Physical Traveling Salesman Problem

Sander Siim and Karl Tarbe, 1st year Computer Science Master's curriculum
Institute of Computer Science, University of Tartu, Tartu, Estonia

Study IT in .ee
sponsored by Skype™

UNIVERSITY OF TARTU

Introduction

The physical traveling salesman problem (PTSP) is an abstraction into the physical world of the well-known traveling salesman optimization problem.

- In regular TSP, the objective is to find a Hamiltonian path in a weighted graph with minimum total cost
- In PTSP, the goal is to minimize the time that it takes to follow the trajectory of the Hamiltonian path, taking into account physical laws of motion
- Solutions for TSP are in general not optimal for PTSP, since force needs to be applied to change direction of motion and sharp turns are costly

GECCO 2005 PTSP Competition

We followed the problem statement from GECCO'05 conference PTSP competition.

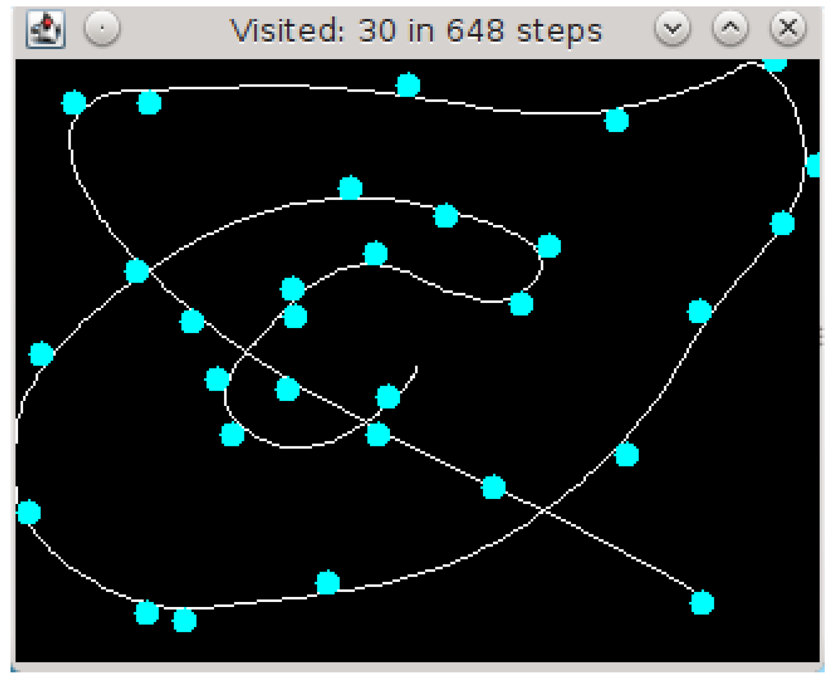

Figure: Best result from Martin Byröd (648 steps)

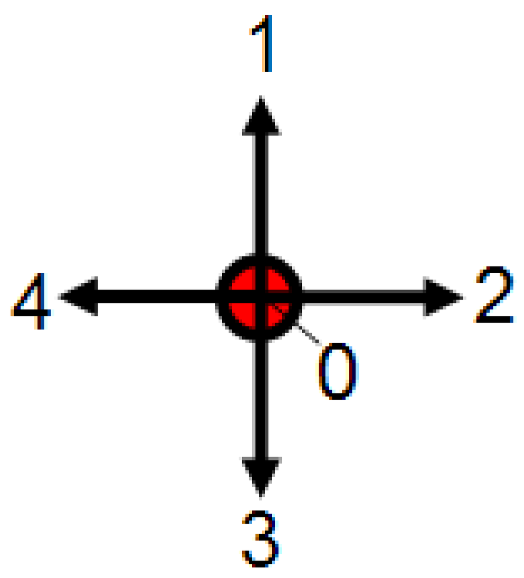

Figure: One of 5 force vectors are applied at every time step

Salesman has a mass of 1 kg. At every time step ($dt = \sqrt{0,1}$), force of 1N can be applied to salesman in either X or Y axis. Newton's equations for velocity and displacement are used for the simulation. Cities are placed in 320x240 grid and each has a radius of 5m.

Some observations about GECCO'05 solutions:

- All best solutions used genetic algorithms
- No crossover was used, only mutations
- Most tried to first find a suitable route for visiting the cities and then optimize the corresponding trajectory

Our Implementation

The natural approach to PTSP is to divide the problem into two phases. In the first phase, we find an optimal route for visiting all the cities. In the second phase, we optimize the corresponding trajectory i.e. the sequence of applied force vectors.

Finding optimal route

For finding a good route candidate, the heuristic should take into account both distances and angles between visited cities. We first experimented with approximating trajectories as circle arcs and using circular motion formulas to estimate traversal time. However, this did not prove to be a good objective estimate.

We ultimately used fitness function from Rok Sibanc's solution, which is defined for every three cities as

$$f_1(A,B,C) = \left(\sqrt{|\vec{AB}|} + \sqrt{|\vec{BC}|}\right) \cdot (3.4 + \cos(\angle ABC)).$$

We used a simple genetic algorithm by mutating the whole population and selecting the best candidates among parents and siblings for the next generation. Mutation was done by swapping two random cities in the route.

Generating optimal trajectory

Our first approach for optimizing the trajectory was to use a proportional-integral-derivative (PID) controller and evolving its parameters with a genetic algorithm. The reasoning was, that this would significantly narrow the search space. However, this had the disadvantage that the salesman always decelerated when approaching a city.

Later, we used a genetic algorithm with tournament selection and a sliding window directly on the force vector sequence. The fitness function f_2 is simple: for trajectory t, fitness is given by

$$f_2(t) = N + (1/d) + (1/T),$$

where N is the number of cities visited, d is the euclidean distance to the next city and T is total elapsed time.

Two types of mutations were used:

- Swap two adjacent force vectors
- Randomly overwrite a vector in the sequence

Best Achieved Solution

Our best solution solved the competition map in 703 steps.

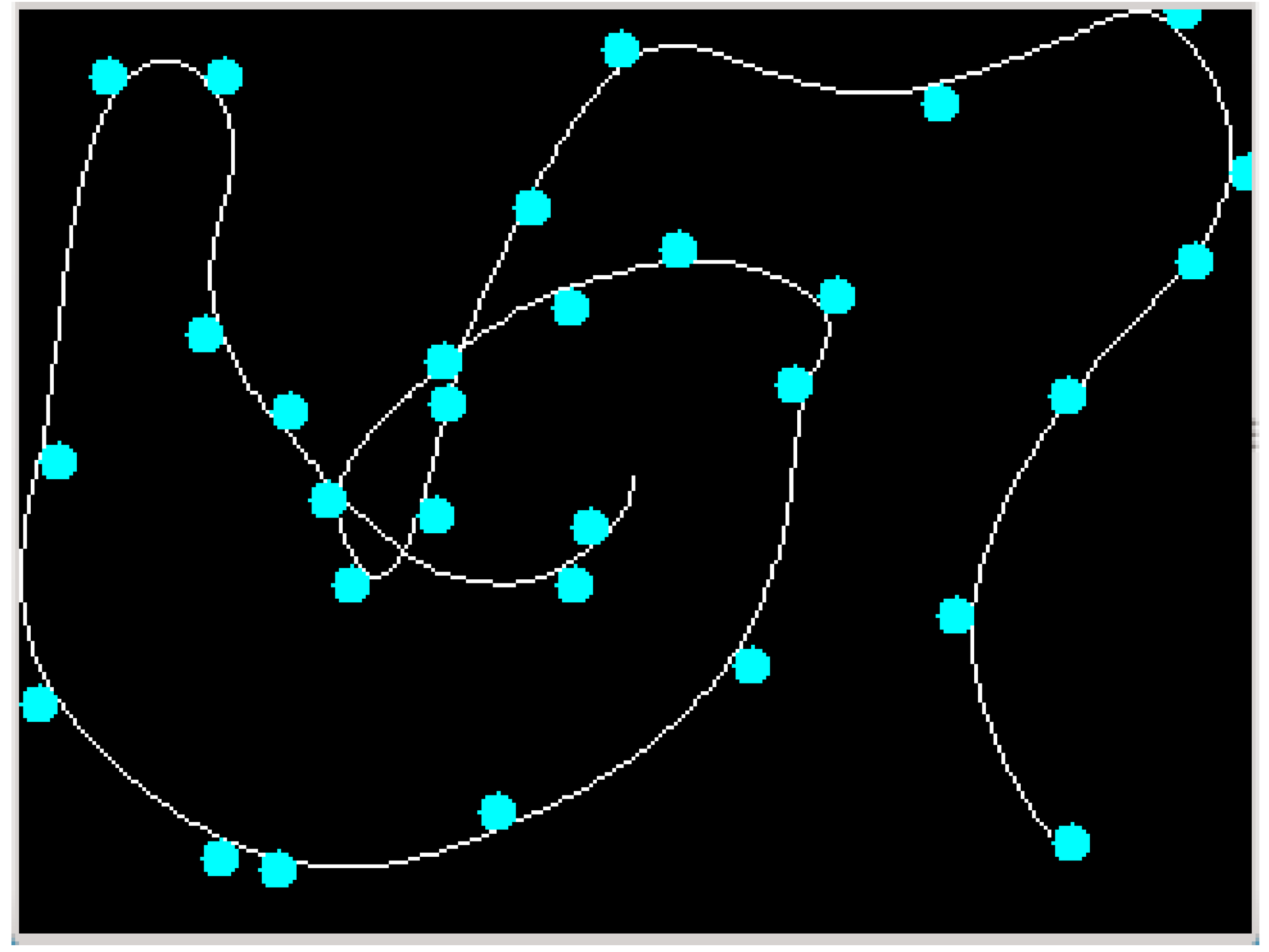

Figure: Our best solution, 703 time steps

Our result is only slightly behind from TOP 4 of GECCO'05. It can be seen that the route is optimized by favoring smooth curves over sharp turns. The total computation for the solution ran under a minute on Intel i5 4200u @1.60 GHz.

Conclusion

- Genetic algorithms have proven useful for PTSP problem, however the main difficulty is finding a subtle balance between diversity of population and convergence rate
- Our solution is definitely in the right direction, but the route could be improved with tweaking genetic algorithm parameters or using clever crossover operations
- A more intelligent fitness function could be constructed which approximates curvature of the whole trajectory
- An even challenging variant of PTSP includes real-time motion planning and obstacles on the map