**Assigning blocks to contours based on the containment of metal**

**Dmitriy Kuptsov, Sovin Stanislav**

**Introduction**

Assigning blocks to contours is a routine task at every mine. The **sort plan (не знаю термина)** is created by the geologists to (do what)? In this document we are going to describe an algorithm that is used to achieve the task.

The structure of the document is the following. First, we will discuss the algorithm and give our thoughts on the complexity. And in the second part, we are going to present the experimental results. Here, we are going to perform microbenchmarking of the algorithm and its parts, validate the correctness of the classification algorithm by visually inspecting the block model and created contours, we are also going to use Micromine software to import the output of our algorithm. And in the final part of this technical report, we are going to give a few thoughts on further improvements to the algorithm using machine learning algorithms, such as deep neural networks.

**Algorithm**

* **Selection of blocks which fall inside the main contour**
* **Assigning blocks to contours based on the containment of metal**
* **Suppression of outliers – contours with small number of blocks**
* **Exporting strings file**

Selection of blocks that fall inside the main contour

The algorithm is simple:

for each block

if ray (that starts at the center of the block) intersects odd number of line segments

mark the block and store it in a separate array

**Algorithm complexity analysis**

We have analyzed the algorithm by breaking it down into smaller parts

Algorithm that selects the blocks that fall into the main contour

O(nm) where n is the number of blocks in the model and m is the number of line segments

Algorithm that assigns blocks to subcontours

The time complexity of this algorithm is O(kv) where k is the number of contours and v is the average number of blocks in each contour

Memory complexity is O(v^2)

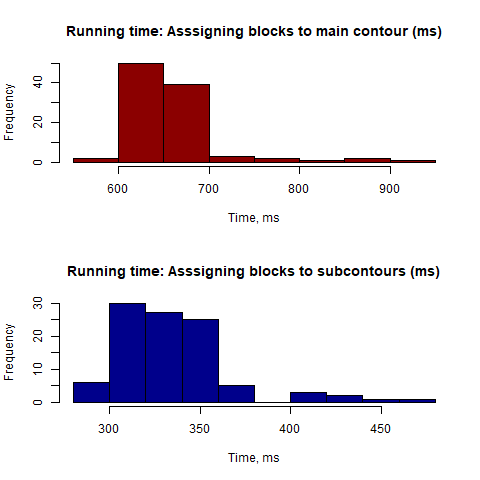
Suppressing the outliers

The time complexity of this algorithm is O(kp^2) where k is the number of contours and p is the average number of blocks that form the border of a contour

**Experimental result**

We have conducted several experiments:

* Microbenchmarking of algorithms
* Validation of the correctness of the obtained results on small data set
* Running the algorithm with large input size (large main contour and smaller block sizes)

****

For block model of size 34095 and the number of blocks in main contour 474, the mean running time was about 657 milliseconds to assign blocks to main contour and 336 milliseconds to create subcontours correspondingly. The standard deviation was 54 and 32 ms accordingly.

**Validation of the results in Micromine software**

**Future ideas**

* Using machine learning to smooth the contours
* Using convex hull algorithm