Simulated annealing for n-queens problem

1 Simulated annealing algorithm

$s \leftarrow s_0$
$e \leftarrow E(s)$
$k \leftarrow 0$
$T_0 = tempestimation(s, E)$
while $k < k_{max}$ and $e > e_{max}$ **do**
  $s_n \leftarrow neighbour(s)$
  $e_n \leftarrow E(s_n)$
  if $P(e, e_n, temp(k)) > random()$ **then**
    $s \leftarrow s_n$
    $e \leftarrow e_n$
  end if
  $k \leftarrow k + 1$
end while
return s

2 Algorithm description

- s is a current state (board with the n-queens)
- s_0 is a initial state that is a board with n-queens, which positions are random.
- s_n is a new state
- $E(s)$ is a utility function for n-queen problem that calculates the number of attacks.
- $tempestimation(s, E)$ is a function that determines the initial value T_0. The function:
  1. run the $neighbour(s)$ function m time on initial state. As a result the m different state will be generated.
  2. Calculate the $E()$ for each m state.
  3. Calculate a mean for $E()$.
  4. Calculate the deviations of each data point from the mean, and square the result of each.

5. Calculate the variance as a mean of all squared deviations.
6. The population standard deviation is equal to the square root of the variance.
7. Return the value of population standard deviation as T_0.

- e_{max} is the value of utility function for the terminal state - solution. For n-queens problem the number of attacs i.e. 0.
- function $neighbour(s)$ generate the new state s_n that differs from state s by randomly moving one randomly chosen queen.
- Calculate the probability from the formula

$$P(e, e_n, temp()) = \begin{cases} exp^{\frac{e-e_n}{temp()}}, & \text{if } e - e_n < 0 \\ 1, & \text{otherwise} \end{cases}$$

- $temp()$ determines value of $T_{k+1} = \alpha * T_k$, where for example $\alpha = 0.95$.
- $random()$ random number from 0 to 1.