

Обратная задача об упаковке в контейнеры при наличии качественных критериев¹

Аннотация. Предлагается новая постановка обратной задачи об упаковке в контейнеры, особенность которой заключается в учете предпочтений лица принимающего решение на множестве объектов, оцененных по многим качественным критериям. Обсуждаются аспекты этой задачи, относящиеся к теории принятия решений при многих критериях. Приводится обзор известных методов решения классической задачи об упаковке в контейнеры и ее обратной задачи (включая многокритериальную задачу).

Ключевые слова: обратная задача об упаковке в контейнеры, отношение предпочтения, порядковая классификация, приближенные алгоритмы, метод ветвей и границ, генетические алгоритмы.

Введение

В различных прикладных областях, таких как, помимо прочего, логистика, системы связи, многопроцессорная обработка, планирование и управление проектами, возникают задачи, которые можно, в общем случае, сформулировать следующим образом:

1) имеется набор объектов (например, грузов, задач, проектов и т.п.), качество которых оценивается по многим критериям;

2) известна потребность каждого объекта в некотором ресурсе (например, грузоподъемности, времени исполнения, затратах и т.п.);

3) такой ресурс может быть неделимым или распределенным на несколько частей (например, на ряд лет, между исполнителями и т.п.).

Требуется:

- упорядочить объекты по качеству в соответствии с предпочтениями лица, принимающего решения (ЛПР);

- выбрать максимальное число объектов с учетом их предпочтительности так, чтобы не нарушалось общее ограничение на ресурс, если он неделимый;

- выбрать максимальное число объектов с учетом их предпочтительности и сгруппировать их по числу частей, на которые разделен ресурс, так, чтобы в каждой группе потребность объектов в ресурсе не превышала его соответствующую часть.

При этом в обоих случаях учет предпочтительности означает, что для каждого невыбранного объекта не должно быть выбранных объектов, менее предпочтительных для ЛПР, вместо которых такой объект мог бы быть выбран без нарушения ограничения на ресурс.

Если отбросить требование учета предпочтительности объектов, такая задача, в первом случае, соответствует задаче о рюкзаке, а во втором – одномерной многокритериальной задаче [1], с единичными коэффициентами в целевой функции и в том, и в другом случае. В свою очередь, такая одномерная многокритериальная задача соответствует задаче об упаковке в контейнеры, в которой, в отличие от ее классического варианта [2], требуется упаковать не все объекты в минимальное число контейнеров, а максимальное число объектов в имеющиеся контейнеры [3, 4]. Задача об упаковке в такой постановке привлекала относительно неболь-

¹Работа частично поддержана РФФИ (проект 16-07-00092).

шое внимание по сравнению с ее классическим вариантом, и в отношении ее названия в литературе нет однозначного мнения. Иногда ее называют обратной задачей об упаковке в контейнеры, а иногда – двойственной.

Требование учета предпочтительности объектов для ЛПР означает, что на множестве объектов должно быть построено бинарное отношение предпочтения в форме строгого, в идеале, или частичного, в общем случае, порядка, и объекты должны упаковываться именно в таком порядке. Такое отношение также можно интерпретировать как отношение предшествования: для любой пары объектов менее предпочтительный объект не должен упаковываться, пока не будет упакован более предпочтительный объект (если позволяет ресурс). В первых публикациях, посвященных задаче об упаковке с отношением предшествования, заданным на объектах [5-7], рассматривался только ее классический вариант (минимизация числа контейнеров) как разновидность задачи о расписании. То же самое относится и к более поздним публикациям в этой области (например, [8]). Кроме того, во всех таких работах соблюдение отношения предшествования означает, что для любой пары объектов, находящихся в отношении предшествования (первый объект должен быть упакован раньше второго), контейнер, в который помещается первый объект, должен иметь не больший номер, чем контейнер для второго объекта. В нашем случае такое соответствие номеров объектов и контейнеров не требуется: более предпочтительный объект может быть упакован в контейнер с большим номером, чем менее предпочтительный. Но попытка упаковки менее предпочтительного объекта может предприниматься только тогда, когда для более предпочтительного объекта не хватает ресурса ни в одном контейнере. Задача об упаковке, в которой требуется упаковать максимальное число объектов, характеризующихся оценками по многим качественным критериям, с учетом предпочтений ЛПР, рассматривалась только в [9].

Задача об упаковке в контейнеры начала активно исследоваться в семидесятые годы прошлого века в следующей «статической» («офлайновой») постановке: имеется набор объектов с известными весами и неограниченное число контейнеров одинаковой грузоподъемности. Требуется упаковать все объекты в

минимальное число контейнеров так, чтобы не нарушалось ограничение на грузоподъемность каждого контейнера. В [10, 11] доказывалось, что даже этот простейший одномерный вариант задачи является NP-трудным, на основании чего делается вывод, что для ее решения построение точного эффективного алгоритма маловероятно.

В получившей широкую известность работе [12] представлена типология задач об упаковке в контейнеры в зависимости от их характеристик. Относительно недавно эта типология была уточнена в [13], где рассматриваются, помимо прочего, следующие характеристики:

1) *число контейнеров*: один или несколько (фиксированное или неопределенное числа);

2) *размерность объектов*, подлежащих упаковке, и, соответственно, контейнеров. Задачам двух- и трехмерной упаковки посвящено множество публикаций (например, [14-27]). Известны также исследования задач упаковки размерности больше трех (например, [28]);

3) *цель упаковки*. Этот критерий используется для определения двух базовых ситуаций: минимизация числа контейнеров для упаковки всех объектов и максимизация числа объектов (или их «ценности», «полезности» и т.п.), которые можно упаковать в имеющиеся контейнеры. Во втором случае предполагается, что имеющихся контейнеров недостаточно для упаковки всех объектов. И здесь возникает комбинаторная задача выбора либо максимального числа объектов, либо объектов с совокупной максимальной «ценностью» или «полезностью», которые можно упаковать в такие контейнеры. В случае минимизации числа контейнеров, в которые требуется упаковать все объекты, задача выбора на множестве объектов, естественно, не возникает. Понятие «ценность» («полезность») объектов интерпретируется в задачах об упаковке как стоимость, прибыль и т.п. Зачастую полезность объектов считается прямо пропорциональной их размеру, так что целевой функцией является максимизация или минимизация веса или длины (одномерные задачи), площади (двумерные задачи) или объема (трехмерные задачи), если используется такая терминология. Целевая функция также может быть определена как минимизация «отходов» или совокупного размера неиспользованных частей контейнеров. Заметим, что во всех таких случаях предпола-

гается, что ценность (полезность и т.п.) объекта задается скалярно, а не оценивается по многим качественным критериям;

4) *разнообразие контейнеров* (контейнеры имеют одинаковые или разные «габариты»);

5) *разнообразие форм объектов*, подлежащих упаковке (при размерности больше 1).

Кроме того, известны задачи динамической (или онлайнной) упаковки (например, [29-32]), когда объекты поступают один за другим, и требуется упаковать каждый объект без информации о том, какими будут другие объекты. При этом на каждом шаге онлайнного алгоритма при появлении нового объекта произведенная до этого упаковка является «безотзывной», т.е. уже упакованный объект нельзя вынуть из одного контейнера и переместить в другой.

Основные проблемы, которые возникают при решении описанной выше обратной задачи об упаковке в контейнеры с учетом предпочтений ЛПП, заключаются в следующем:

1) определение набора качественных критериев для оценки объектов и, собственно, оценка объектов по таким критериям (структуризация);

2) выявление предпочтений ЛПП на множестве многокритериальных объектов для построения на них отношения (частичного) порядка;

3) упаковка максимального числа наиболее предпочтительных объектов в заданное число контейнеров с учетом ограничения на ресурс каждого контейнера (с допаковкой контейнеров менее предпочтительными объектами, также в порядке их предпочтительности, когда для них остается ресурс, а более предпочтительные объекты упаковать не удастся).

Первая и вторая проблемы относятся к теории принятия решений при многих критериях. Эти аспекты кратко рассматриваются в Разделе 1, наряду с формальной постановкой обратной задачи об упаковке при наличии многих качественных критериев. В последующих разделах приводится обзор известных подходов к решению одномерной статической задачи об упаковке в контейнеры и ее обратной (двойственной) задачи, включая многорюкзачную. Предлагаемый обзор не является исчерпывающим, поскольку существует множество направлений исследования этих задач, и число посвященных им публикаций постоянно растет (например, в [33] приводятся сведения о числе опубликованных в западных журналах статей по этой тематике по состоянию на 2014 г.).

1. Формальная постановка задачи, ее структуризация и выявление предпочтений ЛПП

1.1. Формальная постановка задачи

Введем следующие – обозначения:

N – число объектов;

w_i – вес i -го объекта, $i = \overline{1, N}$

$\overline{a_i} = (a_{i1}, a_{i2}, \dots, a_{iK})$ – векторная оценка i -го объекта по K критериям качества, где a_{ik} – оценка i -го объекта по k -му критерию, $k = \overline{1, K}$;

$A = (\overline{a_1}, \overline{a_2}, \dots, \overline{a_N})$ – набор векторных оценок объектов;

$R = \{(\overline{a_i}, \overline{a_j}) \in Ax A\}$ – бинарное отношение предпочтения на множестве векторных оценок объектов: $(\overline{a_i}, \overline{a_j}) \in R$, если объект с номером i не менее предпочтителен для ЛПП, чем объект с номером j ;

M – число контейнеров;

C – грузоподъемность каждого контейнера.

Введем двоичные переменные:

$$x_{im} = \begin{cases} 1, & \text{если } i\text{-й объект упаковывается в } m\text{-й контейнер} \\ 0, & \text{в противном случае} \end{cases}$$

$$y_m = \begin{cases} 1, & \text{если } m\text{-й контейнер используется} \\ 0, & \text{в противном случае} \end{cases}$$

$$(i = \overline{1, N}, m = \overline{1, M}).$$

$$X_i = \max_{m=1, M} (x_{im}), i = \overline{1, N}. \quad \text{Соответственно,}$$

$X_i = 1$, если i -й объект упакован, и 0, в противном случае;

$I_j^+ = \{i | (\overline{a_i}, \overline{a_j}) \in R, X_i = 1\}$ – множество номеров упакованных объектов, не менее предпочтительных для ЛПП, чем объект с номером j ;

$\Delta_m = C - \sum_{i \in I_j^+} w_i x_{im}$ – остаток грузоподъемности в j -м контейнере после упаковки в него объектов с номерами из $I_j^+, 1 \leq j \leq N$. Исходно

$$\Delta_m = C, m = \overline{1, M}.$$

Тогда задача может быть поставлена следующим образом:

$$\max \sum_{i=1}^N X_i \quad (1)$$

$$\sum_{i=1}^N w_i x_{im} \leq Cy_m, \forall m = \overline{1, M} \quad (2)$$

$$\sum_{m=1}^M x_{im} \leq 1, i = \overline{1, N} \quad (3)$$

$$\forall i, j | i = \overline{1, N}, j = \overline{1, N}, i \neq j, (\overline{a_i}, \overline{a_j}) \in R:$$

$$X_i \geq X_j, \text{ если } \exists m, 1 \leq m \leq M \text{ такое, что}$$

$$\Delta_m \geq w_i, \text{ и } X_j \geq X_i, \text{ в противном случае. (4)}$$

Целевая функция (1) означает, что требуется упаковать максимальное число объектов в имеющиеся контейнеры. Здесь используется определенная выше переменная X_i , поскольку неважно, в какой контейнер будет упакован каждый объект.

Ограничение (2) означает, что сумма весов объектов, упакованных в каждый контейнер, не должна превышать грузоподъемность такого контейнера.

Ограничение (3) означает, что каждый объект может быть упакован только в один контейнер.

Ограничение (4) означает, что для любой пары объектов, в которой первый объект не менее предпочтителен, чем второй, но для него не хватает остаточной грузоподъемности ни в одном контейнере. М может быть предпринята попытка упаковки второго объекта.

1.2. Структуризация задачи и выявление предпочтений ЛПР

Согласно [34], задачи принятия решений при многих критериях, такие как задачи выбора, упорядочения или порядковой классификации (многокритериальной сортировки), являются слабоструктурированными или даже не структурированными, поскольку, помимо прочего, основным источником информации для их решения являются суждения (предпочтения) человека, которые в значительной степени субъективны. Структуризация, как отмечается в [35], должна быть первым этапом в описании задачи и ее решении, поскольку именно она дает понимание соответствующих элементов задачи, таких как критерии, их возможные значения и, в случае задачи классификации, наименования классов. Кроме того, на этапе структуризации необходимо упорядочить шка-

лы критериев (их значения) и, в случае задачи классификации, задать порядок на классах в соответствии с предпочтениями ЛПР.

Структуризация задач принятия решения не свойственна повседневной деятельности ЛПР (чего нельзя сказать, например, о задаче выбора, которая, тем не менее, при наличии многих критериев, является, сама по себе, весьма сложной), и, следовательно, необходимы подходы, помогающие ЛПР в ее выполнении. Однако, как ни странно, существует лишь небольшое число публикаций, посвященных этой проблеме. В [36] структуризация задач упорядочения, выбора или классификации названа наиболее сложным этапом их решения. И отмечается, что отсутствие надежной методологии для формальной структуризации делает этот этап чем-то сродни искусству, основанному на интуиции и «мастерстве» ЛПР. Хотя статья [36] опубликована более 30 лет назад, ситуация изменилась незначительно.

В обзоре [37], посвященном задачам и методам принятия решений при многих критериях, делается вывод о том, что структуризация таких задач по-прежнему представляет существенный интерес. В [38, 39] предложен формализованный подход к структуризации задач номинальной классификации (классы не упорядочены) при многих критериях (называемых там признаками, поскольку, помимо прочего, классификация осуществляется не в соответствии с предпочтениями ЛПР, а в соответствии со знаниями эксперта). Вкратце, суть этого подхода (в терминах многопризнаковой задачи классификации, основанной на знаниях) заключается в следующем. Эксперт сначала перечисляет наименования классов, на которые должны быть распределены объекты, а затем либо прямо перечисляет признаки и их значения, либо для каждого класса приводит пример принадлежащего к нему объекта в форме правила «если..., то...», после чего у него выясняется, значения каких признаков отражены в данном примере. После этого для каждого значения каждого признака эксперт должен указать, к каким еще классам может относиться объект, имеющий такое значение данного признака, и какие значения этого признака могут иметь объекты, относящиеся к каждому из других классов. Очевидно, что при использовании такого подхода нельзя гарантировать, что будут получены полные перечни наименований клас-



Рис.1. Общая схема решения обратной задачи об упаковке в контейнеры при наличии многих критериев

сов, признаков и их значений, в связи с чем на этапе выявления правил классификации должна быть предусмотрена возможность расширения структуры задачи [38].

В [40] показано, как этот подход применим для структуризации задачи порядковой классификации в целях выявления наименований критериев, их значений и классов. Его также можно применить, с соответствующими необходимыми изменениями, и для многокритериальной задачи упорядочения/выбора объектов в соответствии с предпочтениями ЛПР.

Когда набор необходимых критериев, набор вербальных значений по каждому критерию и, в случае задачи порядковой классификации, наименования классов будут сформированы (и упорядочены), ЛПР должен:

1) упорядочить значения (шкалы) каждого критерия;

2) оценить объекты, подлежащие упаковке, по таким критериям, то есть поставить в соответствие каждому объекту многокритериальную векторную оценку;

3) разделить их на упорядоченные классы и, если все объекты очередного по предпочтительности класса не могут быть упакованы вместе с объектами более предпочтительных классов, упорядочить объекты такого класса в соответствии с предпочтениями ЛПР.

Для выявления предпочтений ЛПР с учетом многих критериев в целях упорядочения объектов и/или построения их порядковой классификации существует множество подходов и методов. Их обзор выходит за рамки настоящей статьи. Заметим, однако, что одним из перспективных направлений, позволяющих эффективно и надежно (с психологической точки зрения) решить эту задачу, является Вербальный анализ решений [41, 42], в рамках которого разработаны методы полной и согласованной порядковой классификации (например, [40]) и построения частичного упорядочения многокритериальных объектов [43].

Общая схема решения поставленной здесь задачи показана на Рис.1.

2. Приближенные алгоритмы для решения задачи об упаковке в контейнеры и ее обратной задачи

Как уже отмечалось, классическая задача об упаковке в контейнеры заключается в минимизации числа контейнеров известной грузоподъемности, в которые могут быть упакованы все объекты из их заданного списка так, чтобы сумма весов объектов, упакованных в каждый контейнер, не превышала его грузоподъем-

ность. В 70-е годы прошлого века для этой задачи были предложены эффективные приближенные алгоритмы, наиболее известными из которых являются «в первый подходящий» (First Fit-FF), «в наилучший из подходящих» (Best Fit-BF), «в первый подходящий с убыванием» (First Fit Decreasing-FFD) и «в наилучший из подходящих с убыванием» (Best Fit Decreasing-BFD) [2, 44-47]. Во всех этих алгоритмах контейнеры, которые исходно пусты, последовательно нумеруются, начиная с первого. В алгоритмах FF и BF объектам также присваиваются последовательные номера в зависимости от их положения в исходном списке. Алгоритм FF помещает очередной объект в первый из контейнеров, где для него хватает грузоподъемности, а алгоритм BF – в первый из контейнеров, в котором для него не только хватает грузоподъемности, но и ее остаток после его упаковки будет минимальным. В [48, 49] доказывалось, что асимптотический коэффициент приближения алгоритмов упаковки FF и BF равен 1,7, т.е. число определяемых ими контейнеров в худшем случае не отличается от оптимального более чем на 70%. Заметим, что в [50] доказывалось, что для алгоритма BF коэффициент 1,7 является абсолютным коэффициентом приближения.

Алгоритмы FFD и BFD имеют несколько лучшие оценки. Они отличаются от алгоритмов FF и BF тем, что объекты предварительно упорядочиваются по невозрастанию их весов. В [2, 51] доказывалось, что число контейнеров, получаемое в результате применения этих алгоритмов, в худшем случае не больше $11/9 OPT + 4$, где OPT – минимальное число контейнеров, которое можно было бы найти с помощью точного алгоритма. Дальнейшие модификации алгоритмов FFD и BFD предложены, например, в [52]. В них производится предварительная сортировка объектов на группы в соответствии с их размерами и «подходимостью» друг к другу, чтобы наиболее полно использовать грузоподъемность каждого контейнера. Однако ни для одной из этих модификаций пока не получена принципиально лучшая оценка. В [2] отмечается, что поведение перечисленных выше алгоритмов «в среднем» существенно отличается от полученных оценок, что, по-видимому, стало причиной появления статей, посвященных вероятностному исследованию этой задачи (например, [53], где показано, что вероятность нахождения оптимального решения с приме-

нением описанных там приближенных алгоритмов достаточно велика).

Задача упаковки максимального числа объектов в заданное число контейнеров была впервые поставлена в [3]. Там отмечается, что любой алгоритм, имеющий в наихудшем случае разумную оценку относительно оптимального решения, должен пытаться упаковать максимальное число объектов, начиная с самых легких. С такой целью сначала предлагается использовать правило «самый легкий объект первым» (Smallest Piece First-SPF). Согласно этому правилу, очередной объект из списка объектов, упорядоченных по неубыванию их весов, помещается в контейнер наименьшей грузоподъемности, оставшейся после упаковки предыдущих объектов, если его вес не превышает такую остаточную грузоподъемность. Когда очередной объект не помещается ни в один из контейнеров, работа алгоритма прекращается.

В [3] предлагается еще один алгоритм для этой задачи, а именно: «в первый подходящий с возрастанием» (First Fit Increasing-FFI), также основанный на упорядочении объектов по неубыванию, но имеющий лучшую оценку, чем алгоритм SPF. Алгоритм FFI помещает очередной объект, начиная с первого, в контейнер с меньшим номером, в котором для него хватает грузоподъемности. Доказывалось, что в худшем случае максимальное число объектов, которое может быть упаковано этим алгоритмом не меньше $3/4$ от числа объектов в оптимальной упаковке, а его трудоемкость оценивается как $O(n \log_2 n)$, где n – число объектов в исходном списке.

В [3] также обсуждается возможность многократного применения к этой задаче алгоритма FFD с последовательным отбрасыванием самых тяжелых объектов. В [4] этот алгоритм (названный там IFFD от английского Iterated First Fit Decreasing – «итеративный алгоритм «в первый подходящий с убыванием»») детально исследуется и доказывалось, что в худшем случае он позволяет упаковать не меньше $6/7$ объектов, чем в оптимальной упаковке. Однако этот алгоритм имеет большую трудоемкость, чем FFI: $O(n \log_2 n + mn \log_2 m)$, где n – число объектов в исходном списке, а m – число контейнеров.

В [9] алгоритм IFFD был дополнен попыткой доупаковки отброшенных объектов в последний контейнер и показано, что такая попытка может оказаться удачной.

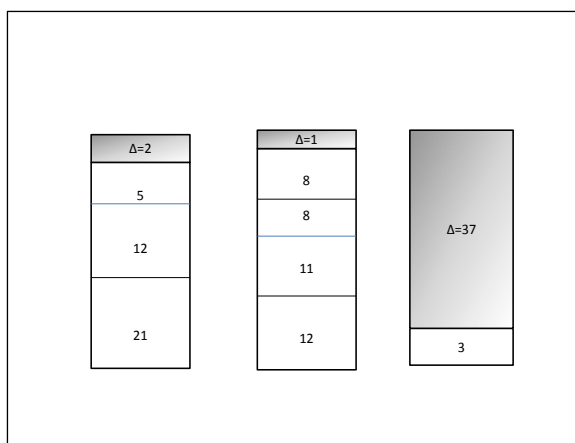


Рис. 2. Первое применение алгоритма IFFD

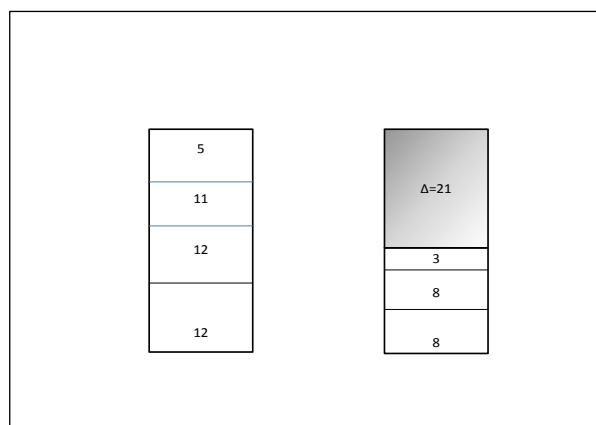


Рис. 3. Второе применение алгоритма FFD

Рассмотрим следующий пример. Имеется два контейнера грузоподъемностью по 40 каждый и восемь объектов с весами: $w_1 = 3$, $w_2 = 8$, $w_3 = 8$, $w_4 = 5$, $w_5 = 11$, $w_6 = 12$, $w_7 = 12$, $w_8 = 21$. Упорядочим эти объекты по невозрастанию их весов и применим для их упаковки алгоритм IFFD. Сначала предположим, что имеется неограниченное число контейнеров грузоподъемности, равной 40, и упакуем их алгоритмом FFD.

Как видно на Рис. 2, для упаковки всех объектов потребовалось 3 таких контейнера. Отбросим самый тяжелый объект с весом $w_1 = 21$ и снова применим алгоритм FFD к оставшимся контейнерам. Как видно на Рис. 3, во втором контейнере осталось достаточно места для отброшенного объекта (с весом, равным 21) и, следовательно, он может быть туда упакован.

3. Точные алгоритмы для основной и обратной задач упаковки в контейнеры

В [54] приводится обзор и сравнение методов линейного программирования для решения классической задачи об упаковке в контейнеры. Первый из таких алгоритмов был предложен в [55]. Он представляет собой простой метод ветвей и границ с поиском в глубину и стратегией ветвления, согласно которой объекты для упаковки выбираются в порядке невозрастания их весов. Значение целевой функции на частичных решениях оценивается снизу в предположении делимости объектов.

Самым известным из таких алгоритмов является МТР [56, 57], основанный на методе ветвей и границ. Его суть заключается в следующем: объекты упорядочиваются по невозрастанию их весов и очередной объект помещается в каждый частично заполненный контейнер, где для него хватает грузоподъемности, и в новый, пустой, контейнер, в результате чего образуется ветвление по этим альтернативным вариантам. Это приводит к тому, что область решения задачи ограничивается $n!$, где n – число объектов, но это весьма пессимистическая верхняя граница, поскольку многие объекты не помещаются в частично заполненные контейнеры.

В каждом узле дерева поиска алгоритм МТР вычисляет пополнение соответствующего частичного решения методами FF, BF и WFD (Worst Fit Decreasing – «в наихудший из подходящих в порядке убывания»): очередной объект помещается в контейнер, в котором ему хватает грузоподъемности, а после его упаковки остаток грузоподъемности будет максимальным). «Частичное решение» означает, что часть объектов уже окончательно упакована. Пополнение частичного решения осуществляется путем использования частично заполненных на данный момент контейнеров и упаковке оставшихся неупакованными объектов в такие контейнеры.

Каждое из этих приближенных решений сравнивается с нижней границей. Если число контейнеров, используемых любым из приближенных решений равно нижней границе, дальнейший поиск ниже данного узла не производится. Если число контейнеров в любом приближенном решении равно нижней границе

исходной задачи, работа алгоритма прекращается, поскольку найденное решение является оптимальным.

Очевидной нижней границей является сумма весов всех объектов, деленная на грузоподъемность контейнера и округленная до ближайшего большего целого. Такая граница в [56, 57] названа границей L1. Лучшая граница получается, если к сумме весов объектов прибавить оценку совокупной грузоподъемности контейнеров, которая останется неиспользованной в любом решении, и полученный результат разделить на грузоподъемность контейнера. В [56, 57] такая граница называется границей L2. В [58] приводится более простой и более интуитивный алгоритм для расчета такой границы. Обзор существующих на данный момент способов определения нижних границ для задачи об упаковке в контейнеры приводится в [53].

Эффективность алгоритма МТР обеспечивается благодаря так называемому «отношению доминирования», общая форма которого предложена в [56, 57]. Для этого вводится понятие «допустимого набора», который определяется как набор объектов, сумма весов которых не превышает грузоподъемность контейнера. Пусть A и B - два допустимых набора. Если набор B можно разбить на подгруппы $B_1, B_2, \dots, B_l$ так, что в наборе A найдется l объектов, таких что вес каждого из них не меньше суммы весов объектов набора B_j ($j = \overline{1, l}$), набор A доминирует набор B . Если все допустимые наборы содержат общий объект x , то для упаковки в контейнер, содержащий x , имеет смысл рассматривать только недоминируемые наборы объектов. В [56, 57] показывается, как использовать отношение доминирования для уменьшения размера подзадач, формируемых в процессе поиска решения.

В [59] описывается точный метод BISON, который использует несколько границ, процедуры уменьшения размера подзадач, эвристики, а также метод ветвей и границ, основанный на новой процедуре ветвления. В [60] предлагается точный алгоритм, основанный на генерации столбцов и методе ветвей и границ.

В [61] приводится обзор точных алгоритмов для решения задачи об упаковке в контейнеры, многорюкзачной задачи и задачи покрытия контейнеров (последняя, называемая там двойственной задачей об упаковке, заключается в

максимизации числа контейнеров, содержащих наборы объектов, суммы весов которых не меньше некоторой константы). Там же предлагается новый алгоритм для задачи об упаковке в контейнеры – алгоритм заполнения контейнеров (Bin Completion-BC), который, в отличие от существующих точных алгоритмов, называемых там «ориентированными на объекты», является «ориентированным на контейнеры». Это означает, что вместо ветвления по числу различных возможных упаковок в контейнеры ветвление осуществляется по числу недоминируемых способов, которыми каждый контейнер может быть заполнен. Утверждается, что алгоритм BC работает асимптотически быстрее, чем алгоритм МТР, и, в частности, более чем в тысячу раз быстрее, чем МТР для задач, в которых требуется упаковать не более 60 объектов. Как отмечается, дополнительным преимуществом этого алгоритма является то, что его гораздо проще и легче реализовать, чем алгоритм МТР. В [62] описываются три способа улучшения алгоритма BC: новый порядок ветвления; последовательная генерация и запоминание в буфере вариантов заполнения, в результате чего избегается генерация экспоненциального числа заполнений, и алгоритм LDS (от Limited Discrepancy Search - «поиск с ограниченными отклонениями»), который изменяет порядок поиска. Утверждается, что только первые два способа улучшения увеличивают скорость нахождения оптимального решения на пять порядков по сравнению с первоначальной версией алгоритма BC.

Как отмечалось во Введении, обратная задача упаковки в контейнеры может быть поставлена как многорюкзачная задача с единичными прибылями. Для классического варианта этой задачи, в которой объекты характеризуются не только весами, но и прибылями, и требуется упаковать в имеющиеся контейнеры (рюкзаки) объекты, обеспечивающие максимальную совокупную прибыль и не нарушающие ограничение на грузоподъемность каждого контейнера, известен алгоритм МТМ [63], который заключается в следующем. Объекты упорядочиваются по невозрастанию отношений прибыли к весу, и каждый объект, выбираемый в таком порядке, помещается как минимум в один контейнер с использованием «жадной» (greedy) процедуры «граница–и–граница» (bound-and-bound). Ветвление осуществляется так, что вы-

бренный объект помещается в контейнеры в порядке неубывания их остаточной грузоподъемности. В каждом узле верхняя граница вычисляется с использованием некоторой релаксации многорюкзачной задачи, которая получается путем объединения всех оставшихся контейнеров в один контейнер с грузоподъемностью, равной совокупной остаточной грузоподъемности в исходной задаче. В результате получается стандартная 0-1 задача о рюкзаке. Такая задача может быть решена любым из известных алгоритмов, а ее оптимальное решение и будет верхней границей для исходной многорюкзачной задачи.

Процедура «граница–и–граница», используемая в алгоритме МТМ, заключается в следующем. С помощью быстрого эвристического алгоритма проверяется, существует ли решение, величина которого равна верхней границе. Если такое решение можно найти, то это означает, что решение, полученное в текущем узле, является оптимальным решением, и можно вернуться обратно. Если же такое решение не найдено, то поиск продолжается под этим узлом. «Жадная» эвристика, используемая в алгоритме МТМ, заключается в решении 0-1 задач о рюкзаке для каждого оставшегося контейнера. Сначала первый контейнер заполняется оптимально с использованием оставшихся объектов, и такие объекты исключаются. Затем второй контейнер заполняется с использованием оставшихся объектов и т.д. Эта процедура повторяется для каждого оставшегося контейнера.

В [64] предлагается точный алгоритм Mulknap для многорюкзачной задачи. Как и МТМ, Mulknap является «ориентированным на объекты» алгоритмом ветвей и границ, который использует верхнюю границу, рассчитанную посредством замещения исходной многорюкзачной задачи 0-1 задачей о рюкзаке, и процедуру «граница–и–граница». Алгоритм Mulknap отличается от алгоритма МТМ тем, что в нем используется иная стратегия подтверждения верхней границы, основанная на разделении решения замещающей однорюкзачной задачи по оставшимся незаполненным контейнерам, объединение объектов и усиление ограничения на грузоподъемность в каждом узле. Приведенные в [64] результаты сравнения алгоритмов Mulknap и МТМ свидетельствуют о том, что для большого числа объектов алгоритм

Mulknap находит оптимальное решение намного быстрее, чем алгоритм МТМ.

В [61] предлагается алгоритм заполнения ВС для многорюкзачной задачи. Он основан на методе ветвей и границ и использует поиск в глубину. Каждый узел в дереве поиска представляет собой максимальную допустимую упаковку в соответствующий контейнер. В каждом узле для оставшейся подзадачи верхняя граница вычисляется путем решения замещающей 0-1 задачи о рюкзаке с использованием алгоритма, основанного на методе ветвей и границ алгоритма нахождения верхней границы для такой задачи [65]. Для возможного уменьшения остаточной задачи применяется процедура редуцирования R2 [65]. Затем выбирается контейнер с наименьшей остаточной грузоподъемностью, и для него находятся недоминируемые упаковки.

Следует заметить, однако, что все эти методы хорошо работают на задачах небольшой размерности, а в случае очень большого числа объектов требуют значительных затрат времени на вычисления.

4. Генетические алгоритмы

В конце восьмидесятых годов прошлого века началась разработка генетических алгоритмов для задачи упаковки в контейнеры (например, [66-68]). В [68] показано, что классический генетический подход неэффективен для определенных типов задач (таких, как задача об упаковке в контейнеры), и предложен генетический алгоритм с группировкой (grouping genetic algorithm), который находит решение за приемлемое время. В [69] этот метод был улучшен за счет использования критерия доминирования [56,57]. Кроме того, там предложен набор эталонных конкретных задач упаковки, который впоследствии был использован многими авторами для тестирования алгоритмов решения этой задачи. В [70] впервые предлагается так называемый гибридный подход, который сочетает в себе генетический алгоритм с вышеперечисленными приближенными алгоритмами. В [71] описывается новый «гиперэвристический» подход, основанный на генетическом алгоритме для нахождения комбинации из четырех известных эффективных приближенных алгоритмов решения одномерных задач упаковки в контейнеры. В качестве таких

приближенных алгоритмов выбраны алгоритмы BFD, NFD (Next Fit Decreasing («в следующий подходящий с убыванием»)), DJD [72] и DJT. Алгоритм NFD помещает объект в текущий контейнер, если не нарушается ограничение на грузоподъемность, или в новый контейнер, который после этого становится текущим. Алгоритм DJD упаковывает объекты, упорядоченные по невозрастанию их весов, в контейнер, пока он не будет полон на одну треть, а затем ищет один, два или три объекта, полностью или хотя бы максимально заполняющие контейнер. Алгоритм DJT аналогичен последнему за тем исключением, что ищется комбинация из не более пяти объектов. После обучения на большом наборе эталонных задач, алгоритм находит оптимальное решение почти для 80% из них, а для других задач получает решение близкое к оптимальному. Доказывается, что этот подход превосходит результаты эвристических алгоритмов, входящих в найденную комбинацию, для 98% задач.

В [73] для рассматриваемой задачи предлагается гиперэвристика, использующая эволюционный алгоритм для формирования наборов атрибутов, которые характеризуют некоторый тип задачи. На этой основе строится классификатор задач, который после обучения выбирает определенную эвристику для новых типов задач. Доказывается, что такой обучаемый классификатор позволяет достичь значительно лучших результатов, чем каждый из используемых приближенных алгоритмов по отдельности (а именно, FFD, DJD и DJT).

Недавно был предложен эффективный генетический алгоритм [74], который обеспечивает перенос лучших генов в хромосомы без потери равновесия между селективным давлением и разнообразием популяции. Такой перенос осуществляется посредством нового набора группирующих генетических операторов, в то время как эволюция балансируется новым алгоритмом воспроизведения, который направляет анализ пространства поиска и предотвращает преждевременную сходимость алгоритма. Представленные результаты вычислительных экспериментов свидетельствуют о том, что этот алгоритм превосходит известные генетические алгоритмы для множества особенно сложных задач упаковки.

Известен также ряд генетических алгоритмов как для задачи о рюкзаке, так и для многорюкочной задачи (например, [75-77]). В [78]

предлагается новый адаптивный алгоритм бинарной дифференциальной эволюции (aBDE) для решения многорюкочной задачи. Особенность этого алгоритма заключается в настройке двух параметров – коэффициента возмущения и коэффициента мутации – посредством линейной процедуры адаптации, которая изменяет их вероятности для каждой генерации. Результаты вычислительных экспериментов, выполненных на 11 вариантах задачи разной степени сложности, свидетельствуют о том, что алгоритм aBDE позволяет получить лучшие решения, чем алгоритм бинарной дифференциальной эволюции BDE [79], стандартный генетический алгоритм и его адаптивная разновидность.

Заключение

Статья носит постановочно-обзорный характер. В ней рассматривается новая постановка обратной задачи об упаковке в контейнеры с учетом предпочтений ЛПП: требуется упорядочить объекты, оцененные по многим качественным критериям, в соответствии с предпочтениями ЛПП и упаковать максимальное число объектов с учетом их предпочтительности в фиксированное число контейнеров без превышения грузоподъемности каждого из них. Учет предпочтительности означает, что для каждого неупакованного объекта не должно быть упакованных объектов, менее предпочтительных для ЛПП, вместо которых он мог бы быть упакован без нарушения ограничения на грузоподъемность контейнера.

Эта задача сочетает в себе аспекты теории принятия решений при многих критериях и комбинаторной оптимизации. В связи с этим, в статье, в первую очередь, изложены основные соображения об упорядочении или порядковой классификации многокритериальных объектов в соответствии с предпочтениями ЛПП. Особо подчеркивается необходимость применения надежных подходов к структуризации задачи в целях выявления набора критериев, их значений и, в случае задачи классификации, наименований классов, а также упорядочения значений критериев и, если применимо, классов в соответствии с предпочтениями ЛПП. Кроме того, в статье содержится обзор известных методов решения классической задачи об упаковке в контейнеры и ее обратной задачи, в том числе в постановке многорюкочной задачи.

Поставленная здесь задача ставилась и решалась только в [9] на основе приближенных алгоритмов. В будущем предполагается использование и сравнение других рассмотренных здесь подходов к ее решению.

Литература

- Martello S., Toth P. (1980). Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research*, 4(4), 276-283.
- Johnson D. S., Demers A., Ullman J. D., Garey M. R., Graham R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4), 299-325.
- Coffman E.G., Jr., Leung J.Y.-T., Ting D. (1978). Bin Packing: Maximizing the Number of Pieces Packed, *Acta Infomat.*, vol. 9, 263-271.
- Coffman Jr, E. G., Leung J. Y. (1979). Combinatorial analysis of an efficient algorithm for processor and storage allocation. *SIAM Journal on Computing*, 8(2), 202-217.
- Graham R. L., Lawler E. L., Lenstra J. K., Kan A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5, 287-326.
- Garey M. R., Graham R. L., Johnson D. S., Yao A. C. C. (1976). Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory, Series A*, 21(3), 257-298.
- Garey M. R., Graham R. L., Johnson D. S. (1978). Performance guarantees for scheduling algorithms. *Operations research*, 26(1), 3-21.
- Dell'Amico M., Diaz J. C. D., Iori M. (2012). The bin packing problem with precedence constraints. *Operations Research*, 60(6), 1491-1504.
- Фуремс Е.М. Модели упаковки в многокритериальных задачах принятия решений при ограниченных ресурсах. Препринт ВНИИСИ, М., 1986. С. 45.
- Karp R.M. (1972). Reducibility among combinatorial problems, in *Complexity of Computer Computations* (R.E. Miller and J.M. Thatcher, eds.), 85-103, Plenum Press.
- Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир. 1982.
- Dyckhoff H., Finke U. (1992). *Cutting and Packing in Production and Distribution: a Typology and Bibliography*, Springer, Berlin.
- Wäscher G., Hausner H., Schumann H. (2007). An improved typology of cutting and packing problems. *EJOR*, 183 (3), 1109-1130.
- Chung F. R., Garey M. R., Johnson D. S. (1982). On packing two-dimensional bins. *SIAM Journal on Algebraic Discrete Methods*, 3(1), 66-76.
- Berkey J. O., Wang P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the operational research society*, 423-429.
- Гимади Э.Х., Залюбовский В.В., Шарыгин П.И. Задача упаковки в полосу: асимптотически точный подход//Известия высших учебных заведений. Математика, № 12, 1997.с.34-44.
- Lodi A., Martello S., Vigo D. (1999). Approximation algorithms for the oriented two-dimensional bin packing problem. *EJOR*, 112(1), 158-166.
- Dell'Amico M., Martello S., Vigo D. (2002). A lower bound for the non-oriented two-dimensional bin packing problem. *Discrete Applied Mathematics* 118(1-2), 13-24.
- Lodi A., Martello S., Vigo D. (2002). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics* 123(1-3), 379-396.
- Lodi A., Martello S., Vigo D. (2002). Heuristic algorithms for the three-dimensional bin packing problem. *EJOR*, 141(2), 410-420.
- Lodi A., Martello S., Vigo D. (2004). TSPack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems. *Annals OR* 131(1-4), 203-213.
- Lodi A., Martello S., Vigo D. (2004). Models and Bounds for Two-Dimensional Level Packing Problems. *J. Comb. Optim.* 8(3), 363-379.
- den Boef E., Korst J., Martello S., Pisinger D., Vigo D. (2005). Erratum to 'The Three-Dimensional Bin Packing Problem': Robot-Packable and Orthogonal Variants of Packing Problems. *Operations Research* 53(4), 735-736.
- Bansal N., Khan A. (2014, January). Improved approximation algorithm for two-dimensional bin packing. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms* (pp. 13-25). SIAM.
- Валеева А.Ф. Применение метаэвристики муравьиной колонии к задачам двухмерной упаковки // Информационные технологии, № 10, 2005. С. 36-43.
- Мухачева Э.А., Валеева А.Ф., Филиппова А.С., Поляковский С.Ю. Задача двумерной контейнерной упаковки: нижние границы и численный эксперимент с алгоритмами локального поиска оптимума // Информационные технологии, № 4, 2006, с.45-52.
- López-Camacho, Eunice, et al. (2013). An effective heuristic for the two-dimensional irregular bin packing problem. *Annals of Operations Research* 206.1: 241-264.
- Lins L., Lins S., Morabito R. (2002). An n-tet graph approach for non-guillotine packings of n-dimensional boxes into an n-container. *EJOR* 141, 421-439.
- Coffman E.G., Jr., Garey M. R., Johnson D. S. (1983). Dynamic bin packing. *SIAM J. Comput.*, 12, 227-258.
- Rhee W.T., Talagrand M. (1993). On-line bin packing with items of random size. *Math. Oper. Res.* 18, 438-445.
- Boyar J., Fayholdt L. M., Larsen K. S., Nielsen M. N. (2001). The competitive ratio for on-line dual bin packing with restricted input sequences, *Nordic J. Comput.*, 8, 463-472.
- Boyar J., et al. (2014). Online bin packing with advice. *Algorithmica*, 1-21.
- Delorme M., Iori M., Martello S. (2015). Bin packing and cutting stock problems: Mathematical models and exact algorithms. In *AIRO 2014 Conference* (p. 77).
- Simon H. (1960). *The New Science of Management Decision*. Harper and Row, New York.
- Saaty T, Shihb H-S (2009), Structures in decision making: On the subjective geometry of hierarchies and networks. *EJOR*, vol. 199, 3:867-872.
- von Winterfeldt D. (1980). Structuring Decision Problems for Decision Analysis. *Acta Psychologica*, vol. 45:71-93.
- Wallenius J., Dyer J., Fishburn P., Steuer R., Zions S., Deb K. (2008). Multiple Criteria Decision Making, Multiattribute Utility Analysis: Recent Accomplishments and What Lies Ahead. *Management Science*, Vol. 54, 7:1336-1349.
- Фуремс Е.М., Структуризация задач классификации, основанных на знаниях// Информационные технологии и вычислительные системы, № 3, 2007.с. 7-17.

39. Furems Eugenia M. (2011). Domain Structuring For Knowledge-Based Multiattribute Classification (A Verbal Decision Analysis Approach), TOP, Springer Berlin / Heidelberg, 19, pp. 402–420.
40. Фуремс Е.М. Многокритериальная порядковая классификация на основе метода STEPCLASS // Искусственный интеллект и принятие решений, № 4, 2012, с. 95-100.
41. Ларичев О.И., Мошкович Е.М. Качественные методы принятия решений. М.: Физматлит. 1996.
42. Ashikhmin I., Furems E., Petrovsky A., Sternin M. (2008). Intelligent DSS Under Verbal Decision Analysis. Encyclopedia of Decision Making and Decision Support Technologies (2 Volumes), Information Science Reference, Hershey • New York, Edited By: Frederic Adam, University College Cork, Ireland; Patrick Humphreys, London School of Economics, UK, Volume 2, pp. 514-527
43. Ashikhmin I., Furems E. (2005). UniCombOS: Intelligent Decision Support System for multi-criteria comparison and choice. Journal of Multi-Criteria Decision Analysis, 13 (2-3), pp.147-157.
44. Гудман С., Хидентниemi С. Введение в разработку и анализ вычислительных алгоритмов. М.: Мир. 19 81. С. 386.
45. Marshal L., Fisher M.S. (1980). Worst-case Analysis of Heuristics Algorithms, Manag. Sci., vol 26, No 1, 1-8.
46. Coffman Jr., Garey M. R., Johnson D. S. (1984). Approximation algorithms for bin-packing—an updated survey. In Algorithm design for computer system design (pp. 49-106). Springer Vienna.
47. Johnson D.S. (1974). Fast algorithms for bin packing. Journal of Computer and System Sciences, vol. 8.3, 272-314.
48. Ullman J. D. (1971). The performance of a memory allocation algorithm. Technical Report 100, Princeton Univ., Princeton, NJ.
49. Garey M. R., Graham R. L., Ullman J. D. (1973), Worst-case analysis of memory allocation algorithms. In Proc. 4th Symp. Theory of Computing (STOC), 143-150, ACM.
50. Dósa G., Sigall J. (2014). Optimal analysis of Best Fit bin packing, Automata, Languages, and Programming. Springer Berlin Heidelberg, 429-441.
51. Coffman Jr., Garey M.R., Johnson D.S. (1978). An Application of Bin Packing to Multiprocessor Scheduling, SIAM J. Comput., vol. 7, No. 1, 1-17.
52. Chi-Chin Yao (1980). New Algorithms for Bin Packing/ Journal of the Association for Computer Machinery, vol 27, No. 2, 207-227.
53. Frederickson G.N. (1980). Probabilistic analysis for simple one- and two-dimensional bin packing algorithms, Information Processing Letters, V. 11, №4-5, 156-161.
54. de Carvalho J.M., Valério (2002). LP models for bin packing and cutting stock problems, EJOR 141/2, 253-273.
55. Eilon S., Christofides N. (1990). The loading problem, Management Science, V. 17, 259267.
56. Martello, S., and Toth, P. (1990), Bin-packing problem. In Knapsack Problems: Algorithms and Computer Implementations. Wiley, chapter 8, 221-245.
57. Martello S., Toth P. (1990). Lower bounds and reduction procedures for the bin packing problem. Discrete Applied Mathematics 28:59-70.
58. Korf R. E. (2002, July). A new algorithm for optimal bin packing. In AAAI/IAAI (pp. 731-736).
59. Scholl, A., Klein, R., Jürgens, C. (1997), BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem, Computers and Operations Research, 24, 627–645.
60. de Carvalho J.M., Valério (1999). Exact solution of bin-packing problems using column generation and branch-and-bound, Annals of Operation Research, 86, 629–659.
61. Fukunaga A.S., Korf R. E. (2007). Bin Completion Algorithms for Multicontainer Packing, Knapsack, and Covering Problems, Journal of Artificial Intelligence Research, 28, 393-429
62. Schreiber E. L., Korf R. E. (2013, August). Improved Bin Completion for Optimal Bin Packing and Number Partitioning. In IJCAI.
63. Martello S., Toth, P. (1981). A bound and bound algorithm for the zero-one multiple knapsack problem. Discrete Applied Mathematics, 3, 275–288.
64. Pisinger D. (1999). An exact algorithm for large multiple knapsack problems. EJOR, 114, 528–541.
65. Martello S., Toth P. (1977). An upper bound for the zero-one knapsack problem and a branch and bound algorithm. EJOR, 1, 169–175.
66. Smith D. (1985). Bin-packing with adaptive search, in: Grefenstette (ed.), Proceedings of an International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum, 202-206.
67. Prosser P. (1988). A hybrid genetic algorithm for pallet loading, in: B. Radig (ed.), ECAI 88 Proceedings 8th European Conference on Artificial Intelligence, Pitman, London, 159-164.
68. Falkenauer E., Delchambre A. (1992). A genetic algorithm for bin-packing and line balancing, Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, Fr, vol. 2, 1186-1192.
69. Falkenauer E.. A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics, 2(1):5–30, 1996.
70. Reeves C. (1996). Hybrid genetic algorithms for bin-packing and related problems, Annals of Operations Research, 63.3, 371-396.
71. Ross P., Marrn-Blazquez J. G., Schulenburg S., Hart E. (2003). Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heuristics. E. Cant.u-Paz et al. (Eds.): GECCO 2003, LNCS 2724, Springer-Verlag Berlin Heidelberg, 1295–1306.
72. Djang P.A., Finch P.R. (1998). Solving one dimensional bin packing problems, Journal of Heuristics, 123-144.
73. Sim K., Hart E., Paechter B. (2012). A Hyper-Heuristic Classifier for One Dimensional Bin Packing Problems: Improving Classification Accuracy by Attribute Evolution. Coello Coello et al. (Eds.): PPSN, Part II, LNCS 7492, 348–357, 2012. Springer-Verlag Berlin Heidelberg.
74. Quiroz-Castellanos, Marcela, et al.(2015). A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Computers & Operations Research 55: 52-64.
75. Khuri S., Bäck T., Heitkötter J. (1994, April). The zero/one multiple knapsack problem and genetic algorithms. In Proceedings of the 1994 ACM symposium on Applied computing (pp. 188-193). ACM.
76. Michalewicz Z., Arabas J. (1994). Genetic algorithms for the 0/1 knapsack problem. In Methodologies for Intelligent Systems (pp. 134-143). Springer Berlin Heidelberg.

77. Cotta C., Troya J. M. (1998). A hybrid genetic algorithm for the 0–1 multiple knapsack problem. In *Artificial neural nets and genetic algorithms* (pp. 250-254). Springer Vienna.
78. André L., Stubs Parpinelli R. (2015). The Multiple Knapsack Problem Approached by a Binary Differential Evolution Algorithm with Adaptive Parameters. *Polibits*, (51), 47-54.
79. Krause J., Parpinelli R. S., Lopes H. S. (2012). Proposta de um algoritmo inspirado em evolucao diferencial aplicado ao problema multidimensional da mochila. *Anais do IX Encontro Nacional de Inteligência Artificial–ENIA*. Curitiba, PR: SBC.

Фуремс Евгения Марковна. Ведущий научный сотрудник ИСА Федерального исследовательского центра «Информатика и управление» РАН. Кандидат технических наук. Автор более 65 научных работ. Область научных интересов: принятие решений при многих критериях, комбинаторная оптимизация, выявление экспертных знаний для создания систем поддержки диагностических решений (приложения в области медицины, техники, геологии, финансов и оценки качества научно-технических переводов). E-mail: fem@mail.ru