

**Применение генетического алгоритма
для рационального размещения прямоугольных деталей**

М.В. Майрамты

Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
362021, г. Владикавказ, ул. Николаева, 44, Россия

Резюме. Цель. Целью работы является проведение сравнительного анализа эффективности применения генетических алгоритмов для поиска рационального размещения прямоугольных деталей на основе численных экспериментов. **Метод.** Существует два основных метода решения задач оптимизации – точные и приближенные. В исследовании рассматривается одно из направлений приближенных алгоритмов – эвристические, которые основаны на предположении о свойствах оптимального решения. В статье рассматривается генетический алгоритм как метод, который позволяет находить такое размещение деталей, которое близко к оптимальному. Для решения задачи задаются малые объекты, которые необходимо разместить без взаимного перекрытия внутри больших объектов так, чтобы целевая функция достигала минимума. Актуальность исследования данной задачи обусловлена ее принадлежностью к классу NP-трудных задач. **Результат.** Разработана программа, реализующая укладку деталей в полубесконечную полосу с использованием генетического алгоритма. Изучено поведение данной программы на разных классах задач с использованием трех процедур размещения. Анализ работы алгоритма проводится на семи категориях известных тестовых наборов. Каждая категория входных данных содержит по три примера с разным количеством элементов в пределах от 16 до 197. **Вывод.** Для решения задач рационального использования материалов актуальным вопросом является разработка и программная реализация эвристических подходов. Данные методы являются эффективными алгоритмами для оптимального использования ресурсов – финансовых, материальных и других. Сильная сторона метаэвристических методов заключается в их способности решения сложных задач без знания пространства поиска, поэтому эти методы дают возможность решать трудноразрешимые задачи оптимизации.

Ключевые слова: генетические алгоритмы, метаэвристики, рациональное размещение, декодер, оптимизация

Для цитирования: М.В. Майрамты. Применение генетического алгоритма для рационального размещения прямоугольных деталей. Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(3):77-85. DOI:10.21822/2073-6185-2025-52-3-77-85

Application of a Genetic algorithm for the rational placement of Rectangular items

M.V. Mairamty

North Caucasian Institute of Mining and Metallurgy
(State Technological University),
44 Nikolaeva Str., Vladikavkaz 362021, Russia

Abstract. Objective. The aim of the work is to conduct a comparative analysis of the efficiency of using genetic algorithms to find a rational placement of rectangular parts based on numerical experiments. **Method.** There are two main methods for solving optimization problems - exact and approximate. The study considers one of the directions of approximate algorithms - heuristic, which are based on the assumption of the properties of the optimal solution.

In particular, this article considers a genetic algorithm as a method that allows you to find such an arrangement of parts that is close to optimal. To solve the problem, small objects are specified that must be placed without mutual overlap inside large objects so that the objective function reaches a minimum. The relevance of the study of this problem is due to its belonging to the class of NP-hard problems. **Result.** A program was developed that implements the placement of parts in a semi-infinite strip using a genetic algorithm. The behavior of this program on different classes of problems using three placement procedures was studied. The analysis of the algorithm's operation is carried out on seven categories of known test sets. Each category of input data contains three examples with a different number of elements in the range from 16 to 197. **Conclusion.** To solve problems of rational use of materials, the development and software implementation of heuristic approaches is a pressing issue. These methods are effective algorithms for the optimal use of resources - financial, material and others. The strength of metaheuristic methods is their ability to solve complex problems without knowledge of the search space, so these methods make it possible to solve difficult optimization problems.

Keywords: genetic algorithms, metaheuristics, rational placement, decoder, optimization

For citation: M.V. Mairamty. Application of a Genetic algorithm for the rational placement of Rectangular items. Herald of the Daghestan State Technical University. Technical Sciences. 2025; 52(3):77-85. (In Russ) DOI:10.21822/2073-6185-2025-52-3-77-85

Введение. В настоящее время значительное внимание уделяется разработке и исследованию эвристических методов оптимизации. В рамках решения множества задач дискретной оптимизации особую актуальность приобретает создание метаэвристических алгоритмов, основанных на известных подходах в данной области.

К числу таких алгоритмов относятся генетические алгоритмы, которые демонстрируют высокую эффективность при решении задач, характеризующихся обширным пространством поиска и множеством экстремумов, а также задач, требующих сложной формализации целевой функции. Эти методы находят применение в ситуациях, когда необходимо использовать многокритериальный поиск, находить приемлемые и рациональные решения в условиях ограниченных ресурсов, а также в задачах, требующих решения в реальном времени.

Совместное использование принципов эволюции и методов случайного поиска стал основой для разработки генетических алгоритмов. Отбирая лучшие наборы элементов в каждой итерации данные алгоритмы пользуются принципами естественного отбора. В свою очередь, операторы, используемые для выведения новых наборов элементов, имитируют генетическое наследование. Эволюционный поиск в задачах рационального использования ресурсов по аналогии с естественными принципами эволюции рассматривается как последовательное преобразование конечного множества промежуточных решений в другое [3].

Постановка задачи. Моделирование естественных эволюционных процессов является основным направлением нахождения рационального решения для задач размещения деталей на полосу. Генетические алгоритмы проецируют на исследовательскую систему адаптацию процессов естественной системы, описывая их абстрактно и математическими формулами.

Работа алгоритма начинается с некоторой начальной информации, так называемой популяции альтернативных решений, конкурирующих между собой. К ним применяются принципы эволюционных процессов для отбора и селекции новых наборов решений.

После каждой итерации оценивается эффективность найденных решений, которые затем подвергаются дальнейшей селекции. Таким образом реализуется принцип естественного отбора. Задача алгоритма заключается в оптимизации целевой функции. Формулирование целевой функции в решении задачи становится ключевым этапом, так как именно она определяет какие решения лучшие, а какие необходимо исключить.

Немаловажную роль играют метод кодирования и разработка принципов размещения предметов. При этом необходимо учитывать определенные условия.

В данной статье решается задача размещения предметов с условием их непересечения и параллельности сторон предметов сторонам укладываемой полосы, так же не допускаются повороты предметов, а произвольные размещения разрешены. Для формальной постановки задачи задаются ширина полосы упаковки W , m прямоугольных объектов, длина l_i и ширина w_i которых известны ($i = \overline{1..m}$), а также необходимо учитывать следующие условия [1]:

– никакие два предмета не должны пересекаться друг с другом, т. е. для $i, j = \overline{1..n}, i \neq j$

$$((x_i \geq x_j + l_j) \vee (x_j \geq x_i + l_i)) \vee ((y_i \geq y_j + w_j) \vee (y_j \geq y_i + w_i)),$$

где (x_i, y_i) – координаты левого нижнего угла детали на полосе, l_i и w_i – длина и ширина детали соответственно, n – количество деталей;

– никакой предмет не пересекает границ полосы;

$$(x_i \geq 0) \wedge (y_i \geq 0) \wedge (y_i + w_i \geq W)$$

– стороны предметов должны быть параллельны граням полосы.

Цель работы алгоритма – минимизировать пространство укладки прямоугольных деталей, тем самым сократить расходы ресурсов, следовательно, целевая функция выглядит следующим образом:

$$(x_i + l_i) \rightarrow \min.$$

Таким образом, математическую постановку задачи можно выразить так [1]:

$$\begin{cases} L = (x_i + l_i) \rightarrow \min \\ y_i + w_i \geq W, & i = \overline{1..m} \\ (x_i \geq x_j + l_j) \vee (x_j \geq x_i + l_i), & i, j = \overline{1..m}, i \neq j \\ (y_i \geq y_j + w_j) \vee (y_j \geq y_i + w_i), & i, j = \overline{1..m}, i \neq j \\ (x_i, y_i \geq 0) \wedge (y_i + w_i \geq W), & i = \overline{1..m} \end{cases}$$

Методы исследования. При использовании эвристических алгоритмов для решения оптимизационных задач в технике, как и в живой природе, текущий набор субоптимальных решений находится по принципу чередования альтернативных решений на каждой итерации. Это изменение направлено на то, чтобы каждая новая генерация решений наилучшим образом адаптировалась к условиям окружающей среды, а именно к ограничениям и условиям рассматриваемой задачи.

В этом контексте общая генетическая совокупность обозначается термином «генотип», тогда как формирование организма происходит через взаимодействие этой генетической совокупности с внешней средой и называется «фенотипом» [3].

Для определения качества и эффективности выбора текущего набора элементов разрабатывается целевая функция. Она показывает целесообразность и пригодность набора элементов, который рассматривается на текущем шаге. Альтернативные решения сравниваются между собой по целевой функции и лучшие из них отбираются для нахождения следующего набора решений. Достижение значений целевой функции и степень реализации разработанного алгоритма показывает эффективность применения генетического алгоритма. Немаловажную роль для эффективности играет определение начального набора элементов и их порядок в наборе.

В практических задачах при применении генетического алгоритма необходимо выбрать и реализовать операторы случайных изменений наборов на каждой итерации, определить способ представления решения, установить критерии целевой функции для определения выбора эффективных решений, сформировать первичный набор элементов.

Для реализации случайных изменений и формирования новых наборов решений адаптируют некоторые естественные генетические операторы: репродукции, селекции, скрещивания, мутации и т.д. После адаптации таким оператором принято считать некоторую языковую конструкцию, которая описывает одно действие из алгоритма последовательных шагов при формировании следующего набора решений. Эффективность решаемой

задачи повышается использованием комбинаций из разных операторов и их модификаций.

Оператор репродукции или селекции описывает процесс нахождения дочерних решений путем отбора элементов из родительских наборов по значению целевой функции. К отобранному набору элементов применяются генетические операторы для создания альтернативного решения. Существует большое число операторов репродукции, но в своей реализации имеются два основных типа:

- Случайный выбор элементов для образования набора решений, где значение целевой функции не учитывается. Имеет значение только число наборов решений на каждой итерации.
- Отбор элементов по «лучшим» значениям целевой функции для формирования новых решений. Предпочтение может отдаваться наборам элементов как с близкими значениями целевой функции, так и сильно отличающихся между собой.

Из всего разнообразия операторов селекции по эффективности можно выделить «турнирный» и «элитный» алгоритмы. При этом турнирная селекция относится к первому типу, т.е. выбор родительского набора элементов происходит случайным образом согласно заданному размеру, а элитная селекция – ко второму типу, т.е. лучшие элементы выбираются сравнением значений целевой функции.

После отбора элементов, над ними производят различные преобразования, после которых формируются новые наборы решений и уже к ним опять применяются операторы репродукции. Процесс повторяется пока не будет найдено рациональное решение или же не будут достигнуты условия окончания поиска (время или количество итераций).

Преобразования над наборами элементов, для нахождения новых решений, представляют собой аналогию процессов скрещивания или мутации в естественной среде. Они представляют собой некоторые языковые конструкции, которые на основе определенного алгоритма действий позволяют сгенерировать новые наборы решений.

Генетический оператор скрещивания принято называть оператором кроссинговера. После его применения всегда получаются два новых набора элементов. В текущей «популяции» решений отбираются два «родительских» набора элементов, выбирается случайная точка скрещивания, со значением, не превосходящим количество элементов в наборе. После применения одного из правил скрещивания получаются два новых набора элементов. На рис. 1 показано одно из возможных правил реализации данного оператора.

$$\begin{array}{l} P_1 : A B C D | E F G H \\ P_2 : G A B E | C D F H \\ \hline P'_1 : A B C D | G E F H \\ P'_2 : G A B E | C D F H \end{array}$$

Рис. 1 – Пример реализации оператора кроссинговера
 Fig. 1 – Example of crossing-over operator implementation

При реализации оператора мутации учитывают два основных шага – определение нескольких позиций в одном наборе элементов и формировании нового набора по некоторому правилу перестановок [2]. На рис. 2 показано одно из возможных правил реализации данного оператора.

$$\begin{array}{l} P: A | B C D | E F | G H \\ P': A G C D B F E H \end{array}$$

Рис. 2 – Пример реализации оператора мутации
 Fig. 2 – Example of the implementation of a mutation operator

В комбинаторных задачах для визуализации допустимых построений используют различные правила преобразования решения в плане упаковки.

Эффективность полученных решений зависит и от способов декодирования, с помощью которого можно вычислить общую длину размещения элементов на полосе, получить графическое представление решения и рассчитать координаты элементов при упаковке [4].

В программной реализации для визуализации и сравнения методов используются следующие три уровневые процедуры укладки деталей на полосу:

1. **Следующий подходящий предмет (NFDL)** – первый элемент помещается на нижнюю левую позицию полосы, соблюдая условия параллельности граням полосы. Затем мысленно проводится вертикальная линия по правой стороне прямоугольника, что позволяет ограничить блок, в который размещен первый элемент.

Каждая следующая подходящая деталь укладывается в этот выделенный блок. При этом должны быть соблюдены условия не пересечения границ полосы и параллельности элементов друг другу. Если же пространства в блоке недостаточно, то элемент размещается в следующую нижнюю левую позицию.

Временная сложность алгоритма NFDL равна $O(n^2)$ из-за необходимости формирования и поиска незаполненных блоков [1]. Схематически декодер NFDL представлен на рис. 3.

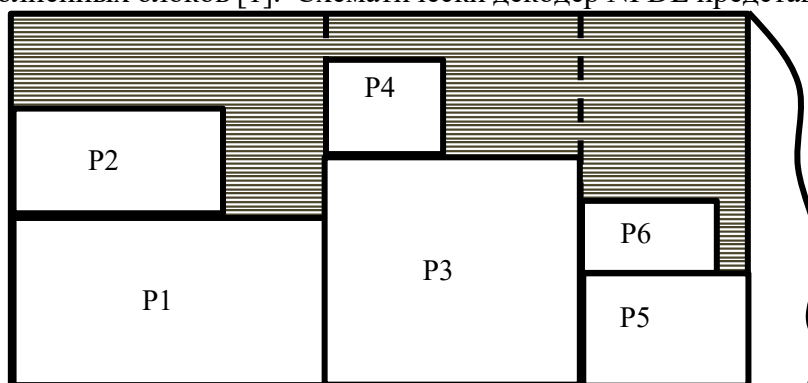


Рис. 3 – Стратегия размещения: следующий по убыванию длины (NFDL)
 Fig. 3 – Placement strategy: next figure in line (NFDL)

2. **Первый подходящий по убыванию длины (FFDL)** – первая деталь из набора помещается в частично заполненный по ширине блок. Далее происходит заполнение этого блока путем подбора подходящей по размеру детали. Если текущий блок заполнен, то формируется следующий блок и в нее размещается первый подходящий из списка элемент [4]. При этом алгоритм FFDL для упаковки требует $O(n^2 \log n)$ времени [5]. Схематически декодер FFDL представлен на рис. 4.



Рис. 4 – Стратегия размещения: первый подходящей по убыванию длины (FFDL)
 Fig. 4 – Placement strategy: the first suitable for decreasing length (FFDL)

3. **Нижний левый (BL)** – пытается разместить прямоугольник p_i ($i = 1, n$) на полосу вниз, насколько это возможно, затем влево, насколько это возможно, снова вниз и т.д. Данный декодер позволяет получить плотный план упаковки полосы, не создавая блоков и тем самым эффективно заполняя пространство упаковки.

Схематически декодер BL представлен на рис. 5.

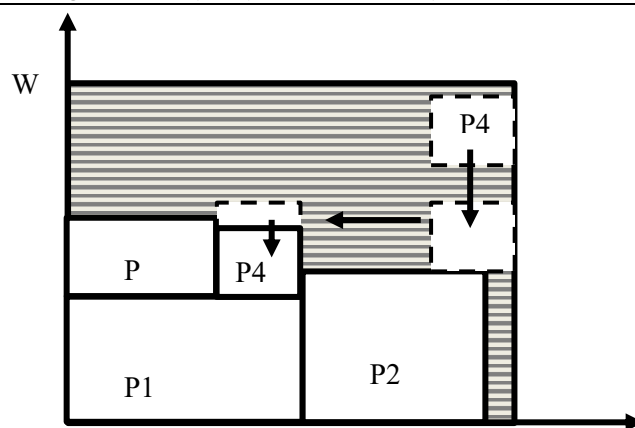


Рис. 5 – Стратегии размещений: нижний левый (BL)

Fig. 5 – Placement strategies: Bottom-left (BL)

Обсуждение результатов. Анализ результатов происходил в соответствии с данными реализованной программы. Запрашиваемые входные данные в программе являются:

1. Количество элементов n .
2. Количество элементов в одном наборе L .
3. Количество наборов решений в одной итерации S .
4. Ширина полосы W [1].

Программа реализовывалась по следующему алгоритму:

Шаг 1. Сформировать начальное размещение объектов на полосе. Каждый набор состоит из элементов и такой набор называется приоритетным списком. В этот список элементы должны входить без повторений.

В одном наборе решений при использовании генетического алгоритма приоритетные списки не должны повторяться, т.е. не должно быть в наборе решений одной итерации одинаковой последовательности укладки элементов. Для каждого приоритетного списка из набора решений рассчитывается значение целевой функции – используемая длина полосы текущим набором элементов.

Шаг 2. Для выбора следующего набора альтернативных решений необходимо определить критерии оптимальности целевой функции. Для этого вычисляется общая занимаемая длина каждого набора элементов и половину лучших приоритетных списков отбирается для формирования нового набора решений.

Отбор происходит с использованием оператора репродукции – отбираются те наборы, у которых значение целевой функции меньше. Для вычисления значения целевой функции по одной из процедур укладки строится карта размещения объектов на основании приоритетного списка.

Шаг 3. На данном шаге к отобранным наборам спискам применяются генетические операторы – кроссинговер или мутация.

Шаг 4. Формируется новый набор альтернативных решений размещения деталей на полосу. Отбирается лучший приоритетный список по целевой функции и строится карту размещения элементов для этого набора. Тот набор, у которой после выполнения всех итераций значение целевой функции минимальное, выводится как наилучшее найденное решение.

Структуру генетического алгоритма можно показать в виде блок-схемы, изображенной на рис. 6.

Разработанная программа с использованием генетического алгоритма анализирует размещение предметов с помощью трех алгоритмов укладки – NFDL, LB, FCNR.

Для анализа работы алгоритма были рассмотрены известные тестовые примеры с их оптимальными решениями. Задача решалась на семи категориях входных данных различной размерности в пределах от 16 до 197 (под размерностью понимается количество элементов) [1].



Рис. 6 – Блок-схема реализованного генетического алгоритма

Fig. 6 – Implemented genetic algorithm block diagram

Результаты, полученные при решении задачи, сведены в табл. 1.

Таблица 1. Сравнение оптимального результата с результатом, полученным при решении генетическим алгоритмом

Table 1. Comparison of the optimal result with the result obtained by a genetic algorithm

Категория Category	Количество элементов Number of el- ements	Ширина по- лосы Band- width	Известный оптимальный результат Known optimal result	Результаты, полученные генетическим алгоритмом Results obtained by the genetic algorithm		
				NFDL	LB	FFDL
I	16	20	20	36	21	35
	17	20	20	37	20	43
	16	20	20	35	21	34
II	25	40	15	23	17	29
	25	40	15	22	26	35
	25	40	15	23	19	20
III	28	60	30	47	32	64
	29	60	30	46	31	69
	28	60	30	56	32	52
IV	49	60	60	109	74	214
	49	60	60	116	71	178
	49	60	60	114	70	166
V	72	60	90	174	95	215
	72	60	90	195	96	259
	72	60	90	172	99	305
VI	97	80	120	266	131	441
	97	80	120	301	124	460
	97	80	120	299	128	329
VII	196	160	240	530	200	1433
	197	160	240	735	255	1202
	196	160	240	628	252	1277

Как видно из результатов (табл.1), генетический алгоритм в комбинации с процедурой упаковки BL находит рациональное решение для рассмотренных примеров.

Для дальнейшего исследования поставленной задачи целесообразно выбирать данную комбинацию методов. А для более эффективных результатов использовать различные усложнения генетических операторов при формировании альтернативного решения.

Для наглядности приведем на рис. 7 гистограмму, где сравним полученные результаты с известными оптимальными. Очевидно, что генетический алгоритм в комбинации

с декодером «Нижний левый» на заданном наборе элементов и при заданных условиях показывает близкие к оптимальному результаты.

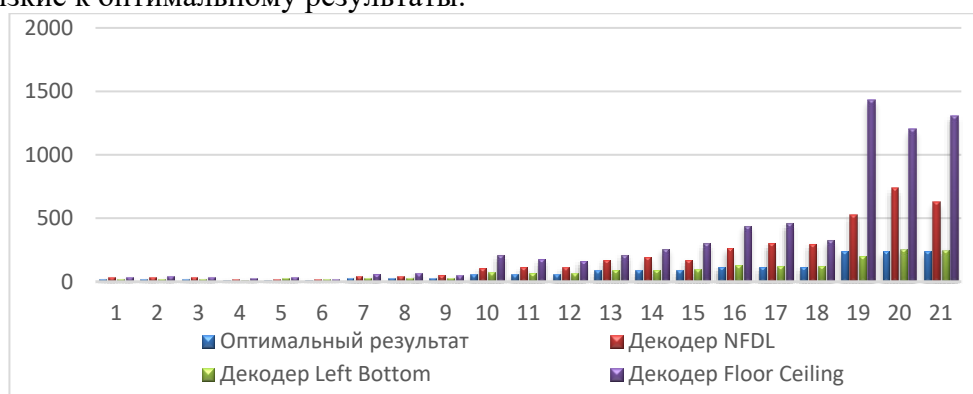


Рис. 7 – Сравнение оптимальных и полученных результатов

Fig. 7 – Comparison of optimal and obtained results

Вывод. Рассматриваемая задача относится к классу NP-трудных задач. Это повышает интерес к нахождению метода решения задачи. Использование точных методов для решения подобных задач является неэффективным, так как сложность используемого точного метода повышается экспоненциально. В связи с этим возникает необходимость разработки и применения приближенных методов, к которым относятся и эвристические алгоритмы. С помощью них, в свою очередь, становится возможным находить субоптимальное решение за приемлемое время. С учетом этого, использование метаэвристических методов для решения данных задач становится эффективным подходом. Акцент в развитии методов направлен на создание математического аппарата на основе единого подхода к описанию задач и на разработку практических моделей и методов для решения задач в определенных технологических условиях.

Задачи упаковки хорошо решают проблему экономии ресурсов, которая является актуальной и современной для всех сфер человеческой деятельности. Высокая сложность и обширная область практического применения задач упаковки обеспечивают актуальность проблемы создания эффективных эвристических и метаэвристических алгоритмов для решения данных задач. Для выявления перспективных путей развития данного направления необходимо проведение дальнейших исследований поведения системы на различных категориях задач и применением различных комбинаций алгоритмов. Это позволит увеличить эффективность применяемых методов и оптимизировать использование ресурсов.

Актуальным направлением исследования так же являются двумерные задачи – упаковка деталей в контейнеры, где появляются дополнительные ограничения по укладке, а именно высота и ширина используемого контейнера для упаковки, а также их количество. Разработка и использование программного обеспечения на основе разработанных метаэвристических алгоритмов актуальна для как проектирования отдельных производственных процессов, так и для сложных систем автоматизации производственного цикла.

Библиографический список:

1. Гладков, Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы // ООО Издательская фирма "Физико-математическая литература", 2009, – 320 с.
2. Майрамты М.В. Сравнительный анализ работы некоторых метаэвристических методов при решении задачи упаковки в полубесконечную полосу // Естественные и технические науки: актуальные вопросы : Сборник статей V Международной научно-практической конференции, 2018, С.13-19.
3. A Novel Genetic Algorithm for the Three-Dimensional Bin Packing Problem with Rotations – L.-Y. Chen, C.-H. Wu. <https://ieeexplore.ieee.org/document/8029218>
4. Валиахметова, Ю.И., Филиппова А.С. Теория оптимального использования ресурсов Л.В. Канторовича в задачах раскроя-упаковки: обзор и история развития методов решения // Вестник Уфимского государственного авиационного технического университета. 2014, Т. 18, № 1(62). – С. 186-197.
5. Sergievskiy M., Syroezhkin S. Use of genetic algorithms for solving problems of optimal cutting // 6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation. ITM Web of Conferences, 2016. <https://www.semanticscholar.org/author/M.-Sergievskiy/70370738>

6. Майрамты М.В. Анализ некоторых метаэвристических методов при решении задачи упаковки в полубесконечную полосу // XII Международной научно-практической конференции «Научное сообщество студентов XXI столетия. Технические науки», 2020, С. 23-29.
7. Тимофеева О.П., Чернышева Т.Ю., Корелин О.Н., Волков А.В. Генетический алгоритм в оптимизации трехмерной упаковки блоков в контейнер // Информатика и управление в технических и социальных системах, 2017, [Интернет-ресурс]. <https://cyberleninka.ru/article/n/geneticheskiy-algoritm-v-optimizatsii-trehmernoy-upakovki-blokov-v-konteyner/viewer> (дата обращения 21.06.2025 г.).
8. Файзрахманов Р.И. Конструктивный вероятностный алгоритм для задачи размещения кругов и прямоугольников / Уфа: Изд-во «Вестник УГАТУ». – 2010. – № 4 (39) –С. 132–138.
9. Yuliia P., Kaidan M., Tchaikovskiy I., Pleskanka M. Research of Genetic Algorithms for Increasing the Efficiency of Data Routing// 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019.
10. Дивеев А.И., Шмалько Е.Ю. Решение задачи двумерной упаковки методом вариационного генетического алгоритма // Cloud of Science, 2016, Т. 3, № 3.- С. 380-395.

References:

1. Gladkov, L.A., Kuraychik V.V., Kuraychik V.M. Genetic algorithms. OOO Publishing company "Physical-mathematical literature", 2009:320 p. (In Russ)
2. Mairamty M.V. Comparative analysis of the work of some metaheuristic methods in solving the problem of packaging in a semi-infinite range. *Natural and technical sciences: current issues : Collection of articles of the V International scientific-practical conference*, 2018;13-19. (In Russ)
3. A Novel Genetic Algorithm for the Three-Dimensional Bin Packing Problem with Rotations – L.-Y. Chen, C.-H. Wu. <https://ieeexplore.ieee.org/document/8029218>
4. Valiakmetova, Y.I., Filipova, A.S. Theory of optimal use of resources by L.V. Kantorovich in the tasks of cutting-packing: review and history of development of methods of solution. *Bulletin of the Ufa State Aviation Technical University*. 2014;18(1)(62):186-197. (In Russ)
5. Sergievskiy M., Syroezhkin S. Use of genetic algorithms for solving problems of optimal cutting. 6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation. ITM Web of Conferences, 2016. <https://www.semanticscholar.org/author/M.-Sergievskiy/70370738>
6. Mairamty M.V. Analysis of some metaheuristic methods in solving the problem of packaging in a semi-infinite band. XII International scientific and practical conference "Scientific community of students of the XXI century. Technical sciences", 2020:23-29. (In Russ)
7. Timofeeva O.P., Chernysheva T.Y., Koroline O.N., Volkov A.V. Genetic algorithm in optimization of three-dimensional packing of blocks in the container // Informatics and management in technical and social systems, 2017, [Internet-resource]. <https://cyberleninka.ru/article/n/geneticheskiy-algoritm-v-optimizatsii-trehmernoy-upakovki-blokov-v-konteyner/viewer> (date of request 21.06.2025 г.). (In Russ)
8. Faizrahmanov R.I. Constructive probabilistic algorithm for the problem of placement of circles and rectangles. *Ufa: Edition "The Messenger of AGU."* 2010; 4 (39):132-138. (In Russ)
9. Yuliia P., Kaidan M., Tchaikovskiy I., Pleskanka M. Research of Genetic Algorithms for Increasing the Efficiency of Data Routing. 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019.
10. Diveev A.I., Shmaslova E.Y. Solving the problem of two-dimensional packaging by the method of a genetic variation algorithm. *Cloud of Science*, 2016;3(3): 380-395. (In Russ)

Сведения об авторе:

Мария Васильевна Майрамты, ассистент, аспирант кафедры «Компьютерное моделирование и автоматизация проектирования»; mairamty95@mail.ru

Information about author:

Maria V. Mairamty, Assistant, Postgraduate Student, Department "Computer modeling and automation of design"; mairamty95@mail.ru

Конфликт интересов/Conflict of interest

Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.

Поступила в редакцию/Received 17.05.2025.

Одобрена после/рецензирования Revised 01.07.2025.

Принята в печать/Accepted for publication 29.08.2025.