

ISSN 2949-5598

# ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 1 2026

Новосибирск  
Издательство Института математики

## ПОЛУДИСКРЕТНОЕ ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ ОБЪЕКТОВ В ЗАДАЧЕ ДВУМЕРНОЙ УПАКОВКИ

*В. В. Акентьев*

Новосибирский гос. университет,  
ул. Пирогова, 2, 630090 Новосибирск, Россия  
E-mail: akentevvv01@gmail.com

**Аннотация.** Рассматривается задача двумерной упаковки (nesting problem), заключающаяся в размещении набора объектов произвольной формы в ограниченную полосу с целью минимизации её используемой длины. Одним из ключевых этапов решения подобных задач является выбор способа моделирования геометрических объектов, который определяет как точность аппроксимации их формы, так и вычислительные затраты алгоритма упаковки. Предлагается новый метод моделирования объектов на основе адаптивных трапеций. В отличие от актуального метода — сегментного подхода — он позволяет значительно снизить избытки аппроксимации, возникающие при несовпадении вершин с узлами сетки, и повысить точность представления формы объектов. Проведённые вычислительные эксперименты на типовых наборах данных ESICUP показали, что использование предложенного метода при выполнении алгоритма упаковки вниз-влево (bottom-left-fill) обеспечивает более плотное размещение объектов при умеренном увеличении вычислительных затрат. Полученные результаты подтверждают эффективность предложенного метода и его практическую применимость в задачах двумерной упаковки. Табл. 2, ил. 11, библиогр. 11.

**Ключевые слова:** задача раскроя, задача упаковки, полудискретное кодирование, алгоритм нижнего левого заполнения.

### Введение

Задача двумерной упаковки объектов произвольной формы (nesting problem) имеет широкий спектр практических приложений. Она возникает в текстильной, металлургической, стекольной и машиностроительной промышленности, где требуется рациональное использование материала при раскрое или компоновке деталей. В литературе данная задача

рассматривается как частный случай общей задачи резки и упаковки (cutting and packing, C&P). В отличие от других подзадач C&P, в задаче вложения рассматриваются как выпуклые, так и невыпуклые многоугольники. Учёт последних существенно усложняет решение по сравнению с постановками, ограниченными выпуклыми объектами.

Формально задача вложения заключается в размещении множества двумерных простых (самонепересекающихся) многоугольников внутри заданных контейнеров при условии отсутствия пересечений между объектами. В качестве контейнеров могут рассматриваться как ограниченные многоугольные области, так и неограниченная по длине полоса. Конкретная целевая функция определяется прикладной областью и может включать минимизацию используемой длины полосы, минимизацию числа контейнеров и другие критерии. Ключевая вычислительная сложность задачи связана с проверкой пересечений объектов произвольной формы, что требует применения эффективных методов вычислительной геометрии и во многом определяет масштабируемость алгоритма.

Производительность алгоритма решения во многом определяется способом моделирования объектов. Существующие подходы условно делятся на два класса:

- точные (методы на основе  $\text{fit polygon}$ ,  $\varphi$ -функций [1] и др.);
- дискретные (пиксельный [2], полосный [3], сегментный [4] методы и др.).

Точные методы обеспечивают корректное геометрическое представление и отсутствие ошибок аппроксимации, однако характеризуются высокими вычислительными затратами и слабой масштабируемостью. Дискретные методы, напротив, позволяют значительно ускорить выполнение основных операций (проверку пересечения, поиск позиции для размещения и само размещение), что особенно важно в эвристических и метаэвристических алгоритмах, где качество решения напрямую зависит от количества итераций. Однако повышение скорости достигается ценой снижения точности аппроксимации формы объектов, что в ряде случаев может приводить к ухудшению качества упаковки.

В последние годы внимание исследователей привлекли полудискретные методы, в которых дискретизация выполняется только по одному направлению. Такой подход обеспечивает баланс между скоростью и точностью [5]. В частности, в [6] показано, что применение полудискретного сегментного кодирования позволяет достигать высокой производительности при умеренных искажениях формы. Однако данный метод чувствителен к выбору шага сетки: если вершины объектов не совпадают с узлами сетки, то аппроксимация может давать значительные избытки, что приводит к ухудшению решения.

В настоящей работе предлагается новый полудискретный метод кодирования объектов, основанный на использовании адаптивных трапеций. Он призван устранить указанный недостаток сегментного подхода, повысив точность аппроксимации без существенного увеличения вычислительных затрат. Целью работы является разработка и исследование предложенного метода, а также экспериментальное сравнение его с существующим сегментным кодированием в контексте применения в жадных алгоритмах двумерной упаковки.

Работа организована следующим образом. В разд. 1 рассматривается историческое развитие дискретных методов. Ограничения актуального сегментного подхода описаны в разд. 2. В разд. 3 представлены новый метод кодирования и его особенности. Результаты вычислительных экспериментов приведены в разд. 4, а в заключении сформулированы основные выводы и направления дальнейших исследований.

## 1. Обзор дискретных кодировок

Важным направлением исследований в области задачи вложения является разработка дискретных кодировок, ориентированных на достижение точной аппроксимации форм объектов при сохранении высокой вычислительной эффективности. Под кодировкой в данном контексте понимается дискретное представление геометрического объекта в виде конечного множества элементов регулярной структуры, позволяющее выполнять основные операции над объектами — проверку пересечений, смещение и размещение — в упрощённой форме. Применение таких кодировок сводит сложные геометрические операции к работе с более простыми структурами, что существенно облегчает проверку пересечений по сравнению с точными методами.

В процессе решения задачи работа ведётся непосредственно с кодировками объектов: объекты считаются пересекающимися, если пересекаются их представления. Поскольку пересечения объектов недопустимы, кодировка должна полностью покрывать исходный объект. При этом упрощения, возникающие в процессе аппроксимации, могут приводить к появлению дополнительных областей — избытков аппроксимации, вследствие чего аппроксимация оказывается несколько увеличенной по сравнению с исходным объектом.

Поскольку большинству дискретных методов построения кодировки характерно использование одномерной или двумерной сетки, её шаг  $R$  выступает параметром, определяющим точность аппроксимации. В типичных случаях уменьшение шага делает дискретное представление ближе к исходной форме объекта и расширяет пространство допустимых решений. Однако такое поведение наблюдается не всегда: если шаг сетки

изначально согласован с геометрией объекта, то его дальнейшее уменьшение может приводить к появлению избыточных областей аппроксимации и, как следствие, к ухудшению точности представления. Тем самым уменьшение шага неизбежно увеличивает вычислительные затраты. Поэтому актуальной задачей является разработка подходов, позволяющих достичь приемлемой точности аппроксимации даже при сравнительно крупном шаге сетки.

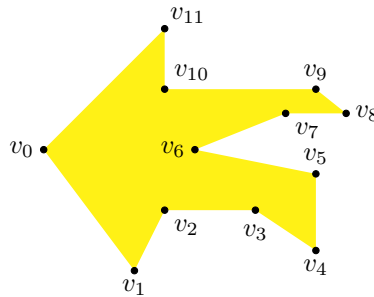


Рис. 1. Исходный объект

Ниже представлены основные подходы дискретного моделирования на примере объекта, приведённого на рис. 1, отражающие последовательное развитие идей и методов к их кодированию.

• **Бинарное кодирование**, предложенное в 1993 г. [2], заключается в построении двумерной сетки и формировании массива, в котором каждой ячейке соответствует клетка сетки. Если клетка или её граница пересекается с объектом, то в соответствующей ячейке массива фиксируется единица, в противном случае — нуль. Таким образом формируется пиксельная аппроксимация объекта (рис. 2а, 2б).

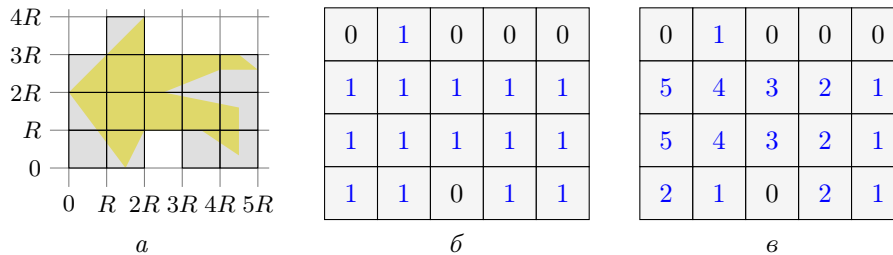


Рис. 2. Методы бинарного кодирования и кодирования переходов:

- а) аппроксимация объекта на сетке; б) массив бинарного кодирования;  
в) массив кодирования переходов

- **Кодирование переходов**, предложенное в 2001 г. [7], представляет собой развитие бинарного подхода. На основе бинарного представления объекта формируется массив тех же размеров, где каждая ячейка получает значение, равное нулю, если соответствующая ячейка в бинарной матрице равна нулю; в противном случае указывается число ячеек, на которое надо переместиться вправо до первой нулевой ячейки или конца массива.

Такой способ кодирования не изменяет геометрическое представление кодировки, но позволяет ускорить проверку пересечений по сравнению с бинарным кодированием, поскольку каждая ячейка хранит дополнительную информацию о расстоянии до области, не содержащей объекта (рис. 2в).

- **Полосное кодирование**, предложенное в 2007 г. [3], является полудискретным методом, так как использует одномерную сетку для кодирования. Объект аппроксимируется множеством прямоугольников, пара сторон каждого из которых опирается на соседние узлы сетки, а объединение всех прямоугольников полностью покрывает объект.

Метод на порядки ускоряет выполнение основных операций по сравнению с двумя описанными выше подходами. Кроме того, полудискретизация обеспечивает более точное приближение формы объектов, что положительно сказывается на качестве решений. Пример полосной аппроксимации объекта приведён на рис. 3.

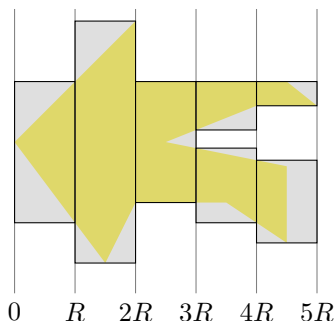


Рис. 3. Полосный метод кодирования

- **Сегментное кодирование**, впервые представленное в 2013 г. [4] и подробно описанное в [6], представляет собой актуальный полудискретный подход. В отличие от полосного метода, здесь для аппроксимации используются трапеции (включая вырожденные случаи с основанием нулевой длины). Основания трапеций располагаются на соседних узлах сетки, а боковые стороны касаются границы объекта. Для обеспечения корректности на трапеции накладывается ограничение: пересечение

$y$ -проекций оснований трапеции на любой узел сетки не пусто. Если допустимы несколько вариантов угла наклона боковой стороны, то выбирается прямой угол.

Элементами кодировки выступают основания трапеций, представляемые сегментами, которым присваиваются метки трёх типов: L (left), R (right) или M (middle), в зависимости от их положения и взаимного пересечения. Так, левый (правый) сегмент трапеции получает метку L (R). Если пересекаются два однотипных сегмента, то они объединяются в один с тем же типом. В случае пересечения разнотипных сегментов соответствующая общая часть выделяется в отдельный сегмент с меткой M, тогда как оставшиеся непересекающиеся участки сохраняют исходную маркировку (рис. 4).

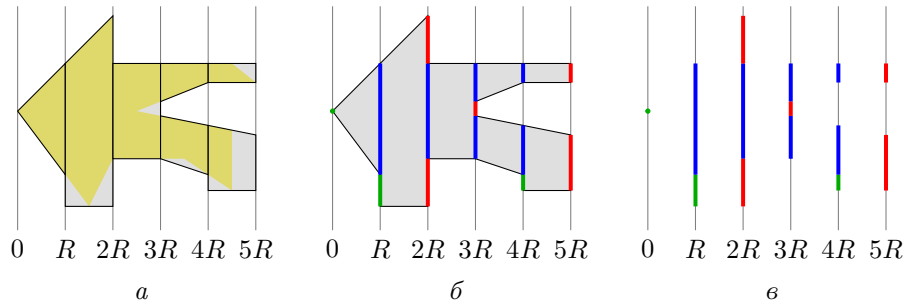


Рис. 4. Сегментный метод кодирования (сегменты L — зелёные, M — синие, R — красные): а) аппроксимация объекта трапециями; б) построение сегментной кодировки; в) сегментная кодировка

В предыдущих методах кодировки объекты считались пересекающимися, если пересекалась хотя бы одна пара их элементарных составляющих. В предлагаемом подходе сохраняется тот же принцип, однако допускается пересечение сегментов типов L и R, что интерпретируется как касание объектов без наложения.

Данный метод обеспечивает существенно более высокую точность аппроксимации по сравнению с предыдущим подходом при сопоставимых вычислительных затратах на выполнение основных операций. Благодаря этому сегментное кодирование в последние годы стало основой для разработки эффективных эвристических алгоритмов решения задачи вложения.

Для более наглядной интерпретации метода в данной работе дополнительно введён шаг, описывающий построение трапеций, что также позволяет интуитивно объяснить корректность сегментного представления.

## 2. Недостатки сегментного кодирования

У актуального полудискретного способа кодирования сохраняется существенный недостаток, связанный с появлением избыточных областей аппроксимации. Эти излишки возникают в тех случаях, когда вершины многоугольника не совпадают с узлами сетки: при дискретизации части фигуры аппроксимация выходит за границы оригинала, что снижает точность представления и может ухудшать качество размещения. Пример возникновения таких избыточных областей показан на рис. 5.

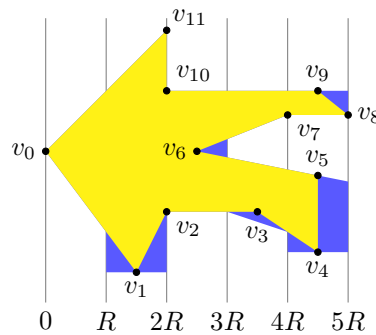


Рис. 5. Избытки аппроксимации сегментной кодировки

Данную проблему можно устранить подбором шага сетки так, чтобы все абсциссы вершин делились на этот шаг, однако требование одновременной целочисленной кратности множества произвольных координат, как правило, неразрешимо на практике. Альтернативой является уменьшение шага для повышения точности аппроксимации. При этом размер кодировки и вычислительная сложность основных операций возрастают пропорционально уменьшению шага, что сводит на нет преимущество полудискретного подхода по вычислительной эффективности.

Для минимизации описанного недочёта разработан новый метод кодирования, который частично снимает проблему избыточной аппроксимации с сохранением высокой вычислительной эффективности; его описание приведено ниже.

## 3. Кодирование адаптивными трапециями

Идея предлагаемого способа кодирования развивает сегментное представление и базируется на той же аппроксимации объекта набором (аппроксимирующих) трапеций, основания которых лежат на соседних узлах сетки, а боковые стороны касаются границ исходного многоугольника, пример которой приведён на рис. 6а.



Ключевое отличие заключается в параметризации выбора угла наклона боковой стороны трапеции в случаях, когда он неоднозначен. Жёсткое правило фиксации прямого угла заменяется параметрическим подходом, при котором значение угла определяется динамически в процессе выполнения основных операций с учётом текущей компоновки объектов.

В качестве элементов кодировки выступают адаптивные трапеции: статичные и нестатичные. Это части аппроксимирующих трапеций, которые не допускают и допускают изменения во время выполнения основных операций соответственно. Статичные трапеции формируются из аппроксимирующих, которые либо не содержат вершин кодируемого объекта (см. трапеции  $T_1$  и  $T_5$  на рис. 6б), либо включают только вершины, соответствующие локально вогнутым углам — вогнутым углам многоугольника, образуемого пересечением этой трапеции с кодируемым объектом (например, трапеции  $T_3$  и  $T_4$ ). Оставшиеся аппроксимирующие трапеции ( $T_2$ ,  $T_6$  и  $T_7$ ), содержащие иные типы вершин, относятся к нестатичным и могут изменяться в процессе выполнения операций. При этом на них накладывается ограничение, что даже после адаптации они обязаны полностью содержать соответствующий фрагмент исходного объекта.

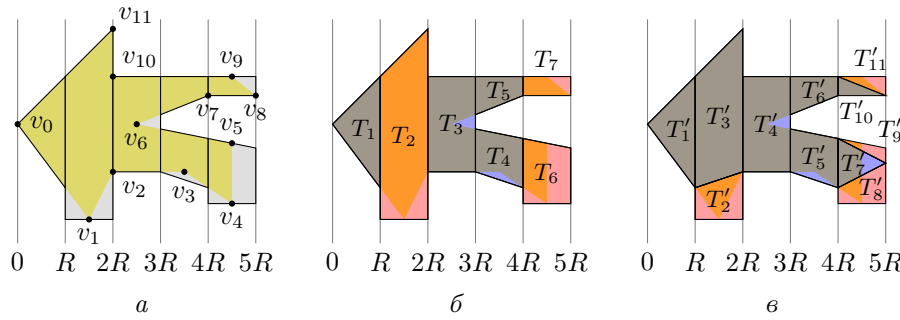


Рис. 6. Построение кодировки адаптивными трапециями (статичные трапеции синие, нестатичные — красные):

- а) аппроксимация объекта трапециями; б) классификация трапеций;  
в) разделение нестатичных трапеций

Для повышения эффективности выполнения базовых операций размеры нестатичных трапеций уменьшаются путём их разбиения на статичную и нестатичные части, при этом статичная часть выбирается максимально возможной. В результате из одного аппроксимирующего элемента выделяются статичная и одна либо две нестатичные части в зависимости от геометрии аппроксимируемой области.

- Пересечение двух узлов сетки. Если аппроксимируемая часть объекта пересекает оба соседних узла сетки, то разбиение порождает одну статичную и одну нестатичную трапеции. Примером служит трапеция  $T_2$

(рис. 6б), которая разбивается на трапеции  $T'_2$  и  $T'_3$  (рис. 6в). При этом если бы вершина  $v_{11}$  лежала между узлами  $R$  и  $2R$  (аналогично вершине  $v_1$ ), то была бы образована ещё одна нестатичная трапеция.

- Двукратное пересечение одного узла. Если аппроксимируемая часть дважды пересекает один и тот же узел сетки, то разбиение приводит к одной статичной и двум нестатичным трапециям. В этом случае выбор основания статичной трапеции со стороны непересекаемого узла является неоднозначным; для устранения неоднозначности и снижения избытков аппроксимации это основание задаётся точкой, ордината которой совпадает с серединой второго основания данной трапеции. Пример иллюстрируется на рис. 6б трапецией  $T_6$ , которая после разбиения образует трапеции  $T'_7$ ,  $T'_8$  и  $T'_9$  (см. рис. 6в).

В обоих случаях у каждой нестатичной трапеции фиксируется одна боковая сторона — та, что непосредственно соприкасается со статичной трапецией, полученной в результате разбиения.

Статичная трапеция однозначно определяется парой своих оснований, тогда как нестатичная адаптивная трапеция характеризуется не только основаниями, но и дополнительными параметрами: множеством точек адаптации, представляющих собой вершины кодируемого объекта вне узлов сетки, лежащие внутри трапеции или на её подвижной боковой стороне, относительно которых выполняется адаптация, а также фиксируемой стороной (верхней или нижней), общей со статичной частью.

**3.1. Построение нестатичной адаптивной трапеции.** Опишем алгоритм построения структуры данных, соответствующей нестатичной адаптивной трапеции. Рассмотрим пару трапеций — статичную и нестатичную, образовавшихся в результате разбиения одной аппроксимирующей трапеции, а также фрагмент объекта, который они совместно кодируют (рис. 7а) в ячейке  $i$ . Как отмечалось ранее, фиксируемой стороной нестатичной трапеции является сторона, общая со статичной частью. Далее изложение ведётся для случая, когда фиксированным является верхнее основание (рис. 7); алгоритм для варианта с фиксированным нижним основанием строится аналогично.

**Определение точек адаптации.** Из множества всех вершин кодируемого объекта выберем только лежащие внутри адаптивной трапеции либо принадлежащие её подвижной боковой стороне. К ним добавим вершины трапеции  $s_l$  и  $s_r$ , принадлежащие фиксируемой боковой стороне. *Точками адаптации* назовём последовательность вершин  $V = (v_1, \dots, v_n)$  выпуклой оболочки полученного множества, являющихся вершинами кодируемого объекта и упорядоченных от левой фиксированной точки  $s_l$  к правой  $s_r$ . В примере на рис. 7 таковыми являются

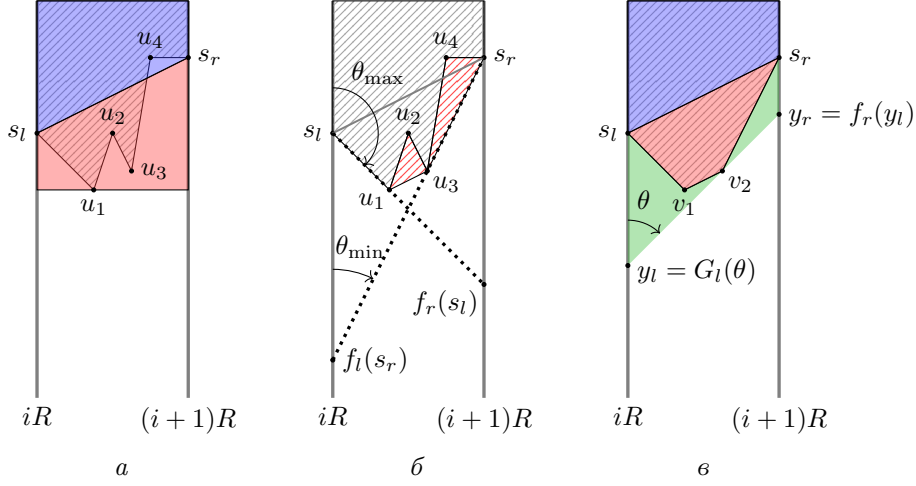


Рис. 7. Построение структуры нестатичной адаптивной трапеции:

- а) кодируемый фрагмент и аппроксимирующие трапеции;
- б) определение точек адаптации и параметров нестатичной трапеции;
- в) демонстрация перестроения подвижной стороны

вершины  $u_1$  и  $u_3$ , переименованные далее в  $v_1$  и  $v_2$  соответственно. Если множество точек адаптации пусто, то данный фрагмент полностью аппроксимируется ранее определённой статичной трапецией, и построения нестатичной трапеции не требуется.

**Функции перестроения подвижной боковой стороны.** Для корректного обновления положения подвижной боковой стороны вводятся функции

$$f_r(y_l) = \min_{v \in V} \left( y_l + \frac{v_y - y_l}{v_x - iR} R \right), \quad (1)$$

$$f_l(y_r) = \min_{v \in V} \left( y_r - \frac{v_y - y_r}{v_x - (i+1)R} R \right). \quad (2)$$

Функция (2) (соответственно функция (1)) по  $y$ -координате левой (правой) подвижной точки возвращает  $y$ -координату правой (левой) подвижной точки таким образом, чтобы прямая, проходящая через эти две точки, являлась касательной к аппроксимируемому фрагменту объекта.

**Адаптация по углу наклона.** Положение подвижной стороны задаётся углом её наклона. Для этого вводится функция, связывающая угол наклона с координатой левой подвижной точки трапеции:

$$g_l(\theta) = \min_{v \in V} (v_y + \text{ctg } \theta (v_x - iR)). \quad (3)$$

Угол наклона боковой стороны, проходящей через две точки  $u$  и  $v$ , определяется функцией

$$\Phi(v, u) = \operatorname{arctg} \left( \frac{v_y - u_y}{v_x - u_x} \right). \quad (4)$$

**Особые углы адаптации.** На основе множества точек адаптации формируется множество *особых углов адаптации*  $\Theta$ , описывающее все положения подвижной боковой стороны, при которых она становится касательной к одной из сторон выпуклой оболочки аппроксимируемого фрагмента:

$$\Theta = \{\Phi(s_l, v_1)\} \cup \bigcup_{i=1}^{n-1} \{\Phi(v_i, v_{i+1})\} \cup \{\Phi(v_n, s_r)\}, \quad v_j \in V. \quad (5)$$

Максимальное и минимальное значения угла наклона подвижной стороны достигаются соответственно при  $\Phi(s_l, v_1)$  и  $\Phi(v_n, s_r)$ ; далее они обозначаются через  $\theta_{\max}$  и  $\theta_{\min}$ .

**Ограничение области адаптации.** Чтобы обеспечить корректную адаптацию боковой стороны, вводится обобщение функции, ограничивающее угол её наклона в пределах допустимой области:

$$G_l(\theta) = \begin{cases} g_l(\theta_{\min}), & \text{если } \theta < \theta_{\min}, \\ g_l(\theta), & \text{если } \theta_{\min} \leq \theta \leq \theta_{\max}, \\ g_l(\theta_{\max}), & \text{если } \theta_{\max} < \theta. \end{cases} \quad (6)$$

На рис. 8 показана визуализация двух из множества возможных представлений кодировки адаптивными трапециями для объекта с рис. 1.

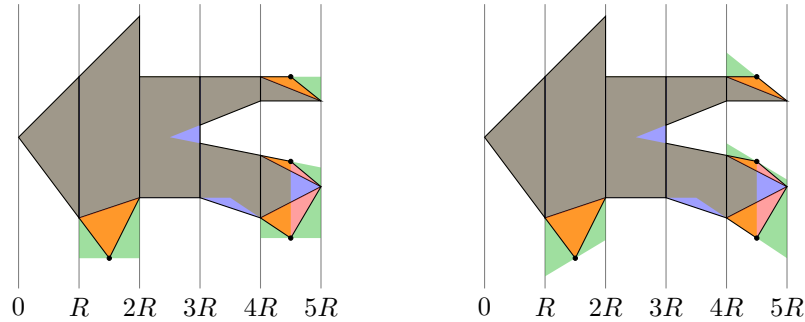


Рис. 8. Кодирование адаптивными трапециями:  
статичные трапеции синие, фиксированная (изменяемая) область  
нестатичных трапеций — красная (зелёная)

**3.2. Проверка пересечений.** Опишем алгоритм проверки пересечений пар адаптивных трапеций, находящихся в одной ячейке. Рассмотрим три возможных случая: пара статичных трапеций, комбинация статичной и нестатичной трапеции и пара нестатичных трапеций.

1. Пара статичных трапеций. Две статичные трапеции пересекаются тогда и только тогда, когда пересекается хотя бы одна пара их оснований или их боковые стороны.

2. Статичная и нестатичная трапеции. Если нестатичная трапеция зафиксирована сверху, то её нижняя сторона адаптируется по верхнему углу статичной трапеции, после чего пересечение определяется по критерию, аналогичному случаю двух статичных трапеций. В случае, когда нестатичная трапеция зафиксирована снизу, аналогично адаптируется её верхняя сторона.

3. Пара нестатичных трапеций. Возможны два случая: обе трапеции зафиксированы с одной стороны или с разных сторон.

Если обе трапеции зафиксированы с одной стороны (например сверху), то первая адаптируется по углу фиксированной боковой стороны второй трапеции, а вторая — по углу фиксированной стороны первой. Далее для адаптированных трапеций пересечение проверяется по критерию для пары статичных.

В случае фиксации трапеций с разных сторон они не пересекаются, если среди объединения множеств их особых углов существует угол, при котором адаптированные трапеции не пересекаются.

Следует отметить, что благодаря особенностям алгоритма построения кодировки количество нестатичных трапеций в представлении объекта ограничено сверху числом его вершин. Это, в свою очередь, ограничивает число сравнительно затратных вычислительных операций адаптации по углу, выполняемых для пары нестатичных трапеций для любого шага сетки. Таким образом, при уменьшении шага сетки рост вычислительных затрат обусловлен преимущественно менее ресурсоёмкими операциями, что обеспечивает масштабируемость метода относительно шага сетки.

**3.3. Размещение.** Опишем процедуру размещения предметов в контейнере, основанную на жадном алгоритме заполнения вниз-влево (bottom-left-fill), который является базовым для большинства эвристических алгоритмов.

Пусть заданы частично заполненный контейнер и предмет. Размещение осуществляется последовательным поиском первой допустимой позиции в кодировке контейнера, начиная из нижнего левого угла. Для каждой проверяемой позиции выполняются следующие шаги.

ШАГ 1. Производится проверка на наличие пересечения адаптивных трапеций размещаемого предмета с трапециями, принадлежащими уже размещённым объектам. Если пересечений не обнаружено, то предмет фиксируется в текущем положении.

ШАГ 2. В случае наличия пересечений предмет сдвигается в новую позицию вдоль сетки на минимальное расстояние, при котором найденные пересекающиеся адаптированные трапеции соприкасаются, не образуя перекрытий.

ШАГ 3. Если после сдвига предмет выходит за рамки текущей ячейки, то поиск позиции продолжается с начала следующей ячейки.

Если при этом кодировка предмета выходит за пределы контейнера по ячейкам, то предмет считается неразмещаемым в данной конфигурации упаковки.

Для формирования общей упаковки используется классический жадный алгоритм *bottom-left-fill*. На первом этапе предметы сортируются в невозрастающем порядке по площади, после чего последовательно размещаются с использованием описанной выше процедуры.

**3.4. Сравнение с сегментным способом кодирования.** Как отмечалось ранее, актуальный полудискретный способ кодирования имеет недостаток, связанный с избытками аппроксимации, возникающими в случаях, когда вершины кодируемого объекта не совпадают с узлами сетки. Предлагаемый метод частично сохраняет данную проблему, однако выражена она значительно слабее. Избытки аппроксимации наблюдаются лишь в двух случаях:

1) для вершин, соответствующих локально вогнутым углам (например, вершины  $v_3$  и  $v_6$  на рис. 6);

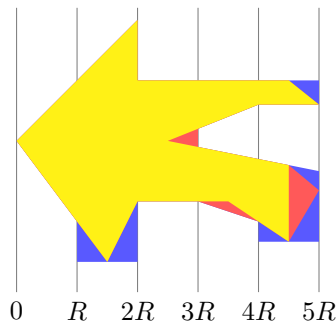


Рис. 9. Избытки аппроксимаций: избытки сегментного кодирования красные и синие, избытки кодирования адаптивными трапециями — красные

2) для фрагментов объекта, пересекающих сетку ровно в двух точках (пример подобного фрагмента показан на рис. 6б и соответствует трапеции  $T_6$ ).

Во всех остальных случаях избытки аппроксимации полностью устранены.

На рис. 9 представлено наглядное сравнение избытков аппроксимации для объекта, показанного на рис. 1, полученных при использовании сегментного способа и кодирования с помощью адаптивных трапеций. Из иллюстрации видно, что применение адаптивных трапеций позволяет существенно сократить избытки аппроксимации по сравнению с исходным методом, что приводит к более точному воспроизведению формы объекта.

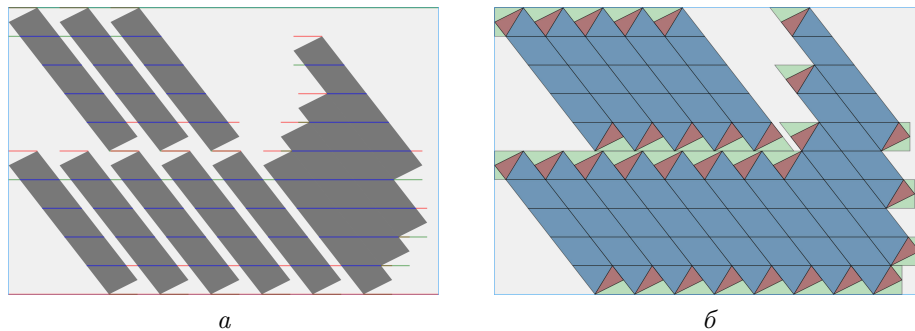


Рис. 10. Сравнение результатов упаковки при использовании различных способов кодирования: а) сегментное кодирование; б) кодирование адаптивными трапециями

На рис. 10 приведён пример упаковки одинаковых прямоугольников алгоритмом bottom-left-fill, использующим различные способы кодирования. Как видно из иллюстрации, при применении предложенного метода удалось разместить 16 прямоугольников, тогда как при сегментном способе — лишь 12. Помимо увеличения числа размещённых объектов новый метод обеспечивает более плотное прилегание сторон многоугольников, что особенно важно при практическом использовании в задачах резки и раскроя на этапе резки. Этот пример наглядно демонстрирует, что снижение избытков аппроксимации непосредственно повышает эффективность упаковки и качество использования доступного пространства.

Таким образом, предложенный метод, во-первых, повышает точность аппроксимации и, как следствие, обеспечивает более плотное размещение объектов, а во-вторых, значительно упрощает задачу подбора шага сетки, поскольку обладает меньшей чувствительностью к вершинам, не совпадающим с её узлами.

#### 4. Результаты вычислительных экспериментов

В рамках настоящего раздела осуществляется сравнение двух методов кодирования объектов задачи двумерной упаковки: сегментного метода кодирования (segment encoding, SE) и метода кодирования адаптивными трапециями (adaptive trapezoid encoding, ATE). Оба метода реализованы на языке C++ и протестированы на процессоре Intel Core i5-10400F. Сравнение подходов осуществляется по результатам, полученным с использованием алгоритма bottom-left-fill, описанным выше.

Вычислительные эксперименты проводились на следующих типовых наборах данных ESICUP: Shirts из [9], Han из [10], а также Trousers и Swim из [11]. Эти наборы включают как выпуклые, так и невыпуклые фрагменты различных размеров. Для оценки масштабируемости дополнительно создан набор Random500, содержащий 500 случайно сгенерированных объектов разной формы и размеров. Каждый объект в этом наборе имеет от одной до трёх копий. Контейнером в экспериментах служил лист фиксированной ширины и неограниченной длины. Во всех тестах не допускались повороты и отражения объектов. Характеристики наборов данных приведены в табл. 1.

Таблица 1

Информация по используемым тестам

| Тест      | Общее число предметов | Число уникальных предметов | Среднее число вершин предметов | Ширина контейнера | Интервал длины $x$ -проекции предметов | $R_b$ |
|-----------|-----------------------|----------------------------|--------------------------------|-------------------|----------------------------------------|-------|
| Han       | 23                    | 20                         | 7,35                           | 58                | [1; 19]                                | 1     |
| Shirts    | 99                    | 8                          | 6,05                           | 40                | [1; 11]                                | 1     |
| Swim      | 48                    | 10                         | 20                             | 5752              | [1; 915]                               | 36    |
| Trousers  | 64                    | 17                         | 6,06                           | 79                | [1; 57]                                | 1     |
| Random500 | 500                   | 246                        | 13,49                          | 600               | [0,001; 63,9]                          | 0,95  |

В качестве шага сетки применяется предложенный в работе [6] шаг  $R_b$ , обеспечивающий компромисс между точностью аппроксимации и вычислительными затратами. Этот шаг определяется по формуле

$$R_b = \max\{P_e, P_p/n_e\},$$

где  $P_e$  — минимальная ненулевая длина  $x$ -проекций всех неперпендикулярных рёбер всех объектов,  $P_p$  — длина  $x$ -проекции наименьшего по ширине объекта, а  $n_e$  — число рёбер этого объекта. Такой выбор шага гарантирует наличие как минимум одного элемента кодирования на каждое ребро при сохранении приемлемых вычислительных затрат.



**4.1. Анализ избытков аппроксимации.** Для оценки зависимости избытков аппроксимации от шага сетки были проведены тесты, результаты которых показаны на рис. 11. На графиках представлено отношение суммарной площади избытков аппроксимации к общей площади кодируемых объектов в зависимости от шага сетки для методов SE и ATE.

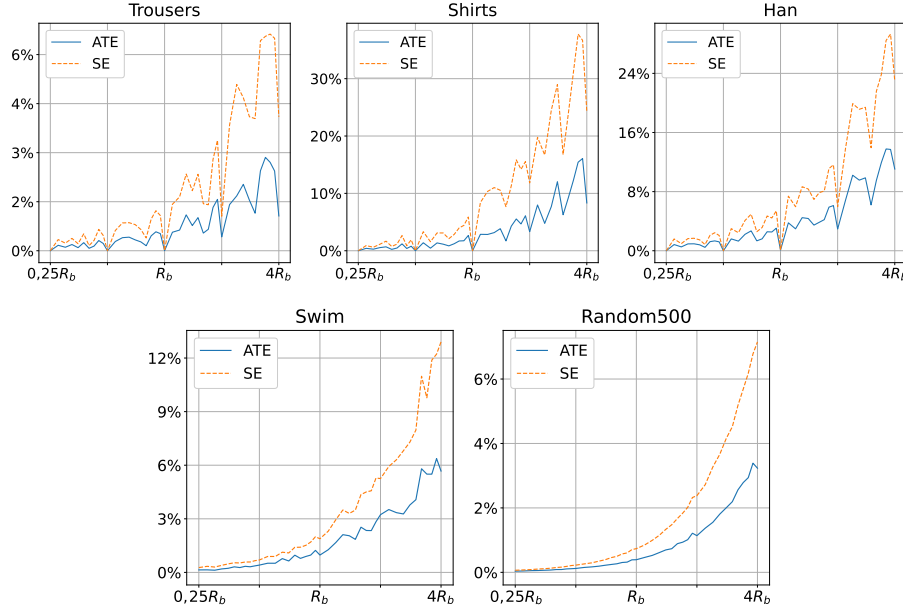


Рис. 11. Зависимость доли избытков аппроксимации от шага сетки

Результаты показывают, что при оптимальном выборе шага сетки удастся полностью устранить избытки аппроксимации (например, для наборов Han, Shirts и Trousers при  $R_b = 1$ ). Однако даже незначительное отклонение от этого значения приводит к заметному росту избытков: до 5% для метода SE и до 2% для ATE. В среднем метод кодирования ATE демонстрирует в 2,2 раза меньшую долю избытков по сравнению с SE, что подтверждает его значительно более высокую точность описания формы объектов.

**4.2. Анализ времени работы.** Для сравнения производительности для каждого метода выполнено по 1000 запусков с шагом сетки  $2^n R_b$ , где  $n = 1, 0, -1, -2, -3$ . В табл. 2 приведены среднее время работы (вместе со среднеквадратичным отклонением) и длина упаковки.

Из табл. 2 видно, что с применением ATE время работы алгоритма увеличивается по сравнению с SE для всех тестов. В среднем время выполнения при кодировании методом ATE примерно в 1,7 раза больше.

Таблица 2

**Сравнение длины и времени упаковки  
для разных способов кодирования**

| Тест      | Шаг сетки | Время, мс       |                 | Длина  |        |
|-----------|-----------|-----------------|-----------------|--------|--------|
|           |           | SE              | ATE             | SE     | ATE    |
| Han       | 2         | $0,1 \pm 0,02$  | $0,2 \pm 0,04$  | 55,0   | 53,0   |
|           | 1         | $0,1 \pm 0,03$  | $0,2 \pm 0,04$  | 50,0   | 50,0   |
|           | 0,5       | $0,2 \pm 0,05$  | $0,3 \pm 0,09$  | 51,0   | 51,0   |
|           | 0,25      | $0,3 \pm 0,07$  | $0,5 \pm 0,10$  | 51,0   | 51,0   |
|           | 0,125     | $0,5 \pm 0,11$  | $0,9 \pm 0,21$  | 51,2   | 51,2   |
| Shirts    | 2         | $0,3 \pm 0,10$  | $0,9 \pm 0,16$  | 74,0   | 74,0   |
|           | 1         | $0,4 \pm 0,14$  | $0,5 \pm 0,13$  | 71,0   | 71,0   |
|           | 0,5       | $0,6 \pm 0,26$  | $0,9 \pm 0,93$  | 70,0   | 70,0   |
|           | 0,25      | $1,0 \pm 0,20$  | $1,6 \pm 0,34$  | 69,5   | 69,5   |
|           | 0,125     | $1,8 \pm 0,35$  | $3,3 \pm 0,69$  | 68,6   | 68,6   |
| Swim      | 72        | $1,0 \pm 0,14$  | $1,8 \pm 0,32$  | 7687,5 | 7687,5 |
|           | 36        | $1,6 \pm 0,23$  | $2,6 \pm 0,46$  | 7615,5 | 7471,5 |
|           | 18        | $2,2 \pm 0,35$  | $3,5 \pm 0,63$  | 7471,5 | 7471,5 |
|           | 9         | $3,3 \pm 0,59$  | $5,9 \pm 1,74$  | 7444,5 | 7399,5 |
|           | 4,5       | $6,3 \pm 1,17$  | $10,4 \pm 2,01$ | 7449,0 | 7417,5 |
| Trousers  | 2         | $0,3 \pm 0,07$  | $0,5 \pm 0,11$  | 285,0  | 285,0  |
|           | 1         | $0,4 \pm 0,13$  | $0,6 \pm 0,19$  | 284,0  | 284,0  |
|           | 0,5       | $0,7 \pm 0,17$  | $1,3 \pm 0,37$  | 284,0  | 284,0  |
|           | 0,25      | $1,3 \pm 0,30$  | $2,6 \pm 0,64$  | 283,8  | 283,8  |
|           | 0,125     | $2,5 \pm 0,61$  | $5,3 \pm 1,24$  | 283,6  | 283,6  |
| Random500 | 1,9       | $99,1 \pm 8,66$ | $138 \pm 14,4$  | 885,3  | 872,0  |
|           | 0,95      | $131 \pm 11,5$  | $161 \pm 15,7$  | 861,6  | 860,6  |
|           | 0,475     | $170 \pm 15,5$  | $233 \pm 21,1$  | 855,4  | 845,4  |
|           | 0,2375    | $265 \pm 21,7$  | $409 \pm 31,0$  | 843,8  | 840,7  |
|           | 0,11875   | $484 \pm 34,1$  | $740 \pm 48,6$  | 841,3  | 840,2  |

Это связано с адаптацией нестатичных трапеций, которая требует дополнительных вычислений. Тем не менее для теста Shirts наблюдается противоположный эффект: при уменьшении шага сетки с  $2R_b$  до  $R_b$  время работы сокращается, поскольку все вершины объектов попадают в узлы сетки, и в кодировке остаются только статичные адаптивные трапеции.

**4.3. Качество упаковки.** Если принять во внимание длину упаковки, то метод кодирования АТЕ в среднем показывает более качественные результаты. В проведённых экспериментах все значения длины упаковки, полученные с использованием АТЕ, оказались не хуже, чем при SE. Это объясняется более точной аппроксимацией формы, позволяющей эффективнее использовать доступное пространство контейнера.

Однако следует отметить, что подобное преимущество не гарантировано для всех случаев. Возможны ситуации, когда улучшение аппроксимации формы приводит к локальному ухудшению общего результата из-за изменения взаимного расположения объектов. Аналогичные эффекты наблюдаются и при уменьшении шага сетки, что обсуждается в работе [6].

Результаты также показывают, что целесообразность применения того или иного метода определяется структурой данных. Для наборов, где удаётся подобрать шаг сетки так, чтобы вершины объектов совпадали с узлами, сегментный метод остаётся более эффективным, так как он обеспечивает аналогичное качество аппроксимации при меньшем времени работы (например, в тестах Han, Shirts и Trousers). Однако для более сложных случаев, где сетка не может быть согласована с вершинами (например, в тестах Swim и Random500), адаптивные трапеции показывают значительное преимущество, обеспечивая более точное кодирование формы и более плотное размещение объектов.

### **Заключение**

В работе представлен новый метод кодирования двумерных объектов с использованием адаптивных трапеций, предназначенный для решения задач резки и раскроя. В отличие от сегментного подхода, предложенный метод обеспечивает более точную аппроксимацию формы объектов и существенно снижает избытки аппроксимации, возникающие при несоответствии вершин узлам сетки. Проведённые вычислительные эксперименты показали, что применение данного метода в жадном алгоритме упаковки позволяет добиться более плотного размещения объектов при умеренном увеличении вычислительных затрат.

Разработанный метод характеризуется хорошей масштабируемостью, что делает его удобным для интеграции в различные эвристические и метаэвристические алгоритмы двумерной упаковки. В перспективе представляется интересным исследование его применения для аппроксимации объектов со сглаженными или криволинейными контурами, не являющимися многоугольниками, путём модификации процедуры адаптации трапеций под такие случаи.

### **Финансирование работы**

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2022-0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

### **Конфликт интересов**

Автор заявляет, что у него нет конфликта интересов.

## Литература

1. Stoyan Y., Terno J., Scheithauer G., Gil N., Romanova T. Phi-functions for primary 2D-objects // Stud. Inform. Univers. 2002. V. 2, No. 1. P. 1–32.
2. Oliveira J. F., Ferreira J. S. Algorithms for nesting problems // Applied simulated annealing. Heidelberg: Springer, 1993. P. 255–273. DOI: 10.1007/978-3-642-46787-5\_13.
3. Ma H., Liu C. C. Fast nesting of 2-D sheet parts with arbitrary shapes using a greedy method and semi-discrete representations // IEEE Trans. Autom. Sci. Eng. 2007. V. 4, No. 2. P. 273–282. DOI: 10.1109/TASE.2006.874973.
4. Akunuru R., Babu N. R. A semi-discrete geometric representation for nesting problems // Int. J. Prod. Res. 2013. V. 51, No. 14. P. 4155–4174. DOI: 10.1080/00207543.2012.751508.
5. Burke E., Hellier R., Kendall G., Whitwell G. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem // Oper. Res. 2006. V. 54, No. 3. P. 587–601. DOI: 10.1287/opre.1060.0293.
6. Chehrazad S., Roose D., Wauters T. A fast and scalable bottom-left-fill algorithm to solve nesting problems using a semi-discrete representation // Eur. J. Oper. Res. 2022. V. 300, No. 3. P. 809–826. DOI: 10.1016/j.ejor.2021.10.043.
7. Babu A. R., Babu N. R. A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms // Comput. Aided Des. 2001. V. 33, No. 12. P. 879–891. DOI: 10.1016/S0010-4485(00)00112-3.
8. Bennell J. A., Oliveira J. F. The geometry of nesting problems: A tutorial // Eur. J. Oper. Res. 2008. V. 184, No. 2. P. 397–415. DOI: 10.1016/j.ejor.2006.11.038.
9. Dowsland K. A., Dowsland W. B., Bennell J. A. Jostling for position: Local improvement for irregular cutting patterns // J. Oper. Res. Soc. 1998. V. 49, No. 6. P. 647–658. DOI: 10.1057/palgrave.jors.2600563.
10. Han G.-C., Na S.-J. Two-stage approach for nesting in two-dimensional cutting problems using neural network and simulated annealing // Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf. 1996. V. 210, No. 6. P. 509–519. DOI: 10.1243/PIME\_PROC\_1996\_210\_150\_02.
11. Oliveira J. F., Gomes A. M., Ferreira J. S. TOPOS — A new constructive algorithm for nesting problems // OR Spectrum. 2000. V. 22, No. 2. P. 263–284. DOI: 10.1007/s002910050105.

Акентьев Всеволод Владиславович

Статья поступила

19 ноября 2025 г.

После доработки —

12 декабря 2025 г.

Принята к публикации

22 декабря 2025 г.

# SEMI-DISCRETE GEOMETRIC REPRESENTATION OF OBJECTS IN THE TWO-DIMENSIONAL NESTING PROBLEM

V. V. Akentev

Novosibirsk State University,  
2 Pirogov Street, 630090 Novosibirsk, Russia  
E-mail: akentevvv01@gmail.com

**Abstract.** The paper addresses the two-dimensional nesting problem, which consists in arranging a set of irregularly shaped objects within a bounded strip so as to minimize its utilized length. One of the key aspects affecting the efficiency of nesting algorithms is the choice of geometric representation, as it determines both the accuracy of shape approximation and the computational cost of the packing process. We introduce a novel object representation method based on adaptive trapezoids. In contrast to the commonly used segment-based approach, the proposed technique significantly reduces approximation errors caused by the misalignment between polygon vertices and grid nodes, thereby improving the fidelity of the geometric model. Computational experiments conducted on standard ESICUP benchmark datasets demonstrate that integrating the proposed representation into a bottom-left-fill packing algorithm yields denser layouts with only a moderate increase in computational effort. The results confirm the effectiveness and practical applicability of the proposed method for two-dimensional nesting problems. Tab. 2, illustr. 11, bibliogr. 11.

**Keywords:** cutting problem, nesting problem, semi-discrete representation, bottom-left-fill algorithm.

## References

1. Y. Stoyan, J. Terno, G. Scheithauer, N. Gil, and T. Romanova, Phi-functions for primary 2D-objects, *Stud. Inform. Univers.* **2** (1), 1–32 (2002).
2. J. F. Oliveira and J. S. Ferreira, Algorithms for nesting problems, in *Appliad Simulated Annealing* (Springer, Heidelberg, 1993), pp. 255–273, DOI: 10.1007/978-3-642-46787-5\_13.

English transl.: *Journal of Applied and Industrial Mathematics* **20** (1) (2026).

3. **H. Ma** and **C. C. Liu**, Fast nesting of 2-D sheet parts with arbitrary shapes using a greedy method and semi-discrete representations, *IEEE Trans. Autom. Sci. Eng.* **4** (2), 273–282 (2007), DOI: 10.1109/TASE.2006.874973.
4. **R. Akunuru** and **N. R. Babu**, A semi-discrete geometric representation for nesting problems, *Int. J. Prod. Res.* **51** (14), 4155–4174 (2013), DOI: 10.1080/00207543.2012.751508.
5. **E. Burke**, **R. Hellier**, **G. Kendall**, and **G. Whitwell**, A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem, *Oper. Res.* **54** (3), 587–601 (2006), DOI: 10.1287/opre.1060.0293.
6. **S. Chehrazad**, **D. Roose**, and **T. Wauters**, A fast and scalable bottom-left-fill algorithm to solve nesting problems using a semi-discrete representation, *Eur. J. Oper. Res.* **300** (3), 809–826 (2022), DOI: 10.1016/j.ejor.2021.10.043.
7. **A. R. Babu** and **N. R. Babu**, A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms, *Comput. Aided Des.* **33** (12), 879–891 (2001), DOI: 10.1016/S0010-4485(00)00112-3.
8. **J. A. Bennell** and **J. F. Oliveira**, The geometry of nesting problems: A tutorial, *Eur. J. Oper. Res.* **184** (2), 397–415 (2008), DOI: 10.1016/j.ejor.2006.11.038.
9. **K. A. Dowsland**, **W. B. Dowsland**, and **J. A. Bennell**, Jostling for position: Local improvement for irregular cutting patterns, *J. Oper. Res. Soc.* **49** (6), 647–658 (1998), DOI: 10.1057/palgrave.jors.2600563.
10. **G.-C. Han** and **S.-J. Na**, Two-stage approach for nesting in two-dimensional cutting problems using neural network and simulated annealing, *Proc. Inst. Mech. Eng., Part B: J. Eng. Manuf.* **210** (6), 509–519 (1996), DOI: 10.1243/PIME\_PROC\_1996\_210\_150\_02.
11. **J. F. Oliveira**, **A. M. Gomes**, and **J. S. Ferreira**, TOPOS — A new constructive algorithm for nesting problems, *OR Spectrum* **22** (2), 263–284 (2000), DOI: 10.1007/s002910050105.

Vsevolod V. Akentev

Received November 19, 2025

Revised December 12, 2025

Accepted December 22, 2025