

УДК 004.021

## Проблема укладки прямоугольников с частично фиксированной позицией на ограниченной плоскости

*Левшин Н.А.*

Московский физико-технический институт (государственный университет)  
Фирма "1С".

Аннотация.

Рассмотрен частный случай two-dimensional strip packing problem. Предложено решение поставленной задачи с помощью жадного алгоритма. Также приведен пример более общего варианта задачи и рассмотрен вариант решения

При работе с любыми объектами, привязанными ко времени важно их наглядное представление. Чаще всего в роли таких объектов выступают пользовательские задачи. Одна из наиболее информативных визуализаций подобных элементов является их укладка на плоскости, где одна из осей представляет шкалу времени. При укладке каждый из объектов представляет собой прямоугольник со следующими свойствами:

- размер
- положение вдоль оси шкалы времени

Задача укладки является частным случаем проблемы двумерной упаковки в полугораниченную полосу (Two Dimension Strip Packing, 2DSP). Она формулируется следующим образом:

*Пусть дано  $N$  прямоугольников  $a_1 \dots a_N$  с фиксированными размерами  $w_i, h_i$ . Назовем полугораниченной полосой область на плоскости, заключенную между функциями  $x > 0, y > 0, x < W$ , где  $W$  – ширина полосы. Требуется расположить прямоугольники внутри полосы так, чтобы они не пересекались, и высота заполненной области была как можно меньше.*

Известно, что задача 2DSP является NP-полной, однако существует множество алгоритмов позволяющих получить за полиномиальное время не оптимальный, но близкий к нему результат. Одним из таких алгоритмов является жадный уровневый алгоритм упаковки First Fit Decreasing Height. Идея заключается в том, что прямоугольники сортируются по убыванию высоты, первый (самый «высокий») элемент располагается в левом нижнем углу полосы, инициализируя прямоугольную область (уровень), отсеченную от полосы продолжением верхнего ребра элемента. Остальные прямоугольники располагаются внутри этой области слева направо, пока есть место. Прямоугольник, не поместившийся на текущем уровне, помещается сверху, образуя следующую область. Последующие элементы размещаются по такому же принципу, но место для расположения ищется не только в текущем уровне, а и во всех ранее образованных.

Существенным отличием поставленной задачи от 2DSP является то, что прямоугольники нельзя располагать где угодно внутри полосы, так как координата  $x$  в них уже фиксирована. Однако можно применить основные принципы из описанного метода, чтобы выработать свой, применимый для текущей задачи. Для этого в первую очередь элементы будут сортироваться не по высоте, а по ширине. Первый элемент располагается внизу полосы и добавляется во множество уже расположенных элементов  $Y$ . Далее, последующие элементы располагают внизу полосы. Если при таком расположении элемент  $x_i$  пересекается границами с каким-либо элементом  $y_i$  из множества  $Y$ , то он поднимается вверх над элементом  $y_i$  и проверка корректности укладки происходит снова. Как только элемент проходит проверку укладки, он добавляется во множество уже расположенных элементов  $Y$ .

Алгоритм показал хорошую скорость работы, а главное, для двух наборов входных данных, для которых большая часть элементов общая, ответ выдаёт визуально схожие укладки. На практике такой алгоритм хорошо подходит для визуализации расписаний событий на месяц, когда

элементы расположены горизонтально вдоль оси времени. Для отображения наименования события требуется ровно одна строка, которая и задаёт размер элемента по высоте символа. Ширина элемента задаётся временными рамками.

К сожалению не всегда такой подход применим. Например, для расписания на день привычнее расположить шкалу времени вертикально. Тогда у элементов фиксированная высота, но ширина может изменяться, и, более того, желательно, чтобы она была максимальной, и больше текста поместилось без переносов. Задача укладки принимает следующий вид:

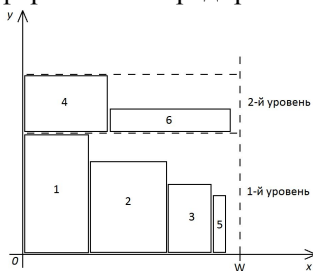
*Пусть дано  $N$  прямоугольников  $a_1 \dots a_N$  с фиксированным размером  $w_i$  и положением  $x_i$ . Назовем полуограниченной полосой область на плоскости, заключенную между функциями  $x > 0, y > 0, x < W$ , где  $W$  – ширина полосы. Требуется расположить прямоугольники внутри полосы так, чтобы они не пересекались, высота заполненной области была как можно меньше, а суммарная площадь прямоугольников была как можно больше.*

При такой модификации изначально поставленной задачи, она уже не является частным случаем 2DSP. Для вычисления оптимальной укладки предлагается использовать тот же жадный алгоритм, но над его результатом произвести некоторую постобработку. Предположим, что все элементы укладки имеют одинаковую высоту, например 1. Тогда можно расположить элементы на полосе, пользуясь ранее предложенным алгоритмом. Постобработка будет происходить в 2 шага:

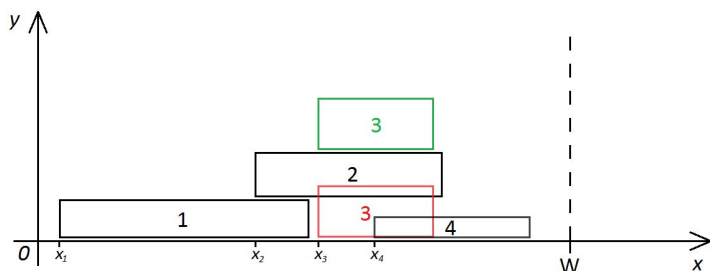
1. Для каждого элемента, который не «зажат» между ближайшими соседями вдоль оси  $Y$ , двигаем его так, чтобы расстояния до соседей были равны. Если пройденное расстояние меньше некоторого порогового значения, изменения для элемента не применяются. Как только ни один элемент нельзя сдвинуть, метод переходит к шагу 2.
2. Каждый из элементов растягиваем вдоль оси  $Y$  вверх до ближайшего соседнего прямоугольника, а затем и вниз до ближайшего соседа.

При такой укладке, доступное пространство используется гораздо эффективнее для отображения информации об объектах, При этом устойчивость алгоритма сохраняется, а скорость работы можно регулировать, изменяя пороговое значение.

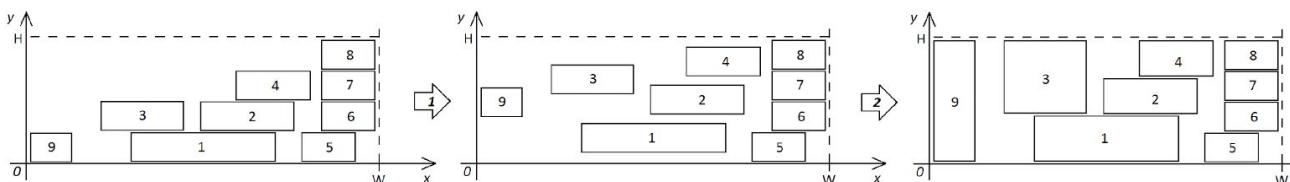
Реализованные алгоритмы написаны на языке C++ и применяются для горизонтальной и вертикальной укладки элементов в поле Планировщика, включенного в мобильную версию платформы «1С: Предприятие 8.3.11».



**Рис 1. Пример работы метода First Fit Decreasing Height**



**Рис 2. Пример работы метода расположения элементов с фиксированной координатой**



**Рис 3. Пример работы метода расположения элементов с фикс. координатой и свободной высотой**

## Литература

1. Andrea Lodi, Silvano Martello, Michele Monaci. Two-dimensional packing problems: A survey. //European Journal of Operational Research 2002. V. 141. P. 241.