Computer Simulations
5^{th} problem set

Summer term 2025
Douwe Jan Bonthuis
Graz University of Technology

Submission deadline:
Schedule A: week of June 16
Schedule B: week of June 23
Please upload your files in the Teach Center before the assessment discussion with your tutor.

Stochastic optimization of the traveling salesman problem

A salesman has to visit N cities exactly once while minimizing the total length of the trip. The aim of this exercise is to find an optimal solution using simulated annealing. The simulated annealing technique can be summarized by saying that the search for the global minimum of a function must be performed by moving downwards most of the time but not necessarily always.

1. **Probing configurations**

(a) Generate a distribution of the positions $\{\boldsymbol{x}\}$ of N towns in two dimensions.[1] The towns are labeled S_i, where the order of the indices $i \in \{1, \dots N\}$ defines a configuration K. Calculate the total length of the round trip of the configuration K, which plays the role of the energy in the simulated annealing,

$$E(\{\boldsymbol{x}\}) = \sum_{i=1}^{N} |\boldsymbol{x}_{i+1} - \boldsymbol{x}_i| \quad \text{with} \quad \boldsymbol{x}_{N+1} = \boldsymbol{x}_1.$$

(b) Propose a new configuration K' by selecting two towns S_a and S_b at random. Exchange their position in the path:

$$K\text{:} \qquad S_1, \dots, S_{a-1}, \underbrace{S_a, S_{a+1}, \dots, S_{b-1}, S_b}_{old}, S_{b+1}, \dots, S_N$$

$$K'\text{:} \qquad S_1, \dots, S_{a-1}, \underbrace{S_b, S_{b-1}, \dots, S_{a+1}, S_a}_{new}, S_{b+1}, \dots, S_N.$$

This procedure reverses the order of the towns between a and b. The corresponding energy difference for the case $a < b$ and $(a, b) \neq (1, N)$ equals

$$E(K') - E(K) = |\boldsymbol{x}_{a-1} - \boldsymbol{x}_b| + |\boldsymbol{x}_a - \boldsymbol{x}_{b+1}| - |\boldsymbol{x}_{a-1} - \boldsymbol{x}_a| - |\boldsymbol{x}_b - \boldsymbol{x}_{b+1}|.$$

Accept K' with the probability

$$p_{acc} = \min\left(1, \frac{p_E(K'|\beta)}{p_E(K|\beta)}\right),$$

where $p_E(K|\beta)$ is given by the Boltzmann distribution $p_E(K(\{\boldsymbol{x}\})|\beta) \propto \exp\left(-\beta E(\{\boldsymbol{x}\})\right)$ and β is a parameter playing the role of the inverse temperature.

[1] Note that the „random" number sequence in pseudo-random number generators is determined deterministically by the seed number. To get a reproducible configuration of towns, the seed number can be set before generating the sequence. Python uses the Mersenne Twister algorithm to generate pseudo-random numbers, producing 53-bit precision floats with a period of $2^{19937} - 1$. Using numpy, the seed number can be set using, e.g. `numpy.random.seed(12345); X = numpy.random.random((N,2))`. In Matlab, the same sequence can be obtained using `rng(12345,'twister'); X = rand(N,2)`. After generating the town locations, you should set the random number seed differently each time for your actual simulated annealing runs. This can be done by using the clock time, for example, which is the default in Python's `numpy.random.seed()`. In Matlab, use `rng('shuffle','twister')`.

2. **Annealing**

(a) To cool the system for this example, it is sufficient to choose the k-th inverse temperature as $\beta_k = \beta_{\mathit{start}}\, k^q$, with $q \simeq 1$. This can result in very fast cooling, and may not be suitable for other applications of simulated annealing. Use $\beta_{\mathit{start}} = 1$. As a rule of thumb, the number of steps per temperature can be taken as $L \simeq N^2$.

(b) For each temperature separately, determine the best energy which has been reached, and the following averages

- average energy: $\langle E\rangle_k \simeq \frac{1}{L}\sum_j E_j$
- scaled variance: $\beta_k^2\langle \Delta E^2\rangle_k \simeq \beta_k^2(\langle E^2\rangle_k - \langle E\rangle_k^2)$

Plot these values (*not* the running averages) versus inverse temperature β_k or versus the index k. Also plot the current path several times during the simulation.

(c) Find and use a suitable convergence criterion. After the system has converged, plot the best path. How long is it?

(d) Can you be certain to have reached the global optimum? Under which conditions is the algorithm guaranteed to converge? Experiment with different choices of parameters, namely with

- different values of q and L;
- a different acceptance probability:

$$p_{\mathit{acc}} = \frac{1}{1 + \exp\left((\beta(E(K') - E(K))\right)}.$$

Qualitatively describe the effects of these parameters and settings on the results in terms of efficiency of the algorithm and quality of the final estimate.

(e) Implement a different cooling strategy by setting $\beta_{k+1} = \beta_k/r$ and using the „specific heat" to determine the cooling rate,

$$r = \frac{\beta_k}{\beta_{k+1}} > 1 - \frac{\delta}{\beta_k\sqrt{\langle E^2\rangle_k - \langle E\rangle_k^2}},$$

with δ being a small value, which can be chosen such that the acceptance rate is reasonably high (typically around 50%). Compare the best path length estimated within a given number of iterations with the results from the cooling strategy of parts (a-d).