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)

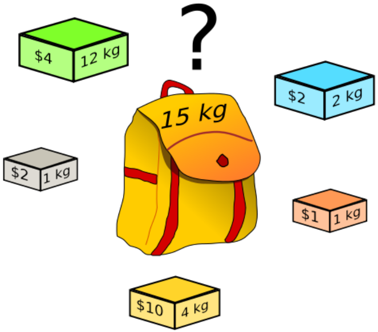
January 8, 2011

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

$$(KP): \qquad \max \sum_{i=1}^{n} b_i x_i$$

$$s.t. \qquad \sum_{i=1}^{n} \omega_i x_i \leq C$$

$$x_i \in \{0,1\}$$

Objective

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

Brute Force Method

Brute Force Method

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

Greedy Algorithm

Greedy Algorithm

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

Greedy Algorithm

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

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 $e_i = p_i / w_i$ 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 $e_i > e_j$.
- 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

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

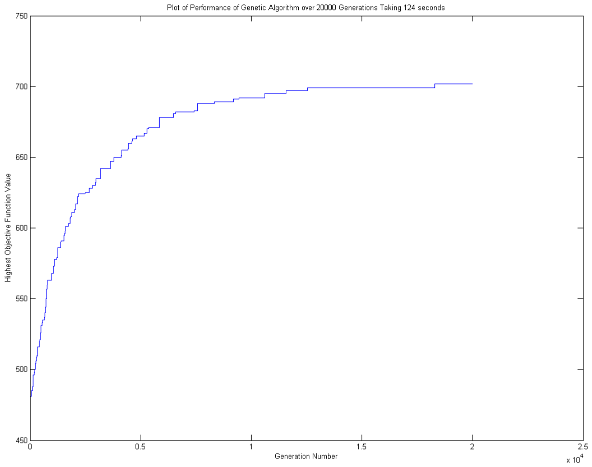
Genetic Algorithm

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

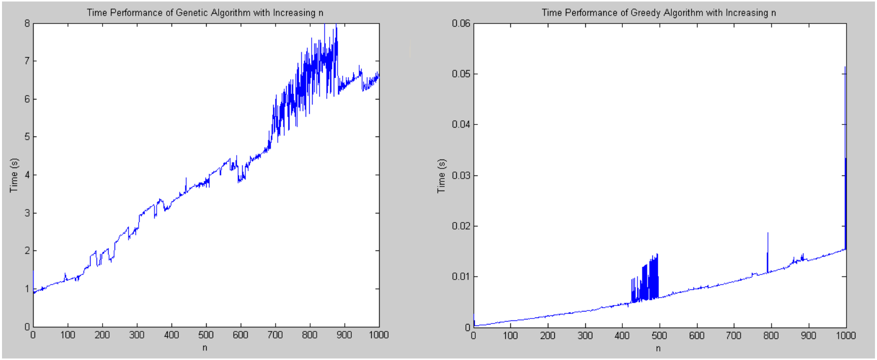
Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
- Crossover
- Mutation
- Next Generation

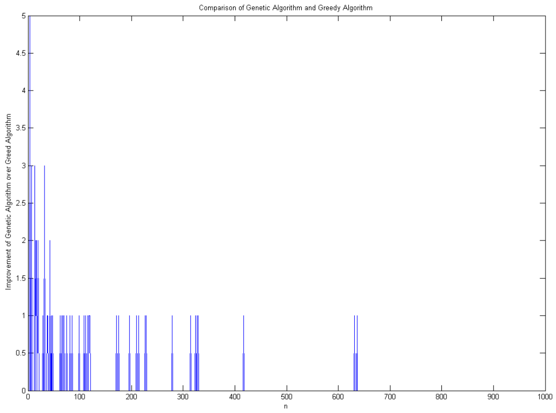
Improvement Over Generations



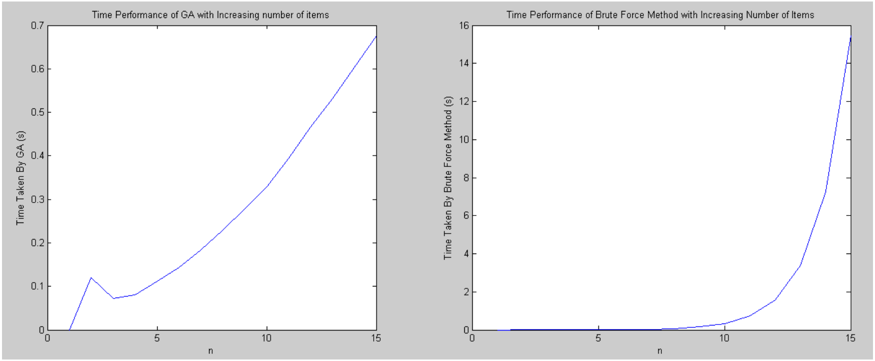
Time Comparison



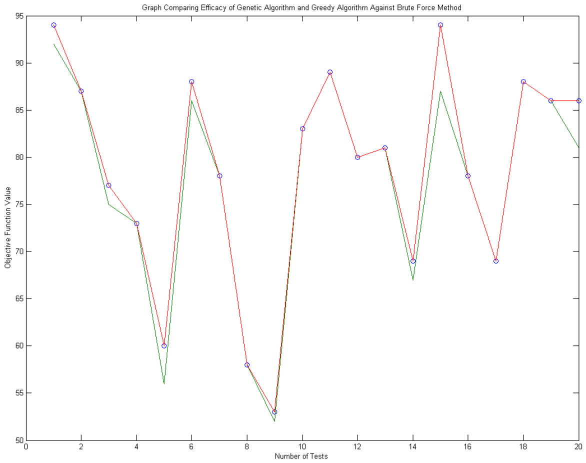
Performance



Time Comparison



Time Comparison



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!!!!