IT3708 - Bio Ai - MOEA

NSGA-II

Sigve Skaugvoll, MIT 2018

Q1: NSGA-II comparison to SPEA

I've chosen to implement the NSGA-II MOEA, and its strategy because it have some advantages when it comes to fitness which I prefered over for example SPEA's advantage when it comes to fitness. NSGA-II uses global nondomination checks amongst the population and has elitism. SPEAs biggest advantage was that fact that fitness was easy to calculate. NSGA-II fitness is not that hard to implement and calculate thus I went for the performance advantage.

One disadvantage of mine implementation against the SPEA is that mine has some parameters that has to be defined, one very specific, is a teta value when creating segments for the initial population. This teta is used to cut the MST edges and thus creating a new segment if the RGB-distance is not within a range for the teta value. Deciding this value is very difficult and image specific. Thus not ideal, but when correctly specified gives an very good advantage. SPEA is parameter-less which makes it much more attractive to use, but does not allow for any specifications towards the problem at hand. NSGA-II also has some good effects of parameters because solutions compete with their crowding distances. So no extra niching parameter is required.

Because of limitations towards the demonstration of this project I decided to go with NSGA-II over SPEA because NSGA-II uses as stated earlier, elitism, which gives a fast convergence. Instead of a slow sorting of the entire population. So if there's a big population we want speed because the task at hand is very computationally heavy. Therefor NSGA-II is a better strategy! Another reason to why I think it's a good strategy to use is because of its performance to fairly easy implementation ratio.

Q2: Chromosome representation

I choose to have a genotype which represents a MST of the image to segment. Then I transform this phenotype into a chromosome genotype, by splitting the genotype when an edge between to nodes in the tree have a value greater than a user specified value. This type of representation is computionaly and memory heavy, but gives quite a good initial segmentation and representation of the problem. With this representation it's also easy to divide all the objects of an image such as pixels, segments, pixel-edges and all the information for a phenotype to

classes. All pixels are Pixel objects and they hold a list of all their neighbors as a directed edge. The chromosome then has a list of its segments, which is of class Segment. A segment is a collection of all the pixels in the “restricted” or cut off mst, where the direct and indirect pixels are within a color range. The fitness for the chromosome is a minimization of the two objectives.

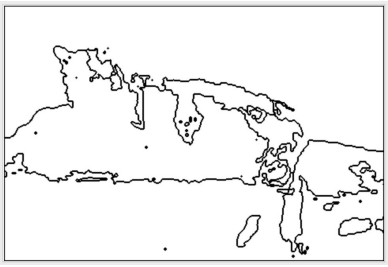
Another representation can be to use a 1D array where the length is number of pixels. This gives that the index represents a node and the value for that index / node is the index to another node, which the first index (not the value) is connected to in the image. This gives a clustering that represents a segmentation. This type of representation would give a lot easier implementation of crossover and mutation than the representation I chose. Because one can simply just select the 1D index and swap with another index in another chromosome. And then one can also just “flip” the value-index in a mutation as long as it does not break any constraints.

I would perhaps argue for that the second representation perhaps would be better because of its simplicity when it comes to crossover and mutation. But the creation of initial population is much harder than the representation I chose. And The initial population is very important in this task, thus I wanted to optimise for as good as possible initial population, and than rather struggle and perhaps miss some good crossover and mutations possibilities because of restrictions from the chosen representation. In the end, I would still prefer the simplicity of the chosen representation and argue for it to be the best.

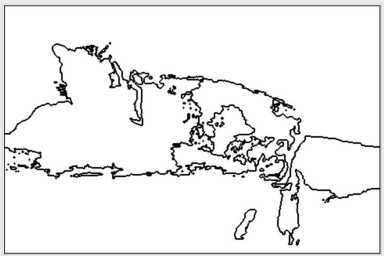
Q3: Effects of weights on weighted sum method

200teta - segmentation color

<0.9,0.1>

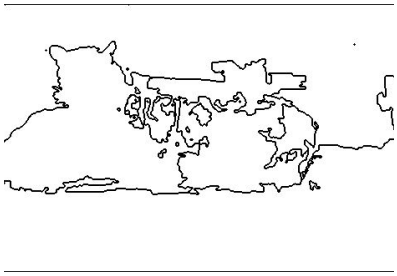


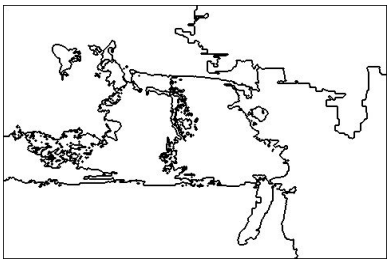
<0.5,0.5>



<0.1,0.9>

100teta - segmentation color





<0.8,0.2> <0.2,0.8>

The effects of different weighting <OD, Edg> is changing the objective focus, do we want to focus on segment deviation or segment edges as we see in picture 4, and 5, we have change the focus on the tiger and background respectively. We can clearly see that the figure and different firm-pattern of the animal becomes more weighted than the foreground when we weight the overall deviation which is the similarity between pixels in segments more than edges and background edges are more clearly when we do the opposite.

I've experienced that finding good combination of weights was really hard and that the weights are not known in advance. Kinda like my segmentation-teta. Thus the weights are very user to user dependent and it's not easy to find the relative importance of the objective, thus they cannot be qualitative.