Experiments with Travelling Salesman Problem

```
clear;
cities = importdata('cities.csv');
nGenes = length(cities.data);
```

Experiment 1

Mutation: Use random swapping of cities when mutating Recombination: Use one point crossover

```
p(1).maxGenerations = 300;
p(1).populationSize = 20;
p(1).crossoverRate = 0.9;
p(1).mutationRate = 1/nGenes;
p(1).nSpecies = p(1).populationSize/4;
p(1).useSpeciation = true;
p(1).useRandomMutation = true;
p(1).useOnePointCrossover = true;
p(1).cities = cities;
name{1} = 'speciation random one point';
```

Experiment 2

Mutation: Swap neighboured cities only when mutating Recombination: Use one point crossover
Speciation: kMeans

```
p(2) = p(1);
p(2).useRandomMutation = false;
p(2).useOnePointCrossover = true;
name{2} = 'speciation neighbour one point';
```

Experiment 3

Mutation: Use random swapping of cities when mutating Recombination: Use two point crossover
Speciation: kMeans

```
p(3) = p(1);
p(3).useRandomMutation = true;
p(3).useOnePointCrossover = false;
name{3} = 'speciation random two point';
```

Experiment 4

Mutation: Swap neighboured cities only when mutating Recombination: Use two point crossover
Speciation: kMeans

```
p(4) = p(1);
p(4).useRandomMutation = false;
p(4).useOnePointCrossover = false;
name{4} = 'speciation neighbour two point';
```

Experiment 5

Mutation: Swap cities randomly when mutating Recombination: Use one point crossover Speciation: none

```
p(5) = p(1);
p(5).useRandomMutation = true;
p(5).useOnePointCrossover = true;
p(5).useSpeciation = false;
name{5} = 'random one point';
```

Experiment 6

Mutation: Use random swapping of cities when mutating Recombination: Use two point crossover Speciation: none

```
p(6) = p(1);
p(6).useRandomMutation = true;
p(6).useOnePointCrossover = false;
p(6).useSpeciation = false;
name{6} = 'random two point';
```

Experiment 7

Mutation: Swap neighboured cities only when mutating Recombination: Use one point crossover Speciation: none

```
p(7) = p(1);
p(7).useRandomMutation = false;
p(7).useOnePointCrossover = true;
p(7).useSpeciation = false;
name{7} = 'neighbour one point';
```

Experiment 8

Mutation: Swap neighboured cities only when mutating Recombination: Use two point crossover Speciation: none

```
p(8) = p(1);
p(8).useRandomMutation = false;
p(8).useOnePointCrossover = false;
p(8).useSpeciation = false;
name{8} = 'neighbour two point';
```

Run Experiments

```
tic;
for i=1:8
    tic;
    clear bestFitness medianFitness;
    parfor run = 1:100
        r(run) = doTsp(p(i));
```

```
        bestFitness(run,:) = r(run).bestFitness;
        medianFitness(run,:) = r(run).medianFitness;
    end
    p(i).medianBestFitness = median(bestFitness,1);
    p(i).medianMedianFitness = median(medianFitness,1);
    toc;
end
```

```
Elapsed time is 56.388082 seconds.
Elapsed time is 51.218856 seconds.
Elapsed time is 53.820428 seconds.
Elapsed time is 50.636977 seconds.
Elapsed time is 33.105312 seconds.
Elapsed time is 32.543430 seconds.
Elapsed time is 29.484379 seconds.
Elapsed time is 29.220968 seconds.
```

```
toc;
```

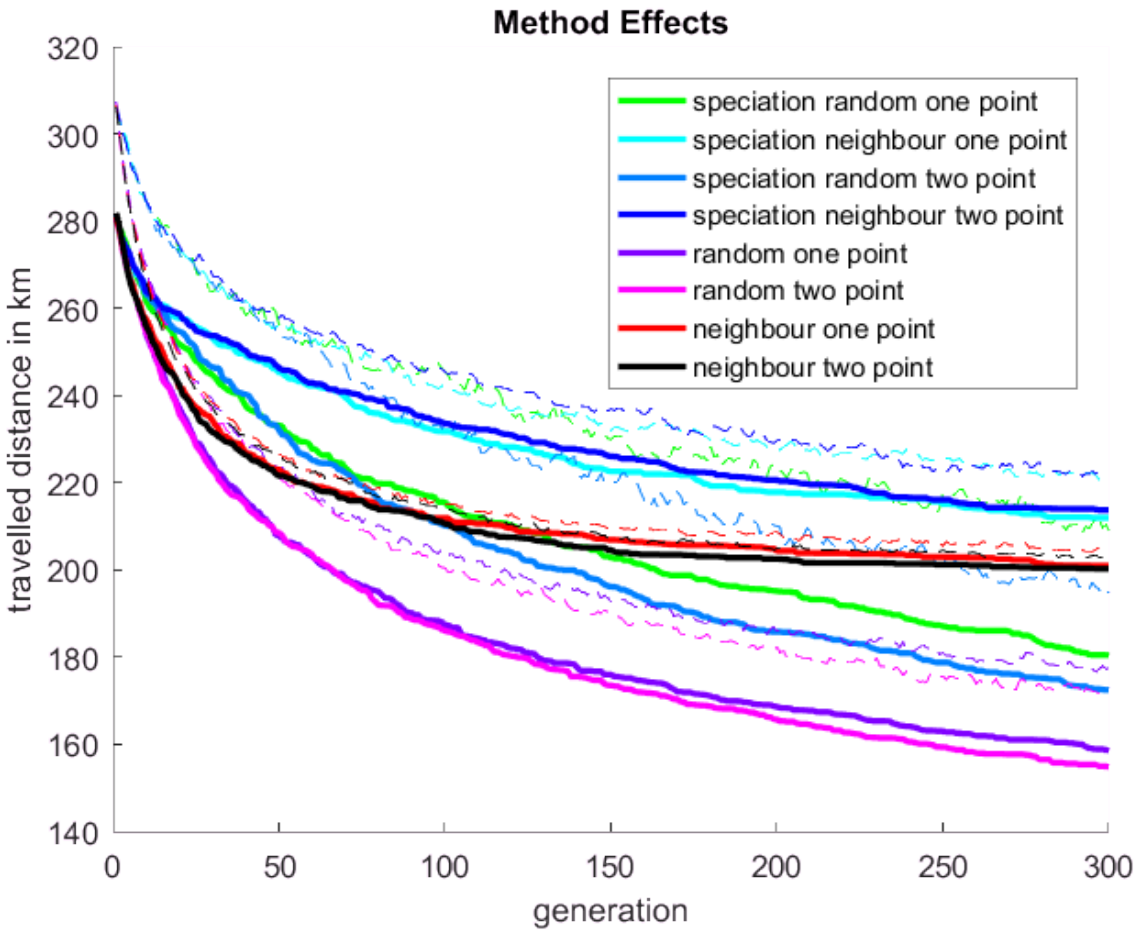
```
Elapsed time is 29.229438 seconds.
```

Plot Routes

for i=1:8 figure(i);clf;hold on; plotRoute(cities, r(i)); title(name{i}); end

Plot Fitnesses

```
figure(1);clf;hold on;
% Helper Library for a better color distribution over plots
colorSet = varycolor(8);
for i=1:8
    lineHandles(i) = plot(-1*p(i).medianBestFitness,'-','LineWidth',2,'Color',colorSet(i,:));
    plot(-1*p(i).medianMedianFitness,'--','Color',get(lineHandles(i),'Color'));
end
ylabel('travelled distance in km');
xlabel('generation');
legend(lineHandles,name,'Location','NorthEast');
title('Method Effects');
```


Method Effects
travelled distance in km
generation
320
300
280
260
240
220
200
180
160
140
0
50
100
150
200
250
300
speciation random one point
speciation neighbour one point
speciation random two point
speciation neighbour two point
random one point
random two point
neighbour one point
neighbour two point