Exercise 1a) **[5 Points]**

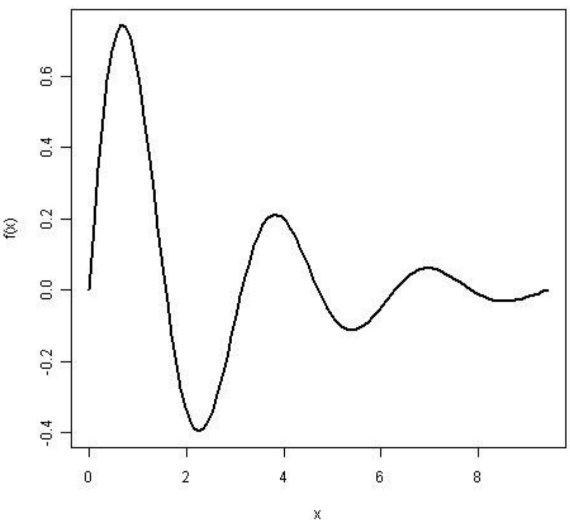
Characterize the following search methods according to their properties, for which kind of task environments (fully / partially observed, deterministic / strategic / stochastic, …) they are best suited:

- Hill climbing / Simulated Annealing
- Local Beam Search
- Genetic Algorithms
- Online Search

Do some algorithmic properties exist which make one algorithm for one kind of problem preferable to some others? Which properties and why?

Exercise 1b) **[5 Points]**

Consider the following function:



When aiming at maximizing the final score: Which solutions are found for hill climbing, stochastic hill climbing, simulated annealing, and beam search, if

- the state space is „fully connected“, that is, each sucessor state x' is reachable from any input state x,
- the successor state space is bounded wrt. $\epsilon > x' - x > 0$. Which solution is found, if the initial state is x=1, or x=4. For Beam search, assume k=3 and x1=1, x2=2, x3=6

If necessary, make further reasonable assumptions.

Exercise 2)

Optimal layout of infinitely tiny objects[1] on a 2D planar surface:

Assume you are having a fully connected graph with arbitrary many nodes (each node has vertices to each other node). Your goal is to draw the graph on a paper, with one constraint: of having minimal „energy“.

„Energy“ corresponds to the well-known fitness function / goal test „function“ which one (better: the algorithm) tries to minimize. That is, for some given planar layout of nodes, utilize the following function to get a global „fitness score“ for each edge between nodes i and j:

$$\sum_{i<j} d_{i,j}^2 + (1 - d_{i,j})$$

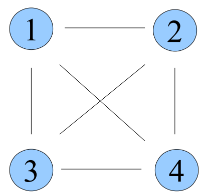
Obviously, if one adjusts the location of just one node, we will get another fitness score, which should preferably be less than the previous one.

Example (schematic drawing, assume those nodes to be on a crisp grid):

Assume the euclidean distance:

$$d_{1,2} = d_{1,3} = d_{2,4} = d_{3,4} = 1$$

$$d_{1,4} = d_{2,3} = \sqrt{2}$$



$$\sum_{i<j} d_{i,j}^2 + (1 - d_{i,j}) = 1^2 + (1-1) + 1^2 + (1-1) + (\sqrt{2})^2 + (1-\sqrt{2}) + (\sqrt{2})^2 + (1-\sqrt{2}) + 1^2 + (1-1) + 1^2 + (1-1)$$

Shortly (!) categorize the problem both according to the task environment properties, and according to the problem definition for search problems (what is the initial state, what are the successor functions, …) you learned in the lecture.

1 For sake of simplicity, we will just focus on a single point for an object, rather to cope with its actual boundaries.

Choose one out of the following (you may not get more than 20 points in total):

Variant a) **[5+5+5+5 points]**

Precisely define the input for the following algorithms:

- A* (or one of its variants)
- Hill climbing / Simulated Annealing
- Local Beam Search
- Genetic Algorithm[2]

Sketch (also possible by handwriting on paper) the initialization of each algorithm and the first two iterations.

Variant b) **[20 points]**

Choose one algorithm out of the ones mentioned in Variant a, and implement the solution.

2 A good tutorial may be found, e.g., here:
http://www.puremango.co.uk/2010/12/genetic-algorithm-for-hello-world/