## **Evolutionary Computation Final Project Proposal**

Finding k-shortest Paths in Network Using Genetic Algorithm 0416235 劉昱劭

[Click here to see shorter description]

## **Abstract**

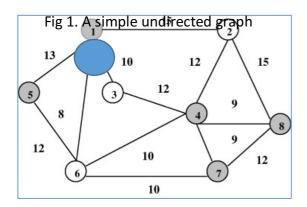
In network topology, we often want to find some critical paths to make better performance or higher utilization of throughput. There are many real time multimedia services, like video streaming, need to guarantee stable bandwidth of data transmission. For this purpose, we need to find some paths with large enough bottleneck bandwidth, and these paths should be as short as possible for less transmission latency. If there are many paths acceptable, we could take these paths as redundancy to avoid congestion.

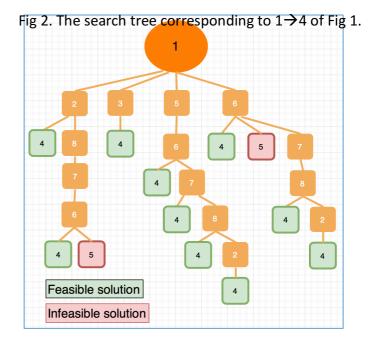
To find **k shortest paths** with **bandwidth constraint** is a significant issue.

## **Problem definition**

Given a network topology (or a graph) G(N, E), a source node S and a destination node D. There is a user-specified parameter k to decide at most how many paths we should find.

The whole search space could be represented by a search tree e.g. Given **G(N,E)** = Fig.1, **Source** = 1, **Destination** = 4. The whole search space is showed as Fig.2.





# Method – Genetic Algorithm

- Individual representation
  - Integer array
  - Each individual shows a path from the source node to the destination node.

#### Crossover

- Use one-cut point
- e.g. 2 parent individuals (2 paths)
  - $\circ\;$  we could cut these 2 individuals at both  $x_3$  and get

<b>X</b> <sub>1</sub>	X <sub>2</sub>	х <sub>3</sub>	X <sub>4</sub>	<b>X</b> 5	x <sub>6</sub>
Source	3	5	6	4	Destination

<b>x</b> <sub>1</sub>	X <sub>2</sub>	x <sub>3</sub>	<b>X</b> 4	<b>X</b> 5	<b>x</b> <sub>6</sub>	<b>x</b> <sub>7</sub>
Source	5	9	8	12	11	Destination

• we could cut these 2 individuals at the same node and get 2 children:

<b>X</b> <sub>1</sub>	X <sub>2</sub>	x <sub>3</sub>	X4	<b>x</b> <sub>5</sub>
Source	5	6	4	Destination

<b>X</b> <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	<b>X</b> 5	x <sub>6</sub>	<b>x</b> <sub>7</sub>	x <sub>8</sub>
Source	3	5	9	8	12	11	Destination

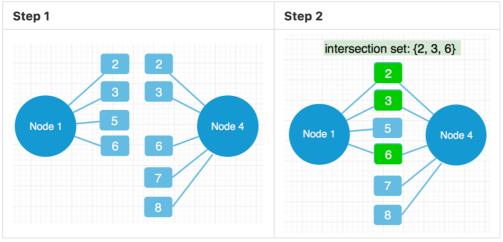
#### Mutation

• individual:

■ now mutate x<sub>2</sub>=3

<b>x</b> <sub>1</sub>	X <sub>2</sub>	х <sub>3</sub>
1	3	4

o Bidirectional BFS to get common neighbor



 $\circ$  mutate  $x_2$  to one of  $\{2, 3, 6\}$ 

<b>x</b> <sub>1</sub>	X <sub>2</sub>	х <sub>3</sub>
1	6	4

## Initialization – Seeding by Iterative-Deepening DFS

- Because the search space could be represented by a search tree, we could apply BFS or DFS on the
  given graph, eventually the search space could be traversed. BFS & DFS could be treated as exhaustive
  search or brute force.
- Let's meet them halfway. Iterative-Deepening DFS is DFS with a max depth  $d_{max}$ , then we could get the all branches within depth  $d_{max}$ , which takes advantages of the width of BFS and the speed of DFS.
- Initialization using Iterative-Deepening DFS (IDDFS)
  - To guarantee diversity and include known good solutions to population.

## **Fitness function**

- Maximize  $f_1(x) = min \{ bandwidth(x) | x \in E_p \} \ge B$ 
  - $\blacksquare$   $f_1(x)$  is the bottleneck bandwidth in the path  $E_p$ . B is the threshold bandwidth.
- Minimize  $f_2(x) = |E_p|$ 
  - lacksquare  $f_2(x)$  represents the number of nodes in the path  $E_p$ , or the path length of  $E_p$ .

We could only optimize  $f_1(x)$  so that this problem could be treated as single objective problem, or optimize both  $f_1(x)$  &  $f_2(x)$  so that this problem becomes multi-objective problem.

## **Related Work**

For KSP problem, Yen's algorithm is very famous and efficient.

Here is a GitHub repository that implements Yen's algorithm in python <a href="https://github.com/Pent00/YenKSP">https://github.com/Pent00/YenKSP</a>
I will compare results of my project with this YenKSP (If I have enough time...)

#### **Future Work**

This work could be regarded as a template of path finding problem. If we change the objective function, this work could have different behavior.

■ E.g. Minimize  $f_3(x) = \sum_{x \in E_p} cost(x)$ It will become a GA to find the shortest path.