2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

doi 10.26089/NumMet.v27r441 УДК 519.8

Решение стохастической задачи о приемке и доставке грузов с временными ограничениями методом адаптивного поиска

Н. В. Царьков

Московский государственный университет имени М. В. Ломоносова,
факультет вычислительной математики и кибернетики,
Москва, Российская Федерация
ORCID: 0000-0002-0097-5188, e-mail: tsarkov90@yandex.com

Аннотация: В работе рассматривается задача о приемке и доставке грузов с временными ограничениями, в которой время проезда между пунктами зависит от момента выезда и является случайной величиной, а время обслуживания в пунктах также считается случайным. Для решения задачи предлагается эвристический алгоритм на основе адаптивного поиска в больших окрестностях. При проверке выполнимости маршрута в алгоритме учитываются временные окна, ограничения на вместимость, вероятности своевременного прибытия в каждый пункт и вероятности успешного выполнения всех маршрутов. Вычислительные эксперименты показывают, что предложенный подход существенно быстрее точной смешанно-целочисленной линейной модели и на сопоставимых сценариях дает решения с тем же значением целевой функции либо с малым относительным отклонением. Проводится анализ влияния гиперпараметров, стохастичности, функций скорости и требований к надежности на время счета и качество решения.

Ключевые слова: задача о приемке и доставке, временные окна, стохастическое время проезда, стохастическое время обслуживания, адаптивный поиск в больших окрестностях, эвристический алгоритм.

Для цитирования: Царьков Н.В. Решение стохастической задачи о приемке и доставке грузов с временными ограничениями методом адаптивного поиска // Вычислительные методы и программирование. 2026. **27**, № 4 666–689. doi 10.26089/NumMet.v27r441

© Н. В. Царьков

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Solution of a stochastic pickup and delivery problem with time windows by the adaptive large neighborhood search

Nikita V. Tsarkov
Lomonosov Moscow State University,
Faculty of Computational Mathematics and Cybernetics, Moscow, Russia
ORCID: 0000-0002-0097-5188, e-mail: tsarkov90@yandex.com

Abstract: The paper considers a stochastic pickup and delivery problem with time windows, where the travel time depends on the departure time and is a random variable, and the service time is also assumed to be random. A heuristic algorithm based on the adaptive large neighborhood search is proposed. Its route-feasibility test accounts for time windows, vehicle capacity, the probabilities of on-time arrival at each node, and the probabilities of successfully completing all routes. Computational experiments show that the proposed approach is substantially faster than an exact mixed-integer linear programming model and, in comparable scenarios, yields solutions with the same objective value or a small relative deviation. The impact of hyperparameters, stochasticity, speed profiles, and reliability requirements on the computation time and solution quality is analyzed.

Keywords: pickup and delivery problem, time windows, stochastic travel time, stochastic service time, adaptive large neighborhood search, heuristic algorithm.

For citation: N. V. Tsarkov, "Solution of a stochastic pickup and delivery problem with time windows by the adaptive large neighborhood search," Numerical Methods and Programming. **27** (4), 666–689 (2026). doi 10.26089/NumMet.v27r441.

1. Введение. Задача о приемке и доставке грузов с временными ограничениями (pickup and delivery problem with time windows, PDPTW) встречается в городских логистических системах, курьерской доставке, перевозках по требованию, медицинской логистике и других сервисах, где необходимо согласованно обслуживать пары пунктов приемки и доставки.

В классич4еской постановке для каждого запроса задана пара пунктов: пункт приемки груза и соответствующий ему пункт доставки. Требуется построить маршруты для транспортных средств так, чтобы каждый запрос обслуживался одной машиной, пункт приемки предшествовал пункту доставки, ограничения на вместимость машин не нарушались, а обслуживание всех пунктов выполнялось в пределах заданных временных окон. Целевая функция обычно связана с минимизацией суммарной стоимости или длины маршрутов. Такая задача является частным случаем задачи маршрутизации транспорта с временными окнами (vehicle routing problem with time windows, VRPTW) и поэтому относится к классу NP-трудных задач [1, 2].

В детерминированной версии задачи предполагается, что время проезда между пунктами заранее известно. Однако в городских условиях это предположение часто нарушается: время в пути зависит от времени суток, дорожной загрузки, погодных условий, дорожно-транспортных происшествий и других случайных факторов. При жестких временных окнах даже небольшое отклонение от ожидаемого времени проезда может привести к нарушению выполнимости (feasibility) маршрута. Поэтому при практическом применении таких моделей важно учитывать не только ожидаемую стоимость маршрута, но и вероятность его выполнения без нарушения временных ограничений.

В настоящей работе рассматривается расширение задачи о приемке и доставке грузов с временными окнами, в котором время проезда между пунктами является случайной величиной, зависящей от момента выезда, а время обслуживания в пунктах при этом также случайно. Для данной постановки предлагается эвристический алгоритм на основе адаптивного поиска в больших окрестностях [3, 4].

Цель работы заключается в разработке и экспериментальном исследовании эвристического алгоритма для стохастической задачи о приемке и доставке грузов с временными ограничениями и случайным временем проезда, зависящим от момента выезда. В отличие от точных методов, основанных на смешанно-целочисленном программировании (mixed-integer linear programming, MILP) [5], эвристический подход не

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

гарантирует доказанной оптимальности, но позволяет получать выполнимые решения за существенно меньшее время на задачах большей размерности.

Основной вклад работы состоит в следующем:

- предложен эвристический алгоритм на основе адаптивного поиска в больших окрестностях для стохастической задачи о приемке и доставке грузов с жесткими временными окнами;
- в алгоритм встроена проверка выполнимости маршрута, учитывающая математическое ожидание и дисперсию времени прибытия в пункты;
- учтены локальные и глобальные требования к надежности — вероятность своевременного прибытия в каждый пункт и вероятность успешного выполнения всего набора маршрутов;
- проведено вычислительное сравнение предложенного алгоритма с точной MILP-моделью при ограниченном времени решения.

Дальнейшая структура статьи организована следующим образом. В разделе 2 приведен обзор литературы. В разделе 3 дана постановка стохастической задачи. В разделе 4 описаны аппроксимации, используемые для проверки вероятностных ограничений. Раздел 5 посвящен алгоритму адаптивного поиска в больших окрестностях в рамках поставленной задачи. В разделе 6 представлены вычислительные эксперименты. В заключении приведены основные выводы и направления дальнейших исследований.

2. Обзор литературы. Задачи маршрутизации транспорта с временными окнами относятся к классическим задачам комбинаторной оптимизации. В одной из базовых работ по этой тематике были предложены тестовые наборы для задачи маршрутизации транспорта с временными окнами и рассмотрены алгоритмы ее решения [1]. Задача о приемке и доставке грузов с временными окнами является расширением этой задачи, так как в ней каждый запрос задается парой связанных пунктов: пунктом приемки и пунктом доставки. При этом необходимо обеспечить выполнение ограничения предшествования: пункт приемки должен быть посещен раньше пункта доставки, соответствующего запросу.

Точные методы решения задачи о приемке и доставке обычно основаны на применении линейного программирования, динамического программирования и методов декомпозиции. В работе [6] использовались комбинированные подходы: основная задача формулировалась как задача разбиения множества (set partitioning problem), а вспомогательная задача сводилась к поиску маршрутов с ограничениями по ресурсам (shortest path problem with resource constraints) [7]. Такой метод позволяет получать оптимальные решения, однако его вычислительная сложность быстро растет с увеличением числа запросов.

В русскоязычной литературе в основном рассматривались задачи, где каждому запросу соответствует ровно один пункт приемки и один пункт доставки. В работе [8] приведены квадратичные, целочисленные линейные и частично целочисленные формализации задачи о приемке и доставке с детерминированными временами проезда и обслуживания.

Из-за высокой вычислительной сложности точных методов при решении задачи о приемке и доставке широко применяются эвристические алгоритмы. В работе [9] исследовался поиск с запретами (tabu search). В работе [10] были предложены эвристический подход и тестовые примеры, которые позже стали стандартными для сравнения алгоритмов. В работе [11] была применена идея удаления связанных запросов с последующим восстановлением решения, которая стала основой для поиска в больших окрестностях (neighborhood search). Гибридный меметический алгоритм для детерминированной PDPTW был предложен в работе [12], где вычислительные эксперименты проводились на бенчмарках с большим числом клиентов.

Существенный вклад в развитие эвристических методов внесла работа [3], в которой был предложен поиск в больших окрестностях. В отличие от локального поиска, оперирующего двумя-тремя пунктами или запросами маршрута, поиск в больших окрестностях перестраивает крупные части маршрутов и благодаря нескольким операторам разрушения и восстановления лучше преодолевает локальные оптимумы. Такой подход показал высокую устойчивость на задачах большой размерности и стал одним из популярных методов для задач маршрутизации [13].

Отдельное направление исследований связано с учетом неопределенности при построении маршрутов. Обзоры стохастических задач маршрутизации приведены в работах [14–16], а динамических задач приемки и доставки — в работе [17]. В этих работах источниками неопределенности являлись спрос, времена обслуживания и проезда, временные окна или появление заявок. Поэтому термин “стохастическая

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

задача о приемке и доставке" в разных исследованиях относится к различным моделям. В рассматриваемой постановке заявки, пары пунктов, объемы грузов и временные окна заданы и известны заранее, а случайными являются времена проезда, зависящие от момента выезда, и времена обслуживания.

Эвристические методы для стохастических вариантов PDPTW встречаются в литературе, однако обычно относятся к другим источникам неопределенности или иной структуре задачи. Например, в работе [18] предложены сценарный подход, метод выборочного среднего (sample average approximation, SAA) и адаптивный поиск в больших окрестностях (adaptive large neighborhood search, ALNS) для PDPTW с регулярными транспортными линиями (scheduled lines) и стохастическим спросом. В работе [19] рассмотрена динамическая PDPTW с неопределенными временными окнами.

Еще одно направление связано с зависимостью времени проезда от момента выезда. В работе [20] рассматривались задачи маршрутизации с зависящими от времени выезда временами проезда и использовался принцип FIFO (first-in-first-out), согласно которому более поздний выезд не должен приводить к более раннему прибытию. Для задачи маршрутизации с такими временами проезда в работе [21] была предложена модель смешанно-целочисленного линейного программирования, учитывающая изменение времени проезда в течение дня.

В работе [22] рассматривалась задача о приемке и доставке с жесткими временными окнами и зависящими от времени выезда временами проезда между пунктами. В ней была предложена модель на основе разбиения множества, метода ветвей и границ и генерации столбцов. Насколько известно автору, для такой постановки эвристические алгоритмы ранее не описывались. В настоящей статье основной акцент сделан на эвристическом алгоритме, который может использоваться в качестве быстрого метода построения выполнимых решений в стохастической постановке. При этом вероятностные ограничения интегрируются непосредственно в процедуру проверки выполнимости маршрутов внутри адаптивного поиска в больших окрестностях.

3. Постановка стохастической задачи. Рассмотрим ориентированный граф $G = (N, A)$. Множество пунктов имеет вид $N = \{0, 2n+1\} \cup P \cup D$, где 0 — начальное депо, $2n+1$ — конечное депо, $P = \{1, \ldots, n\}$ — множество пунктов приемки грузов, а $D = \{n+1, \ldots, 2n\}$ — множество пунктов доставки. Каждому пункту приемки $i \in P$ соответствует пункт доставки $i + n \in D$ [23, 4].

Для каждого пункта $i \in P \cup D$ заданы так называемое временное окно $[e_i, l_i]$ (интервал обслуживания), объем груза q_i и случайное время обслуживания $S_i \geqslant 0$. Для пунктов приемки $q_i > 0$, а для соответствующих пунктов доставки $q_{i+n} = -q_i < 0$. Каждое транспортное средство имеет вместимость $Q > 0$. Множество транспортных средств обозначим через $K = \{1, \ldots, k\}$.

Для каждой дуги $(i,\ j) \in A$ задана стоимость проезда c_{ij} и расстояние $d_{ij} \geqslant 0$. Время проезда $X_{ij}(t)$ по дуге является случайной величиной, зависящей от момента выезда t из пункта i. Таким образом, рассматривается стохастическое время проезда, зависящее от времени выезда.

Решение задачи задается набором маршрутов транспортных средств. Для каждого транспортного средства $k \in K$ маршрут начинается в депо 0, заканчивается в депо $2n+1$ и содержит некоторую последовательность пунктов приемки и доставки. Введем бинарные переменные: $x_{ijk} = 1$, если машина k проходит по дуге $(i,\ j)$, в противном случае $x_{ijk} = 0$.

Также введем случайные величины A_{ik} и D_{ik}, обозначающие соответственно время прибытия машины k в пункт i и время выезда из него после обслуживания, а также переменные Q_{ik}, обозначающие загруженность машины k после выезда из пункта i.

Целевая функция представляет собой суммарную стоимость проезда [23, 4] (не умаляя общности, считаем, что машинами можно пользоваться бесплатно) и определяется выражением

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}\ x_{ijk}.$$

При этом должны выполняться следующие ограничения:

1. Каждый пункт приемки и доставки должен быть посещен ровно один раз, т.е.

$$\sum_{k \in K} \sum_{j:(i,j) \in A} x_{ijk} = 1, \quad i \in P \cup D.$$

2. Для каждого запроса пункты приемки и доставки должны обслуживаться одним и тем же транс-

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

портным средством, т.е.

$$\sum_{j:(i,j)\in A} x_{ijk} = \sum_{j:(i+n,j)\in A} x_{i+n,j,k}, \quad i \in P,\ k \in K.$$

3. Маршрут должен начинаться и заканчиваться в депо, т.е.

$$\sum_{j\in N\setminus\{0\}} x_{0jk} \leqslant 1, \qquad \sum_{i\in N\setminus\{2n+1\}} x_{i,2n+1,k} \leqslant 1, \quad k \in K,$$
$$\sum_{j\in N\setminus\{0\}} x_{0jk} = \sum_{i\in N\setminus\{2n+1\}} x_{i,2n+1,k}, \quad k \in K.$$

4. Если машина приезжает в какой-то пункт, то она должна и выезжать из него, т.е.

$$\sum_{i\in N\setminus\{h\}} x_{ihk} = \sum_{j\in N\setminus\{h\}} x_{hjk}, \quad h \in P \cup D,\ k \in K.$$

5. Далее должно соблюдаться ограничение, согласно которому пункт приемки посещается раньше соответствующего ему пункта доставки, т.е.

$$D_{ik} \leqslant A_{i+n,k}, \quad i \in P,\ k \in K.$$

6. На каждом маршруте должно соблюдаться ограничение на вместимость машины, т.е.

$$0 \leqslant Q_{ik} \leqslant Q, \quad i \in P \cup D,\ k \in K,$$

где Q_{ik} — загрузка транспортного средства k после обслуживания в пункте i.

Также необходимо учитывать стохастическую динамику времени. Если машина k после посещения пункта i едет к пункту j, то время прибытия в j определяется случайным временем выезда из i и случайным временем проезда:

$$A_{jk} = D_{ik} + X_{ij}\left(D_{ik}\right),\ i \in N.$$

Время выезда из пункта j определяется временем начала обслуживания и случайным временем обслуживания:

$$D_{jk} = \max\left\{A_{jk}, e_j\right\} + S_j,\ j \in N.$$

Поскольку в настоящей постановке имеются временные ограничения, то требуется контролировать вероятность своевременного прибытия в каждый пункт. Под своевременным прибытием понимается событие $A_{jk} \leqslant l_j$, то есть прибытие в пункт не позднее конца его интервала обслуживания. Если в каком-либо маршруте машина прибывает хотя бы в один пункт не вовремя, то все решение считается невыполнимым (infeasible). Прибытие раньше начала обслуживания не является нарушением: допускается, что машина может подождать до начала обслуживания. Для каждого пункта $j \in P \cup D$ задается (локальный) уровень надежности пункта (node service level) $\theta_j \in [0,1]$ — минимально допустимая вероятность прибытия в пункт вовремя, которая используется в ограничении вида:

$$\mathbb{P}\left(A_{jk} \leqslant l_j\right) \geqslant \theta_j,\ j \in N,\ k \in K.$$

Кроме локальных ограничений на вероятность своевременного прибытия, задается (глобальный) уровень надежности решения (global service level) Θ — минимальная вероятность успешно пройти все маршруты решения, т.е. прибыть вовремя во все пункты в задаче:

$$\mathbb{P}\left(A_{jk} \leqslant l_j \text{ для всех посещенных пунктов } j \text{ и всех } k \in K\right) \geqslant \Theta.$$

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

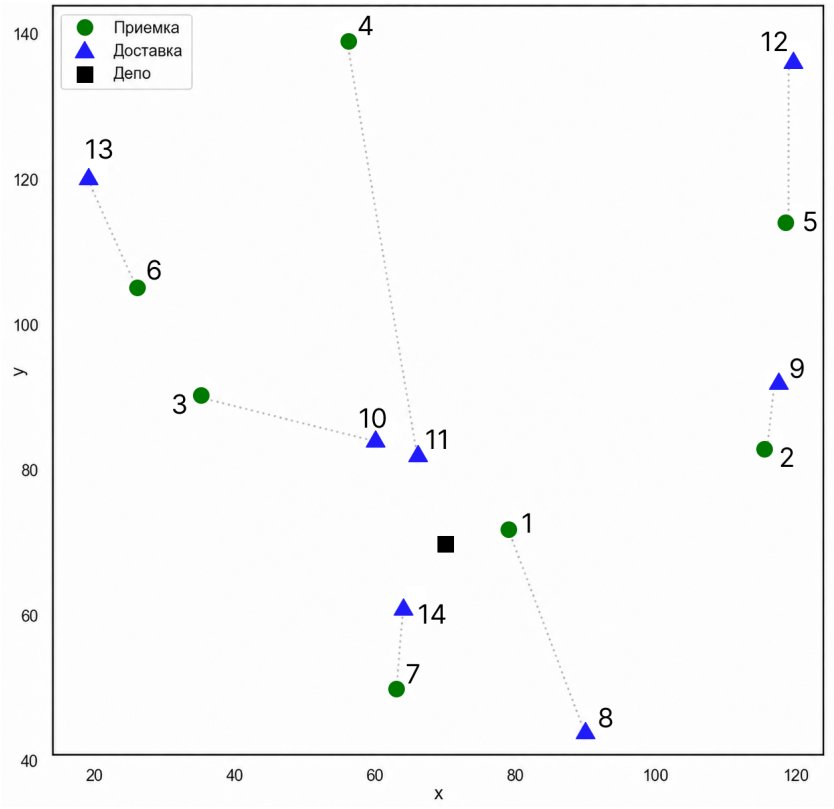


Рис. 1. Расположение пунктов на карте (линиями показана связь между пунктами)

Fig. 1. Locations of the nodes on the map (lines show the connections between nodes)

Таким образом, задача состоит в построении набора маршрутов с минимальной суммарной стоимостью проезда, удовлетворяющих ограничениям посещения пунктов, парности запросов, предшествования, ограничениям на вместимость и вероятностным ограничениям на своевременное посещение пунктов.

На рис. 1 приведено графическое представление задачи PDPTW: на карте изображены пункты, линиями показаны связи между пунктами (для одного запроса меньшее число соответствует пункту приемки, большее число — пункту доставки, квадрат — депо). На рис. 2 a на временном интервале показаны временные окна для каждого пункта. На рис. 2 b для каждого запроса показан размер груза. На рис. 3 продемонстрировано решение задачи, соответствующее данным, представленным на рис. 1, 2. Здесь последовательности стрелок показывают маршруты.

4. Аппроксимация времен прибытия и проверка вероятностных ограничений. Проверка вероятностных ограничений в стохастической задаче является вычислительно сложной процедурой, поскольку времена прибытия в последующие пункты маршрута зависят от случайных времен проезда и обслуживания во всех предыдущих пунктах. Поэтому в алгоритме используется рекуррентный расчет математических ожиданий и дисперсий времен прибытия и выезда вдоль маршрута.

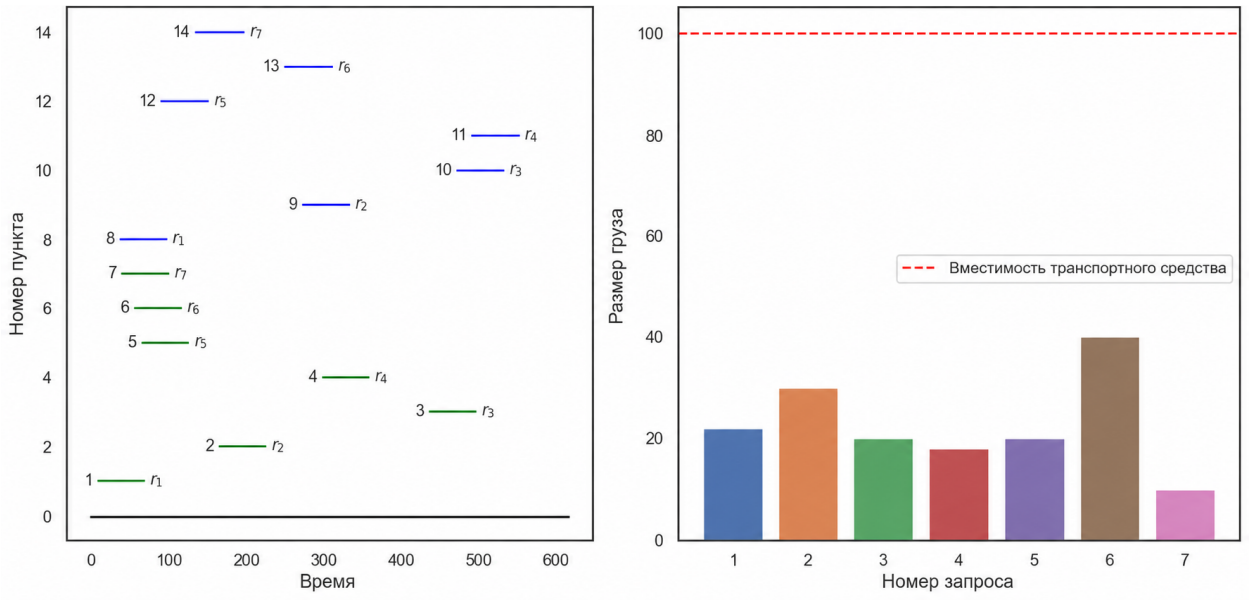
4.1. Время проезда, зависящее от момента выезда. Пусть $v(t)$ — ступенчатая функция скорости, описывающая изменение средней скорости движения в течение планового горизонта. Ожидаемое время проезда $\tau_{ij}(t)$ по дуге $(i,\ j)$ при выезде в момент t определяется из интегрального уравнения:

$$d_{ij} = \int\limits_{t}^{t+\tau_{ij}(t)} v(u)\ du.$$

Такой способ расчета соответствует принципу FIFO: условно, если две машины едут между двумя пунктами, то та машина, которая выехала позже, должна прибыть не раньше другой машины (см. рис. 4, 5). В вычислительных экспериментах функция скорости задается как кусочно-постоянная.

Ожидаемое время проезда и дисперсия времени проезда задаются следующим образом:

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441



a) b)

Рис. 2. Характеристики задачи, представленной на рис. 1: a) временные окна пунктов (зеленый цвет соответствует приемке, синий — доставке); b) размеры грузов и вместимость транспортного средства

Fig. 2. Characteristics of the problem presented in Figure 1: a) node time windows (green lines correspond to pickup, blue — to delivery); b) load sizes and vehicle capacity

$$\mathbb{E}\left[X_{ij}(t)\right] = \tau_{ij}(t), \qquad \mathrm{Var}\left[X_{ij}\right] = \sigma_{ij}^2.$$

В экспериментах стохастичность времени проезда задается коэффициентом вариации cv [22]:

$$\sigma_{ij} = cv \cdot d_{ij}.$$

4.2. Стохастическое время обслуживания. Время обслуживания в пункте i задается случайной величиной S_i. В вычислительных экспериментах используется нормальное распределение:

$$S_i \sim \mathcal{N}\left(\mu_i^S, \left(\sigma_i^S\right)^2\right),$$

где $\mu_i^S = \mathbb{E}\left[S_i\right]$, а $\left(\sigma_i^S\right)^2 = \mathrm{Var}\left[S_i\right]$.

4.3. Рекуррентный расчет математического ожидания и дисперсии времени прибытия. Рассмотрим маршрут $r = (0, v_1, \ldots, v_m, 2n+1)$, где $v_1, \ldots, v_m$ — какие-то пункты в маршруте. Для пункта j маршрута r обозначим случайное время прибытия через A_j^r, а случайное время выезда после обслуживания — через D_j^r. Если предыдущий пункт маршрута — это пункт i, то справедливо равенство

$$A_j^r = D_i^r + X_{ij}\left(D_i^r\right).$$

Точное вычисление параметров распределения для величины A_j^r затруднительно, поэтому используется аппроксимация через первые два момента распределения. Вводятся следующие допущения:

1. Времена проезда между разными пунктами рассматриваются как независимые величины.
2. Времена обслуживания не зависят от времен проезда.
3. Зависимость от времени выезда учитывается в математическом ожидании $\mathbb{E}\left[X_{ij}(t)\right]$.

Математическое ожидание времени прибытия рассчитывается по формуле

$$\mathbb{E}\left[A_j^r\right] = \mathbb{E}\left[D_i^r\right] + \mathbb{E}\left[X_{ij}\left(\mathbb{E}\left[D_i^r\right]\right)\right].$$

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

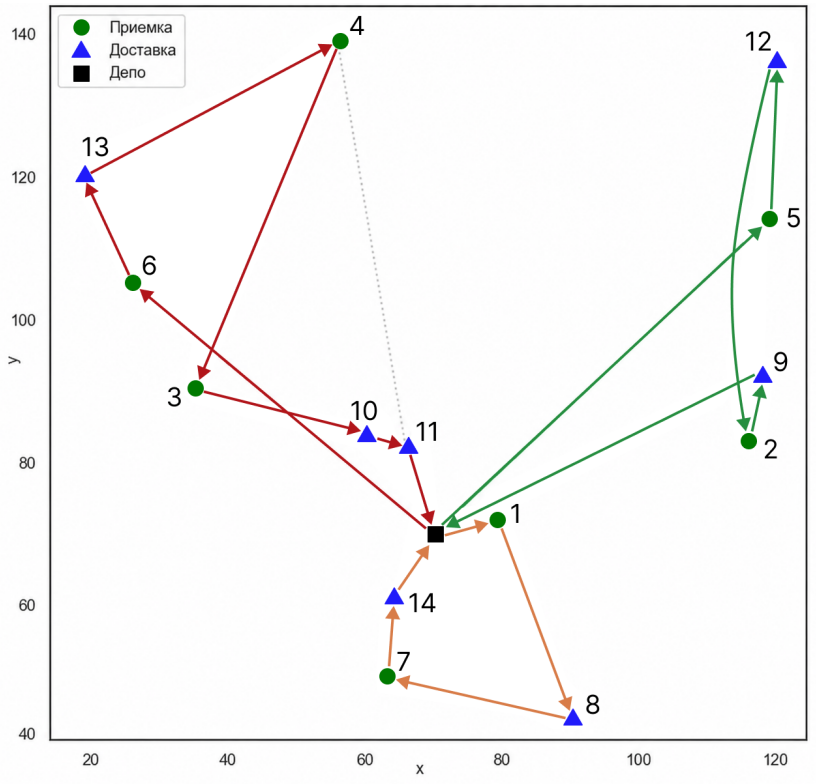


Рис. 3. Решение примера задачи, представленной на рис. 1, 2 (разные цвета линий обозначают маршруты разных транспортных средств)

Fig. 3. Solution of the example problem presented in Figures 1, 2 (different line colors indicate the routes of different vehicles)

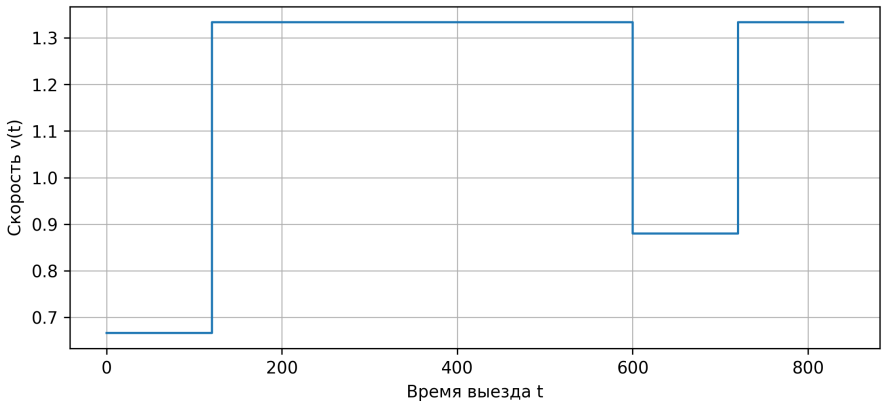


Рис. 4. Пример кусочно-постоянной функции скорости

Fig. 4. An example of a piecewise-constant speed function

В дисперсии времени прибытия учитывается зависимость времени проезда от момента выезда с помощью применения приближения первого порядка [24]:

$$\operatorname{Var}\left[A_j^r\right] \approx \left(1 + \frac{d\mathbb{E}\left[X_{ij}(t)\right]}{dt}\Big|_{t=\mathbb{E}[D_i^r]}\right)^2 \operatorname{Var}\left[D_i^r\right] + \operatorname{Var}\left[X_{ij}\right].$$

Если машина прибывает в пункт раньше начала временного окна, то она ожидает начала обслуживания. Чтобы учесть этот аспект, время начала обслуживания задается как

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

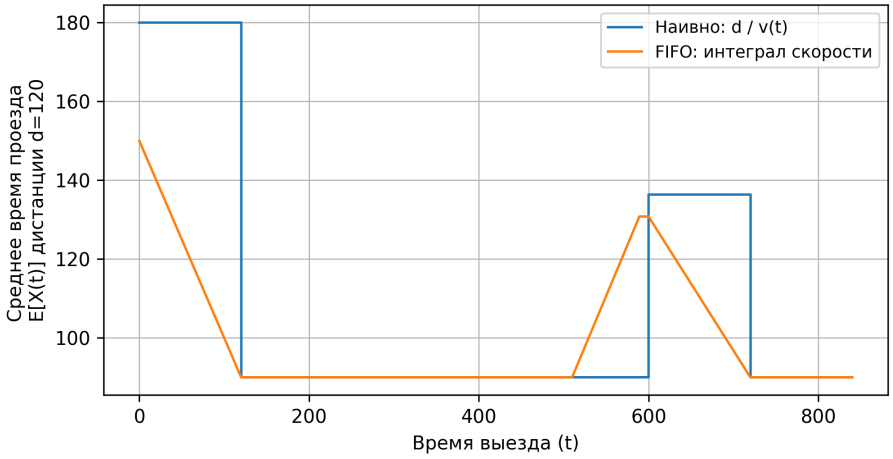


Рис. 5. Сравнение результатов наивного расчета времени проезда и расчета, удовлетворяющего принципу FIFO

Fig. 5. Comparison of the results of the naive travel-time calculation and the FIFO-consistent calculation

$$g_j(t) = \max\{t, e_j\}.$$

Тогда математическое ожидание времени выезда из пункта j равно:

$$\mathbb{E}\left[D_j^r\right] = g_j\left(\mathbb{E}\left[A_j^r\right]\right) + \mathbb{E}\left[S_j\right].$$

Для учета влияния ожидания в пункте на дисперсию времени выезда используется аппроксимация, основанная на производной функции начала обслуживания. Внутри временного окна производная равна единице, а при прибытии раньше начала окна она равна нулю, поскольку раннее прибытие приводит к ожиданию:

$$g_j'(t) = \begin{cases} 0, & t \leqslant e_j, \\ 1, & t > e_j. \end{cases}$$

Тогда дисперсия времени выезда аппроксимируется выражением:

$$\operatorname{Var}\left[D_j^r\right] \approx \left(\overline{g}_j'\right)^2 \operatorname{Var}\left[A_j^r\right] + \operatorname{Var}\left[S_j\right],$$

где $\overline{g}_j'$ — среднее значение производной функции g_j в окрестности $\mathbb{E}\left[A_j^r\right]$. Используемые аппроксимации моментов основаны на подходе из работы [22], но здесь они применяются не внутри точного алгоритма, а в процедуре проверки выполнимости маршрутов при эвристическом поиске решения.

4.4. Проверка локальной и глобальной надежности. После расчета математического ожидания и дисперсии времени прибытия оценивается вероятность своевременного прибытия в пункт. При нормальной аппроксимации она равна

$$P_j^r = \mathbb{P}\left(A_j^r \leqslant l_j\right) \approx \Phi\left(\frac{l_j - \mathbb{E}[A_j^r]}{\sqrt{\operatorname{Var}[A_j^r]}}\right),$$

где $\Phi(\bullet)$ — функция стандартного нормального распределения. Для выполнимости маршрута необходимо выполнение локального вероятностного ограничения

$$P_j^r \geqslant \theta_j \tag{1}$$

для каждого пункта j маршрута.

Следует отметить, что события своевременного прибытия в различные пункты одного маршрута не являются независимыми. Опоздание в одном пункте может влиять на вероятность своевременного обслу-

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

живания последующих пунктов. Точное вычисление совместной вероятности выполнения всех временных ограничений сопряжено с необходимостью работы с многомерным распределением времен прибытия и является ресурсозатратным процессом. Поэтому в настоящей работе используется аппроксимация, предполагающая независимость событий:

$$P_r \approx \prod_{j \in r} P_j^r.$$

Для всего решения $\mathcal{R} = \{r_1, \ldots, r_K\}$ вероятность его успешного выполнения оценивается как

$$P_{\text{global}} \approx \prod_{r \in \mathcal{R}} \prod_{j \in r} P_j^r.$$

Соответственно, глобальное вероятностное ограничение имеет вид:

$$P_{\text{global}} \geqslant \Theta. \tag{2}$$

Таким образом, проверка выполнимости маршрута в стохастической постановке сводится к последовательному пересчету математических ожиданий, дисперсий и вероятностей своевременного прибытия вдоль маршрута.

В завершение формального описания задачи перечислим полный набор ее входных параметров. Их можно разделить на детерминированные и стохастические. К детерминированным параметрам относятся расположение депо, множества пунктов приемки P и доставки D, расстояния d_{ij} и стоимости проезда c_{ij}, парк транспортных средств K, их вместимость Q, временные окна $[e_i, l_i]$, объемы грузов q_i и функция скорости $v(t)$, определяющая математическое ожидание времени проезда $\mathbb{E}[X_{ij}(t)]$. Стохастические параметры задают распределения времени проезда $X_{ij}(t)$ и времени обслуживания S_i, а также требования к надежности: локальные уровни θ_j и глобальный уровень Θ.

5. Алгоритм адаптивного поиска в больших окрестностях.

5.1. Общая схема. Адаптивный поиск в больших окрестностях (adaptive large neighborhood search, ALNS) является итеративным алгоритмом, т.е. строит новое решение из текущего путем последовательного удаления части запросов и их повторной вставки. Более подробное описание алгоритма содержится в статьях [3, 4]. В них также приведено описание операторов разрушения и восстановления, а также показано, какие операторы наиболее эффективны. В настоящей работе основное внимание уделено модификации ALNS для стохастической задачи.

5.2. Оператор восстановления и проверка выполнимости маршрута для стохастической задачи. Напомним, что оператор восстановления получает набор маршрутов, из которых удалена часть запросов, и список этих удаленных запросов. Для каждого запроса необходимо выбрать позиции вставки для пункта приемки и пункта доставки так, чтобы все ограничения оставались выполненными.

Для каждого потенциального варианта вставки выполняется проверка:

a) ограничения предшествования пункта приемки пункту доставки;

b) ограничения на вместимость транспортного средства;

c) в отличие от детерминированного варианта задачи, где используется прямая проверка своевременного прибытия, в стохастической задаче используется проверка локального вероятностного ограничения (1), а именно $P_j^r \geqslant \theta_j$;

d) выполнения глобального вероятностного ограничения (2), т.е. $P_{\text{global}} \geqslant \Theta$.

Проверка выполняется последовательно вдоль маршрута. Если на некотором пункте нарушается хотя бы одно ограничение, дальнейшая проверка прекращается, а соответствующая вставка считается невыполнимой. Такая “ленивая” проверка особенно важна для скорости алгоритма, поскольку оператор восстановления генерирует большое число потенциальных вариантов вставки, которые необходимо проверять на выполнимость.

5.3. Вставки за один проход. Для ускорения поиска на больших размерностях в операторах восстановления используется авторская модификация [4], в которой список возможных вариантов вставок

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

запросов формируется один раз и далее просматривается в порядке увеличения прироста целевой функции. При нахождении выполнимого варианта вставки полный пересчет всех вариантов не осуществляется. Такая модификация снижает вычислительные затраты во время процедуры восстановления, но может ухудшить локальную оптимальность отдельной вставки.

5.4. Оценка вычислительной сложности алгоритма. Пусть n — число запросов, $q = \rho n$ — число удаляемых запросов, N_{iter} — число итераций цикла ALNS. Пусть текущее решение состоит из множества маршрутов $\mathcal{R}$. Для маршрута $r \in \mathcal{R}$ обозначим через $|r|$ число посещенных пунктов в этом маршруте. Среднюю длину маршрута m можно определить следующим образом:

$$m = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} |r|. \tag{3}$$

Эта величина не является заранее заданным параметром для алгоритма и меняется в ходе его работы, однако она удобна для оценки вычислительной сложности проверки вставок. При этом для худшей оценки по числу запросов n удобнее использовать не только среднюю длину маршрута, но и сумму по всем маршрутам.

Проверка выполнимости одного маршрута требует пересчета характеристик вдоль маршрута: при последовательном проходе от начала до конца маршрута пересчитываются загрузка машины, времена прибытия и выезда из пунктов, а в стохастической постановке — также математические ожидания, дисперсии и вероятности своевременного прибытия в последующие пункты маршрута. Если длина маршрута равна $|r|$, то такой пересчет имеет сложность $O(|r|)$.

Для одного запроса число возможных позиций вставки пунктов приемки и доставки имеет порядок $O\left(|r|^2\right)$. Если для каждой пары позиций выполняется полный пересчет маршрута, сложность проверки всех вариантов вставки одного запроса в маршрут r составляет $O\left(|r|^2\right) \cdot O(|r|) = O\left(|r|^3\right)$.

Если вставка одного запроса проверяется по всем маршрутам текущего решения, то более точная оценка сложности вставки представляется в виде $O\left(\sum\limits_{r \in \mathcal{R}} |r|^3\right)$.

Поскольку каждый запрос содержит один пункт приемки и один пункт доставки, суммарное число посещенных пунктов во всех маршрутах не превосходит $2n$:

$$\sum_{r \in \mathcal{R}} |r| \leqslant 2n = O(n).$$

Следовательно, в худшем случае сложность вставки одного запроса составляет $O\left(n^3\right)$.

Таким образом, число машин само по себе не ухудшает худшую асимптотическую оценку по n: наибольшая сложность достигается, когда почти все пункты находятся в одном длинном маршруте. При распределении пунктов между несколькими машинами средняя длина маршрутов уменьшается, и фактическое время проверки вставок обычно оказывается ниже.

Далее возможны разные варианты процедуры восстановления. В варианте с фиксированным порядком вставки удаленные запросы обрабатываются в заранее заданной последовательности. Для каждого запроса перебираются все возможные позиции пунктов приемки и доставки, выбирается лучшая выполнимая вставка, после чего алгоритм переходит к следующему запросу. При таком подходе сложность восстановления q запросов составляет $O\left(q \sum\limits_{r \in \mathcal{R}} |r|^3\right)$.

В худшем случае, когда все запросы находятся в одном маршруте, получаем оценку сложности в виде $O\left(q \cdot n^3\right)$.

Так как $q = \rho n$, а ρ фиксировано, находим худшую оценку $O\left(n^4\right)$ одной итерации ALNS.

С учетом числа итераций сложность цикла ALNS в худшем случае оценивается как $O\left(N_{\text{iter}} \cdot n^4\right)$.

Если использовать среднюю длину маршрута m (формула (3)), то при фиксированном числе используемых маршрутов эту же оценку можно записывать в более компактном виде $O\left(N_{\text{iter}} \cdot q \cdot m^3\right)$.

На практике маршруты распределены между несколькими машинами, а проверка невыполнимых вставок часто завершается досрочно. Поэтому наблюдаемое время работы оказывается существенно ниже оценки худшего случая.

Более дорогой вариант восстановления возникает в случае полной “жадной” вставки. В этом варианте на каждом шаге алгоритм заново рассчитывает лучшие позиции вставки для всех еще не вставленных

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

запросов, после чего выбирается глобально лучший вариант. После каждой вставки структура маршрутов меняется, поэтому ранее рассчитанные стоимости и выполнимости вставок могут стать неактуальными. В результате повторный пересчет по всем оставшимся запросам дает дополнительный множитель q, и сложность процедуры восстановления может достигать $O\left(q^2 \sum\limits_{r\in\mathcal{R}} |r|^3\right)$.

В худшем случае, когда запросы находятся в одном маршруте, получаем оценку сложности $O\left(q^2 \cdot n^3\right)$.

При $q = O(n)$ в худшем случае имеем оценку сложности $O\left(n^5\right)$ на одну итерацию ALNS.

Использование вставок за один проход уменьшает вычислительную сложность процедуры восстановления за счет отказа от полного пересчета всех вариантов вставки после каждого добавленного запроса. Список вариантов вставки формируется один раз, затем они просматриваются в порядке увеличения прироста целевой функции, и выбирается первый найденный выполнимый вариант вставки.

В результате для "жадного" оператора восстановления худшая оценка снижается с $O\left(q^2 \cdot n^3\right)$ до $O\left(q \cdot n^3\right)$, то есть устраняется дополнительный множитель q. В терминах средней длины маршрута это соответствует переходу от $O\left(q^2 \cdot m^3\right)$ к $O\left(q \cdot m^3\right)$, если число используемых маршрутов рассматривается как ограниченное. На практике это ускоряет процедуру восстановления, однако может ухудшать качество локального выбора, поскольку стоимости вставок не пересчитываются после каждого изменения маршрутов.

Сложность процедуры разрушения зависит от используемого оператора. Если множество удаляемых запросов уже выбрано, то их удаление из маршрутов имеет сложность $O(q)$ при хранении позиций пунктов приемки и доставки. Однако выбор самих запросов может требовать дополнительного прохода по текущему решению.

Для случайного удаления [4] сложность составляет от $O(q)$ до $O(n)$. Для операторов, основанных на оценке вклада запроса в стоимость маршрута (целевую функцию) [4], требуется рассчитать показатель для каждого запроса, что дает сложность $O(n)$, а при последующей сортировке — $O\left(n \log n\right)$. Если вместо полной сортировки используется выбор q наихудших элементов, оценка может снижаться до $O\left(n + q \log n\right)$.

Для операторов, учитывающих "похожесть" запросов, сложность выше. Например, в эвристике удаления Шэу [4] на каждом шаге может потребоваться сравнение выбранного запроса со всеми оставшимися, что дает оценку $O(q \cdot n)$. При предварительном построении полной матрицы похожести сложность может достигать $O\left(n^2\right)$.

Таким образом, процедура разрушения решения обычно не является вычислительным "узким местом" ALNS: ее сложность ниже сложности восстановления, где требуется многократная проверка выполнимости пробных вставок. Сводные оценки вычислительной сложности различных этапов ALNS приведены в табл. 1.

Из приведенных оценок следует, что учет стохастичности сам по себе не меняет порядок числа пробных вставок и не добавляет отдельного комбинаторного множителя к схеме ALNS. Основное усложнение возникает внутри проверки одного маршрута: вместе с проверкой логики последовательно пересчитываются математические ожидания, дисперсии и вероятности своевременного прибытия. При используемой аппроксимации этот пересчет остается линейным по длине маршрута, поэтому асимптотический порядок проверки маршрута сохраняется равным $O(m)$, но возрастает постоянный множитель.

Существенное снижение сложности этапа восстановления достигается за счет вставок за один проход. Если после каждой вставки заново пересчитывать лучшие позиции для всех еще не вставленных запросов, то восстановление q запросов имеет оценку $O(q^2 \cdot m^3)$. В используемой схеме для каждого удаленного запроса выполняется один перебор допустимых позиций, поэтому оценка этапа восстановления снижается до $O(q \cdot m^3)$. Таким образом, новизна приведенного анализа состоит в совместном учете вероятностной проверки выполнимости и схемы вставок за один проход, позволяющей уменьшить вычислительную сложность восстановления решения в стохастической постановке.

6. Вычислительные эксперименты.

6.1. Постановка экспериментов. Вычислительные эксперименты проводились для семейства задач PDPTW размерностью 25 (группа pdp_25), 50 (группа pdp_50) и 100 (группа pdp_100) клиентских пунктов. Под клиентскими пунктами здесь понимаются пункты приемки и доставки без учета депо. Наборы pdp_25 и pdp_50 не являются бенчмарк-примерами Li–Lim: они были получены путем случайного выбора пунктов из задач со 100 клиентскими пунктами с сохранением соответствующих пар приемки и

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Таблица 1. Вычислительная сложность этапов алгоритма ALNS

Table 1. Computational complexity of the ALNS stages

Этап Stage	Оценка сложности при средней длине маршрута m Complexity for mean route length m
Случайное удаление запросов Random request removal	от $O(q)$ до $O(n)$ from $O(q)$ to $O(n)$
Удаление запросов на основе стоимости, ожидания или задержки Cost-, waiting- or delay-based removal	от $O(n)$ до $O(n \log n)$ from $O(n)$ to $O(n \log n)$
Проверка одного маршрута Single-route feasibility check	$O(m)$
Перебор позиций для одного запроса Positions for one request	$O(m^2)$
Восстановление одного запроса с полным пересчетом маршрута One-request repair with full route recalculation	$O(m^3)$
Восстановление q запросов при фиксированном порядке вставки Repair of q requests in fixed insertion order	$O(q \cdot m^3)$
Вставка с пересчетом всех оставшихся запросов Insertion with recalculation of remaining requests	$O(q^2 \cdot m^3)$
Вставка за один проход Single-pass insertion	$O(q \cdot m^3)$
ALNS при вставках за один проход ALNS with single-pass insertion	$O(N_{\text{iter}} \cdot q \cdot m^3)$

доставки. Такой способ использовался из-за высокой вычислительной сложности MILP-модели: уже для 50 пунктов решатель в ряде случаев достигал временного лимита, а для 100 пунктов сравнение с MILP не проводилось. Таким образом, для сравнения с MILP использовался авторский набор тестов, построенный по мотивам Li–Lim [10], а на 100 пунктах — бенчмарк Li–Lim.

Каждый пример задачи задает расположение депо и клиентских пунктов, пары приемки и доставки, размеры грузов, временные окна, длительности обслуживания и характеристики парка транспортных средств. Рассматривались семейства LC с группами близко расположенных пунктов, LR с распределением пунктов по всей карте и LRC со смешанной структурой. Графическое представление такого примера, его временные окна, размеры грузов и набор маршрутов показаны на рис. 1–3. При варьировании стохастических параметров, функции скорости и изменении требований к надежности эти исходные входные данные примера сохранялись.

Для ALNS рассматривалась сетка гиперпараметров

$$\rho \in \{0.05,\ 0.15,\ 0.25\}, \qquad N_{\text{iter}} \in \{100,\ 500,\ 1000\}.$$

В экспериментах варьировались следующие параметры:

a) коэффициент вариации для времени проезда $cv \in \{0,\ 0.15,\ 0.30\}$;

b) дисперсия времени обслуживания $\mathrm{Var}[S] \in \{0,\ 1,\ 2\}$;

c) функция скорости: обозначение 100 соответствует постоянной скорости; 90_110 означает, что скорость колеблется около базовой в пределах примерно от 90% до 110%; 70_130 — более сильное изменение скорости, от 70% до 130%; 50_100 соответствует падению скорости до 50% в часы пик и восстановлению до 100% (рис. 6);

d) локальный и глобальный уровни надежности: 70%, 80% и 90%.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Таблица 2. Сравнение качества решений ALNS и COPT-MILP

Table 2. Comparison of ALNS and COPT-MILP solution quality

Размерность Problem size	Результат Result	Количество примеров Number of cases	Доля Share
25	ALNS лучше MILP ALNS better than MILP	1532	52.97%
25	ALNS = MILP	1350	46.68%
25	MILP лучше ALNS MILP better than ALNS	10	0.35%
50	ALNS лучше MILP ALNS better than MILP	1414	95.86%
50	ALNS = MILP	58	3.93%
50	MILP лучше ALNS MILP better than ALNS	3	0.20%

Под входным сценарием понимается комбинация исходного примера задачи, функции скорости, пары параметров стохастичности и пары уровней надежности. В группы pdp_25 и pdp_50 входило по 60 исходных примеров, в группу pdp_100 — 56. Для каждого примера перебирались четыре функции скорости (100, 90_110, 70_130, 50_100), пять пар $(cv, \mathrm{Var}[S])$ и три пары (θ, Θ):

$$\mathcal{V} = \{(0,0),\ (0.15,1),\ (0.15,2),\ (0.30,1),\ (0.30,2)\},$$
$$\mathcal{L} = \{(0.7,0.7),\ (0.8,0.8),\ (0.9,0.9)\}, \qquad \theta_j = \theta \text{ для всех клиентских пунктов } j.$$

Таким образом, параметры cv и $\mathrm{Var}[S]$ варьировались совместно в пределах множества $\mathcal{V}$, а локальный и глобальный уровни надежности — в пределах множества $\mathcal{L}$. Их независимое декартово произведение не использовалось. Полная сетка входных сценариев представляла собой декартово произведение множества исходных примеров, четырех функций скорости, $\mathcal{V}$ и $\mathcal{L}$. На каждый исходный пример приходилось $4 \cdot 5 \cdot 3 = 60$ сценариев. Поэтому для каждой из групп pdp_25 и pdp_50 рассматривалось $60 \cdot 4 \cdot 5 \cdot 3 = 3600$ сценариев, а для pdp_100 — $56 \cdot 4 \cdot 5 \cdot 3 = 3360$. Всего — 10 560 входных сценариев.

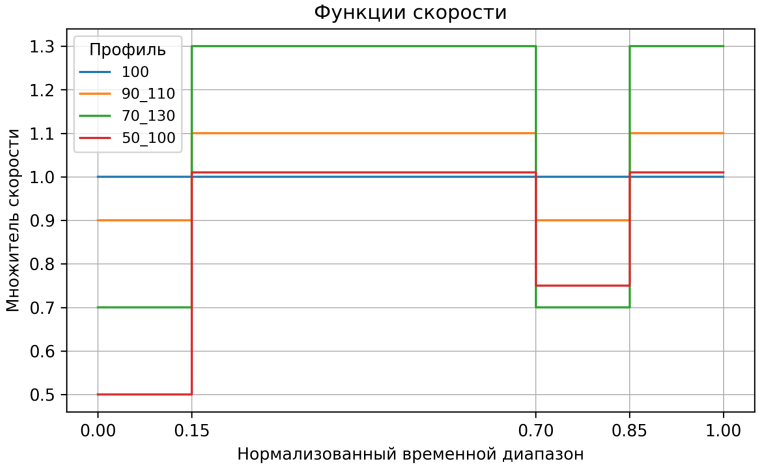


Рис. 6. Функции скорости, используемые в экспериментах

Fig. 6. Speed profiles used in the experiments

Пара гиперпараметров ALNS $(\rho, N_{\mathrm{iter}})$ задает конфигурацию алгоритма для данного входного сценария. Запуски с разными конфигурациями на одинаковых входных данных следует отличать от разных входных сценариев. В разделе 6.4 результаты анализируются отдельно для каждой конфигурации.

Сценарий считался сопоставимым для сравнения ALNS и COPT-MILP, если оба метода вернули выполнимое решение для одной и той же комбинации входных параметров. Запуски, в которых хотя бы один из методов не возвращал выполнимого решения к моменту остановки, исключались из парного сравнения качества. Из 3600 входных сценариев группы pdp_25 в табл. 2 и 3 вошли 2892 сопоставимых сценария, то есть за пределами парного сравнения остались 708 (19.67%). Для группы pdp_50 из 3600 сценариев в сравнение вошли 1475; за его пределами остались 2125 (59.03%). Для группы pdp_100 сравнение с MILP не проводилось. Проверка методом Монте-Карло выполнялась отдельно для найденных ALNS-решений. Ее результаты обсуждаются в разделе 6.7.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Таблица 3. Отклонение и медианное время решения по сопоставимым сценариям

Table 3. Gap and median solution time for comparable scenarios

Размерность Problem size	Количество примеров Number of cases	Медианный *gap* Median *gap*	Средний *gap* Mean *gap*	ALNS время, с ALNS time, s	MILP время, с MILP time, s
25	2892	−0.98%	−10.88%	0.425	3600.004
50	1475	−31.68%	−31.36%	2.421	5400.212

В проведенных экспериментах были сценарии, в которых ALNS не находил выполнимого решения из-за того, что алгоритм не мог построить начальное выполнимое решение. Процедура восстановления устроена таким образом, что она вставляет запросы в разрушенное решение только при условии соблюдения всех ограничений. Во всех таких случаях MILP также не выдавал выполнимого решения и устанавливал невыполнимость задачи. Это наблюдение справедливо для исследованного набора примеров и не дает общей гарантии успешной инициализации ALNS: не доказано, что если ALNS не смог построить начальное решение, то задача невыполнима.

Для проверки надежности найденных решений использовалось моделирование методом Монте-Карло. Для каждого решения генерировались независимые реализации времен проезда и обслуживания, после чего проверялось выполнение всех временных ограничений. Эмпирическая вероятность успешного выполнения решения обозначается через $\widehat{\Theta}$. Решение считается надежным, если выполняется неравенство

$$\widehat{\Theta} \geqslant \Theta.$$

Код ALNS реализован на Python 3.13 с использованием joblib (multiprocessing), numpy и numba. Вычисления проводились на Apple M1 Pro (8 ядер, 3.22 ГГц, 16 ГБ ОЗУ).

6.2. Сравнение ALNS и MILP. Сравнение ALNS и MILP проводилось на общих сценариях, то есть на сценариях, где оба метода нашли выполнимое решение. Для MILP использовался COPT Cardinal Optimizer [25] с ограничением времени решения: для задач размерностью 25 пунктов временной лимит составлял один час, для задач размерностью 50 пунктов — 1.5 часа. Для ALNS устанавливалось ограничение по времени в 1 час. Задача для COPT ставилась в форме MILP, описанной в работе [22].

Для каждого входного сценария выполнялось 8 параллельных запусков ALNS при одинаковых фиксированных гиперпараметрах алгоритма и параметрах стохастичности, но с разным зерном (seed). В табл. 2 и 3 приведены результаты для лучшего выполнимого решения, найденного в этих запусках.

Сравнение проводилось с MILP-реализацией той же стохастической постановки при заданных значениях cv, $\mathrm{Var}[S]$, θ_j и Θ. Детерминированные случаи с $cv = 0$ и $\mathrm{Var}[S] = 0$ использовались только как граничные сценарии в общей сетке экспериментов. В сценариях со случайными временами проезда и обслуживания оба метода проверялись по одним и тем же вероятностным ограничениям, а затем дополнительно валидировались методом Монте-Карло.

Такое сравнение используется как воспроизводимая базовая линия для оценки качества эвристики на малых размерностях. О существовании специализированного общедоступного программного решателя для рассматриваемой стохастической задачи PDPTW с жесткими временными окнами, зависящими от момента выезда, случайными временами проезда и случайными временами обслуживания автору не известно. Ближайший точный подход из литературы представлен в работе [22], но это исследовательская схема, а не готовый универсальный решатель.

Подчеркнем, что сравнение выполняется не всегда с доказанным оптимумом, а с лучшим решением, найденным MILP-решателем к моменту остановки. Поэтому случаи, когда ALNS получает меньшее значение целевой функции, чем MILP, не означают, что алгоритм ALNS находит оптимальное решение. Они показывают, что в пределах заданного лимита времени MILP не всегда успевает найти решение такого же качества.

Для оценки качества решений, полученных алгоритмом ALNS, использовался показатель *gap*:

$$gap = \frac{obj_{\mathrm{ALNS}} - obj_{\mathrm{COPT}}}{obj_{\mathrm{COPT}}}.$$

Отрицательное значение *gap* означает, что ALNS получил меньшее значение целевой функции, чем лучшее решение COPT-MILP на момент остановки.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

На задачах размерностью 25 пунктов было получено 2892 сопоставимых сценария. В 1532 случаях ALNS получил меньшее значение целевой функции, в 1350 случаях значения целевой функции совпали, и только в 10 случаях COPT-MILP получил лучшее решение. Таким образом, ALNS дал решение не хуже COPT-MILP в 99% сопоставимых сценариев.

На задачах размерностью 50 пунктов было получено 1475 сопоставимых сценариев. В 1414 случаях ALNS получил меньшее значение целевой функции, в 58 случаях результаты совпали, и только в 3 случаях COPT-MILP оказался лучше. Следовательно, при увеличении размерности преимущество ALNS по качеству решения при фиксированном времени работы MILP становится более выраженным.

Медианное значение *gap* для задач с 25 пунктами составило -0.98%, а для 50 пунктов оно достигло -31.68%. При этом медианное время работы ALNS составило 0.425 секунды для задач с 25 пунктами и 2.421 секунды для задач с 50 пунктами, тогда как медианное время COPT-MILP было близко к установленному лимиту в 1 час и 1.5 часа.

Эти результаты показывают, что при ограниченном времени вычислений ALNS обеспечивает существенно лучшее соотношение качества решения и времени работы. Особенно заметно это проявляется на задачах большей размерности, где MILP-решатель часто не успевает улучшить найденное решение до уровня, достигаемого эвристическим алгоритмом.

Результаты сравнения приведены в табл. 2. Агрегированные показатели приведены в табл. 3.

6.3. Влияние функции скорости. В табл. 4 отдельно анализируется подвыборка сопоставимых сценариев, в которых COPT-MILP завершал расчет до достижения временного лимита. Поэтому количество примеров в этой таблице меньше, чем в табл. 2 и 3. Такая подвыборка позволяет рассмотреть влияние функции скорости в условиях, когда результат MILP меньше зависит от преждевременной остановки решателя.

Таблица 4. Отклонение и медианное время решения для разных профилей скорости

Table 4. Gap and median solution time for different speed profiles

Размерность Problem size	Функция скорости Speed profile	Количество примеров Number of cases	Медианный *gap* Median *gap*	Средний *gap* Mean *gap*	ALNS время, с ALNS time, s
25	100	435	0.00%	−0.08%	0.358
25	90_110	406	0.00%	−0.28%	0.362
25	70_130	337	0.00%	−0.19%	0.382
25	50_100	265	0.00%	−0.03%	0.393
50	100	24	0.00%	−0.01%	2.169
50	90_110	24	0.00%	−0.09%	2.112
50	70_130	14	0.00%	−0.05%	1.982
50	50_100	13	−0.64%	−3.07%	1.918

Результаты показывают, что на задачах размерностью 25 пунктов медианный *gap* остается близким к нулю для всех функций скорости. Это означает, что на малой размерности ALNS и COPT-MILP часто находят одинаковые или почти одинаковые решения, если оба метода успевают построить выполнимый маршрут.

Для задач размерностью 50 пунктов влияние функции скорости становится более заметным. Наиболее сложным оказывается профиль 50_100, для которого медианный *gap* составляет -0.64%, а средний *gap* равен -3.07%. Это означает, что при более резком изменении скорости во времени MILP-решателю сложнее улучшить решение за фиксированный лимит времени, тогда как ALNS сохраняет способность быстро перестраивать маршруты.

В целом результаты по профилям скорости показывают, что усложнение структуры временной зависимости времени проезда сильнее влияет на точный метод, чем на эвристический. При этом на простых сценариях различия между ALNS и COPT-MILP остаются небольшими, а на более сложных сценариях преимущество ALNS становится заметнее. Результаты по профилям скорости приведены в табл. 4.

6.4. Влияние гиперпараметров ALNS. Напомним, что для ALNS рассматривалась сетка гиперпараметров по двум параметрам: доля разрушения решения ρ и число итераций N_{iter}. Параметр ρ определяет, какая доля запросов удаляется из текущего решения на одной итерации, а N_{iter} задает число итераций алгоритма ALNS.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

На задачах размерностью 25 пунктов (табл. 5) увеличение числа итераций с 100 до 500 заметно улучшает качество решений. При $\rho = 0.05$ доля сценариев, где COPT-MILP оказался лучше ALNS, снижается с 20.5% до 11.5%, а при $\rho = 0.25$ с 4.7% до 1.6%. Дальнейшее увеличение числа итераций с 500 до 1000 дает меньший дополнительный эффект: при $\rho = 0.25$ доля случаев, где COPT-MILP оказался лучше, снижается только с 1.6% до 1.3%, при этом медианное время ALNS увеличивается с 0.547 до 0.988 с.

Для задач размерностью 50 пунктов результаты анализировались также независимо от MILP через отклонение от лучшего найденного ALNS-решения.

Для одного и того же сценария обозначим через $obj^{\text{best}}_{\text{ALNS}}$ наименьшее значение целевой функции среди выполнимых решений, найденных в рассмотренных запусках ALNS. Относительное отклонение от этого значения определяется формулой

$$gap_{\text{ALNS}} = \frac{obj_{\text{ALNS}} - obj^{\text{best}}_{\text{ALNS}}}{obj^{\text{best}}_{\text{ALNS}}}.$$

В табл. 6 приведена медиана этого отклонения в процентах. В отличие от *gap* в разделе 6.2, здесь базой сравнения служит лучшее найденное решение ALNS, а не результат MILP.

Таблица 5. Влияние гиперпараметров ALNS на результаты для задач размерностью 25 пунктов

Table 5. Impact of ALNS hyperparameters on the results for 25-node instances

ρ	N_{iter}	COPT лучше COPT better	Выполнимое решение найдено Feasible solution found	MC валидно MC is validated	Медианный *gap* Median *gap*	Медианное ALNS время, с Median ALNS time, s
0.05	100	20.5%	89.28%	99.49%	−0.41%	0.137
0.05	500	11.5%	89.28%	99.49%	−0.98%	0.332
0.05	1000	10.6%	89.28%	99.42%	−0.98%	0.517
0.15	100	21.4%	89.28%	99.52%	−0.72%	0.143
0.15	500	15.3%	89.28%	99.50%	−0.81%	0.325
0.15	1000	13.5%	89.28%	99.44%	−0.92%	0.509
0.25	100	4.7%	87.74%	99.53%	−1.04%	0.217
0.25	500	1.6%	87.72%	99.56%	−1.49%	0.547
0.25	1000	1.3%	87.57%	99.46%	−1.76%	0.988

Таблица 6. Влияние гиперпараметров ALNS на результаты для задач размерностью 50 пунктов

Table 6. Impact of ALNS hyperparameters on the results for 50-node instances

ρ	N_{iter}	Выполнимое решение найдено Feasible solution found	MC валидно MC is validated	Медианный gap_{ALNS}, % Median gap_{ALNS}, %	Медианное ALNS время, с Median ALNS time, s	P95 ALNS время, с P95 ALNS time, s
0.05	100	84.94%	98.28%	6.97%	0.395	8.949
0.05	500	84.94%	98.25%	3.09%	1.034	8.954
0.05	1000	84.94%	98.15%	2.72%	1.525	8.937
0.15	100	83.67%	98.74%	1.27%	0.641	8.916
0.15	500	83.31%	98.80%	0.00%	1.852	8.914
0.15	1000	83.29%	98.68%	0.00%	3.104	8.915
0.25	100	83.07%	98.88%	0.17%	0.898	8.915
0.25	500	82.82%	98.89%	0.00%	2.733	8.921
0.25	1000	82.71%	98.86%	0.00%	4.589	8.921

При $\rho = 0.05$ увеличение числа итераций с 100 до 500 снижает медианный gap_{ALNS} с 6.97% до 3.09%, а при 1000 итерациях — до 2.72%. Для $\rho = 0.15$ уже при 500 итерациях медианный gap_{ALNS} становится равным 0.00%, а медианное время составляет 1.852 с. При $\rho = 0.25$ медианный gap_{ALNS} также равен 0.00% при 500 и 1000 итерациях, но медианное время выше: 2.733 и 4.589 с соответственно.

Для задач размерностью 100 пунктов сравнение с MILP не проводилось, поэтому масштабируемость ALNS оценивалась по времени работы при базовой конфигурации $\rho = 0.25$, $N_{\text{iter}} = 500$. В этой конфигурации медианное время составило 17.51 с для семейства LC, 18.68 с для LR и 19.10 с для LRC. Таким образом, даже при переходе к задачам размерностью 100 пунктов время работы ALNS остается на уровне десятков секунд.

В целом результаты показывают, что увеличение числа итераций с 100 до 500 дает основной прирост качества, тогда как переход от 500 к 1000 итерациям имеет меньший эффект и приводит к заметному росту

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

времени вычислений. Поэтому конфигурация $\rho = 0.25$, $N_{\text{iter}} = 500$ может рассматриваться как разумный компромисс: для 25 пунктов она дает малую долю случаев, где COPT-MILP оказывается лучше ALNS, для 50 пунктов обеспечивает нулевой медианный gap_{ALNS}, а для 100 пунктов сохраняет приемлемое время работы.

Результаты для задач размерностью 25 пунктов приведены в табл. 5. Результаты для задач размерностью 50 пунктов приведены в табл. 6. Зависимость времени работы от числа итераций и параметра разрушения показана на рис. 7.

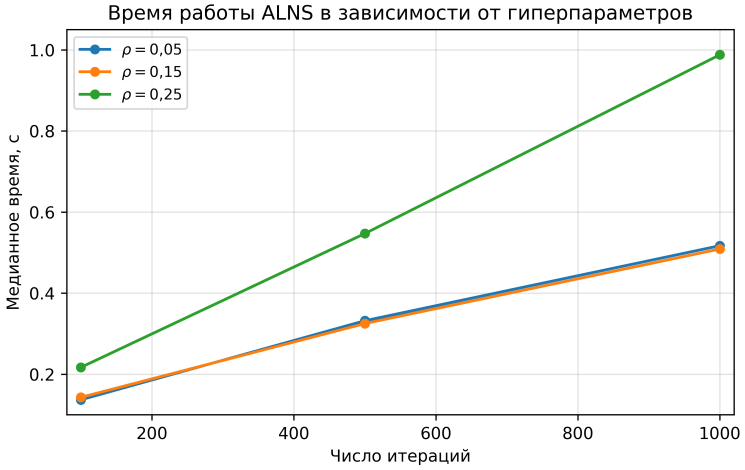


Рис. 7. Медианное время работы ALNS в зависимости от числа итераций и параметра разрушения

Fig. 7. Dependence of the median ALNS runtime on the number of iterations and the removal parameter

6.5. Влияние параметров стохастичности. Для оценки влияния стохастичности были рассмотрены различные значения дисперсии времени обслуживания и коэффициента вариации времени проезда. Параметр cv определяет дисперсию времени проезда, а параметр $\text{Var}[S]$ задает дисперсию времени обслуживания в пунктах.

Увеличение дисперсии времени проезда приводит к росту дисперсий времен прибытия и, как следствие, к снижению вероятности своевременного обслуживания пунктов маршрута. Для выполнения вероятностных ограничений алгоритм вынужден строить более консервативные маршруты, содержащие больший временной резерв. Наибольшая чувствительность ожидаемо наблюдается в сценариях с жесткими временными окнами и высокими требованиями к надежности.

На задачах размерностью 25 пунктов медианный gap остается близким к нулю для всех рассмотренных комбинаций параметров стохастичности. Средний gap также меняется незначительно: от -0.07% до -0.19%. Это говорит о том, что среди сопоставимых сценариев увеличение уровня стохастичности само по себе не приводит к заметному ухудшению качества ALNS относительно COPT-MILP.

Для задач размерностью 50 пунктов медианный gap также остается близким к нулю, однако средний gap становится более чувствительным к отдельным комбинациям параметров. Например, при $\text{Var}[S] = 1.0$ и $cv = 0.3$ средний gap равняется -1.84%, что указывает на наличие отдельных сценариев, где ALNS получает заметно лучшее решение, чем COPT-MILP с временным лимитом на поиск решения.

При интерпретации этих результатов важно учитывать, что таблица строится только по сопоставимым сценариям, где оба метода нашли выполнимые решения, а COPT-MILP завершил расчет до достижения временного лимита (та же подвыборка, что и в табл. 4). Поэтому отсутствие сильного влияния стохастичности на медианный gap не означает, что стохастичность не влияет на сложность задачи вообще. Увеличение дисперсий времен проезда и обслуживания может снижать долю выполнимых решений и усложнять проверку вероятностных ограничений, но среди уже найденных сопоставимых решений относительное качество ALNS и COPT-MILP остается близким.

Агрегированные результаты приведены в табл. 7.

6.6. Влияние требований к надежности. В работе использовались два типа вероятностных ограничений: локальный уровень надежности θ, задающий минимальную вероятность своевременного прибытия в каждый пункт маршрута, и глобальный уровень надежности Θ, задающий минимальную вероятность успешного выполнения всего решения.

Увеличение θ и Θ приводит к сужению множества выполнимых решений. Для выполнения более строгих требований по надежности алгоритм вынужден выбирать маршруты с большим запасом по времени и избегать последовательностей посещений пунктов впритык к границам временных окон. Следовательно, возникает естественный компромисс между стоимостью маршрутов и уровнем надежности: более высокие значения Θ обеспечивают большую вероятность успешного выполнения маршрутов, но требуют

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Таблица 7. Отклонение и медианное время решения для разных параметров стохастичности

Table 7. Gap and median solution time for different stochasticity parameters

Размерность Problem size	Var[S]	cv	Количество примеров Number of cases	Медианный *gap* Median *gap*	Средний *gap* Mean *gap*	ALNS время, с ALNS time, s
25	0.0	0.00	195	0.00%	−0.16%	0.420
25	1.0	0.15	317	0.00%	−0.18%	0.361
25	1.0	0.30	308	0.00%	−0.19%	0.363
25	2.0	0.15	313	0.00%	−0.16%	0.369
25	2.0	0.30	310	0.00%	−0.07%	0.363
50	1.0	0.15	18	0.00%	−0.13%	2.237
50	1.0	0.30	20	0.00%	−1.84%	2.112
50	2.0	0.15	18	0.00%	−0.16%	2.066
50	2.0	0.30	19	0.00%	−0.05%	2.282

дополнительных транспортных затрат.

На практике, как и параметры стохастичности, требования к надежности решения не оказывают существенного влияния на относительный *gap* между ALNS и COPT-MILP. Медианный *gap* равен 0.00% для всех рассмотренных уровней надежности. Это означает, что если оба метода смогли найти выполнимое решение, то повышение требований к надежности не приводит к систематическому ухудшению качества ALNS относительно COPT-MILP.

Агрегированные результаты приведены в табл. 8.

Таблица 8. Отклонение и медианное время решения для разных уровней надежности

Table 8. Gap and median solution time for different reliability levels

Размерность Problem size	θ_j	Θ	Количество примеров Number of cases	Медианный *gap* Median *gap*	Средний *gap* Mean *gap*	ALNS время, с ALNS time, s
25	0.7	0.7	482	0.00%	−0.26%	0.380
25	0.8	0.8	480	0.00%	−0.08%	0.360
25	0.9	0.9	481	0.00%	−0.11%	0.374
50	0.7	0.7	24	0.00%	−0.06%	2.132
50	0.8	0.8	25	0.00%	−1.57%	2.177
50	0.9	0.9	26	0.00%	−0.09%	1.956

6.7. Валидация решений методом Монте-Карло. Для проверки устойчивости найденных решений использовалось моделирование методом Монте-Карло. Для каждого решения генерировались независимые реализации времен проезда и обслуживания. Затем для каждой реализации проверялось, выполняются ли временные ограничения напрямую, без использования вероятностей и нормальной аппроксимации. Эмпирическая вероятность успешного выполнения решения определялась как доля сценариев, в которых все маршруты решения оказались выполнимыми.

В вычислительных экспериментах для каждого решения генерировалось 10 000 сценариев. Решение считалось прошедшим проверку методом Монте-Карло, если эмпирическая вероятность успешного выполнения была не ниже требуемого глобального уровня надежности Θ, в противном случае решение считалось ненадежным.

Результаты показывают согласованность между аналитической проверкой вероятностных ограничений внутри ALNS и последующей симуляционной проверкой. Для задач размерностью 25 пунктов доля решений, проходящих Монте-Карло проверку, среди найденных решений находилась на уровне 99.4–99.6% для разных сочетаний гиперпараметров (табл. 5). Для задач размерностью 50 и 100 пунктов соответствующая доля составляла 98.1–98.9% (табл. 6).

Это означает, что встроенная вероятностная проверка выполнимости маршрутов не только формально фильтрует решения по нормальной аппроксимации, но и хорошо согласуется с независимой симуляционной проверкой. При этом валидация методом Монте-Карло остается необходимой, поскольку анали-

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

тическая проверка использует приближенный расчет математического ожидания, дисперсии и независимую аппроксимацию вероятностей своевременного прибытия. Таким образом, результаты Монте-Карло подтверждают практическую пригодность используемой вероятностной аппроксимации для целей эвристической оптимизации.

6.8. Масштабируемость ALNS по размерности. Для анализа зависимости времени решения от размерности была построена кривая медианного времени ALNS при $N_{\text{iter}} = 500$ и $\rho = 0.25$ для семейств LC (пункты распределены группами по карте), LR (пункты равномерно распределены по всей карте) и LRC (комбинация двух способов распределения пунктов) [23, 4]. Результаты представлены на рис. 8.

Результаты показывают нелинейный рост времени работы алгоритма при увеличении размерности задачи. Для семейства LC медианное время возрастает с 0.51 с на 25 пунктах до 2.26 с на 50 пунктах и 17.51 с на 100 пунктах. Для семейства LR соответствующие значения составляют 0.65, 3.43 и 18.68 с. Для семейства LRC — 0.53, 2.74 и 19.10 с.

Такой рост времени согласуется с оценкой вычислительной сложности процедуры восстановления. При увеличении числа запросов растет количество возможных позиций вставки пунктов приемки и доставки, а каждая проверка вставки требует последовательного пересчета временных и вероятностных характеристик маршрута. Поэтому увеличение размерности приводит к росту времени работы, опережающему линейный.

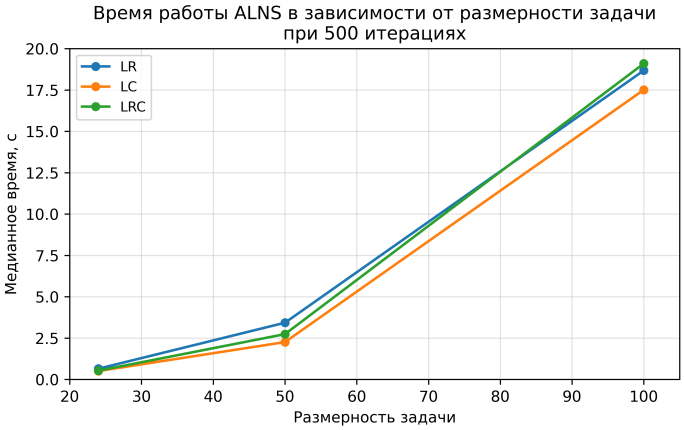


Рис. 8. Медианное время работы ALNS в зависимости от размерности задачи при 500 итерациях

Fig. 8. Dependence of the median ALNS runtime on the problem size at 500 iterations

Тем не менее даже на задачах размерностью 100 пунктов медианное время работы ALNS остается на уровне десятков секунд. Это подтверждает практическую масштабируемость предложенного эвристического подхода по сравнению с точными методами, которые уже на задачах меньшей размерности часто достигают временного лимита.

6.9. Границы применимости. ALNS представляет собой общую метаэвристическую схему, а не метод решения исключительно для транспортных задач. Например, в работе [26] адаптивный поиск в больших окрестностях применялся к смешанно-целочисленным линейным задачам общего вида. Для переноса ALNS на другую задачу необходимо определить представление решения, операторы разрушения и восстановления решения, целевую функцию и проверку ограничений.

Описанная реализация ALNS предназначена для стохастической задачи PDPTW. Все запросы на доставку известны заранее, а неопределенность связана с временами проезда и обслуживания. Для проверки надежности решения применяются допущения и аппроксимации из раздела 4. При зависимых случайных временах либо недостаточной точности нормальной аппроксимации процедуру проверки вероятностных ограничений необходимо пересматривать; выполнимое решение, найденное аналитически, требует отдельной проверки методом Монте-Карло.

Алгоритм не гарантирует нахождения глобального оптимума задачи. Возможна ситуация, когда ALNS не может построить начальное решение для задачи, что может быть как следствием несовместности ограничений задачи, так и ограничений самой эвристики в случае выполнимой задачи. В частности, увеличение дисперсий времен в пути и повышение требований к надежности решения может сделать вероятностные ограничения невыполнимыми. Экспериментальные выводы о качестве и времени работы относятся к исследованным выборкам задач.

7. Заключение. Рассмотрена стохастическая задача PDPTW со случайными временами проезда, зависящими от момента выезда, и случайными временами обслуживания. При жестких временных окнах такая постановка позволяет учитывать не только стоимость маршрута, но и вероятность его выполнения без опозданий.

Для решения задачи был предложен эвристический алгоритм на основе адаптивного поиска в боль-

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

ших окрестностях. Основное отличие предложенного подхода от стандартного ALNS состоит в том, что проверка выполнимости маршрута дополнена вероятностными ограничениями. При каждой проверке маршрута последовательно пересчитываются математические ожидания и дисперсии времен прибытия и выезда, после чего оцениваются локальные вероятности своевременного прибытия в пункты и глобальная вероятность выполнения всего решения.

Вычислительные эксперименты показали, что предложенный алгоритм позволяет получать выполнимые решения за существенно меньшее время по сравнению с MILP-моделью при фиксированном ограничении времени решения. На сопоставимых сценариях, где оба метода находили выполнимые решения, ALNS часто получал такое же значение целевой функции, как MILP, либо лучшее значение по сравнению с лучшим решением MILP на момент остановки. Этот результат не означает превосходства эвристики над точным методом в смысле оптимальности. Он показывает, что при ограниченном времени вычислений точный метод не всегда успевает найти решение сопоставимого качества. Сравнение на сценариях, где MILP завершал работу до достижения лимита времени, показало, что оба подхода часто приходят к одинаковым решениям. Основное преимущество ALNS проявляется в скорости работы и масштабируемости, особенно при увеличении размерности задачи и усложнении структуры зависимости времен проезда между пунктами от времен выезда.

Анализ гиперпараметров ALNS показал, что увеличение доли разрушения решения до $\rho = 0.25$ улучшает качество поиска, поскольку алгоритм получает возможность перестраивать более существенные части маршрутов. При этом увеличение числа итераций с 100 до 500 дает заметный прирост качества, тогда как переход от 500 к 1000 итерациям дает меньший дополнительный эффект при росте времени вычислений.

Результаты по функциям скорости показывают, что более выраженная зависимость времени проезда от момента выезда усложняет задачу для MILP-решателя. При этом ALNS сохраняет способность быстро перестраивать маршруты и находить выполнимые решения.

Влияние стохастичности и требований к надежности проявляется прежде всего через сужение множества выполнимых решений. Более высокие дисперсии времен проезда и обслуживания увеличивают неопределенность времен прибытия, а более строгие уровни надежности требуют дополнительных временных резервов. На сопоставимых сценариях эти параметры не приводят к существенному изменению медианного *gap* между ALNS и MILP, однако они важны для анализа выполнимости и устойчивости найденных решений.

Дополнительная валидация методом Монте-Карло показала высокий уровень согласованности между аналитической вероятностной проверкой, используемой внутри ALNS, и эмпирической вероятностью успешного выполнения маршрутов. Это подтверждает практическую пригодность используемых аппроксимаций математического ожидания, дисперсии и вероятностей своевременного прибытия для целей эвристической оптимизации.

Вместе с тем предложенный подход использует ряд приближений. В частности, вероятность успешного выполнения маршрута оценивается через аппроксимацию локальных вероятностей, а распространение неопределенности вдоль маршрута описывается только через первые два момента распределений. Кроме того, в работе не учитываются корреляции между временами проезда по различным дугам. Эти ограничения необходимо учитывать при интерпретации результатов.

Перспективными направлениями дальнейших исследований являются использование более точных моделей распространения неопределенности, учет коррелированных задержек на дугах, адаптивная настройка параметров ALNS.

Список литературы

1. *Solomon M.M.* Algorithms for the vehicle routing and scheduling problems with time window constraints // Operations Research. 1987. **35**, N 2. 254–265. doi 10.1287/opre.35.2.254.
2. *Braekers K., Ramaekers K., Van Nieuwenhuyse I.* The vehicle routing problem: State of the art classification and review // Computers & Industrial Engineering. 2016. **99**. 300–313. doi 10.1016/j.cie.2015.12.007.
3. *Ropke S., Pisinger D.* An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows // Transportation Science. 2006. **40**, N 4. 455–472. doi 10.1287/trsc.1050.0135.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

4. *Царьков Н.В., Голембиовский Д.Ю.* Обзор методов решения задачи о приемке и доставке с временными ограничениями. Часть II: эвристический подход // International Journal of Open Information Technologies. 2025. **13**, № 9. 91–102.
5. *Aziez I., Côté J.F., Coelho L.C.* Exact algorithms for the multi-pickup and delivery problem with time windows // European Journal of Operational Research. 2020. **284**, N 3. 906–919. doi 10.1016/j.ejor.2020.01.040.
6. *Dumas Y., Desrosiers J., Soumis F.* The pickup and delivery problem with time windows // European Journal of Operational Research. 1991. **54**, N 1. 7–22. doi 10.1016/0377-2217(91)90319-Q.
7. *Desaulniers G., Desrosiers J., Solomon M.M. (eds.)* Column Generation. New York: Springer, 2005. doi 10.1007/b135457.
8. *Бронштейн Е.М., Гиндуллина Э.В., Гиндуллин Р.В.* Формализации задач погрузки и доставки // Вестн. Южно-Урал. гос. ун-та. Сер. Матем. Мех. Физ. 2017. **9**, № 1. 13–21. doi 10.14529/mmph170102.
9. *Nanry W.P., Barnes J.W.* Solving the pickup and delivery problem with time windows using reactive tabu search // Transportation Research Part B: Methodological. 2000. **34**, N 2. 107–121. doi 10.1016/S0191-2615(99)00016-8.
10. *Li H., Lim A.* A metaheuristic for the pickup and delivery problem with time windows // Proc. 13th IEEE Int. Conf. on Tools with Artificial Intelligence. ICTAI 2001, Dallas, TX, USA, November 07–09, 2001. IEEE Press, 2001. pp. 160–167. doi 10.1109/ICTAI.2001.974461.
11. *Shaw P.* Using constraint programming and local search methods to solve vehicle routing problems // Principles and Practice of Constraint Programming–CP98. Pisa, Italy, October 26–30, 1998. Lecture Notes in Computer Science. Vol. 1520. Berlin, Springer, 1998. pp. 417–431. doi 10.1007/3-540-49481-2_30.
12. *Юськов А.Д., Кулаченко И.Н., Мельников А.А., Кочетов Ю.А.* Гибридный алгоритм для двухкритериальной задачи оптимизации трафика в сети // Дискретн. анализ и исслед. опер. 2025. **32**, № 3. 117–144. doi 10.33048/daio.2025.32.827.
13. *Pisinger D., Ropke S.* A general heuristic for vehicle routing problems // Computers & Operations Research. 2007. **34**, N 8. 2403–2435. doi 10.1016/j.cor.2005.09.012.
14. *Gendreau M., Laporte G., Séguin R.* Stochastic vehicle routing // European Journal of Operational Research. 1996. **88**, N 1. 3–12. doi 10.1016/0377-2217(95)00050-X.
15. *Ritzinger U.M., Puchinger J., Hartl R.F.* A survey on dynamic and stochastic vehicle routing problems // International Journal of Production Research. 2016. **54**, N 1. 215–231. doi 10.1080/00207543.2015.1043403.
16. *Oyola J., Arntzen H., Woodruff D.L.* The stochastic vehicle routing problem, a literature review, Part I: models // EURO Journal on Transportation and Logistics. 2018. **7**, N 3. 193–221. doi 10.1007/s13676-016-0100-5.
17. *Cai J., Zhu Q., Lin Q., et al.* A survey of dynamic pickup and delivery problems // Neurocomputing. 2023. **554**, N C. Article Number 126631. doi 10.1016/j.neucom.2023.126631.
18. *Ghilas V., Demir E., Van Woensel T.* A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands // Transportation Research Part B: Methodological. 2016. **91**, N C. 34–51. doi 10.1016/j.trb.2016.04.015.
19. *Györgyi P., Kis T.* A probabilistic approach to pickup and delivery problems with time window uncertainty // European Journal of Operational Research. 2019. **274**, N 3. 909–923. doi 10.1016/j.ejor.2018.10.031.
20. *Ichoua S., Gendreau M., Potvin J.-Y.* Vehicle dispatching with time-dependent travel times // European Journal of Operational Research. 2003. **144**, N 2. 379–396. doi 10.1016/S0377-2217(02)00147-9.
21. *Sun P., Veelenturf L.P., Hewitt M., Van Woensel T.* The time-dependent pickup and delivery problem with time windows // Transportation Research Part B: Methodological. 2018. **116**. 1–24. doi 10.1016/j.trb.2018.07.002.
22. *Wang Z., Dessouky M., Van Woensel T., Ioannou P.* Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times // EURO Journal on Transportation and Logistics. 2023. **12**, N 2. Article Number 100099. doi 10.1016/j.ejtl.2022.100099.
23. *Царьков Н.В., Голембиовский Д.Ю.* Обзор методов решения задачи о приемке и доставке с временными ограничениями. Часть I: точный подход // International Journal of Open Information Technologies. 2025. **13**, № 7. 60–70. https://injoit.ru/index.php/j1/article/viewFile/2160/1934. Дата обращения: 23.10.2026.
24. *Jula H., Dessouky M., Ioannou P.A.* Truck route planning in nonstationary stochastic networks with time windows at customer locations // IEEE Transactions on Intelligent Transportation Systems. 2006. **7**, N 1. 51–62. doi 10.1109/TITS.2006.869596.
25. COPT Cardinal Optimizer. https://www.shanshu.ai/copt. Дата обращения: 23.10.2026.
26. *Hendel G.* Adaptive large neighborhood search for mixed integer programming // Mathematical Programming Computation. 2022. **14**. 185–221. doi 10.1007/s12532-021-00209-7.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

Получена 17 июля 2026 г. | *Принята* 09 октября 2026 г. | *Опубликована* 10 ноября 2026 г.

Информация об авторе

Никита Владимирович Царьков — аспирант; Московский государственный университет имени М. В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, д. 1, стр. 52, 119991, Москва, Российская Федерация.

References

1. M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," Operations Research **35** (2), 254–265 (1987). doi 10.1287/opre.35.2.254.
2. K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The Vehicle Routing Problem: State of the Art Classification and Review," Computers & Industrial Engineering **99**, 300–313 (2016). doi 10.1016/j.cie.2015.12.007.
3. S. Ropke and D. Pisinger, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," Transportation Science **40** (4), 455–472 (2006). doi 10.1287/trsc.1050.0135.
4. N. V. Tsarkov and D. Yu. Golembiovsky, "A Review of Methods for Solving the Pickup and Delivery Problem with Time Windows. Part II: A Heuristic Approach," International Journal of Open Information Technologies **13** (9), 91–102 (2025). [in Russian].
5. I. Aziez, J.-F. Côté, and L. C. Coelho, "Exact Algorithms for the Multi-Pickup and Delivery Problem with Time Windows," European Journal of Operational Research **284** (3), 906–919 (2020). doi 10.1016/j.ejor.2020.01.040.
6. Y. Dumas, J. Desrosiers, and F. Soumis, "The Pickup and Delivery Problem with Time Windows," European Journal of Operational Research **54** (1), 7–22 (1991). doi 10.1016/0377-2217(91)90319-Q.
7. G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), *Column Generation* (Springer, New York, 2005). doi 10.1007/b135457.
8. E. M. Bronshtein, E. V. Gindullina, and R. V. Gindullin, "Formalization of Pickup and Delivery Problem," Vestn. Yuzhno-Ural. Gos. Un-ta. Ser. Matem. Mekh. Fiz. **9** (1), 13–21 (2017). [in Russian]. doi 10.14529/mmph170102.
9. W. P. Nanry and J. W. Barnes, "Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search," Transportation Research Part B: Methodological **34** (2), 107–121 (2000). doi 10.1016/S0191-2615(99)00016-8.
10. H. Li and A. Lim, "A Metaheuristic for the Pickup and Delivery Problem with Time Windows," in *Proc. 13th IEEE Int. Conf. on Tools with Artificial Intelligence. ICTAI 2001, Dallas, TX, USA, November 07–09, 2001* (IEEE Press, 2001), pp. 160–167. doi 10.1109/ICTAI.2001.974461.
11. P. Shaw, "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems," in *Principles and Practice of Constraint Programming–CP98, Pisa, Italy, October 26–30, 1998.* Lecture Notes in Computer Science, Vol. 1520. (Springer, Berlin, 1998), pp. 417–431. doi 10.1007/3-540-49481-2_30.
12. A. D. Yus'kov, I. N. Kulachenko, A. A. Mel'nikov, and Y. A. Kochetov, "A Hybrid Algorithm for a Two-Objective Traffic Engineering Problem," Diskretn. Anal. Issled. Oper. **32** (3), 117–144 (2025). doi 10.33048/daio.2025.32.827.
13. D. Pisinger and S. Ropke, "A General Heuristic for Vehicle Routing Problems," Computers & Operations Research **34** (8), 2403–2435 (2007). doi 10.1016/j.cor.2005.09.012.
14. M. Gendreau, G. Laporte, and R. Séguin, "Stochastic Vehicle Routing," European Journal of Operational Research **88** (1), 3–12 (1996). doi 10.1016/0377-2217(95)00050-X.
15. U. M. Ritzinger, J. Puchinger, and R. F. Hartl, "A Survey on Dynamic and Stochastic Vehicle Routing Problems," International Journal of Production Research **54** (1), 215–231 (2016). doi 10.1080/00207543.2015.1043403.
16. J. Oyola, H. Arntzen, and D. L. Woodruff, "The Stochastic Vehicle Routing Problem, a Literature Review, Part I: Models," EURO Journal on Transportation and Logistics **7** (3), 193–221 (2018). doi 10.1007/s13676-016-0100-5.
17. J. Cai, Q. Zhu, Q. Lin, et al., "A Survey of Dynamic Pickup and Delivery Problems," Neurocomputing **554** (C), Article Number 126631 (2023). doi 10.1016/j.neucom.2023.126631.
18. V. Ghilas, E. Demir, and T. Van Woensel, "A Scenario-Based Planning for the Pickup and Delivery Problem with Time Windows, Scheduled Lines and Stochastic Demands," Transportation Research Part B: Methodological **91** (C), 34–51 (2016). doi 10.1016/j.trb.2016.04.015.

2026, **27** (4), 666–689. doi 10.26089/NumMet.v27r441

19. P. Györgyi and T. Kis, "A Probabilistic Approach to Pickup and Delivery Problems with Time Window Uncertainty," European Journal of Operational Research **274** (3), 909–923 (2019). doi 10.1016/j.ejor.2018.10.031.
20. S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle Dispatching with Time-Dependent Travel Times," European Journal of Operational Research **144** (2), 379–396 (2003). doi 10.1016/S0377-2217(02)00147-9.
21. P. Sun, L. P. Veelenturf, M. Hewitt, and T. Van Woensel, "The Time-Dependent Pickup and Delivery Problem with Time Windows," Transportation Research Part B: Methodological **116**, 1–24 (2018). doi 10.1016/j.trb.2018.07.002.
22. Z. Wang, M. Dessouky, T. Van Woensel, and P. Ioannou, "Pickup and Delivery Problem with Hard Time Windows Considering Stochastic and Time-Dependent Travel Times," EURO Journal on Transportation and Logistics **12** (2), Article Number 100099 (2023). doi 10.1016/j.ejtl.2022.100099.
23. N. V. Tsarkov and D. Yu. Golembiovsky, "A Review of Methods for Solving the Pickup and Delivery Problem with Time Windows. Part I: An Exact Approach," International Journal of Open Information Technologies **13** (7), 60–70 (2025). [in Russian].
24. H. Jula, M. Dessouky, and P. A. Ioannou, "Truck Route Planning in Nonstationary Stochastic Networks with Time Windows at Customer Locations," IEEE Transactions on Intelligent Transportation Systems **7** (1), 51–62 (2006). doi 10.1109/TITS.2006.869596.
25. COPT Cardinal Optimizer. https://www.shanshu.ai/copt. Cited October 23, 2026.
26. G. Hendel, "Adaptive Large Neighborhood Search for Mixed Integer Programming," Mathematical Programming Computation **14**, 185–221 (2022). doi 10.1007/s12532-021-00209-7.

Received *July 17, 2026*
Accepted *October 09, 2026*
Published *November 10, 2026*

Information about the author

Nikita V. Tsarkov — PhD student; Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Leninskie Gory, 1, building 52, 119991, Moscow, Russia.