heuristic crossover and mutation operation have been proposed to prevent premature convergence. One of the permanent challenges for GAs is how to deal with premature convergence due to a sudden and fast reduction of search space and also getting stuck in local optima. heuristic crossover, actually uses four pointers, one pair for each parent, which will move clockwise and counterclockwise. These pointers will be evaluated by means of a fitness function, equal to the inverse distance between two cities, and try not to get stuck in local optima. This approach improves the convergent speed towards the global optimal solution. The heuristic mutation will prevent premature convergence, too. The role of adaptive and nonlinear probabilities of crossover and mutation is to solve the low stability and also slow convergent. It must be mentioned that the implementation of algorithm has no complexity. The method includes the results of 30 times implementation of Slandered Algorithms and proposed method. Initial population for all of these methods is equal to 100 Hipólito Hernández-Pérez et al [15] proposed a branch-and-cut algorithm for one commodity pickup and delivery traveling salesman problem and closely related traveling salesman problem. They used a 0-1 integer linear model for finding a optimal solution. In branch-and-cut algorithm follows a branch-and- bound scheme, in which new lower bounds are computed by solving linear program (LP) relaxation of the problem. The relaxation is tightened by adding valid inequities to the current LP, according to the cutting plane approach. They have implemented the algorithm with basic ingredients, where the initial heuristic is based only on the TSP nearest insertion and where the cutting plane does not consider the TSP 2-matching inequalities

In this thesis, we have applied Steady-State Genetic algorithm to solve the TSPPD. In our GA, we implemented pheromone based crossover operator which is used applied to construct the offspring. The pheromone based crossover contains both local and global information. Local information used here includes edge lengths and adjacency relations, whereas global information is stored as pheromone trails. The utilization of global information in crossover can increase the chance to find a better solution, and then improve the performance of GA.

2.1 The Proposed GA for TSPPD

In our proposed GA, path representation is used. After initializing the parameters used in algorithm, the GA starts from a random population of individuals (feasible tours), and iteration until some pre-defined stopping criteria is satisfied. Each individual is evaluated in each iteration (generation) based on its fitness function. During iteration, individuals are probabilistically selected from the population according to the binary tournament selection. Then the selected individuals are recombined or mutated to generate offspring. To accelerate the convergence of GA, a 2-opt local search procedure is used after recombination (crossover).The basic design of our algorithm includes all these steps:

2.1.1 Initial Population Generation

In our algorithm, we have used an adapted nearest neighbor heuristic to generate initial population. We have used iteratively method to generate initial population. During iteration, a city is selected uniformly at random in the individual. Now we will choose the nearest city, according the distance, from the cities that have not appeared in the individual. The procedure is repeated until all cities have appeared in the individual and feasible individual has been generated. The others individuals of population are filled in the same way.

The adapted heuristic is described as follows:

Step 1: Take an empty individual.
Step 2: Select uniformly random city t in the individual and check the feasibility of the city
Step 3: Insert the city into the individual if city is feasible else go to step 2.
Step4: Select the nearest one city according the distance that has not been appeared in the individual.
Step 5: Check the feasibility of the nearest city.
Step 6: If nearest city is feasible then insert it into the individual and go step 4.
Step 7: Procedure is repeated until the entire individuals have been filled.