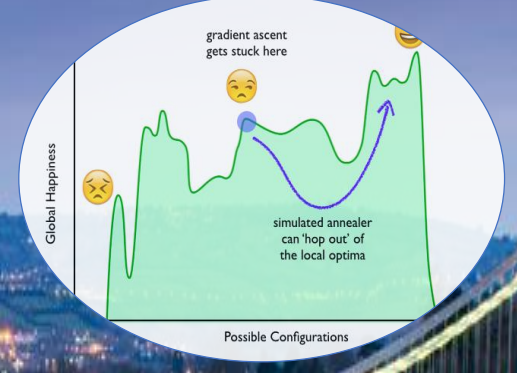
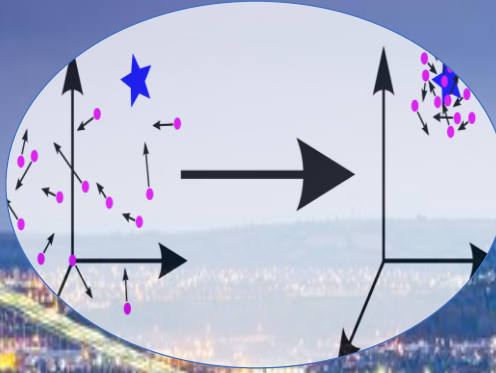


## Reason for simulated annealing algorithm and Particle Swarm Optimization

First, the structure of these two algorithms are easy to understand with a simple logic. Second, simulated annealing algorithm accepts a worse solution based on a probability, which is able to avoid a local optimum and obtain a global optimum probably. Third, particle swarm optimization is a parallel algorithm with particles best solution and best known solution, it is able to accelerate the speed finding the global optimal solution.

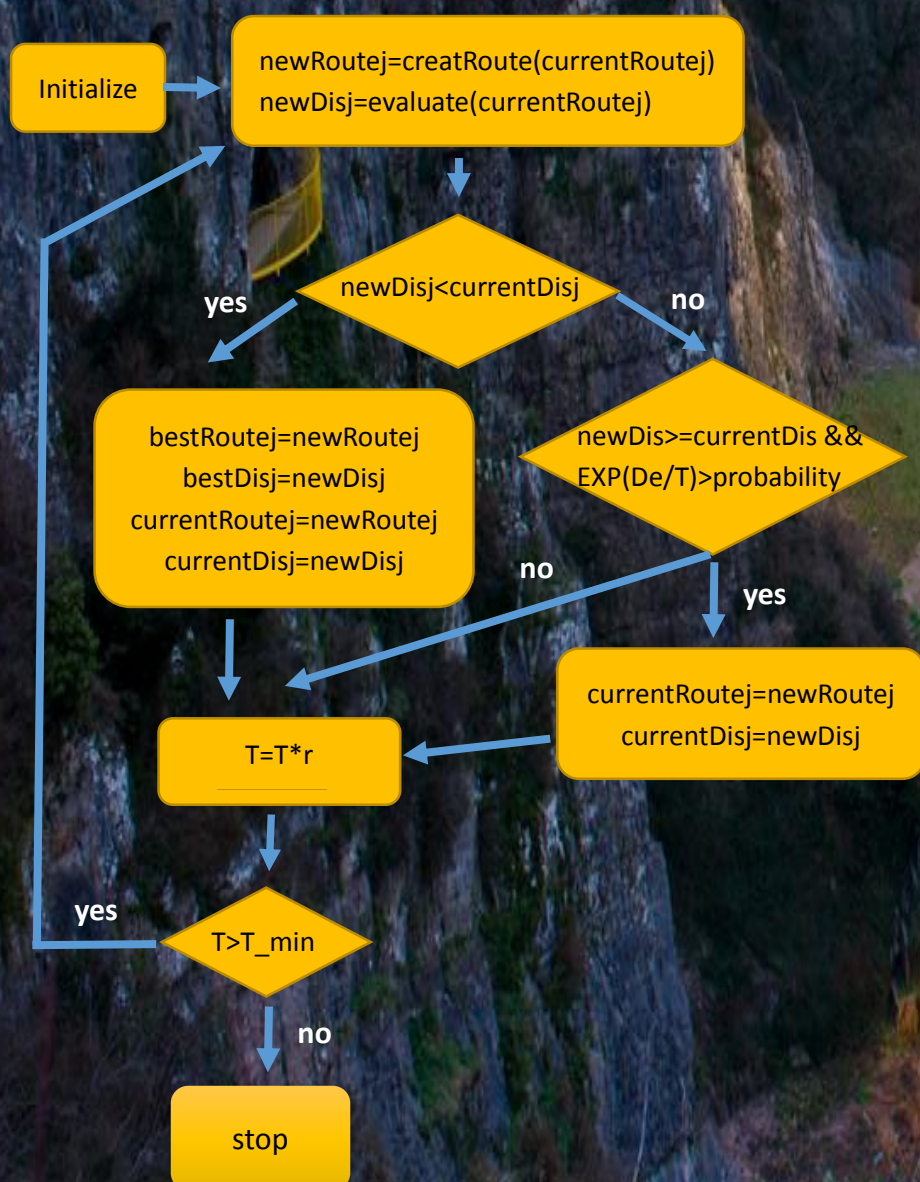


## Source of this algorithm

I combined the idea and advantages of both the algorithms and modified some parts, for example I modified the new solution creation function with a random inverse permutation in the annealing algorithm. As both the algorithms are ordinary classic algorithms, I relearned them in the Wikipedia.

## Details and explanation of the algorithm.

runAnneal(): used to run the annealing algorithm.  
creatRoute(): create a new solution based on the current solution  
invPermutate(): used in the createRoute() to make an inverse permutation of a part of the solution.  
evaluate(): evaluate the cost of the solution  
showRoute(): show the details(routes) of the best solution



## Parameters setting

currentRoutej[]: the current solution of the problem of particle j  
newRoutej []: a new solution created from currentRoute[] of particle j  
int bestRoutej [] : the best solution which updating  
currentRouteDis j, newRouteDis j, bestRouteDis j: the cost of solution  
r: cooling rate for controlling the speed for annealing  
T: the the initial temperature of the annealing algorithm  
T\_min : the min of temperature for stopping the annealing  
dE j : the temperature variation of particle j

