Introduction to Artificial Intelligence

Bachelor in Informatics Engineering and Bachelor in European Computer Science Engineering

Year 2 – 1st semester 2023/2024

Practical Work No. 2
Optimization Problem

1. Goals

Design, implement and test optimization methods that find good quality solutions for different instances of the problem described below.

2. Minimum cost subset

Given a graph and an integer value k, this problem consists of finding a subset of vertices of size k, **such that all vertices have at least one connection** and the cost of the edges within the subset is minimal.

Formally, the problem is defined as:

Data

- an undirected graph $G = (V, A)$, composed of a set V of vertices connected to each other by weighted edges A
- an integer k

Problem

- find a subset of vertices S, of size k, such that $S \subseteq V$, **with all vertices of S having at least one connection,** to minimize the total cost of the edges of that subset
- The aim of this problem is therefore to minimise.

Instances for testing

Some instances of this issue are available in Moodle to test the algorithms. The algorithms to be implemented must be prepared to handle graphs containing up to 500 vertices. The graphs are stored in the files according to the adjacency list representation. Vertices are identified by integers starting from 1.

- The first line contains the value of k.
- Next, in a line starting with "***p edge***", are the number of vertices and edges.
- Each of the following lines begins with the letter **e**, and specifies an edge between two vertices and its cost.

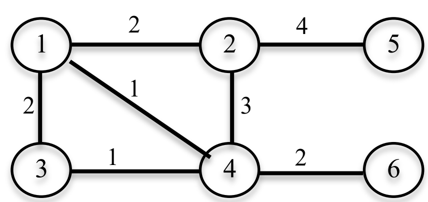
The following illustrates how the information is stored in the test instances (text files are available in *Moodle*):

`k 18`	Value of *k*
`p edge 45 918`	Graph with 45 vertices and 918 edges
`e 2 1 4`	First edge connects vertices 1 and 2 with cost 4
`e 3 1 2`	Second edge connects vertices 1 and 3 with cost 2
`e 3 2 5`	
`e 4 1 15`	...
`e 4 2 10`	
`...`	
`e 45 39 3`	
`e 45 40 5`	
`e 45 41 12`	
`e 45 42 10`	Last edge connects vertices 42 and 45 with cost 10

Example:

Consider a graph G with 6 vertices and 7 edges, with the information represented below (***test.txt***). The image on the right illustrates the graph. The goal is to find a solution to the problem for k = 4.

```
k 4
P Edge 6 7
e 1 2 2
e 1 3 2
e 1 4 1
e 2 4 3
e 2 5 4
e 3 4 1
e 4 6 2
```



Consider three possible solutions (there will be others):
S1 = {1, 2, 5, 6} – invalid solution, vertex 6 has no links
S2 = {1, 2, 3, 4} – cost of edges = 9
S3 = {2, 3, 4, 6} – cost of edges = 6

Since this is a minimization problem, solution **S3** is the best of the three.

Note: In the implementation of the optimization algorithms, it is advisable to use binary representation for the solutions. For example, for S1 = [1 1 0 0 1 1], for S2 = [1 1 1 1 0 0], for S3 = [0 1 1 1 0 1].

3. Optimization Methods

You should implement and evaluate the capacity of different optimization algorithms to find good quality solutions to the described problem. The following 3 methods should be implemented and compared. Several instance files (test.txt, file1.txt, file2.txt, file3.txt, file4.txt and file5.txt), with different complexities, are available on Moodle.

The algorithms to be implemented and tested are as follows:

1. Local search algorithm (choose one between Climbing Hill, Simulated Annealing, or other, as long as approved by the teacher)
    - Choose an appropriate representation for the solutions, binary representation is advised.
    - Implement the appropriate evaluation function.
    - Propose strategies to deal with invalid solutions (penalization, reparation, ...).
    - Explore at least two different neighbourhoods.

2. Evolutionary algorithm
    - Use the same representation and evaluation function as for the local search algorithm.
    - Propose strategies to deal with invalid solutions (penalization, reparation, ...).
    - Explore at least two different crossover operators and two different mutation operators.
    - Two different selection methods should be tested.

3. Hybrid method combining the previous two approaches.
    - At least two different hybrid approaches using the two algorithms implemented in points 1 and 2 shall be explored.

4. Experimental study

The experimental study should analyse the parameters and components of each algorithm that may have an influence on its performance.

In the local search algorithm, you should vary:

- Number of iterations
- If you choose simulated annealing: *Tmin, Tmax,* Cooling function
- Neighbourhoods
- Accept Solutions with Equal-Cost

In the evolutionary algorithm, you should vary:

- Population size
- Crossover/Mutation Operators
- Crossover/Mutation Operator Probabilities
- Selection methods
- Penalization vs reparation strategies to deal with invalid solutions.

In the hybrid approaches, you should:

- Choose the best algorithms among the previous experiments to build the hybrid approaches (best local search + best evolution)
- Compare the two hybrid approaches
- Present a comparison table with the results obtained by:
    - Best Local Search
    - Best evolutionary
    - Hybrid approach 1
    - Hybrid approach 2

Experiments should be repeated at least 10 times, and the conclusions of the study should be based on the comparison of average values.

5. Evaluation criteria

- Originality and correctness of the implemented algorithms
    - Local search (10%)
    - Evolutionary algorithm (20%)
    - Hybrid approaches (10%)
- Experimentation and analysis
    - Local search (15%)
    - Evolutionary algorithm (25%)
    - Hybrid approaches (10%)
- Documentation and presentation (10%)

6. Guidelines

- The work must be carried out **in groups of two students**. In exceptional cases work can be done individually, check with your teacher if that is possible
- The works will be subject to mandatory presentation, on a date to be announced.
- Grading: **6 values (in 20)**

7. Submission

- Deadline: **11:59 p.m. on December 17, 2023**
- A penalty of 25% **will be given** for each day of delay
- A compressed file in ZIP format must be submitted, properly identified with the names and numbers of the students. This file should contain the following:
    - **Report** addressing at least the following points
        - Representation used for the problem, description of the evaluation function, and optimization goal
        - Description of the algorithms and/or heuristics used. Explain which neighbourhoods, selection methods and genetic operators have been implemented
        - Justification of the main decisions
        - Results of the tests and their analysis. The results to be shown in the report should be a summary supported on tables/graphs showing averages of several repetitions and their conclusions. The complete study must be attached in an Excel file.
    - **Source code**, executable and test samples
    - **Excel file** with the tests for each algorithm.

- The work must be delivered through the *Moodle* platform by the indicated deadline.
- The presentations will be in the practical classes between the 18th and the 21st of December and on other days, if necessary. Presentations are mandatory and will be scheduled with the teachers of the practical classes.