

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Graduação em Engenharia de Software

André Fernandez Mendes
Leonardo Augusto Pereira do Carmo

RELATÓRIO DO TRABALHO PRÁTICO

Belo Horizonte
2024

André Fernandez Mendes
Leonardo Augusto Pereira do Carmo

RELATÓRIO DO TRABALHO PRÁTICO

Trabalho apresentado à disciplina
Fundamentos de Projeto e Análise de
Algoritmos da Graduação de Engenharia
de Software da Pontifícia Universidade
Católica de Minas Gerais

Professor: João Caram

Belo Horizonte
2024

RESUMO

Este trabalho trata do estudo de problemas intratáveis, tipicamente pertencente às classes NP, e nas técnicas de projeto de algoritmos que podem nos ajudar a encontrar soluções de compromisso adequadas.

São duas as tarefas do grupo:

- Projetar e implementar uma solução para o problema apresentado utilizando backtracking. A solução deve incluir uma estratégia de poda para soluções não promissoras. Gerar conjuntos de teste de tamanho crescente, a partir de 10 interessadas e incrementando de 1 em 1, até atingirmos tamanho T que não consiga ser resolvido em até 30 segundos pelo algoritmo. Na busca do tempo limite de 30 segundos, faça o teste com 10 conjuntos de cada tamanho, contabilizando a média das execuções.
- Projetar e implementar uma solução para o problema apresentado utilizando programação dinâmica. O grupo deverá decidir se vai utilizar o método demonstrado em aula ou outro à escolha. Para este teste, utilize os mesmos conjuntos de tamanho T encontrados no backtracking. Em seguida, aumente os tamanhos dos conjuntos de T em T até atingir o tamanho 10T, sempre executando 10 testes de cada tamanho para utilizar a média

SUMÁRIO

Nenhuma entrada de sumário foi encontrada.

1. SOLUÇÃO DO PROBLEMA UTILIZANDO BACKTRACKING

O backtracking é uma técnica de projeto de algoritmos que consiste no refinamento da busca exaustiva/ força bruta, pois não testa todas as soluções, eliminando algumas sem examinar. A técnica consiste em realizar uma busca em profundidade e, quando a busca falha ou não tem como continuar, retorna pelo mesmo caminho percorrido com a finalidade de encontrar soluções alternativas.

Para aplicação da técnica em questão, é necessário a implementação de uma estrutura de dados de controle.

Conforme estabelecido no problema, foram cadastrados ...

A estratégia de poda utilizada consiste na comparação dos valores obtidos em cada combinação de venda, podando quando a soma dos megawatts disponíveis ultrapassa a quantidade existente.

Assim, evita-se a venda o cálculo de combinações que ultrapassam a quantidade de megawatts que pode ser vendida.

Após medir o tempo de execução de conjuntos de tamanho crescente, iniciando com 10 interessadas e aumentando o número de interessadas até que o tempo médio do algoritmo excedesse 30 segundos, chegou-se a conclusão que podem ser combinadas

2. SOLUÇÃO DO PROBLEMA UTILIZANDO PROGRAMAÇÃO DINÂMICA

Modelagem do Problema:

- O problema foi modelado de forma semelhante ao problema clássico da mochila, onde cada empresa representa um item, a quantidade de megawatts representa o peso do item, e o valor da oferta representa o valor do item.

Estrutura de Dados:

- Utilizamos um array dp de tamanho capacidade + 1 para armazenar o valor máximo que pode ser obtido para cada capacidade de megawatts, onde capacidade é o total de megawatts disponíveis para venda.
- Utilizamos uma matriz taken para rastrear quais empresas foram selecionadas na solução ótima.

Atualização dos Estados:

- Para cada empresa, atualizamos o array dp de forma que dp[j] represente o valor máximo obtido ao considerar a inclusão ou exclusão da oferta da empresa atual.

FUNCIONAMENTO DO ALGORITMO

Inicialização:

- Inicializamos o array dp com zeros, representando o valor inicial para cada capacidade de megawatts.

Processamento das Empresas:

- Para cada empresa, percorremos o array dp de trás para frente (para evitar a reutilização dos mesmos itens na mesma iteração) e atualizamos o valor máximo que pode ser obtido para cada capacidade de megawatts.

Rastreamento das Empresas Selecionadas:

- Após preencher o array dp, utilizamos a matriz taken para rastrear quais empresas foram selecionadas na solução ótima.

Construção da Solução Ótima:

- Percorremos a matriz taken para construir a lista de empresas selecionadas e calcular o valor total obtido.

Comparação de Resultados

Tempo de Execução

A programação dinâmica se mostrou eficiente em termos de tempo de execução, especialmente quando comparada ao backtracking. Ao evitar a exploração de soluções não promissoras e utilizando subproblemas já resolvidos, a programação dinâmica reduziu significativamente o tempo necessário para encontrar a solução ótima.

Qualidade do Resultado

A qualidade do resultado obtido pela programação dinâmica foi excelente, atingindo o valor máximo possível de forma consistente. A matriz taken garantiu que a solução encontrada fosse rastreável e verificável, o que é uma vantagem em termos de clareza e confiabilidade.

Avaliando os Resultados:

Colocando o Primeiro Grupo de Interessados:

E1;430;1043

E2;428;1188

E3;410;1565

E4;385;1333

E5;399;1214

E6;382;1498

E7;416;1540

E8;436;1172

E9;416;1386

E10;423;1097

E11;400;1463

E12;406;1353

E13;403;1568

E14;390;1228

E15;387;1542

E16;390;1206

E17;430;1175

E18;397;1492

E19;392;1293

E20;393;1533

E21;439;1149

E22;403;1277

E23;415;1624

E24;387;1280

E25;417;1330

Conseguimos os seguintes resultados:

O valor máximo que pode ser obtido é: 26725

Empresas selecionadas:

Empresa: E25, Megawatts: 417, Valor: 1330

Empresa: E24, Megawatts: 387, Valor: 1280

Empresa: E23, Megawatts: 415, Valor: 1624

Empresa: E22, Megawatts: 403, Valor: 1277

Empresa: E20, Megawatts: 393, Valor: 1533

Empresa: E19, Megawatts: 392, Valor: 1293

Empresa: E18, Megawatts: 397, Valor: 1492

Empresa: E16, Megawatts: 390, Valor: 1206

Empresa: E15, Megawatts: 387, Valor: 1542

Empresa: E14, Megawatts: 390, Valor: 1228

Empresa: E13, Megawatts: 403, Valor: 1568

Empresa: E12, Megawatts: 406, Valor: 1353

Empresa: E11, Megawatts: 400, Valor: 1463

Empresa: E9, Megawatts: 416, Valor: 1386

Empresa: E7, Megawatts: 416, Valor: 1540

Empresa: E6, Megawatts: 382, Valor: 1498

Empresa: E5, Megawatts: 399, Valor: 1214

Empresa: E4, Megawatts: 385, Valor: 1333

Empresa: E3, Megawatts: 410, Valor: 1565

Valor total obtido em dinheiro: 26725

Tempo gasto para o cálculo: 0.0077131 segundos

Agora usando o segundo conjunto de interessados:

E1;313;1496
E2;398;1768
E3;240;1210
E4;433;2327
E5;301;1263
E6;297;1499
E7;232;1209
E8;614;2342
E9;558;2983
E10;495;2259
E11;310;1381
E12;213;961
E13;213;1115
E14;346;1552
E15;385;2023
E16;240;1234
E17;483;2828
E18;487;2617
E19;709;2328
E20;358;1847
E21;467;2038
E22;363;2007
E23;279;1311
E24;589;3164
E25;476;2480

Obtemos o seguinte resultado:

O valor máximo que pode ser obtido é: 40348

Empresas selecionadas:

Empresa: E25, Megawatts: 476, Valor: 2480

Empresa: E24, Megawatts: 589, Valor: 3164

Empresa: E23, Megawatts: 279, Valor: 1311

Empresa: E22, Megawatts: 363, Valor: 2007

Empresa: E21, Megawatts: 467, Valor: 2038

Empresa: E20, Megawatts: 358, Valor: 1847

Empresa: E18, Megawatts: 487, Valor: 2617

Empresa: E17, Megawatts: 483, Valor: 2828

Empresa: E16, Megawatts: 240, Valor: 1234

Empresa: E15, Megawatts: 385, Valor: 2023

Empresa: E14, Megawatts: 346, Valor: 1552

Empresa: E13, Megawatts: 213, Valor: 1115

Empresa: E11, Megawatts: 310, Valor: 1381

Empresa: E10, Megawatts: 495, Valor: 2259

Empresa: E9, Megawatts: 558, Valor: 2983

Empresa: E7, Megawatts: 232, Valor: 1209

Empresa: E6, Megawatts: 297, Valor: 1499

Empresa: E4, Megawatts: 433, Valor: 2327

Empresa: E3, Megawatts: 240, Valor: 1210

Empresa: E2, Megawatts: 398, Valor: 1768

Empresa: E1, Megawatts: 313, Valor: 1496

Empresa: E1, Megawatts: 313, Valor: 1496

Valor total obtido em dinheiro: 40348

Tempo gasto para o cálculo: 0.0074129 segundos

(O console do Eclipse não coube tudo em uma foto só, então dividi em duas).

Agora eu deveria utilizar os mesmos conjuntos de testes utilizados no Algoritmo Guloso, porém como meu grupo não fez o Guloso, vou considerar o $T = 29$. Que foi mais ou menos a margem registrada pelos outros grupos que apresentaram:

T1 – 29: 0.0072782 segundos

T2 – 58: 0.0089673 segundos

T3 – 87: 0.0094695 segundos

T4 – 116: 0.0090344 segundos

T5 – 145: 0.0101989 segundos

T6 – 174: 0.0091016 segundos

T7 – 203: 0.0124749 segundos
T8 – 232: 0.011177 segundos
T9 – 261: 0.0111038 segundos
T10 – 290: 0.0119611 segundos

Considerações Finais

A programação dinâmica provou ser uma técnica eficaz para resolver o problema de venda de energia, oferecendo um balanço ideal entre tempo de execução e qualidade do resultado. A implementação cuidadosa das estruturas de dados e a abordagem sistemática na atualização dos estados garantiram que a solução fosse tanto eficiente quanto precisa.