ESSAY

The cheapest way to travel system

Prepared by: Tulebayeva Balganym

Akhanov Nurdaulet

Kenen Almat

Contents

1. PROBLEM DESCRIPTION
2. APPLICATION OF GENETIC ALGORITHM
3. TECHNOLOGIES
4. CONCLUSION

PROBLEM DESCRIPTION

The problem of getting from city A to city B by airplane seems to be easy problem if we have large budget and time. But what if we want to find a solution that balances between durability and cost? Many solutions focus only on finding the cheapest trip. In this project we will try to find solution that depends on both frontiers and that is close to satisfactory solution. We simply call that shortest path problem and will try to give definition to the shortest path in technical part of the solution.

Dijkstra’s algorithm is the most popular approach for solving this problem, but the algorithm is very time consuming in large graphs. So, our aim is not to find the best solution, but to end up with similar solution to the best one.

The shortest path problem is typical in the world of combinatorial systems. This project will attempt to apply a Genetic algorithm to solve this problem based on an air trip system. This is based on the analogy of finding the shortest path (i.e. the shortest possible distance) between two towns or cities in a graph or a map with potential connections (assuming that the path distances are always positive). Typically this is represented by a graph with each node representing a city and each edge being a path between two cities. So, applying a genetic algorithm is an interesting idea.

This is clearly different from traditional algorithms that try to compare every possibility to find the best solution, which might be a time consuming algorithm for a graph containing a large number of nodes and edges.

Genetic algorithm is search and optimization technique modeled from natural selection, genetic and evolution. It mimics natural evolution process for producing better results.

APPLICATION OF GENETIC ALGORITHM

A typical genetic algorithm requires:

1. a genetic representation of the solution domain (chromosomes)
2. a fitness function to evaluate the solution domain.

New offsprings are generated/evolved from the chromosomes using operators like *selection*, *crossover* and *mutation.*

We take a path between two cities A and B as a sequence of cities and we consider a flight between any two adjacent cities in this sequence. First city in this sequence will be our starting city A and last city will be our ending city B.

We take this sequence as a chromosome and each city in sequence as a gene. For practicability we consider sequences with length maximum of 10. Yet, we have to able to calculate fitness function for a chromosome. Without loss of generality we take fitness function as follows:

$$fitness = \frac{10000}{\sum dist(t_{i-1}, t_i)}$$

where

$$dist(x, y) = duration(x, y) * cost(x, y)$$

Indeed, if we consider duration of a flight, we can discover that the shorter the duration the more pleasant a flight. Same thing with the cost of a flight. So, we found that fitness is inversely dependent from both of the magnitudes. We take duration in hours and cost in dollars.

In addition, we do not want to work with extremely small numbers, that is why we multiply the value by some constant 1000.

Selection operation.

Selection operation in this project remains as it is in most of the applications of genetic algorithm. Chromosomes with higher fitness level have higher chances to be selected.

Crossover operation.

A crossover operation happens only between chromosomes that have cities in common. It follows up simple logic: we can not come up with better solution by mixing solutions that do not work with each other.

Example

Consider following two chromosomes.

We denote with **underlined bold** the chosen crossover point and with **bold** all possible crossover points.

1	2	**6**	8	**<u>4</u>**	10	9	**5**

1	**6**	3	7	**<u>4</u>**	11	**5**

One of the possible outputs would be:

1	6	3	7	4	10	9	5

1	2	6	8	4	11	5

Probability of crossover operation is 0.99.

Mutation operation.

Mutation operation means changing a city to another city in a sequence . And changing city means changing of exactly two flights.

Probability of crossover operation is 0.01.

TECHNOLOGIES

- *React, D3js* for visualization
- *Python* for parsing and computation
- some *public APIs* for parsing information about flights: http://www.itnext.in/articles/1003456/9-flight-apis-you-can-create-apps-on-to-get-live-flight-data

CONCLUSION

We believe that the proposed GA provides a good example of a sophisticated GA application for a representative combinatorial optimization problem and that some of the ideas can be successfully applied to the design of Genetic Algorithms for other combinatorial optimization problems.