An Improved Method For Knapsack Problem.

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessalegn Hirpa(AIMS), Morgan KAMGA(WITS), Alain MBEBI(AIMS), Claude Michel NZOTUNGICIMPAYE(AIMS), Blessing OKEKE, Milaine SEUNEU(AIMS), Simphiwe SIMELENE(WITS), Luyanda NDLOVU(WITS), Joseph KOLOKO (UP)
Introduction of the Knapsack Problem

Objective

Algorithms
   Brute Force Method
   Greedy Algorithm Method
      Description of the Greedy Algorithm
      Problems and Benefits of this Methods

Proposed Improvements

Comparison of Results

Conclusion
Introduction

- Knapsack problem consists of finding the best packing configuration to maximise benefit while abiding by the weight constraint
- Knapsack Problem cannot be solved in polynomial time
The Knapsack Problem
The Knapsack Problem
Mathematical Formulation

\[
(KP) : \quad \max \sum_{i=1}^{n} b_i x_i \\
\text{s.t.} \quad \sum_{i=1}^{n} \omega_i x_i \leq C \\
\quad \quad x_i \in \{0, 1\}
\]
Objective

- Brief review of existing methods
- Benefits and Pitfalls
- Implement an Algorithm
- Improve optimality while being mindful of time constraints
Brute Force Method

- Enumerates every possible packing configuration
- Choose the best solution
- Optimality is ensured
- Extremely costly in time, for large n

Franklin DJEUMOU (WITS), Byron JACOBS (WITS), Dessalegn Hirpa (AIMS), Morgan KAMGA (WITS), Alain MBEBI (AIMS), Claude Michel NZOTUNGICIMPAYE (AIMS), Blessing OKEKE, Milaine SEUNEU (AIMS), Simphiwe SIMELENE (WITS), Luyanda NDLOVU (WITS), Joseph KOLOKO (UP)

An Improved Method For Knapsack Problem.
Brute Force Method

- Enumerates every possible packing configuration
- Choose the best solution
- Optimality is ensured
- Extremely costly in time, for large n
Greedy Algorithm

Let $e_i = p_i / w_i$ be the efficiency of item $i$. The greedy algorithm first sorts items in the decreasing order with respect to their efficiency. i.e item $i$ comes before item $j$ if $e_i > e_j$. It then selects the most efficient item available and places it in the knapsack, reducing the knapsack's available capacity.
Greedy Algorithm

- Let $e_i = p_i / w_i$ be the efficiency of item $i$. 

Franklin DJEUMOU (WITS), Byron JACOBS (WITS), Dessalegn Hirpa (AIMS), Morgan KAMGA (WITS), Alain MBEBI (AIMS), Claude Michel NZOTUNGICIMPAYE (AIMS), Blessing OKEKE, Milaine SEUNEU (AIMS), Simphiwe SIMELENE (WITS), Luyanda NDLOVU (WITS), Joseph KOLOKO (UP)

An Improved Method For Knapsack Problem.
Greedy Algorithm

- Let $e_i = p_i/w_i$ be the efficiency of item $i$.  

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessalegn Hirpa(AIMS), Morgan KAMGA(WITS), Alain MBEBI(AIMS), Claude Michel NZOTUNGICIMPAYE(AIMS), Blessing OKEKE, Milaine SEUNEU(AIMS), Simphiwe SIMELENE(WITS), Luyanda NDLOVU(WITS), Joseph KOLOKO (UP)

An Improved Method For Knapsack Problem.
Greedy Algorithm

- Let \( e_i = \frac{p_i}{w_i} \) be the efficiency of item \( i \).
- The greedy algorithm first sorts items in the decreasing order with respect to their efficiency. i.e item \( i \) comes before item \( j \) if \( e_i > e_j \).
Greedy Algorithm

- Let $e_i = p_i / w_i$ be the efficiency of item $i$.
- The greedy algorithm first sorts items in the decreasing order with respect to their efficiency. i.e item $i$ comes before item $j$ if $e_i > e_j$. 
Greedy Algorithm

- Let $e_i = \frac{p_i}{w_i}$ be the efficiency of item $i$.
- The greedy algorithm first sorts items in the decreasing order with respect to their efficiency. i.e item $i$ comes before item $j$ if $e_i > e_j$.
- It then selects the most efficient item available and places it in the knapsack, reducing the knapsack’s available capacity.
Problems and Benefits of this Methods

The Greedy Algorithm does not solve the problem to optimality. It rather finds a local optimal solution. It operates in linear time, which is extremely efficient. Will occasionally produce the optimal result.
Problems and Benefits of this Methods

- The Greedy Algorithm does not solve the problem to optimality.
Problems and Benefits of this Methods

- The Greedy Algorithm does not solve the problem to optimality.
- It rather finds a local optimal solution.
Problems and Benefits of this Methods

- The Greedy Algorithm does not solve the problem to optimality.
- It rather finds a local optimal solution.
- It operates in linear time, which is extremely efficient.
- Will occasionally produce the optimal result.
Proposed Improvements

- Use Genetic Algorithm
- Include the Greedy Solution in the population
Genetic Algorithm
Genetic Algorithm

- Generate Population
Genetic Algorithm

- Generate Population
- Include Greedy Solution
Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
- Crossover
Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
- Crossover
- Mutation
Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
- Crossover
- Mutation
- Next Generation
An Improved Method For Knapsack Problem.

Improvement Over Generations

Plot of Performance of Genetic Algorithm over 20,000 Generations Taking 124 seconds

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessalegn Hirpa(AIMS), Morgan KAMGA(WITS), Alain MBEBI(AIMS), Claude Michel NZOTUNGICIMPAYE(AIMS), Blessing OKEKE, Milaine SEUNEU(AIMS), Simphiwe SIMELENE(WITS), Luyanda NDLOVU(WITS), Joseph KOLOKO (UP)
Time Comparison

**Outline**
- Introduction of the Knapsack Problem
- Objective
- Algorithms
- Proposed Improvements
- Comparison of Results
- Conclusion

**Franklin DJEUMOU (WITS), Byron JACOBS (WITS), Dessalegn Hirpa (AIMS), Morgan KAMGA (WITS), Alain MBEBI (AIMS), Claude Michel NZOTUNGICIMPAYE (AIMS), Blessing OKEKE, Milaine SEUNEU (AIMS), Simphiwe SIMELENE (WITS), Luyanda NDLOVU (WITS), Joseph KOLOKO (UP)**

**An Improved Method For Knapsack Problem.**
Performance

Comparison of Genetic Algorithm and Greedy Algorithm

Performance of Genetic Algorithm over Greedy Algorithm

Franklin DJEUMOU (WITS), Byron JACOBS (WITS), Dessalegn Hirpa (AIMS), Morgan KAMGA (WITS), Alain MBEBI (AIMS), Claude Michel NZOTUNGICIMPAYE (AIMS), Blessing OKEKE, Milaine SEUNEU (AIMS), Simphiwe SIMELENE (WITS), Luyanda NDLOVU (WITS), Joseph KOLOKO (UP)
An Improved Method For Knapsack Problem.

Time Comparison

- Time Performance of GA with Increasing number of items
- Time Performance of Brute Force Method with Increasing Number of Items
Time Comparison

Graph Comparing Efficacy of Genetic Algorithm and Greedy Algorithm Against Brute Force Method

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessalegn Hirpa(AIMS), Morgan KAMGA(WITS), Alain MBEBI(AIMS), Claude Michel NZOTUNGICIMPAYE(AIMS), Blessing OKEKE, Milaine SEUNEU(AIMS), Simphiwe SIMELENE(WITS), Luyanda NDLOVU(WITS), Joseph KOLOKO (UP)

An Improved Method For Knapsack Problem.
Conclusion

- Genetic Algorithm has a small time cost for a potential improvement
- Further improvements can be made to GA by generating a initially fit population, through small amounts of brute force
- The crossover technique can be further optimized for large $n$
Thank you!!!!
Thank you!!!!
Any question is most welcome!!!!