Simulated Annealing (Kirkpatrick, Gelatt, Vecchi 1983)

- *Simulated Annealing (SA)* is a *stochastic, solution-improvement metaheuristic for* global *optimization*
  - Note: k-opt algorithms are problem-specific (TSP-specific) local search heuristics that *can* be used inside SA

  The basic idea consists in trying to *escape from local optima* by accepting, with a probability that decreases during the search, also moves that do not improve the current solution

  - On the other hand, in the basic implementations of k-opt local searches, the search stops has soon as an improvement to the current solution has not been found in its neighborhood

- Simulated Annealing was one of the first metaheuristics, and is now a "mature" metaheuristic method (more the 1,000 papers) that has been applied to a number of application domains both in continuous and combinatorial optimization

- Several results for the convergence to the global optimum are available for SA

- It is simple to implement and, if carefully designed and tuned, SA can provide good results

A physics-inspired metaheuristic

- SA was inspired by the process of *annealing of solids in metallurgy*
- Annealing is a thermal process for obtaining minimal energy states of a solid through a heat bath:
  1. The temperature of the solid is increased until it melts
  2. The temperature is slowly decreased through a quasi-static process until the solid reaches a minimal energy state in which a regular crystal structure appears
  3. The objective is to have a crystalline state with large crystals and as few as possible defects in the crystal structure
- *By analogy*, *the configuration of the atoms can be seen as the current 'solution' to an iterative optimization process, with the associated energy being the 'value' of the solution*
  - According to the temperature, the atom configuration of the solid can 'move' from one state to the other with a certain probability. The probability of a transition to a configuration with a higher energy is proportional to the temperature of the heating bath. Therefore, at lower temperatures only transitions to lower (better!) energy states are likely to happen
- Computer simulations of the annealing process:
  - *Monte Carlo* techniques have been extensively used to simulate the annealing process
  - SA is derived from the *Metropolis-Hastings algorithm* (Metropolis et al., 1953; Hastings, 1970), a Monte Carlo algorithm for the generation of sample states in a thermodynamic system
  - In more general terms, the Metropolis-Hastings algorithm can generate a *Markov chain* whose states correspond to samples drawn from a target probability distribution. The distribution does not need to be given explicitly, but it is only necessary to provide a function that dominates the target density. Therefore, the method can be effectively used when it is not possible to sample directly from a target distribution

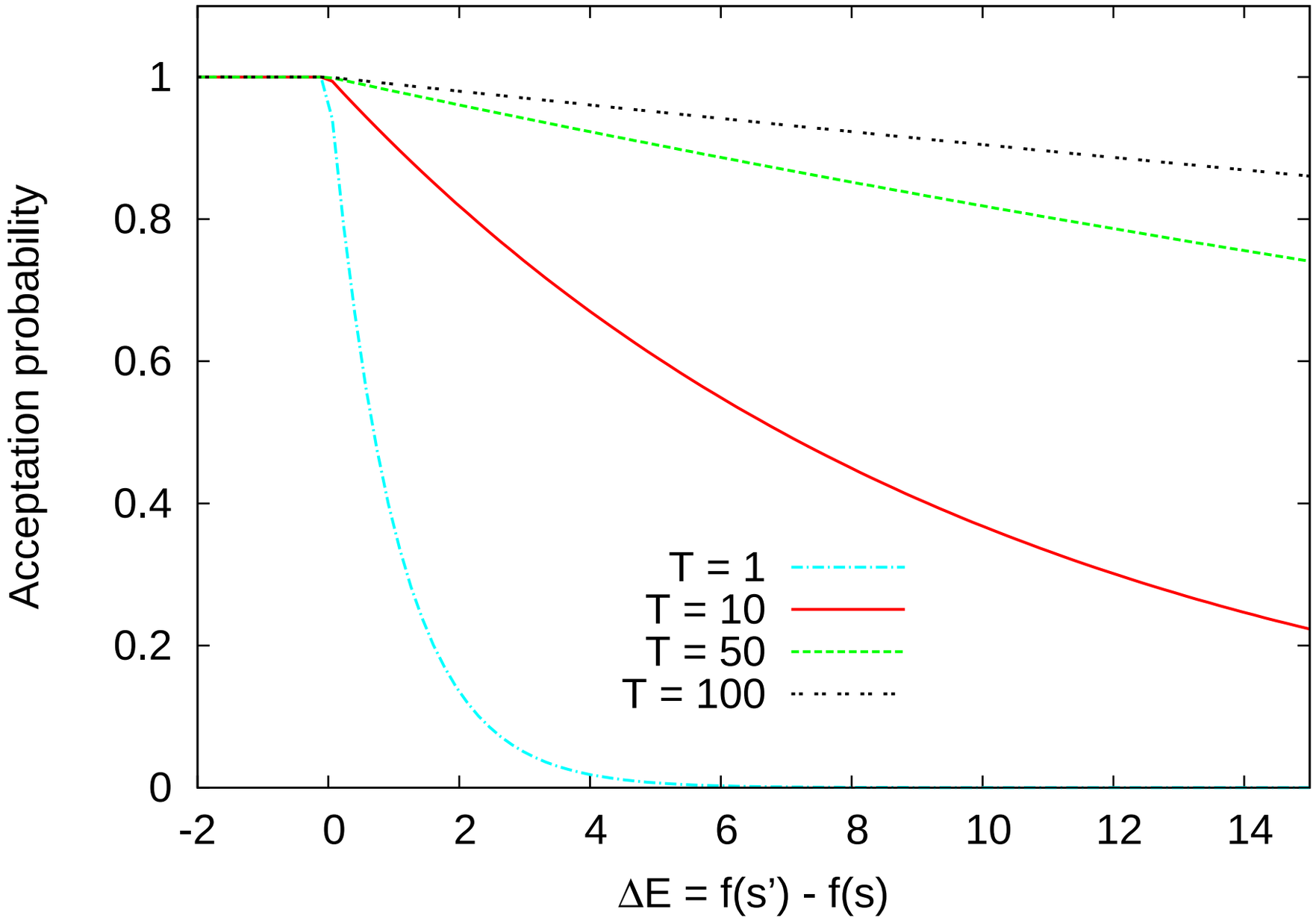
SA, general strategy

1. Start from an initial *current* solution (e.g., generated by a construction heuristic)
2. At each iteration a new solution *next* is *randomly chosen from the neighborhood* $\mathcal{N}(current)$ of the current solution
   - E.g., if the selected neighborhood function is the k-exchange one, a solution can be randomly drawn from the k-exchange neighborhood of the current solution by selecting at random k edges, removing them from the solution, and then rewiring the resulting k fragments of solution
3. If $f(next) < f(current)$, where f is the value of the objective function, *next* is set as the current solution point and the subsequent iteration will *explore* next*'s neighborhood*
4. Otherwise, if $f(next) > f(current)$ the selection between $next$ and $current$ is taken according to a *stochastic decision* based on the function $exp\Big(- \frac{f(next) - f(current)}{T} \Big)$ that considers the *'energy'* difference between the two solutions $current$ and $next$ and the value of a parameter T playing the role of *temperature*
5. Temperature is slowly *decreased across the iterations*
6. If the *termination criterion* is not met, the algorithm iterates actions 2-5. Termination can be defined according to a number of criteria, such that:
   - max cpu time
   - max number of solution transitions
   - no improvement for a certain number of iterations

Simulated Annealing metaheuristic

procedure Simulated_Annealing()
  $S = \{$set of all feasible solutions$\}$;
  $\mathcal{N} =$ neighborhood structure defined over S;
  $s \leftarrow$ Generate a starting feasible solution; // e.g., with a construction heuristic
  $s^{best} \leftarrow s$;
  $T \leftarrow$Determine a starting value for temperature;
  while (NOT YET *frozen*) // termination criterion
    while (NOT YET AT *equilibrium* FOR THIS TEMPERATURE)
      $s' \leftarrow$Choose a random solution from neighborhood $\mathcal{N}(s)$; // e.g., select a random 2-opt move
      $\Delta E \leftarrow f(s') - f(s)$;
      if $(\Delta E \leq 0)$ // downhill, locally improving move
        $s \leftarrow s'$;
        if $(f(s) < f(s^{best}))$
          $s^{best} \leftarrow s$;
      else // uphill move
        $r \leftarrow$Choose a random number uniformly from [0,1];
        if $(r < e^{-\Delta E/T})$ // accept the uphill, not improving, move
          $s \leftarrow s'$
      end if
    end while
    $T \leftarrow$Lower the temperature according to the selected cooling schedule;
end while
return s^{best};

Effect of temperature on the acceptation probability of uphill moves



Convergence properties of SA

$$P(\text{accepting } s') = \begin{cases} 1 & \text{if } f(s') \leq f(s) \\ \exp\left(-\frac{f(s') - f(s)}{T_k}\right) & \text{if } f(s') > f(s) \end{cases}$$

- Acceptation probability P depends only on the current solution and on the previous one $\rightarrow$ The solution sequence can be seen as a *Markov chain*.

- This property has been used to make a number of theoretical studies on Simulated Annealing. If T_k decreases slowly enough the algorithm will *asymptotically converge in probability to the global optimum*.
  - ✦ In general terms, *convergence can be guaranteed* if at each step the temperature drops no more quickly than $C/\log n$, where C is a constant and n is the number of steps take so far

- To guarantee convergence is often necessary to have a number of iterations k greater than the number of solutions of the problem!

- For TSP:

$$k = O\left(n^{n^{2n-1}}\right), \quad \text{while the number of solutions is 'only' } n!$$

- Cooling schedules that work in practice lack of convergence properties :(

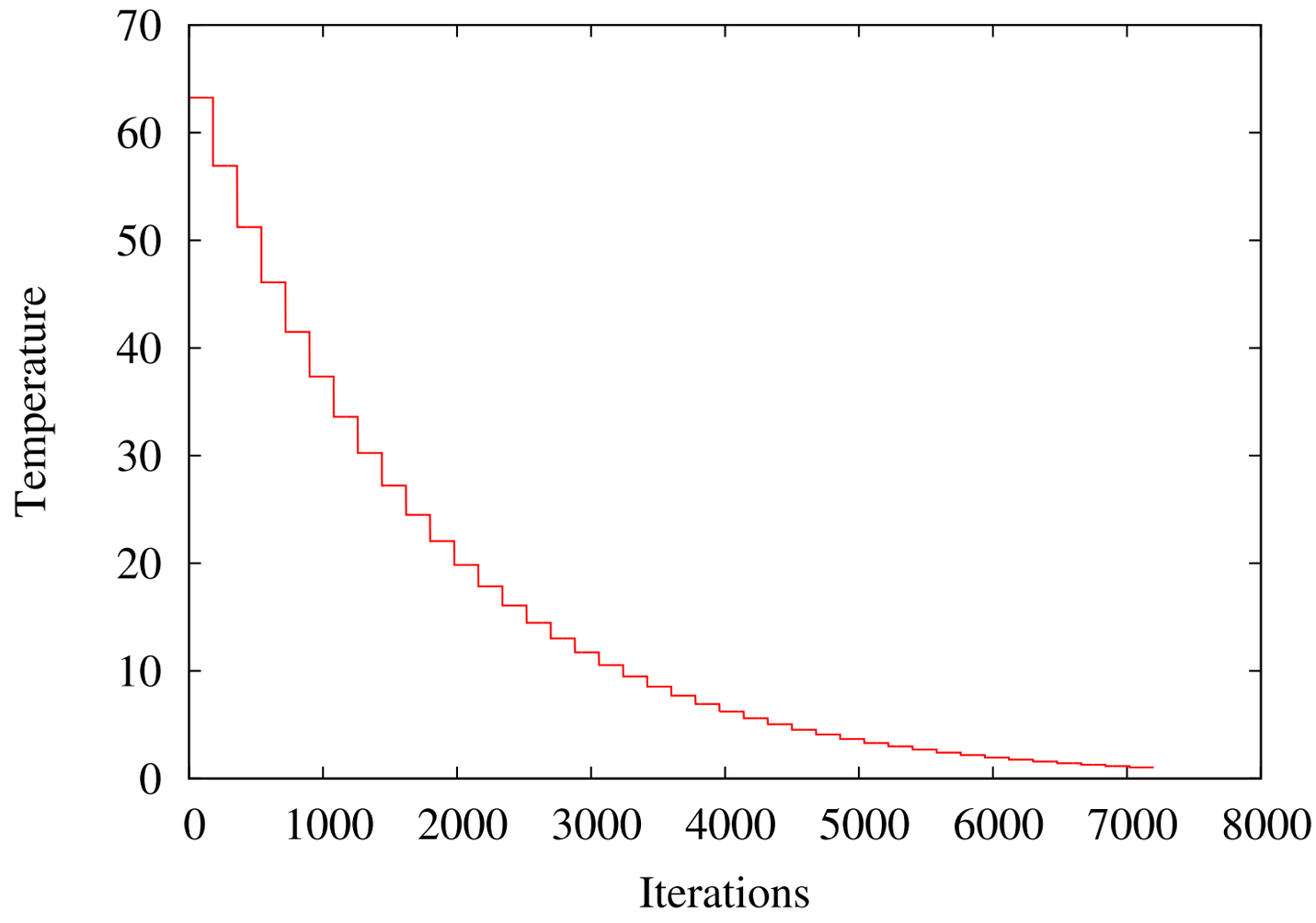
Strategies for temperature decrease

- In most of the applications, a so called *exponential cooling* strategy is adopted, such that the temperature drops roughly as $C^n,\ C \in (0,1)$: a fixed number of moves is performed at each temperature, after which one arbitrarily declares *"equilibrium"* and reduces the temperature by a standard factor, $T_{k+1} = \gamma T_k$, where $\gamma \in [0,1]$ is a constant ($\gamma = 0.95$ is a common choice)

  Under such an exponential cooling regime, the temperature reaches values sufficiently close to zero after a polynomially-bounded amount of time and the *"frozen"* state can be declared

- The definition of an exponential cooling, requires:
  - *Starting temperature value*:
    - For Euclidean TSP instances a popular starting value for the temperature is $L/\sqrt{n}$, where L is the length of the starting tour and n is the number of cities (Bonomi and Lutton, 1984)
    - Johnson (1995) proposes $1.5(L/\sqrt{n})$ that allows an expected initial acceptance of about 50%
  - *Temperature length*, that is, the number of steps between two temperature updates
    - It should be at least proportional to the *size of the neighborhood*, to guarantee a satisfactory level of local search at a given temperature
    - *Example:* in the case the 2-exchange neighborhood is used, temperature length should equal to $\alpha n(n-1)$, where α is a proportionality constant (e.g., $\alpha \in \{1, \ldots, 100\}$ in the experiments reported in the following table)

Example of temperature decrease using exponential cooling



- Example for a Euclidean TSP with $n = 10$ nodes and initial tour cost of cost $L = 200$
- $\alpha = 2$, $T_0 = L/\sqrt{n} = 63.25$, temperature length = $\alpha n(n-1) = 180$,
- $T_{k+1} = \gamma T_k, \ \gamma = 0.9$

Performance results for SA vs. 2-opt and 3-opt

		Random Euclidean Instances					
		Average Percent Excess			Running Time in Seconds		
Variant		10^2	$10^{2.5}$	10^3	10^2	$10^{2.5}$	10^3
SA_1 (Baseline Annealing)	$\alpha=1$	3.4	3.7	4.0	12.40	188.00	3170.00
SA_1 + Pruning	$\alpha=1$	2.7	3.2	3.8	3.20	18.00	81.00
SA_1 + Pruning	$\alpha=10$	1.7	1.9	2.2	32.00	155.00	758.00
SA_2 (Pruning + Low Temp)	$\alpha=10$	1.6	1.8	2.0	14.30	50.30	229.00
SA_2	$\alpha=40$	1.3	1.5	1.7	58.00	204.00	805.00
SA_2	$\alpha=100$	1.1	1.3	1.6	141.00	655.00	1910.00
2-Opt		4.5	4.8	4.9	0.03	0.09	0.34
Best of 1000 2-Opts		1.9	2.8	3.6	6.60	16.20	52.00
Best of 10000 2-Opts		1.7	2.6	3.4	66.00	161.00	517.00
3-Opt		2.5	2.5	3.1	0.04	0.11	0.41
Best of 1000 3-Opts		1.0	1.3	2.1	11.30	33.00	104.00
Best of 10000 3-Opts		0.9	1.2	1.9	113.00	326.00	1040.00
Lin-Kernighan		1.5	1.7	2.0	0.06	0.20	0.77
Best of 100 LK's		0.9	1.0	1.4	4.10	14.50	48.00