Solving the Minimum Hitting Set Problem using Genetic Algorithms

Drumia Petru-Sebastian

1. Problem Description

The **Minimum Hitting Set (MHS)** problem is a classical optimization problem. Given:

- A finite universal set $U = \{1, 2, ..., n\}$
- A collection of subsets $S = \{S_1, S_2, ..., S_m\}$ where each $S_i \subseteq U$

The goal is to find the smallest subset $H \subseteq U$ such that H intersects every subset in S, i.e., $H \cap S_i \neq \emptyset$ for all $i \in \{1, ..., m\}$. This subset H is known as a *hitting set*.
Example: In the below example we can see that:

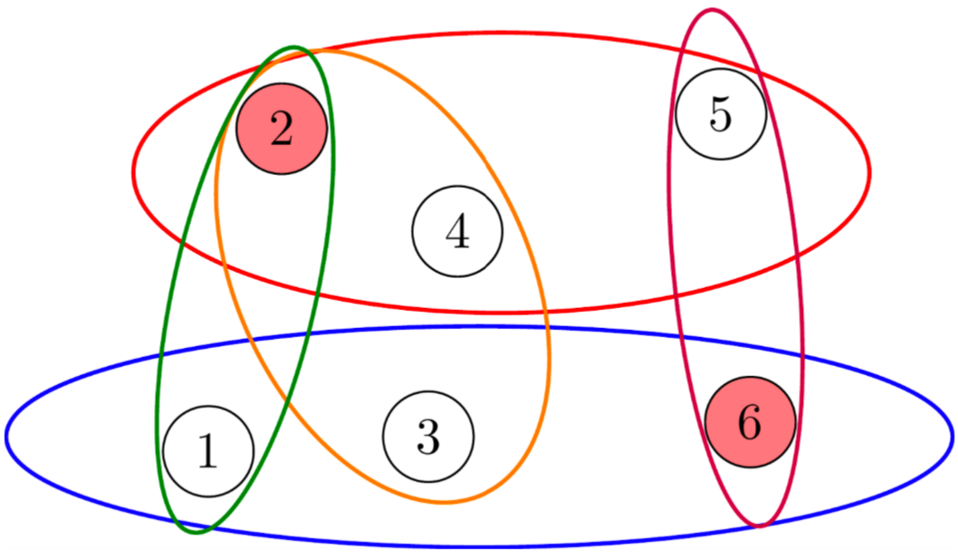

Figure 1: Visual representation of hitting set.

$$U = \{1, 2, 3, 4, 5, 6\}$$

$$S = \{\{1, 2\}, \{2, 3, 4\}, \{2, 4, 5\}, \{5, 6\}, \{1, 3, 6\}\}$$

The minimum hitting set for this example is $H = \{2, 6\}$, as seen by the red nodes.

2. Genetic Algorithm Overview

We employ a **Genetic Algorithm (GA)** to approximate a solution to the MHS problem. A GA is a population-based metaheuristic inspired by natural selection. It evolves a population of candidate solutions over iterations using selection, crossover, and mutation.

Representation

Each individual is a binary string of length n (size of U). A 1 at position i indicates that element i is included in the hitting set. For example the individual 0011001 has the following nodes included in the hitting set: $\{3, 4, 7\}$ while 0100100 has $\{2, 5\}$.

Initial Population

We generate an initial population of 150 individuals randomly.

Fitness Evaluation

The *objective function* of an individual is the number of elements selected (number of 1s). The individual is first **repaired** if it is not a valid hitting set, by greedily adding nodes with the highest frequency in S until all subsets are hit.

Fitness is computed by normalizing the objective values:

$$\text{fitness} = 1 - \frac{\text{objective} - \min}{\max - \min}$$

To maintain diversity and discourage elitism, fitness of the top 100 elite individuals is modified:

- If fitness > mean fitness: subtract mean from it.
- Else: fitness becomes 0.

Selection

Tournament selection is used with group size $k = 3$. For each individual, k candidates are selected at random and the one with the best fitness is chosen.

Crossover and Mutation

- **Crossover**: Single-point crossover is applied between pairs of individuals.
- **Mutation**: Each bit has a small chance ($\frac{1}{n}$), where n is the number of nodes, to be flipped.

Elitism and Iteration

At each generation:

1. Top 100 individuals are preserved, after changing the fitness function. These 100 go directly to the next iteration without applying genetic changes to them.
2. Remainder of population is created via selection, crossover, and mutation.

The process repeats for 5000 generations.

3. Results

A series of experiments were concluded to determine the efficancy of the algorithm. The datasets used were taken from here, a link which is found in the original presentation, of the problem, PACE 2025 - Hitting Set.

Results			
Datasets	GA alg hitting set cardinality	Time Taken ()	Optimal hitting set cardinality
bremen subgraph 300	96	206.96	85
bremen subgraph 100	32	69.68	30
bremen subgraph 50	18	46.00	18
bremen subgraph 200	64	139.32	58
bremen subgraph 20	10	27.32	10

We can observe that all taken times are under 300 seconds (5 min) which means that it fits the requirement present on the problem page.

Next steps in improving the algorithm may be: changing the selection, crossover and mutation methods so that they can better fit our problem and our needs, trying to perform a different type of local search (instead of repairing them, or changing the way we are repairing them), testing with different types of hyper-parameters for the GA and also optimizing the code where is possible for a faster runtime.