

Genetic Algorithm / Backpack problem

Implement the genetic algorithm as discussed in the presentation.

The screenshot shows a software application titled 'MainWindow' with various sliders and buttons for configuring a genetic algorithm. The sliders are set to: Capacity 350, #Items 100, Min Size 15, Max Size 100, Min Val 1, and Max Val 8. A 'Generate Items' button is located below the sliders. The application displays the following statistics:

- 100 Items, Total value: 480, Total size: 5773
- 15 Items, Total value: 104, Total size: 342 (under the 'Ratio' table)
- 15 Items, Total value: 105, Total size: 350 (under the 'Genetic' section)

At the bottom, there are sliders for Population (1000), Crossover % (95), Mutation % (5), and Elitism % (3).

Compare the result with the "ratio" variant, where the relatively most expensive objects are packed first. Also experiments with the parameters of the algorithm (crossover, mutation, population, elitism).

The project with the UI is attached to this document.

Most of the helper methods are also already programmed, so you can focus on the implementation of the algorithm and how to handle crossover and mutation.

The following methods have to be implemented (they currently hold a compileable dummy implementation):

Backpack.cs

- Mutate
- CrossOver
- Repair

ConventionalAlgos.cs

- GetItemsBaseOnValue
- GetItemsBaseOnSize
- GetItemsBaseOnRatio
- GetItemsThatFit

GeneticAlgorithmEngine

- section FindBest
 - FindBest
- section Next Generation
 - CreateNextGeneration
 - GetBestAndCalculateTotalFitness
 - AddElite
 - ShouldTerminate
- section Selection
 - SelectCandidateByRouletteWheel
 - SelectCandidateViaTournament