NAT-DL: Self-study questions **Set 1 (week 3)**

1. The knapsack (or: rucksack) problem is as follows: Given a set of weights W, and a target weight T, find a subset of W whose sum is as close to T as possible.

   Example: $W = \{5, 8, 10, 23, 27, 31, 37, 41\}$, $T = 82$

   a) Solve the instance of the knapsack problem given above.

   b) Consider solving the knapsack problem using the canonical GA. How can a solution be encoded as a chromosome?

   c) What fitness function can be used for the knapsack problem, so that better solutions have higher fitness?

   d) Given your answer to question b, what selection methods would be appropriate?

   e) Consider also the case that the total weight of the subset must not exceed T. Would you need to change your approach?
2. **Computer exercise**: Implement a simple GA with fitness-proportionate selection, roulette-wheel sampling, N=100, p_c=0.7, p_m=0.001. As fitness, use the integer value that is obtained when considering the genome of an individual as a binary number.
   a) How does the number of generations needed to find the optimum depend on the size of the genome? This is just an exercise, not an assignment, so just try a few sizes and record the result.
   b) Compare your result with a hill-climbing algorithm on the same problem.
   c) For a fixed size of the genome (D=20) the time to find the solution depends on the parameters p_c and p_m as shown in the figure 1 below (green (x) is for random, red (+) for zero initialisation). Discuss the figure.
   d) The second figure show analogous results for a “deceptive” fitness function over a discrete search space {0,1,….,31}, with $F(x) = x^2$ for x = 1, …, 30, and $F(0)$ = 961 and $F(31)$ = 0. Why are the results here independent of p_c? Why is the curve monotonous?
3. **The travelling salesperson problem** asks to find the shortest path through a set of N cities given the pairwise distances. Create randomly the (x,y) positions of 20 - 50 cities, determine their distances, and then mutate (how?) a string representing the tour and evolve a tour that leads back to the starting city with the shortest distance.
4. **Termination**: The generational process in GA is repeated until a termination condition has been reached. Termination is needed to qualify as an algorithm (by definition). What termination criteria are suitable in GAs and similar algorithms?
5. **Natural evolution**: Recall what you know about natural evolution (DNA, genomes, natural selection etc.). What features of biological evolution are reflected in Gas, and how could GAs be improved by including additional features into the algorithm design?

Fig. 1

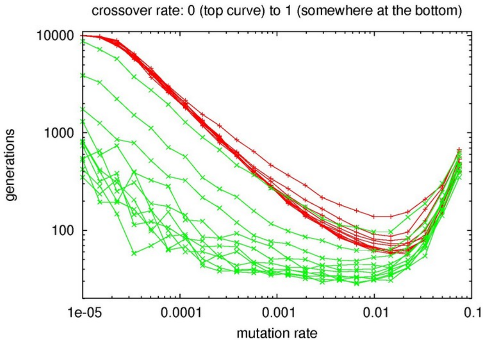

Fig. 2 (same axis titles)