XE33SCP: Individual project

Task is to

1. Design an evolutionary algorithm for chosen problem.
2. Implement and experimentally evaluate the proposed algorithm.
3. Write a report consisting of two parts
   a) Specification of chosen solution representation and definition of the fitness function. Its minimum should be the optimal solution for your task. We shall approve the representation and fitness function if they are at least a bit meaningful; otherwise we shall show you the right way.
   b) Report on the solution process of your task (the first part will be included). If you want to change the approved representation or fitness function, you can, but you must explain why you chose to make the change and in what aspects it is better than the original specification.

List of topics:

1. **Genome Sequence analysis**

Given: A set of fragmentary gene segments. Only four letters will be used: A (adenine), C (cytosine), G (guanine), and T (thymine). The segments have the same length. For example, an input could look like:

TCGG
GCAG
ATCG
CAGC
GATC

Goal is to reconstruct the gene sequence i.e. to reassemble the fragments into the shortest possible gene that can be made from these pieces. The rules for assembling the final sequence are:

1.1.1. The sequence must use all the segments (and only the segments) provided
1.1.2. You cannot flip any of the segments.
1.1.3. The best answer is the shortest answer.

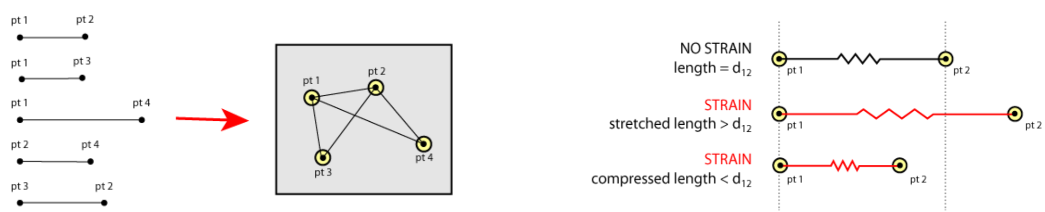
Example: CAGCAGATCGG and GATCGGCAGC are possible final sequences of length 11 and 10, respectively.

2. **Design of Molecule**

You are trying to model the structure of a molecule (which for the sake of simplicity will be limited to two dimensions).

Given: A collection of plastic connecting rods that represent the distance between pairs of atoms in the molecule, given in a symmetric distance matrix. A -1 in the [i,j] location signifies that there is no connection specified between the two atoms in question.

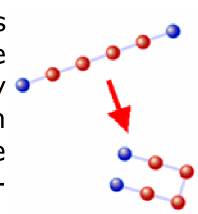
Goal is to fit the connecting rods together as well as possible into a consistent two dimensional shape so that none of the points falls outside the box limits.



This isn't always possible, so some of the lengths you specify will be slightly longer or shorter than their preferred length. **Stretching** or **compressing** the connecting rods is legal, but it causes strain, and you want to minimize strain as much as possible. The strain on the rod between any two atoms is defined as the absolute value of the difference between the preferred distance and the actual distance between atoms i and j.

3. **Protein Folding**

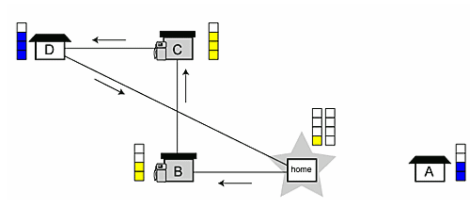
Proteins, which are initially synthesized in the cell as long chains of amino acid building blocks, automatically fold down into more compact working shapes. Exactly how they do this, and how they do it quickly and consistently, is poorly understood. The problem is so notoriously difficult that even dramatic simplifications are difficult to solve. This modification concerns a simplified two-dimensional version of the problem.



Given: An amino acid sequence of a simplified protein. In our idealized world, there are only two kinds of amino acids: hydrophobic amino acids (in red) and hydrophilic amino acids (in blue). Since proteins exist in a watery environment, the hydrophobic amino acids prefer to cling to each other, compact and apart from the water as much as possible.

Goal is to determine a final "optimally folded" configuration. Each potential arrangement of the protein sequence is associated with a free energy level, which we call the "result" of your folding algorithm. An optimal folding (and there may be more than one) corresponds to the configuration with the lowest possible result, which can be thought of as a minimum energy configuration. The free energy (result) is calculated by totaling the distance between all pairs of hydrophobic amino acids.

4. Truck routing



Given: A map populated with certain stations that contain either some amount of freight (a warehouse) or some amount of fuel (a gas station) for your truck.

Goal is to maximize the amount of freight that you can put in your truck and bring back to your home base. You will be penalized (to a lesser degree) for the amount of gas you collect. So if two trucks bring home the same amount of freight, the one that picks up less fuel is preferred. The actual distance that you travel does not by itself affect the score. Here are some guidelines.

1.1.4. Your truck must perform a closed circuit (don't run out of gas!)

1.1.5. There is no limit to how much freight you can carry

1.1.6. There is no limit to how much gas you can carry

1.1.7. Stations can be visited no more than once

1.1.8. When you reach a station, you are required to pick up all available material, whether freight or gas (which is then depleted)

1.1.9. One unit of gas will move your truck exactly one unit of distance, regardless of how much freight or gas you are carrying

5. Gerrymandering

Gerrymandering is the process of carving up an electoral district into strange shapes in order to derive a political advantage. You are given the task of preparing for an upcoming election in the state of Rectanglia. As the director of redistricting, your job is to divide the state into N districts of equal population.

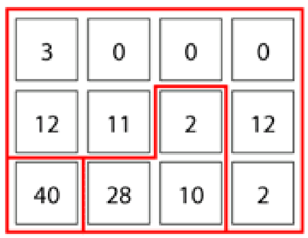
Given: A matrix A in which each element corresponds to the population in a given square mile.

Goal is to find a matrix B that indicates which voting district each square mile belongs in and voting districts must be contiguous, or connected in the four-neighbor sense (no diagonal connections).

Example. Suppose, given the following census data for Rectanglia in matrix A, you are told to divide the state into N = 3 districts of equal population.

$$A = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 12 & 11 & 2 & 12 \\ 40 & 28 & 10 & 2 \end{bmatrix}$$

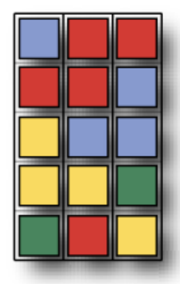
Here's one way to optimally divide the state.



$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 3 & 2 & 2 & 1 \end{bmatrix}$$

6. Blockbuster

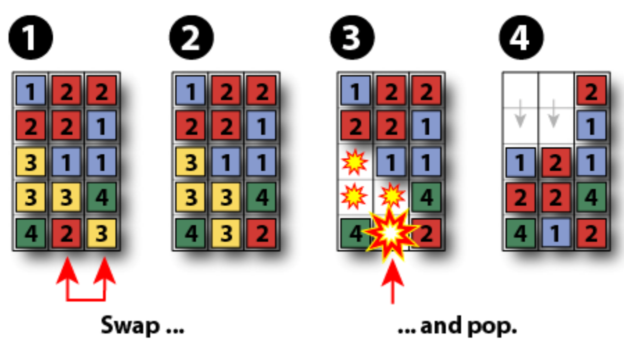
Given: A grid of numbers which you can imagine corresponds to a box filled with colored blocks.



You remove a block by popping it. When a block pops, it vanishes and all blocks above it fall down one space as though pulled down by gravity. Furthermore, all similarly colored contiguous blocks also disappear. In fact, in order to pop a block, it must have at least one similar neighbor. Isolated blocks cannot be popped.

Goal is to empty the box. In order to maximize the effectiveness of your block removal, you can perform block swaps. Pick a pair of adjacent blocks (adjacent is defined vertical and horizontal only, no diagonal swaps) and exchange their positions. You may not swap a block with empty space.

Example. Starting with the board on the left, we will first swap two blocks at the bottom and then remove four contiguous blocks.



7. **Optimal allocation of new water reservoirs**

Given: A map of M cities, each defined by its coordinates [x,y].

Goal is to build new reservoirs in $N<M$ cities so that the total length of the pipelines connecting all cities with its nearest reservoir is minimized.

Variants: Try to use different metrics (euclid, manhattan, ...)

8. **Parking problem**

Given: A set of M parking spaces on a parking lot and a set of N cars (where $M \geq N$). For each pair car-space is defined a time needed to park the car at the given space (If certain space is disabled for a car then the link is rewarded with an infinite time). Parking one at some space does not affect parking other cars at remaining positions.

Goal: Find such an assignment of cars to parking spaces that the total sum of time needed to park all cars is minimized.

9. **Distribution of beer**

Given: A map of beer consumers (pubs) and a list of daily orders; a pub either demands certain number of full barrels or wants to get rid of some empty ones. For the shipment of barrels we can use a truck with limited capacity (max. number of barrels – both empty and full).

Goal is to find the shortest tour through all pubs such that every pub is visited exactly once and none of the following situations happen

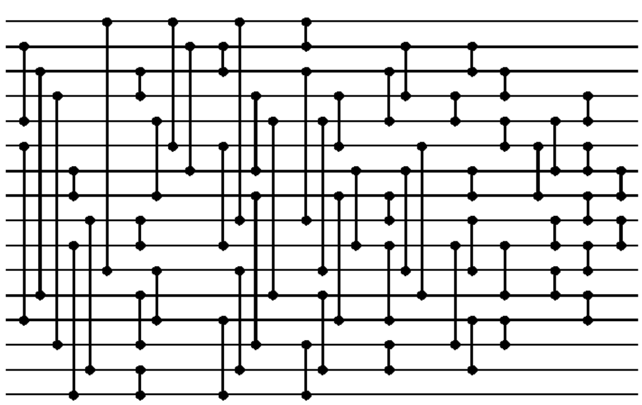
a. we are approaching a pub that demands more full barrels than we have got on the truck, or
b. we have not got enough free space on the car to be able to load all empty barrels from the pub.

10. **Sorting networks (Wikipedia)**

Given: A sequence of N numbers and a set of comparators, where each comparator connects two wires and sorts the values by outputting the smaller value to one wire, and a larger value to the other.

Goal is to design for the given sequence of numbers a sorting network with minimal number of comparators.

Example:



Each vertical line represents a comparator that switches the input numbers if they are not in desired order.

11. Piecewise linear approximation of a set of points

Given: A set of points [*x*,*y*].

Goal is to find a piecewise linear curve that minimizes the chosen objective (e.g. minimal sum of squared deviations or sum of absolute deviations).

Input parameter is the desired number of linear segments.

12. Magic squares

Given: A matrix *N* x *N* and a set of numbers from interval <1, N^2>.

Goal is to place the number into the table so that every number is used just once and the numbers in all rows, all columns, and both diagonals sum to the same constant.

13. Robotic arm with multiple joints

Given: A robotic arm consisting of a number of segments connected by 2-degree of freedom joints. All segments are of the same length. The first joint is connected to the origin at <0,0>. The target position is at <x,y>.

Goal is to set angles of the joints so that the robotic arm reaches the target position.

Parameters of the problem are the lengths of the segments and the numbers of segments.

14. Extending tram line system

Given: A map of the city with existing tram stops and the system of tram lines. Vertices represent the tram stops and the edges represent direct connection between two stops. Every edge is assigned a value specifying the the time needed to walk the way from one station to the other one. If the tram is used then the time reduces k-times.

Goal is to design a new line of length *m* so that the sum of travel times from a given node *v* to all other stops is minimized. The new line must be connected and without loops and branches.

15. Satisfiability problem (SAT)

Given: A set of logical formulas composed of boolean variables x_1, x_2, ..., x_N. Formulas are in *conjunctive normal form* (CNF)

i. NOT operators are only applied directly to variables, such as not(x12),
ii. a clause is a disjunction (OR) of literals, such as (x5 or not(x12)),
iii. a formula is a conjunction (AND) of clauses, such as (x5 or not(x12)) and (x3 or not(x7) or x11).

Goal is to assign logical values to all variables in such a way that all formulae evaluate to true.

16. Allocation of modules to processors

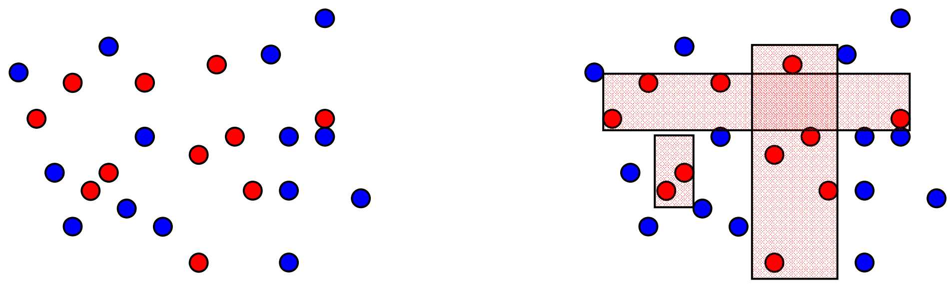
Given: A set of *N* modules; each module has assigned its computational complexity. A table of intermodule communication *C*, where position *C*[*i*,*j*] represents a communication rate between modules *i* and *j*.

Goal is to find an allocation of modules to *M* processors such that the overall computational load is evenly distributed to processors and the interprocessor communication is minimized.

17. Covering problem

Given: A set of points $X=\{(x_i,y_i): i = 1..n\}$ that is split into two disjuntive sets of positive *P* and negative *N* cases.

Goal is to cover all positive points by a minimal set of rectangular patches so that none of the negative points is covered.



a) input data b) solution

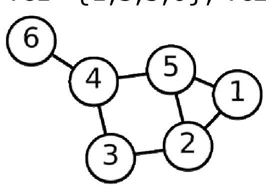
18. Vertex cover problem

Given: An undirected graph with *N* vertices and *M* edges given by a symmetric binary matrix $G[N \times N]$, where $G(i, j) = 1$, iff there is an edge between *i* and *j*.

Vertex cover is a subset *V'* of the vertices of the graph which contains at least one of the two endpoints of each edge.

Goal is to find a vetrex cover of minimum size.

Example: vc1={1,3,5,6}, vc2={2,4,5}



19. Approximating euclidean distance

Motivation: The euclidean distance must be computed in many tasks. However, it is not cheap operation since it requires a square root. There are many efforts to use approximations of euclidean distance which are cheaper, but not precise.

One of the approaches is to use the following approximation for D-dimensional space:

$$d_1(x_1,x_2) = a_1\left|\Delta_{\pi(1)}\right| + a_2\left|\Delta_{\pi(2)}\right| + \ldots + a_D\left|\Delta_{\pi(D)}\right|$$

where

$$\Delta_d = x_{1,d} - x_{2,d}$$

is the distance of the points in the d-th coordinate, and

$$\pi$$

is a permutation which sorts the deltas, so that

$$\left|\Delta_{\pi(1)}\right| \geq \left|\Delta_{\pi(2)}\right| \geq \ldots \geq \left|\Delta_{\pi(D)}\right|.$$

Goal: Using any instance of evolutionary algorithm, the goal is to find values of individual coefficients *a* that would result in an approximation that is as close as possible to the right values of distance.

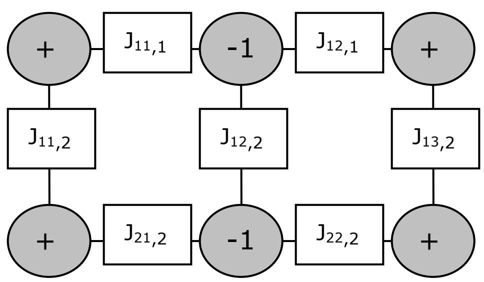
Notes:

- Choose a set of points in a grid that will serve for the evaluation.
- Try to find coefficients *a* at least for 2- and 3- dimensional spaces.
- Try to use various reasonable measures of quality of the approximation (Average error? Maximal error? Another measure?).
- Display graphically the differences between real Euclidean distance and your approximation.

20. Ground state of spin glass model

Given: A set of spins with possible positions -1 and +1. The spins are situated in a 2-D rectangular grid. The nearest neighbors *i*, *j* are connected with linkages whose strengths are defined as coupling constants J_{ij}. Energy of the model is

$$H = -\sum_{i,j} J_{i,j} s_i s_j$$



The goal is to implement and test a suitable evolutionary method that finds the state (set of spin orientations) with minimum energy (ground state). Consider just 2D model with $M \times N$ spins. The inputs will be the values of $J_{i,j}$. To prove the functionality, the method must be able to find exact solution of Ising ferromagnet, a special case of spin glass model, where $J_{i,j}$ = J>0.

21. Mastermind®

The Challenge: This task is a variation on the classic Mastermind® game. Each puzzle is an unknown sequence of colored pegs and the object is to guess which pegs they are. On each round of the game, you make a guess at the puzzle and receive feedback in the form of black and white pegs to let you know how many of your pegs are correct.

Goal: A row vector of positive integers represents the sequence of pegs, e.g. [4 24 7 3]. Each puzzle can have any number of pegs (length of vector) and any number of possible colors (maximum integer). Your task is to use a genetic algorithm to evolve the best guess for a particular sequence. The candidate solutions earn a black peg for each peg that is exactly right and a white peg for each peg that is in the solution, but in the wrong position (for scoring purposes, a black peg is worth twice as much as a white peg).

For example:

The true solution:
[4 2 4 7 3]
Guess:
[4 3 4 2 4]
Black pegs: 2 (for the first and third pegs in the guess)

White pegs: 2 (for the second and fourth pegs in the guess, which match the fifth and second pegs of the sulution but are in the wrong place).