736 Computational Intelligence - Coding Assignment Report

Anthony DeFoe, anthony.defoe@ndsu.edu

# Overview

The goal of this assignment is to design an optimization algorithm to select optimal locations for installing electric vehicle charging stations in a hypothetical city by applying Genetic Algorithm techniques covered in the course.

In this report, we will cover the problem domain, the implementation of a genetic algorithm to reach an optimal solution, experimentation on the algorithm, and a discussion of the algorithm’s performance.

# Problem Formulation

*Environment Definition*

To model the electric vehicle charging station optimization problem, I first need to identify the key components that describe both the city environment and the possible decisions that the optimization algorithm can/will make.

The “city” can be modeled as a 2D grid, where every point on the map corresponds to a pair of (x,y) coordinates. All coordinates lie within the city’s bounds such as:

For this problem, there are possible station positions *P* where the simulated city could install charging stations. For example, they could correspond to parking lots, malls, or gas stations. We can denote this as:

Where *N* is the total number of possible charging stations. Each of these locations will have a fixed cost *ci* which models the total installation and maintenance cost.

Next, to model where demand for charging access exists, we identify demand points *D* on the coordinate plane.

Where *M* is the total number of demand points. Each point *j* also has a weight which reflects the demand density or population at that demand point.

We will use Euclidean distance between possible charging point *i* and demand site *j*:

A demand point is considered covered if where *R* is the coverage radius (for example, 5 miles). In this context, “miles” correspond to units of distance on the coordinate grid; for instance, the distance between points (1, 1) and (1, 4) is 3 units, or equivalently, 3 miles.

To develop a chromosome representation of the possible layouts from which we can assess total fitness, we denote decision variable where each potential charging site is represented by a binary variable:

This directly corresponds to a gene in the genetic algorithm chromosome. A chromosome of length *L* (one bit per possible charging station site) defines which grid coordinates are chosen for charging stations. For example:

Means that sites 1, 4, and 5 are selected.

To measure demand point coverage, we will use a coverage variable , indicating whether it is served by any selected charging station:

This tells the algorithm how well the city’s population is covered by the current configuration of stations generated by the algorithm.

*Objective Function*

Each chromosome in the genetic algorithm represents one possible layout of charging stations on the grid. The genes correspond to the decision variables . The fitness of each chromosome is computed from coverage and cost:

Where controls the trade-off between coverage and cost.

The objective function can be written as the maximization of the above formula:

*Constraints*

1. Coverage constraint: A demand point can only be considered “covered” if it lies within the radius *R* of at least one active section.
2. Capacity constraint: The capacity constraint ensures that the total cost of selected charging stations does not exceed the available budget, much like a real city. This prevents the algorithm from simply choosing every available location to maximize coverage.

Where *B* is the total available budget.

1. Variable constraints:

# Algorithm Design

To meet the objectives of maximizing demand coverage while minimizing cost, we implement a Genetic Algorithm (GA). The GA provides a population-based, evolutionary approach that searches for an optimal combination of charging station locations. Each iteration (generation) of the algorithm applies processes inspired by natural evolution: selection, crossover, and mutation to improve candidate solutions over a length of time.

The algorithm uses the fitness (objective) function defined earlier:

This function rewards configurations that maximize demand coverage while penalizing high installation and operating costs. The weighting factor controls the balance between these competing objectives.

*Chromosome Representation*

Each candidate solution is represented as a binary chromosome consisting of genes, one for each potential station site:

If , a station is built at site ; if , the site is not used. This structure directly maps to the binary decision variable defined in the problem formulation and allows simple bitwise genetic operations.

*Algorithm Process*

The algorithm begins by generating an initial population of random chromosomes. Each gene is initialized as 0 or 1 with equal probability (0.5). This random initialization ensures that the starting solutions are diverse while preventing early bias towards any specific configurations.

Each chromosome’s quality is measured by the fitness function . For every candidate solution, the algorithm:

1. Calculates the distance between every demand point and each selected site.
2. Marks a demand point as covered if any selected station lies within the coverage radius .
3. Computes total demand coverage and total cost .
4. Evaluates overall fitness .

If the total cost exceeds the available budget , the fitness value is reduced by a penalty so that unfeasible solutions are less likely to be selected. The penalized fitness is defined as:

Here, and are normalized coverage and cost ratios. The first term encourages coverage, the second accounts for general cost trade-offs through 𝜆, and the third introduces an additional penalty (weighted by α) if the total cost surpasses the budget limit. This prevents the algorithm from favoring unrealistically expensive configurations while still allowing exploration near the budget boundary.

Next, chromosomes are chosen for reproduction according to their fitness. We will use a roulette-wheel selection so that each individual’s probability of being selected for reproduction is proportional to its fitness value:

High-fitness solutions have a greater chance of producing offspring, while weak solutions still retain a small probability of being selected to maintain diversity.

Pairs of selected chromosomes undergo single-point crossover. A random crossover point is chosen, and all genes after that point are exchanged between two parents to create two offspring. For example:

Crossover is applied with a probability of approximately to ensure a balance between exploration of new solutions and preservation of existing performative solutions.

After crossover, each offspring is subjected to bit-flip mutation with a small probability ranging between 1-3% (will be fined tuned during testing phase). Mutation introduces random changes in individual genes by flipping 0s to 1s or 1s to 0s.

This ensures that the algorithm continues to explore new areas of the solution space and avoids becoming trapped in local optima. Mutation represents the natural process of random variation in biological evolution, providing diversity across generations.

Once new offspring are generated, they replace the previous population to form the next generation. To preserve the best solution discovered so far, the algorithm implements elitism (the fittest individual from the current generation is automatically carried forward into the next one without alteration). This guarantees that the best-found configuration is never lost due to random variation, ensuring steady progress toward the optimal solution.

The algorithm repeats the evaluation, selection, crossover, and mutation steps for a set number of generations or until the improvement in average population fitness becomes negligible. The stopping criteria include:

1. A maximum number of generations.
2. A convergence threshold where no significant improvement is observed.

Both criteria will be evaluated during the testing phase.

The algorithm repeats the evaluation, selection, crossover, and mutation steps for a set number of generations or until the improvement in average population fitness becomes negligible.

Once the termination condition is met, the GA outputs the best-performing configuration found during the evolutionary process. The final results include:

1. The selected charging station layout (.
2. The number of covered demand points.
3. The total installation cost.
4. The corresponding fitness score .

# Experimentation

todo