IT3708 - Project 3

Magnus Moan

March 2017

1 Task 1: Finding segmentations

I've chosen to implement an Strength Pareto Evolutionary Algorithm 2 (SPEA2) to find image segmentations. The SPEA2 algorithm maintains a pareto front in a constant size archive. The solutions to be added to the archive is found by selecting the best individuals from both the current archive and the current population (details below). The archive does at all time contain the current Pareto front (Non dominated solutions). If the number of non dominated solutions exceeds the size of the archive I remove those solutions that are most similar to the other solutions in the front. This way I make sure that the archive is as diverse as possible. A non dominated solution is a solution with a fitness less than 1. The SPEA2 fitness-measure takes into account how many solutions that dominate the given solution, and also how similar the solution is to the other solutions. Mathematically we define fitness for individual i as:

$$f(i) = f_r(i) + d(i) \tag{1}$$

$f_r(i)$ is the raw fitness of individual i. It measures how many solutions that dominate the current solution.

$$f_r(i) = \sum_{j \in dom(i)} |\{k | j \in dom(k)\}| \tag{2}$$

$dom(i)$ is the set of solutions that *dominates* i. Dominates between 2 solutions is defined as usual, solution s_1 dominates solution s_2 is s_1 is better than s_2 in all objectives. Finally we have the density of a solution i:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{3}$$

σ_i^k is the distance from solution i to the k^{th} nearest neighbor, where k is:

$$k = \sqrt{N + A} \tag{4}$$

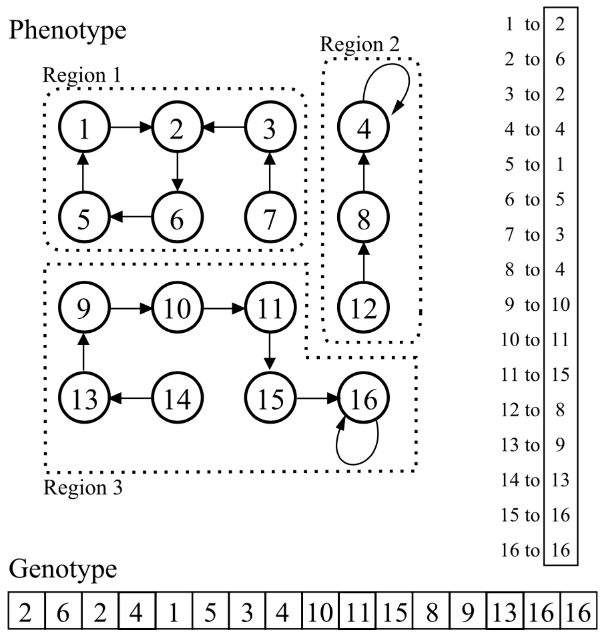
Here N is population size and A is the archive size.

If the number of solutions with fitness less than 1 is less than the archive size (A) we add the best dominated solutions to the archive.
The next parents for the next generation is chosen by performing binary tournament selection between randomly chosen solutions in the archive. After selecting parents from the archive, the next population is generated by performing crossovers and mutations on the parents.

2 Task 2: Implementation

My initial population is created by clustering pixels using the k-means algorithm. The k-means algorithm segments the pixels by adding pixels to the segment which they have the best most similarity with. Similarity is measured by difference in RGB-values. After the k-means algorithm is done, I separate the k-means segments based on pixels connectivity. Two pixels are connected if they are in the same segment, and they are neighbors in the image. Two pixels are neighbors if they are beside eachother or above/below eachother. The segments I get after this process is then transformed into chromosomes (my genotypes). My chromosomes are one dimensional arrays with length equal to the number of pixels. Each entry in the chromosome array is a pointer. See figure below for illustration (source: `http://ieeexplore.ieee.org/document/4983250/`).

The main loop of my algorithm is as follows:



1. Calculate objectives for all chromosomes
2. Calculate fitness of all chromosomes (as described above)
3. Fill the archive with non-dominated solutions.
4. If the archive isn't filled, fill the rest of the spots with the best dominated chromosomes. If the archive have too many individuals, remove the least diverse solutions (as described above).
5. Select parents for the next generation from the archive (using binary tournament selection).
6. Create the next population by performing crossover and mutations on the parents. Every new individual is the result of a crossover, and some individuals does also get mutated (with a 20% chance).

A few notes about the algorithm. I perform a DFS search on my chromosome to find which pixels that are placed in the same segment. This DFS search is how I transform my genotype (chromosome) into my phenotype. When I have the phenotype I can easily compute edge value, overall deviation and connectivity. The binary tournament selection picks 2 random individuals from the archive and compare their fitness. In 80% of the situations the one with the highest fitness is put into the parent-pool, in the other 20% both individuals are put into the parent-pool. Note that this selection technique will add some individuals from the archive more than once into the parent-pool.

Crossover is done by choosing two random parents. Then I cut each parent in two parts, at the same random point in the genotype. A new individual is formed by merging the first half of one parent with the second half of the other parent.

Mutations are performed by merging a randomly picked segment with its most similar neighbor (in terms of RGB-centroid).

Task 3, 4: Test images and Pareto fronts

I've chosen the first test image as my test image.

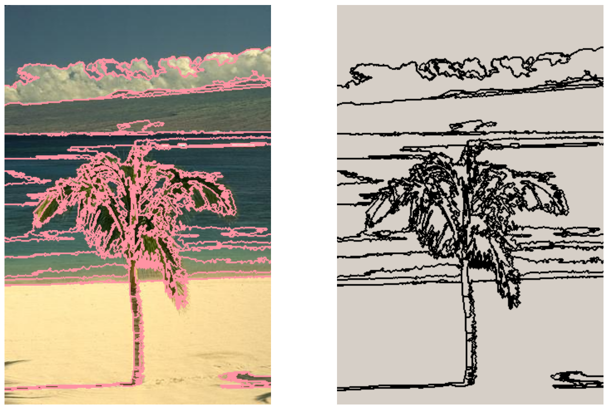


Figure 1: Connectivity and edge values: 50 segments

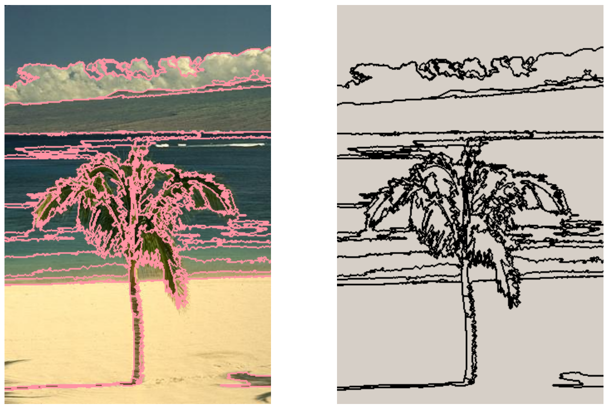


Figure 2: Overall deviation and connectivity: 25 segments

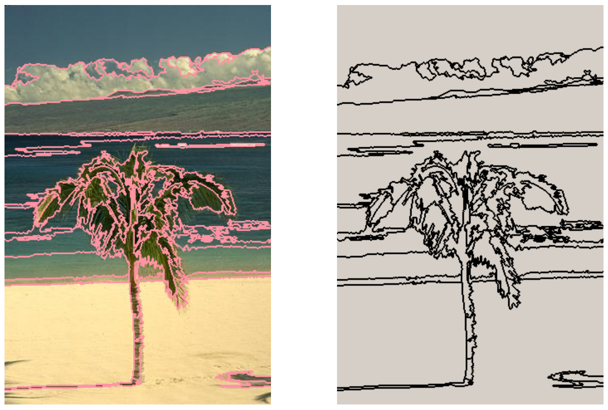


Figure 3: Overall deviation and edge value: 70 segments

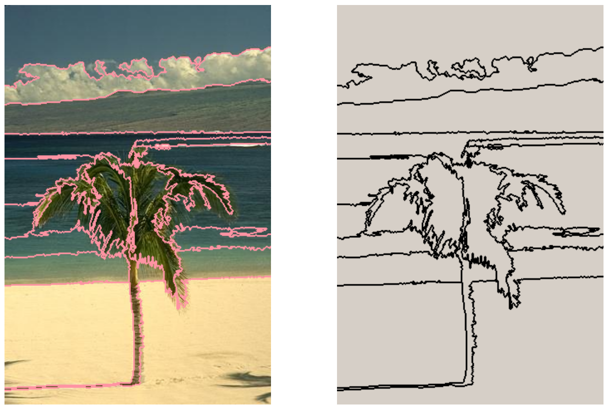
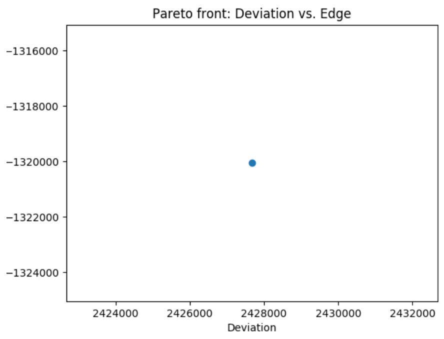
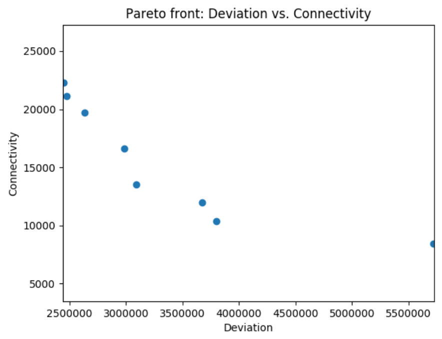


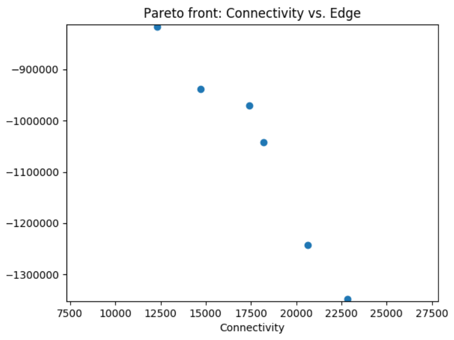
Figure 4: All objectives: 20 segments



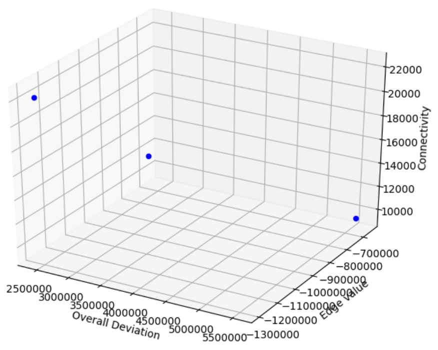
(a) Deviation vs. edge value



(b) Connectivity vs. overall deviation



(a) Connectivity vs. edge value



(b) All objectives

Task 5: Parameters

- Number of generations: 50
- Population size: 30
- Archive size: 5
- Mutation rate: 20%
- Binary tournament rate: 80%