

## lab9

February 3, 2024

```
[1]: # 9. Write a program for Genetic algorithm to maximize  $f(x)=x^2+4x-4$ 

import numpy as np

# Define the objective function to maximize
def objective_function(x):
    return -x**2 + 4*x - 4

# Genetic Algorithm parameters
population_size = 50
generations = 100
mutation_rate = 0.1

# Initialization: Generate a random population
population = np.random.uniform(-10, 10, size=population_size)

# Main loop
for generation in range(generations):
    # Evaluate the fitness of each individual in the population
    fitness = objective_function(population)

    # Select the top individuals based on fitness
    sorted_indices = np.argsort(fitness)[::-1]
    selected_population = population[sorted_indices[:population_size]]

    # Crossover: Create new individuals by combining pairs of selected
    ↪ individuals
    crossover_population = np.random.choice(selected_population,
    ↪ size=population_size)

    # Mutation: Introduce random changes to some individuals
    mutation_mask = np.random.rand(population_size) < mutation_rate
    mutation_population = np.random.uniform(-1, 1, size=population_size)
    crossover_population[mutation_mask] += mutation_population[mutation_mask]

    # Replace the old population with the new one
    population = crossover_population
```

```
# Find the best individual in the final population
best_individual_index = np.argmax(objective_function(population))
best_x = population[best_individual_index]

# Print the result
print(f"Optimal x: {best_x}")
print(f"Optimal f(x): {objective_function(best_x)}")
```

```
Optimal x: -1.4298916788497522
Optimal f(x): -11.764156928642771
```

```
[ ]:
```