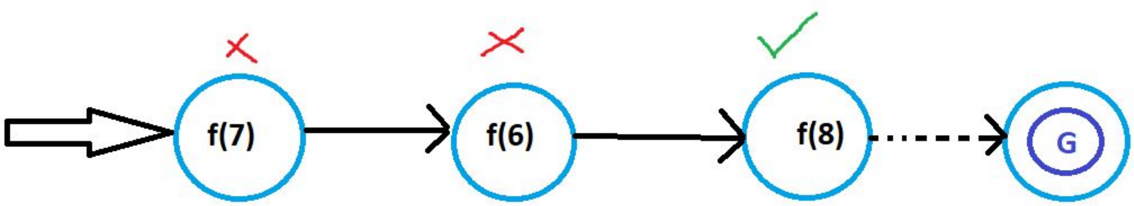
Assignment 11 Hill Climbing

Make yourself familiar with the Hill Climbing Algorithm as introduced in the lecture.

- What are the advantages and disadvantages of using the Hill Climbing algorithm as optimization algorithm?

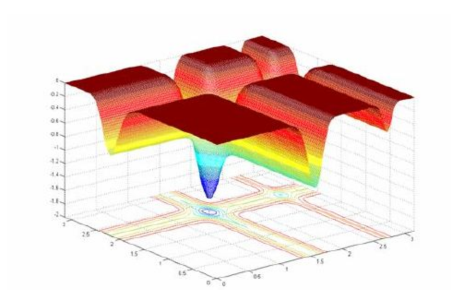
Advantages:

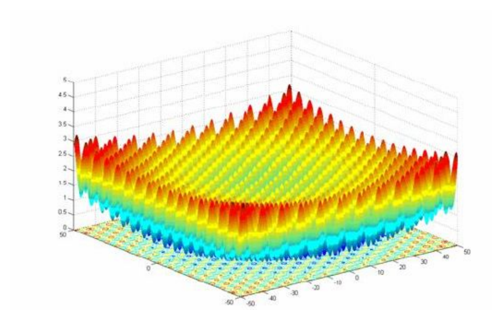
Local method: Decides what to do next by looking at the **immediate** step.



Disadvantages:

1) Random initializations (start), due to which repetitions are required to find out whether global optimum is reached.
2) Sampling strategy, (randomly or deterministically) results in time and budget to find a satisfactory neighbourhood.
3) Fitness Landscapes: Often has Local maxima, Plateaus and Ridges.





- Make yourself familiar with the Traveling Salesman Problem (TSP) as introduced in the lecture. Your next task is to apply the Hill Climbing algorithm to the TSP. The distance matrix mat_{dist} indicates the distances between the cities c_1 to c_5:

$$\text{mat}_{dist} = \begin{array}{cc} & \begin{array}{ccccc} c_1 & c_2 & c_3 & c_4 & c_5 \end{array} \\ \begin{array}{c} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{array} & \begin{bmatrix} 0 & 110 & 350 & 220 & 70 \\ 110 & 0 & 455 & 260 & 170 \\ 350 & 455 & 0 & 420 & 490 \\ 220 & 260 & 420 & 0 & 170 \\ 70 & 170 & 490 & 170 & 0 \end{bmatrix} \end{array}$$

  where c_1: Los Angeles, c_2: San Diego, c_3: San Francisco, c_4: Las Vegas, c_5: California City.

  The Salesman does not need to come back to the city where he started, i.e. after visiting all five cities he will stay in the last city.

  Use the permutation representation of the TSP and exchange two cities as a neighborhood function. The fitness function of the problem is the sum of all distances, which should be minimized. Start with c_2-c_4-c_1-c_5-c_3 as initial solution x_c and apply the algorithm until the stopping criterion is fulfilled.

Ordering distances from lowest to highest: 70<110<170<220<260<350<420<455<490

c2-c4-c1-c5-c3

procedure hill-climber
begin
  local ← FALSE
  x_c ⟵ **c2-c4-c1-c5-c3**
  repeat
    select x_n = **c2-c1-c4-c5-c3**
    if **990** $\leq$ **1040**
      then $x_c = x_n$
    else *local* ← TRUE
  until *local*
end

c2-c1-c4-c5-c3

procedure hill-climber

begin

local ← FALSE

x_c ← **c2-c1-c4-c5-c3**

repeat

select x_n = **c2-c1-c5-c4-c3**

if **770** $\leq$ **990**

then $x_c = x_n$

else *local* ← TRUE

until *local*

end

c3 has the highest cost so not desirable to start with for cost optimization.
All other permutations yield a distance higher than 770.
Inspecting current solution with most of the least cost (green region).
So the final solution which yields the minimal distance is C_2-C_1-C_5-C_4-C_3