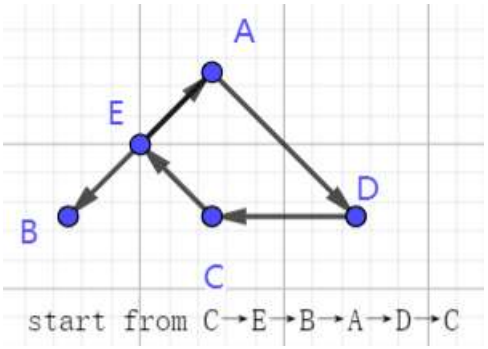
Optimization Methods

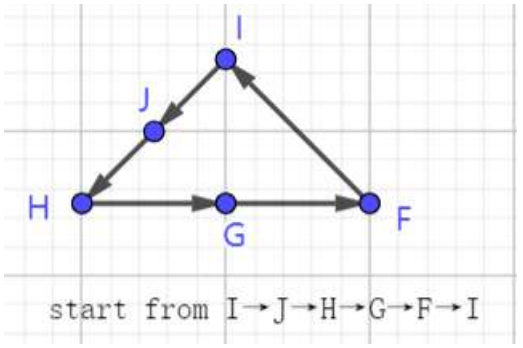
Lab 4 Session

12112218 史纪茂

Task 1

Create a TSP problem where a non-optimal greedy solution is obtained from an initial city and the obtained greedy solution cannot be improved by inversion-based local search. It is enough to show that one greedy solution cannot be improved. You do not have to show that any greedy solution from an arbitrary initial city cannot be improved.





Notify: BE<AE

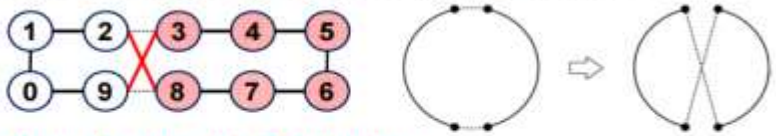
In left one, we let it should start from E. After using greedy algrithom, we get this solution which can't be improved by inversion-based local search. At the same time, it is not the optimal solution.

In the right one, it start from I, and then do as the upon graph. And this is the optimal solution.

Lab Session Task 2

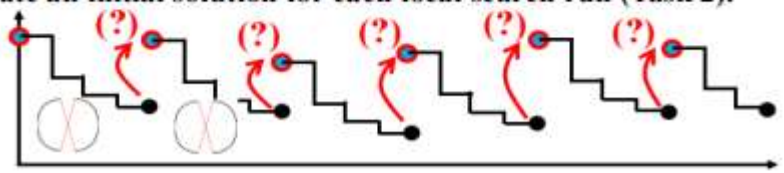
Local search

- Neighborhood structure: Inversion (two-edge change)



Creation of a new initial solution:

- 1st initial solution: Greedy solution
- Other initial solutions: Explain your own idea about how to create an initial solution for each local search run (Task 2).

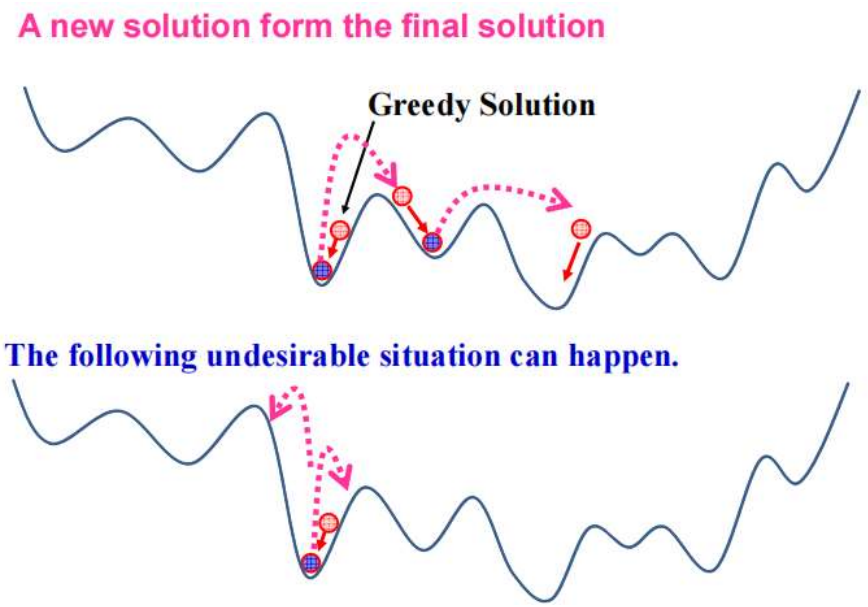


First, we represent the solution of TSP by string. Define the distance between 2 points as the closest length between 2 points.
In the following example, we can find that point 1 and 4 have a path with the length of 2.

1	2	3	4	5	6
4	2	3	1	5	6

Therefore, we can create our initial solution by the following instruction.

To create an initial solution for a new local research run, we can exchange the positions of two points which are originally 2 or more apart.



The new initial solution can be far enough away from the last solution, because we exchange 2 edges for 5 times at least to restore the original state. The inversion-based distance between the new initial solution and the last final solution is 5 at least.
Therefore, it is enough for us to jump out of the local optimal trap.

Lab Session Task 3

Specify neighborhood structures $\mathcal{N}_1$, $\mathcal{N}_2$, $\mathcal{N}_3$, ... for designing a Variable Neighborhood Descent (VND) algorithm for a TSP problem.

Since we need a set of neighborhood structures which become larger and larger, we can consider the following one.

N_i means arbitrary select i points to exchange their index.

We can clearly say that the size of neighborhood structures increase as long as i increase. At the same time, we can guarantee that with the increase of i, when i become the total number of points n, the neighborhood structure is equal to the whole solution space of TSP problem.