Project teams - Algorithmics 2019

Please, fill in your project - title, team members, link to your own project environment (code repo, Drive Doc, or whatever…)

TEMPLATE (Please, don't edit):

Title
Team: Member 1; Member 2
2-3 sentences describing the project idea and goals
Link: where is project relevant information during project

Add your project topic and team here:

EXAMPLE PROJECT:

Building artificial general intelligence using deep K-nearest neighbour algorithm
Team: Jaak Vilo, Novin Shahroudi, Dmytro Fishman

First we want to develop a new algorithm by making the KNN algorithm deeper. Make our own version of AlphaGo by training a K-nearest neighbour algorithm. Use transfer learning to make our AlphaGo model mimic all models, in order to achieve an artificial general intelligence (AGI) We are very sure that this will work out as we are using the latest smartphone to make calculations. Also, we are ready to invest into buying another smartphone to exploit the wisdom of crowds.

Link: https://github.com/skyfallen/AGI (this is an example)

Knapsack problem applied to a roleplaying game
Team: Jack CROSSLEY

- The objective is to create a text based choose your own adventure where you get to choose between different treasure chests and then merchants. An algorithm will then maximise your value/weight ratio and when bartering try to do the same while taking into account cost. (Implementing a bartering aspect to the Knapsack algorithm

Link: where is project relevant information during project

Image Approximation with Genetic Algorithm
Team: Ergo Nigola

Objective is to use genetic algorithms to approximately represent images in various ways. Similar in concept to the “Evolution of Mona Lisa”, but using also some other ways for representing images, e.g. voronoi diagram, circles. Also experiment with different fitness functions, mutations, limits (e.g. limit on number of items used).

Link: https://github.com/ENigola/image-approx

Visualization of the outcome of an algorithm for finding most influential nodes out of a dataset
Team: Behrad Moeini, Hasan Mohammed Tanvir, Robin Sulg

The objective of this project is to find the most influential node of a graph network and visualize the outcome of the algorithm. In order to measure the most influential node, we propose to calculate measure of centrality and combine these to find the most influential node in the given graph network. To consider our endeavor successful the following must be achieved.

- Implement the measure of centrality, (e.g. eigenvector centrality, degree centrality, etc.) and calculate an influence score based on which the nodes will be ranked.
- Visualize the outcome of algorithm and make it accessible on the world wide web
- User interactivity

Link: http://algo-tu-2020.ee.s3-website.eu-central-1.amazonaws.com/

Optimization of travelling salesman by meta-heuristics

Team : Mehdi Hatamian

Heuristics are problem specific approximating solutions and this is exactly for problem that are difficult to solve such as Travelling salesman problem. By heuristic we want to get a good guess of best solution and by meta heuristic we want to get it faster.In this optimization we are working on an optimized solution for travelling salesman by meta-heuristics. the objective is to work on constraint in optimization problem we want to consider it by simulated annealing and tabu search .

Link of repositories Link of Poster

Comparison of maze solving algorithms with visualization GUI

Team : Quazi Saimoon Islam

The objective of the project is to implement different path solving algorithms for mazes. In short, the aim to develop a visualization tool through which different path solving algorithms i.e. Djikstra, A*, DFS etc. can be tested and their search processes visualized. The tool should be used in an educational capacity to understand how these algorithms work and why their solutions may be different. I believe that such a simple tool can be quite informative for newbies to path solving algorithms and provide a deeper understanding of their behaviours. The general operation of the tool would be to :

- Upload an image of a 2D maze (preferably B&W)
- Select the desired algorithm to run with related options
- The tool outputs a Gif of the search for the shortest paths (which nodes were visited and in which order) and finally saves the shortest path solutions.

Link: https://tinyurl.com/ucsmqox

Ant colony optimization in a maze with visualization

Team: Anne Ott, Villem Tõnisson, Alo Aasmäe

The objective of this project is to implement ant colony in maze. The user is able to generate a maze with different parameters and place an ant colony starting location. Then the algorithm will start, where the objective is to find the shortest path to food. But as time goes on, the nutritional value of different food locations will decrease, which will force the ants to search for new and more nutritious meals. In the end the algorithm should be able to balance out between different food locations to get the best possible nutrition.

Link: https://github.com/aloaas/algoproject

Animal Environmental Evolution

The goal of this project is to create an environment filled with foods of varying quality. Several “animals” are then placed into the ring, where they have to compete on who reaches the food first. Every animal that has collected enough food reproduces, with both animals getting slight mutations (alterations) in their statistics that change their decision process on what food to go

after, or if they can scare other animals away from the food. Other animals will either continue to live if they managed to get a passable amount of food, but don't mutate, or will be removed if they didn't get enough food.
The three main statistics that each animal currently has are speed, size and "quality". The first two directly correlate to their base statistics and how much energy they consume every time they move, while quality is a preferential statistic that rarely changes, but alters what type of quality food the animal goes at minimum, since the better the food quality, the more points it gives. If there isn't any food for their preferred quality, they try to find the next best one.
The animals follow a simple greedy pattern of picking out the nearest food based on its quality, their own size and whether or not the food is occupied.
All of the animals statistics are also visualized on a 3D graph next to the area to represent what type of animals thrive in the environment over time.

Team: Karl - Walter Sillaots

Link: https://github.com/K-WS/Algorithmics_Evolution
Poster PDF: Algorithmics_Evolution.pdf

Shopping optimizer

Given a list of desired items, shops with coordinates and product lists, find a list of ordered cities which minimizes the cost of purchased items.
Team: Taavi Luik, Galina Avital Pass, Ida Maria Orula

Based on no. 4 of suggested project ideas in 2019, we aim to create/use and analyze empirically and mathematically (an) algorithm(s) to solve the following problem: a list of items must be purchased from shops, each shop having a subset of items available and a cost associated with travelling to that shop; which subset and order of shops minimizes the total cost?

Link: https://github.com/cairus/shopping

Teaching AI to play Snake using genetic algorithms

Team: Meelis Perli, Alvin Meltsov, Claudia Kittask

Goal of our project is to train a neural network (NN) to play the classic game of Snake. We will be using genetic algorithms (GA) to train the weights of the feed-forward network, using weights as genes and snake instances as population. During our training we will look at the performance of different input featuresets in regards to the efficiency and speed of training the network.

Link: https://github.com/MeelisPerli/AlgorithmicsSnekGame

Visualizing Time Series with Dynamic Time Warping

Team: Nshan Potikyan, Shota Amashukeli

We build a web tool that lets the user visualize a pair of time series and detect the similarity/dissimilarity between them. Our implementation works well for already processed time series objects, as well as for raw audio files in any popular formats such as .wav, .mp3, .ogg etc. The tool creates both interactive and static graphs, which help to further investigate any particular regions of interest.

Link: https://github.com/NshanPotikyan/TimeSeriesVisualization

---

Board Game "Dobutsu Shogi" Simulation and AI

Team: Markus Rondo

The goal is to build a playable game of a shogi board game variant in Unity using C# and to implement an AI that can play the game using the minimax algorithm with alpha-beta pruning. The optimizational effects of alpha-beta pruning and different heuristics will be investigated.

Link: https://bitbucket.org/nurm/janggi

Parallelizable barcode scanning algorithm on images

Team: Erik Amor

The idea of the project is to write and test the performance of a barcode detection algorithm that would be able to detect the number encoded as a barcode from a captured image. The algorithm should traverse the image only once and one row at a time. This allows the algorithm to be run in parallel logic units on an embedded FPGA (running on actual FPGA is not part of the project). The results of individual rows are then combined to increase the accuracy of the algorithm. The algorithm should handle only barcodes that are in specific orientation or close to it, meaningful barcode can be seen on one row of pixels. The barcodes recognised in this project will be of UPC-E standards.

Link: Project repository

Assumption of Species Evolution Using Genetic Algorithm

Team: Navid Bamdad Roshan, Afsana Khan, Souvik Paul

The idea of the project is to perform crossovers and mutations among genes of 100 different species over a fixed number of generations and determine the how the species have evolved based on their modes of locomotion such as swimming, running, jumping, flying, gliding. Each of these modes of locomotion would have specific genes to denote them which is termed as reference genes. The crossover and mutated genes of the species after a fixed number of generations will be matched with the reference genes in order to find the maximum similarity. Determining similarity is done here using the Lowest Common Subsequence (LCS) algorithm. The gene length and number of generations will be taken as user input from a GUI. For convenience the reference genes and species genes are generated randomly. Lastly a bar plot showing the distribution of evolution of the species after some generations can be visualized.

Link: **Project Repository**

Generation of course timetable using artificial immune system algorithm
Team: Siim Kurvits ; Karl-Gustav Kallasmaa , Laura Katrin Leman
Implementation of artificial immune system negative selection algorithm to create a university course timetable. The timetable's fitness is evaluated by soft constraints such as penalising the last slots of the day and rewarding consecutive lecture-practice combinations.
Link: TBA

Comparison and visualization of sieve algorithms

Team: Aleksander Parelo, Mark-Eerik Kodar, Laura Liis Metsvaht

The idea of the project is to visualize two different sieve algorithms for finding prime numbers. The chosen sieve algorithms were the sieve of Eratosthenes and the Sieve of Sundaram. The visualization shows the algorithm progress step-by-step. This can be used to compare the performance of the two algorithms. We also compared the actual run-time of the two algorithms to see how the number of steps actually influences run-time.

Link: **visualization**, project repo

Visualization of Binary Space Partitioning

Team: Alessandro Stranieri

When a scene is rendered, for example in a video-game, there is the problem of the order in which the scene's objects such as walls should be drawn. With a naive Painters Algorithm implementation, ideally one would draw farther objects first, moving closer to the viewer. Before the advent of dedicated hardware, sorting objects for every point of view would have been really taxing on the rendering performance. The Binary Space Partitioning algorithm allows to pre-compute a partitioning of the space so that a fast ordering of objects is possible, as they are stored into a tree-structure. This allows to: i) create a correct order of drawing from any position of the camera/player and ii) eventually split objects that cannot be unambiguously sorted. In this project we would like to create a tool to demonstrate the how the BSP algorithm works. The tool should allow to:
1. Visualize a simple 2D map by drawing walls
2. Visualize the partitioning of the map
3. Visualize the sorting of the partitioned walls from the point of view of a camera/player

Link: https://github.com/alessandrostranieri/ut_bsp_tool

Dynamic Time Warping and Edit Distance Visualisation

Team: Yana Halas, Tetiana Shtym, Dmytro Zabolotnii

It is hard to overestimate the power of visualisation tools when it comes to the learning process. Visualisations are able to convey complex concepts in a simple manner, making knowledge accessible to more people.
During this semester we have been successfully using many visualisation tools to gain a better understanding of various algorithms. However, for Dynamic Time Warping and Edit Distance Algorithms such visualisation tools seemed not to be presented. Therefore, we decided to create ones within the project, hopefully, making studying process smoother for the next generations.

Link: Visualisation, Project Repository

Genetic Algorithm - Path finding

Team: Quentin Fabre

Genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. That's the process of natural selection where the fittest is selected for reproduction and do the same until get the main aim. Here I will create a maze with a point to reach. I will create one 2D interface of the genetic components evolution.The motivation of this project is to know how does it work clearly, and trying to "vulgarise" the way to think about the genetics algorithms. A lot of people think that's really difficult but I will try to show them that's not so hard and try to make them understand with the 2D interface.

Evolution Simulator
Team: Madis Janno

A simulation of creatures which attempt to seek food, die, and multiply. Multiplying can result in mutations which alter the stats of the creatures. The goal of the project is to see how different rulesets affect the way creatures develop over hundreds of generations and to show how very simple principles result in behaviour which resembles optimization.

Link: https://github.com/madisjanno/EvoSim

Building an image from polygons using genetic algorithm
Team: Liina Anette Pärtel, Karl Riis

We got this idea from the proposed list of projects, nr 14. The main idea is to recreate some given image using polygons of different sizes and colors, starting from a blank image. The image is initialized with some random polygons and genetic algorithm is used to reshape and recolor these polygons, hopefully creating something close to the original image.

Link: https://github.com/liinaanette/image-building-from-polygons

---

The Perfect Snake Game
Team: Youssef Sherif Mansour Mohamed - Abdelrhman Elsayed Hassan Eldallal
The goal of the project is to solve the classic game of snake perfectly. We used an algorithm based on Hamiltonian Cycle generation, where we use different Hamiltonian path every game to solve it. We used Backbite transformation to generate the different paths. In addition a number of optimization techniques were considered.
Link: https://github.com/dallal9/PerfectSnake

Multi-Object Tracking Framework
Team: Jan Aare van Gent
The goal of the project is to create a simple, yet fully functional tracking framework based on global nearest neighbour (GNN) tracking and ensemble Kalman filters for state estimation and noise filtering. The 3D detections will be provided from camera and LiDAR scans.
Link: proprietary code (Milrem Robotics)

Bin Packing Problem
Team: Mirjam Iher; Indrek Pertman
The idea is to implement an efficient bin packing algorithm that would pack items with random dimensions into a finite number of finite capacity bins or boxes. The packing order and placement will be visualised.
Link: Bitbucket

A shortest path in a brave new virtual world

Team: Dmytro Urukov, Sergei Zaiaev

The idea is to create a web-based application for building virtual worlds and executing different shortest path finding algorithms (Dijkstra, A*, etc.). The virtual world builder should be interactive and contain different types of available blocks, including portals.

Link: github: algorithms, application, demo: link

Genetic algorithm for image revolution
Team: Farhan Syakir, Roland Heichun Shum, Heidi Korp

The idea is to implement different approaches of genetic algorithm for merging images to be different resulting images which are similar to the target image. 3 different approaches have been used. They are the size mutation, swapping and color value mutation.
Link: github: algorithm

Chess AI
Team: Gamal Elkoumy, Mahmoud Kamel, Nesma Mahmoud

The aim of this project is to use advanced algorithms for enhancing the run time/ improve and AI performance in the chess game. Using AI to generate all possible moves (Search Space) from any chess board state, then looking for the best move within a reasonable time-limit.
Link: github: here, Poster: here
