Jacob Hopkins  
9/15/2020  
Minimum k – Clustering

I have chosen minimum k clustering. This is a problem in which we have points represented within space, and we want to group/subset/partition/cluster the data into k number of clusters. This algorithm is generally solved with k means clustering as well as other forms like min max k means clustering as well as other algorithms like SOMs. This is a process of assigning k random cluster centers and assigning points to the nearest cluster center. Then by finding the mean /average/center for all of the points in each cluster we reassign the cluster center and recheck each point reassigning those who become closer to a new/different positioned cluster center.  
  
 Genetic algorithms work on a large range on problems. They generally involve populations of solutions and new populations are created through genetic operations. These operations include selection, crossover, and mutation. We need DNA or chromosomes to represent the states in order to get fitness and breed to the next generation. Those chromosomes can be mutated by random changes, overrepresented when successful (elitism), and removed from the mating pool when poorly performing. With clustering we can even do fusion and fission. Fusion is taking two unique allele values in a chromosome and combining them into one or combining them into one cluster. Fission is the act of taking two single allele values and breaking them into a cluster.  
  
 In my research of GA’s and using them with this problem I have found so far two approaches for representing the states of clusters in regards to the points. Hruschka, Campelo, and Castro approached this problem by representing a parallel array to the points array in which each value in the parallel array is a “cluster ID” to give reference to which cluster said point belongs. Secondly Maulik and Bandyopadhyay used a k length array with a tuple containing the cluster centers, which comes with the trade off that each iteration points need to be assigned to the cluster centers before fitness evaluation. I think I will be using the clusterID’s and keeping point counts lower for now.

1-point Mutation can be changing the point or cluster ID randomly with a low chance rate. Cross over can be a one-point midpoint cross over, where we split the strings and create two children from them. Or a two-point cross over, or even a k point cross over. Or even a uniform cross over. I am more interested in the uniform cross over.

Fitness is a complex thing. People say that the CGA should not only minimize the distance between points and cluster centers but should have a highest fitness when the right/best/correct number of clusters is selected. This means it should be changing the cluster centers, and the cluster count as well. Average Silhouette Width developed by Kaufman and Rousseeuw is used in to determine the number of clusters needed, and some also use it for the objective function of a GA.  
  
 Personally I do not fully understand this yet, and will look more into it and other algorithms for finding the fitness of my population. So far I have a graphical viewer for seeing the points, and clusters. I have the k means algorithm going, and will be adding the GA shortly.