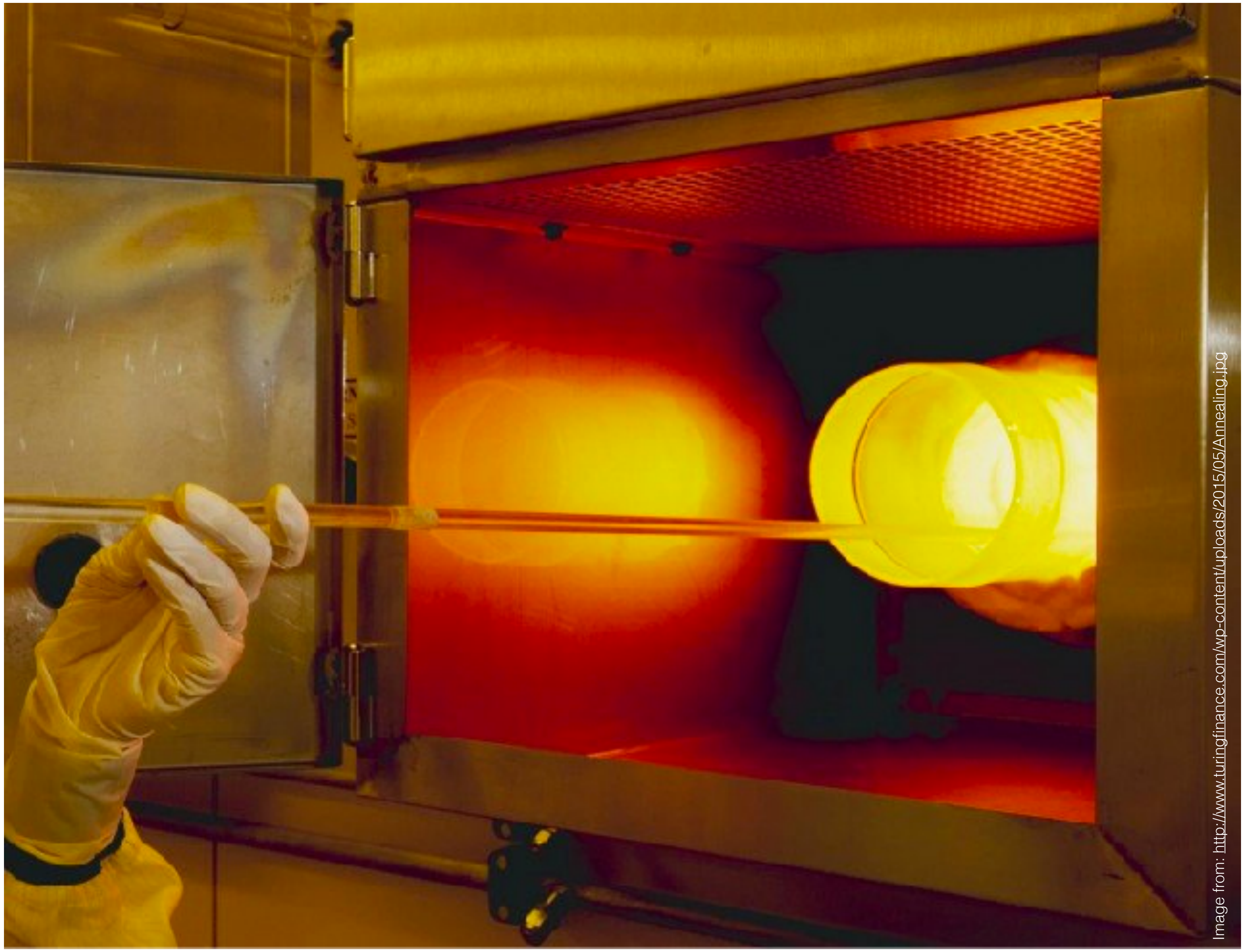


Image from: http://www.turingfinance.com/wp-content/uploads/2015/05/Annealing.jpg

Simulated Annealing — Part 2

Leandro L. Minku

Simulated Annealing

Simulated Annealing (assuming maximisation)

1. current_solution = generate initial solution randomly

2. Repeat:

    2.1 rand_neighbour = generate random neighbour of current_solution

    2.2 If quality(rand_neighbour) <= quality(current_solution) {

        2.2.1 With some probability,
        current_solution = rand_neighbour

    } Else current_solution = rand_neighbour

    2.3 Reduce probability

  Until a maximum number of iterations

Metallurgy Annealing

- A blacksmith heats the metal to a very high temperature.
- When heated, the steel's atoms can move fast and randomly.

Image from: http://www.stormthecastle.com/indeximages/sting-steel-thumb.jpg

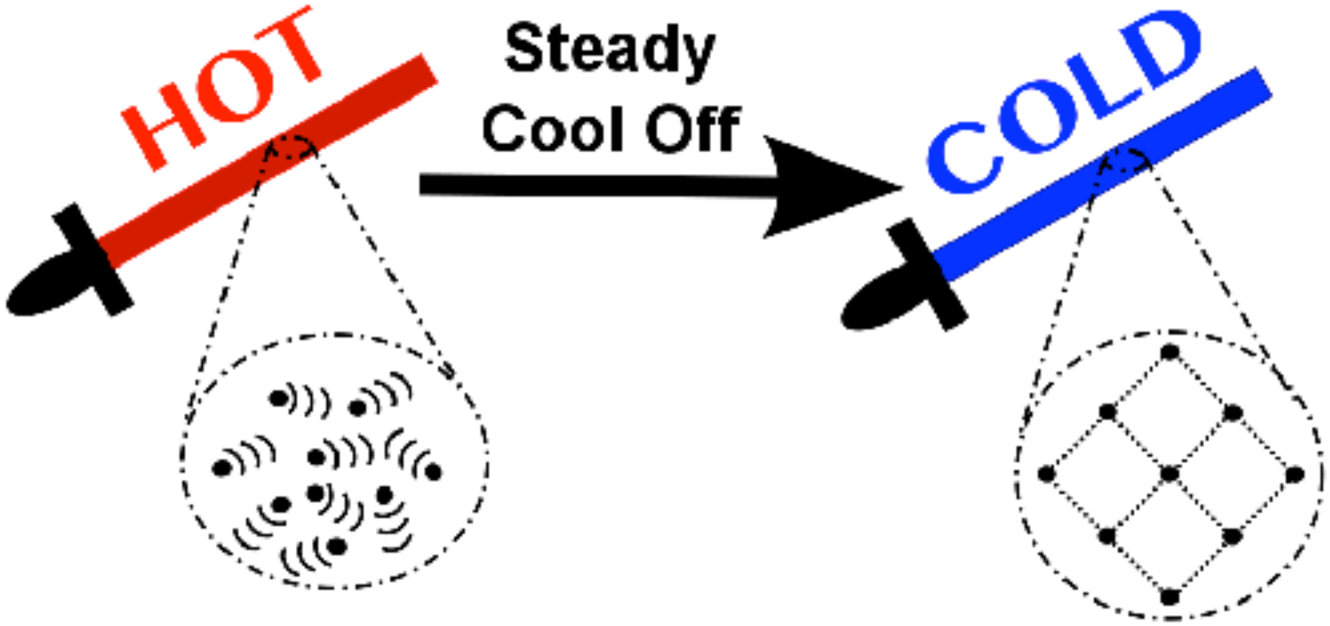


Image from: http://2.bp.blogspot.com/--kOlrodykkg/UbfVZ0_l5HI/AAAAAAAAAJ4/0rQ98g6tDDA/s1600/annealingAtoms.png

- The blacksmith then lets it cool down slowly.
- If cooled down at the right speed, the atoms will settle in nicely.
- This makes the sword stronger than the untreated steel.

Probability Function

Probability of accepting a solution of equal or worse quality, inspired by thermodynamics:

$$e^{\Delta E/T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

e = 2.71828…

Exponential Function

(<=0)

$e^{\Delta E/T}$

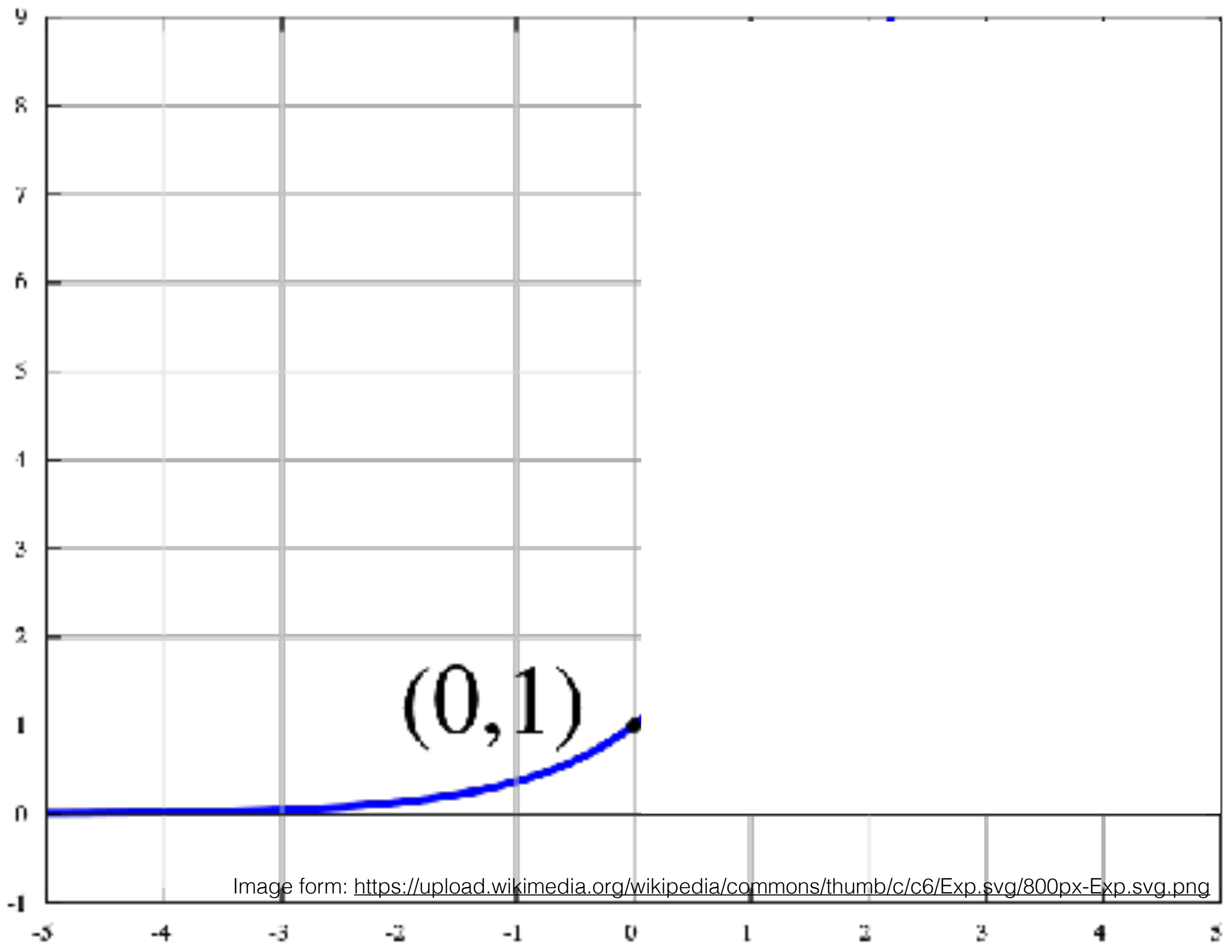


Image form: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Exp.svg/800px-Exp.svg.png

$\Delta E/T$

Exponential Function

$e^{\Delta E/T}$

$e^{\Delta E/T}$

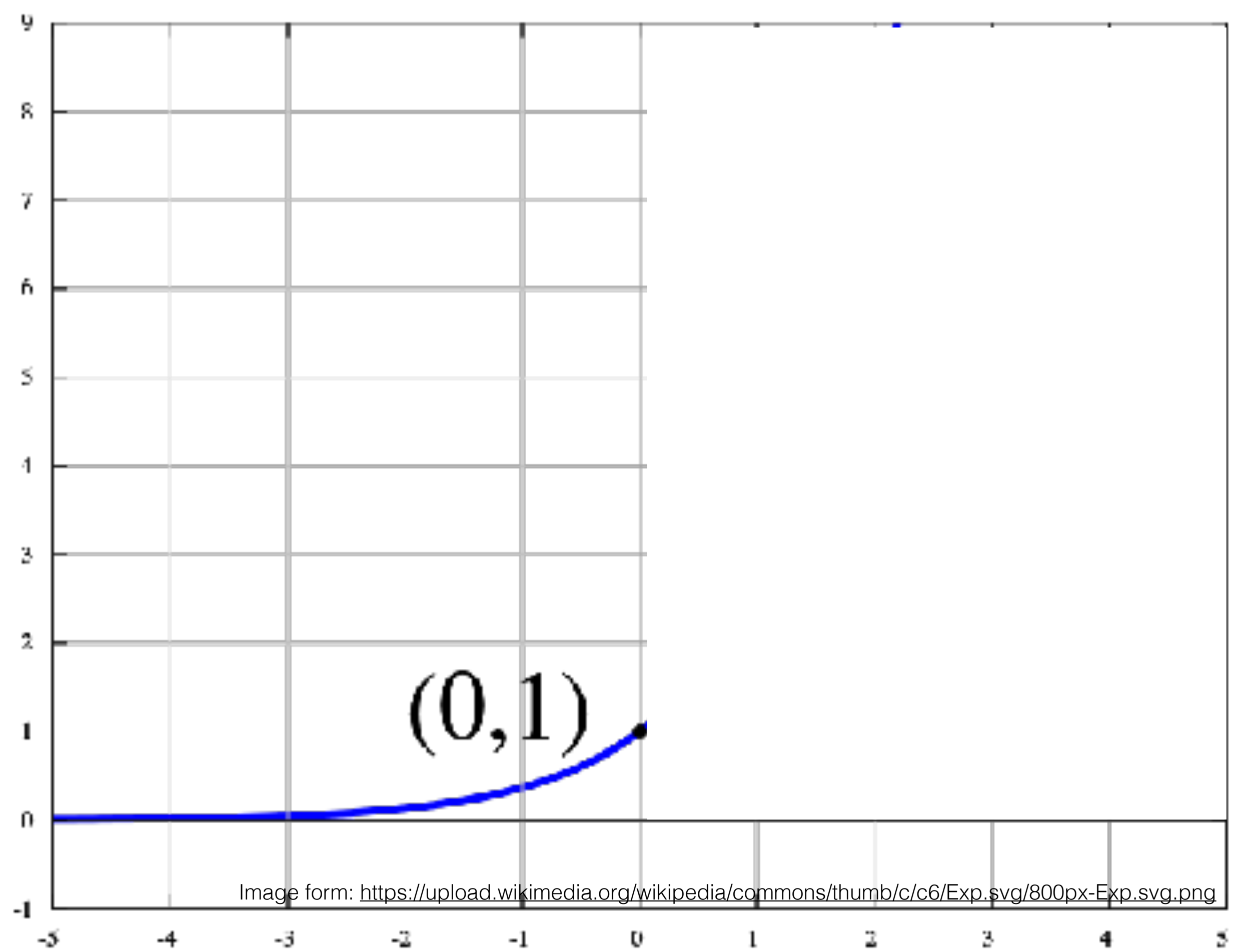


Image form: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Exp.svg/800px-Exp.svg.png

$\Delta E/T$

Exponential Function

$e^{\Delta E/T}$

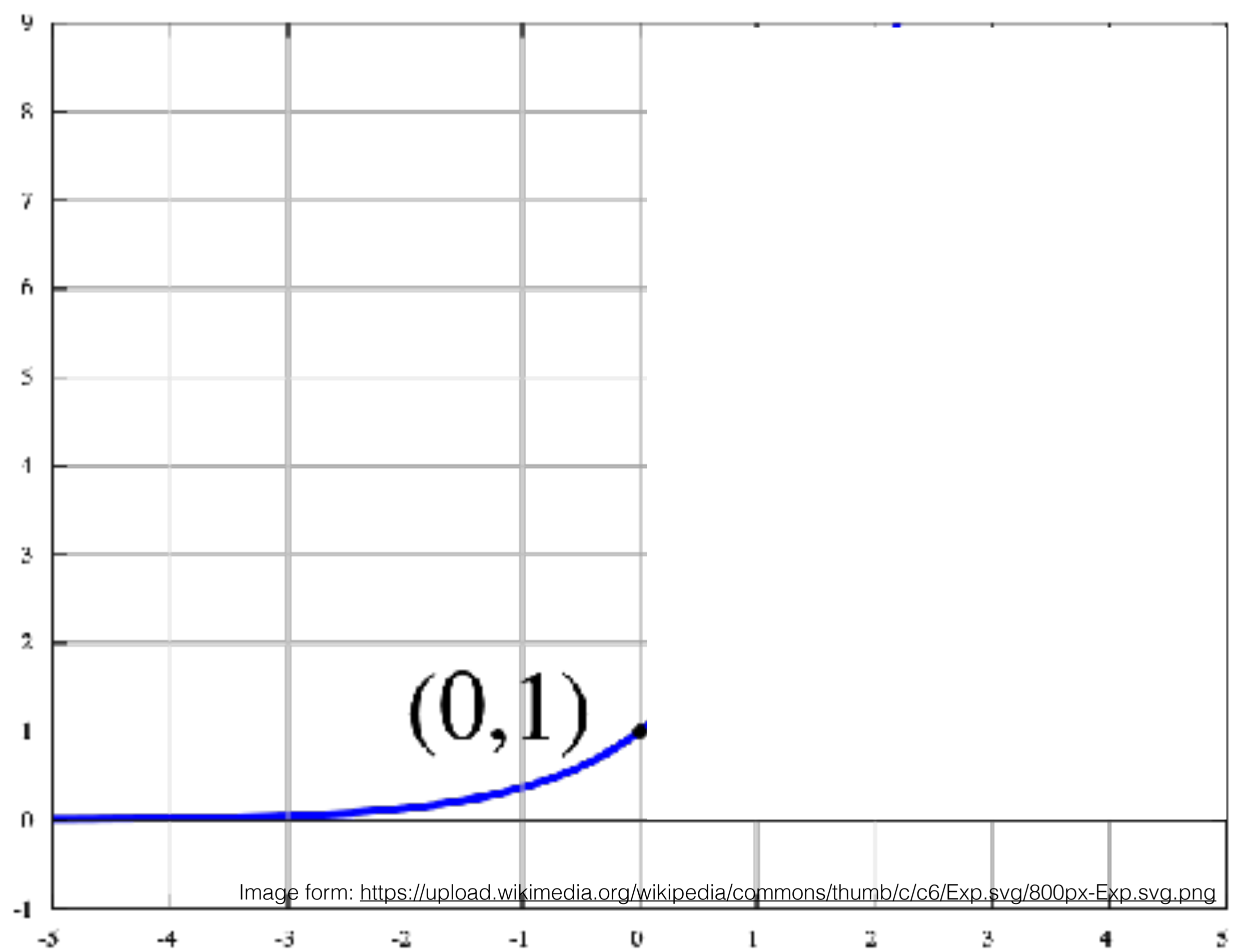


Image form: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Exp.svg/800px-Exp.svg.png

$\Delta E/T$

$e^{\Delta E/T}$

But never reaches zero

How Does ΔE Affect the Probability?

Probability of accepting a solution of equal or worse quality:

$$e^{\Delta E / T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)

(<=0)

Assuming maximisation…

T = temperature

(>0)

The worse the neighbour is in comparison to the current solution, the less likely to accept it.

How Does ΔE Affect the Probability?

Probability of accepting a solution of equal or worse quality:

$$e^{\Delta E / T}$$

But never reaches zero

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

We always have some probability to accept a bad neighbour, no matter how bad it is.

How Does ΔE Affect the Probability?

Probability of accepting a solution of equal or worse quality:

$$e^{\Delta E / T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

The better the neighbour is, the more likely to accept it.

How Should the Probability be Set?

- **Probability to accept solutions with much worse quality should be lower.**
    - **We don’t want to be dislodged from the optimum.**

- High probability in the beginning.
    - More similar effect to random search.
    - Allows us to explore the search space.

- Lower probability as time goes by.
    - More similar effect to hill-climbing.
    - Allows us to exploit a hill.

How Does T Affect the Probability?

Probability of accepting a solution of **equal or worse quality**:

$<=0$

$$e^{\Delta E/T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

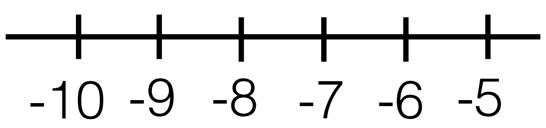
Assuming maximisation…

T = temperature
(>0)

How Does T Affect the Probability?

Probability of accepting a solution of equal or worse quality:

$$e^{\overbrace{\Delta E}^{<=0}/T}$$



ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

If T is higher, the probability of accepting the neighbour is higher.

How Does T Affect the Probability?

Probability of accepting a solution of equal or worse quality:

<=0

$$e^{\Delta E / T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

If T is lower, the probability of accepting the neighbour is lower.

How Does T Affect the Probability?

Probability of accepting a solution of equal or worse quality:

$$e^{\overset{<=0}{\Delta E}/T}$$

ΔE = quality(rand_neighbour) - quality(current_solution)
(<=0)

Assuming maximisation…

T = temperature
(>0)

So, reducing the temperature over time would
reduce the probability of accepting the neighbour.

How Should the Temperature be Set?

- High probability in the beginning.
  - More similar effect to random search.
  - Allows us to explore the search space.

  T should start high.

- Lower probability as time goes by.
  - More similar effect to hill-climbing.
  - Allows us to exploit a hill.

  T should reduce slowly over time.

Image from: http://static.comicvine.com/uploads/original/13/130470/2931473-151295.jpg

How to Set and Reduce T?

- T starts with an initially high pre-defined value (parameter of the algorithm).
- There are different update rules (schedules)…
- Update rule:
  - $T = \alpha T$,
    α is close to, but smaller than, 1
    e.g., $\alpha = 0.95$

Simulated Annealing

Simulated Annealing (assuming maximisation)

Input: initial temperature Ti

1. current_solution = generate initial solution randomly
2. **T = Ti**
3. Repeat:

    3.1 rand_neighbour = generate random neighbour of current_solution

    3.2 If quality(rand_neighbour) <= quality(current_solution) {

        3.2.1 With probability $e^{\Delta E/T}$,

            current_solution = rand_neighbour

    } Else current_solution = rand_neighbour

    3.3 **T = schedule(T)**

Until a maximum number of iterations

Simulated Annealing

Simulated Annealing (assuming maximisation)

Input: initial temperature Ti, minimum temperature Tf

1. current_solution = generate initial solution randomly
2. **T = Ti**
3. Repeat:

 3.1 rand_neighbour = generate random neighbour of current_solution

 3.2 If quality(rand_neighbour) <= quality(current_solution) {

  3.2.1 With probability $e^{\Delta E/T}$,

   current_solution = rand_neighbour

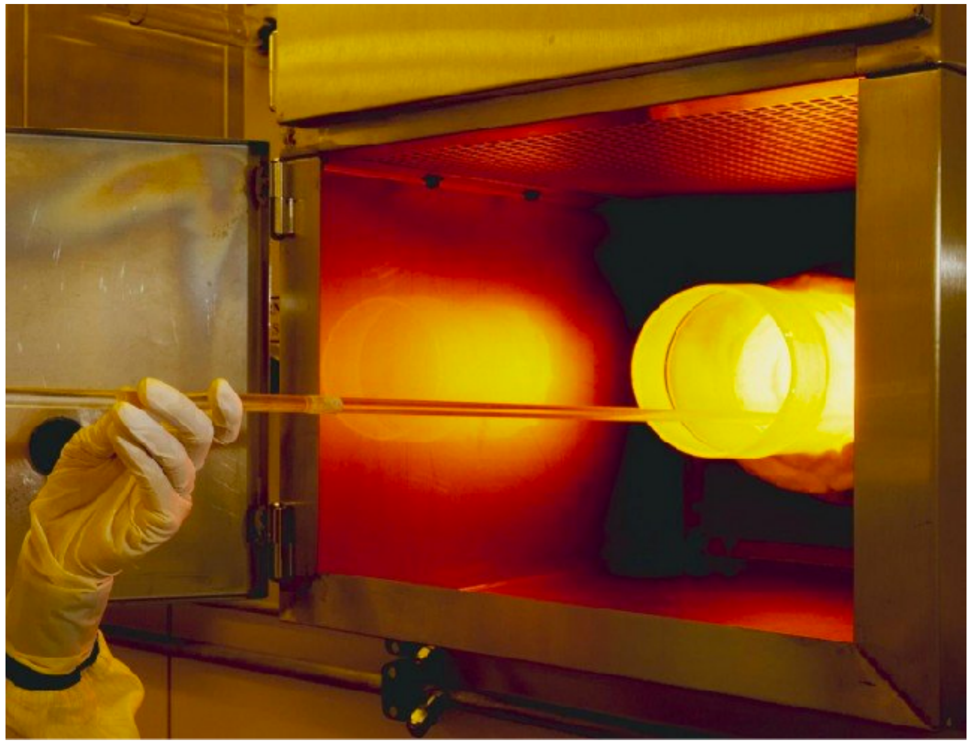
 } Else current_solution = rand_neighbour

 3.3 **T = schedule(T)**

until a minimum temperature Tf is reached or
until the current solution "stops changing"

Local Search

- Simulated annealing can also be considered as a local search, as it allows to move only to neighbour solutions.
- However, it has mechanisms to try to escape from local optima.



Optimality

Is simulated annealing guaranteed to find the optimum?

- Simulated annealing is not guaranteed to find the optimum in a reasonable amount of time.
- Whether or not it will find the optimum depends on the termination criteria and the schedule.
- If we leave simulated annealing to run indefinitely, it is guaranteed to find an optimal solution, depending on the schedule used.
- However the time required for that can be prohibitive — even more than the time to enumerate all possible solutions using brute force.
- Therefore, the advantage of simulated annealing is that it can frequently obtain good (near optimal) solutions, by escaping from several poor local optima in a reasonable amount of time.

Time and Space Complexity

- Time complexity:
  - We will run more or less iterations depending on the schedule and minimum temperature / termination criterion.
  - It is possible to compute the time complexity to reach the optimal solution, but it varies depending on the problem and may be even worse than the brute force time complexity, as mentioned in the previous slide.
- Space complexity:
  - Depends on how the design variable is represented in the algorithm.

Summary

- The probability of accepting neighbouring solutions of equal or worse quality than the current solution is inspired by metallurgy annealing.
- A “temperature” is used to control how low the probability is.
- A schedule is used to reduce the “temperature” over time.
- The worse a neighbour is, the lower the chances of accepting it.

Next

- Dealing with constraints.