ISSN 2658-5014 (Print)
2022. Т. 4, № 2 (9). С. 37–42
СИИТ
СИСТЕМНАЯ ИНЖЕНЕРИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
ISSN 2686-7044 (Online)
http://siit.ugatu.su

УДК 004.02365
doi 10.54708/26585014_2022_42937

Особенности метода поиска с запретами для задачи упаковки

А. Р. Усманова [1], **Ю. И. Валиахметова** [2]

[1] kfmn2004@mail.ru, [2] julikas@inbox.ru

ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)

Поступила в редакцию 17 августа 2022 г.

Аннотация. Рассматривается задача упаковки предметов в контейнеры, являющаяся NP-полной. Авторы исследуют один из популярных метаэвристических методов – поиск с запретами (Tabu Search, TS). Предлагается использовать математическую модель родственной задачи – задачи расписания для параллельных процессоров. Авторами предложен алгоритм, позволяющий свести процесс получения оптимального решения в задаче упаковки к получению допустимого решения в задаче расписания процессоров. Рассмотрены две окрестности соседних решений полиномиальной сложности. В работе обсуждаются способы построения одной из самых существенных составляющих метода поиска с запретами – списка запретов (Tabu List, TL). Предложена модель с динамической длиной списка. Также предложена оценочная функция, имеющая константную сложность вычисления. Приведены некоторые результаты численного эксперимента, демонстрирующего эффективность предложенных подходов.

Ключевые слова: поиск с запретами; эвристический алгоритм; оптимизация; двумерная упаковка.

ВВЕДЕНИЕ

Метод поиска с запретами (TS, Tabu Search) является одним из наиболее эффективных метаэвристических методов. Он был предложен Ф. Гловером в 80-е годы [1].

Отличительной чертой этого метода является процесс введения и снятия некоторых искусственных ограничений задачи в процессе поиска решения [2].

Метод поиска с запретами и его вариации нашли широкое применение в решении разных оптимизационных задач NP-сложности: задач о составлении расписания [3], задач об упаковке [4], задач о выборе оптимального маршрута [5] задач о размещении и ряда других оптимизационных задач [6, 7].

Метод поиска с запретами дает результаты, близкие к оптимальным, за приемлемое время, что позволяет использовать его в оптимизационных задачах наряду с другими метаэвристическими методами.

Актуальность проблемы создания эффективных алгоритмов для задачи упаковки обусловлена как широким практическим применением задач раскроя-упаковки (Р-У) в различных отраслях производства, так и многообразием постановок задач Р-У, трудностью создания адекватных математических моделей и выбора методов их решения. Задачи Р-У представляют собой важный раздел задач дискретной оптимизации и исследования операций. В рамках решения этих задач исследуются общие проблемы, характерные для указанных областей математики.

СПИСОК ЗАПРЕТОВ

Основной проблемой оптимизационных метаэвристических методов является необходимость перехода от локально оптималь-

ных решений к глобальному оптимуму. В методе поиска с запретами эта проблема решается введением специальной конструкции – списка запретов – TL (Tabu List).

В этом списке запоминаются ранее рассматривавшиеся решения (или их некоторые характеристики) и при очередном выборе нового решения запрещается выбирать из окрестности решения, содержащиеся в этом списке.

В общем виде список запретов можно описать как множество некоторых ограничений $TL = \{t_j\}$, $j = 1,\ldots, p$; где p – длина списка запретов. Если решение удовлетворяет некоторому $t_j \in TL$, $j = 1,\ldots, p$, будем называть его запрещенным. Пусть текущая итерация имеет номер i, тогда обозначим очередное решение через $x(i)$, а список запретов через $TL(i)$.

Обозначим множество допустимых соседних решений, не содержащее запрещенных решений через $N'(x(i)),TL(i)) \subseteq N(x)$. Одним из важнейших в данном алгоритме является вопрос об оптимальной длине списка запретов и его структуре.

Структура TL обычно определяется спецификой задачи, основная трудность заключается в минимизации сложности проверки, является ли решение запрещенным.

Во многих задачах в списке запретов хранятся не сами ранее посещенные решения, а параметры оператора, применение которого может привести к такому решению. Например, если оператор представляет собой перестановку двух компонент вектора, то список запретов может содержать номера компонент вектора, которые запрещено переставлять.

Считается, что эффективность алгоритма во многом зависит от длины списка запретов. С одной стороны, слишком короткий список запретов может привести к зацикливанию, а с другой стороны, слишком большой список запретов ведет к дефициту «хороших» решений в окрестности и увеличивает время проверки выполнения условий запрета для рассматриваемых решений.

При работе со списком TL статической длины, то есть постоянной в течение всего времени решения задачи, обычно используется циклический список, в котором после заполнения всего списка старые запреты вытесняются новыми, то есть каждый запрет хранится в течение определенного числа итераций, равного длине списка. В последнее время перспективным считается ведение списка запретов с динамической длиной.

Наиболее общую стратегию можно описать так: следует уменьшать длину списка при нахождении в «хорошей» области пространства решений, то есть если происходит улучшение целевой функции в течение нескольких последних итераций, и необходимо увеличить длину списка в противном случае.

Первая рекомендация помогает «расширить» окрестность решений с тем, чтобы увеличить вероятность нахождения оптимума (локального или глобального), а вторая, напротив, позволяет «сузить» поиск в неперспективной области для того чтобы не тратить много времени на просмотр «плохих» решений и как можно быстрее перейти в более хорошую область.

Другой составной частью алгоритма, связанной с окрестностью соседних решений, является список разрешающих условий – ACL (ACL, Aspiration Criteria Level). Установившегося русского термина для перевода английского названия нет, и в дальнейшем он будет называться именно так. Список содержит набор условий, выполнение которых для некоторого запрещенного решения влечет за собой включение его в окрестность. Эти условия выбираются, исходя из специфики задачи.

МЕТОД ПОИСКА С ЗАПРЕТАМИ

Приведем общее описание алгоритма поиска с запретами. В конкретных приложениях некоторые пункты алгоритма могут быть пропущены или же добавлены новые.

Алгоритм TS

1. Найти начальное решение x_0. Положить номер текущей итерации $i = 0$. В качестве наилучшего решения (рекорда) положить $x^* = x_0$. Сформировать списки TL и ACL.

2. Сгенерировать окрестность соседних решений $N'(x(i),TL(i), ACL(i))$.

3. Выбрать лучшее $y \in N'$ по отношению к целевой или оценочной функциям и положить текущее решение $x(i) = y$.

4. Обновить списки TL и ACL.

5. Если выполнены условия запуска процедур диверсификации или интенсификации, то выполнить их (при этом может измениться текущее решение).

6. Если $x(i)$ лучше x^* по отношению к целевой функции, то сменить рекорд $x^* = x(i)$.

7. Если выполнены условия останова, то алгоритм заканчивается с выдачей рекорда. Иначе перейти к шагу 2.

Приведем описание алгоритма TS для задачи упаковки в контейнеры.

Метод поиска с запретами является наиболее существенной частью общего алгоритма решения рассматриваемой задачи. Одной из характерных особенностей предлагаемого алгоритма является то, что задача упаковки решается в форме задачи расписания. Пусть имеется m идентичных процессоров и n работ с временами выполнения $w_1, w_2,\ldots,w_n$. Требуется сопоставить каждую работу какому-либо процессору, так, чтобы минимизировать максимальное время работы процессоров.

Задача BPP-2. При заданных исходных данных найти

$$x_{ij} \in \{0,1\}, i \in N = \overline{1,n},\ j \in M = \overline{1,m}, \qquad (1)$$

где

$$x_{ij} = \begin{cases} 1, \text{ если } i\text{-я работа сопостпавлена} \\ j\text{-му процессору} \\ 0, \text{ иначе} \end{cases} \qquad (2)$$

$$\sum_{j=1}^{m} x_{ij} = 1, i \in N, \qquad (3)$$

при которых величина

$$T = \max_j \left(\sum_{i=1}^{n} x_{ij} \cdot w_i\right) \qquad (4)$$

достигает минимума.

Таким образом, метод поиска с запретами применяется к некоторому начальному недопустимому решению, в котором предметы, сопоставленные нескольким контейнерам превышают его емкость. Цель алгоритма – преобразовать решение таким образом, чтобы оно стало допустимым в исходной задаче.

На первом этапе получаем начальное решение для задачи в исходной форме любой простой эвристикой, в частности, авторы использовали метод «случайная перестановка с параметром»-RPP [3].

На следующем этапе все предметы из наименее заполненного контейнера перемещаются в другие контейнеры, и значение целевой функции уменьшается на единицу.

Алгоритм перемещения довольно простой и состоит в следующем: самый тяжелый предмет из «уничтожаемого» контейнера перемещается во второй из наименее заполненных контейнеров, второй по возрастанию веса предмет перемещается в третий из наименее заполненных контейнеров и так далее, до тех пор, пока все предметы из данного контейнера не будут перемещены. Работа алгоритма демонстрируется на рис. 1, где контейнеры расположены по возрастанию заполненности (суммарного веса сопоставленных им предметов) снизу вверх, а предметы в «уничтожаемом» контейнере расположены по возрастанию веса слева направо.

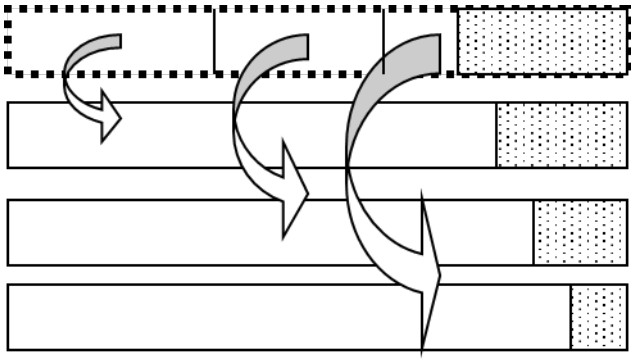

Рис. 1. Алгоритм «уничтожения» контейнера

Сложность алгоритма можно считать константой, так как число предметов в контейнере не зависит от общего числа предметов, а только от отношения среднего веса предмета к емкости контейнера. В решаемых тестовых задачах это число не превосходит 5 для наборов задач 1–4 и 4 для наборов 5–8, а так как уничтожается самый легкий контейнер, то обычно число предметов в нем еще меньше этой границы.

Очевидно, в результате работы алгоритма «уничтожения» контейнера решение

становится недопустимым, некоторые контейнеры переполнены. Поэтому на следующем этапе метод поиска с запретами пытается преобразовать полученное недопустимое решение в допустимое, где отсутствуют переполненные контейнеры.

Если это удается, то вновь уничтожается самый легкий контейнер и повторяется поиск допустимого решения методом TS. Процесс заканчивается, если на очередном шаге получено допустимое решение со значением целевой функции, совпадающим с нижней границей, или если не удалось получить допустимое решение. В последнем случае результатом будет предыдущее допустимое решение, полученное до уничтожения контейнера.

На следующем этапе метод поиска с запретами пытается преобразовать полученное недопустимое решение в допустимое, где отсутствуют переполненные контейнеры.

Может показаться, что такая схема приводит к слишком большим затратам вычислительного времени, так как несколько раз решается задача BPP-2 методом TS. В действительности же для числа контейнеров, больших, чем $N_0 + 1$ допустимое решение находится очень быстро. Число итераций, необходимых для нахождения допустимого решения при заданном значении числа контейнеров, становится большим только при числе контейнеров равным оптимальному значению. Это демонстрируется на рис. 2, где показаны доли числа итераций, необходимых для нахождения допустимых решений при заданном числе контейнеров.

Из рисунка видно, что для нахождения решения для 105, 104, 103 и 102 контейнеров понадобилось число итераций, составляющих всего по 1 % от общего числа итераций (а именно от 7 до 11 итераций), при уменьшении числа контейнеров до 101 и 100 число итераций незначительно возросло и составило 2 % и 3 % (18 и 29 итераций). Основная доля итераций – 91 % (830 итераций) приходится на нахождение допустимого решения с оптимальным числом контейнеров, равным 99. Эксперимент был проведен с задачей № 1 из тестового набора 2. Для задач из других наборов соотношение числа итераций качественно остается таким же. Причем доля числа итераций, выполняемых при нахождении оптимального решения, возрастает с ростом размерности задачи.

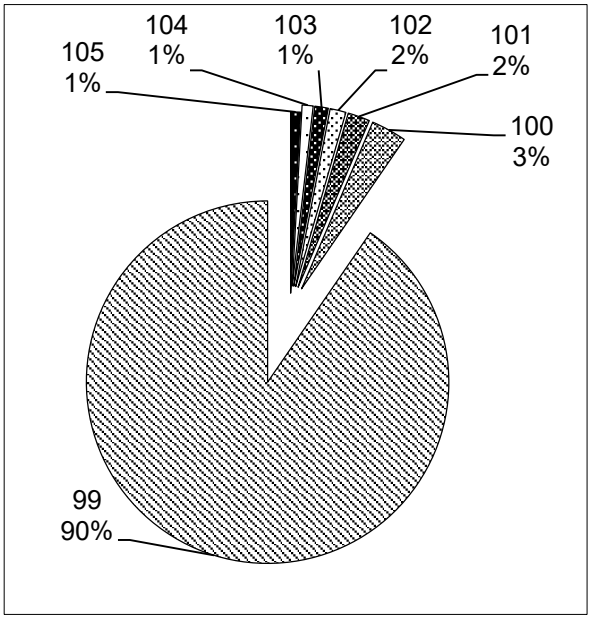

Рис. 2. Доля числа итераций для нахождения допустимого решения при заданном числе контейнеров. Задача № 1, тестовый набор 2

Для задач набора 1 с числом предметов 120 эта доля составляет порядка 70–85%, а для задач наибольшей размерности набора 4 с числом предметов 1000 уже порядка 99 %. Таким образом, наибольшие затраты времени возникают при поиске решения для оптимального числа контейнеров, а при поиске допустимого решения для большего числа контейнеров затраты времени незначительны.

Подобная схема напоминает метод дихотомии и была использована в работе [6]. Отличие состоит в том, что в указанной работе предварительно вычислялись различные нижние границы, и затем для получения начального решения применялся алгоритм WFD с заданным числом контейнеров (равным наилучшей из полученных нижних границ). При этом вычисление каждой из границ имело линейную или квадратичную сложность от размерности задачи, и алгоритм WFD также имел сложность порядка $O(n \cdot \log(n))$. В предлагаемом здесь методе при получении начального решения только один раз выполняется алгоритм RPP со

сложностью O(n·log(n)), что позволяет улучшить вычислительное время.

Приведем формальное описание общего алгоритма.

Алгоритм Упаковка

1. Положить номер шага $I = 0$. Получить начальное решение x_0 методом RPP со значением целевой функции Nrpp. Положить текущее значение числа контейнеров N_i = Nrpp. Положить наилучшее значение целевой функции $N^* = N_i$.

2. Вызов алгоритма «уничтожения» контейнера. $N_i = N_i - 1$.

3. Вызов алгоритма поиска с запретами – TS.

4. Положить $i = i + 1$. Если удалось получить допустимое решение, то $N^* = N_i$, шаг 5. Иначе шаг 6.

5. Если $N_i = N_0$, где нижняя граница N_0 вычисляется по формуле, то получено оптимальное решение, остановка работы. Иначе шаг 2.

6. Остановка алгоритма с выдачей результата N*.

ЗАКЛЮЧЕНИЕ

Предлагаемая реализация метода поиска с запретами для задачи упаковки в контейнеры (Bin Packing Problem-BPP) в основном совпадает с описанной выше общей схемой метода. С другой стороны она имеет ряд особенностей. В приведенной табл. 1 показаны основные характеристики компонент предлагаемого алгоритма.

Поясним два момента. Во-первых, в алгоритме отсутствует список разрешающих условий. Это обусловлено тем, что применение стандартных разрешающих условий в лучшем случае не приводит к улучшению поиска, а в худшем ведет к зацикливанию.

Во-вторых, остановка работы алгоритма происходит при выполнении любого из двух условий. Первое состоит в том, что получено допустимое решение – общий вес предметов в любом контейнере не превышает его емкости.

Второе заключается в превышении заданного числа итераций. Это число выбирается экспериментально и зависит от размерности задачи. Для задач набора 1 со 120 предметами максимальное число итераций принималось равным 1000, а для задач набора 4 с 1000 предметов максимальное число итераций брали равным 100000.

Таблица 1

Характеристика компонент метода TS для задачи BPP-2

Компоненты метода	*Характеристика компонент*
Окрестность решений	Использование двух полиномиальных окрестностей. Рандомизация окрестностей
Целевая/оценочная функция	Оценочная функция обеспечивает градиентный спуск по окрестности. Сложность вычисления значения функции – константа
Список запретов	Различные структуры списка
Список разрешающих условий	Отсутствует
Интенсификация	Косвенная, осуществляется управлением списка запретов
Диверсификация	Быстрый алгоритм «обмена 2 на 1»
Условия остановки	1) получение допустимого решения; 2) превышение заданного числа итераций

При выполнении первого условия алгоритм возвращает найденное допустимое решение, если же выход произошел при превышении максимального числа итераций, алгоритм возвращает решение с наименьшим суммарным переполнением контейнеров, найденное в процессе поиска.

Выбор длины списка запретов зависит от размерности решаемой задачи и мощности окрестности. При коротком списке запретов алгоритм может зациклиться [8]. При длинном списке может оказаться так, что большая часть окрестности будет запрещена, что также не приведет к хорошим результатам.

СПИСОК ЛИТЕРАТУРЫ

1. **Боровых Н. И., Красоткина О. В**. Применение алгоритма поиска с запретами в задаче автоматизированного составления оптимального штатного расписания // Известия ТулГУ. Технические науки. 2019. Т. 2, № 6. С. 218–227. **[** N. I. Borovyh, O. V. Krasotkina, “Application of the search algorithm with prohibitions in the problem of automated compilation of the optimal staffing schedule”, (in Russian), in *Izvestiya TulGU. Tehnicheskie nauki*, vol. 6, no. 2, pp. 218-227, 2019. **]**

2. **Руднев А. С.** Вероятностный поиск с запретами для задачи упаковки кругов и прямоугольников в полосу // Дискретный анализ и исследование операций. 2018. Т. 16, № 4. С. 61–86. **[** A. S. Rudnev, “Probabilistic tabu search for the problem of packing circles and rectangles into a strip”, (in Russian), in *Diskretny`j analiz i issledovanie operacij*, vol. 16, no. 4, pp.61-86, 2018. **]**

3. **Усманова А. Р.** Вероятностные жадные эвристики для задачи упаковки в контейнеры // Первая всероссийская научно-практическая конференция по вопросам решения оптимизационных задач в промышленности: сб. докладов Оптим-2001. СПб.Ж ОПТИМ, 2001. С. 141–145. **[** A. R. Usmanova, “Probabilistic Heuristics for the container packing problem”, (in Russian), in *The First All-Russian Scientific and Practical conference on the optimization problem in industry solution: Collection of reports Optim-2001.* St. Petersburg, 2001. **]**

4. **Optimizing** base station location and configuration in UMTS networks / E. Amaldi, et al. // ORSA Journal on computing. 1989. Vol. 1, No. 3. Part I. Pp. 190-206.

5. **Glover F**. Tabu search – part II // ORSA Journal on computing. 1990. Vol. 2, No. 1. Pp. 4-32.

6. **Martello S.,Toth P.** Knapsack Problems, Algorithms and Computer Implemetations. England: John Wiley and Sons Ltd., 1990. 296 p.

7. **A Tabu** Search Algorithm for Application Placement in Computer Clustering / J. P. Van der Gaast, et al. // Computers & Operations Research. 2014.Vol. 50, No. 1. Pp. 38-46.

8. **Vogt L., Poojari C. A., Beasley J. E**. A tabu search algorithm for the single vehicle routing allocation problem // Journal of the Operational Research Society. 2007. Vol. 58, No. 4. Pp. 467-480.

ОБ АВТОРАХ

УСМАНОВА Анжелика Рашитовна, доц. каф. ВМК. Инженер-программист (УГАТУ, 1997). Канд. физ.-мат. наук (БГУ, 2002). Иссл. в обл. эвристических алгоритмов решения задач дискретной оптимизации.

ВАЛИАХМЕТОВА Юлия Ильясовна, доц. каф. ВМиК. Дипл. инж. (УГАТУ, 2004). Канд. техн. наук по математическому моделированию, численным методам и комплексам программ (УГАТУ, 2008). Иссл. в обл. эвристических алгоритмов решения задач дискретной оптимизации.

METADATA

Title: Features of tabu search method for the packaging problem.

Authors: A. R. Usmanova [1], Yu. I. Valiakhmetova [2]

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

Email: [1] kfmn2004@mail.ru, [2] julikas@inbox.ru

Language: Russian.

Source: SIIT (scientific journal of Ufa State Aviation Technical University), vol. 4, no. 2 (9), pp. 37-42, 2022. ISSN 2686-7044 (Online), ISSN 2658-5014 (Print).

Abstract: The paper considers the problem of packing objects into containers, which is NP-complete. The authors explore one of the popular metaheuristic methods – tabu search (TS). It is proposed to use a mathematical model of a related problem – the scheduling problem for parallel processors. The authors propose an algorithm that makes it possible to reduce the process of obtaining an optimal solution in the packing problem to obtaining an acceptable solution in the processor scheduling problem. Two neighborhoods of neighboring solutions of polynomial complexity are considered. The paper discusses ways to construct one of the most essential components of the tabu search method, the tabu list (TL). The authors proposed a model with a dynamic list length. An evaluation function is also proposed, which has a constant computational complexity. Some results of a numerical experiment demonstrating the effectiveness of the proposed approaches are presented.

Key words: tabu search; heuristic algorithm; optimization; bin-packing.

About authors:

USMANOVA, Angelika Rashitovna, Assoc. Prof., Dept. of CMC. Engineer-programmer (USATU, 1997). Cand. of Phys.-Math. Sci. (BSU, 2002).

VALIAKHMETOVA, Yuliya Ilyasovna, Assoc. Prof., Dept. of Computational Mathematics and Cybernatics. Dipl. Engineer (USATU, 2004). Dr. of Tech. Sci. (USATU, 2008).