

Report IT3708 – Project 3

1. The MOEA finds segmentation by splitting the image into random segments. Then it merges neighboring segments, biased towards segments with similar centroid pixels, until a preset number of segments has been reached. Overall I had huge trouble with finding any good segmentation strategy which in terms led to very poor results for the final segmentation.

2. The algorithm implements NSGA-II. The core implementation is similar to that described in the paper from itslearning[1]. Specific to this domain each chromosome representation has two main attributes. (1) an array the size of the image and (2) a list of segments and their coordinates on the image. The array contains, for each pixel, an integer value which references the segment this pixel belongs to. Correspondingly, each segment is represented as a list with XY coordinates. This representation yields an overhead of $2N$ for each chromosome, where N is the total number of pixels in the image.

For crossover, two parents are selected using binary tournament selection. Here, parents with a higher Pareto-front rank (assigned by *fast-non-dominant-sort* as described in [1]) are preferred. If the rank is equal, chromosomes with a higher crowding distance value will be selected.

The child is created from a merger between the two parent chromosome segments. While there exist some pixels not assigned to a segment, the algorithm picks any of these pixels, and a random parent. This pixel is the seed point for a region growing, using the segment assigned to this pixel in the parent and unassigned pixels in the child chromosome as boundaries for the region. In other words, while the neighboring pixel belongs to the same segment in the parent AND the neighboring pixel is an unassigned pixel in the child, this pixel is accepted in the new segment region.

After generating an offspring population this is added to the previous generation and Pareto-fronts are formed by using the *fast-non-dominant-sort* algorithm. This approach secures elitism for the implementation. The use of *crowding-distance-assignment* (also from [1]) ensures the diversity among the solutions. The next generation is created by inserting each front into the population. If the front is bigger than the space remaining in the population (constrained by parameter), the individuals with the highest crowding distance are picked.

There are three mutation operators. **(1) Border-Pixel-Operator:** Picks number (decreasing with iterations) of boundary pixels (i.e. pixels which lie on the edge of two segments) and moves them to the neighboring segment. **(2) Merge-Segment-Operator:** Merges two neighboring segments (either biased towards similar segments or random). **(3) Split-Segment-Operator:** Splits a segment into two segments.

	0	1	2	3	
0	0	0	1	1	Seg 0: ([0,0],[0,1],[1,1],[1,0])
1	0	0	1	3	Seg 1: ([0,2],[0,3],[1,2])
2	3	3	3	3	Seg 2: ([3,3],[3,2])
3	3	3	2	2	Seg 3: ([1,3],[2,0],[2,1],[2,2],[2,3],[3,0],[3,1])

Parameter values for the implementation were:

POP_SIZE = 50
 NEXT_GEN = 25
 NEAREST_NEIGHBORS (L) = 4
 MUTATION = 0.4
 ITERATIONS = 100
 SEGMENT_NUMBER = 5

Figure 1: Chromosome Representation

Here are the results from the algorithm. The crossover is clearly not good enough, and, subjectively, it even seems that the initial segmentation is better than the one produced by the algorithm (but the initial segmentations were discarded by the algorithm by the objective function). Also, the initial segmentation is not good enough to produce a solid foundation for the genetic algorithm.

3 Objectives:

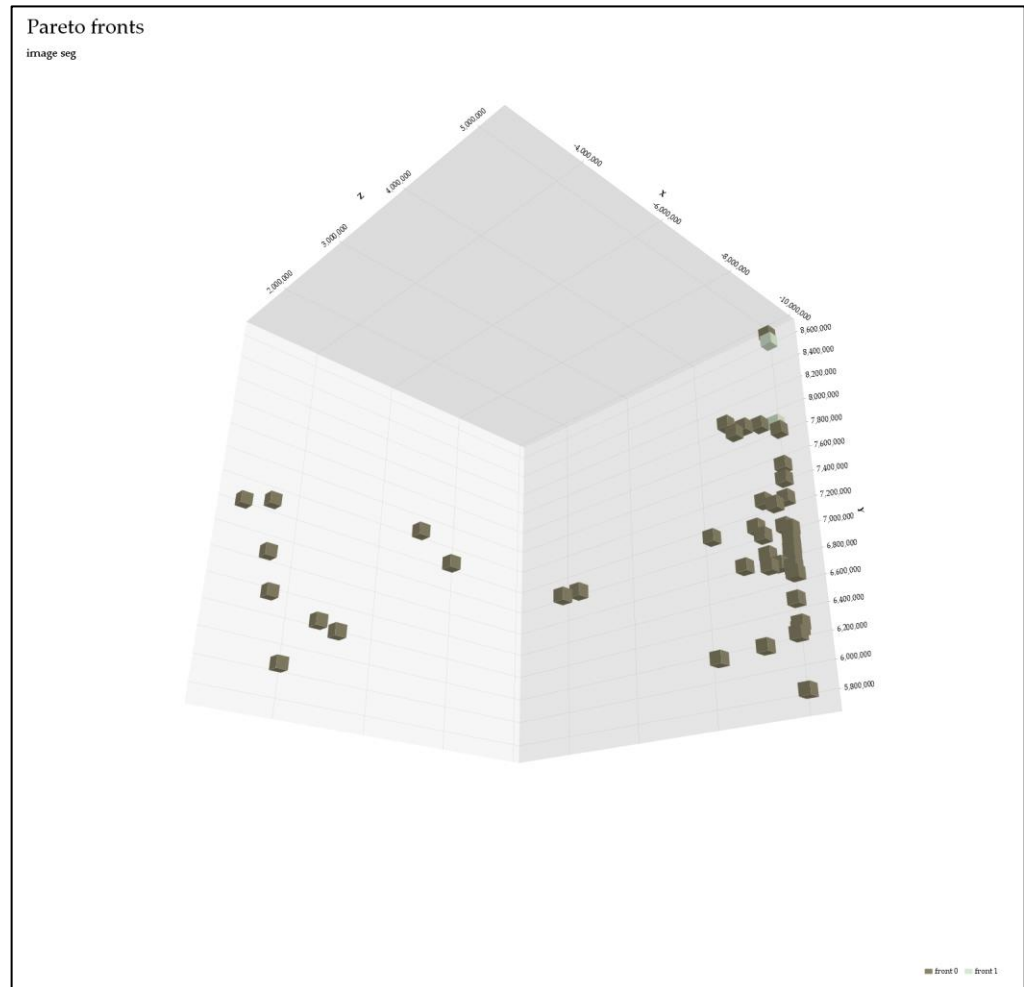


Figure 2: Pareto Fronts for 3 Objectives

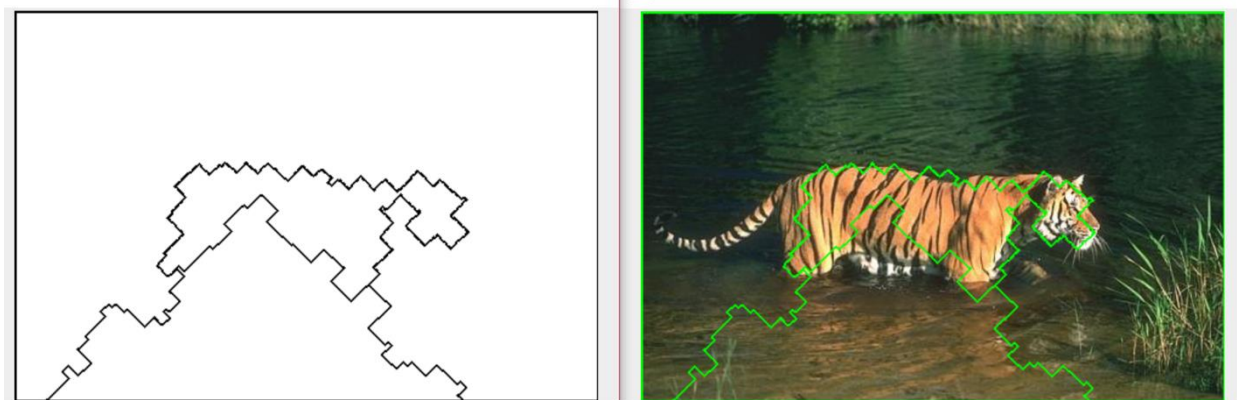


Figure 3: Segmentation for 3 objectives

2 Objectives:

Connectivity/Deviation:

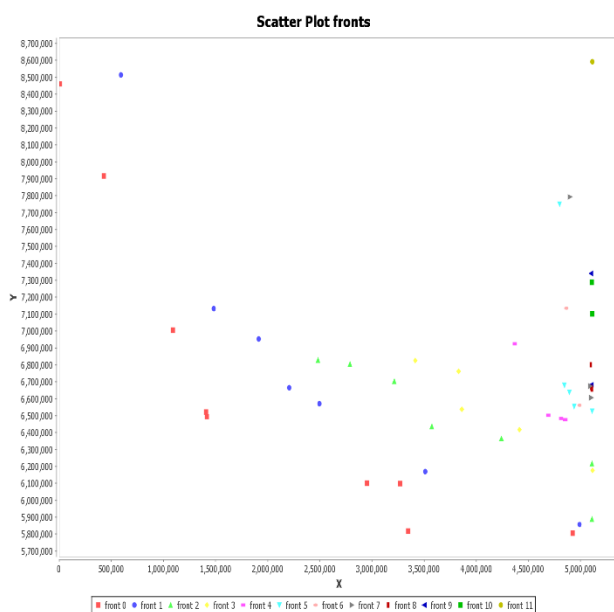


Figure 5: Pareto fronts Connectivity/Deviation Objectives

Edge/Connectivity:

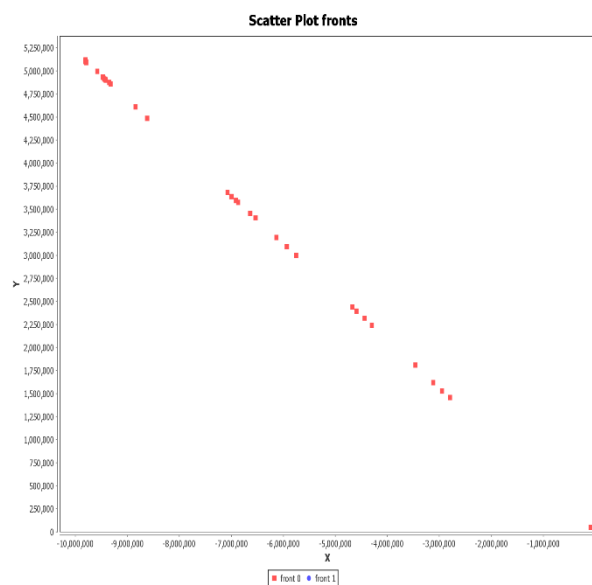


Figure 6: Pareto fronts Edge/Connectivity Objectives

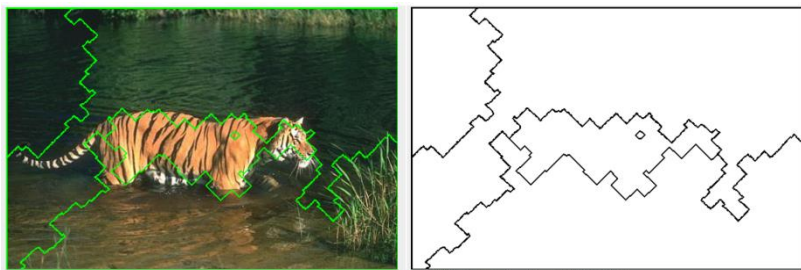


Figure 7: Segmentation for Connectivity/Deviation Objectives

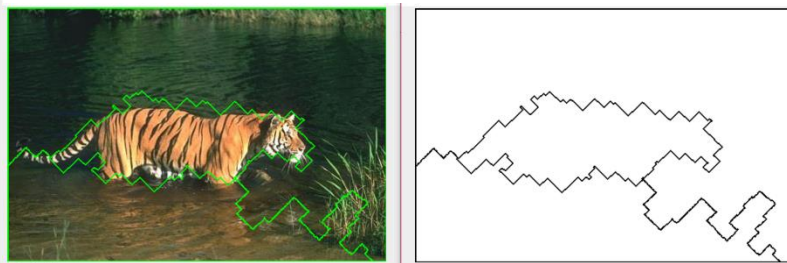


Figure 8: Segmentation for Edge/Connectivity Objectives

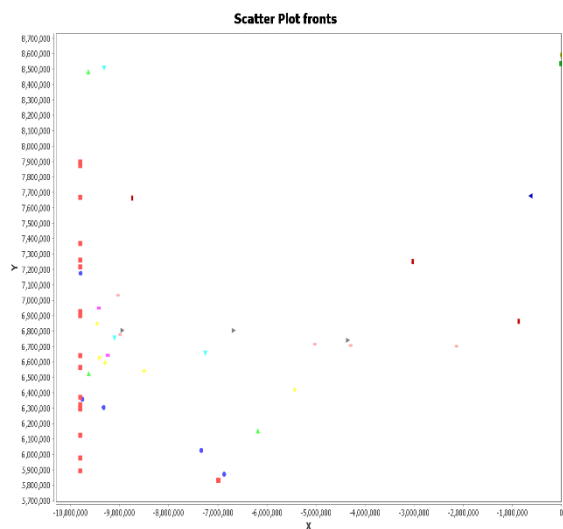


Figure 9: Pareto Fronts Edge/Deviation

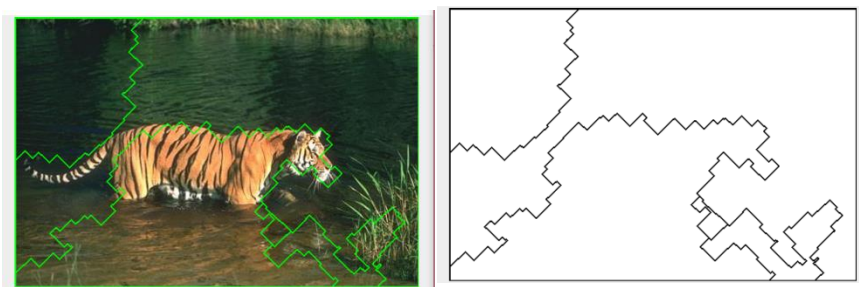


Figure 10: Segmentation Edge/Deviation Objectives

References:

[1] Kalyanmoy Deb, Associate Member, IEEE, Amrit Pratap, Sameer Agarwal, and T. Meyarivan: *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 2, APRIL 2002.