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)

<u>Januarv</u> 8. 2011

Outline

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

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

向下 イヨト イヨト

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

・ 同 ト ・ ヨ ト ・ ヨ ト

The Knapsack Problem

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg An Improved Method For Knapsack Problem.

イロン イヨン イヨン イヨン

Э

The Knapsack Problem



イロン イヨン イヨン イヨン

Э

Mathematical Formulation





- - 4 回 ト - 4 回 ト

æ

Objective

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

・ 同 ト ・ ヨ ト ・ ヨ ト

3

Brute Force Method Greedy Algorithm Method

・ロト ・回ト ・ヨト ・ヨト

æ

Brute Force Method

Brute Force Method Greedy Algorithm Method

・ 同 ト ・ ヨ ト ・ ヨ ト

Brute Force Method

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

Brute Force Method Greedy Algorithm Method

・ロト ・回ト ・ヨト ・ヨト

æ

Greedy Algorithm

Brute Force Method Greedy Algorithm Method

・ロン ・回と ・ヨン ・ヨン

æ

Greedy Algorithm

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

Brute Force Method Greedy Algorithm Method

・ロン ・回と ・ヨン ・ヨン

æ

Greedy Algorithm

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

Brute Force Method Greedy Algorithm Method

- 4 同 6 4 日 6 4 日 6

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.

Brute Force Method Greedy Algorithm Method

- 4 同 6 4 日 6 4 日 6

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.

Brute Force Method Greedy Algorithm Method

- 4 同 6 4 日 6 4 日 6

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.

Brute Force Method Greedy Algorithm Method

・ロン ・回と ・ヨン ・ヨン

æ

Problems and Benefits of this Methods

Brute Force Method Greedy Algorithm Method

- 4 同 6 4 日 6 4 日 6

э

Problems and Benefits of this Methods

 The Greedy Algorithm does not solve the problem to optimality.

Brute Force Method Greedy Algorithm Method

(4月) イヨト イヨト

3

Problems and Benefits of this Methods

- The Greedy Algorithm does not solve the problem to optimality.
- It rather finds a local optimal solution.

Brute Force Method Greedy Algorithm Method

・ 同 ト ・ ヨ ト ・ ヨ ト

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

・ 同 ト ・ ヨ ト ・ ヨ ト

Genetic Algorithm

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg An Improved Method For Knapsack Problem.

イロン 不同と 不同と 不同と

Э

Genetic Algorithm

Generate Population

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg An Improved Method For Knapsack Problem.

- 4 回 2 - 4 回 2 - 4 回 2 - 4

Э

Genetic Algorithm

- Generate Population
- Include Greedy Solution

| 4 回 2 4 U = 2 4 U =

Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection

(4回) (4回) (4回)

Genetic Algorithm

- Generate Population
- Include Greedy Solution
- Selection
- Crossover

(4回) (4回) (4回)

Genetic Algorithm

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

・ 同 ト ・ ヨ ト ・ ヨ ト

Genetic Algorithm

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

→ ∃ →

Improvement Over Generations



Time Comparison



イロト イヨト イヨト イヨト

Performance



Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg An Improved Method For Knapsack Problem.

イロン イヨン イヨン イヨン

Э

Time Comparison



イロン イヨン イヨン イヨン

Э

Time Comparison



Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg

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

・ 同 ト ・ ヨ ト ・ ヨ ト

 The crossover technique can be further optimized for large n

Thank you!!!!

Franklin DJEUMOU(WITS), Byron JACOBS(WITS), Dessaleg An Improved Method For Knapsack Problem.

・ロン ・回と ・ヨン ・ヨン

Э

Thank you!!!! Any question is most welcome!!!!

伺下 イヨト イヨト

э