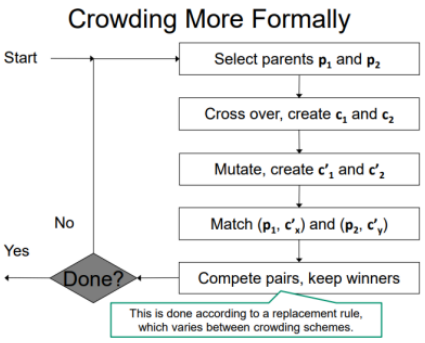
Bio-Inspired Artificial Intelligence

Project 1

Parte 1

Methodology

The evolutionary process follows the steps reported in the image:




```
for indice=1:50
    global curr_pop
    # Evaluate population
    fitness, total_weight = evaluate_population(curr_pop, profits, weights, MAX_CAPACITY)
    # Select indices of parents to crossover
    crossovering_parents = tournament_selection(fitness, 400, 10)
    # Crossover
    offspring = two_point_crossover(curr_pop[crossovering_parents, :])
    curr_pop = vcat(curr_pop, offspring)
    # Mutation
    mutation_max_n_genes!(curr_pop, PROB_MUTATION, GENES_MUTATED)
    # Survival selection
    fitness, total_weight = evaluate_population(curr_pop, profits, weights, MAX_CAPACITY)
    best_candidates_indices = elitism(fitness, N_POP)
    # update curr_pop
    curr_pop = curr_pop[best_candidates_indices, :]
    # update fitness history
    push!(mean_fitness, mean(fitness))
    push!(max_fitness, maximum(fitness))
    push!(min_fitness, minimum(fitness))
end
plot_fitness_evolution(mean_fitness, max_fitness, min_fitness)
```

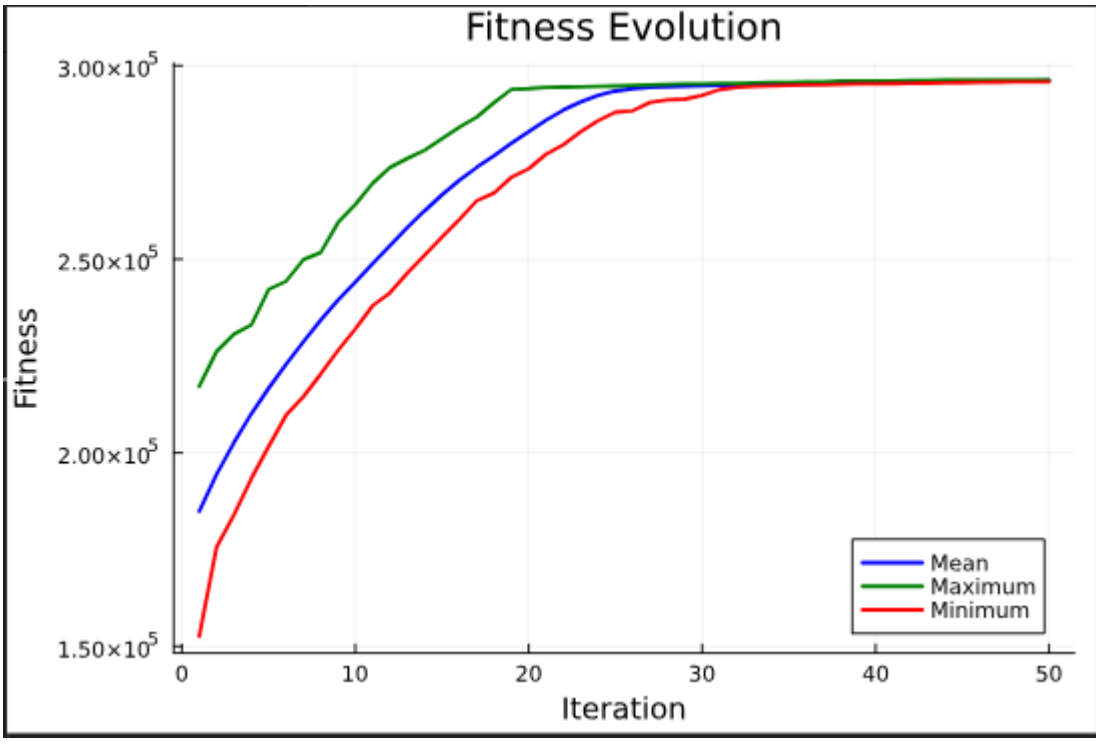
In particular, the initial version of the project included:

- **Initialization**: It starts with a population of 1000 random species.
- **Evaluate Population**: The sum of the profits of the items included in the knapsack minus a penality for any excess weight beyond the knapsack's capacity.
- **Parent Selection**: Tournament selection of size 10.
- **Crossover**: Two-point crossover.
- **Mutation**: Mutation of two random genes, each with a probability of 0.8.
- **Survival Selection**: Elitism to select the best 1000 individuals from the population.
- **Termination**: The iteration stops after the 50th cycle.

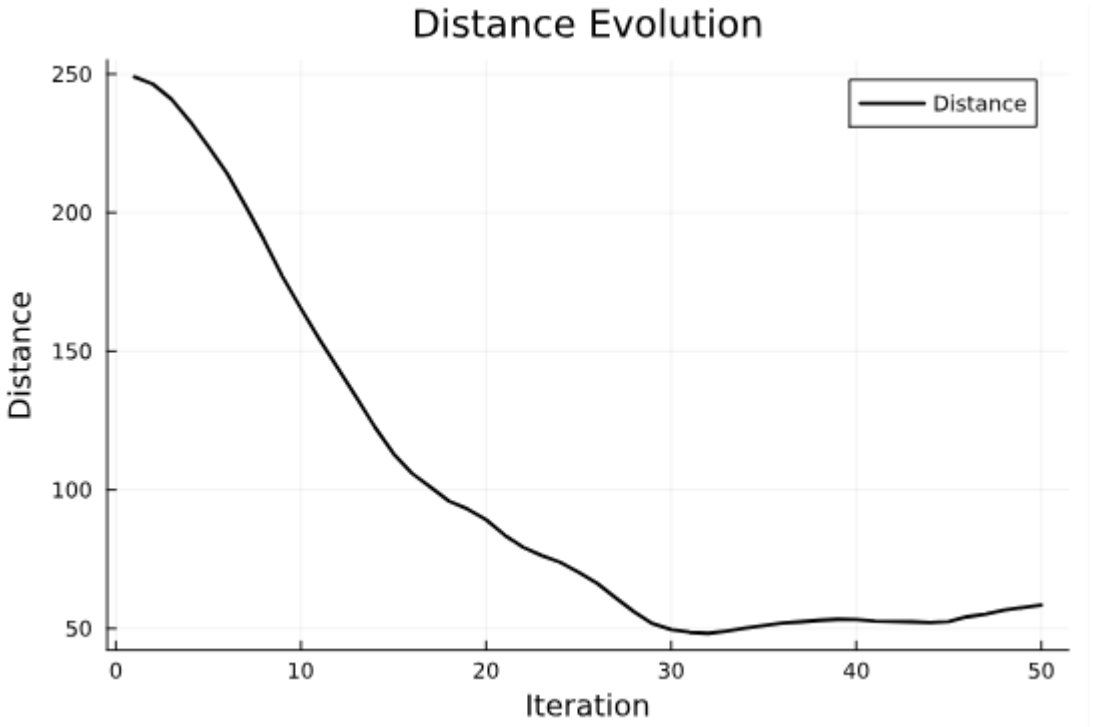
4. Results

At the end of 50 iterations, the following parameters were recorded:

- **Best Fitness**: The highest fitness value obtained.
- **Best Solution**: The configuration of items that maximizes profit while respecting capacity constraints.
- **Fitness Evolution Graph**: Displays the progression of solution improvement.



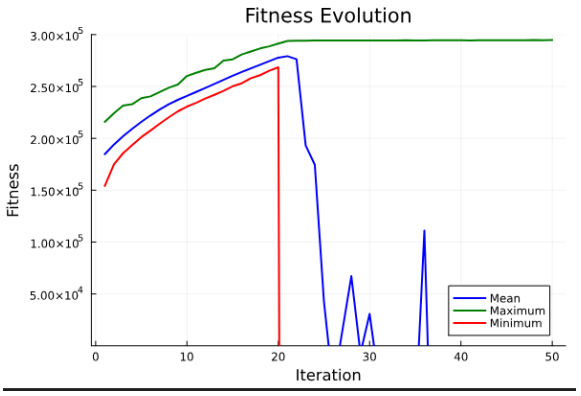

```
Fitness of the best individual: 296419
Weight of the best individual: 300985/280785
```

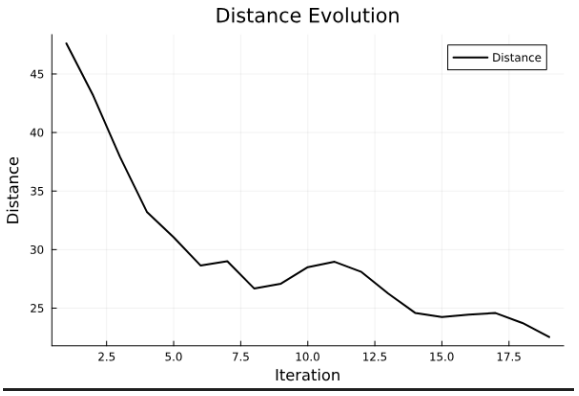


These are the final results for the first implementation of the evolutionary algorithm on the knapsack problem. We can observe that the solution slightly exceeds the maximum weight but is very close to the optimal solution, which is 296,835.

To obtain a solution with an admissible weight, we increased the penalty function by raising the cube of the weight difference relative to the imposed weight limit. Those are the results:

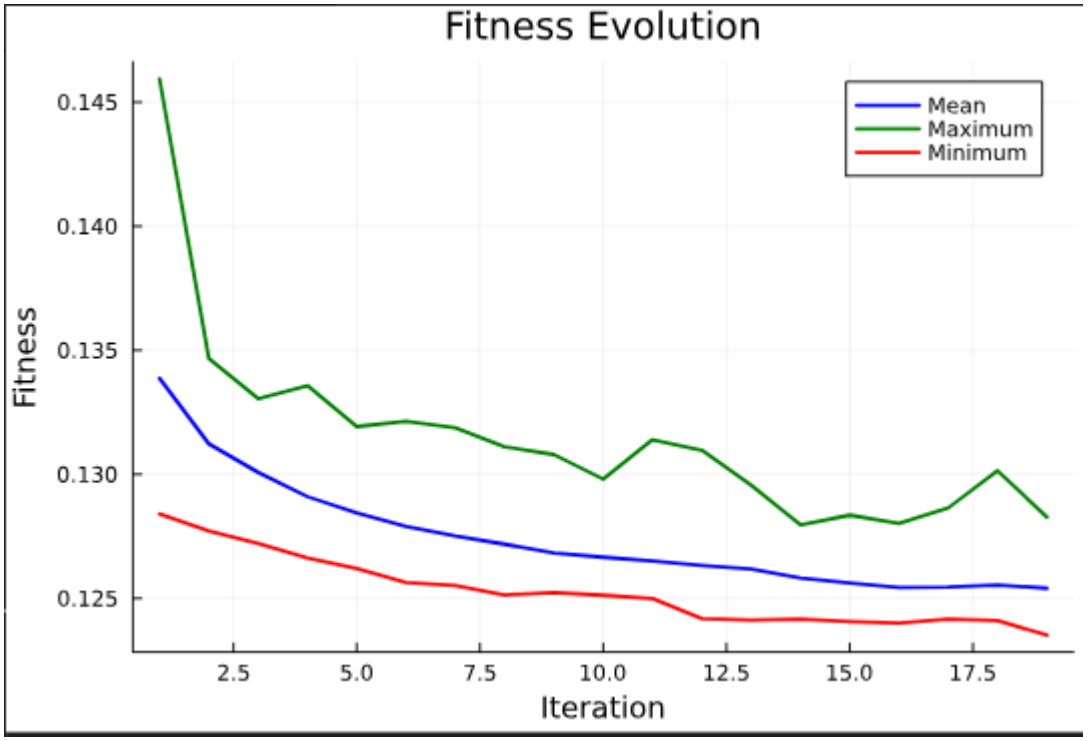
```
Fitness of the best individual: 294738
Weight of the best individual: 280744/280785
```

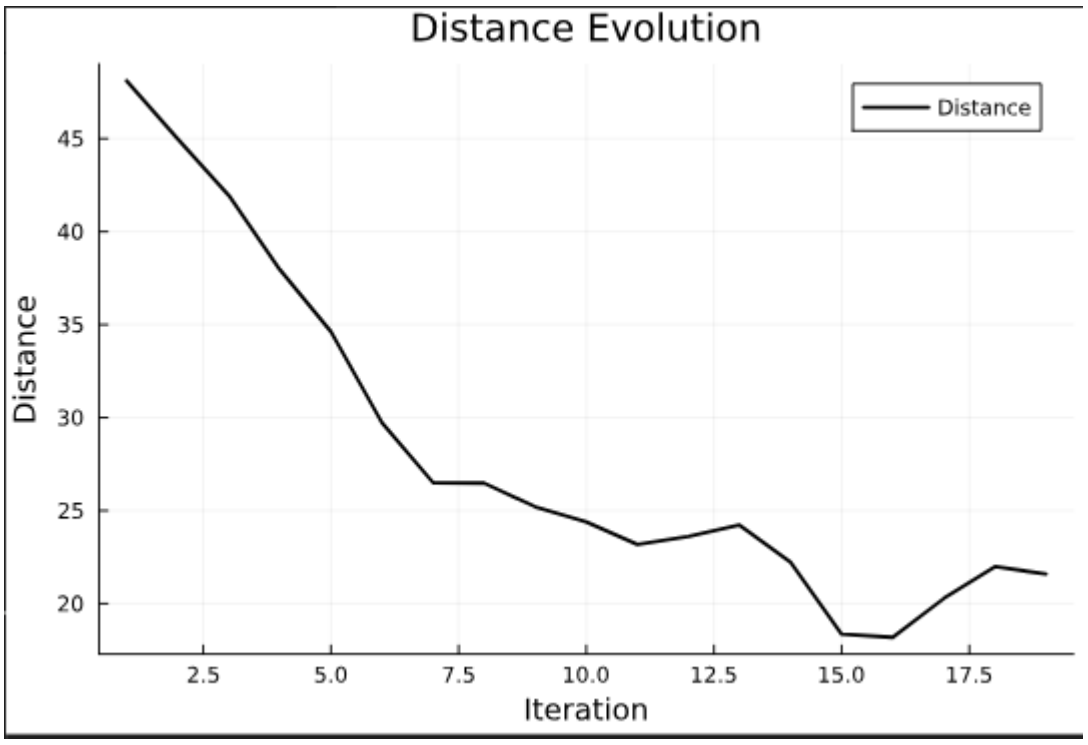




Part 2

In this case, the goal is to minimize the RMSE of a linear regression model to achieve optimal feature selection. We only needed to modify the fitness function while keeping in mind that it is a minimization problem. Those are the results with a population of 100 and a mutation of four random genes, each with a probability of 0.8





At the iteration number 20 we found a good solution with rmse=0.1235.

We also tried Roulette Wheel Selection did not found a possible solution due to its tendency to over-prioritize strong individuals early on, reducing genetic diversity. Tournament Selection, on the other hand, provided a more balanced approach, leading to better exploration of the search space.

Sin problem

the problem was once again to maximize the fitness score. the problem seemed easier to solve: in a few iterations with just a population of 10 individals found a close solution: