# Payload Selection for Space Mission: CSIS 2430 Programming Project 3 Report

## 1. Introduction

The 0/1 knapsack problem is a classic combinatorial optimization task in which we are given a set of items, each with an associated weight and value (or rating), and a maximum capacity. The goal is to select a subset of items whose total weight does not exceed the capacity, while maximizing the sum of their values. In its general form the problem is NP-complete, and exact worst-case algorithms run in pseudo-polynomial time.

In our “payload selection” scenario, each item represents a scientific instrument or resource module with a known mass (weight) and an anticipated scientific utility score (rating). The objective is to pack a spacecraft’s cargo hold—limited by a maximum payload mass—in a way that maximizes overall mission return.

This project implements and compares:  
- Three simple greedy heuristics:  
 1. Greedy by highest rating  
 2. Greedy by lowest weight  
 3. Greedy by highest rating-to-weight ratio  
- An exact dynamic-programming solver that guarantees the optimum for integer weights.

We evaluate each approach on both a small sample dataset and a larger randomly generated dataset, measuring solution quality (total rating vs. optimal) and runtime performance.

## 2. Methods

Greedy by Rating: Sort items in descending order of rating and add each item if it fits. Time complexity: O(n log n).

Greedy by Weight: Sort items in ascending order of weight and add each item if it fits. Time complexity: O(n log n).

Greedy by Ratio: Compute density ρ = rating/weight. Sort in descending order of density and add items if they fit. Time complexity: O(n log n).

## 3. Implementation Details

Language: Python 3.13.3, standard libraries (csv, argparse).

Project structure:  
- data/: CSV files  
- src/: Implementation modules  
- tests/: Unit tests  
- benchmarks/: Benchmark scripts  
- generate\_random\_items.py: Data generator

## 4. Experiments

Sample Dataset: 5 items, capacity=10.

Random Dataset: 500 items (weight 1-20, rating 1-100), capacity=500.

## 5. Results

### 5.1 Sample Dataset Benchmarks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Capacity | Heuristic | Total Rating | Time (s) | Approx. Ratio |
| 50 | Ratio | 38.50 | 0.000008 | 1.013 |
| 50 | Weight | 38.50 | 0.000003 | 1.013 |
| 50 | Rating | 38.50 | 0.000002 | 1.013 |
| 50 | DP Opt | 38.00 | 0.000024 | 1.000 |
| 100 | Ratio | 38.50 | 0.000002 | 1.013 |
| 100 | Weight | 38.50 | 0.000002 | 1.013 |
| 100 | Rating | 38.50 | 0.000001 | 1.013 |
| 100 | DP Opt | 38.00 | 0.000040 | 1.000 |
| 200 | Ratio | 38.50 | 0.000001 | 1.013 |
| 200 | Weight | 38.50 | 0.000001 | 1.013 |
| 200 | Rating | 38.50 | 0.000001 | 1.013 |
| 200 | DP Opt | 38.00 | 0.000074 | 1.000 |

### 5.2 Random Dataset Results (Capacity = 500)

|  |  |  |
| --- | --- | --- |
| Heuristic | Total Rating | Weight Used |
| Ratio | 7969.80 | 499.70 |
| Weight | 6327.70 | 496.80 |
| Rating | 4673.10 | 499.40 |

## 6. Discussion

The ratio heuristic consistently yields the best solution quality, especially on large datasets. Greedy methods run in O(n log n), while DP is O(nW), which limits its scalability.

## 7. Conclusion

The ratio-based greedy offers near-optimal solutions quickly, making it suitable for practical payload selection. The DP solver guarantees optimality but at greater computational cost.

## 8. References

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.
2. Martello, S., & Toth, P. (1990). Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons.
3. CSIS 2430 Programming Project 3 specification (Spring 2025).