Genetic Algorithm: Bin Packing Problem

Matthew D. Branson

Department of Computer Science

Missouri State University

Springfield, MO

branson773@live.missouristate.edu

Abstract

This paper presents the implementation and analysis of a...

Index Terms

Genetic Algorithms, Bin Packing Problem, Optimization, Heuristic Search

I. Q1: Encoding and Initialization

Each solution (individual) is represented by a list of integers indicating bin assignments. For example: `[0, 1, 0, 2, 1, 1, 2, 1, 3, 0]`.

- This means that 10 orders are assigned to bins 0 through 3. (Orders can be 10, 25, 50, and 100).
- You can assume the maximum number of boxes (bins) equals the number of items.
- Implement a function to generate a random initial population of such individuals.

II. Q2: Function Evaluation

A. Objective Function

Implement a function to evaluate the objective function f, defined as the number of unique bins used (i.e., the goal is to minimize f).

B. Constraint Handling

Also compute the constraint violation g, defined as the number of bins whose total weight exceeds 10 kg.

Examples:

- If no bin exceeds the limit, $g = 0$ (feasible).
- If one bin exceeds the limit, $g = 1$.
- If all bins exceed the limit, $g = N$ (worst case).

C. Evaluation Function

Each individual will thus be evaluated using both f and g.

III. Q3: GA Operations

A. Tournament Selection

- For each selection, choose two individuals at random and determine a winner.
- Repeat until the mating pool has N individuals.
- Then select pairs from the mating pool for crossover.

B. Winner Selection Rules

- If both individuals are feasible ($g_1 = 0$, $g_2 = 0$), the one with lower f wins.
- If one individual is feasible and the other is not, the feasible one wins.
- If both are infeasible, the individual with the lower g wins.

C. Crossover

Implement one-point or two-point crossover between two selected parents.

D. Mutation

For each gene in the chromosome, mutate it with a probability $p_m = \frac{1}{\text{number of boxes}}$.

- Mutation means reassigning the gene to a new random bin (0 to $N - 1$).
- Ensure that genes only hold integer values.

IV. Q4: GA Execution

- Run the GA for 50 generations with a population size of 20 considering 10, 25, 50, and 100 orders separately.
- Vary the number of population size and max generation to see the effect on performance.
- Plot the best and average fitness (i.e., number of bins used) per generation.
- Report the best solution found and its bin-wise packing configuration.

V. Q5: Analysis and Comparison

A. Convergence Analysis

Compare the convergence behavior of different runs (e.g., different random seeds or order sizes).

B. Performance Comparison

Discuss whether the algorithm tends to find feasible solutions early or late in the process.

VI. Conclusion

This study successfully implemented...