Genetic Algorithm:

1. Gene Model:
   In this problem, every group have only two ways to color it, so I model this in to 0 and 1, and this is the definition of our gene locus, and all locus from colorable graph form a chromosome, so every chromosome is a color method for entire color bounding box, and is one solution to this problem.
2. Mutation Method, MOGAR:
   (1) First sort the rows by score, I keep the highest score chromosome at top.
   (2) For each row, I random a number x and to see if x<a_i, I will choose this row to mutate, given $a_i = \frac{i-1}{N}, i \in N$, we can see that the row with the best score is located at the top and thus $a_i = 0$, this row is never choose, thus never changed, and the lower the row is, the higher this mutation probability is.
   (3) After I choose the row to mutate, I perform mutation on those loci where their b_j is high, and $b_j = 1 - |p_{j0} - 0.5| - |p_{j1} - 0.5|, \forall j \in columns$, where $p_{jX} = \frac{\sum_{i=1}^{N}(N+1-i)\times\delta_{ij}(X)}{\sum_{m=1}^{N} m}$, where $\delta_{ij}(X) = 1$ if Matrix(i,j)=1, vice versa, this indicates some kind of statistics, because higher score in general will encounter similar structure that cause good result, so if one column is randomly distributed, then it has higher probability to changed.
   (4) I will replace the rows whose score lie in the last 20% with randomly generated chromosomes.
3. Mutation Method, MOGAC:
   (1) Sort rows by score, the same as MOGAR
   (2) For each column, I random a number y to see if y<b_j, as given above, and if it is true, I choose this column to mutation the last $b_j \times N$ rows' j^{th} loci.
4. Implementation:
   At first, I set a maximum iteration time, and a parameter γ, and I create an m by n chromosome matrix MC, where m=100*groups and n=groups, and in each iteration I random a number z, and if z< γ, I mutate the chromosome using MOGAR , otherwise I use MOGAC, and after that I update the score, and then do the next iteration, until maximum iteration is reached or is about exceed the contest run time threshold.