Stochasticity and Simulated Annealing

Alexander G. Ororbia II

Biologically-Inspired Intelligent Systems

CSCI-633

1/29/2026

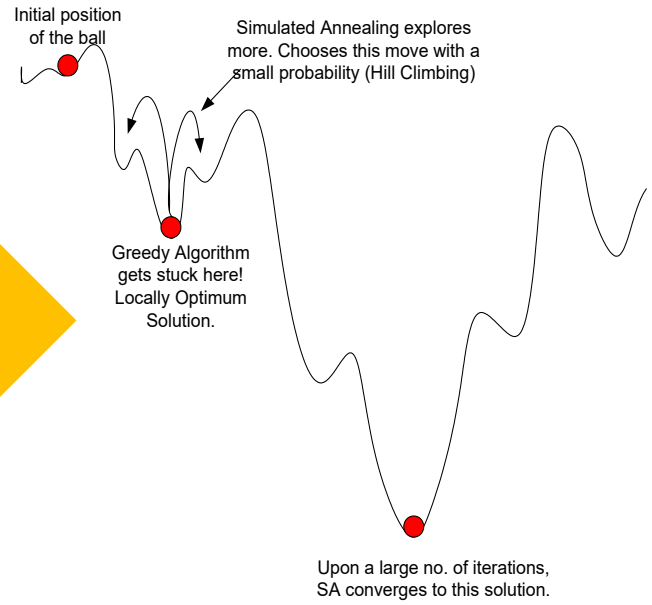
Towards Simulated Annealing (SA)

- Hill-climbing that never makes downhill (or “bad”) moves, gets stuck in local maxima easily (incomplete)
- Pure random walks (moving to successor state uniformly at random from set of successors) is complete BUT extremely inefficient
- Why not combine both? You get SA (“shaking” algo)
- We introduce temperature *T* that decreases over time
  - At start, we allow for move bad moves when T is high
  - Towards end, we permit few(er) bad moves when T is low
- VSLI layout problems (historical), factory layouts, etc.

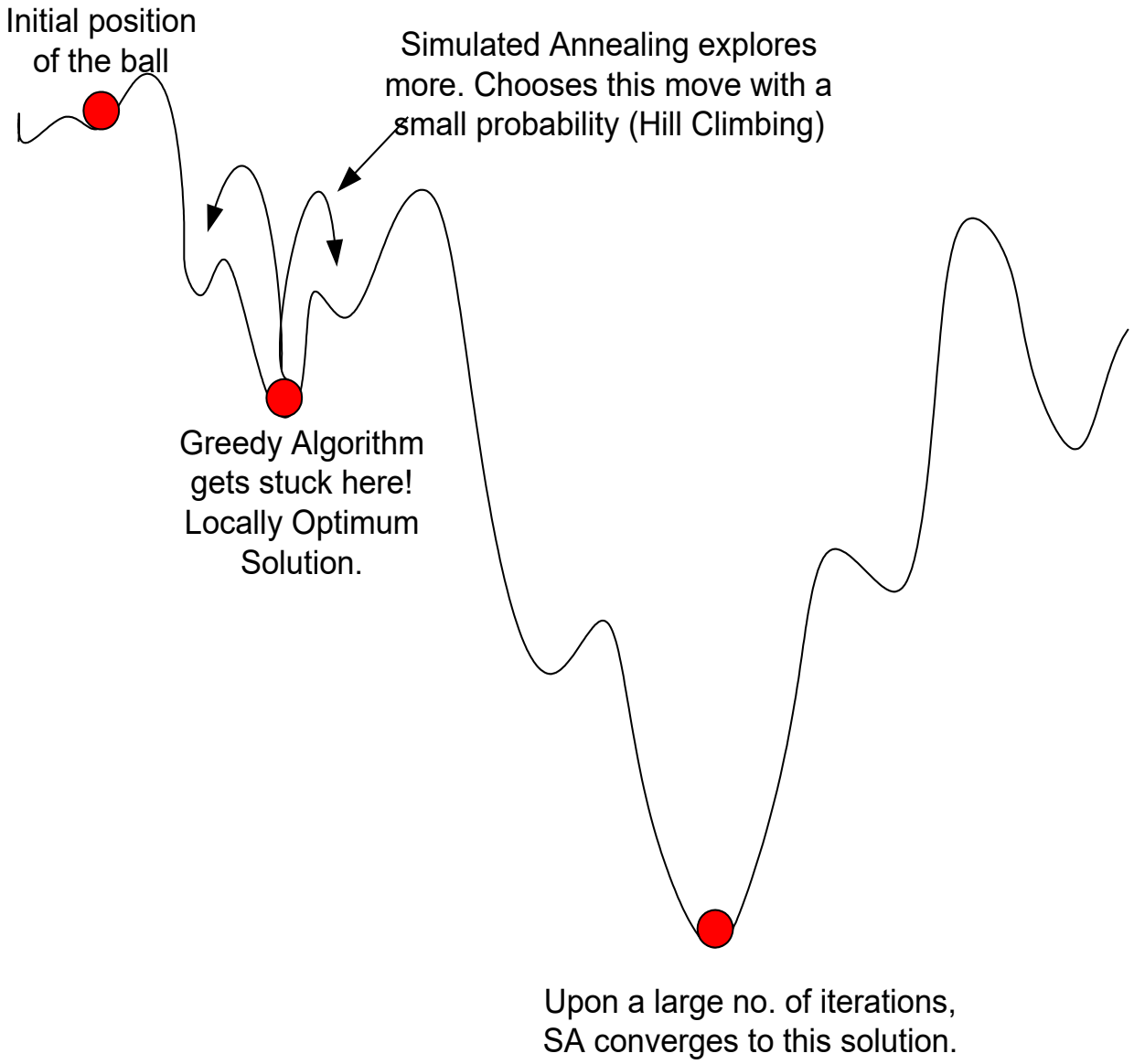
Simulated Annealing Mechanics

Simulated Annealing (SA)

- The “shaking algorithm”
- Hill-climbing + controlled random walks
  - Accept all good moves and some bad ones (over time, controlled by temperature *T*)



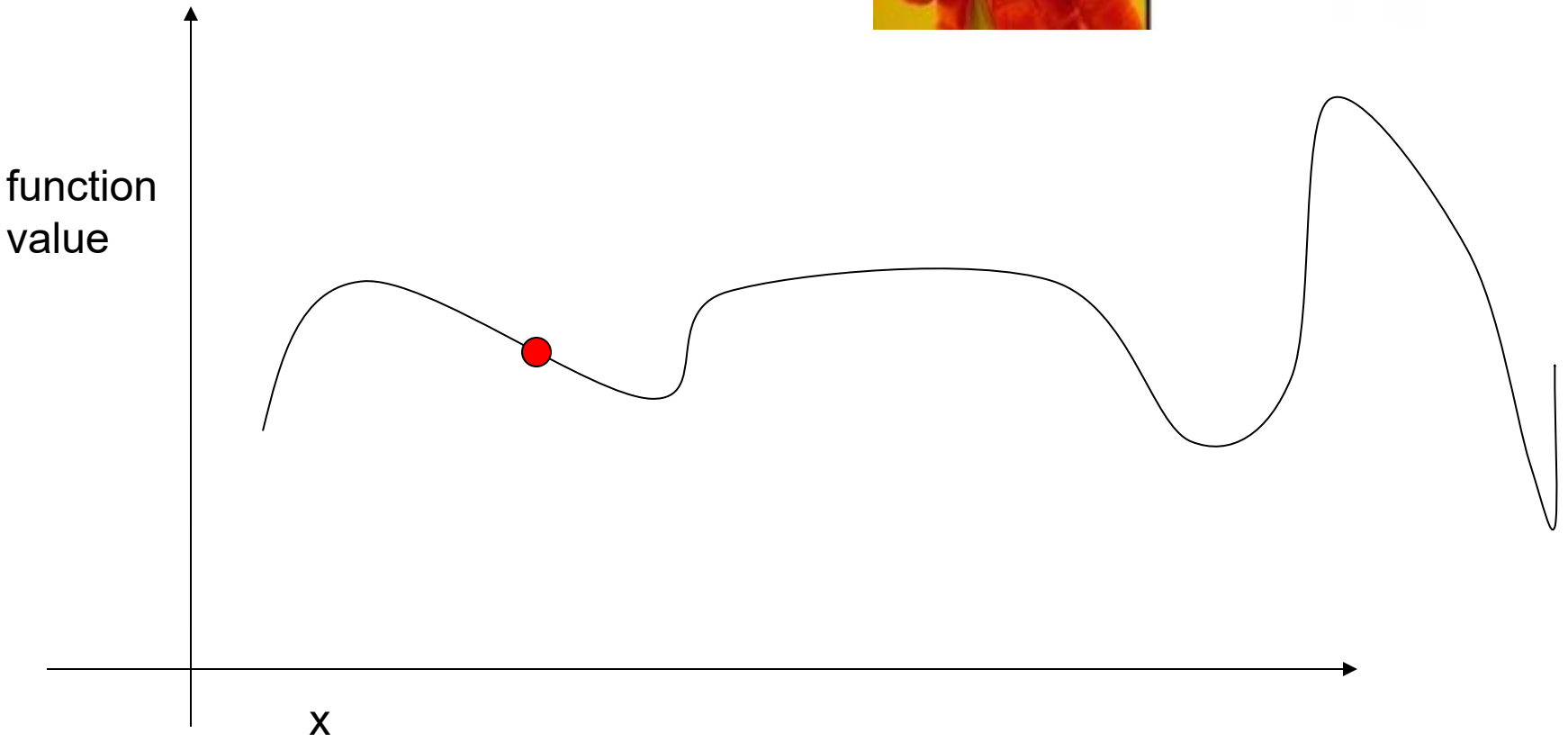
“Ball on Terrain” Example – SA vs Greedy Hill-Climbing



Simulated Annealing

- Simulated Annealing (this will be you, soon enough!)
- SHC + Restarts (this is you now, as you work on HW 1)

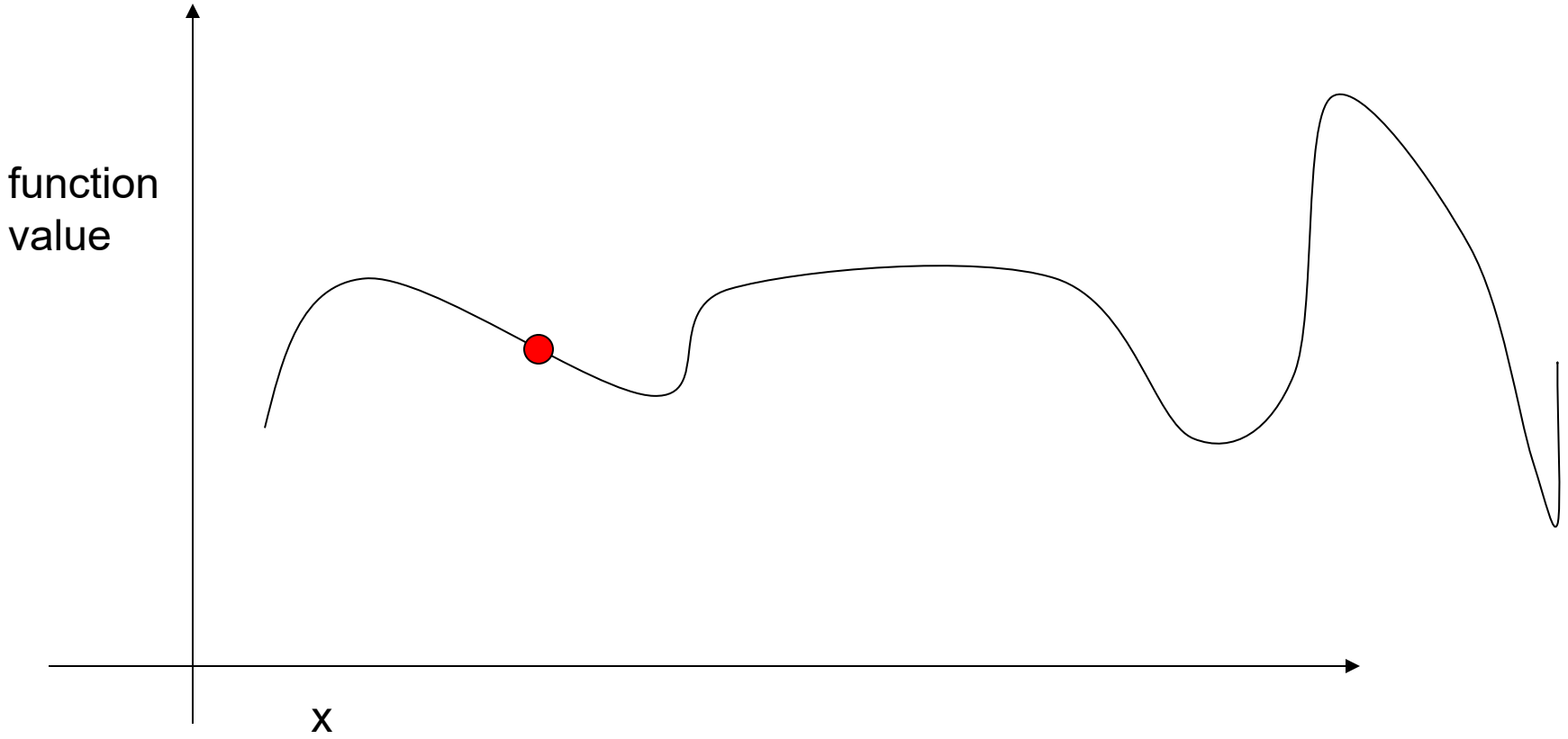
- Random Starting Point



Simulated Annealing

T = Very High

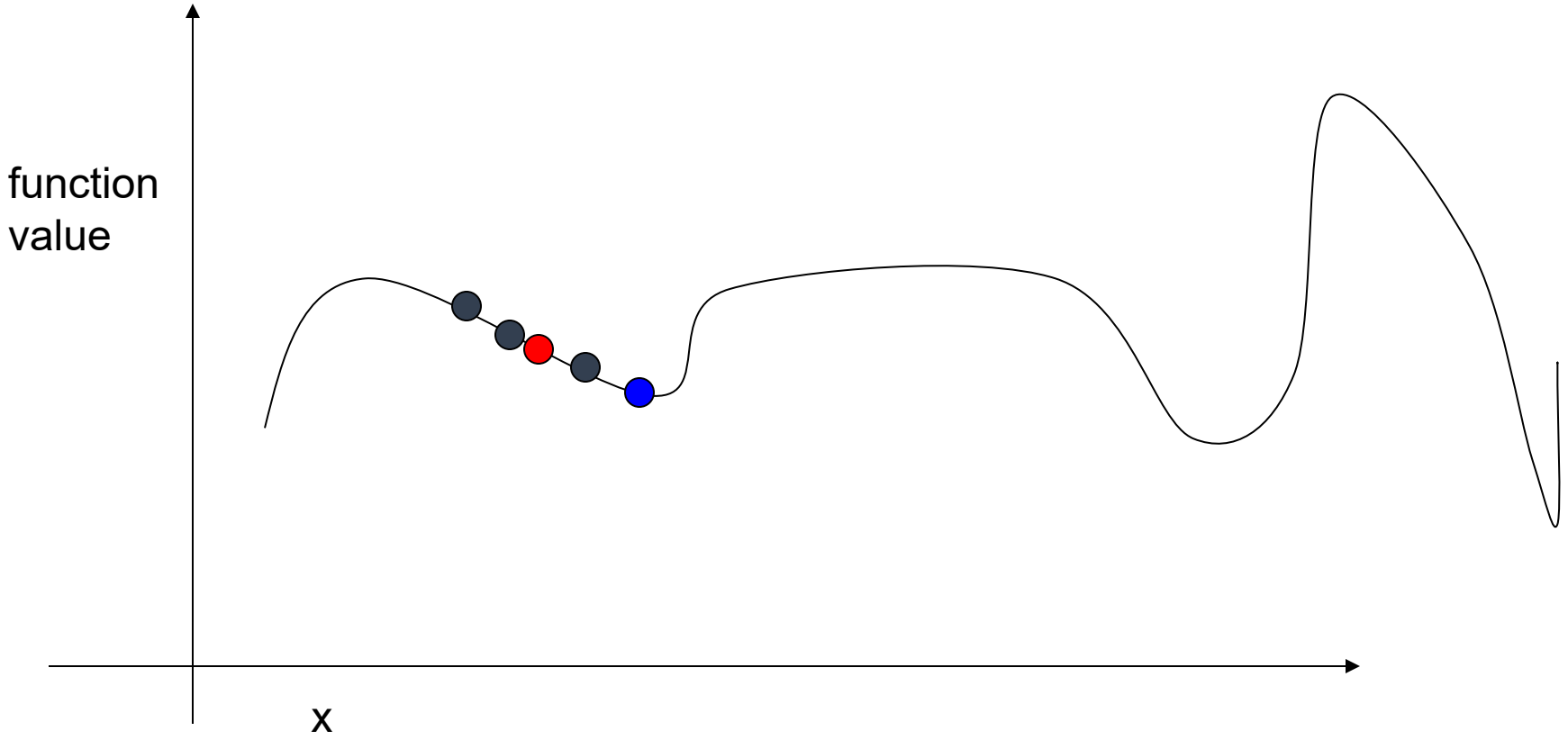
- Random Starting Point



Simulated Annealing

T = Very High

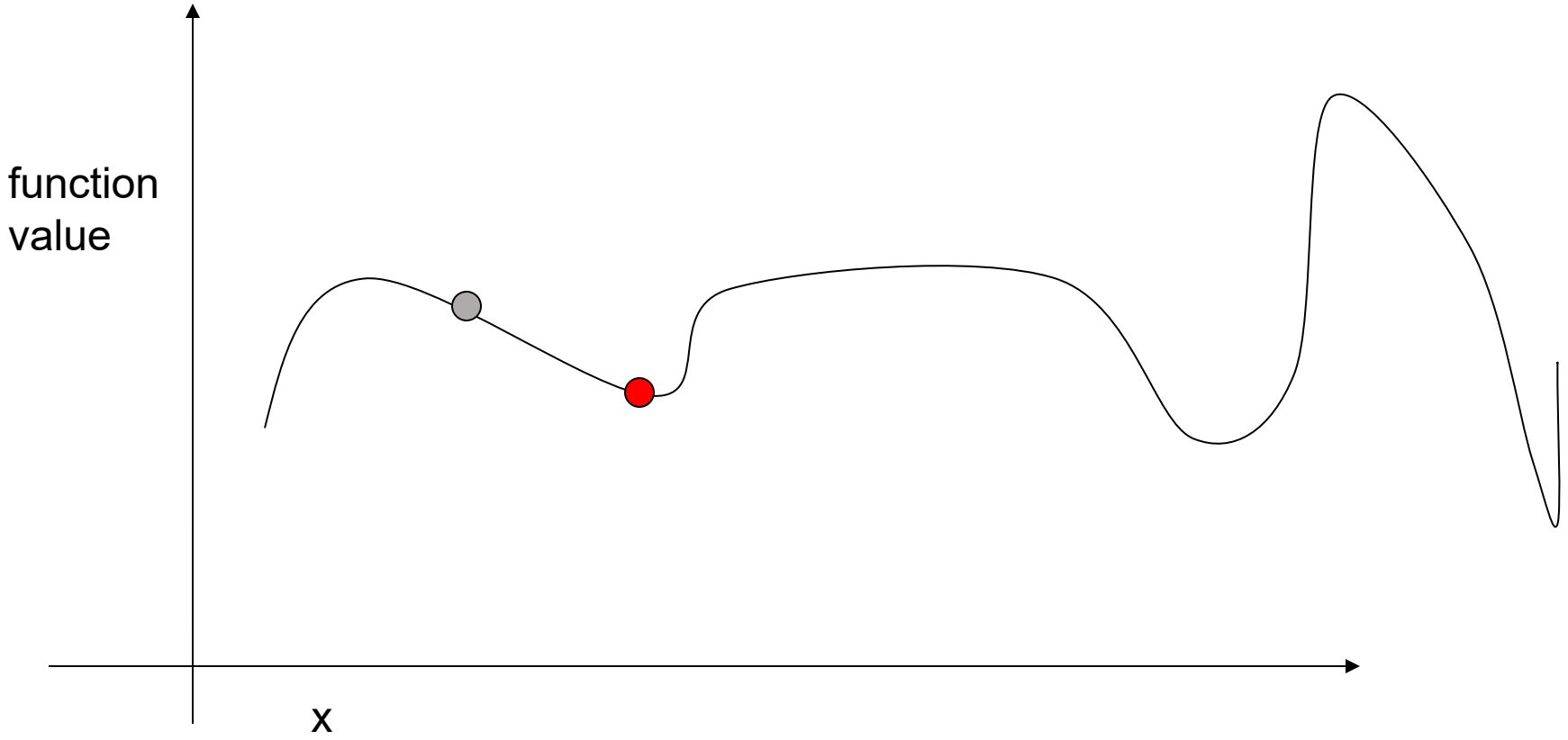
- Random Step



Simulated Annealing

T = Very High

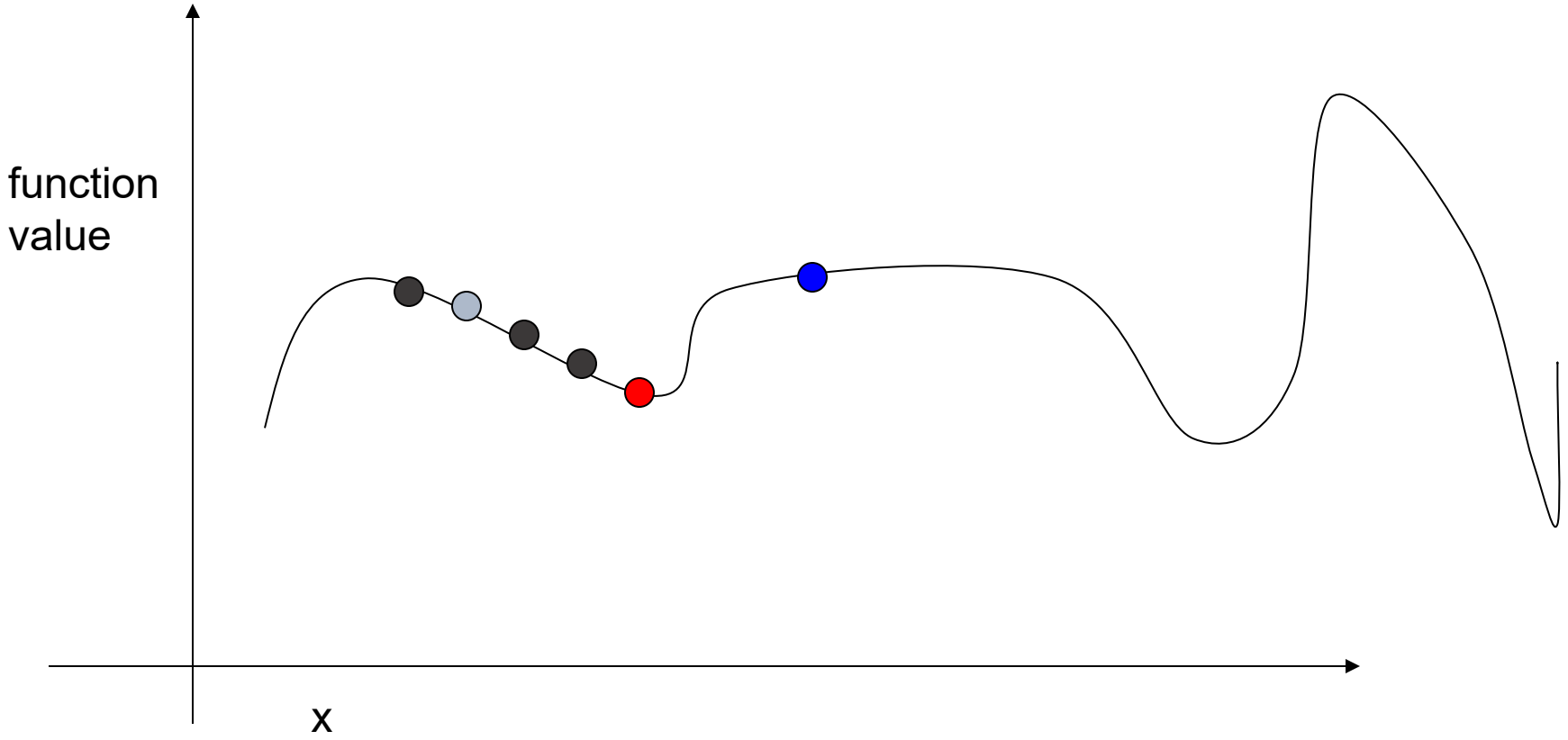
- Even though E is lower, accept



Simulated Annealing

T = Very High

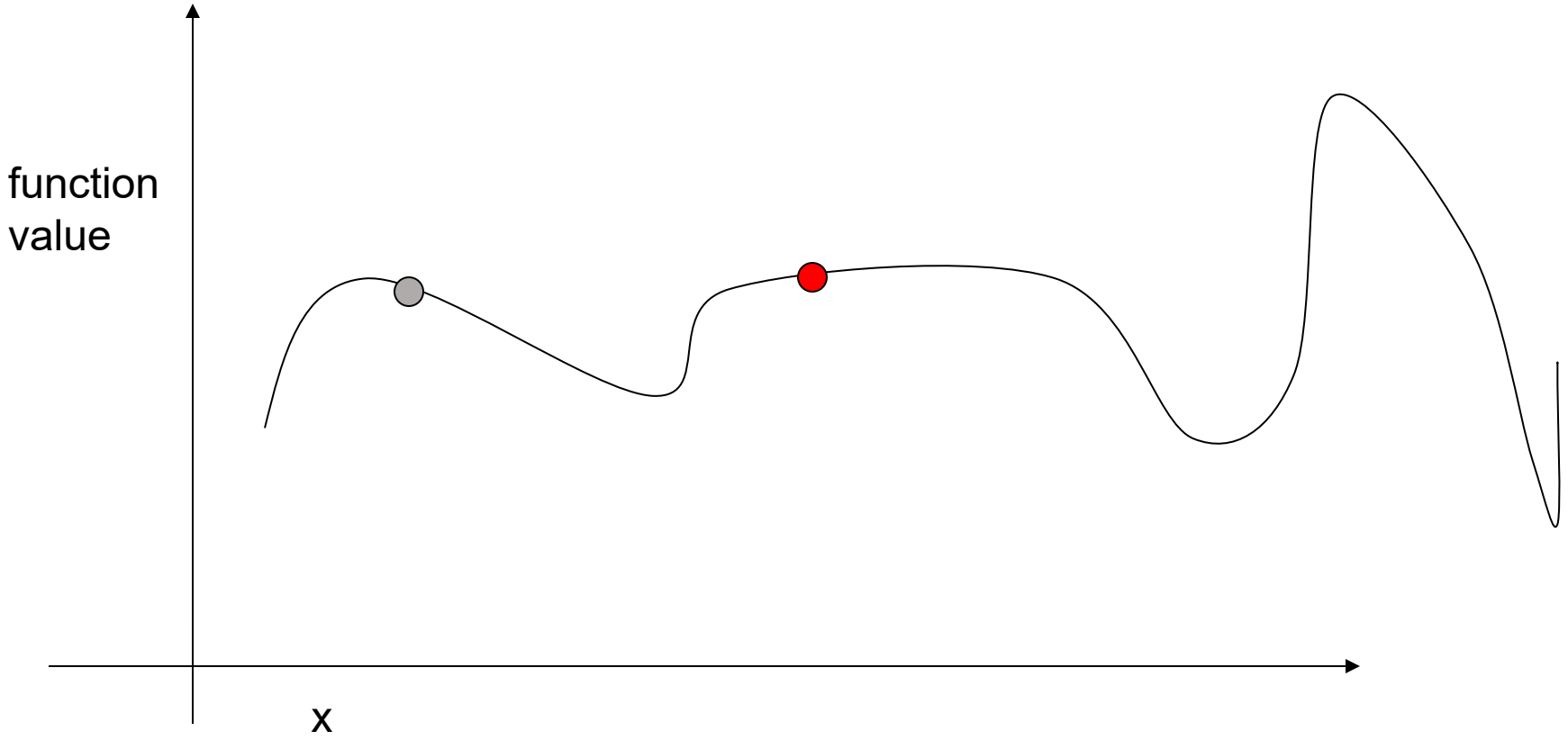
- Next Step; accept since higher E



Simulated Annealing

T = Very High

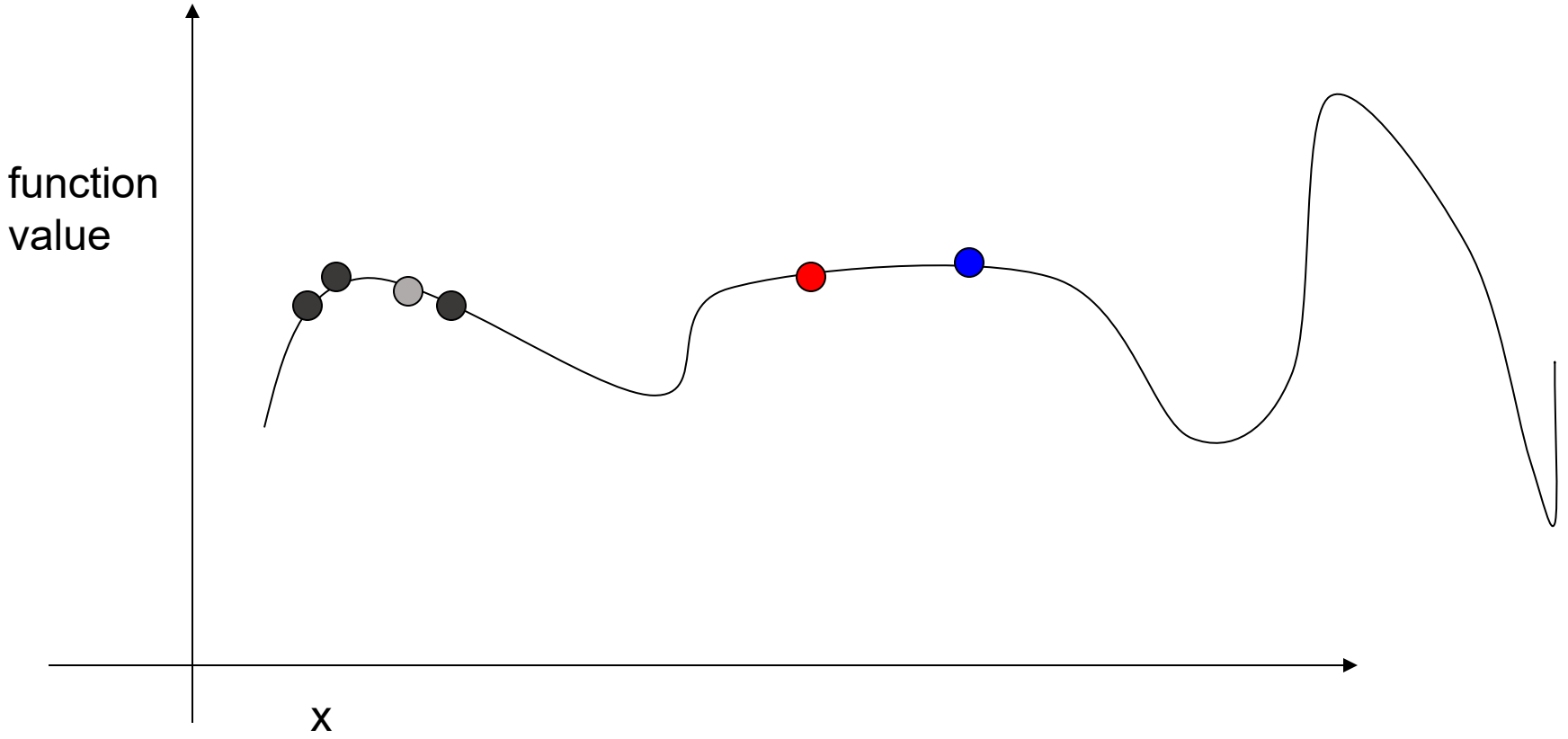
- Next Step; accept since higher E



Simulated Annealing

T = Very High

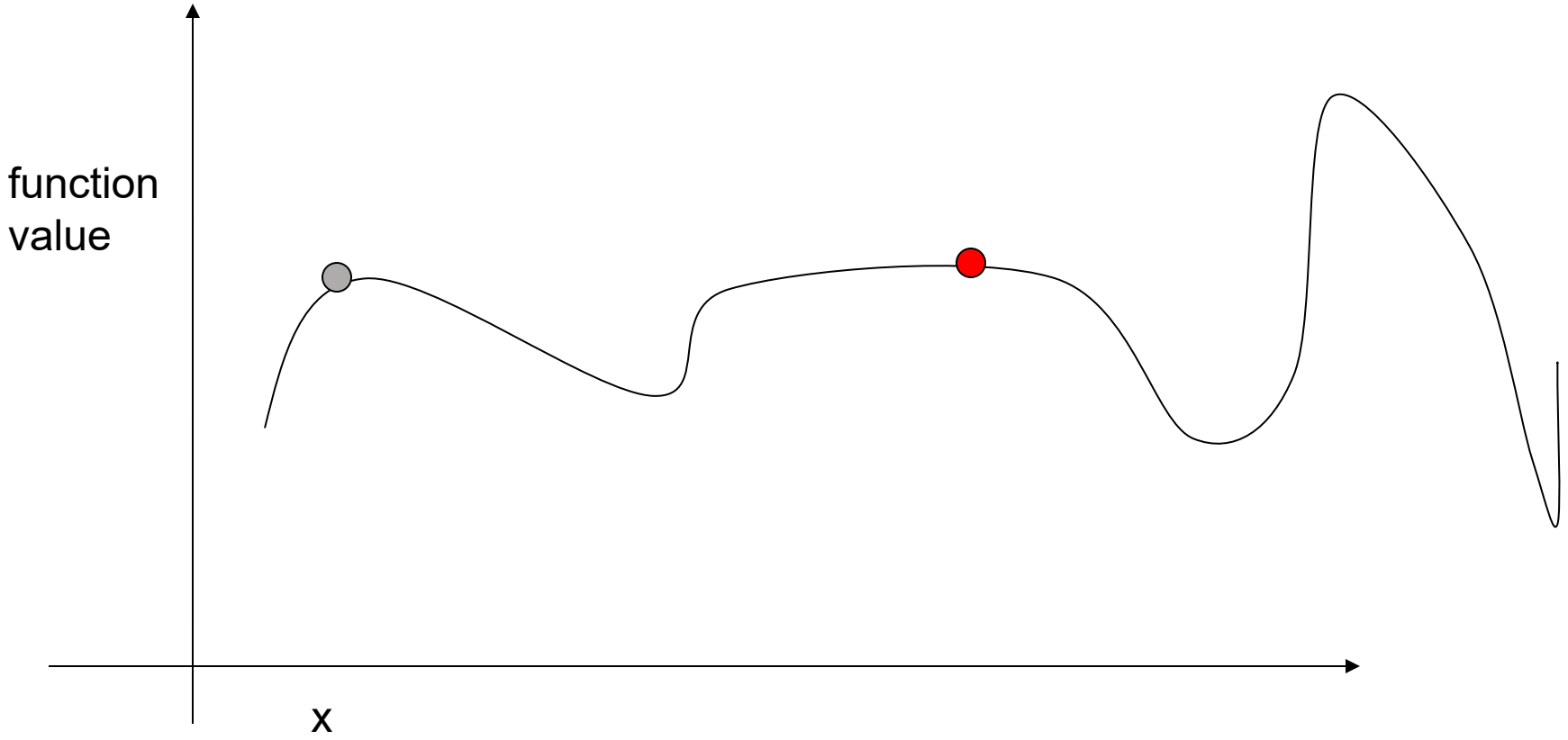
- Next Step; accept since higher E



Simulated Annealing

T = High

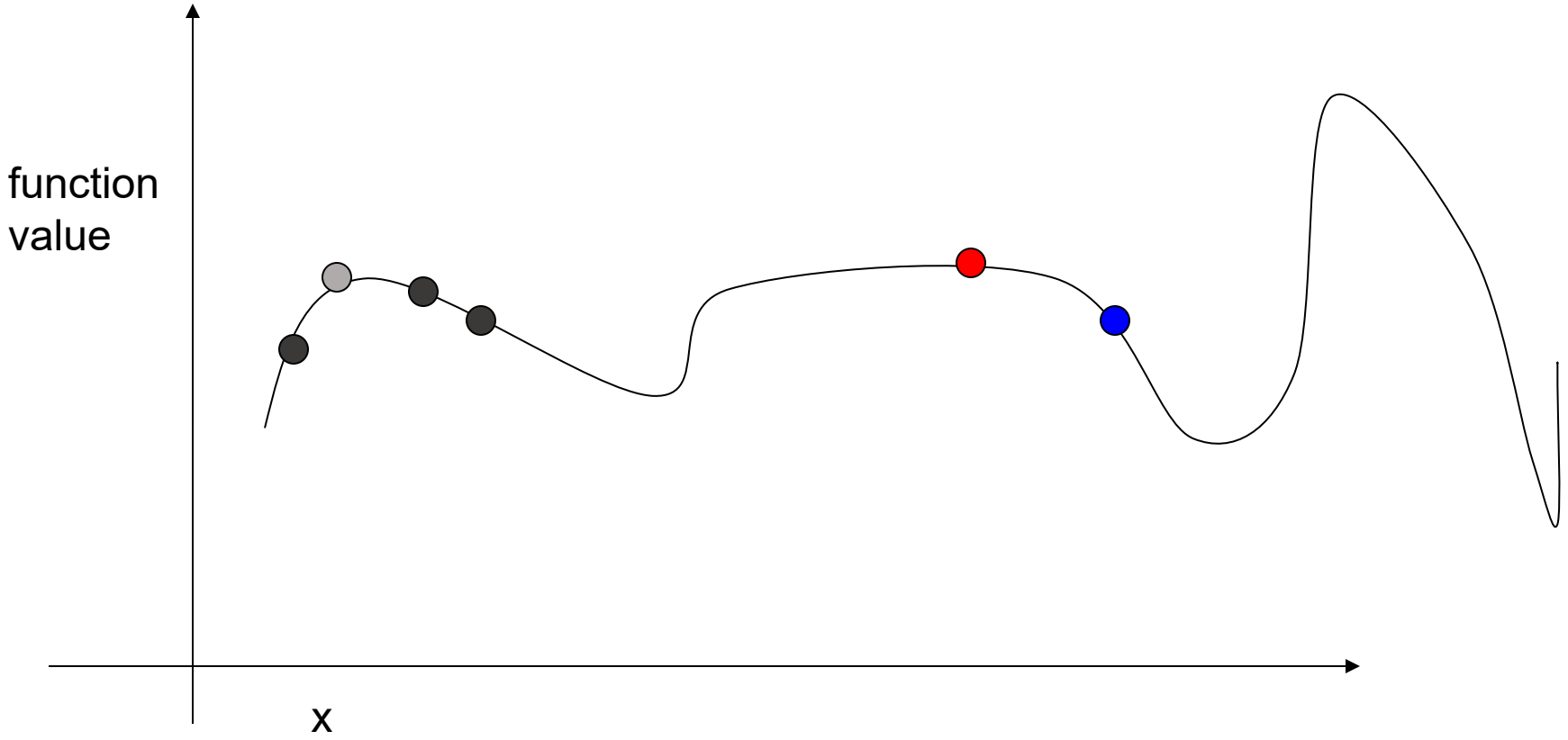
- Next Step; accept even though lower



Simulated Annealing

T = High

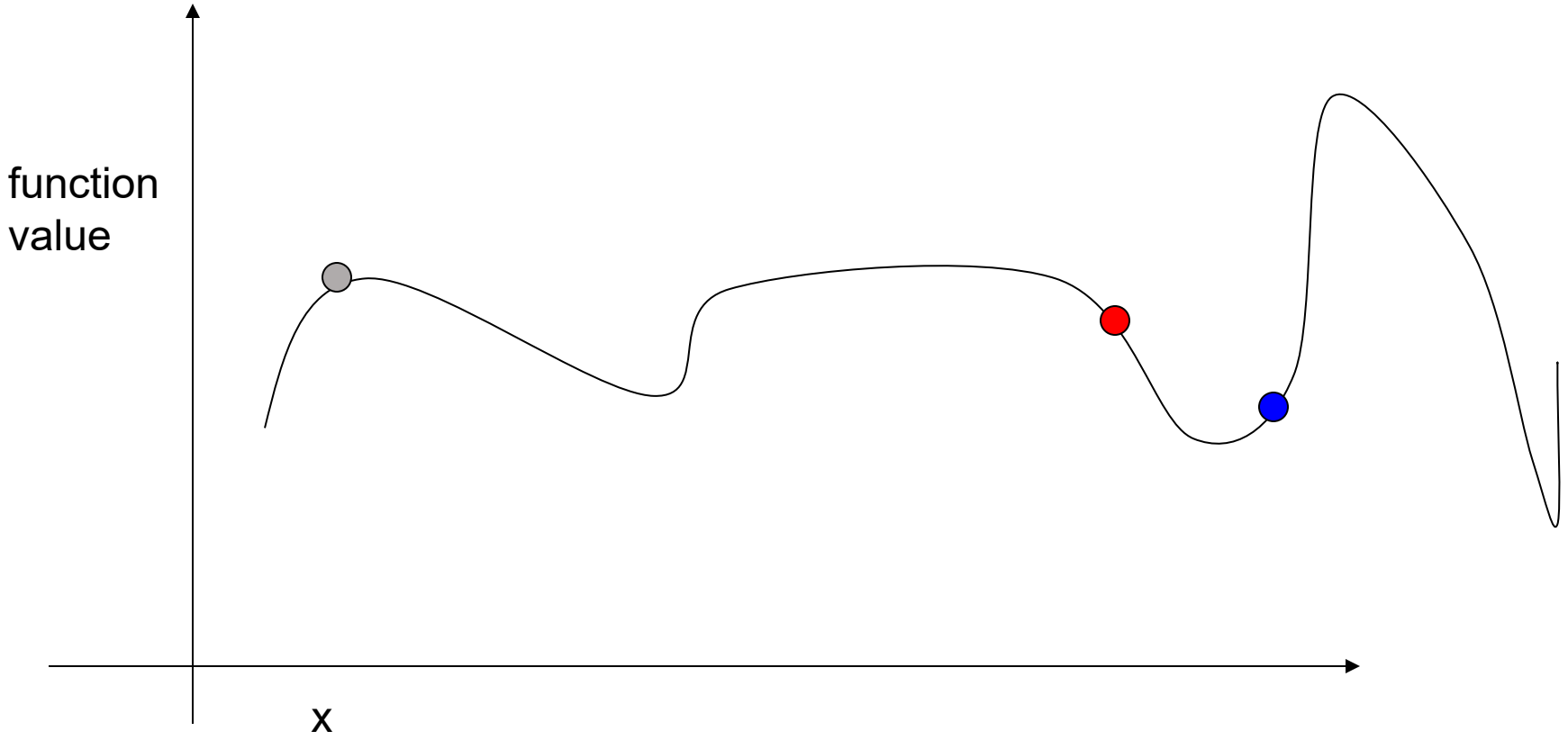
- Next Step; accept even though lower



Simulated Annealing

T = High

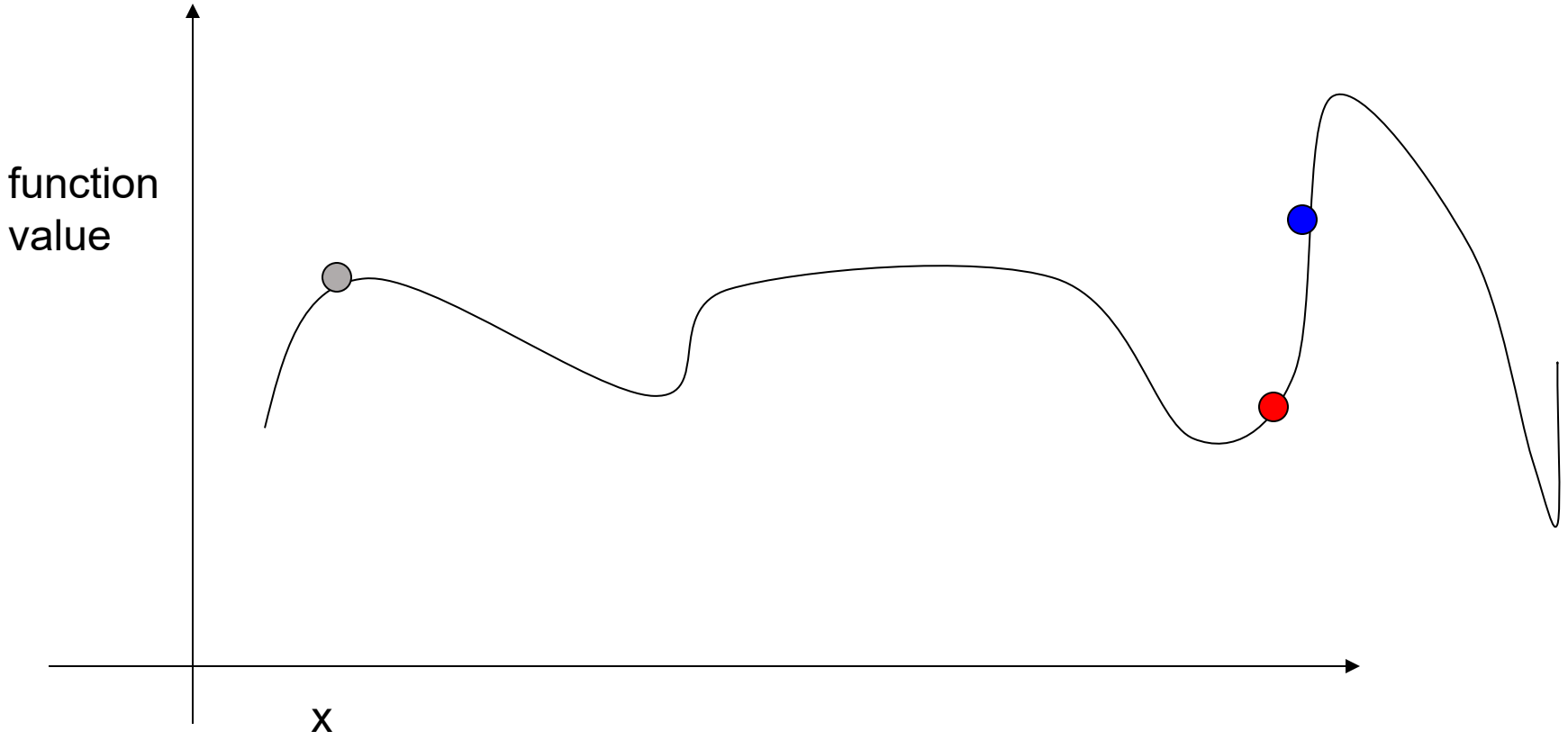
- Next Step; accept even though lower



Simulated Annealing

T = Medium

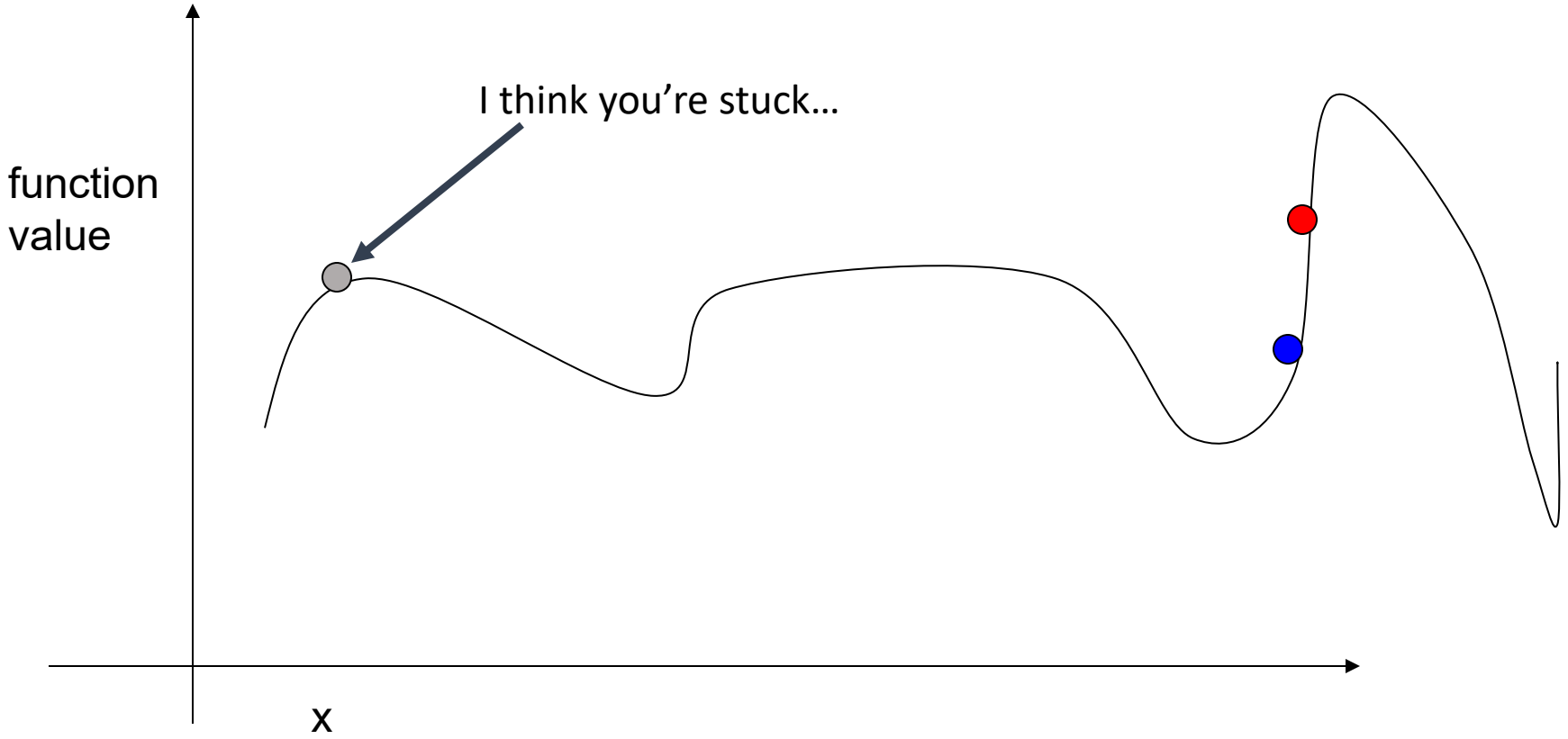
- Next Step; accept since higher



Simulated Annealing

T = Medium

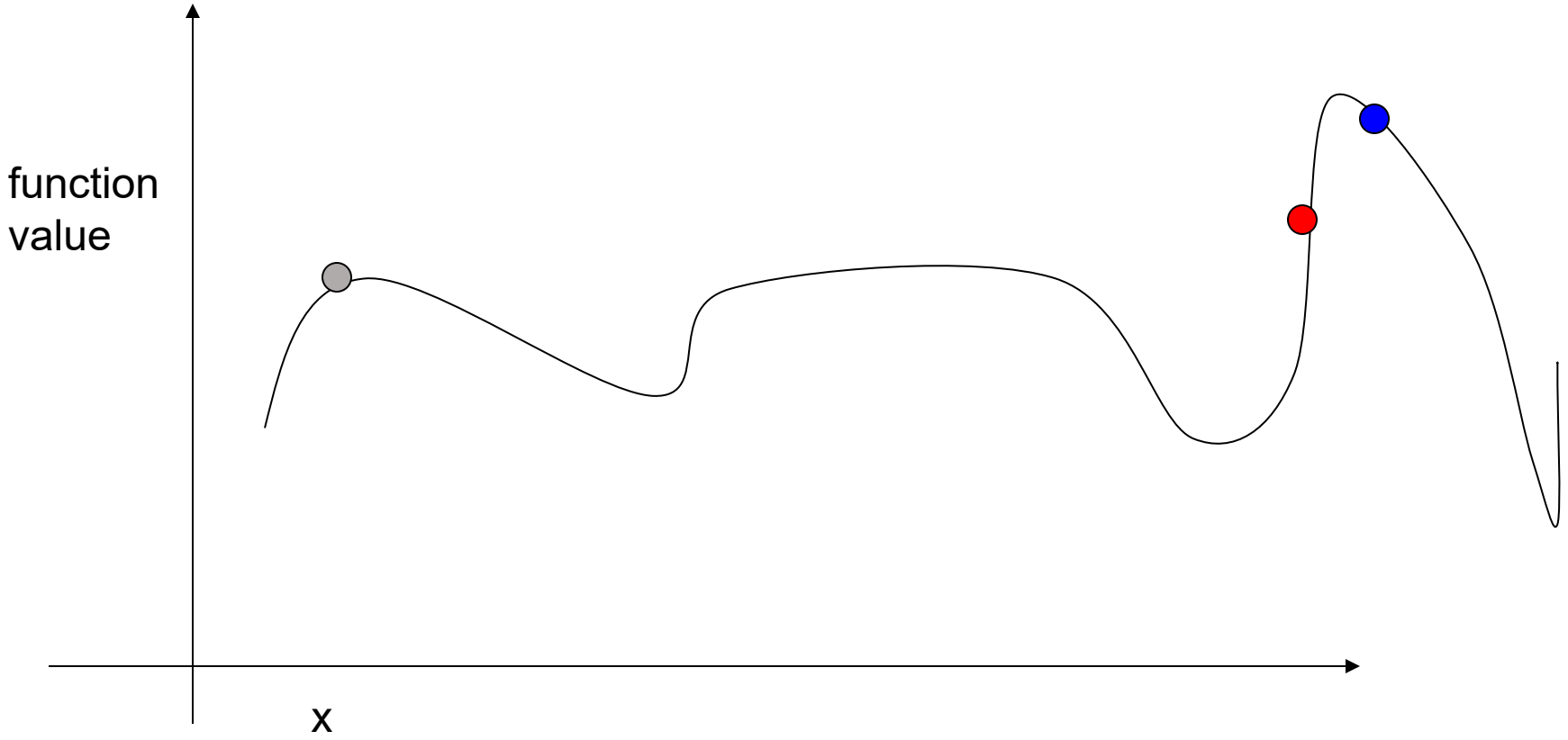
- Next Step; lower, but reject (T is falling)



Simulated Annealing

T = Medium

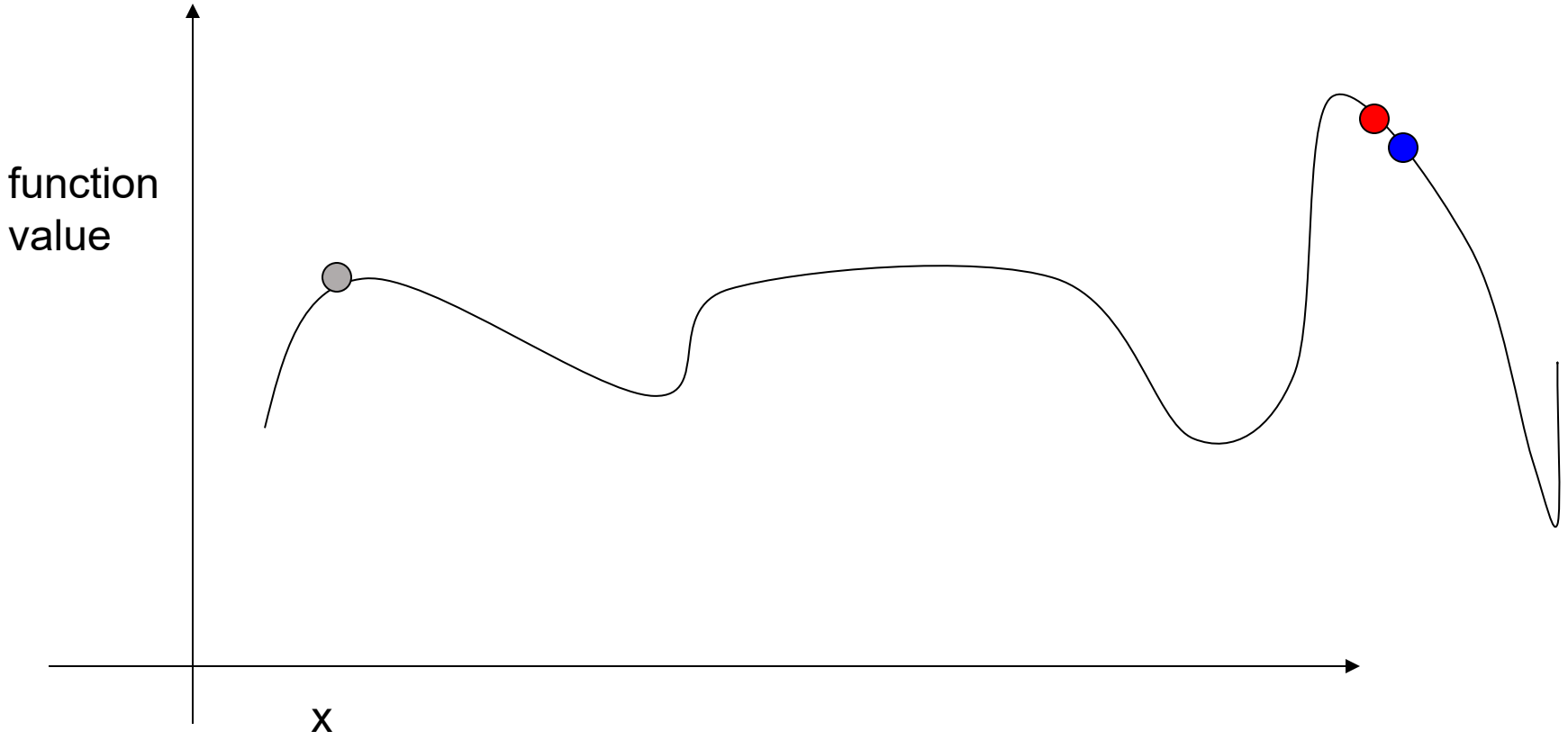
- Next Step; Accept since E is higher



Simulated Annealing

T = Low

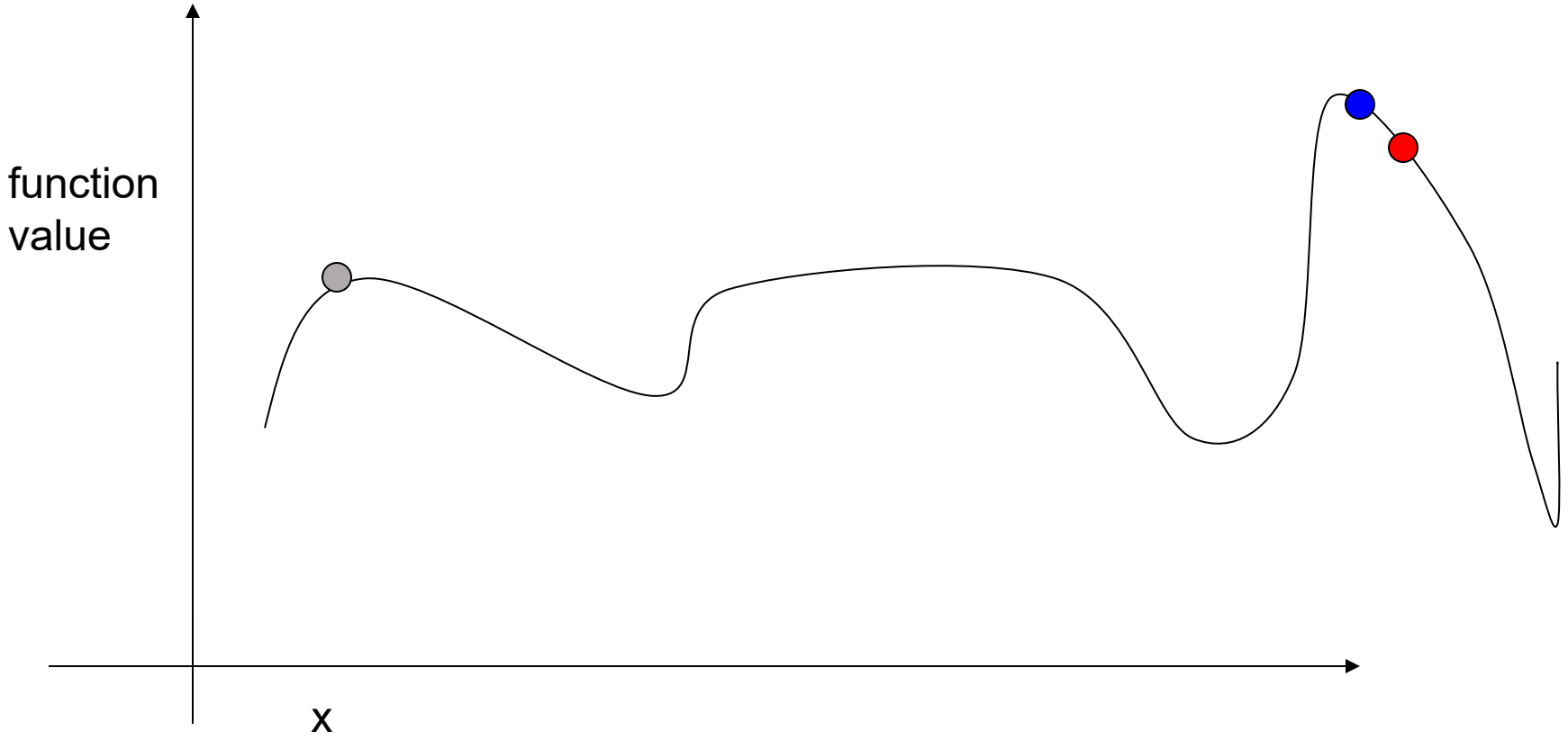
- Next Step; Accept since E change small



Simulated Annealing

T = Low

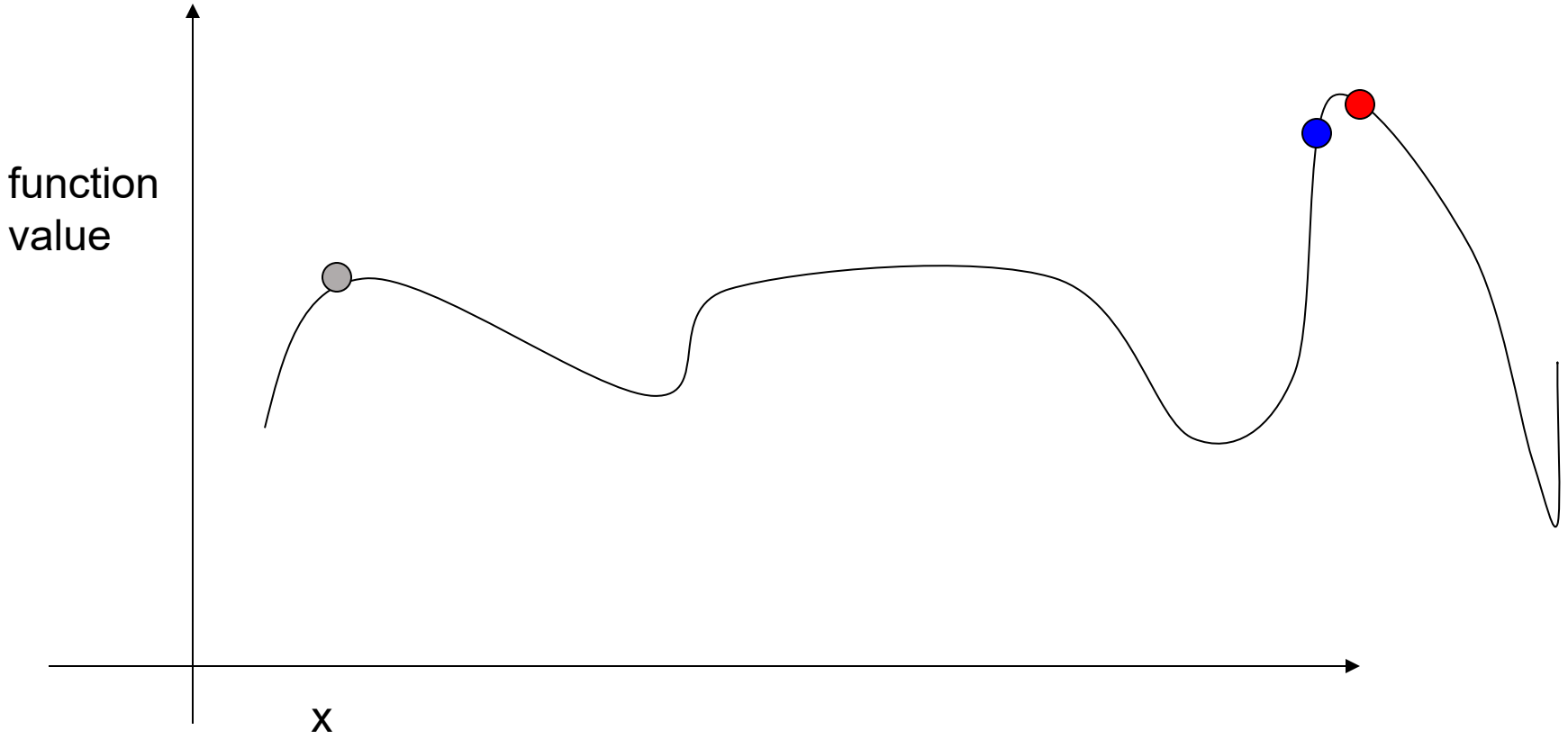
- Next Step; Accept since E larger



Simulated Annealing

T = Low

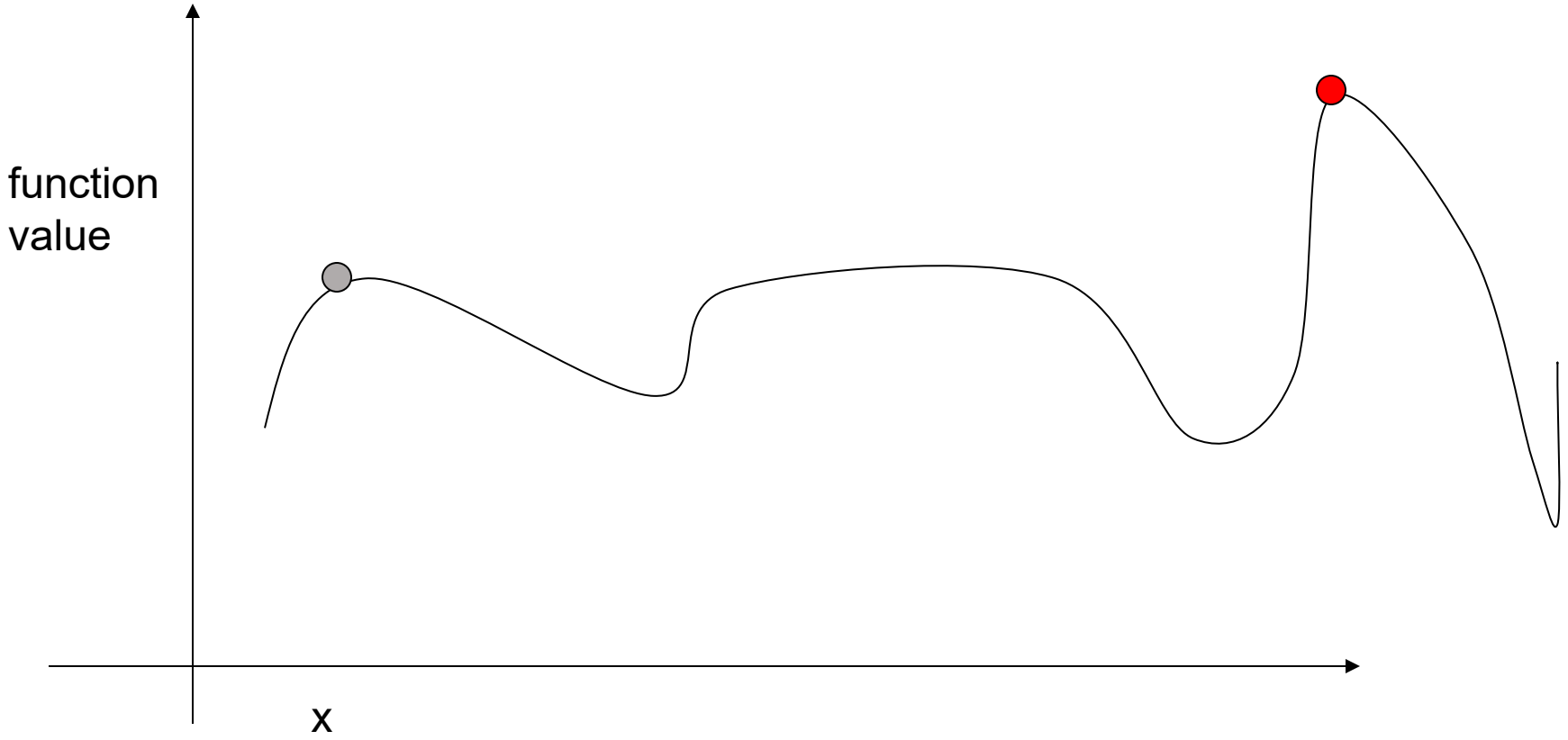
- Next Step; Reject since E lower and T low



Simulated Annealing

T = Low

- Eventually converge to maximum



Simulated Annealing Algorithm

Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, ..., x_d)^T$
Initialize the initial temperature T_0 and initial guess $\boldsymbol{x}_{(0)}$
Set the final temperature T_f and the max number of iterations N
Define the cooling schedule $T \mapsto \alpha T$, $(0 < \alpha < 1)$
while ($T > T_f$ and $t < N$)
    Drawn $\boldsymbol{\epsilon}$ from a Gaussian distribution
    Move randomly to a new location: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{\epsilon}$ (random walk)
    Calculate $\Delta f = f_{t+1}(\boldsymbol{x}_{t+1}) - f_t(\boldsymbol{x}_t)$
    Accept the new solution if better
    if not improved
        Generate a random number r
        Accept if $p = \exp[-\Delta f/T] > r$
    end if
    Update the best $\boldsymbol{x}_*$ and f_*
    $t = t + 1$
end while

Questions?