```
def fitness-function(sequence):
    return fitness_value
population_size =100
crossover_rate =0.7
mutation_rate =0.01
num_iterations =10
num_genes =100
population: intialize_population(population_size, num_genes)
for generation in range(num_genes):
    fitness_values =[fitness_function(individual) for
                      individual in population]
    selected_population: select_population(population,
                          fitness_value)
    offspring_population =[crossover(selected_population[i],
     selected_population[i+1], crossover_rate)
     for i in range(0, len(selected_population), 2)]
  mutated_population [mutate(offspring, mutation_rate) for
   offspring in off-spring population)

  print ("Best solution": best_individual)
  print ("Best fitness": best_fitness)
  Output:
Best solution [-0.8033799  1.99833483  0.46499453
               2.55857116  0.16386048  2.23798819]

Best Fitness: 52.26534065842369
```