

8- The different ways to implement the tabu list

The goal of the tabu list is to prevent search to explore solution that have been seen previously

There are many versions of a tabu list:

- ➔ Visited solutions kept in tabu list for N iterations
- ➔ Keep in tabu list all states that have the same fitness as already visited states (if we have visited a state with fitness n, then all neighbours with fitness n go into the tabu list)
- ➔ Movements (of transformations) that is forbidden, for example, it is impossible to go left for the next 2 moves -> TO avoid going back to x if we just moved from x to x', the inverse movement is in the tabu list

Updating the tabu list -> adding and removing items: short- and long-term memory

- ➔ Short term memory:
 - Use a finite size tabu list, so as we add to the tabu list, the oldest item is deleted
 - Associate a duration to the tabu items (banned time), a movement is banned for N iterations -> N is defined based on knowledge of the problem or by trial and error
 - ➔ Long term memory
 - Used to prevent that some movements are never used; the long-term memory can disregard the forbidden movement in the short-term list
 - To implement the long-term memory, we track the movements / states / transitions never done/visited, and at a certain point we force the algorithm to visit them
- ➔ Short list = exploration -> no diversification
- ➔ Long list = exploitation -> with diversification

- We can also put fitness above all, making it ok to revisit a state that has a higher fitness, even if it was in the taboo list

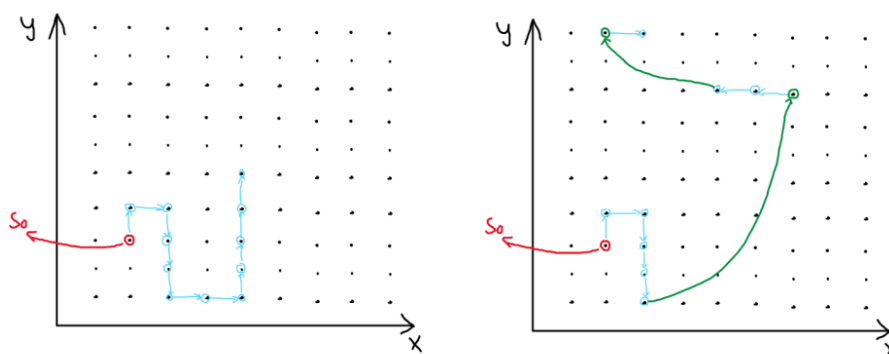


Figure 3: Exploration of Z^2 plane, left without diversification, right with diversification