

УДК 004.021

DOI: [10.26102/2310-6018/2024.46.3.014](https://doi.org/10.26102/2310-6018/2024.46.3.014)

Формализация задачи оптимизации рендера компьютерной трехмерной графики как вариант многомерной задачи о рюкзаке

В.Н. Мымлик[✉], О.А. Антамошкин, М.М. Фарафонов

Сибирский федеральный университет, Красноярск, Российская Федерация

Резюме. Работа посвящена решению задачи оптимизации рендера компьютерной трехмерной графики, а именно рендер-конвейера. Данная работа сводит упомянутую проблему к многомерному варианту широко известной комбинаторной оптимизационной задачи о рюкзаке. Центральным элементом такой оптимизации является емкость, которую в текущем контексте играют ограниченные аппаратные возможности пользователя, и предметы для укладывания в емкость, роль которых играют различные пиксельные шейдеры. Емкость ограничена величинами ресурсов аппаратуры, а предметы-шейдеры имеют два основных свойства: полезность, выраженную в некоторой величине вклада в качество итоговой картинки, и вес, которым является их вычислительная стоимость для каждого ресурса аппаратуры. Основной задачей в данном контексте является разработка системы, которая будет способна решать такую задачу о рюкзаке в реальном времени для определения в каждый момент времени наилучшей возможной комбинации шейдеров. Таким образом, можно будет получить наилучшее качество изображения и избежать простоя или перегрузки аппаратуры. Основное применение описанная система найдет в сфере компьютерных игр, веб-рекламы, создании фильмов и других сферах, использующих компьютерную графику. Среди ключевых проблем при разработке описанной системы можно выделить сложность в определении вклада каждого отдельного шейдера в результат ввиду субъективности такой оценки. Другой сложностью является поиск компромисса между точностью решения задачи о рюкзаке и скоростью получения результата, с учетом условия, что система должна работать в реальном времени и не замедлять работу программы, для которой выполняется оптимизация.

Ключевые слова: задача о рюкзаке, рендеринг, трехмерная графика, рендер-конвейер, оптимизация, нейронные сети.

Для цитирования: Мымлик В.Н., Антамошкин О.А., Фарафонов М.М. Формализация задачи оптимизации рендера компьютерной трехмерной графики как вариант многомерной задачи о рюкзаке. *Моделирование, оптимизация и информационные технологии*. 2024;12(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=1632> DOI: 10.26102/2310-6018/2024.46.3.014

Formalization of the computer three-dimensional graphics rendering optimization problem as a variant of the multidimensional knapsack problem

V.N. Mymlikov[✉], O.A. Antamoshkin, M.M. Farafonov

Siberian Federal University, Krasnoyarsk, the Russian Federation

Abstract. The work is devoted to solving the problem of optimizing the rendering of computer three-dimensional graphics, namely the rendering pipeline. This work reduces the mentioned problem to a multidimensional version of the well-known combinatorial optimization knapsack problem. The central element of this optimization is capacity, which in the current context is the user's limited hardware capabilities, and the items to be placed in the capacity, which are various pixel shaders. The capacity is limited by the values of the hardware resources, and the shader items have two properties - utility,

expressed in some value of contribution to the quality of render, and weight, which is their computational cost. The main challenge in such a context is to develop a system that will be able to solve such a knapsack problem in real time, in order to determine at each moment the best possible combination of shaders. Thus, it will be possible to obtain the best image quality and avoid downtime or overloading of the hardware. The main application of the described system will be in the sphere of computer games, web advertising, movie making and other spheres using computer graphics. Among the key problems in the development of the described system is the difficulty in determining the contribution of each individual shader to the result, due to the it's subjectivity. Another difficulty is finding a compromise between the accuracy of the knapsack problem solution and the speed of obtaining the result, taking into account the condition that the system must work in real time and not slow down the program for which the optimization is being performed.

Keywords: knapsack problem, rendering, 3D graphics, render pipeline, optimization, neural networks.

For citation: Mymlikov V.N., Antamoshkin O.A., Farafonov M.M. Formalization of the computer three-dimensional graphics rendering optimization problem as a variant of the multidimensional knapsack problem. *Modeling, Optimization and Information Technology*. 2024;12(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=1632> DOI: 10.26102/2310-6018/2024.46.3.014 (In Russ.).

Введение

Процесс отображения трехмерной графики – это достаточно длинная последовательность операции по обработке информации о виртуальных объектах, описываемых в виде вершин, линий, полигонов и множества других данных, таких как текстуры и карты нормалей. Цель этой обработки заключается в трансформации упомянутой информации в двухмерное изображение, а называется этот процесс рендером трехмерной графики. Рендеринг включает в себя множество различных этапов, начиная от преобразований координат вершин и заканчивая постобработкой отрисованного кадра. Все эти действия составляют рендер-конвейер – программную систему, осуществляющую отображение трехмерной графики.

На сегодняшний день существует актуальная проблема оптимизации производительности программного обеспечения, которая связана не только с очевидным стремлением к экономии денег и ресурсов, но также не в последнюю очередь продиктована стремлением к сохранению окружающей среды. В частности, неоптимальная работа программного обеспечения приводит к большим затратам электроэнергии, как следствие, к большему количеству сжигаемого топлива и, наконец, к ухудшению экологической обстановки.

На данный момент существует огромное множество различных технологий трехмерного рендера, каждая со своими преимуществами в виде улучшения качества картинки и недостатками, в числе которых может быть большая вычислительная цена. Также, несмотря на наличие относительно универсальной логической модели рендер-конвейера, существует множество его программных реализаций, среди которых можно выделить такие, как OpenGL от компании CGI¹, Direct3D от корпорации Microsoft [1], Vulcan от Chronos Group [2], а также вариативный графический конвейер в игровом движке Unity [3]. Такое множество реализаций не позволяет единообразно оптимизировать все существующие графические технологии, также непосредственно сами компоненты рендера по отдельности уже оптимизированы настолько, насколько это возможно. Потому наиболее разумным подходом к улучшению ситуации в данной области выглядит применение системного подхода и сведение задачи оптимизации рендер-конвейера к варианту задачи о рюкзаке.

¹Гинсбург Д., Пурномо Б. *OpenGL ES 3.0. Руководство разработчика*. Москва: ДМК Пресс; 2015. 448 с.

Модель системы оптимизации рендер-конвейера

Общая абстрактная структура рендер-конвейера включает в себя 4 крупных этапа: этап приложения, этап обработки геометрии, растеризацию и обработку пикселей. Первый этап предполагает определение объектов для отображения и различные действия в виртуальном мире, не связанные непосредственно с отрисовкой. Первый этап выполняется на CPU, после чего вся необходимая информация переносится на графический процессор и дальнейшие действия происходят уже на нем. Второй этап предполагает серию матричных трансформаций над данными о вершинах и полигонах моделей, составляющих отображаемую сцену, а также некоторые необязательные операции, такие как тесселяция. Следующий этап – растеризация, предполагает обработку полигонов, в частности, определение фрагментов – частей пикселей, пересекающихся с тем или иным полигоном. Финальный этап заключается в определении цвета каждого пикселя при помощи применения к данным, полученным от предыдущих этапов пиксельных шейдеров.

Основной задачей настоящей работы является создание системы оптимизации рендер-конвейера путем разработки способа решения задачи рюкзака в соответствующем контексте за сравнительно малое количество времени и с достаточной точностью для обеспечения высокого качества результатов. Система, которой предстоит решать эту задачу, будет состоять из трех основных компонентов: непосредственно самого рендер-конвейера, экспертной системы оценки влияния шейдеров и решателя задачи о рюкзаке. Работа системы оптимизации будет выполняться на процессоре, и в ее задачи будет входить формирование списка пиксельных шейдеров, которые должны быть применены на этапе обработки пикселей. Для определения значимости шейдеров будет применяться система экспертной оценки, которая будет логически размещена после завершения рендера, когда картинка уже сформирована. Система будет выполнять поиск наилучшей комбинации шейдеров на всем протяжении работ приложения, что позволит эффективно использовать все ресурсы системы, избегая простоя. Описанная структура представлена на Рисунке 1.

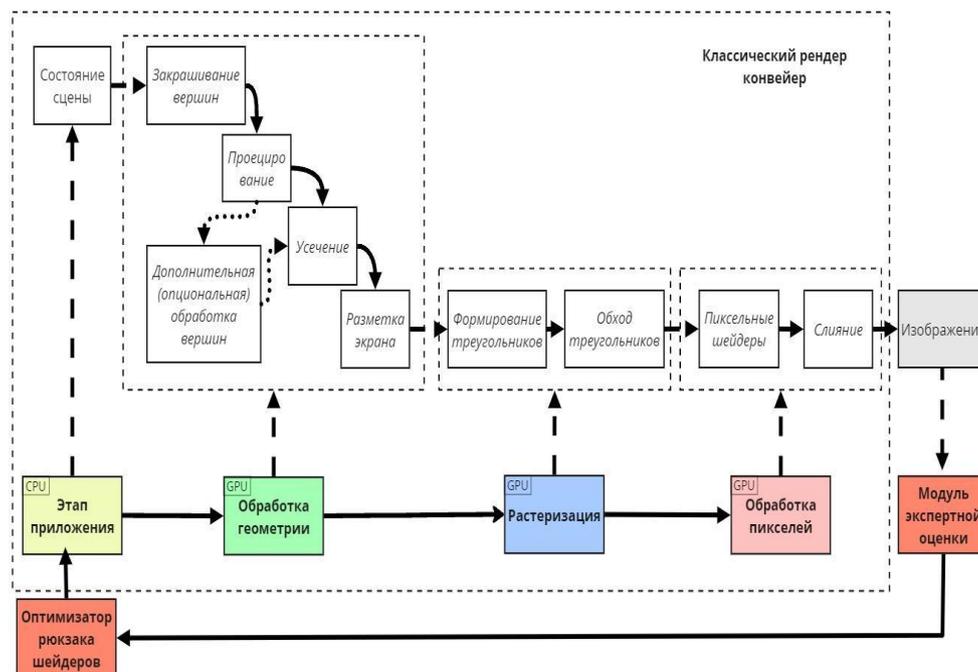


Рисунок 1 – Схема организации предложенной системы
Figure 1 – Scheme of organization of the proposed system

Формализация задачи оптимизации рендер-конвейера как варианта задачи о рюкзаке

Задача о рюкзаке в общем виде предполагает ситуацию, в которой существует некая емкость, характеризуемая числовым значением ее величины (или множеством числовых значений, если речь идет о многомерном рюкзаке), и множество предметов, которые в данную емкость можно положить [4]. Предметы описываются двумя характеристиками: ценностью (в случае их загрузки в рюкзак) и весом (возможно многомерным), который ограничивает число предметов для укладывания в рюкзак.

В контексте оптимизации рендера ситуацию можно трактовать как вариант решения задачи многомерного рюкзака. Центральным элементом, который будет подвергаться оптимизации, является множество пиксельных шейдеров, которые могут быть включены в конвейер. Именно эти шейдеры будут предметами, ценность которых заключается в том, насколько сильно улучшается итоговое качество картинки при включении того или иного шейдера в процесс отображения. Весом предметов-шейдеров является их вычислительная стоимость – объем выделенной видеопамяти, время на загрузку необходимых данных (например текстур), время выполнения инструкций, составляющих шейдер и т. д. В математической форме такая задача выглядит следующим образом: имеется n шейдеров. Для каждого i – го шейдера определены вероятные вычислительные затраты на j – ый ресурс системы, выполняющей отрисовку графики $w_i^j \geq$, и величина улучшения качества картинки в результате внедрения в рендер-конвейер i -ого шейдера $v_i > 0, i = 1, 2, \dots, n$. Ограничение суммарной вычислительной сложности шейдеров в рюкзаке задается свободной мощностью системы W . Необходимо:

$$\text{максимизировать } \sum_{i=1}^n v_i x_i \quad (1)$$

$$\text{с ограничениями } \sum_{i=1}^n w_i^j x_i \leq W, j = 1, \dots, m; x_i \in \{0,1\}, i = 1, \dots, n. \quad (2)$$

Существенную сложность при такой постановке задачи составляет определение числовой характеристики улучшения качества итогового изображения, обеспечиваемого шейдерами. Качество изображения – величина крайне субъективная, и определение ее числового значения – задача нетривиальная. В частности, возникают такие проблемы, как «проблема среднего пользователя», сложность адекватной оценки результатов и т. д. Однако без этой числовой величины постановка задачи будет неполной и будет невозможно приступить к ее решению, потому необходимым компонентом такой системы является разработка экспертной системы для оценки качества изображений и выявления вклада отдельных шейдеров в это качество. На текущий момент есть одна работа, в рамках которой были получены некоторые данные о значимости шейдеров при рендере, хотя данная работа и охватывает лишь небольшое число шейдеров [5].

Возможные подходы к решению задачи

Задача рюкзака известна уже более века, и для нее было разработано множество различных алгоритмов решения. Основной сложностью является NP-полнота данной задачи, что делает невозможным быстрое получение точного и быстрого решения для такой задачи. Потому каждый известный алгоритм – это некоторый компромисс между скоростью получения результата и его точностью. В зависимости от варианта такого компромисса алгоритмы можно разделить на две группы: точные и приближительные [6].

Точные алгоритмы в результате выполнения предоставляют истинное, верное решение задачи о рюкзаке, однако выглядят как не самый лучший вариант решения задачи из-за их очень большого времени работы, которое критически важно при работе

в реальном времени. Тем не менее, эти алгоритмы могут пригодиться для формирования эталонных примеров решения задачи в описываемом контексте, например, для формирования обучающего набора данных для обучения нейронных сетей в случае их использования в проекте. Среди точных алгоритмов решения классической задачи о рюкзаке наиболее популярным решением является динамическое программирование [7, 8], для которого существуют различные модификации, в частности, для работы с многомерным вариантом задачи [9]. Другим популярным решением для данной вариации задачи является метод ветвей и границ, а также разные его модификации [9-11].

Приблизительные методы решения задачи о рюкзаке выглядят наиболее перспективными благодаря относительно большей скорости работы, чем у точных алгоритмов. Данная группа алгоритмов объединяет в себе множество самых разнообразных подходов, такие как жадные алгоритмы, решения, использующие жадный подход как один из этапов работы [12, 13] и приближенные схемы полиномиального времени [14, 15]. Также к данной группе относятся технологии машинного обучения, в частности, нейронные сети [16, 17] и биоинспирированные алгоритмы [18, 19], среди которых можно отдельно выделить эволюционные методы [20, 21]. Однако далеко не все из них применимы для многомерной постановки задачи о рюкзаке. В частности, согласно исследованию из работы [22], использование схем полиномиального времени в многомерных задачах о рюкзаке невозможно. Тем не менее, в данной сфере все же есть ряд решений, которые кратко описаны в Таблице 1.

Таблица 1 – Сравнение существующих методов решения многомерной задачи о рюкзаке
Table 1 – Comparison of existing methods for solving the multidimensional knapsack problem

Группа методов	Авторы	Название метода	Характеристика результатов
Точные методы	Boyer V., El Baz D. [9]	Комбинация динамического программирования и метода ветвей и границ	Тесты проводились на примерах из Chu and Beasley [23]. При $n \times m = 500 \times 30$ время работы составляет ~26 секунд.
	Martello S., Toth P. [10]	Метод ветвей и границ	Тесты проводились на примерах из Chu and Beasley [23]. При $n = 10000$, для двумерной задачи о рюкзаке время работы находится в районе 200–300 секунд.
	Boussier S. et al. [11]		В работе представлены значения $n = 250, 500$ и $m = 5, 10$ и использованы примеры из OR-Library. При $m = 5$ и $n = 500$ или $m = 10$ и $n = 250$ время работы варьируется от 0 до ~2000 сек. При $m = 10$ и $n = 500$ время работы измеряется часами.

Таблица 1 (продолжение)

Table 1 (continued)

Приблизительные методы	Varnamkhandi M. J., Lee L.S. [20]	Модифицированный генетический алгоритм	Алгоритм тестировали на примерах из работ Chu and Beasley и Qian and Ding. Алгоритм превосходит по метрике Percentage Deviation множество других эвристик при работе не дольше 500 CPU seconds.
	Djannaty F., Doostdar S. [21]	Модифицированный генетический алгоритм	Алгоритм тестировали на примерах из разных сетов, в частности WEISH23, PB6, HP2, SENTO1. Время работы в секундах варьируется от 3 до 350.
	Mingo López L. F., Gómez Blas N., Arteta Albert A. [18]	Алгоритм роя частиц с генетическими операциями	Тестирование проводилось на примерах из сетов SENTO и WEISH. При $n=60$ и $m = 30$ время нахождения оптимума для разных примеров составляет от 3,6 до 5 секунд. При $n = 50$ и $m = 5$ время нахождения оптимума составляет от 1,4 до 3,6 секунд.
	Zhang S., Liu X., Wang M. [19]	Алгоритм муравьиной колонии	Предложенный алгоритм авторы сравнили с тремя другими вариантами Ant colony и GA, предложенное решение лучшее. Примеры для тестирования были взяты из сета Chu and Beasley. Время получения результата в пределах 400 CPU секунд.
	Kong X. et al. [24]	Harmony Search	Минимальное время работы при $n = 500$ и $m = 5$ составляет от 8 до 24 CPU секунд. При $n = 500$ и $m = 30$ время составляет от 31 до 115 CPU секунд.
	Zhou Y., Kuang Z., Wang J. [16]	Transiently Chaotic Neural Network	Примеры взяты из сета WEISH. При комбинации параметров $m/n = 5/90$ время работы составляет 0,79 секунд.

Специфика решения задачи о рюкзаке

Помимо разработки непосредственно системы оптимизации рендера, важной задачей настоящей работы является поиск новых решений задачи о рюкзаке. В ходе текущего исследования планируется решать задачу о рюкзаке путем создания и обучения нейронной сети, ввиду ее гибкости. Нейронные модели в процессе обучения формируют обобщенное представление алгоритма решения задачи о рюкзаке при помощи своих весовых коэффициентов, при этом подразумевается, что у решения такого типа задач есть общая для всех случаев закономерность.

Одна из концепций, которая нуждается в экспериментальной проверке, заключается в том, что нейронная модель, возможно, сможет лучше решать задачу о рюкзаке, если помимо данных о предметах предоставлять ей некоторую дополнительную контекстную информацию о параметрах задачи. В случае оптимизации рендер-конвейера такой дополнительной информацией может быть карта признаков, полученная путем свертки изображения, полученного при рендере предыдущего кадра или же трехмерная карта признаков, полученная путем трансформации данных, описывающих сцену в набор признаков. В теории в этом случае нейронная сеть будет иметь данные о содержимом сцены, что позволит предсказать наиболее эффективные в текущем контексте шейдеры.

Обсуждение

Одной из ключевых сложностей представленной работы является возможность развития сценария, при котором, система будет замедлять работу рендер конвейера, так как ее код будет выполняться на CPU. Необходимо определить степень влияния системы оптимизации на процесс рендера, его скорость и прочие характеристики, и, по возможности, уменьшить негативное влияние на производительность. Одним из возможных путей такой оптимизации является реализация решателя задачи о рюкзаке в виде нейронной сети, большая часть работы которой будет производиться на графическом процессоре, что позволит решать упомянутую задачу в параллельном режиме и за минимально возможное время.

Другой важной проблемой является проблема взаимного влияния шейдеров друг на друга, в частности, включение одного шейдера может уменьшать или увеличивать эффект другого, потому рассмотрение итогового качества картинки как простой арифметической суммы величин воздействия всех шейдеров может быть ошибочным. В частности, использование технологии трассировки лучей, очевидно, уменьшит или вовсе устраним видимое воздействие других решений для имитации поведения света, например, Ambient Occlusion. Необходимо всестороннее исследование данного вопроса, которое, однако, не относится к тематике настоящего проекта.

Также присутствует проблема, связанная с существенным отличием рендер-конвейера в тех случаях, где применяется технология трассировки лучей. Некоторые этапы классического рендера в нем отсутствуют, при этом добавляются новые этапы. Кроме того, у технологии трассировки лучей важную роль имеют дискретные числовые параметры, описывающие поведение лучей – количество отскоков в зависимости от поверхности, коэффициенты отражения/преломления, также свою лепту вносит наличие или отсутствие подавителя шума. Помимо очевидного отличия конвейера у разных технологий рендера, единство также отсутствует и в сфере классического рендера. Хотя логическая модель рендера довольно общая и универсальная, на практике есть множество реализаций, которые были упомянуты ранее. Потому, систему оптимизации рендера придется, так или иначе, адаптировать к этим различиям для достижения желаемой универсальности.

Также к важным моментам разработки описанной системы относится тестирование полученного результата, которое нужно произвести на различных конфигурациях оборудования для проверки универсальности и применимости в самых разных условиях, при разном количестве доступных вычислительных ресурсов.

Заключение

Компьютерная трехмерная графика является неотъемлемой частью современного мира и на ее отображение ежедневно уходит огромное количество электроэнергии,

времени и прочих ресурсов, потому крайне важно, чтобы ее формирование и обработка происходили настолько эффективно, насколько возможно. Система оптимизации рендер-конвейера, сводящая проблему оптимизации трехмерной графики к задаче о рюкзаке, будет востребована во множестве самых разных сфер. Среди них можно выделить сферу развлечений, в частности, компьютерные игры, сферу кинематографа, где предложенное решение позволит уменьшить затраты на создание сложных графических сцен, также определенный интерес может присутствовать в сфере рекламы и веб-графики, где нужно представить привлекательный внешний вид без перегрузки аппаратуры пользователя.

В рамках данной статьи была формализована задача оптимизации ресурсов рендер-конвейера путем сведения ее к варианту многомерной задачи о рюкзаке. Был произведен анализ возможных путей решения поставленной проблемы и определены наиболее существенные возможные затруднения при решении поставленной задачи.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Luna F.D. *Introduction to 3D Game Programming with DirectX 12*. Dulles: Mercury Learning and Information; 2016. 900 p.
2. Castorina M., Sassone G. *Mastering Graphics Programming with Vulkan: Develop a modern rendering engine from first principles to state-of-the-art techniques*. Birmingham: Packt Publishing Ltd; 2023. 382 p.
3. Dickinson C. *Unity 2017 Game Optimization: Optimize all aspects of Unity performance*. Birmingham: Packt Publishing Ltd; 2017. 376 p.
4. Martello S., Toth P. *Knapsack Problems: Algorithms and Computer Implementations*. Chichester: John Wiley & Sons Ltd; 1990. 308 p.
5. Peresunko P., Mamatin D., Antamoshkin O., Peresunko E., Nikitin A. Models of Experts for Shaders Estimation of Rendering Complex 3D Scenes in Real Time. In: *2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), 10–12 November 2021, Lipetsk, Russia*. IEEE; 2021. pp. 895–897. <https://doi.org/10.1109/SUMMA53307.2021.9632071>
6. Cacchiani V., Iori M., Locatelli A., Martello S. Knapsack problems – An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research*. 2022;143. <https://doi.org/10.1016/j.cor.2021.105693>
7. Andonov R., Poirriez V., Rajopadhye S. Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research*. 2000;123(2):394–407. [https://doi.org/10.1016/S0377-2217\(99\)00265-9](https://doi.org/10.1016/S0377-2217(99)00265-9)
8. Pferschy U., Scatamacchia R. Improved dynamic programming and approximation results for the knapsack problem with setups. *International Transactions in Operational Research*. 2017;25(2):667–682. <https://doi.org/10.1111/itor.12381>
9. Boyer V., Baz D.E., Elkihel M. Solution of multidimensional knapsack problems via cooperation of dynamic programming and branch and bound. *European Journal of Industrial Engineering*. 2010;4(4):434–449. <https://doi.org/10.1504/EJIE.2010.035653>
10. Martello S., Toth P. An Exact Algorithm for the Two-Constraint 0–1 Knapsack Problem. *Operations Research*. 2003;51(5):826–835. <https://doi.org/10.1287/opre.51.5.826.16757>
11. Boussier S., Vasquez M., Vimont Y., Hanafi S., Michelon P. A multi-level search strategy for the 0–1 Multidimensional Knapsack Problem. *Discrete Applied Mathematics*. 2010;158(2):97–109. <https://doi.org/10.1016/j.dam.2009.08.007>

12. Dantzig G.B. Discrete-Variable Extremum Problems. *Operations Research*. 1957;5(2):266–288. <https://doi.org/10.1287/opre.5.2.266>
13. Al-douri T., Hifi M., Zissimopoulos V. An iterative algorithm for the Max–Min knapsack problem with multiple scenarios. *Operational Research*. 2021;21(1):1355–1392. <https://doi.org/10.1007/s12351-019-00463-7>
14. Chan T.M. Approximation Schemes for 0-1 Knapsack. In: *1st Symposium on Simplicity in Algorithms (SOSA 2018): Open Access Series in Informatics (OASICs): Volume 61, 07–10 January 2018, New Orleans, LA, USA*. 2018. pp. 5:1–5:12. <https://doi.org/10.4230/OASICs.SOSA.2018.5>
15. Jin C. An Improved FPTAS for 0-1 Knapsack. In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019): Leibniz International Proceedings in Informatics (LIPIcs): Volume 132, 09–12 July 2019, Patras, Greece*. 2019. pp. 76:1–76:14. <https://doi.org/10.4230/LIPIcs.ICALP.2019.76>
16. Zhou Y., Kuang Z., Wang J. A Chaotic Neural Network Combined Heuristic Strategy for Multidimensional Knapsack Problem. In: *Advances in Computation and Intelligence: Third International Symposium on Intelligence Computation and Applications, ISICA 2008: Proceedings, 19–21 December 2008, Wuhan, China*. Berlin, Heidelberg: Springer; 2008. pp. 715–722. https://doi.org/10.1007/978-3-540-92137-0_78
17. Afshar R.R., Zhang Y., Firat M., Kaymak U. A State Aggregation Approach for Solving Knapsack Problem with Deep Reinforcement Learning. In: *12th Asian Conference on Machine Learning, ACML 2020: Proceedings of Machine Learning Research: Volume 129, 18–20 November 2020, Bangkok, Thailand*. PMLR; 2020. pp. 81–96.
18. Mingo López L.F., Gómez Blas N., Arteta Albert A. Multidimensional knapsack problem optimization using a binary particle swarm model with genetic operations. *Soft Computing*. 2017;22(8):2567–2582. <https://doi.org/10.1007/s00500-017-2511-0>
19. Ke L., Feng Z., Ren Z.H., Wei X. An ant colony optimization approach for the multidimensional knapsack problem. *Journal of Heuristics*. 2010;16(1):65–83. <https://doi.org/10.1007/s10732-008-9087-x>
20. Jalali Varnamkhasti M., Lee L.S. A Fuzzy Genetic Algorithm Based on Binary Encoding for Solving Multidimensional Knapsack Problems. *Journal of Applied Mathematics*. 2012;2012. <https://doi.org/10.1155/2012/703601>
21. Djannaty F., Doostdar S. A Hybrid Genetic Algorithm for the Multidimensional Knapsack Problem. *International Journal of Contemporary Mathematical Sciences*. 2008;3(9):443–456.
22. Kulik A., Shachnai H. There is no EPTAS for two-dimensional knapsack. *Information Processing Letters*. 2010;110(16):707–710. <https://doi.org/10.1016/j.ipl.2010.05.031>
23. Beasley J.E. OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*. 1990;41(11):1069–1072. <https://doi.org/10.1057/jors.1990.166>
24. Kong X., Gao L., Ouyang H., Li S. Solving large-scale multidimensional knapsack problems with a new binary harmony search algorithm. *Computers & Operations Research*. 2015;63(C):7–22. <https://doi.org/10.1016/j.cor.2015.04.018>

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Мымликов Владислав Николаевич, Vladislav N. Mymlikov, PhD Student, Siberian аспирант, Сибирский федеральный университет, Красноярск, Российская Федерация.

e-mail: vmymlikov-ki21@stud.sfu-kras.ru

ORCID: [0000-0001-7631-4901](https://orcid.org/0000-0001-7631-4901)

Антамошкин Олеслав Александрович, доктор технических наук, доцент, профессор кафедры программной инженерии, Сибирский федеральный университет, Красноярск, Российская Федерация.

e-mail: oantamoskin@sfu-kras.ru

ORCID: [0000-0002-5976-5847](https://orcid.org/0000-0002-5976-5847)

Oleslav A. Antamoshkin, Doctor of Technical Sciences, Docent, Professor of the Chair of Software Engineering, Siberian Federal University, Krasnoyarsk, the Russian Federation.

Фарафонов Максим Михайлович, аспирант, Сибирский федеральный университет, Красноярск, Российская Федерация.

e-mail: mfarafonov@sfu-kras.ru

ORCID: [0000-0002-2218-538X](https://orcid.org/0000-0002-2218-538X)

Maxim M. Farafonov, PhD Student, Siberian Federal University, Krasnoyarsk, the Russian Federation.

Статья поступила в редакцию 18.07.2024; одобрена после рецензирования 23.07.2024; принята к публикации 30.07.2024.

The article was submitted 18.07.2024; approved after reviewing 23.07.2024; accepted for publication 30.07.2024.