Simulate Annealing utilizes exploration and exploitation approach. Exploration prevents us from being stuck at local optima and exploitation takes us to optima in first place. The extreme version of exploration is random walk, of exploitation - hill climbing

In SA if a move is a good, the probability to move to it is higher (in original paper it is 1, however this is not required for algorithm to work). If it is a bad, the probability to move to it is lower but still existent. This probability depends on Temperature. If the Temperature is high, we are more likely to take *bad* decisions.

Let *c* be *current node*, *n* be *next node* and *eval* function be the function that evaluates the node. Then the goodness of move when maximizing can be defined by `ΔE = eval(n) - eval(c)` (flip the signs if minimizing).
In annealing we want 2 parameters: ΔE that defines how good the move *n* is and another parameter called Temperature (T) that will control how this ΔE influences the probability of taking action *n*, making it possible to choose bad moves so we do not get stuck in local optima.

$$\frac{\Delta E}{T}$$

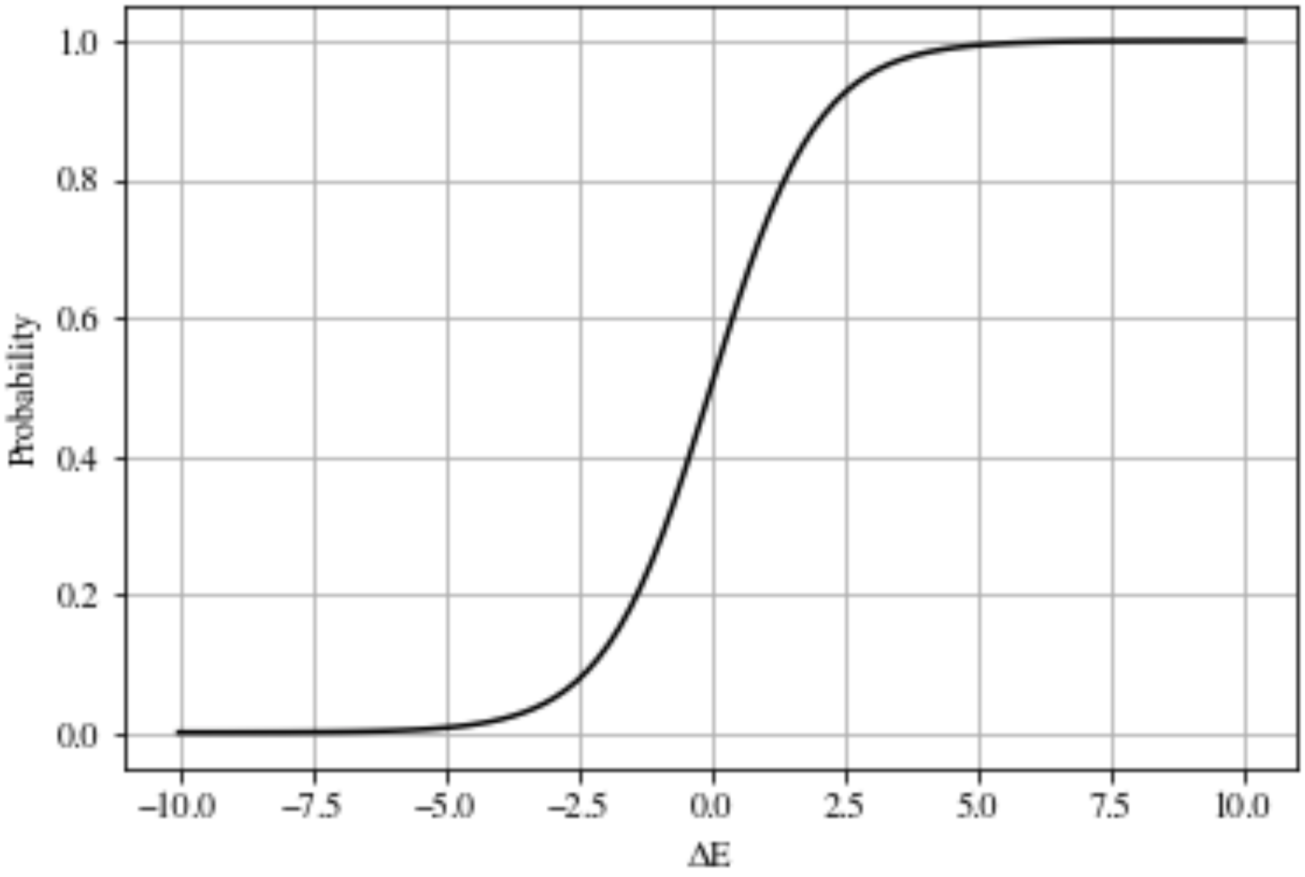
So we need a function that will assign a probability 0-1 to a real input depending on its value. Low probability to negative inputs, high probability to positive inputs. There are many functions that can do that, in this example we will use a sigmoid function. However, in original paper the function was `exp(-ΔE/T)`.

$$S(x) = \frac{1}{1 + e^{-x}}$$

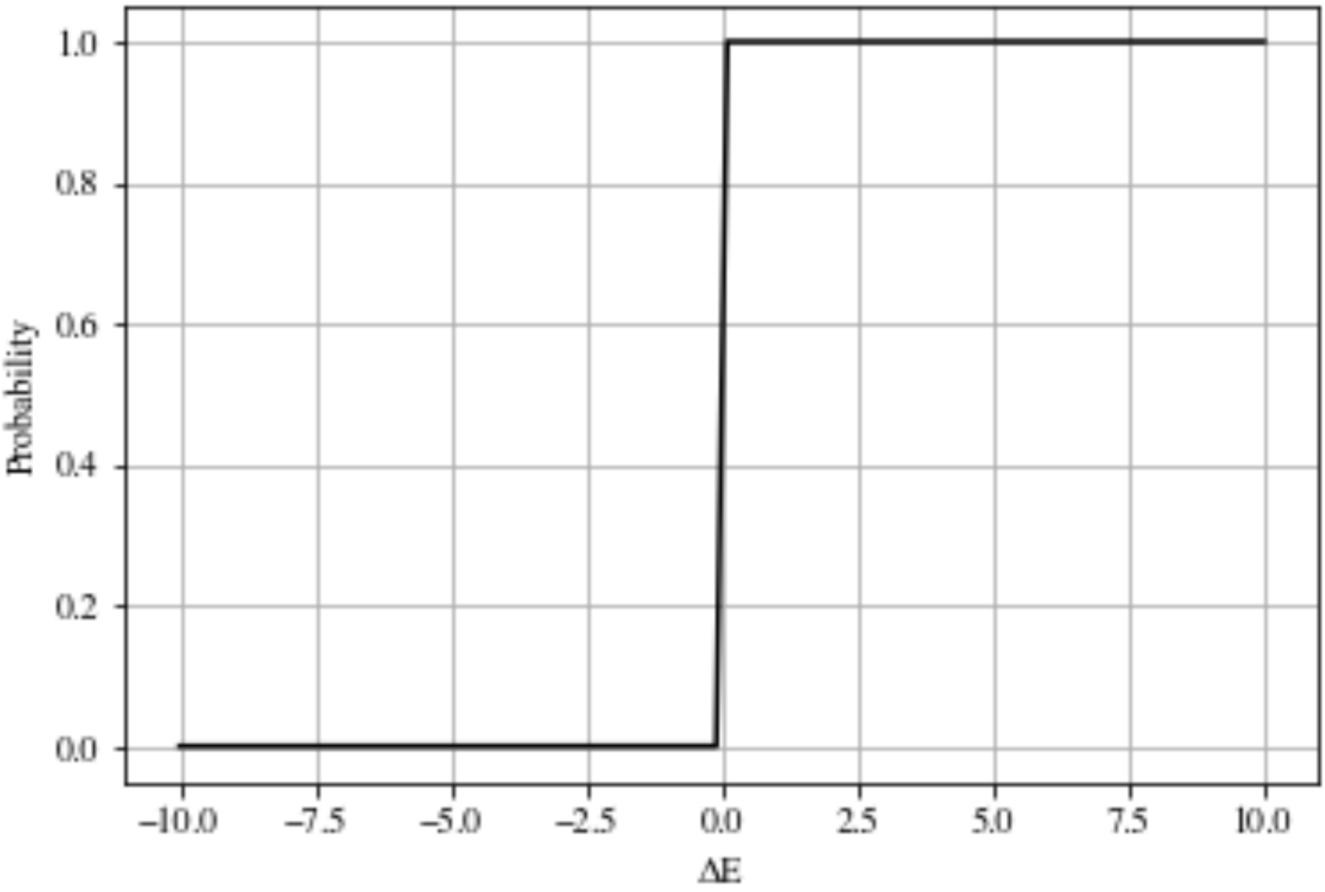
By applying the ideas discussed above, we get an *acceptance probability function* (and again, it can be different, but the idea of using ΔE/T should remain)

$$P(c, n) = \frac{1}{1 + e^{\frac{\Delta E}{T}}}$$

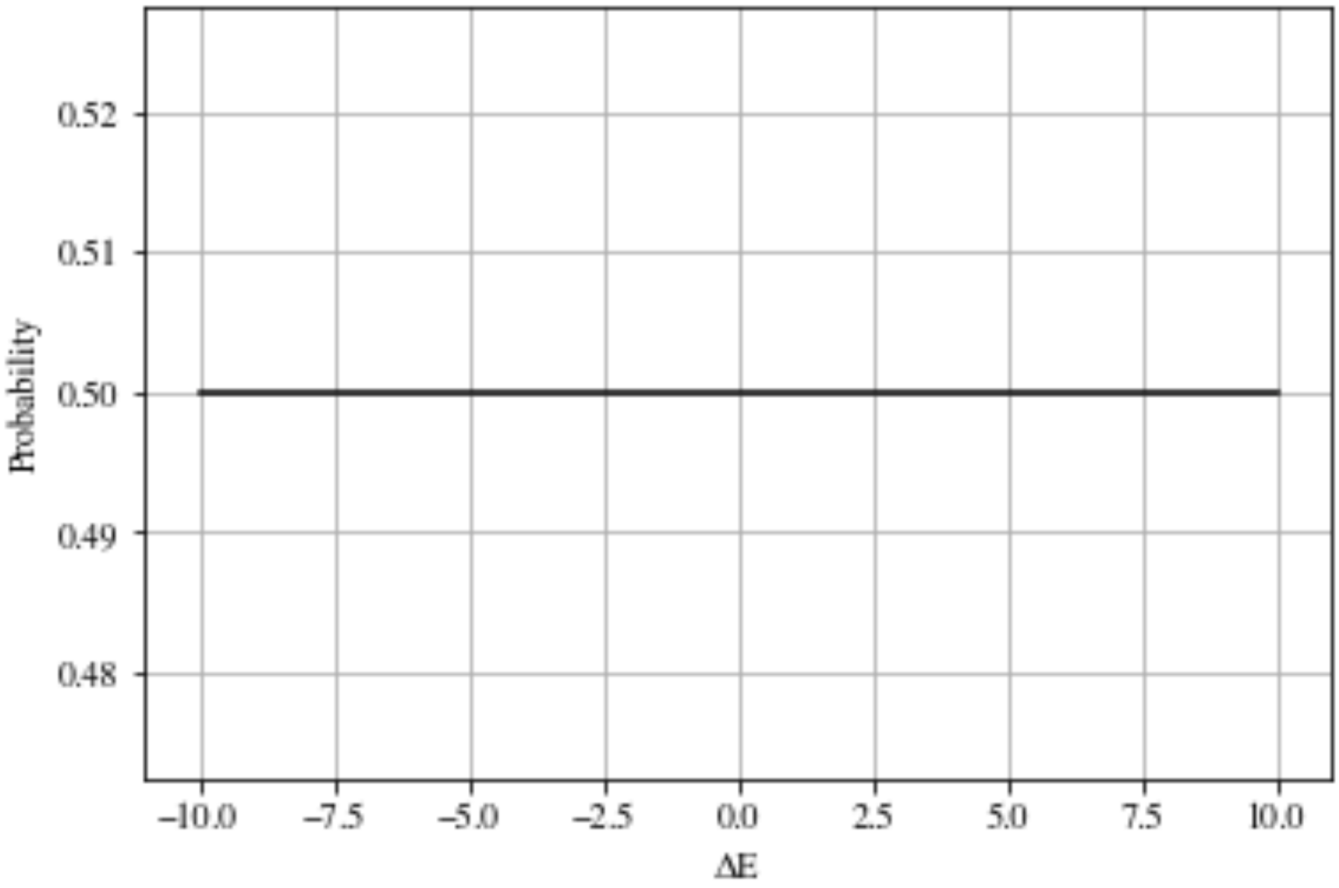
Here is what our probability curve looks like for a constant T = 1 if `-10 <= ΔE <= 10`.



For comparison, in hill climbing we make action only when `ΔE > 0` and probability curve will be

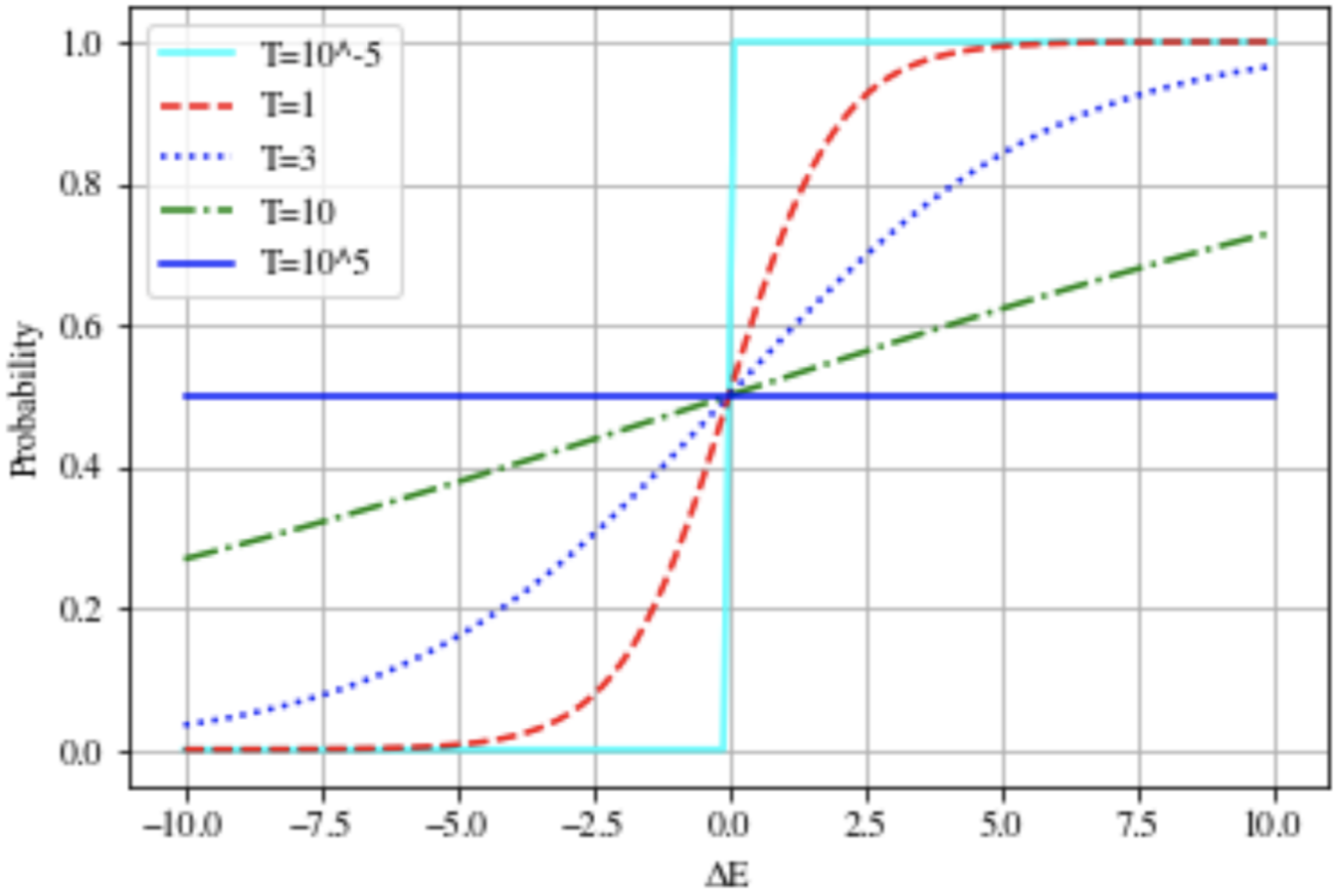


In random walk, we have the same probability when taking any action (0.5) and probability curve is



Let's play with Temperature and see how it affects the probability curve. High temperature should make our decisions worse, low temperature - better. Maybe try it with extremely low and

extremely high Temperatures?



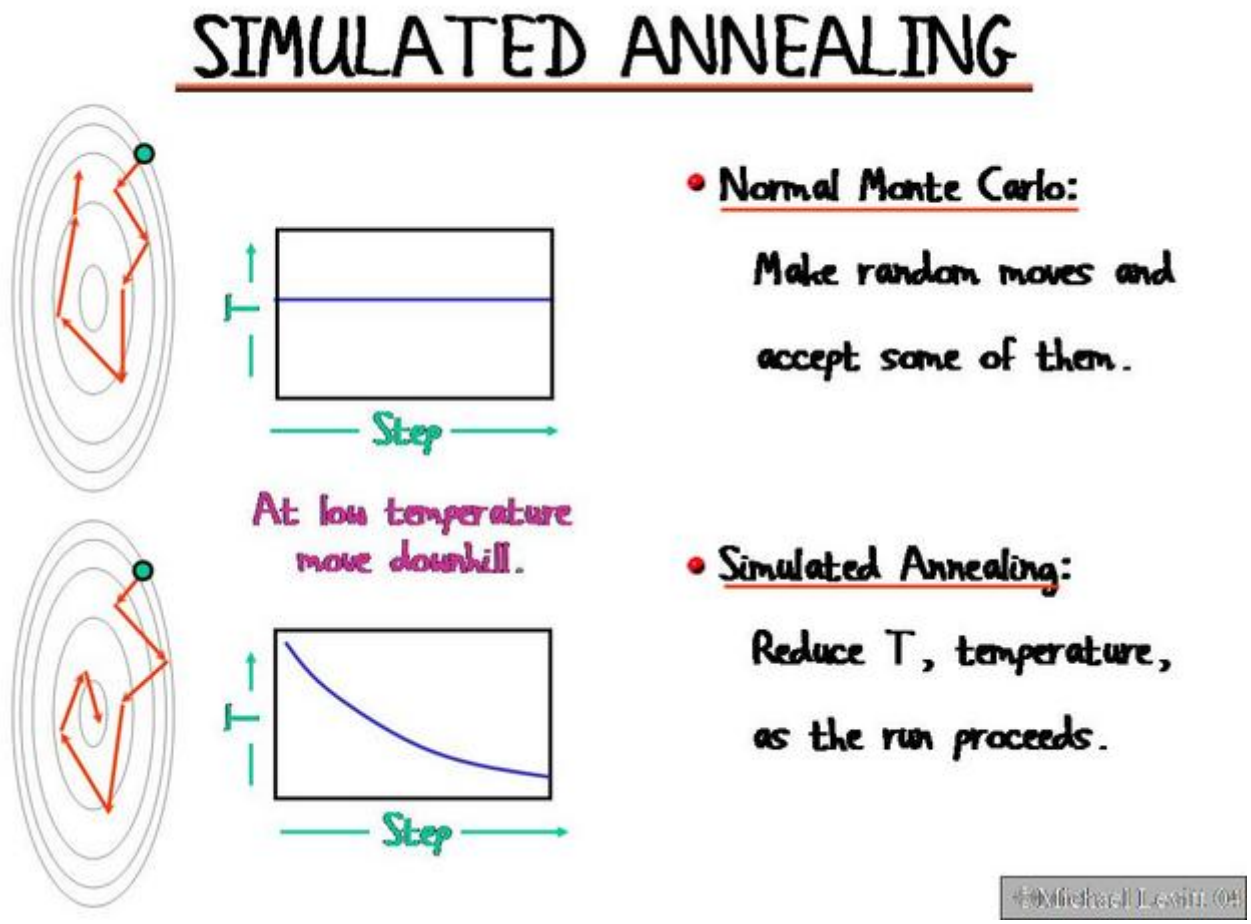
The lower the Temperature, the more the graph looks like Hill Climbing. The higher the temperature, the more the graph looks like Random Walk. So if we want the probability function to make more random choices (explore), we can increase the Temperature. If we want it to follow the gradient and find the minima/maxima, decrease the Temperature. Again, this property is nicely demonstrated when we use a sigmoid **acceptance probability function**, but might not be present if you use some other

In the original description of simulated annealing, the probability was equal to 1 when the new node had a better evaluation that the previous node, the procedure always moved downhill when it found a way to do so, irrespective of the temperature. Many descriptions and implementations of simulated annealing still take this condition as part of the method's definition. However, this condition is not essential for the method to work.

Here is the pseudocode of Simulated Annealing:

- Let $s = s_0$
- For $k = 0$ through $k_{\max}$ (exclusive):
  - $T \leftarrow$ temperature(1 - $(k+1)/k_{\max}$)
  - Pick a random neighbour, $s_{\text{new}} \leftarrow \text{neighbour}(s)$
  - If $P(E(s), E(s_{\text{new}}), T) \geq \text{random}(0, 1)$:
    - $s \leftarrow s_{\text{new}}$
- Output: the final state s

This approach is very similar to Markov Chain Monte Carlo sampling methods. We use only knowledge about the previous node to choose a new node, which is similar to Markov Chains (although, SA is not time-homogenous as we have T). Also, we decide whether to take a proposed move or not based on a random number, which is similar to Monte Carlo.



And this is because SA was inspired by Metropolis random walk algorithm, which is an MCMC sampling method (in fact it was the first MCMC algorithm that was ever made). An extended version of it is widely known today. It is called Metropolis-Hastings algorithm, which unlike the original Metropolis random walk that could only sample well from symmetrical proposal distributions, can work with general case.
So in both SA and M-H we explore a surface stochastically, using the Metropolis acceptance criterion for proposed points.

Simulated annealing was also inspired by real annealing process (materials science). In annealing, atoms migrate in the crystal lattice and the number of dislocations decreases, leading to a change in ductility and hardness. By cooling the material in a controlled way, we can achieve the properties that we want, which is like a physical optimization process (cool slowly → less dislocations in crystal lattice → material is hard and elastic).

References

Optimization - I (Simulated Annealing)
(ML 18.1) Markov chain Monte Carlo (MCMC) introduction
A Short History of Markov Chain Monte Carlo; Subjective Recollections from Incomplete Data
Markov Chains Clearly Explained! Part - 1
An introduction to the Random Walk Metropolis algorithm
Optimization - I (Simulated Annealing)
Optimization by Simulated Annealing
Diser/Materials/Simulated Annealing
The Metropolis–Hastings algorithm
What is the difference between Simulated Annealing and Monte-Carlo Simulations?
What is the relationship between Metropolis Hastings and Simulated Annealing?
Markov Chain Monte Carlo & Simulated Annealing