (в соответствии с их специализацией).
2. Выбор оптимального поста. Среди совместимых постов выбирается тот, который освободится раньше всего (имеет минимальное время завершения предыдущих работ).
3. Назначение и расчет времени начала, окончания обслуживания и освобождения поста. Время начала обслуживания автомобиля определяется как максимум из времени его прибытия и времени освобождения выбранного поста. Время окончания равно сумме времени начала и нормативного времени выполнения работы. Время освобождения выбранного поста обновляется значением времени окончания обслуживания.

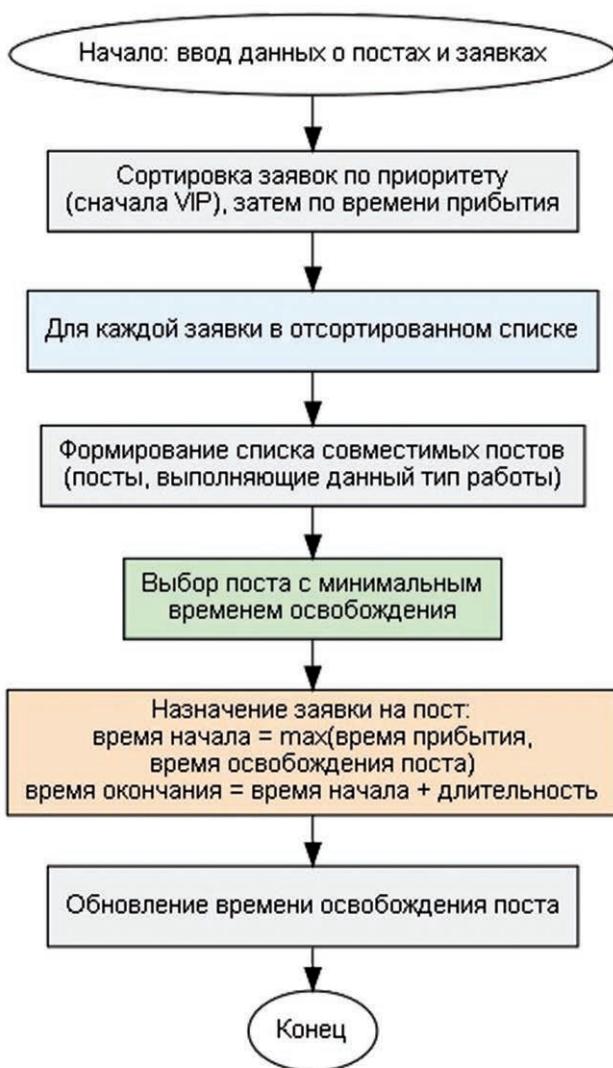


Рис. 1. Алгоритм формирования расписания распределения автомобилей по постам технического обслуживания и ремонта (схема составлена авторами)

Fig. 1. Algorithm for forming the schedule to distribute vehicles among maintenance and repair posts (scheme compiled by the authors)

Шаг 4. Формирование итогового расписания.

После обработки всех заявок формируется полное расписание, содержащее для каждой заявки:

- идентификатор клиента,
- назначенную работу и ее тип,
- выделенный пост,
- время начала и окончания обслуживания,
- время ожидания (разность между началом и моментом прибытия).

Особенностью алгоритма является использование жадной стратегии. На каждом шаге принимается локально оптимальное решение (минимизация времени начала), что обеспечивает низкую вычислительную сложность $O(n \cdot m)$, где n – число заявок, m – число постов. Алгоритм корректно обрабатывает случаи, когда ни один пост не может выполнить работу, выдавая предупреждение. Алгоритм позволяет соблюдать баланс целей, когда приоритеты управляют очередностью, а выбор самого раннего свободного поста способствует равномерной загрузке и сокращению простоев [7].

В результате формируется расписание для каждой заявки с учетом ее характеристик и параметров занятости постов. Оно представлено в виде временной диаграммы Ганта (рис. 2) [8].

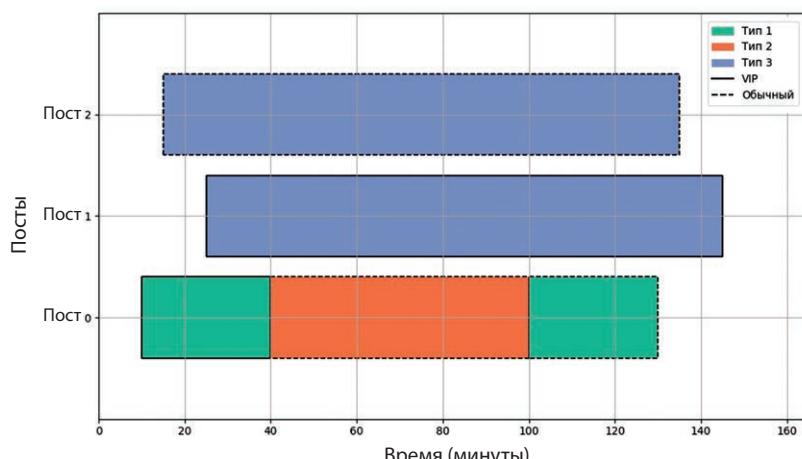


Рис. 2. Распределение автомобилей по постам (диаграмма построена авторами)
Fig. 2. Vehicle distribution among service station posts (diagram complied by the authors)

Реализация `sto_scheduler_app.py` выполнена на языке Python 3.10 с использованием следующих компонентов. Ядро расчета реализовано на основе встроенных структур данных. Интерфейс: веб-приложение на Streamlit, обеспечивающее настройку конфигурации постов, ввод и редактирование списка клиентов, визуализацию расписания в виде диаграммы Ганта. Визуализация реализована с помощью библиотеки matplotlib с экспортом в PNG/PDF для отчетов [9].

Программный код структурирован в соответствии с принципами модульности, что позволяет легко интегрировать решение в существующие информационные системы СТО (рис. 3–5).

Оценка эффективности предложенного подхода может быть проведена путем сравнительного эксперимента с использованием двух групп: контрольной группы, предполагающей формирование расписания администратором СТО вручную (традиционный метод), и экспериментальной группы, в которой расписание генерируется разработанным алгоритмом.

Важными показателями такой оценки являются:

- среднее и максимальное время ожидания клиента,
- коэффициент загрузки постов,

Пилотная система управления СТО

Распределение автомобилей по постам с учетом специализации, приоритетов и минимизации ожидания.

1. Настройка постов СТО

Количество постов

3 - +

Пост 0

ТО (легкое) Диагностика Капитальный ремонт

Пост 1

ТО (легкое) Диагностика Капитальный ремонт

Пост 2

ТО (легкое) Диагностика Капитальный ремонт

Рис. 3. Ввод количества постов и их специализации (скрин главного окна программы)

Fig. 3. Filling in the number of service station posts and their specialization (Screenshot of the main program window)

3. Результаты распределения

Клиент (прибыл)	Тип работы	Приоритет	Назначение на пост	Начало	Окончание	Ожидание (мин.)
10	ТО (легкое)	VIP		0	10	40
0	Диагностика	Обычный		1	0	60
30	ТО (легкое)	Обычный		2	30	60



Рис. 5. Вывод и визуализация результатов распределения клиентов по постам техобслуживания и ремонта (скрин окна программы)

Fig. 5. Output and visualization of customer distribution by maintenance and repair posts (program window screenshot)

- количество простоев продолжительностью более 30 минут,
- время реакции на приоритетные заявки.

Для статистической обработки полученных данных можно использовать описательные методы (среднее, стандартное отклонение) [10].

Предложенный подход демонстрирует высокую практическую применимость методов теории расписаний в сфере автосервиса. Эвристический алгоритм обеспечивает баланс между качеством расписания и вычислительной простотой, что критично для предприятий малого и среднего бизнеса.

Для объективного сравнения эффективности стандартной практики распределения заявок и разработанного алгоритма на основе теории расписаний была построена имитационная модель работы станции технического обслуживания. Модель воспроизводит работу СТО с пятью постами, различающимися по специализации, и обрабатывает поток из 120 клиентских заявок, сформированный на основе реалистичных статистических распределений (времени прибытия – пуассоновский процесс, типов работ – согласно средней структуре спроса в городском автосервисе, доли VIP-клиентов – 15 %).

В рамках модели реализованы два сценария планирования:

- стандартный подход: в нем распределение автомобилей по постам имитирует типичную ручную практику FCFS (First Come, First Served) с учетом совместимости постов, но без оптимизации по времени освобождения и учета приоритета (VIP-клиенты обслуживаются в порядке очереди);
- предлагаемый подход: распределение выполняется по разработанному алгоритму ЕАСВ (Earliest Available Compatible Bay) с сортировкой заявок по приоритету и выбором поста с минимальным временем освобождения.

Результаты моделирования представлены в таблице 1.

Таблица 1. Оценка эффективности предлагаемого подхода на имитационной модели по сравнению с ручным распределением заявок по постам

Table 1. Efficiency assessment of the suggested approach based on simulation model compared with the manual distribution of applications by service station posts

Показатель	Стандартный подход (FCFS)	Предлагаемый подход (ЕАСВ)	Абсолютное улучшение	Относительное улучшение
Среднее время ожидания клиента, мин	49.3	38.1	-11.2	-22.7 %
Максимальное время ожидания, мин	115	86	-29	-25.2 %
Среднее время ожидания VIP-клиентов, мин	43.8	29.0	-14.8	-33.8 %
Коэффициент загрузки постов, %	71.6	88.4	+16.8	+23.5 %
Общая продолжительность смены (makespan), мин	425	398	-27	-6.4 %
Количество простоев >30 мин на пост	15	6	-9	-60.0 %

Результаты имитационного эксперимента показали, что даже при одинаковом входящем потоке заявок и идентичной конфигурации постов применение алгоритма ЕАСВ, основанного на принципах теории расписаний, обеспечивает систематическое улучшение ключевых операционных показателей [11, 12].

4. Заключение / Conclusions

Разработанное приложение позволяет оперативно реагировать на изменения потока заявок и эффективно использовать производственные ресурсы. Наиболее значимый эффект достигнут в сервисной составляющей: снижение среднего времени ожидания более чем на 22 %, а для приоритетных клиентов – почти на 34 %. Одновременно достигается более рациональное использование производственных мощностей: коэффициент загрузки постов возрастает до 88,4 %, а число неэффективных простоев сокращается на 60 %.

Направлением дальнейших исследований является апробация разработанного метода в условиях работы реального предприятия, интеграция приложения с CRM-системами для автоматического импорта заявок, использование методов машинного обучения для прогнозирования времени выполнения работ и расширение модели на случай неопределенности (формирование стохастического расписания).

По результатам исследования можно сделать вывод, что эффективность предложенного подхода значительно выше стандартного. Оценивалась производительность по следующим показате-

лям: среднему и максимальному времени ожидания клиента, коэффициенту загрузки постов, количеству простоеv и времени приоритетной заявки.

Данный программный продукт может быть интересен предприятиям технического сервиса автомобилей и специальной техники для управления загрузкой сервисной зоны как госкорпораций, так и частного бизнеса, так как обеспечивает баланс между вычислительной эффективностью и качеством расписания.

 **Вклад авторов:** Козин Е. С.: разработка идеи, гипотезы исследования, планирование и проведение исследования, включая сбор данных и их анализ. Мальшаков А. В.: проведение исследования, включая сбор данных и их анализ, редактирование рукописи, создание визуализаций и других графических представлений данных.

Author contributions: Evgeniy S. Kozin: development of research ideas, hypothesis, planning and conducting research, including data collection and analysis. Albert V. Malshakov: conducting research, including data collection and analysis, editing the manuscript, and creating visualizations and other graphical representations of the data.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest. The authors declare no relevant conflict of interest.

Список литературы

1. Зуб И. В., Ежов Ю. Е., Касаткин Г. Д. Применение теории расписаний в организации работы морского порта. В сб.: *Логистика: современные тенденции развития: Материалы XXIV Международной научно-практической конференции, Санкт-Петербург, 10–11 апреля 2025 года*. Санкт-Петербург: Государственный университет морского и речного флота им. адмирала С. О. Макарова; 2025. С. 162–167.
2. Григорьева Е. Г., Курко Д. И. Сравнительный анализ некоторых алгоритмов из теории расписаний. В сб.: *Донецкие чтения – 2025: образование, наука, инновации, культура и вызовы современности: Материалы X Международной научной конференции, посвященной 60-летию создания Донецкого научного центра, Донецк, 05–07 ноября 2025 года*. Донецк: Донецкий государственный университет; 2025. С. 186–188.
3. Могилев А. А. Обзор методов решения задач теории расписаний. *Информатика, вычислительная техника и инженерное образование*. 2019;(4):19–32. <https://www.elibrary.ru/pafcss>
4. Pinedo M. L. *Scheduling: Theory, Algorithms, and Systems*. 6th edition. New York: Springer; 2022. URL: <https://link.springer.com/book/10.1007/978-3-031-05921-6>
5. Brucker P. *Scheduling Algorithms*. 5th edition. Berlin: Springer; 2007. URL: http://old.math.nsc.ru/LBRT/k5/Scheduling/BruckerSchedulingAlgorithms_Full.pdf
6. Denisov M. V., Kamaev V. A., Kizim A. V. Organization of the repair and maintenance in road sector with ontologies and multi-agent systems. *Procedia Technology*. 2013;9:819–825. <https://doi.org/10.1016/j.protcy.2013.12.091>
7. Куприянов Б. В., Рощин А. А. Решение flow shop задачи теории расписаний в рекурсивном представлении. В сб.: *XIV Всероссийское совещание по проблемам управления, Москва, 17–20 июня 2024 года*. Москва: Институт проблем управления им. В.А. Трапезникова РАН; 2024. С. 1860–1864. <https://vsru2024.ipu.ru/proceedings/1860.pdf>
8. Денисюк Д. М., Архипова Н. А. Теория расписания в задачах железнодорожного транспорта. В сб.: *Дни студенческой науки: Сборник материалов 52-й научной конференции обучающихся ПривГУПС, Самара, 01–25 апреля 2025 года*. Самара: Приволжский государственный университет путей сообщения; 2025. С. 67–69.
9. Захаров В. В., Соколов Б. В. Системно-кибернетический подход к описанию и решению задач теории расписаний. *Процессы управления и устойчивость*. 2025;12(S2-1):233–234.
10. Козин Е. С., Мальшаков А. В. Моделирование изменения интенсивности эксплуатации автомобилей в течение года. *Транспортное машиностроение*. 2024;(1)64–71. <https://doi.org/10.30987/2782-5957-2024-1-64-71>

11. Козин Е. С., Мальшаков А. В. Использования нейронных сетей для распознавания запасных частей для обслуживания автомобилей. В сб.: *Научная территория: технологии и инновации: Материалы Международной научно-практической конференции, Тюмень, 2023*. Тюмень: Тюменский индустриальный университет; 2023 С. 215–217. <https://www.elibrary.ru/oykzbq>
12. Веревкин Н. И., Мальшаков А. В. Влияние отрицательных температур на прочностные характеристики пневмоподвески. *Научное обозрение*. 2017;(6):57–61.

References

1. Zub I. V., Yezhov YU. YE., Kasatkin G. D. Application of scheduling theory in organizing the work of a seaport. In: *Logistika: sovremennoye tendentsii razvitiya: Materialy XXIV Mezhdunarodnoy nauchno-prakticheskoy konferentsii, Saint Petersburg, April 10–11, 2025*. Saint Petersburg: Admiral Makarov State University of Maritime and Inland Shipping; 2025. P. 162–167. (In Russ.)
2. Grigor'yeva YE. G., Kurko D. I. Comparative analysis of some algorithms from scheduling theory. In: *Donetskiye chteniya – 2025: obrazovaniye, nauka, innovatsii, kul'tura i vyzovy sovremennosti: Materialy X Mezhdunarodnoy nauchnoy konferentsii, posvyashchennoy 60-letiyu sozdaniya Donetskogo nauchnogo tsentra, Donetsk, November 5–7, 2025*. Donetsk: Donetsk State University; 2025. P. 186–188. (In Russ.)
3. Mogilev A. A. Review of methods for solving the scheduling theory problems. *Informatika, vychislitel'naya tekhnika i inzhenernoye obrazovaniye*. 2019;(4):19–32. (In Russ.) <https://www.elibrary.ru/pafcss>
4. Pinedo M. L. *Scheduling: Theory, Algorithms, and Systems*. 6th edition. New York: Springer; 2022. URL: <https://link.springer.com/book/10.1007/978-3-031-05921-6>
5. Brucker P. *Scheduling Algorithms*. 5th edition. Berlin: Springer; 2007. URL: http://old.math.nsc.ru/LBRT/k5/Scheduling/BruckerSchedulingAlgorithms_Full.pdf
6. Denisov M. V., Kamaev V. A., Kizim A. V. Organization of the repair and maintenance in road sector with ontologies and multi-agent systems. *Procedia Technology*. 2013;9:819–825. <https://doi.org/10.1016/j.protcy.2013.12.091>
7. Kupriyanov B. V., Roshchin A. A. Solution of the flow shop problem in scheduling theory in recursive representation. In: *XIV Vserossiyskoye soveshchaniye po problemam upravleniya, Moscow, June 17–20, 2024*. Moscow: V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences; 2024. P. 1860–1864. (In Russ.) <https://vspu2024.ipu.ru/proceedings/1860.pdf>
8. Denisyuk D. M., Arkhipova N. A. Scheduling theory in railway transport problems. In: *Dni studencheskoy nauki: Sbornik materialov 52-yy nauchnoy konferentsii obuchayushchikhsya PriVGUPS, Samara, April 1–25, 2025*. Samara: Volga State Transport University; 2025. P. 67–69. (In Russ.)
9. Zakharov V. V., Sokolov B. V. System-cybernetic approach to the description and solution of problems in the theory of schedules. *Control processes and stability*. 2025;12(S2-1):233–234. (In Russ.)
10. Kozin E., Malshakov A. Simulation of changes in car operation rate during the year. *Transport Engineering*. 2024;(1)64–71. (In Russ.) <https://doi.org/10.30987/2782-5957-2024-1-64-71>
11. Kozin Ye. S., Mal'shakov A. V. Using neural networks to recognize spare parts for car maintenance. In: *Nauchnaya territoriya: tekhnologii i innovatsii: Materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii, Tyumen, 2023*. Tyumen: Industrial University of Tyumen; 2023. P. 215–217. (In Russ.) <https://www.elibrary.ru/oykzbq>
12. Verevkin N. I., Mal'shakov A. V. Negative temperature impact on the durability properties of air suspension. *Nauchnoye obozreniye*. 2017;(6):57–61. (In Russ.)



Информация об авторах

Козин Евгений Сергеевич, канд. техн. наук, доцент, доцент кафедры сервиса автомобилей и технологических машин, Тюменский индустриальный университет, Тюмень, Российская Федерация, kozines@tuuiu.ru, <https://orcid.org/0000-0002-6774-3285>

Мальшаков Альберт Владимирович, канд. техн. наук, доцент кафедры интеллектуальных систем и технологий, Тюменский индустриальный университет, Тюмень, Российская Федерация, albert_nord@mail.ru

Information about the authors

Evgeniy S. Kozin, Cand. Sci. (Engineering), Associate Professor, Associate Professor in the Department of Car Service and Technological Machines, Industrial University of Tyumen, Tyumen, Russian Federation, kozines@tyuiu.ru, <https://orcid.org/0000-0002-6774-3285>

Albert V. Malshakov, Cand. Sci. (Engineering), Associate Professor, Associate Professor in the Department of Intelligent Systems and Technologies, Industrial University of Tyumen, Tyumen, Russian Federation, albert_nord@mail.ru

Получена 20 сентября 2025 г., одобрена 11 декабря 2025 г., принята к публикации 19 декабря 2025 г.
Received 20 September 2025, Approved 11 December 2025, Accepted for publication 19 December 2025