# Applying Genetic Algorithms to Solve the Minimum Dominating Set Problem Using Evolutionary Operations

Taha M.Ali 4634

**Problem Statement**

The Minimum Dominating Set (MDS) problem is a fundamental concept in graph theory, where the objective is to identify the smallest subset of vertices in a graph such that every other vertex is either included in this subset or is adjacent to at least one vertex from it. This problem can be formally defined as follows:

**Definition**

Given a graph $G=(V,E)$:

- A dominating set $D \subseteq V$ satisfies the condition that for every vertex $v \in V \setminus D$, there exists at least one vertex $u \in D$ such that $(u,v) \in E$.
- The Minimum Dominating Set seeks to minimize the cardinality of D, making it the smallest possible dominating set for the graph.

**Complexity**

The MDS problem is classified as **NP-hard**, indicating that no polynomial-time algorithm is known to solve all instances of this problem efficiently. The complexity arises from the combinatorial nature of the problem, which involves examining numerous subsets of vertices to find the minimum set that satisfies the dominating condition.

**Chromosome Encoding**

In the proposed solution, each index in the chromosome represents a vertex in the graph, and its corresponding value, referred to as a "gene," is either 0 or 1:

- A value of 1 indicates that the vertex is included in the minimum dominating set.
- A value of 0 implies that the vertex is dominated by an adjacent vertex.
  Thus, the genotype encoding utilizes binary representation.

**Example**

Consider a small graph $G=(V,E)$ where:

- $V=\{v1,v2,v3,v4\}$
- $E=\{(v1,v2),(v1,v3),(v2,v4)\}$

**Chromosome**

$$C = [1,0,1,0]$$

I[0]=1 → Vertex v1 is in the dominating set.
i[1]=0 → Vertex v2 is dominated by an adjacent vertex (in this case, v1).
i[2]=1 → Vertex v3 is in the dominating set.
i[3]=0 → Vertex v4 is dominated by an adjacent vertex (in this case, v2).

**Fitness Function**

The quality of a solution x is determined by evaluating its ability to cover all vertices in the graph. Two sets are defined: the solution set S and the dominating set D. The fitness function is given by:

$$f(x)=w+(d-s)$$

where w represents a constant factor for encouraging good solutions.

**Selection Operation**

Roulette wheel selection is employed based on fitness values. The probability pi of selecting an individual i is calculated as:

$$pi=\Sigma j=1nf(xj)f(xi)$$

where f(xi) is the fitness of the i-th individual, and n is the total number of individuals. This method ensures that solutions with higher fitness are more likely to be chosen.

**Crossover Operation**

The solution described in this work utilizes a one-point crossover. Alternative crossover methods can also be applied without significantly affecting the ability to find good solutions within the search space.

**Mutation Operation**

A one-bit flip mutation is used, with a mutation rate of 0.01.