** If you don't want to generate new random set of cities use "evaluate" button instead of "generate" button. The plotted graph will always correspond to currently selected parameters, i.e. crossover and selection strategy, population size etc.

My comments over implementing travelling salesman problem using genetic algorithm -

1. When no of cities increases due to the high time complexity of brute force and dynamic programming travelling salesman problem takes a lot of time to compute and produce answer. When I tried to approach the problem using Genetic Algorithm, even though I used a slow language like JavaScript until the number of cities reached over 100 the result was produced within a second, even though I was running the same algorithm for many different population sized at the same time. However the answers were fairly optimal, and for the same input changed in case of reevaluation.
2. In repeated trials over same input range, two point crossovers usually produced better results than one point crossover. However due to the random nature of the algorithm it was not always true.
3. In repeated trials over same input range, Elitism in many cases reached higher best-fitness in earlier generation(i.e. within 10 generations) over roulette wheel. However this was not always true. And in latter generations increase of fitness was very little in both selection strategies, thus producing almost identical fitness in later stages.
4. When the population size increases, I noticed that there is a growth in best-fitness computation. However as the population size increases this value comes to a still. I.e. before a saturation value (in my case it was mostly near 400) as population increased the results got slightly better.
5. In case of latter generations, the selected chromosomes had less changes with generation progression.