Travailing Salesmen problem(TSP) solving by genetic algorithm(GA).
Tobias Kin Hou Lei
April 21, 2011

Usage

limited for ten cities only since this competition required us to solve a ten cities TSP problem. In order to run the simplePlot.py you need to install *numpy* and *matplotlib.pyplot.*

Bash command lines

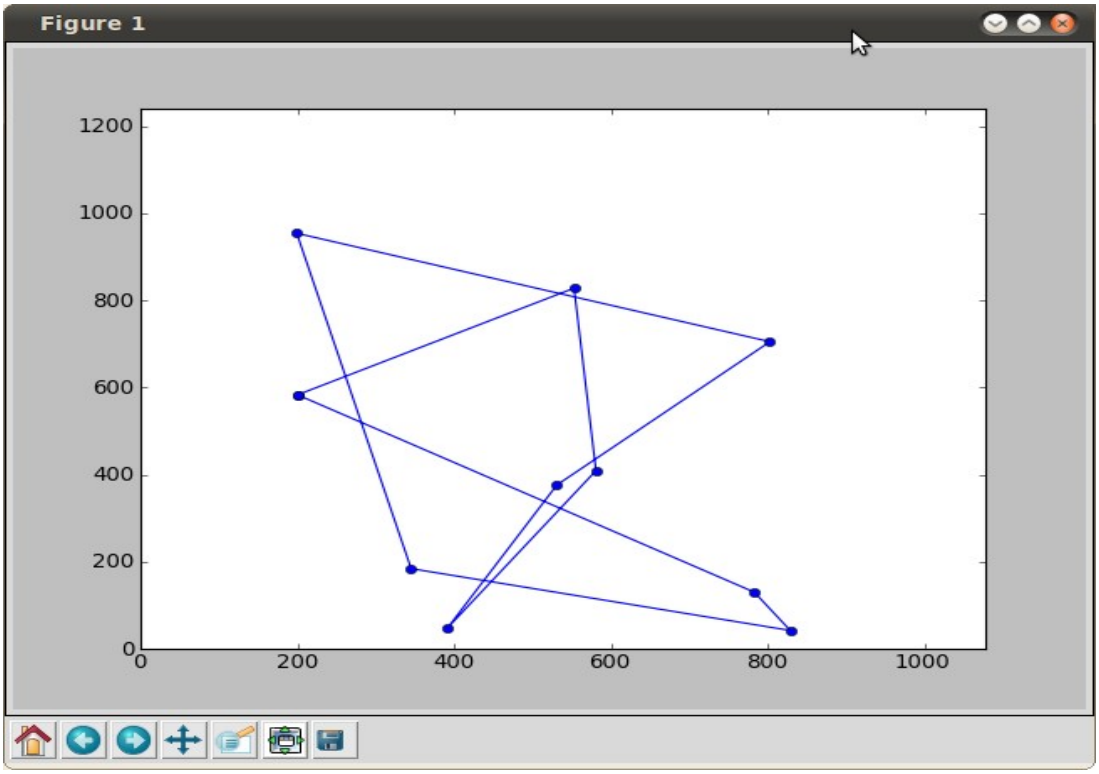
// Generate a problem randomly
./CitiesGenerator [NUMBER_OF_CITIES] > [FILENAME]

// Solve the generated problem
./tspSolver [FILENAME] [#of unchange] [#of population] [# of parents] [# of swap cities] > [OUTPUT_FILENAME]

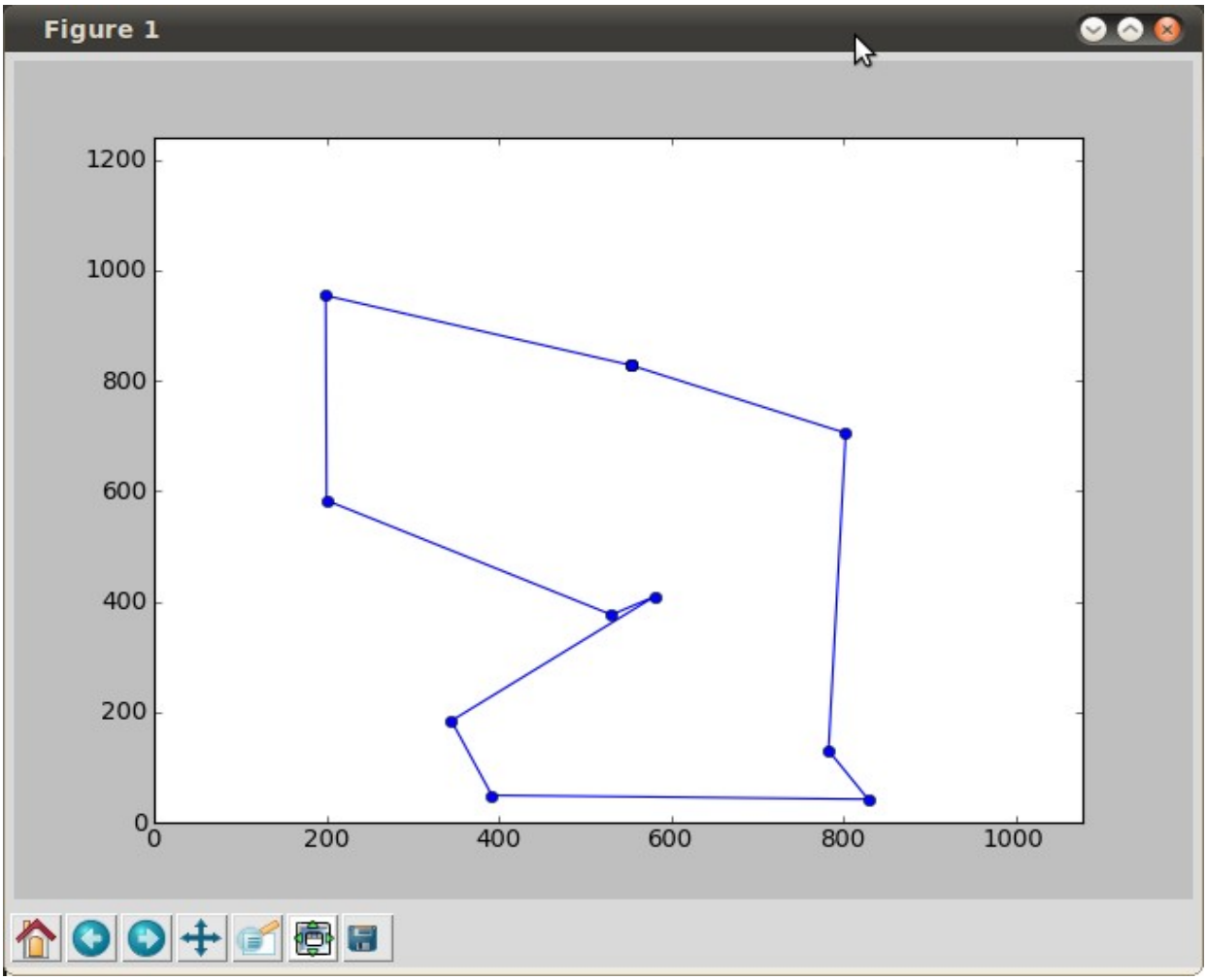
// Gisplay the result
python simplePlot.py [OUTPUT_FILENAME]

Example

./CitiesGenerator 10 > t1.tsp
./tspSolver t1.tsp 2000 20 3 5 > t1.sol
python simplePlot.py t1.sol



Input



Output

How does it work?
It follows the traditional GA algorithm, but I added some personal favors such as choosing half of routes that have the highest cost and half of the lowest cost and randomly mutate cities in a route.

Pseudo-code

```
procedure GA
      t=0;
      // randomly generate some initial problems
      initialize population P(t)

      // calculate the cost for these problems
      evaluation P(t)

      until Done { /* Based on some #of unchange paramater */
            t = t + 1
            /*
            select 50% highest cost and 50% lowest cost route as the
            parent
            */
            parent selection P(t)

            /*
            combine parents by swapping the cities in each tour so we divide a tour
            into 3 parts equally and swap the 1st part with the 3rd part
            */
            recombination P(t)

            /*
            randomly swap # of swap cities cities in each tour in the
            population. It doesn't matter if r is randomly generating
            twice or many times.
            */
            mutation P(t)

            evaluation P(t)

            /*
            pruned out the routes that have lowest cost in order to maintain
            the population
            */
            survival P(t)
      }
end procedure
```

Improvements
We could work on the parameter setting and find out the best parameters. Also, we could try to change the way to recombination and mutation. There are a lot of different ways to recombine and mutate.