

Кафедра системного программирования СПбГУ

# Nesting в библиотеке CoreCVS

Курсовая работа

Свирин Евгений Александрович, 343 группа  
научный руководитель: Пименов А. А.

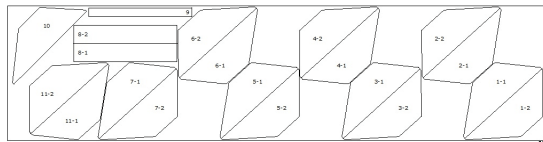
25 мая 2020 г.

Задача раскроя — вырезание деталей из некоторого объекта, минимизируя растрату материала.  
1939 — формальная постановка задачи Л.В. Канторовичем.

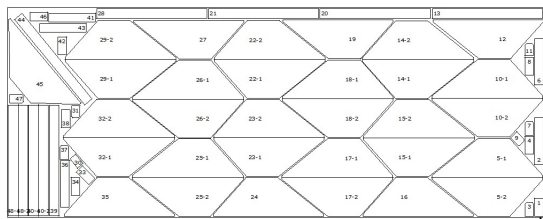


Задача раскроя в промышленности

Мой научный руководитель получил заказ на эффективную в плане утилизации материала раскладку некоторых деталей по листу металла для последующей вырезки, воспользовавшись промышленной программой, он обнаружил неудовлетворительный результат – было очевидно, что раскладка неоптимальна, оставляла очень много места между деталями, которое пошло бы на утилизацию. Это называется задачей раскроя – вырезание деталей из некоторого объекта, минимизируя растрату материала. Впервые математически сформулирована была Леонидом Витальевичем Канторовичем в 1939. Нетрудно догадаться, что технологии по эффективному решению такой задачи широко применяются в промышленности, в частности, например: сталепрокатной, стекольной, пленочной промышленности а также при лазерной резке.



Пример работы ProNest 2019



Пример раскладки вручную

Верхний рисунок - результат работы промышленной программы пронест, этот пример даёт понять, что даже промышленные программы предоставляют далеко не оптимальный ответ в некоторых случаях.

**Целью** проделанной работы являлось спроектировать и реализовать программу, решающую задачу раскроя для выпуклых многоугольников.

Задачи:

- Изучение подходов к решению задачи раскроя
- Выбор нескольких алгоритмов по решению этой задачи, исходя из изученного материала
- Реализация этих алгоритмов в CoreCVS
- Сравнение их между собой и аналогами для оценки эффективности и доработки
- Реализация GUI и поддержки файловых форматов

В итоге из соображений, что даже платные раскройщики иногда весьма неэффективны, а хороших открытых раскройщиков очень мало, появилась идея, ставшая целью: спроектировать и реализовать программу раскройщик. И для того чтобы с чего-то начать, было решено реализовать программу решающую частный случай задачи: разместить внутри прямоугольной корзины многоугольники, минимализуя значение высоты самой высокой вершины, то есть минимализуя вертикальную заполненность. Программу было решено встроить в coreCVS, библиотеку, полезную для компьютерного зрения, реализующую геометрические примитивы, так нужные этой работе.

Промышленные компании:

- Eastman Machine Company
- SIGMATEK
- HyperTherm

SVGNest — самая популярная реализация с открытым доступом.

Как уже было отмечено, есть множество компаний чьим продуктом являются программы раскройщики, соответственно список некоторых. Код у них закрыт, в процессе работы ориентировался на опубликованные примеры их работы. Opensource раскройщики очень редки, самый популярный на GitHub - SVGNest, умеет раскладывать не только выпуклые многоугольники, имеет схожую с реализованным раскройщиком логику работы.

## Эвристики:

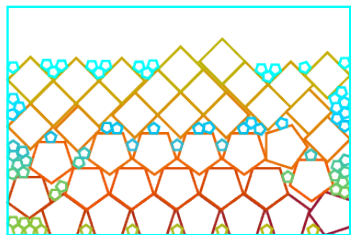
- Выборочные эвристики
  - First Fit Decreasing (FFD)
  - First Fit Increasing (FFI)
  - First Fit (FF)
- Эвристики размещения
  - Bottom-Left Placement
  - Constructive Approach (CA)
- LowerMassCenter

В López-Camacho E. et al. An effective heuristic for the two-dimensional irregular bin packing problem //Annals of Operations Research. – 2013. – Т. 206. – №. 1. – С. 241-264 предоставлен обзор этих эвристик.

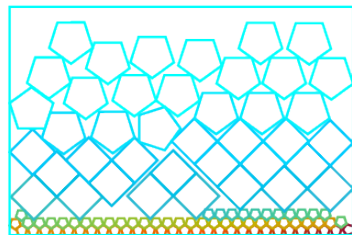
Задачи раскрыя решаются с помощью эвристик, их комбинаций, на основе которых подбираются приближенные решения, поскольку получение точных решений чрезвычайно времязатратно. Среди эвристик можно выделить две категории: выборочную и размещения. Выборочные Эвристики определяют которую из фигур помещать на очередной итерации алгоритма. Эвристики размещения же указывают на то как распологать очередной объект. Примеры выборочных: FF – никак не менять исходный порядок помещаемых фигур, FFD, FFI отсортировать фигуры в начале алогоритма по площади по убыванию и возрастанию соответственно.

Среди эвристик размещения есть, например, Bottom-Left placement, диктующий из всех возможных позиций выбирать нижнюю левую, а есть CA-подход, который указывает, что нужно оптимизировать покрывающий прямоугольник.

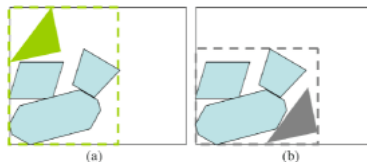
Кроме того бывают эвристики не попадающие под категории, например lower the masses, которая предлагает понизить всем фигурам центр масс а потом помещать их.



First Fit Decreasing



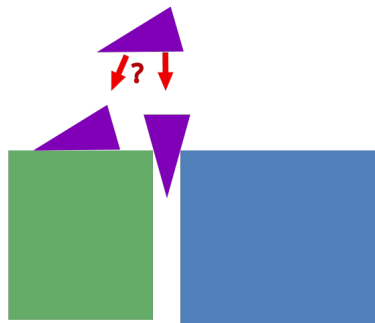
First Fit Increasing



## Идея Constructive Approach

Верхние две картинки демонстрируют работу эвристик First Fit Decreasing и First Fit Increasing. более тёплым цветом рисуются фигуры, добавленные раньше. Как видите FFD, в данном примере позволяет фигурам поменьше попасть в пространства между большими помещенными фигурами, образуя более оптимальную раскладку чем FFI. Снизу CA подход. Стоит отметить что судя по изученным примерам промышленные компании используют CA локально, но при этом оптимизируют ту же метрику что в нашей задаче, как это реализовать конкретно не ясно, поэтому это единственная из перечисленных эвристик, которая не была имплементирована.

Новая эвристика повышает приоритет позициям, соответствующим фигурам с пониженным центром масс.



Спорная ситуация помещения вращений

Также была придумана новая эвристика, которая повышает приоритет позициям соответствующим фигурам с пониженным центром масс. Новая эвристика выделяет вращения с самым низким центром масс и при подсчете какую позицию фигуры считать самой низкой "занижает" им высоту пропорционально вертикальной проекции фигуры. Альтернативная версия эвристики сравнивает центры масс, "занижая" центр масс того же класса вращений. Вот на картинке присутствуют два разных вращения одного треугольника, дабы не менять "приземистости" картины, новая эвристика скорее всего скажет выбрать левую позицию для размещения



Из комбинаций различных эвристик было составлено 10 алгоритмов.

Самые эффективные:

- Alg1 — алгоритм, использующий новую эвристику
- Alg2 — использует измененную новую эвристику

Набор данных:

- Небольшой набор данных, уже имевшийся
- Аппроксимация промышленных примеров
- Дополнительные составленные примеры

Из комбинаций различных эвристик было составлено 10 алгоритмов. Alg1, Alg2 – самые эффективные алгоритмы. Они используют эвристики FFD и LM. Для размещения же они используют новую эвристику, в первом алгоритме её предикат выбора использует вертикальную проекцию фигуры, а во втором дополнительно задействует центры масс.

Набор данных для тестирования представляет собой небольшой набор данных, уже умевшийся, аппроксимацию промышленных примеров и дополнительные примеры составленные мной.

Значения — вертикальная заполненность.

SVGNest — алгоритм самого популярного открытого раскройщика.

Alg1 в сравнении с SVGNest:

минимальный выигрыш - 0.015

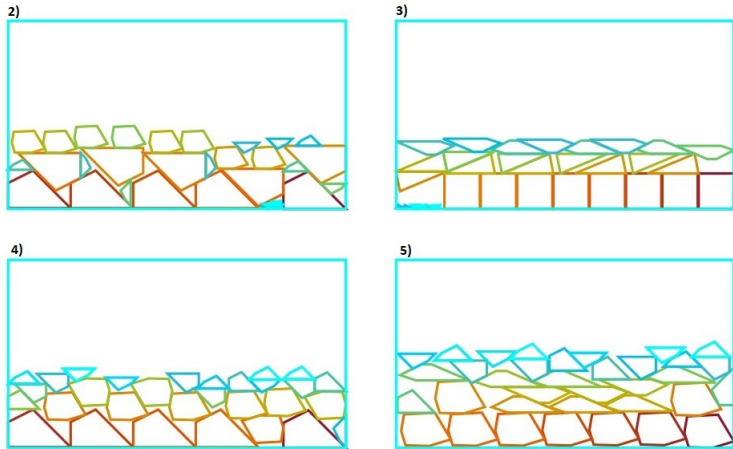
средний выигрыш - 0,065

максимальный выигрыш - 0.169

Test	1	2	3	4	5
Alg1	0.680	<b>0.431</b>	<b>0.373</b>	<b>0.431</b>	<b>0.564</b>
Alg2	<b>0.649</b>	0.446	0.406	0.440	0.585
SVGNest	0.695	0.493	0.542	0.461	0.613

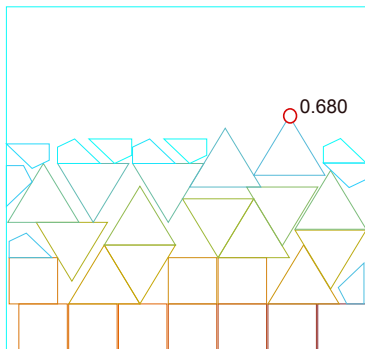
Таблица Выборка некоторых алгоритмов с различными тестами

Было проведено сравнение с SVGNest, на всем наборе данных Alg1 и Alg2 выигрывают, вот таблица-выборка некоторых алгоритмов с различными тестами. Метрика — вертикальная заполненность, то есть минимализация стороны сонаправленной со стороной корзины минимального прямоугольника, покрывающего фигуры. Строка SVGNest соответствует работе алгоритма SVGNest, сравнение проводилось при одинаковом максимальном допустимом количестве поворотов — 4, то есть кроме фигуры самой фигуры для помещения рассматривается ещё 3 её вращения. Такое число было выбрано так как другого интерфейс SVGNest не поддерживает.

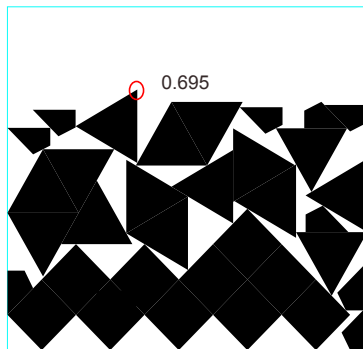


Результат работы Alg1 над тестами 2, 3, 4, 5

Вот визуализация некоторых текстов.



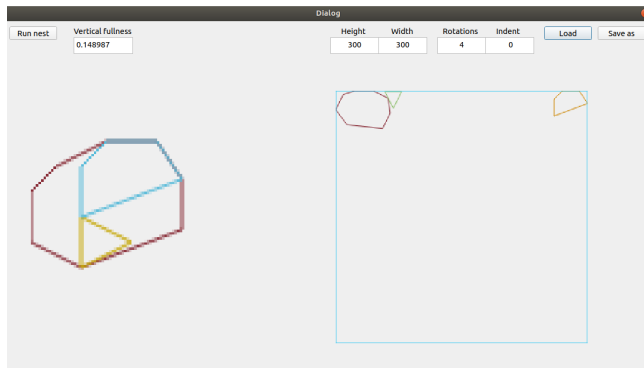
Результат работы Alg1  
над тестом 1



Результат работы  
SVGNest над тестом 1

А это результаты работы над 6 тестом, как видно  
раскройщик corecvs справился лучше.

- Поддержка файловых форматов
  - SVG (Scalable Vector Graphic)
  - DXF (Drawing Exchange Format)
- GUI



Визуализация работы раскройщика

Как уже было отмечено для работы и отладки тестировщика была добавлена поддержка файловых форматов SVG и DXF на основе инструментов предлагаемых coreCVS. И, используя, QTWidgets реализован GUI.

- Изучен ряд статей по данной теме, придумана и реализована новая эвристика, основывающаяся на повышении приоритета позициям, соответствующим фигурам с пониженным центром масс
- Составлено несколько алгоритмов по решению поставленной задачи
- В библиотеке CoreCVS реализовано несколько алгоритмов nesting
- Произведено сравнение реализованных алгоритмов и аналогов
- Добавлена поддержка форматов SVG и DXF, реализован GUI

В результате проделанной работы был изучен ряд статей по выбранной теме, придумана и реализована новая эвристика, основывающаяся на повышении приоритета позициям, соответствующим фигурам с пониженным центром масс.

На основе изученного и придуманного составлено несколько алгоритмов по раскройке. Эти алгоритмы были реализованы в CoreCVS Произведено сравнение реализованных алгоритмов и аналогов, а также добавлена поддержка форматов svg и dxf с помощью реализованных в CoreCVS парсеров, реализован GUI посредством QT widgets.