ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Artificial Intelligence course

6th Semester

Bachelor in Informatics and Computer Engineering

Simulated Annealing

Nuno Leite

12 May 2022

Copyright note

- These slides were partially based on the book:
  El-Ghazali Talbi, *Metaheuristics – From Design to Implementation*, 2009, Wiley

Simulated annealing optimization process

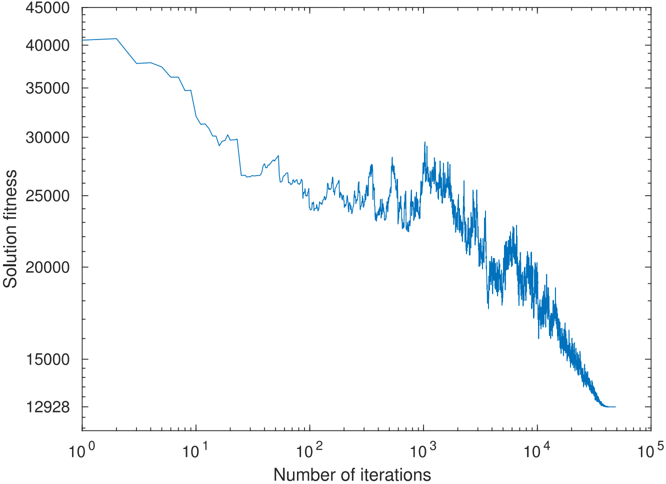


Fig.3. Solution fitness evolution for ITC2007's Instance 4 when applying simulated annealing.

Simulated annealing (SA)

- At the beginning, SA accepts worse and better solutions because the temperature is high
- At the end of the optimization process, the temperature is very low and SA only accepts better solutions; worse solutions are discarded
    - It behaves like Hill-climbing

SA search

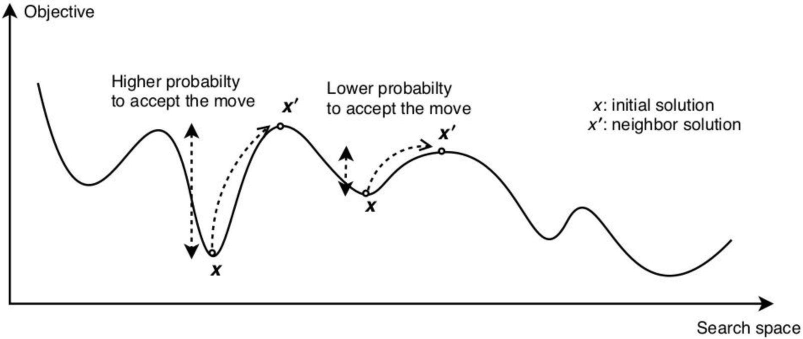


FIGURE 2.25 Simulated annealing escaping from local optima. The higher the temperature, the more significant the probability of accepting a worst move. At a given temperature, the lower the increase of the objective function, the more significant the probability of accepting the move. A better move is always accepted.

Probability function of temperature

$$P(\Delta E, T) = \mathrm{e}^{-\frac{f(s') - f(s)}{T}}$$

SA algorithm template

Algorithm 2.3 Template of simulated annealing algorithm.

Input: Cooling schedule.
$s = s_0$; /* Generation of the initial solution */
$T = T_{max}$; /* Starting temperature */
Repeat
  Repeat /* At a fixed temperature */
    Generate a random neighbor s' ;
    $\Delta E = f(s') - f(s)$;
    If $\Delta E \leq 0$ **Then** $s = s'$ /* Accept the neighbor solution */
    Else Accept s' with a probability $e^{\frac{-\Delta E}{T}}$;
  Until Equilibrium condition
  /* e.g. a given number of iterations executed at each temperature T */
  $T = g(T)$; /* Temperature update */
Until Stopping criteria satisfied /* e.g. $T < T_{min}$ */
Output: Best solution found.

Simulated Annealing (algorithm for minimization of f)

1. Make $T = T_{max}$ and
   Choose a solution u (at random)
2. Select a neighbour of u, say v
   If $f(v) < f(u)$ make $u \leftarrow v$;
   Else make $u \leftarrow v$ with probability

   $$p(fu, fv) = \exp((fu - fv)/(T \cdot fu))$$

   Repeat Step 2 k times
3. Make $t \leftarrow t + 1$;
   Set $T = T(t)$ see Eq.(1)
   If $T < T_{min}$ *Stop*;
   Else go to Step 2.

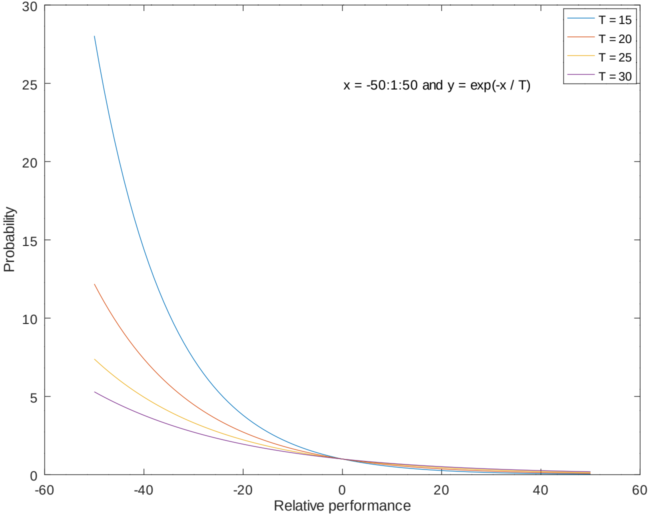
Simulated annealing (SA)

- A well known approach to vary temperature T is to mimic the temperature exponentially-decreasing function (of time) used for metal annealing, described as:

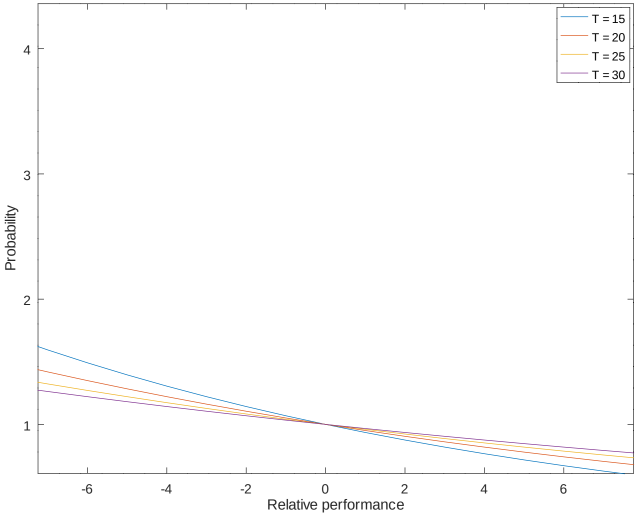
$$T(t) = T_{max} \exp(-R \cdot t) \tag{1}$$

where, R is a temperature decreasing rate and T_{max} is the initial temperature (a large value compared to R)

The effect of temperature

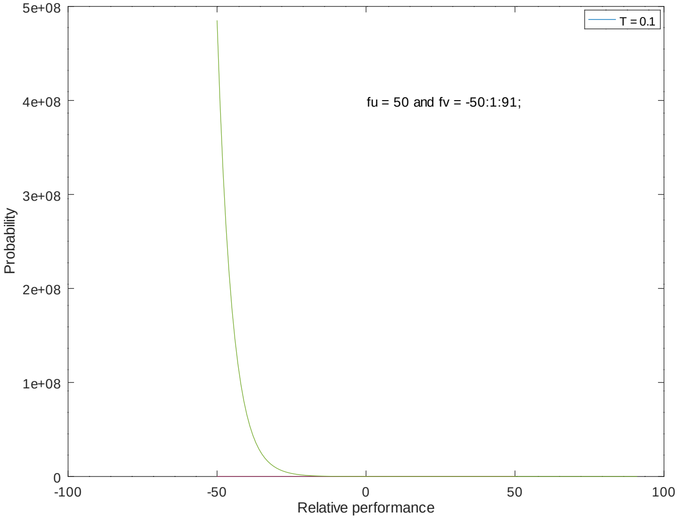


Zooming...


T = 15
T = 20
T = 25
T = 30
4
3
2
1
Probability
-6
-4
-2
0
2
4
6
Relative performance

Analysis

- We can see that when temperature is lower ($T = 15$), probability function gives higher values for <u>better</u> values of x
  - x represents the relative performance between the current solution, s, and the neighbour solution, $s^{'}$, i.e., $f(s^{'}) - f(s)$
  - We can also see that high temperatures lead to a higher chance of accepting worse solutions (see the previous slide, where when relative performance > 0, i.e., the neighbour is worse, the probability is still near 1)



5e+08
4e+08
3e+08
2e+08
1e+08
0
Probability
T = 0.1
fu = 50 and fv = -50:1:91;
-100
-50
0
50
100
Relative performance

Analysis

- In the previous slide, the probability function

$$p(fu, fv) = \exp((fu - fv)/(T \cdot fu))$$

  is used, where u is the current solution and v is the neighbour solution, and are equivalent to s and $s^{'}$, respectively
- fu and fv are the costs of the solutions, and are equivalent to $f(s)$ and $f(s^{'})$, respectively
- This is the probability function used in the SA algorithm presented in slide 8