



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Bacharelado em Engenharia de Software

Fundamentos de Projeto e Análise de Algoritmos

Trabalho Prático

**Laura Lourdes Coutinho Rodrigues
Pedro de Sousa Motta**

Belo Horizonte

2024

a) Solução com Backtracking

Na abordagem de backtracking, foram testadas todas as opções possíveis de lances garantindo que a soma total dos megawatts não exceda a quantidade e buscando a combinação que maximize o valor obtido. Para evitar que a execução se prolongue indefinidamente, há um limite de tempo de 30 segundos para a busca.

Parâmetros:

- *lances*: Uma lista de tuplas, onde cada tupla representa um lance e contém dois elementos: a quantidade de recursos (mw) e o valor do lance (dinheiros).
- *energia_disponivel*: A quantidade total de recursos disponíveis para alocação.

Retorna:

- O valor total máximo que pode ser obtido com a alocação ótima dos recursos disponíveis.

b) Solução com Algoritmo Guloso

Na abordagem gulosa, foram utilizadas duas estratégias diferentes: escolher os primeiros lances com maior valor e escolher os primeiros lances com o maior valor por megawatt. O objetivo é maximizar o valor total sem ultrapassar a energia disponível.

Guloso pelo maior valor:

Ordena os lances em ordem decrescente de valor e, em seguida, seleciona os lances até que o limite de energia disponível seja alcançado ou excedido.

Parâmetros:

- *lances*: Uma lista de tuplas, onde cada tupla representa um lance e contém dois elementos: a quantidade de energia (lote) e o valor do lance.
- *energia_disponivel*: A quantidade de energia disponível para ser alocada.

Retorna:

- O valor total alcançado pela seleção dos lances dentro do limite de energia disponível.

Guloso pelo maior valor por megawatt:

Ordena os lances em ordem decrescente de valor por unidade de energia (valor/megawatt) e, em seguida, seleciona os lances até que o limite de energia disponível seja alcançado ou excedido.

Parâmetros:

- *lances (list of tuples)*: Uma lista de tuplas, onde cada tupla representa um lance e contém dois elementos: a quantidade de energia (lote) e o valor do lance.
- *energia_disponivel (int/float)*: A quantidade de energia disponível para ser alocada.

Retorna:

- *int/float*: O valor total alcançado pela seleção dos lances dentro do limite de energia disponível.

c) Solução com Divisão e Conquista

A estratégia de divisão e conquista foi dividir uma lista de lances em duas menores, resolver independentemente e encontrar a melhor combinação entre as soluções das metades.

Parâmetros:

- *lances*: Uma lista de tuplas, onde cada tupla representa um lance e contém dois elementos: a quantidade de energia (em megawatts) e o valor do lance (em unidades monetárias).

- *energia_disponivel*: A quantidade de energia disponível para ser alocada (em megawatts).

- *memo*: Um dicionário usado para armazenar os resultados de subproblemas já calculados.

Padrão é None, o que significa que nenhum resultado de subproblema é inicialmente armazenado.

d) Solução com Programação Dinâmica

Na programação dinâmica, foi utilizado uma abordagem semelhante ao problema da mochila, onde é utilizado uma tabela para armazenar os valores máximos possíveis para diferentes capacidades de energia, as linhas representam os lances e as colunas a energia disponível.

Resultados

1) Considerando 25 empresas para o primeiro e segundo conjunto

Resultados Laura e Pedro respectivamente

Resumo dos Resultados:

Conjunto 1:

Guloso (Maior Valor) => Resultado: 26725, Tempo: 0.02s
Guloso (Maior Valor por MW) => Resultado: 26725, Tempo: 0.01s
Divisão e Conquista => Resultado: 7480, Tempo: 0.00s
Programação Dinâmica => Resultado: 26725, Tempo: 0.03s
Backtracking => Resultado: 26725, Tempo: 13.76s

Conjunto 2:

Guloso (Maior Valor) => Resultado: 38673, Tempo: 0.00s
Guloso (Maior Valor por MW) => Resultado: 39271, Tempo: 0.01s
Divisão e Conquista => Resultado: 13899, Tempo: 0.00s
Programação Dinâmica => Resultado: 40348, Tempo: 0.03s
Backtracking => Resultado: 40348, Tempo: 13.70s

Resumo dos Resultados:

Conjunto 1:

Guloso (Maior Valor) => Resultado: 26725, Tempo: 0.00s
Guloso (Maior Valor por MW) => Resultado: 26725, Tempo: 0.01s
Divisão e Conquista => Resultado: 7480, Tempo: 0.00s
Programação Dinâmica => Resultado: 26725, Tempo: 0.03s
Backtracking => Resultado: 26725, Tempo: 14.32s

Conjunto 2:

Guloso (Maior Valor) => Resultado: 38673, Tempo: 0.00s
Guloso (Maior Valor por MW) => Resultado: 39271, Tempo: 0.01s
Divisão e Conquista => Resultado: 13899, Tempo: 0.00s
Programação Dinâmica => Resultado: 40348, Tempo: 0.03s
Backtracking => Resultado: 40348, Tempo: 14.06s

2) Considerando 25 empresas para o primeiro e 50 empresas para o segundo conjunto

Resumo dos Resultados:

Conjunto 1:

Guloso (Maior Valor) => Resultado: 26725, Tempo: 0.01s
Guloso (Maior Valor por Mw) => Resultado: 26725, Tempo: 0.01s
Divisão e Conquista => Resultado: 7480, Tempo: 0.00s
Programação Dinâmica => Resultado: 26725, Tempo: 0.03s
Backtracking => Resultado: 26725, Tempo: 14.06s

Conjunto 2:

Guloso (Maior Valor) => Resultado: 37755, Tempo: 0.00s
Guloso (Maior Valor por Mw) => Resultado: 39271, Tempo: 0.02s
Divisão e Conquista => Resultado: 12469, Tempo: 0.00s
Programação Dinâmica => Resultado: 40348, Tempo: 0.06s
Backtracking => Resultado: 40348, Tempo: 30.00s

3) Considerando 50 empresas para o primeiro e 25 empresas para o segundo conjunto

Resumo dos Resultados:

Conjunto 1:

Guloso (Maior Valor) => Resultado: 37755, Tempo: 0.01s
Guloso (Maior Valor por Mw) => Resultado: 39271, Tempo: 0.01s
Divisão e Conquista => Resultado: 12360, Tempo: 0.01s
Programação Dinâmica => Resultado: 40348, Tempo: 0.06s
Backtracking => Resultado: 40348, Tempo: 30.01s

Conjunto 2:

Guloso (Maior Valor) => Resultado: 26725, Tempo: 0.00s
Guloso (Maior Valor por Mw) => Resultado: 26725, Tempo: 0.00s
Divisão e Conquista => Resultado: 7586, Tempo: 0.00s
Programação Dinâmica => Resultado: 26725, Tempo: 0.05s
Backtracking => Resultado: 26725, Tempo: 7.21s