
ИНФОРМАТИКА И УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ И СОЦИАЛЬНЫХ СИСТЕМАХ

УДК 656.2

DOI: 10.46960/1816-210X_2023_3_7

ОПТИМИЗАЦИЯ ЗАГРУЗКИ ВАГОНА НА ОСНОВЕ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ С МЕМОИЗАЦИЕЙ И КОЭФФИЦИЕНТОМ НЕУДОБСТВА

Л.Ю. КатаеваORCID: 0000-0001-9217-9324 e-mail: kataeval2010@mail.ru

Нижегородский государственный технический университет им. Р.Е. Алексеева
Филиал Самарского государственного университета путей сообщения
в г. Нижнем Новгороде
Нижний Новгород, Россия

Д.А. МасленниковORCID: 0000-0002-7819-4007 e-mail: dmitrymaslennikov@mail.ru

Нижегородский государственный технический университет им. Р.Е. Алексеева
Нижний Новгород, Россия

На основе решения типовых задач упаковки показано, что классические эвристики хорошо работают для упаковки разнообразных объектов по размеру и уступают предложенному методу на основе программирования с мемоизацией и определением коэффициента неудобства для каждого типа объектов при большом количестве размещаемых объектов каждого типа. Отличительная особенность данной задачи – наличие большого количества грузов с ограниченным набором их типов. Для решения можно использовать метод динамического программирования, суть которого можно свести к рекурсивному разбиению загружаемого вагона размещаемым объектом на подконтейнеры по плоскостям. В дальнейшем вагон также может называться подконтейнером. Недостатком такого подхода является избыточное разбиение пространства за счет гильотинной резки даже в случае отсутствия ее необходимости, что несколько снижает эффективность упаковки, но позволяет получить оптимальное решение в такой постановке. Предлагаемый алгоритм позволяет осуществить более плотную упаковку, минимизируя количество используемых контейнеров. Предложенное решение актуально в условиях интенсификации эксплуатации подвижного состава и перехода на высокоскоростное движение, особенно при использовании прямоугольных объектов.

Ключевые слова: загрузка вагона, оптимизация, динамическое программирование, коэффициент неудобства.

ДЛЯ ЦИТИРОВАНИЯ: Катаева, Л.Ю. Оптимизация загрузки вагона на основе динамического программирования с мемоизацией и коэффициентом неудобства / Л.Ю. Катаева, Д.А. Масленников // Труды НГТУ им. Р.Е. Алексеева. 2023. № 3. С. 7-17. DOI: 10.46960/1816-210X_2023_3_7

OPTIMIZATION OF WAGON LOADING BASED ON DYNAMIC PROGRAMMING WITH MEMOIZATION AND INCONVENIENCE FACTOR

L.Yu. KataevaORCID: 0000-0001-9217-9324 e-mail: kataeval2010@mail.ru

Nizhny Novgorod State Technical University n.a. R.E. Alekseev
Branch of Samara State Transport University in the city of Nizhny Novgorod
Nizhny Novgorod, Russia

D.A. Maslennikov

ORCID: **0000-0002-7819-4007** e-mail: **dmitrymaslennikov@mail.ru**

Nizhny Novgorod State Technical University n.a. R.E. Alekseev

Nizhny Novgorod, Russia

Abstract. Under conditions of more intensive operation of rolling stock and transition to high-speed traffic, the issues of more efficient operation of rolling stock become urgent. The use of typical rectangular objects for transportation makes this task urgent. A distinctive feature of the problem under consideration is the presence of a large number of cargoes with a limited set of types. To solve this problem we can use the method of dynamic programming, the essence of which can be reduced to the recursive division of the loaded wagon by the placed object into subcontainers on the planes. Hereafter, the wagon can also be referred to as a subcontainer. The disadvantage of this approach is excessive space partitioning due to guillotine cutting, even if it is not necessary, which somewhat reduces the efficiency of packaging, but it allows us to obtain an optimal solution in this formulation. It is shown that classical heuristics work well for packing objects of different sizes and give way to the proposed method based on programming with memoization and defining inconvenience coefficients for each type of objects with a large number of objects of each type. This allows for more dense packing by minimizing the number of containers used.

Key words: loading of a railway car, optimization, dynamic programming, inconvenience coefficient.

FOR CITATION: L.Yu. Kataeva, D.A. Maslennikov. Optimization of wagon loading based on dynamic programming with memoization and inconvenience factor. Transactions of NNSTU n.a. R.E. Alekseev. 2023. № 3. Pp. 7-17.
DOI: 10.46960/1816-210X_2023_3_7

Введение

Решение задачи оптимального размещения грузов в вагоне позволяет увеличить эффективность их эксплуатации и существенной экономии ресурсов. Данные задачи являются NP-трудными задачами и даже при применении эвристических методов достаточно сложны для 3D-случая [1, 2]. Разработанные алгоритмы для данной задачи [3, 4] и комбинаторные методы имеют экспоненциально растущую сложность [5]. Динамическое программирование позволяет оценивать последствия каждого шага на горизонт вплоть до полного решения, однако данный подход применяется для более простых задач (упаковка груза в одномерные контейнеры). Следует отметить всплеск интереса к таким задачам, начиная с 2012 г., что может объясняться появлением новых технических перспектив полной автоматизации процесса погрузки и новых возможностей вычислительных технологий [6-9]. Наибольшее количество научных исследований проводится в сфере разработки алгоритмов решения задачи о размещении блоков заданных размеров в контейнере-параллелепипеде.

Начиная с первых железнодорожных перевозок и по настоящее время, люди постоянно обращаются к задаче об оптимальной загрузке вагонов при перевозке грузов. На заре развития перевозок люди использовали технические приемы, позволяющие обеспечить безопасность и сохранность грузов (рис. 1). Впоследствии появление вычислительных машин и развитие математической науки позволило создать большое разнообразие алгоритмов для решения поставленной задачи (рис. 2).

Алгоритмы, в основе которых лежит динамическое программирование в качестве целевой функции, как правило, используют минимизацию затрат на перевозки при заданных ограничениях. Эволюционные принципы используются при реализации различных видов генетических алгоритмов с учетом доступного пространства для укладки грузов и ограничений на размеры ящиков. Методы грубой силы по своей сути являются алгоритмами перебора всех возможных вариантов укладки. Такие методы работают медленно и, как правило, с ограниченным набором ящиков. Эвристические алгоритмы (рой частиц, отжиг и др.) используют, по сути, стохастический поиск. Они более производительны и не имеют жестких размеров на количество укладываемых предметов и их разнообразие. При этом следует отметить, что каждый из данных численных алгоритмов находит применение в решении рассматриваемого класса задач в зависимости от конкретных условий.

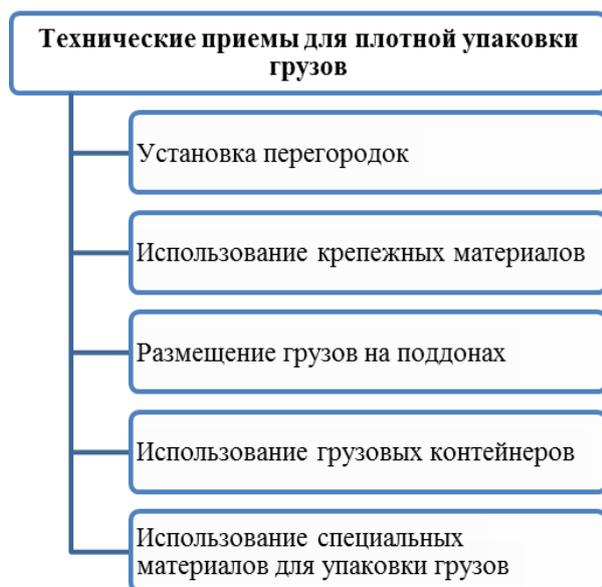


Рис. 1. Наиболее распространенные технические приемы, применяемые на железнодорожном транспорте

Fig. 1. The most common technical methods used in rail transport

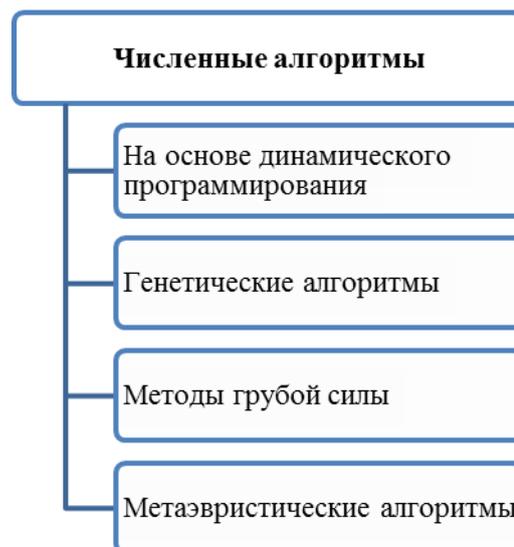


Рис. 2. Численные алгоритмы для решения задачи оптимизации упаковки грузов

Fig. 2. Numerical algorithms for solving the problem of packing optimization

Эффективность каждого из перечисленных методов зависит от ограничений на размеры и количество, требований к оптимальности, количеству укладываемых грузов и др. Именно большое количество параметров задачи определяет, какой из перечисленных алгоритмов наиболее приемлем в конкретном случае. При этом надо учитывать, что методы полного перебора весьма неэффективны при больших наборах данных и конкретных требованиях к результату, зато генетические и метаэвристические методы с данными задачами справляются более успешно. Для упрощения алгоритмов и увеличения их эффективности при решении задачи о загрузке вагона группами контейнеров разного размера можно использовать наборы ключевых эвристик и их комбинаций (рис. 3).

В настоящее время, в связи с возросшим грузооборотом на транспорте, возрастает интерес к задаче об оптимальной загрузке вагонов. С одной стороны, разработка современных алгоритмов позволит не только решить задачу эффективной упаковки грузов в вагоне, но и даст возможность автоматизировать сам процесс погрузки, создавая карту последовательности с указанием места положения каждого груза. С другой стороны, более плотная укладка уменьшает вероятность повреждения груза и обеспечивает сбалансированное распределение груза с учетом центра тяжести вагона и равномерного распределения нагрузки по вагону. Все это может повысить безопасность и надежность перевозки грузов. Напротив, в результате неправильной укладки грузов в вагоне ущерб может быть причинен не только грузоотправителю и грузополучателю, но также и компании, осуществляющей перевозку.

Особо отметим, что при нарушениях правил перевозки опасных грузов последствия могут сказаться не только на экономике региона, но и страны в целом. Вот почему авторы полагают, что решение данной задачи на современном этапе развития железнодорожного транспорта становится основой эффективности и экономичности транспортировки грузов. Существующие алгоритмы предназначены для упаковки грузов различных размеров, а случай ограниченного количества типов предметов недостаточно исследован. Эти алгоритмы далеко не совершенны и имеют ряд ограничений, не способствующих их успешному внедрению в процесс полной автоматизации процесса загрузки вагонов. В то же время развитие ма-

тематики и техники позволяет взглянуть на решение задачи оптимальной загрузки вагонов по-новому, находя нетривиальные решения для задачи загрузки вагонов типовыми грузами.

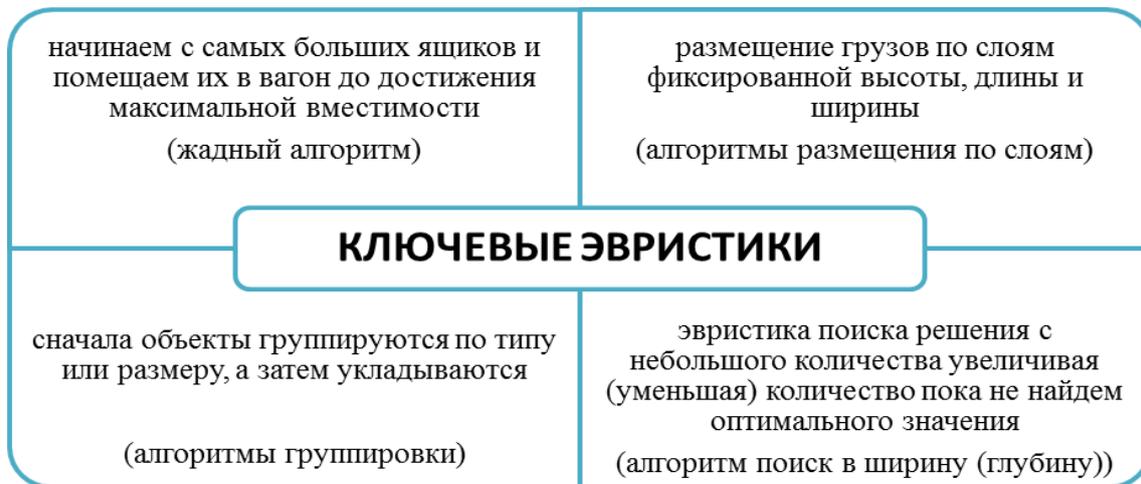


Рис. 3. Эвристики, часто используемые при решении задачи оптимизации загрузки вагона

Fig. 3. Frequently used heuristics used in solving the wagon load optimization problem

Цель данного исследования – разработка более простого подхода к решению задачи оптимальной загрузки вагонов ограниченным набором типов грузов при их количестве введением алгоритма склейки подконтейнеров, снижения размерности задачи до трех и снятия ограничения на горизонт планирования.

Формулировка задачи об оптимальной загрузке вагонов ограниченным набором типов грузов при неограниченном их количестве и методы ее решения

Предположим, что имеется множество вагонов, в каждом из которых есть пространство для груза в форме прямоугольного параллелепипеда. Требуется загрузить все имеющиеся объекты в минимально возможное количество вагонов. В качестве целевой функции рассматривается отношение объема объектов к отношению объема использованных вагонов. Для теоретического исследования при вычислении целевой функции объем последнего вагона принимается равным объему груза в нем. Этот прием позволяет более сравнивать различные алгоритмы размещения груза.

Рассматривается задача размещения n типов объектов с размерами $c_{i,j}$, в минимально возможное количество контейнеров с размерами b_j . Для каждого объекта также определяются допустимые варианты вращения. Количество каждого типа объекта q_i . Общее количество объектов h . Код допустимых типов вращения req_i . Центр масс каждого объекта соответствует его геометрическому центру. $i = \overline{1, n}$ – номер типа объекта, $j = \overline{1, 3}$ – номер координаты. Вращение груза рассматривается как перестановка размеров $[c_{i,1}, c_{i,2}, c_{i,3}]$. Код допустимых типов вращения определяет допустимые значения типов поворота для всех объектов соответствующего типа с учетом соотношений:

$$req_i = \begin{cases} 1, & \text{запрещено вращать данный объект} (r = 1) \\ 2, & \text{разрешается горизонтальное вращение} (1 \leq r \leq 2) \\ 3, & \text{разрешается любое вращение} (1 \leq r \leq 6) \end{cases}$$

Типы поворотов $r = \overline{1,6}$ определяются перестановками размеров соответственно $[c_{i,1}, c_{i,2}, c_{i,3}] [c_{i,2}, c_{i,1}, c_{i,3}] [c_{i,1}, c_{i,3}, c_{i,2}] [c_{i,3}, c_{i,1}, c_{i,2}] [c_{i,2}, c_{i,3}, c_{i,1}] [c_{i,3}, c_{i,2}, c_{i,1}]$.

Постановка не включает требования к наличию опоры для последующих объектов, т.е. допустимо размещение объектов, при котором они не перекрываются друг с другом, не выходя за границы контейнера. Решением задачи оптимизации является определение числа контейнеров n_b , для каждого объекта номера контейнера, куда его следует разместить, типа поворота и положения. Формально такое решение представляется набором переменных n_b – число контейнеров; $x_{m,0}$ – номер контейнера, в который помещается объект; $x_{m,1}, x_{m,2}, x_{m,3}$ – минимальные координаты объекта в контейнере, r_m – тип поворота; m – номер объекта. Для формулирования ограничения обозначим координаты каждого объекта как номер контейнера и координаты в нем, что задает пространство $\{1, 2, \dots, n_b\} \times R^3$. Номер контейнера обозначается нулевой координатой, а пространственное положение координатами от 1 до 3. В таком пространстве множество точек контейнера определяется соотношением:

$$B_i = \{p \in \{1, 2, \dots, n_b\} \times R^3 \mid (p_0 = t) \wedge (0 \leq p_1 \leq b_1) \wedge (0 \leq p_2 \leq b_2) \wedge (0 \leq p_3 \leq b_3)\}.$$

Множество точек объекта определяется:

$$O_m = \{p \in \{1, 2, \dots, n_b\} \times R^3 \mid (p_0 = x_{m,0}) \wedge (x_{m,1} \leq p_1 \leq x_{m,1} + c_{m(i,r_1)}) \wedge (x_{m,2} \leq p_2 \leq x_{m,2} + c_{m(i,r_2)}) \wedge (x_{m,3} \leq p_3 \leq x_{m,3} + c_{m(i,r_3)})\}$$

где $c_{m(i,r_j)}$ – размеры объекта m . Тройка чисел (i, r, j) определяет соответственно номер типа объекта, номер поворота и номер координаты.

Ограничения, накладываемые на размещение объектов в контейнерах: $\forall m_1, m_2, m_1 \neq m_2 : \mu(O_{m_1} \cap O_{m_2}) = 0$, где μ – мера множества, соответствующая объему множества. Это ограничение требует, чтобы объекты не перекрывались, при этом допускается только пересечение границ объектов. В данной постановке предметы не могут выходить за границы контейнеров, а также запрещено размещать предметы в большем числе контейнеров, чем было выбрано $\forall m : \mu\left(O_m \setminus \left(\bigcup_{t=1}^{n_b} B_t\right)\right) = 0$.

Целью размещения является минимизация числа используемых контейнеров для размещения всех объектов. Для более эффективного сравнения различных подходов к решению задачи в качестве целевой функции, из числа использованных контейнеров вычитается доля незанятого объема в последнем контейнере. Такой подход позволяет выявить более эффективный метод упаковки объектов, даже если он не привел к уменьшению числа контейнеров. Целевая функция определяется соотношением $F = n_b - 1 + V \rightarrow \min$, где V – доля объема последнего контейнера, заполненного объектами.

В данной работе рассматривается случай ограниченного количества размеров однотипных прямоугольных объектов. Для ее решения можно использовать метод динамического программирования, суть которого сводима к рекурсивному разбиению контейнера размещаемым объектом на подконтейнеры по плоскостям. В дальнейшем контейнер также может называться подконтейнером. Недостатком такого подхода является избыточное разбиение пространства за счет гильотинной резки даже в случае отсутствия ее необходимости, что несколько снижает эффективность упаковки, но позволяет получить оптимальное решение в такой постановке. Еще одним недостатком классического подхода динамического программирования является увеличение размерности таблицы для хранения результатов на число типов объектов, что может приводить к неприемлемому увеличению объема вычислений и оперативной памяти.

Для устранения описанных недостатков в данной работе предложено отказаться от учета числа оставшихся предметов и введения алгоритма склейки подконтейнеров, имеющих

общую грань. Вместо учета числа оставшихся объектов используется бинарная переменная, показывающая, есть ли предметы данного типа, которые следует упаковать. Недостаток такого подхода проявляется в случае, когда объекты данного типа закончились, в алгоритм динамического программирования оценивал каждый подконтейнер с учетом возможности размещения объектов этого типа. В этом случае алгоритм динамического программирования пересчитывает ценность объемов уже без учета возможности размещения данного предмета, что приводит к снижению эффективности упаковки. Достоинством данного подхода является снижение размерности таблицы в алгоритме динамического программирования до размерности геометрии, что дает возможность выполнять решение за разумное время. В отличие от известных эвристик, такой подход позволяет оценить эффект укладки в контейнер каждого объекта без ограничения горизонта планирования.

Проблему необоснованного разделения объемов за счет гильотинного разреза можно смягчить путем введения процедуры склеивания объемов, имеющих смежные грани. Суть этой процедуры состоит в том, что при разбиении объема на подконтейнеры выполняется проверка того, есть ли у него общая грань со всеми остальными незаполненными подконтейнерами. В этом случае происходит объединение двух смежных подконтейнеров в один. За счет использования хэш-таблиц время выполнения данного процесса не зависит от числа проверяемых подконтейнеров. Процедура склеивания позволяет существенно увеличить эффективность упаковки объектов. Алгоритм динамического программирования не учитывает повышение эффективности упаковки за счет склейки подконтейнеров, что может приводить к недооценке склеиваемых объемов. Данный недостаток сглаживается в случае, если алгоритм может размещать сразу группу однотипных объектов. В этом случае уменьшается потребность в склеивании подконтейнеров и их оценка более точна.

Предложенный в данном исследовании алгоритм динамического программирования реализован в отдельной функции DP . В классическом варианте реализации данного метода объем вычислений напрямую зависит от структуры таблицы для хранения промежуточных результатов. Размерность таблицы определяется как сумма числа измерений геометрии и типов объектов (если они учитываются в таблице), что может приводить к нарастанию вычислительной сложности, выходящей за пределы практической применимости. В настоящей работе применяется упрощенная версия алгоритма, которая не учитывает количество оставшихся предметов. Это позволяет сократить размерность задачи до размерности геометрии. Для повышения производительности используется мемоизация, благодаря этому вычисления выполняются только для тех размеров подконтейнеров, которые используются.

В качестве входных аргументов выступают вектор (B), состоящий из значений размеров анализируемого контейнера (подконтейнера), бинарный вектор (Q), показывающая для каждого типа объекта его наличие или отсутствие и вектор весов объектов (W), показывающий ценность единицы объема для каждого объекта, предназначенный для работы различных эвристик. Выходными аргументами в рамках одной задачи однозначно определяются для подконтейнера с размерами B с учетом возможности положить в него доступные объекты согласно вектору Q и их весам W являются оценка (v) подконтейнера (то есть максимальная ценность объектов, которую можно достичь, размещая объекты в подконтейнере с точки зрения алгоритма динамического программирования), вектор (P), содержащий число объектов каждого типа, размещаемых в подконтейнере, номер типа объекта (ib), размещение которого в углу данного подконтейнера обеспечит максимальную оценку всего подконтейнера, номер перестановки размеров объекта (rb), соответствующий его вращению (может принимать целые значения от 1 до 6 по числу перестановок трех размеров)

Для решения поставленной задачи на первом этапе необходимо проверить, есть ли вычисленный результат для данного набора аргументов (B, Q, W), к которому имеется доступ в кэше. Если результат уже есть, он может быть использован без повторного вычисления. Если результат отсутствует в кэше, происходит переход к алгоритму, и результаты вычислений добавляются в кэш.

Далее, для каждого объекта и возможных способов его расположения выполняется попытка поместить его в текущий подконтейнер. Если это возможно, оставшееся пространство разбивается на три подконтейнера. Затем для каждого из них рекурсивно запускается алгоритм с теми же значениями аргументов Q и W . Таким образом, для объекта i и его поворота r определяется следующая величина:

$$v_{i,r} = (c_{i,1}c_{i,2}c_{i,3})W_i + DP(B_x, Q, W) + DP(B_y, Q, W) + DP(B_z, Q, W),$$

где B_x, B_y, B_z – представляют собой размеры прямоугольных подконтейнеров, которые вместе с объектом под номером i и с учетом его поворота r полностью заполняют контейнер V . Значения функции DP вычисляются рекурсивно с учетом кэширования. После проведения описанной процедуры для всех типов объектов и всех их допустимых поворотов, определяются значения i и r , которые обеспечат максимальное значение оценки подконтейнера (v). Такое решение является оптимальным с точки зрения динамического программирования.

Разработанный алгоритм решения задачи оптимизации загрузки обладает весьма полезной особенностью – возможностью совмещения с другими эвристиками и метаэвристиками. Наличие индикатора эффективности размещения каждого эффекта, работающего без ограничения горизонта планирования, может быть добавлено в качестве корректирующего коэффициента, что может повысить эффективность использования эвристик. Добавление каждого объекта в контейнер не требует полного пересчета таблицы динамического программирования, что обеспечивает высокую производительность при совмещении данного результата с эвристиками.

Для повышения эффективности работы алгоритма вводится величина характеризующая неудобство объекта для загрузки, которые будут в последствии загружаться последними. Для объекта типа i относительное неудобство (W_i) вычисляется по формуле и определяет его ценность:

$$W_i = (b_1b_2b_3) / DP(b, Q_i, W'),$$

где Q_i – вектор из n элементов, в котором координата с номером i равна 1, а остальные 0, $W' = 1$ – вектор из n единиц.

Полученная оценка соответствует части объема контейнера, который может быть заполнен, используя только предметы одного типа. Чем больше заполненная объемная доля контейнера, тем более удобным считается этот тип объектов. По результатам вычислений вес каждого объекта назначается равным относительному неудобству (W_i). Использование данной эвристики приводит к приоритетному помещению неудобных объектов. Удобные объекты помещаются в том случае, когда это способствует заполняемости контейнера за счет комбинирования типов объектов.

На следующем этапе переходим к алгоритму «Размещение объектов». При создании нового контейнера, весь его объем помещается в единственный подконтейнер в стеке. Для верхнего подконтейнера применяется функция «Динамическое программирование» с учетом текущих весов и доступных на данный момент типов объектов, с помощью которой определяется, какой объект и в каком положении следует разместить. Если таких объектов нет, данный подконтейнер удаляется. В противном случае выполняется помещение объекта, а оставшееся пространство разбивается на три подконтейнера методом, аналогичным описанному в алгоритме динамического программирования. После этого каждый новый подконтейнер добавляется в стек и выполняется поиск подконтейнеров со смежными гранями. Если поиск завершается успехом, выполняется их слияние. Это снижает негативные эффекты от гильотинного разбиения пространства. Если объекты всех типов закончились, то задача решена. Если в данном контейнере не остается подконтейнеров, контейнер считается полностью упакованным, и для оставшихся объектов создается следующий контейнер.

Результаты исследования и их обсуждение

Для тестирования предложенного алгоритма была рассмотрена модельная задача по укладке 8-ми предметов W1 размерами $10 \times 10 \times 2$ и 16-ти предметов W2 размерами $2 \times 2 \times 2$ без ограничений вращения в контейнеры размером $12 \times 12 \times 12$. Данная задача имеет оптимальное решение, сводящееся к полной загрузке одного контейнера всеми имеющимися предметами. Схема размещения показана на рис. 4. Это решение очевидно является оптимальным. Решение, основанное на методе [10, 11] обеспечивает загрузку только 4 из 8 предметов W1 и 16 предметов W2, что соответствует заполняемости 53,7 %.

Основной причиной высокой эффективности алгоритма, предложенного в настоящей работе для данного примера, является его способность оценивать ценность подконтейнеров с учетом возможности размещения в них объектов на полную глубину планирования. Однако данное преимущество имеет место только для ограниченного класса задач с большим количеством однообразных объектов. Для более детального исследования работы алгоритма в реальных условиях рассматривался типичный вагон, представленный на рис. 5 и имеющий размеры $15724 \times 2764 \times 3050$ мм, его вес 26 т, а объем 138 м^3 . Исходные данные для задачи приведены в табл. 1, а загруженность вагонов – в табл. 2.

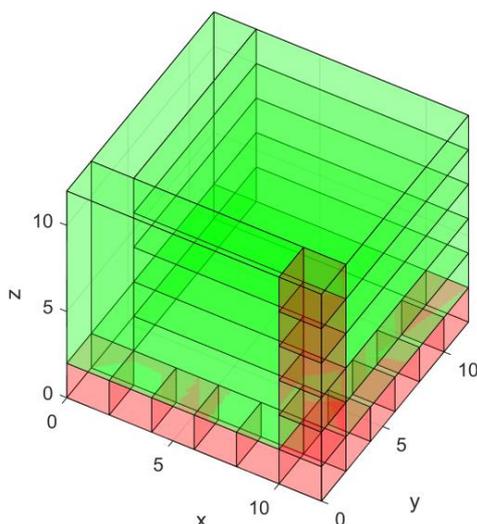


Рис. 4. Схема размещения объектов для модельной задачи

Fig. 4. Layout of objects for the model problem



Рис. 5. Крытый типовой вагон

Fig. 5. Covered typical wagon

Таблица 1.
Характеристики контейнеров и объектов

Table 1.
Containers and objects characteristics

	Длина	Ширина	Высота	Количество
Контейнер	15724	2764	3050	-
Объект 1	1500	1800	1100	81
Объект 2	1400	1200	1100	97
Объект 3	1200	1400	1400	32
Объект 4	1600	1500	1700	70
Объект 5	1300	1900	1100	88

Таблица 2.
Заполняемость контейнеров

Table 2.
Occupancy of containers

№ контейнера	1	2	3	4	5	6	7	8	9
Заполняемость, %	94,04	94,04	94,04	93,21	82,54	90,85	90,85	90,85	38,91

Наиболее неудобный предмет соответствует желтому цвету, поэтому он является приоритетным при упаковке, в то время как другие предметы упаковываются по остаточному принципу. Тем не менее, за счет различия в размерах такой подход позволяет поучить высокую степень заполняемости пространства контейнера. На рис. 6 показана схема упаковки, идентичная для первых трех вагонов. Как видно из табл. 2, они упакованы наиболее плотно.

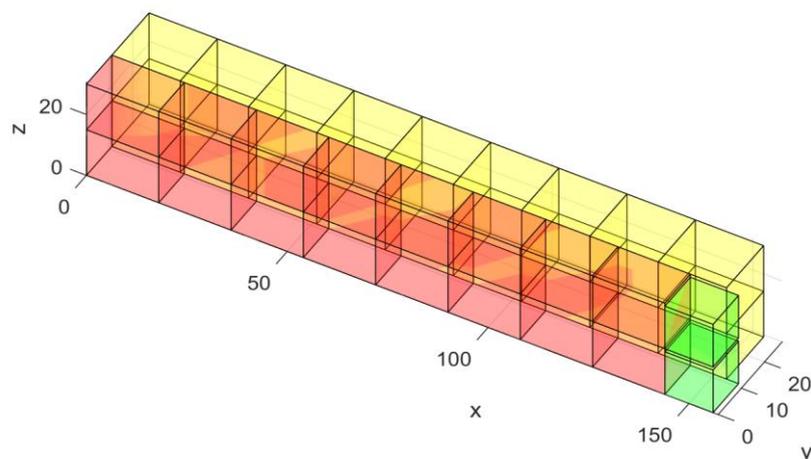


Рис. 6. Размещение груза первого вагона

Fig. 6. Cargo placement of the first wagon

Падение эффективности упаковки за счет исчерпания предметов одного типа показана на рис. 7. В процессе размещения предметов происходит исчерпания предметов, обозначенных красным цветом, вследствие чего это место занимает менее эффективно предметами, соответствующими синему цвету. Причиной этого является оценка подконтейнеров алгоритмом динамического программирования без учета реального числа оставшихся предметов.

Нивелирование этого недостатка метода возможно только путем повышения размерности алгоритма динамического программирования.

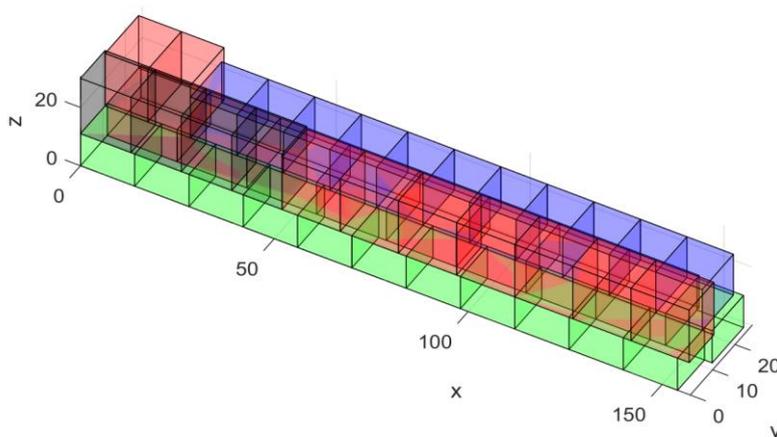


Рис. 7. Размещение груза в пятом вагоне при исчерпании одного из типов объектов

Fig. 7. Cargo placement in the fifth car when one of the types of objects is exhausted

Особенностью задач по упаковке грузов во множество контейнеров является большая вероятность низкой заполняемости последнего из них. С практической точки зрения незанятое место в последнем контейнере является негативным фактором, тем не менее, при теоретических исследованиях отсутствие груза для более полной загрузки последнего контейнера не является недостатком алгоритма упаковки, поэтому предлагается пренебречь незанятым пространством последнего контейнера. Это повышает чувствительность целевой функции даже к несущественным изменениям качества алгоритма.

Выводы

Использование разработанного алгоритма обеспечивает более плотную загрузку контейнеров в случае, когда существует небольшое количество типов-размеров объектов, а число самих объектов велико. Классические эвристики для решения задачи упаковки контейнера эффективны, как правило, для случая различных размеров объектов, так как не предназначены для просмотра вариантов упаковки объектов по всему доступному объему.

Предлагаемый метод, основанный на динамическом программировании с мемоизацией и вычислением коэффициента неудобства объектов, позволяет получить эффективность упаковки, минимизируя количество использованных вагонов.

Библиографический список

1. **Курейчик, В.В.** Применение генетического алгоритма решения задачи трехмерной упаковки / В.В. Курейчик, Д.В. Заруба, Д.Ю. Запорожец // Известия Южного федерального университета. Технические науки. 2012. Т. 132. № 7. С. 8-14.
2. **Курейчик, В.В.** Гибридный подход для решения задачи 3 -х мерной упаковки / В.В. Курейчик, А.Е. Глушенко, А.Н. Орлов // Известия Южного федерального университета. Технические науки. 2016. №. 6 (179). С. 45-53.
3. **Гладков, Л.А.** Решение задачи трехмерной упаковки разногабаритных объектов с использованием бионических методов / Л.А. Гладков, Н.В. Гладкова, Е.С. Скубриева // Известия Южного федерального университета. Технические науки. 2013. Т. 144, №. 7. С. 35–41.

4. **Псиола, В.В.** О приближенном решении трехмерной задачи об упаковке на основе эвристик // Интеллектуальные системы. 2007. Т. 11. № 1. С. 83-100.
5. **Ху, Т.Ч.** Комбинаторные алгоритмы / Т.Ч. Ху, М.Т. Шинг. – Нижний Новгород: Изд-во Нижегородского госуниверситета им. Н.И. Лобачевского, 2004. – 330 с.
6. **Юдаков, П.В.** Задача о трехмерной упаковке и методы ее решения. Обзор // Инженерный вестник. 2015. № 6. С. 12.
7. **Ягудин, Р.Р.** Решение задачи оптимизации упаковки многогранников в параллелепипедную область на основе построения годографа вектор – функции плотного размещения // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика, телекоммуникации и управление. 2012. № 5 (157). С. 58-62.
8. **Акеб, Н. А** Look-Forward Heuristic for Packing Spheres into a Three-Dimensional Bin // 2014 Federated Conference on Computer Science and Information Systems. 2014. P. 397-404.
9. **Чеканин А.В.** Алгоритмы эффективного решения задачи ортогональной упаковки объектов / А.В. Чеканин, В.А. Чеканин // Журнал вычислительной математики и математической физики. – 2013. Т. 53. №. 10. С. 1639-1648.
10. **Li X.** A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins / X. Li, Z. Zhao, K.A. Zhang // IIE Annual Conference. Proceedings. – Institute of Industrial and Systems Engineers (IIE), 2014. С. 2039.
11. **Dube, E.** Optimizing three-dimensional bin packing through simulation / E. Dube, L.R. Kanavathy, P. Woodview // Sixth IASTED International Conference Modelling, Simulation, and Optimization. – 2006.

*Дата поступления
в редакцию: 10.05.2023*

*Дата принятия
к публикации: 11.07.2023*