

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И ОБРАБОТКА ДАННЫХ

Научная статья

УДК 51-3

DOI: 10.18101/2304-5728-2026-1-63-71

ОБ УНИВЕРСАЛЬНОЙ ОРГАНИЗАЦИИ ВЛОЖЕННЫХ ЦИКЛОВ НА ПРИМЕРЕ СЕГМЕНТАЦИИ ТРАНСПОРТНОЙ ЗАДАЧИ

© **Погодин Игорь Евгеньевич**

доктор физико-математических наук, профессор,
Военно-морской политехнический институт
Россия, 198516, г. Санкт-Петербург, ул. Разводная, 15
ierogodin@mail.ru

© **Удовиченко Андрей Сергеевич**

кандидат физико-математических наук, доцент,
Балтийский государственный технический университет «Военмех»
имени Д. Ф. Устинова
Россия, 190005, г. Санкт-Петербург, ул. 1-я Красноармейская, 1
a_udovichenko@list.ru

Аннотация. Для упрощения обычно весьма трудоемкой процедуры решения транспортных задач предложено два пути их возможной автоматической сегментации, которая заключается в выборе малых замкнутых подзадач транспортных задач произвольных размеров.

При этом указан общий и весьма перспективный путь перехода от задач, требующих большого количества вложенных циклов, к одномерной задаче с единственным переменным параметром, равным количеству не требующихся теперь вложенных циклов. Для этого число, выражающее общее количество требующихся комплектов индексов, записывается в системе счисления с основанием, равным числу вложений, и величина, оказывающаяся на каждой из позиций полученного разложения, используется в качестве счетчика соответствующего цикла. Количество требующих вложения циклов практически не ограничено.

Ключевые слова: транспортная задача, мощности производителей, емкости потребителей, фрагментация, фильтрующие векторы, вложенные циклы.

Для цитирования

Погодин И. Е., Удовиченко А. С. Об универсальной организации вложенных циклов на примере сегментации транспортной задачи // Вестник Бурятского государственного университета. Математика, информатика. 2026. № 1. С. 63–71.

Введение

В теории исследования операций и методов оптимизации выделяются так называемые транспортные задачи (ТЗ) как в классическом, так и в различных модифицированных вариантах. Суть этой задачи заключается в поиске оптимального в смысле минимизации полных затрат на перевозку однотипных грузов от N производителей с различными мощностями к M потребителям с различными объемами запросов при известных затратах на перевозки по различным участкам $\{a_{ij}\}$.

ТЗ известна около 100 лет и допускает целый ряд способов решения, как излагавшихся в [2], так и обсуждаемых ниже.

ТЗ обычно решаются методами линейного программирования и широко применяются в логистике, планировании поставок и распределении ресурсов. Выбор подходящего алгоритма зависит от размеров $M \times N$ матрицы спроса-предложения, требований к точности и вычислительной эффективности.

Известно несколько алгоритмических подходов, каждый из которых имеет свои преимущества и недостатки. Выбор оптимального метода зависит от размера задачи, наличия дополнительных ограничений. Эти подходы обеспечивают высокую точность и гибкость, однако для больших размеров эффективны методы минимальной стоимости и методы разбиения. В дальнейшем развитие направлено на интеграцию этих подходов с параллельными вычислениями и комбинированными алгоритмами, чтобы обеспечить быстрые и надёжные решения даже при сложных ограничениях.

Современные требования к эффективности и гибкости логистических систем привели к расширению классической модели. Важными расширениями являются ТЗ с центрами расщепления, ограничениями по маршруту, многокомпонентной поставкой, динамическими изменениями спроса-предложения и стохастическими параметрами. Каждое из этих расширений влечёт за собой новые сложности: необходимость введения искусственных переменных, добавление ограничений на ребра графа, решение систем линейных уравнений с несколькими потоками, а также учёт неопределённости и времени.

В результате возникло множество специализированных алгоритмов, способных эффективно решать расширенные задачи. Несмотря на это, существует потребность в систематическом обзоре существующих методов, их сравнительном анализе и формулировке рекомендаций по выбору подходящего алгоритма в зависимости от конкретных ограничений и размеров задачи.

Кратко опишем существующие методы. Для небольших и средних ТЗ наиболее эффективным сочетанием является **метод Фогеля**, который обеспечивает быстрое и точное решение без сложной реализации. Выбор конкретного метода зависит от конкретных ограничений задачи, доступных вычислительных ресурсов и требований к точности.

Классический симплекс-метод для ТЗ основан на принципе построения начального базисного решения, (например, по правилу «северо-западного угла») и последующего улучшения через циклические операции. Преимущества этого метода заключаются в скорости получения решения для небольших и средних размеров и использовании структуры задачи. Ограничение в том, что требуется начальное базисное решение; в худшем случае может иметь экспоненциальную зависимость сложности [5; 6].

Алгоритм наименьшей стоимости основан на принципе представления ТЗ в виде сети с вершинами (поставщиками и потребителями) и центральным узлом; поиске потока с минимальной стоимостью (например, с помощью алгоритма Дейкстры с приоритетной очередью). Преимущество — в возможности использовать готовые библиотеки сетевого потока; масштабируется на большие сети, ограничение в требовании более сложной модели (потоки, ограничения на ребра).

Метод потенциалов является модификацией симплекс-метода решения задач линейного программирования применительно к ТЗ. Он позволяет, отправляясь от некоторого допустимого решения, получить оптимальное решение за конечное число шагов. Основан на принципе предпочтений строк/столбцов по разнице между двумя наименьшими стоимостями с дальнейшим применением транспортного симплекс-метода. Преимущество — в скорости приближения, по этой причине частое использование в качестве начального решения, однако не гарантирует оптимальность.

Ступенчатый метод (метод циклов) Используется для улучшения плана в случае его неоптимальности. В свободной ячейке строится цикл и перераспределяются поставки для уменьшения общей стоимости. Проводится анализ циклов в матрице после получения базисного решения для локальных улучшений. Этот метод позволяет улучшать решение до оптимального, однако требует полного перебора циклов, хотя обычно быстро сходится.

Модифицированный метод распределений основан на принципе использования индексов (U- и V-потенциалов) для оценки потенциалов и последующего улучшения решения. Его преимущество — в достаточно быстрой сходимости и широком применении в практике, однако сильно зависит от степени корректности начальных шагов.

Итеративные алгоритмы основаны на принципе задания начального приближения (например, случайного распределения) и корректировки через линейные программные шаги. Преимущество заключается в возможности комбинировать их с другими методами, а ограничение — в возможном требовании многих итераций для сходимости.

Генетические (эволюционные) методы основаны на принципе поиска оптимального решения в пространстве распределений через специальные алгоритмы, они шире по области применения (например, с дополни-

тельными ограничениями), однако не гарантируют глобальную оптимальность решения

Смешанные целочисленные приемы программирования добавляют целочисленные ограничения (например, целочисленность поставок товаров). Они позволяют решать задачи с дискретными переменными, однако подходят лишь для небольших размеров задач.

Метод *simplex* для случая сегментации ТЗ реализован А. С. Удовиченко на языке Python при помощи библиотек *pulp* и *scipy*:

Алгоритм:

1. Ищутся любые «самодостаточные» группы (*subset-sum + backtracking*).

Функция “*find_balanced_groups*” ищет любые подмножества производителей и потребителей, где сумма поставок равна сумме потребностей. Для небольших их наборов она использует полный перебор подсетов, а для больших — ограничивает глубину поиска, чтобы избежать чрезмерно сложных вычислений.

2. Для каждой найденной группы формируется подзадача и затем решается.

При помощи линейного программирования (*scipy.optimize.linprog*).

“*solve_transport_subproblem*” формирует линейную задачу и решает её с помощью “*linprog*” (метод “*simplex*”). Возвращается матрица распределения и общая стоимость

3. Выводятся решения каждой подзадачи и их сумма.

“*segment_and_solve*” разбивает исходную задачу на найденные группы, решает каждую подзадачу отдельно и сохраняет решения в глобальной матрице распределения для удобства просмотра.

Ограничения метода:

Для больших задач (несколько сотен производителей/потребителей) поиск групп может стать узким местом. В таком случае можно применить более продвинутые эвристики (например, жадный алгоритм разбиения по суммам, кластеризация) или использовать MILP для поиска оптимального разбиения.

Общим признаком практически всех перечисленных методов является то, что их трудоемкость сильно возрастает с увеличением числа участников. Поэтому является актуальным поиск возможностей сегментации ТЗ [1], когда некоторая часть производителей может точно обеспечить суммарные запросы части потребителей, что позволяет разбить исходную задачу на несколько независимых, более мелких подзадач. При значительных размерах исходной ТЗ выявление таких подзадач «вручную» становится довольно трудоемким, желательно эти комбинаторные поиски осуществлять с помощью специального компьютерного алгоритма.

Поскольку актуальность такого подхода высока, известны попытки его реализации, например, где сначала ищут любые «самодостаточные» группы (сумма поставок равна сумме потребностей) с помощью функций “*find_balanced_groups*”. Для небольших наборов используется полный

перебор «подсетов», а для больших — ограничивается глубиной поиска, чтобы избежать объемных вычислений. Для больших задач (несколько сотен производителей или потребителей) поиск групп может стать узким местом.

Поэтому здесь предложены пути прямого решения ТЗ с помощью достаточно простой и «прозрачной» сегментации без использования не всегда доступных и недостаточно изученных процедур. Разработке таких алгоритмов посвящена данная работа, кратко изложенная в [4].

Алгоритм предполагает 2 этапа:

1-й этап. Сначала требуется создать множество (2^N-1 (или 2^M-1)) занумерованных «фильтрующих» N - (или M)- мерных векторов (ФВ), состоящих из нулей и единиц, после попарного перемножения элементов которых на элементы вектора мощностей (и емкостей) подобно скалярному произведению получают пробные выборки элементов таблицы ТЗ меньших размеров.

Упорядоченная система (матрица) ФВ размером $((2^N-1), N)$ сначала содержит только нули, затем поочередно (в цикле) на все N позиций устанавливается первая « a » единица, сначала как одиночная (единственная в ФВ:), затем (в новом вложенном цикле) справа от нее на каждую из свободных позиций до N -й включительно поочередно устанавливаются вторые в ФВ « b » единицы. Далее после вторых единиц устанавливаются третьи в ФВ « c » единицы, также сначала как одиночные, после чего в очередном вложенном цикле везде справа от третьих единиц устанавливаются четвертые единицы « d » и т.д.

При создании очередного ФВ всегда сразу ему присваивается увеличивающийся на единицу порядковый номер ФВ в системе. Процесс завершается, когда последняя установленная единица оказывается на N -й позиции в ФВ.

В табл. 1 представлен пример реализации 1-го этапа предложенного алгоритма для создания ФВ с упорядоченной установкой (слева направо) первых « a », вторых « b », третьих « c » единиц в трехпозиционных ($N=3$) ФВ.

Таблица 1

Пример построения системы ФВ для трехпозиционного
 (трехразрядного) вектора

Порядко- вый номер создаваемо- го ФВ	1-я позиция в ФВ	2-я пози- ция в ФВ	3-я позиция в ФВ
№ 1	1a		
№ 2		1a	
№ 3			1a
№ 4	1a	1b	
№ 5	1a		1b
№ 6		1a	1b
№ 7	1a	1b	1c

2-й этап. Для каждой из усеченных таким образом выборок мощностей (и емкостей) подсчитывается суммарное значение их элементов, в случае их равенства выделяется малая замкнутая подзадача. В оставшейся части ТЗ можно снова попытаться этим же методом найти новую малую подзадачу (либо она сразу выделяется алгоритмом), что приводит к фрагментации ТЗ с уменьшением объема работы по решению ТЗ.

Так, при $\vec{A} = (30,10,20)$, $\vec{B} = (10,10,40)$ алгоритм выбирает следующие возможные комбинации ФВ (табл. 2)

Таблица 2

Пример подлежащей фрагментации ТЗ размером (3*3)

$A_i \setminus B_j$	10	10	40
30	a ₁₁	a ₁₂	a ₁₃
10	a ₂₁	a ₂₂	a ₂₃
20	a ₃₁	a ₃₂	a ₃₃

Для производителей и потребителей в ТЗ из примера табл. 2 с помощью построенных систем ФВ алгоритм выделяет следующие варианты замкнутых подзадач (табл. 3)

Таблица 3

Пример выделенных с помощью ФВ подзадач ТЗ размером (3*3)
(табл. 2)

Производители			Потребители		
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	0	1
0	0	1	1	1	0
0	1	0	1	0	0
0	1	0	0	1	0

Рассмотренный метод вложенных циклов для построения «фильтрующих» векторов с монотонно увеличивающимися номерами позиций прост и нагляден. Однако на этом пути при изменении числа участников (производителей или потребителей), т. е. соответствующих длин векторов, требуется построение новой программы под новое число вложенных циклов.

Поэтому здесь предлагается рассматривать «фильтрующие» векторы в виде N -позиционных (N – число участников) $2^N - 1$ чисел в N -ричной системе счисления со значениями:

$$\alpha_{s,j} = E \left[\frac{Z}{N^{j-1}} - N * E \left[\frac{Z}{N^j} \right] \right] + 1 \quad (1)$$

для j -й позиции s -го ФВ. Единица в конце выражения добавлена потому, что в N -ричной системе счисления используются числа от 0 до $(N-1)$, в то время как позиции (разряды) в N -разрядном числе должны нумероваться от 1 до N . Здесь функция $E[\dots]$ — «антье» (entier) — означает выбор наибольшего целого, не превосходящего указанного в скобках аргумента. Каждый выделяемый таким образом разряд соответствует счетчику конкретного вложенного цикла в первом методе.

Среди полученных таким образом ФВ следует отбросить, например, те, у которых значения в отдельных позициях не обнаруживают строго монотонного роста (возможно, наоборот, строго монотонное убывание). Числа в отобранных ФВ указывают места (позиции) расположения единиц. Для трехмерного примера (табл. 1) это векторы: 1, 2, 3, 12, 13, 23, 123.

Множество этих ФВ для вектора мощностей $\{A_i\}$ представляет собой выборки для сравнения с аналогично полученными выборками потребностей $\{B_j\}$ и фрагментации, т.е. выбора малых замкнутых подзадач в исходной ТЗ (см. 2-й этап в изложенном ранее методе с вложенными циклами).

Существенно, что на этом пути можно работать с одной неизменной программой, в которой остается лишь соответствующим образом изменять числовые значения параметров N и M (количество производителей или потребителей).

Представление ряда натуральных чисел в N -ричной системе счисления создает пространство с равномерным распределением. С его помощью можно организовать, например, вычисление определителей произвольного порядка «по определению» [3], когда требуется перебрать и просуммировать все возможные произведения N элементов матрицы, в которых ровно по одному разу (без повторений) представлены каждая строка и каждый столбец со знаковым коэффициентом, равным минус единице в степени числа «нарушений порядка» по одному из измерений после упорядочения по другому измерению. Для этого из всех N -ричных чисел требуется исключить те, которые содержат повторяющиеся значения в рядах.

Заключение

Предложено два пути автоматической фрагментации ТЗ (выбор малых замкнутых подзадач ТЗ произвольных размеров). При этом указан общий путь перехода от задач с большим количеством вложенных циклов к одномерной задаче с единственным переменным параметром, соответствующим количеству вложенных циклов.

Литература

1. Ассаул В. Н., Погодин И. Е. Выводимые планы и декомпозиция транспортной задачи // Известия высших учебных заведений. Сер. Экономика, финансы и управление производством. 2024. № 1(59). С. 55–60.
2. Вентцель Е. С. Исследование операций. Москва: Советское радио, 1972. 552 с.
3. Гантмахер Ф. Р. Определители и матрицы. Москва: Физматлит, 1967. 576 с.
4. Gavrilko A. V., Pogodin I. E. On segmentation of a transportation problem and universal organization of a system of cycles. *Academic science - problems and achievements XXXVIII: Proceedings of the Conference* (Bengaluru, India, 15-16.12.2025, Vol. XXXVIII. Bengaluru, Karnataka, India: Pothi.com). 2025: 157–161.
5. Korte B., Vygen J. *Combinatorial Optimization: Theory and Algorithms*. Springer. 2012: 664.
6. Schrijver A. *Theory of Linear and Integer Programming*. John Wiley & Sons. 1998: 484.

Статья поступила в редакцию 12.02.2026; одобрена после рецензирования 25.02.2026; принята к публикации 16.03.2026.

ON THE UNIVERSAL ORGANIZATION
OF NESTED CYCLES ON THE EXAMPLE
OF THE TRANSPORT PROBLEM SEGMENTATION

Igor E. Pogodin

Double Dr. Sci. (Phys. and Math.), Professor,
academic title: Professor,
Professor of the Department of Mathematics at Naval Polytechnic Institute
198516 Sanct-Petersburg, Razvodnaya str.15, Russia

Andrew S. Udovichenko

Cand. Sci. (Phys. and Math.)
Associate Professor of the Department of Software Engineering and
Intelligent Systems of the Baltic State Technical University "Voenmeh"
named after D.F. Ustinov

Abstract. To simplify the usually very time-consuming procedure of solving transport problems, two ways of their possible automatic segmentation are proposed, which consists in selecting small closed subtasks of transport problems of arbitrary sizes. At the same time, a general and very promising way of transition from problems requiring a large number of nested cycles to a one-dimensional problem with a single variable parameter equal to the number of nested cycles that are no longer required is indicated. To do this, the number expressing the total number of required index sets is written in the number system with a base equal to the number of investments, and the value that appears in each of the positions of the resulting decomposition is used as the counter for the corresponding cycle. In this case, the number of cycles that require investment is practically unlimited.

Keywords: transportation problem, manufacturer powers, consumer capacities, fragmentation, filtering vectors, nested loops.

For citation

Pogodin I. E., Udovichenko A. S. On the Universal Organization of Nested Cycles on the Example of the Transport Problem Segmentation // Bulletin of Buryat State University. Mathematics, Informatics. 2026. N. 1. P. 63–71.

The article was submitted 12.02.2026; approved after reviewing 25.02.2026; accepted for publication 16.03.2026.