The general design idea behind this operator is to apply EA method to randomly generate tours, in which we try to find the best one that can beat the tour read from the function linkerntour.java. In order to achieve this goal, four main steps need to be developed.

- To design a crossover function implementing the idea from the PMX.
- To design a mutation function implementing the idea from inversion.
- To design a selection function implementing the idea from tournament.
- To enable the elitism method in order to keep the best one from previous generation.

The crossover rate is set to 0.9 and the mutation rate is set to 0.3. The reason of using such high mutation rate and crossover rate is because we want more changes for the tour. From those changes, we are able to find out a shorter one than the given one.

The test case is shown as below:

```
TSP tsp = new TSP(instance);
int[] Best = tsp.runGA(20, 10000, 0.9, 0.3, 12000000);
System.out.println("Before: "+tsp.Distance_1);
System.out.println("After: "+tsp.Distance_2);
```

The relevant result is described as below:

```
Before: 2610.0
After: 2589.0
```

As it can be seen, the tour read from the function linkertour.java is 2610.0 in total. After applying the EA operator, the shortest has become 2589.0. For more details, please check the code implemented for this exercise.