Optimization & Physics
Exercise 3

HS 07
H. G. Katzgraber

The goal of this exercise is to learn about genetic algorithms.

Problem 3.1 Genetic algorithms applied to spin glasses

We look again at the Edwards-Anderson Ising spin glass and use it as a playground for genetic algorithms. In this case, a chromosome corresponds to a spin configuration x and the fitness function to the Edwards-Anderson Ising spin glass Hamiltonian $H(x)$. Start with a pool of M chromosomes and alter these by mutations or by crossovers. In a mutation a spin is flipped with some probability p. For the crossover you can take two randomly chosen parents (chromosomes), split them at a random crossover point s and combine the four parts to two new offsprings. Keep only the two fittest of the four chromosomes. The mutations should happen less often than the crossovers. Make sure to allow for mutations in only 10 % of your chromosome-modifications and do a crossover otherwise. Plot the Hamiltonian evaluated for the fittest individual at logarithmic time steps and the fitness distribution of your final chromosome pool.

Problem 3.2 Job scheduling using genetic algorithms

Applications of genetic algorithms are of course not restricted to spin glasses and we will consider a toy-example of job scheduling in this exercise. Suppose you own a little business and have engaged three workmen W_1, W_2, W_3 and there are six tasks T_1, ..., T_6 to be done by them today in their available work hours.

	Available work hours
W_1	08.00-17.00
W_2	13.00-17.00
W_3	08.00-14.00

	Duration	Target time
T_1	4 h	17.00
T_2	3 h	17.00
T_3	3 h	13.00
T_4	3 h	17.00
T_5	2 h	08.00
T_6	3 h	17.00

In a schedule, the tasks have to be assigned to the workmen and can be represented as a list $\{\{w_1, t_1\}, ..., \{w_6, t_6\}\}$, with $w_i \in \{1, 2, 3\}$ specifying the workmen and $t_i \in \{1, 2, 3, 4, 5, 6\}$ the task. By convention, a workman performs a task T_{t_i} before T_{t_j}, if $i < j$.
This list can also act as a representation for a chromosome. Since a task should not be done multiple times, a mutation is described by a flip of two randomly chosen tasks. The crossover is done in the following way: Given two chromosomes $\{\{w_1, t_1\}, ..., \{w_6, t_6\}\}$ and $\{\{w_1', t_1'\}, ..., \{w_6', t_6'\}\}$ a random crossover point s is chosen, and the pairs $\{w_1, t_1\}, ..., \{w_s, t_s\}$ are replaced by $\{w_i', t_i' = t_k\}, ..., \{w_j', t_j' = t_l\}$, $k, l \in \{1, ..., s\}$ and $i < j$ to build a new offspring. To not upset your customers, you want to optimize the schedule such that the tasks are done before the deadline. As a cost function you can use the following:

$$f(t_{\text{end}}) = \begin{cases} 0 & \text{if } t_{\text{end}} < t_{\text{target}} - t_{\text{duration}} \\ -m_1(t_{\text{target}} - t_{\text{duration}} - t_{\text{end}}) & \text{if } t_{\text{target}} - t_{\text{duration}} < t_{\text{end}} < t_{\text{target}} \\ -m_2(t_{\text{target}} - t_{\text{duration}} - t_{\text{end}}) + p & \text{if } t_{\text{end}} > t_{\text{target}} \end{cases}$$

where t_{end} is the time a task is finished, m_1 and m_2 two positive constants and $p > 0$ a penalty. Find an optimal schedule for your staff using genetic algorithm. You may in addition optimize your schedule such that the idle time is minimal.