электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

Задача о трехмерной упаковке и методы ее решения. Обзор

06, июнь 2015

Юдаков П. В.[1,*]

УДК: 658.7

[1]Россия, МГТУ им. Н.Э. Баумана
[*]judakoff@mail.ru

Введение

Задача трехмерной упаковки представляет из себя естественную эволюцию классической одномерной и двухмерной задачи. Наиболее часто используемое практическое применение данной задачи – транспортировка груза, который будет упакован в контейнеры или кузовы транспортных средств или упаковка груза на паллеты. Кроме редких исключений большинство статей, посвященных этой задаче, рассматривают размещение товара в трехмерные ящики. Но, даже с учетом этого ограничения, масштаб и разброс приложений очень велик. Несмотря на промышленную и экономическую важность таких приложений, публикаций на эту тему было издано заметно меньше, чем по одномерной или двухмерной задаче. Однако в последние годы заметен быстрый рост в исследованиях этой проблемы, включая новые направления поиска решений вкупе с маршрутизацией и планированием.

Упаковка коробок в контейнер, кузов транспортного средства или на паллету является одной из самых сложных проблем упаковки из-за ограничений, которые возникают в реальном мире. Например, при попытке организовать оптимальную упаковку с наиболее эффективным использованием имеющихся ресурсов недооценка количества необходимых контейнеров для транспортировки может обойтись очень дорого из-за задержки и перевозки дополнительных контейнеров, которые недостаточно полно используются, превращаясь в потраченный впустую ресурс. Касательно размеров коробок, существует множество ограничений, связанных с распределением веса, укладкой, устойчивостью упаковки. Более того, клиенты могут ограничивать грузы, которые необходимо перевозить вместе или, например, разгрузка транспорта может происходить в нескольких местах. В работе [1] представлен комплексный обзор таких ограничений.

Цель данной работы заключается в обзоре статей, посвященных трехмерной загрузке контейнера, причем большее внимание будет уделяться практическому применению методологий решения этой проблемы, представленных у различных авторов со всего мира за последние годы.

1 Классификация задач

Как было сказано раньше, мы будем рассматривать в качестве грузов лишь трехмерные объекты формы параллелепипеда. Груз будем называть коробками, а пространство, в которое он упаковывается – контейнером. Различаем два типа набора коробок:

- однородные – когда в наборе мало различных типов коробок, а количество коробок велико;
- разнородные – когда в наборе много различных типов коробок, а количество коробок мало.

Если контейнеров несколько, то они могут быть идентичными, однородными или разнородными.

Для начала рассмотрим задачи, целью которых будет упаковать все коробки в минимально возможное число контейнеров. Таким образом, комбинируя различные типы коробок и контейнеров, выделим шесть основных типов проблем упаковки:

1) SSSCSP (Single stock-size cutting stock problem) – если контейнеры идентичны, а коробки однородны;
2) SBSBPP (Single bin-size bin packing problem) – если контейнеры идентичны, а коробки разнородны;
3) MSSCSP(Multiple stock-size cutting stock problem) – если контейнеры и коробки однородны;
4) MBSBPP (Multiple bin-size bin packing problem) если контейнеры однородны, а коробки разнородны;
5) RCSP (Residual cutting stock problem) – если контейнеры разнородны, а коробки однородны;
6) RBPP (Residual bin packing problem) – если контейнеры и коробки разнородны.

Когда необходимо заполнить всеми коробками максимальный по объему набор из представленных контейнеров, различают семь различных типов задач:

1) IIPP (dentical item packing problem) – если единственный контейнер и коробки идентичны;
2) SLOPP (Single large object placement problem) – если единственный контейнер и однородные коробки;
3) SKP (Single knapsack problem) – если единственный контейнер и коробки различны;
4) MILOPP (Multiple identical large object placement problem) – если несколько идентичных контейнеров и однородные коробки;
5) MHLOPP (Multiple heterogeneous large object placement problem) – если контейнеры однородны или разнородны, а коробки однородны;
6) MIKP (Multiple identical knapsack problem) – если несколько идентичных контейнеров, а коробки разнородны;

7) MHKP (Multiple heterogeneous knapsack problem) – если контейнеры однородны или разнородны, а коробки разнородны.

Во всех перечисленных задачах подразумевается, что размеры контейнеров фиксированы. Однако если у контейнера два размера фиксированы, а длина или высота варьируется, то это задача SCIMP (single container input minimization problem), но мы в рамках данной работы ее рассматривать не будем.

2 Ограничения задачи

Во всех рассматриваемых задачах присутствуют стандартные, общие ограничения:

- коробки могут размещаться только таким образом, чтобы их грани были параллельны стенам контейнера;
- все коробки должны быть упакованы внутрь контейнера;
- коробки не должны пересекаться друг с другом.

В одних задачах коробки должны плотно прилегать друг к другу, в других допустим небольшой свес. Если необходимо учитывать центр тяжести объектов, то предполагается, что он совпадает с геометрическим центром объекта.

Помимо перечисленных основных ограничений в зависимости от конкретной задачи возникают различные дополнительные ограничения, связанные с ориентированностью коробок, стабильностью, плотностью, распределением веса, максимально возможным весом коробок, помещаемых в контейнер, приоритетом коробок, их несущей способностью и необходимостью выгрузки в нескольких пунктах. Более подробно все эти ограничения рассмотрены и классифицированы в работе [1].

Всего возможно 6 различных ориентаций коробки в пространстве относительно осей координат, что используется в работах [2-4]. Зачастую в задачах есть запреты на определенные ориентации или на поворот относительно определенных осей. Такие задачи рассмотрены в работах [5, 6]. В некоторых статьях [7, 8] любой поворот коробок вообще запрещен.

Стабильность упаковки является важным условием успешно решенной задачи, особенно остро проявляющимся на практике. Неустойчивая упаковка может привести к повреждению груза и усложнениям при погрузке или выгрузке. Использование крепежных ремней или заполнение пустого пространства между коробками специальным объемным материалом применяется на практике, однако это повышает стоимость упаковки, перевозки и вообще нежелательно. Самый верный способ достижения стабильности упаковки заключается в таком заполнении контейнера, когда каждая коробка целиком находится на плоской поверхности [2]. Другие авторы предлагают ослабить это ограничение и разрешают частичные свесы с максимально допустимым значением [9, 10]. В некоторых работах используют подход, согласно которому хотя бы одна сторона каждой упакованной коробки должна соприкасаться с другими коробками. Два различных подхода для измерения стабильности груза были предложены в работе [11].

Ограничения укладки подразумевают то, как коробки будут лежать друг на друге с учетом их несущей способности. Каждая коробка имеет максимальный вес, который она может выдержать, зависящий от ориентации коробки. В работе [11] аргументировано, что несущая способность коробки обеспечивается, в первую очередь, за счет ее боковых стенок. Поэтому в некоторых ситуациях возможно размещение идентичных коробок наверх уже упакованных, однако могут возникнуть проблемы при размещении коробки в половину меньшего размера и веса посередине сверху. Ограничения по высоте, на которую может быть установлена тяжелая коробка также связаны с соображениями стабильности.

Вес грузов в отдельности и распределение веса по контейнеру являются важнейшими ограничениями при погрузке или перемещении уже упакованного контейнера. Идеальное распределение веса достигается в том случае, когда центр тяжести контейнера совпадает (или близок) с его геометрическим центром на полу. При загрузке наземного транспорта необходимо учитывать весовые нагрузки на оси транспортного средства и максимально допустимый суммарный вес контейнера.

Ситуация множественной выгрузки контейнера в разных пунктах назначения подробно рассмотрена в работах [11-13]. Коробки с разными грузами пакуются отдельно определенным образом, чтобы максимально эффективно произвести выгрузку в такой ситуации. Поэтому необходимо продумывать такую упаковку коробок, чтобы на всем протяжении пути во время выгрузок/перегрузок избегать необходимости перемещения большого количества коробок, в идеале лишь выгружая только необходимый в этом пункте назначения груз.

Все рассмотренные выше ограничения, непосредственно влияющие на упаковку контейнера, должны быть объединены с эвристикой для решения поставленной задачи наиболее эффективно.

В работе [14] автор вводит еще два ограничения, которые влияют на упаковку ящиков в разные контейнеры:

- разделение грузов – когда необходимо перевезти за один раз, например, продукты питания и парфюмерию, поэтому их приходится разделять по разным контейнерам;
- полнота упаковки – когда в упаковку должны быть включены еще и грузы, принадлежащие к определенной подгруппе.

3 Эвристики расположения

В любом приближенном решении поставленной задачи должен присутствовать механизм, который бы позволил определить, как именно располагать коробки в контейнерах, необходимо ли заново генерировать начальное приближение, или текущая установка все же входит в приближенное оптимальное решение. Такой механизм в этой работе называется эвристикой расположения. Для загрузки контейнера существует множество подобных эвристик. Если набор коробок однороден, то чаще всего используют «постройку стены» или «постройку слоя». В случае разнородных коробок с грузом целесообразно устанавливать коробки по одной.

В схемах с постройкой стены или слоя коробки одного типа собираются в строки или столбцы, чтобы заполнить одну вертикальную или горизонтальную стенку свободного пространства. Список свободных мест создается для всех возможных расположений. Как только свободное пространство включает в себя стену или слой, генерируются новые свободные места. Как правило, если получилась стена или слой, оставшееся свободное место рассматривается, как более маленький контейнер. Оба приближения имитируют ручную упаковку, пытаясь создать плоские грани результирующей упаковки.

Наряду с решением о том, как упаковывать коробки, приходит необходимость определить, какой тип коробки будет упакован следующим. Общие подходы основаны на определении очереди заранее и на динамическом определении очереди. В первом случае очередь создается, применяя определенные критерии, такие как объем коробки, число коробок, упаковочная площадь или конкретные размеры коробки. Динамическая очередь, как правило, базируется на пустом пространстве, допустимом для упаковки новых коробок после установки очередной коробки, объеме еще неупакованных коробок одного типа и свободной упаковочной площади.

Рассмотрим перечисленные эвристики подробней.

3.1 Построение стены

Исходный вариант такой эвристики заполняет контейнер набором башен с коробками (стенами) по всей глубине контейнера. Башни строятся последовательно, поэтому проблемы касаемо распределения веса решаются обменом, чередованием или зеркальным отображением упаковки, чтобы при этом коробки не пересекались друг с другом.

Самое раннее упоминание об этой эвристике было представлено в работе [15] в 1980 г. Авторы предложили двузонную эвристику, основанную на настоящих промышленных задачах из 800 коробок и не более чем 20 типов грузов. Заполняя очередное пустое пространство, а вначале – весь контейнер, для каждой стены (башни) выбирается первая (нижняя) коробка таким образом, чтобы стена была не очень глубокой и не очень мелкой, используя следующую систему рангов. Для начала выбирается тип коробок, у которого минимальный размер является максимальным среди всех других кандидатов. Однако под этот критерий могут подойти сразу несколько типов. Тогда выбирается набор, состоящий из максимального числа коробок. Если по-прежнему нет однозначности, то выбираются коробки, у которых наибольший размер является длиной. После того, как все коробки какого-нибудь одного типа упакованы, этот тип становится открытым. Открытые типы дают преимущество, потому что тип с наибольшим числом коробок получает самый высокий ранг. Если длина незаполненного контейнера слишком мала, обычно меньше какого-то определенного значения, оставшиеся коробки более не упаковываются по правилам этой эвристики.

Основываясь на этой схеме, в работе [16] приведено сравнение 14 различных эвристик, объединяющих шесть правил рангов и три метода заполнения контейнера, с различными ранговыми критериями. Не выбрав одну лучшую эвристику, авторы предложили гибридную, объединяющую все 14 эвристик для выбора лучшего решения.

В работе [17] также используется базовая схема из статьи [15], но с двумя модификациями. Первая модификация относится к получению новых пустых мест для упаковки. В то время, как в основополагающей работе предполагается, что ширина исходного пространства является шириной новых создаваемых столбцов незанятого пространства, эти авторы ограничивают пространство в области над упакованными коробками, используя решения стабильности первоначального подхода. Вторая модификация ограничивает ширину новой стены, чтобы она была не больше, чем у предыдущей стены, в результате чего происходит слияние незанятого пространства в более крупные и более удобные пространства и увеличивается стабильность упаковки.

Вместо использования фиксированных критериев сортировки, в работах [18, 19] определяется последовательность коробок для упаковки благодаря алгоритму поиска на дереве и использованию базовых принципов, описанных в статье [15]. В модификации базовых принципов по версии, представленной в работе [19], алгоритм поиска на дереве разработан для поиска набора глубин стен и ширин полосок, которые могут быть повернуты вертикально или горизонтально, чтобы составить стену. Первые значения различных глубин получаются путем выбора в каждом узле решения ширины полосы. Лишь фиксированное число подузлов рассчитывается для каждого узла решения, также допускается бэктрэкинг. Различные глубины стен получаются благодаря использованию 27 ранговых критериев в каждом узле решения.

В работе [11] рассматривается загрузка контейнера грузом, который необходимо доставить в несколько различных мест. Авторы используют подход к упаковке, в котором каждая партия по очереди формируется так, что сначала упаковывается груз, который необходимо доставить последним. Во время упаковки каждая следующая коробка и ее расположение выбирается так, чтобы максимизировать оставшееся незанятое пространство. Так как при этом нет однозначности, авторы выбирают коробку, чтобы в оставшемся пространстве выступ от установленной коробки по длине был минимальным. Есть еще два правила для исключения множественности, в которых выбирается коробка с наибольшим объемом, а затем выбирается пространство с минимальной шириной. Здесь базовый принцип по очереди добавляет отдельные башни или ширину контейнера вместо стен целиком.

В работе [20] коробки упаковываются в единственный контейнер с известными размерами с использованием базового принципа из работы [15]. Длина (толщина) каждой стены определяется длиной «слоя определения коробки» (layer determining box – LDB), который является первой, обычно самой большой коробкой, размещенной в этой стене. Два непересекающихся пространства генерируются, как только LDB был установлен в контейнере. Рядом с LDB образуется пространство с такой же высотой. Над LDB также образуется пространство сквозь всю ширину контейнера. Каждую коробку лучше поставить в стену, нежели между соседними стенами. В работе [21] развивается концепция LDB таким образом, чтобы генерировались боки, содержащие конкретное число стен вместо одиночных вертикальных стен. Коробки могут соединять смежные стены, но не два отдельных блока. Ранговый критерий выбора типа коробок и их ориентации основан на базовой работе [11]. Когда невозможно расположить очередной LDB, текущая стена считается последней и ее глубина расширяется до конца контейнера вместо длины LDB. В работе [22] также используется эвристика построения стены как части гибридного генетического алгоритма.

1) В работах [23, 24] применяются кубовидные модели глубиной коробки, которые более эффективны, чем стены. Во-первых, куб состоит из множества коробок одного типа, и могут быть присоединены по высоте или ширине. Во-вторых, два таких кубоида располагаются один за одним или один над другим. Все свободные места сохраняются в списке, а пространство с минимальным объемом всегда заполняется первым. Также используются два альтернативных правила для получения локального расположения внутри упаковочного пространства:
2) общий объем устанавливаемых коробок должен быть максимальным;
3) использование двойного критерия – минимально возможного объема потерь и максимально возможного эффективного объема.

В работе [6] описана система упаковки контейнера для определения и визуализации вручную или в автоматическом режиме пошагового заполнения контейнера, включающее в себя ориентацию каждой устанавливаемой коробки и поиск ее определенного места в контейнере. Коробки упаковываются в вертикальные полосы, которые затем объединяются в стены. Метод пространственного представления матрицы применяется при поиске и слиянии незаполненного пространства.

3.2 Построение слоя

Эвристика построения слоя является одной из базовых эвристик расположения, однако некоторые исследователи усовершенствовали этот подход. Упаковка производится следующим образом: коробки по очереди устанавливаются на пол контейнера, создавая базовый слой. Как только этот слой заполнится, новый слой образуется поверх базового и создание новых слоев происходит до тех пор, пока в оставшуюся незанятую высоту контейнера уже не поместится ни один слой. Визуализация данного принципа представлена на рис. 1. Важно отметить, что некоторые авторы в своих работах под слоями имеют ввиду стены, рассмотренные выше.

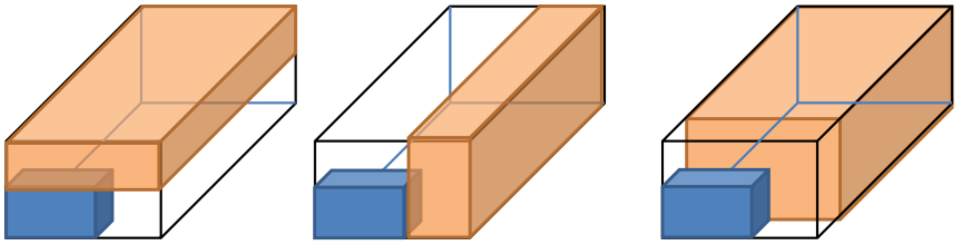

Рисунок 1. Принцип построения слоя

Работа [11] появилась благодаря задаче упаковки паллеты. Каждый слой содержит не более двух различных типов коробок, причем тип коробки и ее ориентация выбираются исходя из загруженности площади погрузки, на которую устанавливается слой. Такой подход решает проблемы стабильности, возникающие в результате значительных перепадов высот в стенах из коробок, которые проявляются в других подходах. В работе [25] авторы также придерживаются этого подхода как части итеративной эвристики, позволяющей решать проблемы упаковки «неудобных» типов коробок. Вводится определенный коэффициент, пересчитывающийся после каждой итерации для оценки «неудобности» полученной упаковки. В результате работы такой схемы «неудобные» коробки будут упакованы раньше.

Работа [26] основана на предыдущей статье. В ней предложен алгоритм, успешно работающий с ограничениями, связанными с несущей способностью коробок. Авторы предлагают итеративно упаковывать слои коробок в контейнер снизу (с уровня пола) вверх. Каждый слой состоит не более чем из двух прямоугольных блоков. Коробки одного блока должны быть одного типа и одной ориентации. На каждом шаге алгоритм проверяет разницу между весом выбранного блока и максимально допустимым весом на упаковочной поверхности. Коробки со слабой несущей способностью не могут быть упакованы на первоначальных итерациях. Следовательно, возможности для размещения коробок на более высоких слоях рассматриваются на каждом шаге алгоритма.

В работе [27] предложен похожий подход. Однородные коробки группируются по высоте, а затем группы сортируются по убыванию их высоты. Внутри каждой такой группы коробки сортируются по убыванию площади основания. Коробки упаковываются с конца контейнера по очереди, начиная с тех, которые занимают большее пространство. В работе [28] автор, развивая эту идею, придерживается простой стратегии сортировки – коробки принадлежат одной группе, если они находятся внутри определенного допуска высот. Также, как и в статье [17], здесь коробки сортируются по убыванию площади основания внутри своих групп. Двухмерная эвристика упаковки создает слой путем оценки и вычисления позиций кандидатов. После генерации слоев, которые вмещают в себя все коробки, решается одномерная проблема упаковки для установки слоев со своими высотами в ящик конечной высоты. Эта эвристика называется first-area second (HA) и комбинируется вместе с табуированным поиском.

3.3 Другие эвристики расположения

Пусть даны однородные блоки, состоящие из коробок определенного типа. Упаковка контейнера такими блоками дает несколько преимуществ.

1) Такую схему проще организовать и на такую загрузку требуется меньше времени.
2) Нет необходимости в пересортировке груза после разгрузки, так как коробки одного типа упакованы вплотную друг к другу.
3) Проблема несущей способности коробок здесь менее сложна из-за того, что укладываются идентичные коробки.

4) Блочная структура обеспечивает дополнительную стабильность упаковке благодаря тому, что одинаковые коробки, установленные в прямоугольную форму, не могут с легкостью проскальзывать.

В работе [29] разработан гибридный подход, в котором табуированный поиск и эвристика построения блоков запускаются многократно. Типы груза сортируются по убыванию объема для получения начального решения. Блоки составляются из коробок одного типа таким образом, чтобы наиболее эффективно заполнить доступное для упаковки место. В некоторых ситуациях в блок добавляется еще один тип коробок для улучшения заполнения. В работе [30] также используется стратегия блочного размещения для определения приоритета загрузки в рамках одного контейнера. Определены пятиблочные оценочные функции и используется поиск на дереве лучшего блочного расположения для каждой такой функции. Автор статьи [3] тоже использует структуру типа дерева для представления контейнера ветвей взаимоисключающих подпространств, оставшихся после упаковки блока идентичных коробок. В работе [31] предложена эвристика построения блоков, в которой несколько различных типов коробок могут составлять блок. А для выбора блока установки используется алгоритм многослойного поиска.

В работе [32] предложена стратегия «максимального пространства». Интервалы «максимального пространства» – набор самых больших кубических пространств, которые полностью покрывают свободный объем контейнера. Эти пространства могут наслаиваться друг на друга. После установки очередной коробки пространства генерируются и сохраняются в списке. Все интервалы, недостаточно большого размера для вмещения в себя коробки или включенные в другие пространства, удаляются из списка. Следующая коробка упаковывается в ближайшее пространство к левому верхнему углу контейнера достаточного размера для помещения в него рассматриваемой коробки. Разработано множество эффективных алгоритмов, использующих такую стратегию. Например, в работах [9, 10] используется эвристика, которая размещает башни, ряды, стены или слои коробок одного типа. Ее идея заключается в определении максимально свободных пространств и заполнении их наиболее удачной конфигурацией идентичных коробок на основе одного или двух критериев. На каждой итерации грузом заполняется максимальное пространство, находящееся ближе всего к углу контейнера. Эта эвристика объединяется с жадным случайным адаптивным поиском (GRASP) в работе [10] и локальным поиском с чередующимися окрестностями (VNS) в работе [9]. Вот другие недавние работы, показывающие хорошие результаты на основе такой же идеи: [33-36].

В работе [2] коробки упаковываются в башни или стеки при работе с разнородными коробками. Все башни содержат базовую коробку, установленную непосредственно на пол контейнера. Каждая из оставшихся остальных коробок устанавливается на коробки таким образом, чтобы их нижняя грань полностью стояла на уже упакованной коробке. Жадный алгоритм используется для минимизации неиспользованного пространства над базовыми коробками. На втором шаге алгоритма решается задача двумерной упаковки всех базовых коробок на пол контейнера.

Автор работы [37] решает задачу упаковки одного контейнера используя и коробочный, и блочный подход. Предложенная жадная эвристика сортирует коробки по убыванию их объемов. Каждая комбинация ориентаций коробки и пустых пространств рассматривается с использованием определенного критерия для выбора наиболее соответствующего пустого пространства для размещения коробки. Основной принцип заключается в том, что после упаковки очередной коробки суммарный объем пустого пространства, в которое не может быть установлена ни одна из неупакованных коробок, получается минимальным. Из-за неоднозначности такого критерия, предпочтение при выборе отдается пространству, наиболее близко находящемуся к нижнему заднему левому углу контейнера. Затем решения дополняются и улучшаются с учетом различных последовательностей загрузки на дереве решений, где ветви образуются различными типами коробок и ориентациями коробок.

В работе [38] коробки сортируются на основании двойственного критерия – отношения объема к площади основания. Авторы предлагают новый подход под названием MBP (multifaced buildup algorithm), в котором коробки могут быть установлены на любую стену контейнера и каждая стена может быть использована в качестве базы или пола. Как только коробки были установлены на стену контейнера в качестве базы, остальные коробки могут быть установлены на верх этих коробок.

В работе [39] автор избегает зависимости упаковки от очереди коробок путем получения возможного расположения для каждой коробки. Для увеличения эффективности этого подхода используется метод пространственного представления. Пространственная матрица, состоящая из двумерных матриц, представляет позиции и размеры всех упакованных коробок и пустых пространств. В работах [40, 41] также используется прием пространственного представления для моделирования процесса упаковки коробок и отслеживания как упакованного, так и свободного пространства в контейнере. По сравнению с базовой работой [39], этот алгоритм позволяет пользователю предопределять позиции некоторых коробок. Пространственные матрицы также используются в работе [42], в которой разработана новая эвристика специально для учета несущей нагрузки. Такой подход гарантирует, что все коробки установлены на пол контейнера или их основания находятся в непосредственном контакте с другими коробками. Эта эвристика упаковывает одну коробку на каждой итерации с использованием системы оценки, состоящей из пяти критериев для выбора варианта размещения. Доступные поверхности для упаковки хранятся и обновляются в двух матрицах – в первой помещается высота поверхности, а во вторую заносится соответствующая несущая способность.

Все рассмотренные выше эвристики моделируют доступное для упаковки свободное пространство, однако есть и другие подходы. В работах [43, 44] используется альтернативная стратегия определения точек расположения (но в ранней работе они назывались угловыми точками, а в более поздней – экстремальными). Эти точки являются кандидатами для размещения в них неупакованных коробок. Рассмотрим упаковку как трехмерную поверхность (свесы не разрешены), тогда интересующие нас точки появляются в местах контакта между двумя коробками или коробки и контейнера и расположены в точке совпадения трех ребер, образуя «вогнутый» угол. В работе [43] автор для начала распределяет коробки по контейнерам перед тем как определить позицию каждой коробки. Каждый контейнер образуется из дерева решений, где каждый узел является частным решением и для каждого доступного угла следующей коробки существует набор дочерних узлов, в которых коробки упакованы по убыванию своего объема. А в работе [44] происходит одновременное распределение коробок по контейнерам вместе с упаковкой, используя для этого хорошо известную эвристику (best-fit decreasing heuristic).

Похожий принцип упаковки предложен в работах [45, 46] для упаковки коробок в единственный контейнер. Автор ввел термин «уровень пещерности» для ситуаций, когда коробка находится в углу или в «пещере» – пустом пространстве, окруженном коробками, если это возможно. «Уровень пещерности» позволяет хранить все доступные углы в списке, а затем получает все комбинации углов, типов коробок и допустимых ориентаций для выбора лучшего набора, основываясь на системе рангов. В работе [47] представлена модификация базового подхода для уменьшения пространства поиска. А в работе [48] авторы придумали улучшение уже для модифицированного подхода, добавив в алгоритм фазу локального поиска.

В работе [49] используется шаблон упаковки L-формы, получающийся при упаковке коробок вдоль основания и одной вертикали. В работе [50] предложена эвристика для решения этой же проблемы, но с одинаковыми коробками. Эвристика на каждой своей итерации создает слой. Разрешено переориентирование контейнера, а слои могут быть построены от любой стены или даже от пола, а не только от стены контейнера, как это происходит в большинстве известных подходов.

В работе [51] предложено представление упаковки в виде направленного графа. Требуются три направленных графа для отображения ребер x, y, z контейнера. Авторы утверждают, что вырождение и симметрия приводят к достаточному снижению размера задачи и успешному решению. Представлены результаты для 6 возможных ориентаций каждой из 394 коробок.

4 Последовательность упаковки

Упорядочивание коробок перед упаковкой сильно влияет на эффективность той или иной эвристики упаковки. Модификации эвристик зачастую включают в себя изменения касательно построения очереди коробок. Рассмотрим наиболее часто используемые подходы, описанные в литературе. Разобьем все упорядочивания на две группы:

1) статические;
2) динамические.

Статическое упорядочивание подразумевает использование фиксированного ранжирования коробок или типов коробок перед непосредственно упаковкой. В динамическом упорядочивании используются критерии, определяющие следующую коробку или место для упаковки во время самого процесса упаковки. Очевидно, что существует много различных способов создания последовательности коробок для упаковки в контейнер при соблюдении правил статического и динамического упорядочивания. Хоть до сих пор не придумано наилучшего подхода, удовлетворяющего всем задачам, наиболее часто используется сортировка по объему.

5 Улучшение эвристик

В то время как использование эвристик позволяет получить быстрое и в основном удовлетворяющее качеству решение, расширение пространства поиска с помощью улучшенных эвристик обычно приводит к значительному преимуществу. В общем, улучшение сводится к работе с одним или несколькими полными решениями, делаются соседние шаги для поиска лучшего решения. Этот процесс варьируется в сложности от простых структур и критериев отбора, которые лишь улучшают шаги поиска решения до сложных областей и критериев, позволяющих находить множество локальных оптимумов, включая многие реализации стандартных метаэвристик.

Большинство эвристик расположения основаны на порядке сортировки или перестановки коробок. Для решения этой проблемы часто используют генетический алгоритм (GA), так как целое множество генетических операторов, представленных в литературе, работают с сортировкой и перестановкой.

Одно из самых ранних использований GA было представлено в работе [53] в 1994 году, здесь алгоритм использовался для определения ширины стен. Каждая хромосома состоит из набора целых чисел. Каждое число отображает стратегию упаковки, используемую в соответствующей стене, и все эти стены объединяются в целый контейнер. Родительские хромосомы выбираются случайным образом, а шанс выбора каждой хромосомы напрямую связан с ее значением фитнес-функции. Скрещивание или мутация разрешены родительским хромосомам для генерации двух дочерних хромосом. Лучший набор стен будет достигнут в результате процесса получения новых поколений.

GA в работе [2] решает двухмерную подзадачу. Как было описано ранее, коробки складываются в башни. Башни образуются, исходя из двумерной задачи. Хромосома отражает решение с помощью вектора расположения, который показывает последовательность расположений оснований башен и две возможных ориентации каждого основания башни. Каждая хромосома трансформируется в решение путем определения списка доступных для размещения углов с последующей установкой коробки в первом возможном углу. Популяция ранжируется исходя из значения фитнесс-функции, чем выше ранг, тем раньше коробка будет установлена. Поскольку хромосомы представляют из себя расстановки оснований башен, используются операторы перестановки для генерации потомков в качестве возможных решений. Скрещивание и мутация чередуются случайным образом. Похожее применение GA предложено в работе [22]. Однако, здесь GA представляет слои (стены) из коробок, построенные в результате применения базовой эвристики. Она используется для генерации начального решения, а новые слои образуются из потомков после скрещивания или мутации. Хромосома, содержащая лучший план упаковки, подвергается пересортировке для достижения наилучшего распределения веса. В работе [54] предприняты попытки по улучшению качества решения с применением распараллеливания GA. Предложенная модель разворачивается в локальной компьютерной сети, и каждый отдельный экземпляр GA-задачи назначается определенной рабочей станции.

В работе [55] предложено два различных подхода для загрузки коробок в единственный контейнер. Каждая частица хромосомы состоит из двух частей – самой коробки и ее ориентации, пронумерованной от 1 до 6. Первоначально все хромосомы заполняются номерами коробок в последовательности уменьшения их объемов со случайно выбранными ориентациями. Используется метод селекции по принципу колеса рулетки, один оператор скрещивания с механизмом восстановления и мутация, случайно меняющая коробки или ориентацию. Лучшее решение копируется на следующую популяцию.

В работах [33, 56] также используется структура перестановки, но нет необходимости в корректировке потомков хромосомы благодаря стратегии случайного ключа. Эта стратегия заключается в перестановке коробок и случайному присвоению вещественного числа каждой коробке. Сортировка коробок по случайному числу рождает новые перестановки. Авторы в два раза увеличили длину хромосомы, чтобы включить в нее эвристику расположения для каждой коробки.

В работе [57] предложена альтернативная популяционная эвристика, очень похожая на GA, но построена на наблюдениях за поведениями пчел. Алгоритм пчелиного поиска работает с популяцией решений и вычисляет их фитнесс-функцию, затем выбираются перспективные решения и подвергаются фазе локального поиска. Лучшие решения и некоторые из вновь созданных переходят на следующую популяцию. Слои получаются согласно базовой эвристике, цель алгоритма поиска заключается в том, чтобы найти альтернативные ширины слоев при лучшем общем качестве упаковке, при необходимости вращая коробки.

Табуированный поиск (TS) является одним из наиболее популярных методов решения задач комбинаторной оптимизации, что не могло не отразиться на частоте применения его в трехмерной упаковке. В работах [23, 24] применяется TS вкупе с эвристикой построения стены для упаковки однородных коробок в единственный контейнер. Полное решение вначале генерируется согласно базовой эвристике. TS оперирует с очередью упаковки и допустимыми свободными пространствами. Список табу содержит в себе полные очереди упаковки в выбранной окрестности. Распараллеливание TS было предложено позже в работе [24] для улучшения качества решения. Основной подход описан в статье [58], где используется алгоритм имитации отжига (SA) как альтернативы TS при использовании базовой эвристики. Позже SA был объединен с TS и сформирован новый гибридный алгоритм. Это было сделано для того, чтобы усилить сильные стороны обоих алгоритмов и избежать их недостатков. Таким образом, SA используется для получения хорошего начального решения, с которыми начинает работать TS в течении одной или более итераций. Также проводились работы по распараллеливанию SA и гибридного алгоритма.

В работе [29] используется поиск среди коробок, отнесенных к пространствам. Вместо изменения свободного пространства, принадлежащего коробке, алгоритм меняет очередность упаковки, меняя коробки местами в пределах определенной окрестности. Длина списка табу зависит от размерности задачи.

Наиболее очевидный способ объединения TS и эвристики расположения для решения задачи со многими контейнерами заключается в использовании TS для присваивания ящиков контейнерам с последующим применением эвристики расположения для определения слоев упаковки. Этот вариант и предложен в работе [59]. В работе [62] рассмотрен двухуровневый TS. Первый уровень похож на TS из предыдущей статьи, используется для присваивания коробок контейнерам без выбора их расположения. Затем в окрестностях меняется порядок установки путем перестановки и перераспределения коробок между контейнерами. Второй уровень TS применяется для установки коробок в каждом контейнере с помощью эвристики критических точек и поиска на интервале графа, предложенного в работах [60, 61]. Каждая пара коробок в контейнере согласно заданному подходу будет сформирована в семь доступных комбинаций.

Вместо присваивания коробок контейнерам, в работе [28] TS используется для присваивания коробок двумерным слоям. Предложенный унифицированный подход к применению TS также успешно использован и в двумерной упаковке. Алгоритм выбирает коробки для каждого слоя и позволяет уменьшить суммарное число слоев. Перемещения в окрестностях позволяют освободить специальный целевой слой, содержащий меньше всего элементов. Единственная коробка из целевого слоя объединяется с коробками в k других контейнеров. Окрестность содержит разные значения k и различные коробки в целевом слое. Список табу содержит штрафы перемещений в конце каждого хода.

Управляемый локальный поиск (GLS) имеет много общего с TS в плане использования памяти для управления поиском. В работе [63] впервые разрешено пересечение между коробками внутри контейнера. Решается задача упаковки в несколько контейнеров. Ищется возможное расположение коробок в одном контейнере. Как только оно найдено, контейнер исключается, а коробки распределяются по оставшимся контейнерам и поиск начинается снова. Допускаются локальные перемещения ящиков – движения ящика вдоль оси координат в контейнере или перенос ящика на те же координаты, но в другой контейнер. Параметры, уводящие локальные поиск от локального оптимума определены объемом перекрытия между коробкам i, j и суммарным объемом коробок i, j. Быстрый локальный поиск в данном случае используется для нахождения локального минимума.

В работе [32] локальные перемещения заключаются в замене двух коробок в очереди упаковки, а поиск регулируется по стандартному графику охлаждения. Результаты подходят для двумерной задачи. В работе [64] также используются локальные перемещения в очереди коробок. Однако явным недостатком такого подхода является вычислительная стоимость восстановления решения после каждого шага. Автор отмечает, что при количестве ящиков больше 64 такое улучшение приносит лишь замедление процесса получения решения и не рекомендуется для использования.

В то время как GA, TS и SA являются стандартным набором из трех метаэвристик, жадный алгоритм со случайным адаптивным поиском (greedy randomized adaptive search procedure – GRASP) используется реже. GRASP разработан для использования жадной эвриситики получения решения и изменения параметров процесса получения решения, создавая случайные выборки из ограниченного списка кандидатов вместо наиболее жадного выбора. Таким образом, этот подход обладает очевидными преимуществами для использования его в эвристиках расположения для упаковки контейнера.

В работе [12] рассматривается задача открытого размещения, в которой коробки группируются в наборы согласно заказам клиентов и упаковываются в контейнер по-отдельности. Таким образом доказывается, что возможно решать каждую подзадачу отдельно для уменьшения размера каждого такого набора. Авторы представляют задачу в виде графа и показывают, что оптимальное решение достигается тогда, когда максимальная клика равна числу коробок. Они решают задачу максимальной клики с использованием GRASP.

В работе [17] все типы коробок сортируются по объему занимаемого пространства. Базовый жадный алгоритм выбирает первый тип коробок. Используя GRASP, выбирается случайная коробка из списка лучших кандидатов с учетом порогового параметра со значением между нулем и единицей. Если параметр равен единице, выбор производится случайным образом, а когда он равен нулю, используется последовательный выбор. Полученное решение затем улучшается с помощью локального поиска, в котором в конце очереди коробки меняются местами, запрещая возвращать последнюю с конца коробку на свою предыдущую позицию. Похожее применение GRASP используется в работе [10] и объединяется с эвристикой расположения максимального пространства. Здесь кандидаты попадают в ограниченный список согласно своему рангу (шанс оказаться в начале списка составляет $100\% * a$, где a – критерий ранжирования, $a \in [0; 1]$) без использования порогового параметра вхождения. Первоначально a выбирается случайно из набора возможных значений. В процессе поиска вероятности зависят от каждого возможного значения, основанного на его прошлых успехах.

Следуя успешным применениям GRASP, некоторые авторы решили объединять эвристику максимального пространства с алгоритмом VNS. В работе [9] предложено пять типов локальных перемещений для уменьшения фазы VNS и дополнительное перемещение для фазы смешивания:

- процедура обрезания слоя, которая удаляет столбцы или строки из слоя и пытается заполнить пустое пространства другими коробками;
- добавление колонки в пустое пространство;
- добавление одной коробки в пустое пространство;
- очистка уже упакованной области;
- заполнение области таким образом, чтобы одни коробки в одной окрестности обладали максимальным объемом, а в соседней области – наиболее оптимальной формы.

Смешивание окрестности затрагивает от 10% до 30% коробок текущего решения и заменяет их на основе критерия лучшего объема.

В то время как метаэвристические подходы, описанные выше, стремятся быть менее жадными или по крайней мере включать в себя фазы, позволяющие разнообразить поиск в пространстве решений, некоторые авторы предпочитают лишь запускать жадный поиск много раз подряд. Например, в работах [65, 66] используется локальный поиск по перестановкам и ориентациям во время операции обмена. Множественные начальные решения обеспечивают более широкий поиск по пространству решений.

Другой подход заключается в применении жадного выбора после просмотра нескольких шагов, что уменьшает краткосрочную фокусировку на решении. В работе [34] описан поиск на дереве, просматривающий на несколько шагов вперед, который определяет очередное расположение на основе лучшего узла расширенного дерева глубиной d, где для каждого узла раскрыты m лучших дочерних узлов. В предшественнике этой работы, статье [37], используется ограниченный поиск по дереву для получения решения. Вначале происходит поиск по дереву в ширину с ограничением на число рассматриваемых дочерних узлов (на глубину поиска). Для того, чтобы избежать доминирующего пути, если два узла находятся на одной глубине, а значения вычисляемой функции у них одинаковы и обе коробки упакованы, то один из узлов удаляется. Однако авторы называют поиск по дереву пробным методом, надеясь на дальнейшие исследования в данном направлении.

6 Точные методы

По сравнению с числом всевозможных эвристик, количество точных алгоритмов решения трехмерной задачи упаковки в контейнер ограничено. Одна из причин этого заключается в сложности представления вероятной упаковки и введения ограничений из реальных задач. Даже если такой метод решения найден, остается сложность в формулировке решения, зачастую являющимся большим из-за числа коробок и контейнеров. Тем не менее за все время существования этой проблемы были проведены исследования и разработаны различные подходы, которые и рассмотрим дальше.

В работе [67] описан алгоритм, решающий задачу упаковки коробок различных типов, со своими заранее известными размерами и числом коробок, в набор контейнеров различных размеров с целью максимизировать суммарный объем упакованных коробок. Первоначально такая задача названа многомерной задачей о рюкзаке, включая в себя целое число, присвоенное каждому типу контейнеров, упакованному в соответствии с данной схемой.

Так как число таких схем чрезмерно велико, сложность задачи и получение оптимального решения являются серьезной проблемой. Из-за этого авторы прибегают к процедуре «генерации колонок», в которой используется целочисленное программирование, а именно дискретная задача о рюкзаке со сторонними ограничениями, сама по себе являющейся сложной для решения. Поэтому авторы предлагают эвристики решения такой дискретной задачи. В работе [14] рассмотрена похожая проблема, с некоторыми ограничениями и целевой функцией, минимизирующей число контейнеров благодаря двухфазному алгоритму генерации колонок. На первой фазе алгоритма ограниченное число колонок генерируются априори, используя эвристику, которые затем передаются для решения задачи целочисленного программирования во второй фазе алгоритма. Такой подход обладает несомненным плюсом – возможностью включения набора реальных ограничений, таких как ориентация коробки или стабильность упаковки. Результаты вычислений, представленные автором на примере решения тестовой задачи показали, что затраты на вычисление задачи линейного программирования занимает не более пяти секунд от всего времени решения, а рассмотренный алгоритм превосходит аналогичные в работах [68, 69].

Поскольку генерация колонок приводит к успешному решению, то не удивительно, что в работе [70] описан похожий алгоритм для MBSBPP, где целью также является минимизация числа используемых контейнеров. В эту задачу включены и некоторые ограничения, а именно возможность свеса коробок и ортогональная упаковка. Наиболее затратная в вычислительном плане проблема здесь заключается в загрузке одного контейнера. Вместо генерирования требуемых колонок, что дорого, авторы прибегают к стратегии, генерирующей необходимые приближения к колонкам, названные прототипами колонок. По сравнению с алгоритмом, представленным в работе [71], этот метод находит решение на 50% более близкое к оптимальному решению.

Другой подход к математическим моделям, когда переменные определяются в зависимости от каждой конкретной упаковки, заключается в том, чтобы использовать формулы, в которых переменные определены для каждой конкретной установленной коробки. Такой подход позволяет избежать проблем при работе с большим набором упаковок и вместо этого полагается на дискретизацию пространства контейнера и явное представление решения того, будут ли коробки установлены в рассматриваемые текущие координаты или нет.

В работе [72] представлена такая модель как объединение линейного программирования для общей задачи трехмерной упаковки контейнера с целью минимизации пустого пространства и числа используемых контейнеров. Модель используется для решения тестовой задачи из трех разных контейнеров и шести коробок с использованием LINGO, причем само решение заняло 15 минут. Авторы также представили результаты другого примера, состоящего из одного контейнера и шести коробок и с целью минимизации длины контейнера. Многогранное исследование и улучшение такого подхода представлено в работе [73]. В статье показаны некоторые фасеточные результаты и верные неравенства, с рекомендациями использовать их совместно с методом ветвей и отсечений для решения поставленной задачи. В работе [74] приведено эмпирическое доказательство того, что возможно вычислить задачу при такой постановке максимум при 20 коробках с использованием современных решателей. Поэтому автор предложил альтернативную постановку, основанную на пространственно-индексных переменных, которая обладает лучшей эффективностью, чем в работах [72, 73], зависящих от размеров конкретных задач и времени вычисления их оптимума.

В работе [75] рассмотрена объединенная задача целочисленного программирования упаковки в однотипные контейнеры, а также представлены верные неравенства и результаты вычислений. Другая модель, основанная на декартовой системе координат, представлена в работе [76] для решения задачи с ограничениями для обеспечения вертикальной, горизонтальной стабильности и учета несущей способности коробок. Результаты вычислений в этой статье показывают, что такая модель способна в большинстве случаев решать (оптимально) случайно генерируемые случаи, в которых используются четыре типа коробок до 20 штук в каждом типе максимум и от 10 до 100 контейнеров. Сложность в нахождении оптимального решения увеличивается с числом контейнеров также как и при сходстве в размерах коробок.

В работе [25] использован эвристический алгоритм для рассматриваемой нами задачи с учетом ограничений по весу на осях, связанных с нагрузкой на грузовики, груженые контейнерами, при их выгрузке из корабля. Учет этого условия вводит нелинейности в постановку задачи целочисленного программирования. Поэтому авторы придумали и описали эвристическую процедуру, но использовали именно целочисленное программирование, смешанное или двоичное, вкупе с алгоритмом для улучшения упаковки.

Наконец, необходимо упомянуть работу [33], в которой используются приближенные алгоритмы для решения нашей задачи. Алгоритмы основаны на идее решения набора задач о рюкзаке для формирования стеков, с дальнейшим помещением стеков в контейнер вследствие решения неограниченной задачи о рюкзаке.

7 Реализации в зависимости от типа задачи

7.1 Упаковка единичного контейнера

Всего рассмотрено 25 работ, в которых рассматривается задача SLOPP, причем методология сильно варьируется. В тринадцати статьях рассмотрены только эвристики расположения, в семи из которых [6, 13, 15, 16, 27, 41, 78] используют статическое ранжирование и критерий расположения. В оставшихся шести работах используется динамический выбор очередной коробки или места для упаковки [11, 26, 21, 3] или поиск на дереве [8, 51] для того, чтобы исследуемый алгоритм стал менее жадным. В одиннадцати работах используются или разрабатываются новые улучшенные эвристики для решения поставленной задачи. В двух из них [33, 79] используется GA, в еще двух [24, 29] используется TS в работе [58] используется SA, в оставшихся [9, 10, 17, 37, 42, 77, 80] используются различные вариации адоптированного локального поиска. А в работе [76], в единственной из всех рассмотренных работ, используются точные методы решения.

В 33 работах рассмотрено решение SKP. Методологии также сильно различаются по типу и сложности. В десяти работах рассмотрены только эвристики расположения, в половине из которых [5, 20, 38, 81, 82] авторы используют статическое ранжирование и критерий расположения. В оставшейся половине работ применяется динамический выбор очередной коробки или места для упаковки [7, 39, 80, 83] или поиск на дереве [18]. В 19 статьях применяются улучшенные эвристики, из которых в восьми случаях [2,22, 33, 54, 84-87] используется GA, в работе [29] авторы используют TS, в статье [4] применяется SA, а в оставшихся работах [9, 10, 19, 45, 47, 48, 83, 88] используются различные вариации адоптированного локального поиска. В четырех работах [73, 76, 89, 90] используются точные методы решения.

7.2 Упаковка в несколько контейнеров

Для решения этой задачи в основном используется одна из трех стратегий.

1) Последовательная стратегия, используемая в работах [52, 55, 91, 92] заключается в том, что контейнеры упаковываются один за другим. Новый контейнер начинает

заполняться только в том случае, когда в текущий контейнер невозможно упаковать ни одну из оставшихся коробок. Однако, одним из недостатков такого метода является то, что в контейнере, упакованном последним, может оказаться слишком много свободного пространства, потому что крупногабаритные или коробки неудобной формы обычно остаются напоследок.

2) Предварительная стратегия – первым делом распределяет коробки по контейнерам, и во время этой стадии не производится упаковка. Эвристика применяется для непосредственно упаковки соответствующих коробок в соответствующие контейнеры. Ящики, не поместившиеся в свои контейнеры, упаковываются в другие контейнеры согласно специальным механизмам. Работа [59] является типичным примером применения такой стратегии.
3) Одновременная стратегия, применявшаяся в работах [72, 93, 94] заключается в том, что предпринимаются попытки упаковать коробки одновременно в несколько контейнеров. Удивительно, но такой подход более эффективен, чем последовательная стратегия из работы [37].

Однако есть несколько исключений, когда используются смешанные стратегии. В работе [71] сначала коробки укладываются в определенные наборы (шаблоны), а затем уже сформированные наборы упаковываются в один контейнер, причем время упаковки зависит от конкретного набора. В работе [62] автор получает начальное решение, используя последовательную стратегию, применяя улучшенную фазу вкупе с предварительной стратегией. Наконец, в работе [63] начальное решение получается генерированием нескольких стен перед тем, как объединить их в контейнеры.

В семи работах рассмотрена задача SSSCSP. Среди них в работах [11, 68] показана лишь эвристика расположения, а в статьях [37, 71, 72, 95] разработаны улучшенные эвристики. Работа [96] – единственная, в которой рассматривается план погрузки контейнеров на судно путем сопоставления такой задачи с SSSCSP в трехмерной упаковке.

Всего рассмотрено 14 работ, относящихся к задаче SBSCBPP. Большинство работ [43, 44, 80, 83, 91, 97-99] здесь направлены на построение эвристики расположения. Среди этих улучшенных эвристик чаще всего [28, 59, 62, 100] используется TS, что удивительно, в то время как в работе [63] автор предложил применять GLS. Работа [75] единственная, в которой применяются точные методы, хотя они встроены в более обширную эвристику, когда в работе [101] автор предложил новую нижнюю границу.

Среди работ, рассматривающих решение задачи MSSCSP, лишь в статье [68] рассмотрена только эвристика расположения, в то время как все остальные работы [14, 71, 72, 102] посвящены улучшенным эвристикам.

Все четыре работы [59, 93, 102, 103], посвященные задаче MBSBPP, содержат улучшенные эвристики. В работе [72] автор утверждает, что с помощью представленного в статье подхода возможно решить любой тип задачи упаковки из рассмотренных нами, в том числе и RCSP. Однако, конкретно алгоритма для решения RCSP предложено не было.

В двух работах [59, 93] рассмотрена задача RBPP. Также как и в RCSP, в работе [77] предложенный алгоритм способен решить задачу MILOPP, но все равно этот тип задач остается практически нерешаемым. MIKP является третьим типом задачи, оставшимся неизученным. И было найдено лишь две работы [14, 67], в которых решается задача MHLOPP. Методы, изложенные в статье [103] также решают и задачу MHKP.

8 Оценка эффективности алгоритмов

Для оценки эффективности работы методов упаковки разработаны специальные наборы тестовых данных в зависимости от разновидности решаемой задачи. В данной работе использовались три каноничных набора.

1) Набор из 47 примеров, состоящих из однородных коробок разного числа и заполнением единичного контейнера постоянных размеров, но различных среди примеров [68]. Оценивается суммарное среднее время работы алгоритма. Лучшие результаты у алгоритмов (согласно рис. 2): 12 – построение колонн без учета ограничения стабильности [70], 13 – построение колонн с учетом ограничения стабильности [70].

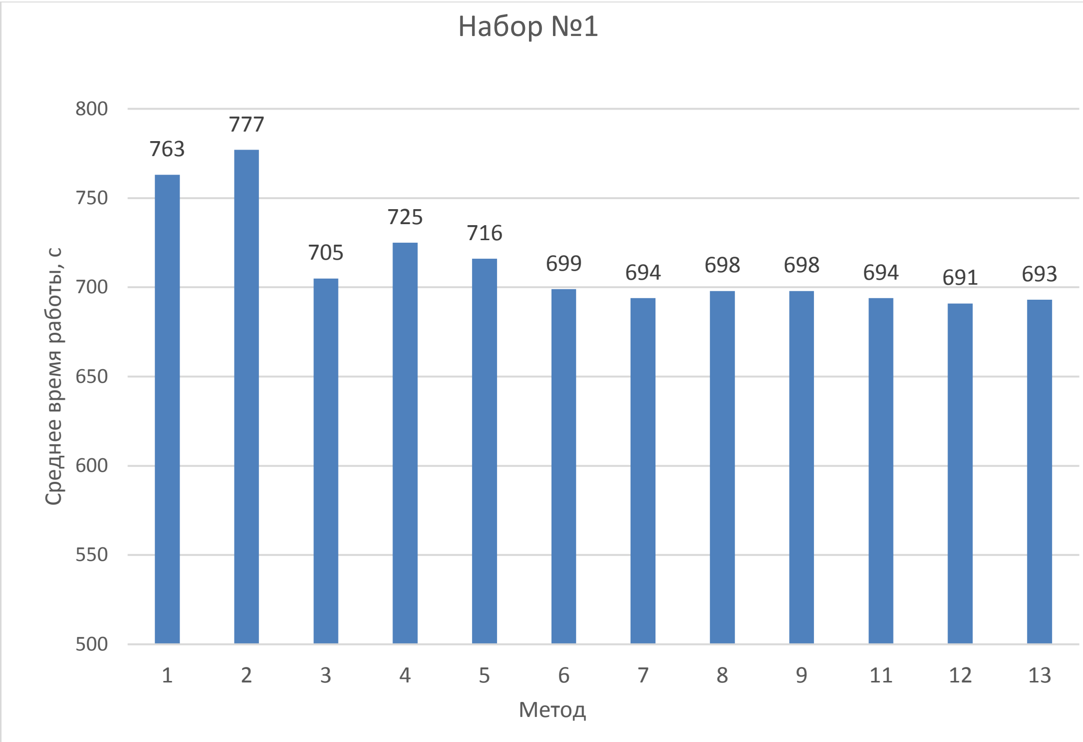

Рисунок 2. Среднее время работы алгоритмов на тестовом наборе №1

2) Набор из 15 примеров, состоящих из однородных коробок и одного контейнера, который необходимо заполнить [27]. При этом могут остаться неупакованные коробки. Оценивается плотность полученной упаковки в процентах. Диаграмма с результатами работы алгоритмов представлена на рис. 3. Наиболее эффективными оказались алгоритмы: 14 – приближение, основанное на оценке уровня пещерности [47], 16 –алгоритм на основе поиска на дереве [30], 17 – итеративное приближение на основе динамических приоритетах коробок [25].

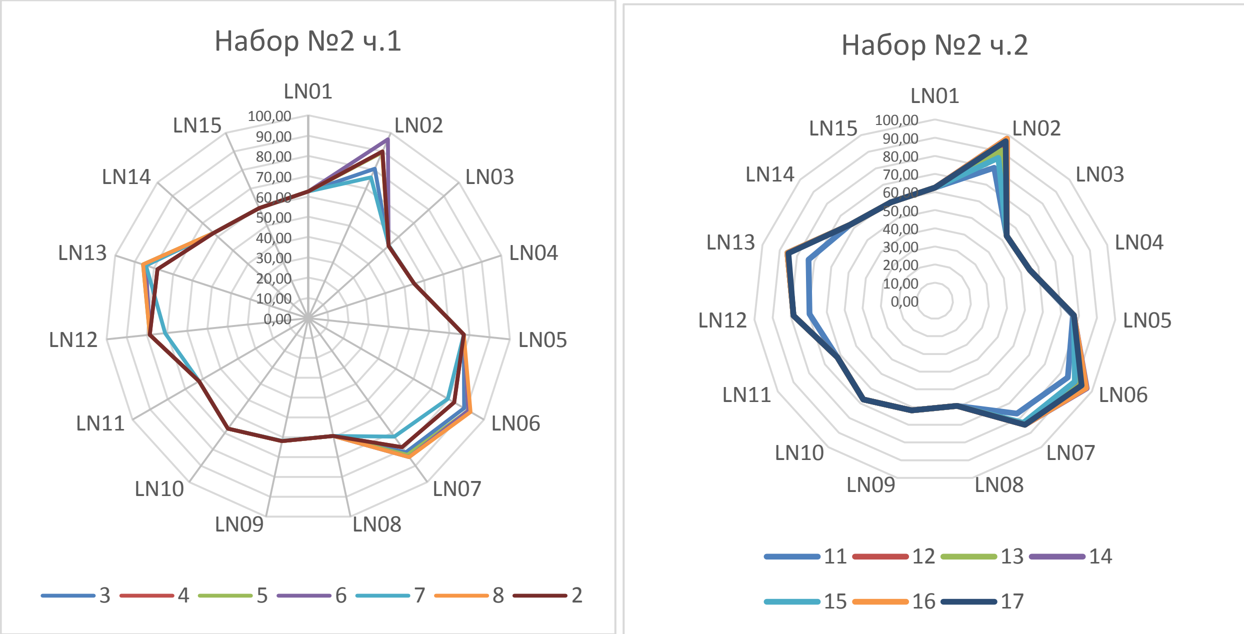

Рисунок 3. Плотность получившейся упаковки на тестовом наборе №2

3) Набор из 8 классов примеров, состоящих из разнородных коробок, которыми необходимо заполнить минимальное число одинаковых контейнеров [28]. Оценивается время работы алгоритма. На рис. 4 представлены результаты работы алгоритмов – среднее время работы по классам с различными габаритами коробок и суммарное время работы по всем классам. Наиболее эффективными оказались алгоритмы: 8 – двухуровневый табуированный поиск [62], 6 – алгоритм экстремальных точек [63].

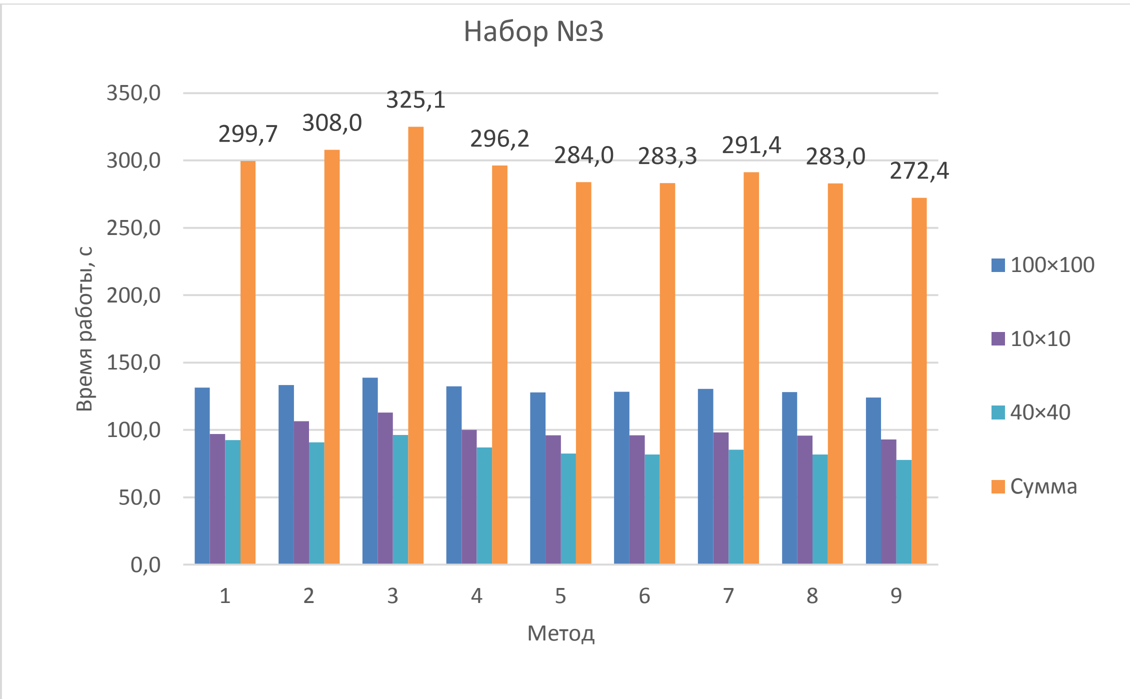

Рисунок 4. Среднее время работы алгоритмов на тестовом наборе №3

9 Заключение

В данной работе был произведен литературный обзор статей, посвященных задаче трехмерной упаковке в контейнер, рассмотрены основные идеи подходов к решению этой задачи, их реализации и оценки эффективности. Для этого было рассмотрено в общей сложности 103 работы, охватывающие всевозможные варианты рассматриваемой задачи и использующие широкий спектр методологий, которые были представлены в данной работе, а также эвристики расположения, улучшенные эвристики и точные методы.

На основании проведенного анализа можно сделать следующие выводы.

1) Намного меньше работ посвящены загрузке нескольких контейнеров по сравнению с загрузкой одного контейнера, причем большинство из них лишь проецируют модель упаковки одного контейнера на несколько. Но даже среди этого небольшого набора еще меньше исследований решают вариант задачи с разнородными контейнерами.
2) Среди всех работ чаще всего не учитываются ограничения реальных задач. Лишь в нескольких статьях эти ограничения охватываются всесторонне, большинство работ фокусируется на нескольких определенных ограничениях.

Список литературы

1. Bortfeldt A., W¨ascher G. Constraints in container loading: a state-of-the-art review // European Journal of Operational Research. 2013. Vol. 229. Is.1. P. 1–20. DOI:10.1016/j.ejor.2012.12.006
2. Gehring H., Bortfeldt A. A genetic algorithm for solving the container loading problem // International Transactions in Operational Research. 1997. Vol 4. Is.5-6. P. 401–418. DOI:10.1016/S0969-6016(97)00033-6
3. Wang Z., Li K.W., Levy J.K. A heuristic for the container loading problem: a tertiary-tree-based dynamic space decomposition approach // European Journal of Operational Research. 2008. Vol 191. Is.1. P. 86–99. DOI:10.1016/j.ejor.2007.08.017
4. Egeblad J., Pisinger D. Heuristic approaches for the two- and three-dimensional knapsack packing problem // Computers & Operations Research. 2009. Vol. 36. Is. 4. P. 1026–1049. DOI:10.1016/j.cor.2007.12.004
5. Haessler R.W., Talbot F.B. Load planning for shipments of low density products // European Journal of Operational Research. 1990. Vol 44. Is.2. P. 289– 299. DOI:10.1016/0377-2217(90)90364-H
6. Chien C.F., Deng, J.F. A container packing support system for determining and visualizing container packing patterns // Decision Support Systems. 2004. Vol 37. № 1. P. 23–34. DOI: 10.1016/S0167-9236(02)00192-6
7. Scheithauer G. Algorithms for the container loading problem // Operations Research Proceedings. 1991. P. 445–452. DOI: 10.1007/978-3-642-46773-8_112

8. Morabito R., Arenales M. An AND/OR-graph approach to the container loading problem // International Transactions in Operational Research. 1994. Vol. 1. Is. 1. P. 59–73. DOI:10.1016/0969-6016(94)90046-9
9. Parreño F., Alvarez-Valdes R., Oliveira J., Tamarit J. Neighborhood structures for the container loading problem: a VNS implementation // Journal of Heuristics. 2010. Vol. 16. Is. 1. P. 1–22. DOI:10.1007/s10732-008-9081-3
10. Parreño F., Alvarez-Valdes R., Tamarit J.M., Oliveira J.F. A maximal-space algorithm for the container loading problem // INFORMS Journal on Computing. 2008. Vol. 20. Is. 3. P. 412–422.
11. Bischoff E.E., Ratcliff M.S.W. Issues in the development of approaches to container loading // Omega-international Journal of Management Science. 1995. Vol. 23. Is. 4. P. 377–390. DOI:10.1016/0305-0483(95)00015-G
12. Lai K.K., Xue J., Xu B. Container packing in a multi-customer delivering operation // Computers & Industrial Engineering. 1998. Vol. 35. In. 1-2. P. 323–326. DOI:10.1016/S0360-8352(98)00085-0
13. Ren J., Tian Y., Sawaragi T. A tree search method for the container loading problem with shipment priority // European Journal of Operational Research. 2011. Vol. 214. Is. 3. P. 526–535. DOI:10.1016/j.ejor.2011.04.025
14. Eley M. A bottleneck assignment approach to the multiple container loading problem // OR Spektrum. 2003. Vol. 25. Is. 1. P. 45–60. DOI: 10.1007/s002910200113
15. George J.A., Robinson D.F. A heuristic for packing boxes into a container // Computers & Operations Research. 1980. Vol. 7. Is. 3. P. 147–156. DOI:10.1016/0305-0548(80)90001-5
16. Bischoff E.E., Marriott M.D. A comparative evaluation of heuristics for container loading // European Journal of Operational Research. 1990. Vol. 44. Is. 2. P. 267–276. DOI: 10.1016/0377-2217(90)90362-F
17. Moura A., Oliveira J.F. A GRASP approach to the container-loading problem // IEEE Intelligent Systems. 2005. Vol. 20. P. 50–57. DOI: 10.1109/MIS.2005.57
18. Chien C.F., Wu W.T. A recursive computational procedure for container loading // Computers & Industrial Engineering. 1998. Vol. 35. Is. 1-2. P. 319–322. DOI:10.1016/S0360-8352(98)00084-9
19. Pisinger D. Heuristics for the container loading problem // European Journal of Operational Research. 2002. Vol. 141. Is. 2. P. 382–392. DOI: 10.1016/S0377-2217(02)00132-7
20. Gehring H., Menschner K., Meyer M. A computer-based heuristic for packing pooled shipment containers // European Journal of Operational Research. 1990. Vol. 44. Is. 2. P. 277–288. DOI:10.1016/0377-2217(90)90363-G
21. Davies A.P., Bischoff E.E. Weight distribution considerations in container loading // European Journal of Operational Research. 1999. Vol. 114. Is. 3. P. 509– 527. DOI:10.1016/S0377-2217(98)00139-8

22. Bortfeldt A., Gehring H. A hybrid genetic algorithm for the container loading problem // European Journal of Operational Research. 2001. Vol. 131. Is. 1. P. 143– 161. DOI:10.1016/S0377-2217(00)00055-2
23. Bortfeldt A., Gehring H. Applying tabu search to container loading problems // (Symposium on Operations Research (SOR'97) Jena, September 3–5, 1997). Operations Research Proceedings 1997. Springer Berlin Heidelberg. 1998. Vol. 1997, 1998. P. 533–538. DOI: 10.1007/978-3-642-58891-4_84
24. Bortfeldt A., Gehring H., Mack D. A parallel tabu search algorithm for solving the container loading problem // Parallel Computing. 2003. Vol. 29. Is. 5. P. 641–662. DOI:10.1016/S0167-8191(03)00047-4
25. Lim L.C.A., Ma H., Xu J., Zhang X. An iterated construction approach with dynamic prioritization for solving the container loading problems // Expert Systems with Applications. 2012. Vol. 39. Is. 4. P. 4292–4305. DOI:10.1016/j.eswa.2011.09.103
26. Ratcliff M.S.W, Bischoff E.E. Allowing for weight considerations in container loading // OR Spektrum. 1998. Vol. 20. Is. 1. P. 65–71. DOI: 10.1007/BF01545534
27. Loh T.H., Nee A.Y.C. A packing algorithm for hexahedral boxes // Proceedings of the Conference of Industrial Automation. Singapore. 1992. P. 115–126.
28. Lodi A., Martello S., Vigo D. Heuristic algorithms for the three-dimensional bin packing problem // European Journal of Operational Research. 2002. Vol. 141. Is. 2. P. 410–420. DOI:10.1016/S0377-2217(02)00134-0
29. Liu J., Yue Y., Dong Z., Maple C., Keech M. A novel hybrid tabu search approach to container loading // Computers & Operations Research. 2011. Vol. 38. Is. 4. P.797–807. DOI:10.1016/j.cor.2010.09.002
30. Ren J., Tian Y., Sawaragi T. A tree search method for the container loading problem with shipment priority // European Journal of Operational Research. 2011. Vol. 214. Is. 3. P. 526–535. DOI:10.1016/j.ejor.2011.04.025
31. Zhang D., Peng Y., Leung S.C.H. A heuristic block-loading algorithm based on multi-layer search for the container loading problem // Computers & Operations Research. 2012. Vol. 39. Is. 10. P. 2267–2276. DOI:10.1016/j.cor.2011.10.019
32. Lai K.K., Chan J.W.M. Developing a simulated annealing algorithm for the cutting stock problem // Computers & Industrial Engineering. 1997. Vol. 32. Is. 1. P. 115–127. DOI:10.1016/S0360-8352(96)00205-7
33. Gonçalves J.F., Resende M.G.C. A parallel multi-population biased random-key genetic algorithm for a container loading problem // Computers & Operations Research. 2012. Vol. 39. Is. 2. P. 179–190. DOI:10.1016/j.cor.2011.03.009
34. Zhu W., Lim A. A new iterative-doubling Greedy-Lookahead algorithm for the single container loading problem // European Journal of Operational Research. 2012. Vol. 222. Is. 3. P. 408–417. DOI:10.1016/j.ejor.2012.04.036

35. Zhu W., Oon W.C., Lim A., Weng Y. The six elements to block-building approaches for the single container loading problem // Applied Intelligence. 2012. Vol. 37. Is. 3. P. 431–445. DOI:10.1007/s10489-012-0337-0
36. Araya I., Riff M.C. A beam search approach to the container loading problem // Computers & Operations Research. 2014. Vol. 43. P. 100–107. DOI:10.1016/j.cor.2013.09.003
37. Eley M. Solving container loading problems by block arrangement // European Journal of Operational Research. 2002. Vol. 141. Is. 2. P. 393–409. DOI:10.1016/S0377-2217(02)00133-9
38. Lim A., Rodrigues B., Wang Y. A multi-faced buildup algorithm for three-dimensional packing problems // Omega. 2003. Vol. 31. Is. 6. P. 471–481. DOI:10.1016/j.omega.2003.08.004
39. Ngoi B.K.A., Tay M.L., Chua E.S. Applying spatial representation techniques to the container packing problem // International Journal of Production Research. 1994. Vol. 32. Is. 1. P. 111–123. DOI: 10.1080/00207549408956919
40. Chua C.K., Narayanan V., Loh J. Constraint-based spatial representation technique for the container packing problem // Integrated Manufacturing Systems. 1998. Vol. 9. Is. 1. P. 23–33. DOI: 10.1108/09576069810196814
41. Chien C.F., Lee C.Y., Huang Y.C., Wu W.T. An efficient computational procedure for determining the container loading pattern // Computers & Industrial Engineering. 2009. Vol. 56. Is.3. P. 965–978. DOI:10.1016/j.cie.2008.09.019
42. Bischoff E.E. Three-dimensional packing of items with limited load bearing strength // European Journal of Operational Research. 2006. Vol. 168. Is. 3. P. 952– 966. DOI:10.1016/j.ejor.2004.04.037
43. Martello S., Pisinger D., Vigo D. The three-dimensional bin packing problem // Operations Research. 2000. Vol. 48. Is. 2. P. 256–267. DOI:10.1287/opre.48.2.256.12386
44. Crainic T.G., Perboli G., Tadei R. Extreme point-based heuristics for three-dimensional bin packing // INFORMS Journal on Computing. 2008. Vol. 20. Is. 3. P. 368–384.
45. Huang W., He K. A caving degree approach for the single container loading problem // European Journal of Operational Research. 2009. Vol. 196. Is. 1. P. 93– 101. DOI:10.1016/j.ejor.2008.02.024
46. Huang W., He K. A new heuristic algorithm for cuboids packing with no orientation constraints // Computers & Operations Research. 2009. Vol. 36. Is. 2. P. 425–432. DOI:10.1016/j.cor.2007.09.008
47. He K., Huang W. A caving degree based flake arrangement approach for the container loading problem // Computers & Industrial Engineering. 2010. Vol. 59. Is. 2. P. 344–351. DOI:10.1016/j.cie.2010.05.007
48. He K., Huang W. An efficient placement heuristic for three-dimensional rectangular packing // Computers & Operations Research. 2011. Vol. 38. Is.1. P. 227– 233. DOI:10.1016/j.cor.2010.04.015

49. Han C.P., Knott K., Egbelu P.J. A heuristic approach to the three dimensional cargo loading problem // Computers & Industrial Engineering. 1986. Vol. 11. Is. 1 - 4. P. 109–113. DOI:10.1016/0360-8352(86)90059-8
50. George J.A. A method for solving container packing for a single size of box // Journal of the Operational Research Society. 1992. Vol. 43. Is. 4. P. 307–312. DOI:10.1038/sj/jors/0430402
51. Lins L., Lins S., Morabito R. An n-tet graph approach for non-guillotine packings of n-dimensional boxes into an n-container // European Journal of Operational Research. 2002. Vol. 141. Is. 2. P. 421–439. DOI:10.1016/S0377-2217(02)00135-2
52. Xue J., Lai K.K. Effective methods for a container packing operation // Mathematical and Computer Modelling. 1997. Vol. 25. Is. 2. P. 75–84. DOI:10.1016/S0895-7177(97)00008-3
53. Hemminki J. Container Loading with Variable Strategies in Each Layer // Institute for Applied Mathematics, University of Turku, Turku, Finland. 1994.
54. Gehring H., Bortfeldt A. A parallel genetic algorithm for solving the container loading problem // International Transactions in Operational Research. 2002. Vol. 9. Is. 4. P. 497–511. DOI:10.1111/1475-3995.00369
55. Wu Y., Li W., Goh M., De Souza R. Three-dimensional bin packing problem with variable bin height // European Journal of Operational Research. 2010. Vol. 202. Is. 2. P. 347–355. DOI:10.1016/j.ejor.2009.05.040
56. Gonçalves J.F., Resende M.G. A biased random key genetic algorithm for 2D and 3D bin packing problems // International Journal of Production Economics. 2013. Vol. 145. Is. 2. P. 500–510. DOI:10.1016/j.ijpe.2013.04.019
57. Dereli T., Das G.S. A hybrid ‘bee(s) algorithm’ for solving container loading problems // Applied Soft Computing. 2011. Vol. 11. Is. 2. P. 2854–2862. DOI:10.1016/j.asoc.2010.11.017
58. Mack D., Bortfeldt A., Gehring H. A parallel hybrid local search algorithm for the container loading problem // International Transactions in Operational Research. 2004. Vol. 11. Is. 5. P. 511–533. DOI:10.1111/j.1475-3995.2004.00474.x
59. Jin Z., Ito T., Ohno K. The three-dimensional bin packing problem and its practical algorithm // JSME International Journal Series C - Mechanical Systems, Machine Elements and Manufacturing. 2003. Vol. 46. Is. 1. P. 60 – 66. DOI:10.1299/jsmec.46.60
60. Fekete S.P., Schepers J. A new exact algorithm for general orthogonal d-dimensional knapsack problems // Conference: European Symposium on Algorithms. (5th Annual European Symposium Graz, Austria, September 15–17, 1997) Proceedings. Berlin: Springer-Heidelberg. 1997. P. 144 – 156. DOI: 10.1007/3-540-63397-9_12
61. Fekete S.P., Schepers J. A combinatorial characterization of higher-dimensional orthogonal packing // Mathematics of Operations Research. 2004. Vol. 29. Is. 2. P. 353–368.

62. Crainic T.G., Perboli G., Tadei R. TS^{2}PACK: a two-level tabu search for the three-dimensional bin packing problem // European Journal of Operational Research. 2009. Vol. 195. Is. 3. P. 744–760. DOI:10.1016/j.ejor.2007.06.063
63. Farøe O., Pisinger D., Zachariasen M. Guided local search for the three-dimensional bin-packing problem // INFORMS Journal on Computing. 2003. Vol. 15. Is. 3. P. 267–283.
64. Faina L. A global optimization algorithm for the three-dimensional packing problem // European Journal of Operational Research. 2000. Vol. 126. Is. 2. P. 340– 354. DOI:10.1016/S0377-2217(99)00292-1
65. Takahara S. A simple meta-heuristic approach for the multiple container loading problem // Browse Conference Publications. (Systems, Man and Cybernetics. 8-11 Oct. 2006. Taipei). IEEE International Conference on Systems, Man and Cybernetics. 2006. Vol. 3. P. 2328–2333. DOI: 10.1109/ICSMC.2006.385210
66. Takahara S. A multi-start local search approach to the multiple container loading problem // Greedy Algorithms / Witold Bednorz. Vienna: IN-TECH. 2008. 586 p. Ch. 4. P. 55–68. DOI: 10.5772/6358
67. Mohanty B.B., Mathur K., Ivancic N.J. Value considerations in three-dimensional packing: a heuristic procedure using the fractional knapsack problem // European Journal of Operational Research. 1994. Vol. 74. Is. 1. P. 143 – 151. DOI:10.1016/0377-2217(94)90212-7
68. Ivancic N.J., Mathur K., Mohanty B.B. An integer programming based heuristic approach to the three-dimensional packing problem // Journal of Manufacturing and Operations Management. 1989. Vol. 2. Is. 4. P. 268–298.
69. Bortfeldt A. Eine Heuristik für Multiple Containerladeprobleme // OR Spektrum. 2000. Vol. 22. Is. 2. P. 239–261. DOI: 10.1007/s002910050104
70. Zhu W., Huang W., Lim A. A prototype column generation strategy for the multiple container loading problem // European Journal of Operational Research. 2012. Vol. 223. Is. 1. P. 27–39. DOI:10.1016/j.ejor.2012.05.039
71. Che C. H., Huang W., Lim A., Zhu W. The multiple container loading cost minimization problem // European Journal of Operational Research. 2011. Vol. 214. Is. 3. P. 501–511. DOI:10.1016/j.ejor.2011.04.017
72. Chen C.S., Lee S.M., Shen Q.S. An analytical model for the container loading problem // European Journal of Operational Research. 1995. Vol. 80. Is. 1. P. 68–76. DOI:10.1016/0377-2217(94)00002-T
73. Padberg M. Packing small boxes into a big box // Mathematical Methods of Operations Research. 2000. Vol. 52. Is. 1. P. 1–21. DOI: 10.1007/s001860000066
74. Allen S.D., Burke E.K., Mareček J. A space-indexed formulation of packing boxes into a larger box // Operations Research Letters. 2012. Vol. 40. Is. 1. P. 20–24. DOI:10.1016/j.orl.2011.10.008

75. Hifi M., Kacem I., Nègre S., Wu L. A linear programming approach for the three-dimensional bin-packing problem // Electronic Notes in Discrete Mathematics. 2010. Vol. 36. P. 993–1000. DOI:10.1016/j.endm.2010.05.126
76. Junqueira L., Morabito R., Yamashita D.S. MIP-based approaches for the container loading problem with multi-drop constraints // Annals of Operations Research. 2012. Vol. 199. Is. 1. P. 51–75. DOI:10.1007/s10479-011-0942-z
77. Hifi M. Approximate algorithms for the container loading problem // International Transactions in Operational Research. 2002. Vol. 9. Is. 6. P. 747–774. DOI:10.1111/1475-3995.00386
78. Christensen S.G., Rousøe D.M. Container loading with multi-drop constraints // International Transactions in Operational Research. 2009. Vol. 16. Is. 6. P. 727– 743. DOI:10.1111/j.1475-3995.2009.00714.x
79. Kang K., Moon I., Wang H. A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem // Applied Mathematics and Computation. 2012. Vol. 219. Is. 3. P. 1287–1299. DOI:10.1016/j.amc.2012.07.036
80. Burke E.K., Hyde M.R., Kendall G., Woodward J. Automating the packing heuristic design process with genetic programming // Evolutionary Computation. 2012. Vol. Is. 1. 20. P. 63–89. DOI: 10.1162/EVCO_a_00044
81. Chien C.F., Wu W.T. A framework of modularized heuristics for determining the container loading patterns // Computers & Industrial Engineering. 1999. Vol. 37. Is. 1–2. P. 339–342. DOI:10.1016/S0360-8352(99)00088-1
82. Liu W.Y., Lin C.C., Yu C.S. On the three-dimensional container packing problem under home delivery service // Asia-Pacific Journal of Operational Research. 2011. Vol. 28. Is. 5. P. 601–621. DOI: 10.1142/S0217595911003466
83. Lim A., Zhang X. The container loading problem // Proceedings of the 2005 ACM Symposium on Applied Computing (SAC). (Santa Fe, New Mexico, USA, March 13-17, 2005). ACM New York, NY, USA. 2005. P. 913–917. DOI: 10.1145/1066677.1066888
84. Lin J.L., Foote B., Pulat S., Chang C.H., Cheung J.Y. Hybrid genetic algorithm for container packing in three dimensions // Proceedings of the Ninth Conference on Artificial Intelligence for Applications. (1-5 Mar, Orlando, FL, 1993). IEEE Computer Society Press, Orlando. 1993. P. 353–359. DOI: 10.1109/CAIA.1993.366589
85. Yeh J.M., Lin Y.C., Yi S. Applying genetic algorithms and neural networks to the container loading problem // Journal of Information and Optimization Sciences. 2003. Vol. 24. Is. 3. 423–443. DOI: 10.1080/02522667.2003.10699576
86. Liang S.C., Lee C.Y., Huang S.W. A hybrid meta-heuristic for the container loading problem // Communications of the International Information Management Association. 2007. Vol. 7. Is. 4. P. 73–84.

87. Hasni H., Sabri H. On a hybrid genetic algorithm for solving the container loading problem with no orientation constraints // Journal of Mathematical Modelling and Algorithms in Operations Research. 2013. Vol. 12. Is. 1. P. 67 – 84. DOI:10.1007/s10852-012-9179-3
88. Fanslau T., Bortfeldt A. A tree search algorithm for solving the container loading problem // INFORMS Journal on Computing. 2010. Vol. 22. Is. 2. P. 222 – 235.
89. Fekete S.P., Schepers J., van der Veen J.C. An exact algorithm for higher-dimensional orthogonal packing // Operations Research. 2007. Vol. 55. Is. 3. P. 569– 587.
90. Junqueira L., Morabito R., Yamashita D.S. Three-dimensional container loading models with cargo stability and load bearing constraints // Computers & Operations Research. 2012. Vol. 39. Is.1. P. 74–85. DOI:10.1016/j.cor.2010.07.017
91. De Castro Silva J.L., Soma N.Y., Maculan N. A greedy search for the three-dimensional bin packing problem: the packing static stability case // International Transactions in Operational Research. 2003. Vol. 10. Is. 2. P. 141–153. DOI:10.1111/1475-3995.00400
92. Thapatsuwan P., Pongcharoen P., Hicks C., Chainate W. Development of a stochastic optimisation tool for solving the multiple container packing problems // International Journal of Production Economics. 2012. Vol. 140. Is. 2. P. 737–748. DOI:10.1016/j.ijpe.2011.05.012
93. De Almeida A., Figueiredo M.B. A particular approach for the three-dimensional packing problem with additional constraints // Computers & Operations Research. 2010. Vol. 37. Is.11. P. 1968–1976. DOI:10.1016/j.cor.2010.01.010
94. Soak S.M., Lee S.W., Yeo G.T., Jeon M.G. An effective evolutionary algorithm for the multiple container packing problem // Progress in Natural Science. 2008. Vol. 18. Is. 3. P. 337–344. DOI:10.1016/j.pnsc.2007.11.007
95. Kang M.K., Jang C.S., Yoon K.S. Heuristics with a new block strategy for the single and multiple containers loading problems // Journal of the Operational Research Society. 2010. Vol. 61. Is. 1. P. 95–107. DOI: 10.1057/jors.2008.120
96. Sciomachen A., Tanfani E. A 3D-BPP approach for optimising stowage plans and terminal productivity // European Journal of Operational Research. 2007. Vol. 183. Is. 3. P. 1433–1446. DOI:10.1016/j.ejor.2005.11.067
97. Miyazawa F.K., Wakabayashi Y. Three-dimensional packings with rotations // Computers & Operations Research. 2009. Vol. 36. Is. 10. P. 2801–2815. DOI:10.1016/j.cor.2008.12.015
98. Amossen R.R., Pisinger D. Multi-dimensional bin packing problems with guillotine constraints // Computers & Operations Research. 2010. Vol. 37. Is. 11. P. 1999–2006. DOI:10.1016/j.cor.2010.01.017
99. Epstein L., Levy M. Dynamic multi-dimensional bin packing // Journal of Discrete Algorithms. 2010. Vol. 8. Is. 4. P. 356–372. DOI:10.1016/j.jda.2010.07.002
100. Lodi A., Martello S., Vigo D. TSpack: a unified tabu search code for multi-dimensional bin packing problems // Annals of Operations Research. 2004. Vol. 131. Is. 1-4. P. 203–213. DOI: 10.1023/B:ANOR.0000039519.03572.08

101. Boschetti M.A. New lower bounds for the three-dimensional finite bin packing problem // Discrete Applied Mathematics. 2004. Vol. 140. Is. 1-3. P. 241–258. DOI:10.1016/j.dam.2003.08.004
102. Brunetta L., Gr´egoire P. A general purpose algorithm for three-dimensional packing // INFORMS Journal on Computing. 2005. Vol. 17. Is. 3. P. 328–338.
103. Ceschia S., Schaerf A. Local search for a multi-drop multi-container loading problem // Journal of Heuristics. 2013. Vol. 19. Is. 2. P. 275–294. DOI:10.1007/s10732-011-9162-6