7- Tabu Search: main principles and convergence

Used for quadratic assignment problem and TSP

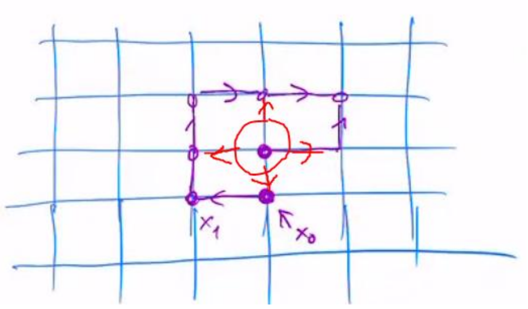
- The search space is explored by going from neighbour-to-neighbour xn → xn+1 ∈ V(xn)
- The xn+1 is chosen by the search operator, as the non-tabu xn+1 that locally optimizes fitness -> (independent of the current fitness of xn -> can be worse) -> if many points have the same fitness, it is random choice
- A tabu list, is a list of already explored states, so that we don't go back
- However, it is possible for a state to leave the tabu list -> after a certain amount of iterations

Tabu list -> prevents us from going back

The fact that xn+1 can have worse fitness than xn means we won't get stuck in local max

A state can go out of tabu list, so that the code doesn't get stuck

- This is why we need the list to be updated (if we are in red state, we are stuck):



Convergence:

To converge means to find the global optimum!!

-tabu search will converge if:

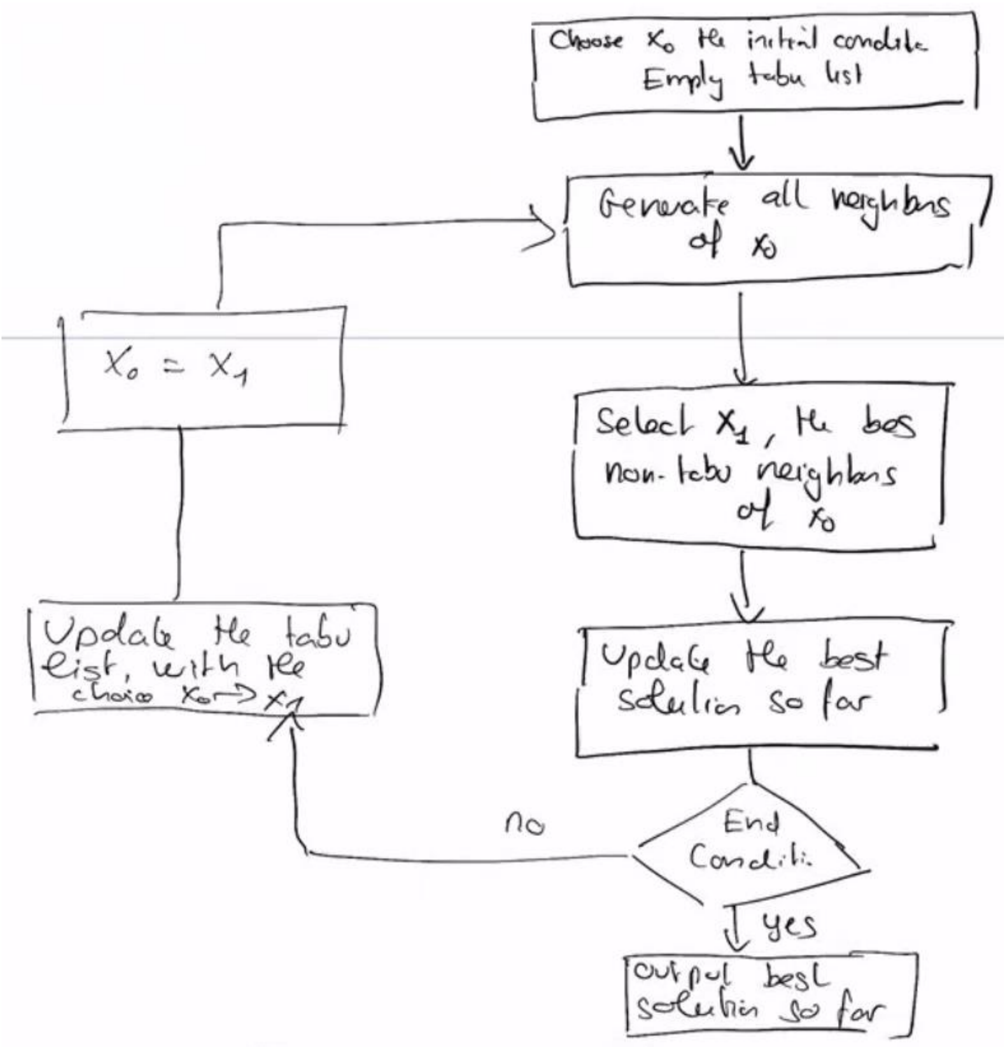
- the search space is finite

- neighbourhood is symmetric: x in v(y) ⇔ y in v(x)

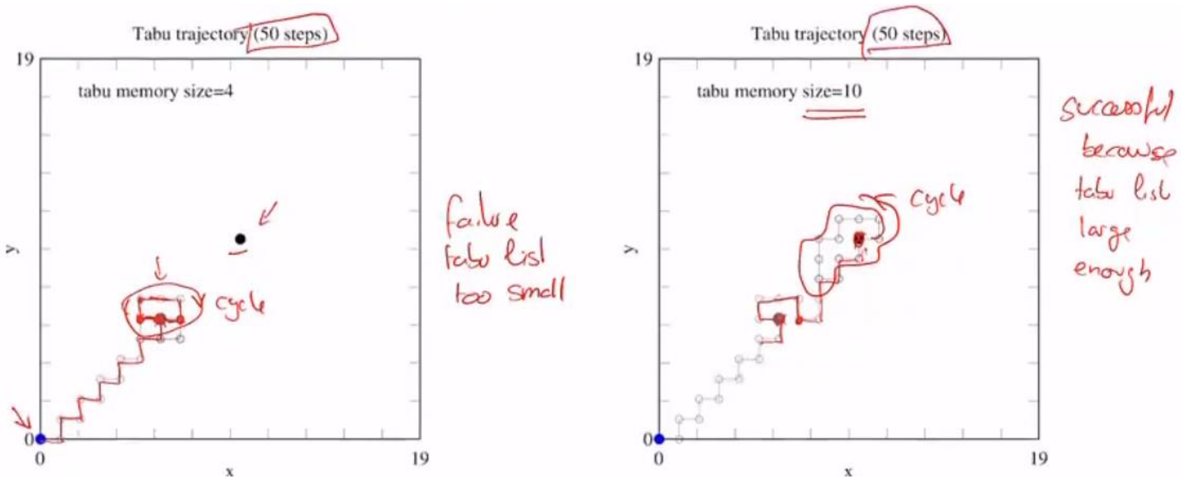
- any state in the search space is reachable by a finite number of steps

- Memorizes all the visited points, but allows the trajectory to use the oldest tabu point to unlock itself -> will visit all the search space -> and find the optimal solution

Flowchart of tabu list algorithm:


Choose x0 the initial condit.
Empty tabu list
Generate all neighbors of x0
Select x1, the bes non-tabu neighbors of x0
Update the best solution so far
End Conditi.
yes
no
output best solution so far
Update the tabu list, with the choice x0 -> x1
X0 = X1

Example of the same search space, with 2 different tabu list memories:


Tabu trajectory (50 steps)
tabu memory size=4
x
y
cycle
failure
tabu list
too small
Tabu trajectory (50 steps)
tabu memory size=10
cycle
successful
because
tabu list
large
enough