

- 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)
 - we could cut these 2 individuals at both x_3 and get

x_1	x_2	x_3	x_4	x_5	x_6
Source	3	5	6	4	Destination

x_1	x_2	x_3	x_4	x_5	x_6	x_7
Source	5	9	8	12	11	Destination

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

x_1	x_2	x_3	x_4	x_5
Source	5	6	4	Destination

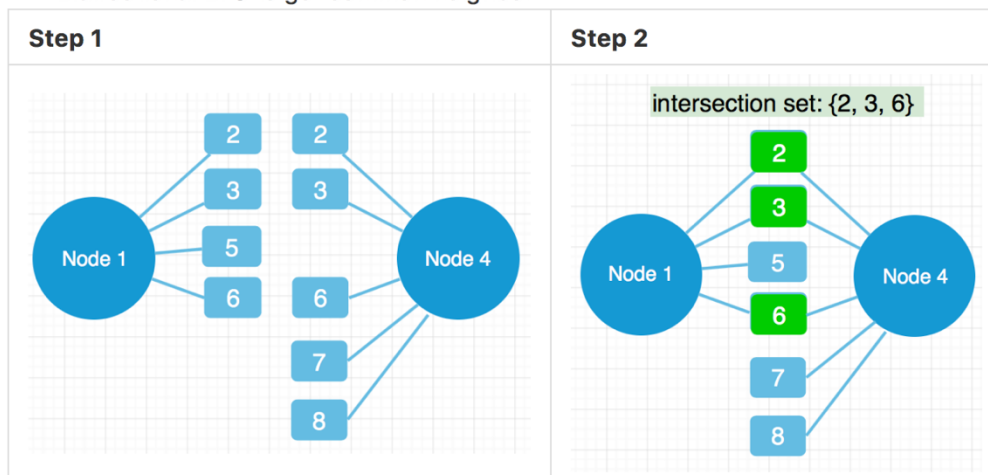
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Source	3	5	9	8	12	11	Destination

● Mutation

- individual:
 - now mutate $x_2=3$

x_1	x_2	x_3
1	3	4

- Bidirectional BFS to get common neighbor



- mutate x_2 to one of {2, 3, 6}

x_1	x_2	x_3
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 $f_1(x) = \min \{ bandwidth(x) \mid x \in E_p \} \geq B$
 - $f_1(x)$ is the bottleneck bandwidth in the path E_p . B is the threshold bandwidth.
- Minimize $f_2(x) = |E_p|$
 - $f_2(x)$ 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

<https://github.com/Pent00/YenKSP>

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.