The general idea behind this operator is to generate several solutions and then apply a crossover function to find out a solution that is better the solution without using any EA this crossover operator. The general steps for solving this exercise are described as below.

- To design a crossover function by using the idea from ordered crossover method.
- To design a selection function by using the idea from tournament method.
- To enable elitism method in order to find out the best one.

The tour is remained unchanged, while the parking plan is changed by using the crossover operator.

By combining different parking plans and an unchanged tour, the operator is able to generate different solutions. From these solutions, we choose the best one with the large objective value.

A testing case is described as below:

```
TTPSolution bestSolusion = cottp.RunGA(GENERATIONS_SOL, CROSS_RATE_SOL, MAX_RUN_TIME_SOL);

// print to file
bestSolusion.writeResult(resultTitle);

// print to screen
System.out.println("\nBefore: ");
solution.println();
System.out.println("After: ");
bestSolusion.println();
```

The relevant result is shown as below:

```
Before:
220535.0 987172.0 1207772.0 2613 4156.272548277357 301039.5808678119 2819
After:
220367.0 987340.0 1207840.0 2613 4156.578798333247 301040.769355619 4178
```

As it can be seen, the objective value is changed a little bit, which turns out the algorithm Implemented for this exercise works well.