



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

## Engenharia de Software

Fundamentos de Projeto e Análise de Algoritmos

Prof(a): João Caram Santos de Oliveira

Bernardo Oliveira Pires  
Nathan Gonçalves de Oliveira  
Vitor de Souza Xavier

## Problema

Na segunda parte da disciplina, estamos focados no estudo de problemas intratáveis, tipicamente pertencentes às classes NP, e nas técnicas de projeto de algoritmos que podem nos ajudar a encontrar soluções de compromisso adequadas.

“Uma empresa produtora de energia possui uma quantidade  $X$  de energia, medida em megawatts, para vender. Seu objetivo é vender sua energia produzida, obtendo o maior valor possível no conjunto de suas vendas. As vendas serão realizadas por leilão: cada empresa interessada dará um lance por um lote de  $K$  megawatts, oferecendo um valor  $V$  por este lote. As interessadas só comprarão um lote do tamanho exato da oferta.

## Solução

O grupo decidiu que cada membro implementaria uma solução:

Nathan Oliveira	Programação Dinâmica
Vitor Xavier	Backtracking
Bernardo Oliveira	Algoritmo Guloso

Na geração de lances fornecida pelo professor, os valores representados abaixo foram pré-definidos para início da solução do problema. Para resolução do problema é considerada a existência de 2 conjuntos de testes apresentados abaixo.

\* Conjunto de empresas interessadas 1  
(nome, quantidade, valor):

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

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

## Backtracking

A função de backtracking foi implementada para maximizar o lucro na venda de energia para empresas interessadas, garantindo que a quantidade de energia disponível não seja excedida.

O backtracking busca realizar uma exploração sistemática de todas as opções possíveis. A função de backtracking gera soluções parciais incrementais, tentando incluir ou excluir cada lance de uma empresa. Claro que, são realizados testes para verificar se a solução parcial é viável ou se pode ser descartada. Se caso a solução parcial não atender aos critérios é realizada a poda, ou seja, o algoritmo abandona esse ramo da busca. A cada vez que uma solução completa é alcançada, o algoritmo verifica se essa solução é a melhor que a melhor solução encontrada até agora.

O critério de poda funciona da seguinte maneira:

- A poda recebe a lista de lances, a seleção atual de lances, o índice atual, a quantidade disponível de energia e o lucro atual;
- Calcula o lucro total da seleção atual de lances;
- Verifica se a seleção parcial já é pior que a melhor solução encontrada, realizando a poda:
  - Se a quantidade de energia necessária para o lance atual for maior do que a quantidade disponível, a seleção é descartada.
  - Retorna true se a poda for válida, indicando que o ramo da árvore de busca pode ser podado.

```

Melhor lucro: 26725
Energia sobrando: 412 MW
Lances selecionados:
ID: E3, Quantidade: 410, Valor: 1565
ID: E4, Quantidade: 385, Valor: 1333
ID: E5, Quantidade: 399, Valor: 1214
ID: E6, Quantidade: 382, Valor: 1498
ID: E7, Quantidade: 416, Valor: 1540
ID: E9, Quantidade: 416, Valor: 1386
ID: E11, Quantidade: 400, Valor: 1463
ID: E12, Quantidade: 406, Valor: 1353
ID: E13, Quantidade: 403, Valor: 1568
ID: E14, Quantidade: 390, Valor: 1228
ID: E15, Quantidade: 387, Valor: 1542
ID: E16, Quantidade: 390, Valor: 1206
ID: E18, Quantidade: 397, Valor: 1492
ID: E19, Quantidade: 392, Valor: 1293
ID: E20, Quantidade: 393, Valor: 1533
ID: E22, Quantidade: 403, Valor: 1277
ID: E23, Quantidade: 415, Valor: 1624
ID: E24, Quantidade: 387, Valor: 1280
ID: E25, Quantidade: 417, Valor: 1330
Tempo de execução: 0.3885056 segundos

```

Solução caso de teste 1

```

Melhor lucro: 40348
Energia sobrando: 38 MW
Lances selecionados:
ID: E1, Quantidade: 313, Valor: 1496
ID: E2, Quantidade: 398, Valor: 1768
ID: E3, Quantidade: 240, Valor: 1210
ID: E4, Quantidade: 433, Valor: 2327
ID: E6, Quantidade: 297, Valor: 1499
ID: E7, Quantidade: 232, Valor: 1209
ID: E9, Quantidade: 558, Valor: 2983
ID: E10, Quantidade: 495, Valor: 2259
ID: E11, Quantidade: 310, Valor: 1381
ID: E13, Quantidade: 213, Valor: 1115
ID: E14, Quantidade: 346, Valor: 1552
ID: E15, Quantidade: 385, Valor: 2023
ID: E16, Quantidade: 240, Valor: 1234
ID: E17, Quantidade: 483, Valor: 2828
ID: E18, Quantidade: 487, Valor: 2617
ID: E20, Quantidade: 358, Valor: 1847
ID: E21, Quantidade: 467, Valor: 2038
ID: E22, Quantidade: 363, Valor: 2007
ID: E23, Quantidade: 279, Valor: 1311
ID: E24, Quantidade: 589, Valor: 3164
ID: E25, Quantidade: 476, Valor: 2480
Tempo de execução: 0.4799949 segundos

```

Solução caso de teste 2

No primeiro caso, o algoritmo conseguiu um lucro de 26725, com 412 MW de energia sobrando. O tempo de execução foi de aproximadamente 0.39 segundos. A quantidade significativa de energia sobrando sugere que a seleção de lances não foi totalmente eficiente em maximizar a utilização da energia disponível.

No segundo caso, o algoritmo obteve um lucro maior de 40348, utilizando quase toda a energia disponível, com apenas 38 MW sobrando. O tempo de execução foi

de cerca de 0.48 segundos. Este resultado indica uma combinação de lances mais eficiente, maximizando o lucro e utilizando quase toda a energia disponível.

O algoritmo de backtracking está funcionando corretamente e é eficiente para maximizar o lucro na venda de energia, mesmo com múltiplos lances. Ele é capaz de encontrar a melhor combinação de lances rapidamente, garantindo bons resultados e desempenho consistente. Note que ambos os casos mostraram tempos de execução rápidos, abaixo de 0.5 segundos, demonstrando a eficiência do algoritmo em encontrar a melhor combinação de lances.

## Algoritmo Guloso

O método guloso foi implementado com o objetivo de obter uma solução eficiente para a venda de energia para empresas, dentro da capacidade total de energia disponível. Este método é conhecido por sua capacidade de fornecer soluções rápidas e razoavelmente boas, embora não garanta o resultado ótimo global.

O método guloso opera com uma abordagem “miopia”, onde faz escolhas baseadas em critérios de seleção que parecem melhores no momento, sem considerar o efeito completo dessas escolhas no resultado final.

### Estratégias/Critério de Implementação:

- **Guloso por Maior Valor:** Prioriza as ofertas com os maiores valores monetários, ordenando todas as ofertas disponíveis pelo valor e as selecionando até que o limite de energia seja alcançado ou esteja próximo.
- **Guloso por Melhor Custo/Benefício:** Foca em ofertas com a melhor relação valor por megawatt, ordenando as ofertas para maximizar o retorno por unidade de energia consumida.

### Resultados:

Guloso por maior valor lista de testes 1:

```
Guloso por maior valor
Selecionando oferta: E23: 415 MW, 1624 dinheiros
Selecionando oferta: E13: 403 MW, 1568 dinheiros
Selecionando oferta: E3: 410 MW, 1565 dinheiros
Selecionando oferta: E15: 387 MW, 1542 dinheiros
Selecionando oferta: E7: 416 MW, 1540 dinheiros
Selecionando oferta: E20: 393 MW, 1533 dinheiros
Selecionando oferta: E6: 382 MW, 1498 dinheiros
Selecionando oferta: E18: 397 MW, 1492 dinheiros
Selecionando oferta: E11: 400 MW, 1463 dinheiros
Selecionando oferta: E9: 416 MW, 1386 dinheiros
Selecionando oferta: E12: 406 MW, 1353 dinheiros
Selecionando oferta: E4: 385 MW, 1333 dinheiros
Selecionando oferta: E25: 417 MW, 1330 dinheiros
Selecionando oferta: E19: 392 MW, 1293 dinheiros
Selecionando oferta: E24: 387 MW, 1280 dinheiros
Selecionando oferta: E22: 403 MW, 1277 dinheiros
Selecionando oferta: E14: 390 MW, 1228 dinheiros
Selecionando oferta: E5: 399 MW, 1214 dinheiros
Selecionando oferta: E16: 390 MW, 1206 dinheiros
Valor máximo obtido: 26725
```

Guloso por Melhor Custo/Benefício lista de teste 1:



Guloso por Valor/Megawatt

Selecionando oferta (VM):	E15:	387 MW,	1542 dinheiros
Selecionando oferta (VM):	E6:	382 MW,	1498 dinheiros
Selecionando oferta (VM):	E23:	415 MW,	1624 dinheiros
Selecionando oferta (VM):	E20:	393 MW,	1533 dinheiros
Selecionando oferta (VM):	E13:	403 MW,	1568 dinheiros
Selecionando oferta (VM):	E3:	410 MW,	1565 dinheiros
Selecionando oferta (VM):	E18:	397 MW,	1492 dinheiros
Selecionando oferta (VM):	E7:	416 MW,	1540 dinheiros
Selecionando oferta (VM):	E11:	400 MW,	1463 dinheiros
Selecionando oferta (VM):	E4:	385 MW,	1333 dinheiros
Selecionando oferta (VM):	E12:	406 MW,	1353 dinheiros
Selecionando oferta (VM):	E9:	416 MW,	1386 dinheiros
Selecionando oferta (VM):	E24:	387 MW,	1280 dinheiros
Selecionando oferta (VM):	E19:	392 MW,	1293 dinheiros
Selecionando oferta (VM):	E25:	417 MW,	1330 dinheiros
Selecionando oferta (VM):	E22:	403 MW,	1277 dinheiros
Selecionando oferta (VM):	E14:	390 MW,	1228 dinheiros
Selecionando oferta (VM):	E16:	390 MW,	1206 dinheiros
Selecionando oferta (VM):	E5:	399 MW,	1214 dinheiros

Valor máximo obtido por Valor/Megawatt: 26725

Guloso por maior valor lista de testes 2:

Guloso por maior valor

Selecionando oferta: E24: 589 MW, 3164 dinheiros

Selecionando oferta: E9: 558 MW, 2983 dinheiros

Selecionando oferta: E17: 483 MW, 2828 dinheiros

Selecionando oferta: E18: 487 MW, 2617 dinheiros

Selecionando oferta: E25: 476 MW, 2480 dinheiros

Selecionando oferta: E8: 614 MW, 2342 dinheiros

Selecionando oferta: E19: 709 MW, 2328 dinheiros

Selecionando oferta: E4: 433 MW, 2327 dinheiros

Selecionando oferta: E10: 495 MW, 2259 dinheiros

Selecionando oferta: E21: 467 MW, 2038 dinheiros

Selecionando oferta: E15: 385 MW, 2023 dinheiros

Selecionando oferta: E22: 363 MW, 2007 dinheiros

Selecionando oferta: E20: 358 MW, 1847 dinheiros

Selecionando oferta: E2: 398 MW, 1768 dinheiros

Selecionando oferta: E14: 346 MW, 1552 dinheiros

Selecionando oferta: E6: 297 MW, 1499 dinheiros

Selecionando oferta: E1: 313 MW, 1496 dinheiros

Selecionando oferta: E13: 213 MW, 1115 dinheiros

Valor máximo obtido: 38673

Guloso por Melhor Custo/Benefício lista de teste 2:

```
Guloso por Valor/Megawatt
Selecionando oferta (VM): E17: 483 MW, 2828 dinheiros
Selecionando oferta (VM): E22: 363 MW, 2007 dinheiros
Selecionando oferta (VM): E4: 433 MW, 2327 dinheiros
Selecionando oferta (VM): E18: 487 MW, 2617 dinheiros
Selecionando oferta (VM): E24: 589 MW, 3164 dinheiros
Selecionando oferta (VM): E9: 558 MW, 2983 dinheiros
Selecionando oferta (VM): E15: 385 MW, 2023 dinheiros
Selecionando oferta (VM): E13: 213 MW, 1115 dinheiros
Selecionando oferta (VM): E7: 232 MW, 1209 dinheiros
Selecionando oferta (VM): E25: 476 MW, 2480 dinheiros
Selecionando oferta (VM): E20: 358 MW, 1847 dinheiros
Selecionando oferta (VM): E16: 240 MW, 1234 dinheiros
Selecionando oferta (VM): E6: 297 MW, 1499 dinheiros
Selecionando oferta (VM): E3: 240 MW, 1210 dinheiros
Selecionando oferta (VM): E1: 313 MW, 1496 dinheiros
Selecionando oferta (VM): E23: 279 MW, 1311 dinheiros
Selecionando oferta (VM): E10: 495 MW, 2259 dinheiros
Selecionando oferta (VM): E12: 213 MW, 961 dinheiros
Selecionando oferta (VM): E14: 346 MW, 1552 dinheiros
Selecionando oferta (VM): E11: 310 MW, 1381 dinheiros
Selecionando oferta (VM): E2: 398 MW, 1768 dinheiros
Valor máximo obtido por Valor/Megawatt: 39271
```

## Programação Dinâmica

A Programação Dinâmica divide os problemas em subproblemas menores, realizando os cálculos apenas uma vez, para evitar redundância e diminuir o tempo de execução do algoritmo, encontrando assim, a melhor solução final.

O algoritmo de Programação Dinâmica para solucionar o problema do leilão de energia, foi pensado para retornar o melhor lucro na venda de energia. O algoritmo recebe do usuário a quantidade de energia disponível, em MW e quantidade de lances. A partir daí, o usuário deve preencher o ID da empresa, a quantidade de energia e o valor de cada lance.

Após digitar todas as informações, o código retorna o melhor lucro, com a quantidade de energia que sobrou, a lista com os lances selecionados e o tempo de execução. Semelhantemente ao problema da mochila, o algoritmo utiliza uma tabela para armazenar os valores máximos para as capacidades de energia. As linhas representam os lances e as colunas as quantidades de energia.



O código abaixo mostra o preenchimento da tabela dinâmica, sendo “i” a as linhas, e “j” as colunas:

```
for (int i = 1; i <= n; i++) {  
  
    for (int w = 0; w <= quantidadeDisponivel; w++) {  
  
        EmpresaInteressada lance = lances.get(i - 1);  
  
        if (lance.getQuantidade() <= w) {  
  
            if (lance.getValor() + dp[i - 1][w - lance.getQuantidade()] > dp[i - 1][w]) {  
  
                dp[i][w] = lance.getValor() + dp[i - 1][w - lance.getQuantidade()];  
  
                solucao[i][w] = true;  
  
            } else {  
  
                dp[i][w] = dp[i - 1][w];  
  
            }  
  
        } else {  
  
            dp[i][w] = dp[i - 1][w];  
  
        }  
  
    }  
  
}
```

Melhor lucro: 26725  
Energia sobrando: 412 MW  
Lances selecionados:  
ID: E25, Quantidade: 417, Valor: 1330  
ID: E24, Quantidade: 387, Valor: 1280  
ID: E23, Quantidade: 415, Valor: 1624  
ID: E22, Quantidade: 403, Valor: 1277  
ID: E20, Quantidade: 393, Valor: 1533  
ID: E19, Quantidade: 392, Valor: 1293  
ID: E18, Quantidade: 397, Valor: 1492  
ID: E16, Quantidade: 390, Valor: 1206  
ID: E15, Quantidade: 387, Valor: 1542  
ID: E14, Quantidade: 390, Valor: 1228  
ID: E13, Quantidade: 403, Valor: 1568  
ID: E12, Quantidade: 406, Valor: 1353  
ID: E11, Quantidade: 400, Valor: 1463  
ID: E9, Quantidade: 416, Valor: 1386  
ID: E7, Quantidade: 416, Valor: 1540  
ID: E6, Quantidade: 382, Valor: 1498  
ID: E5, Quantidade: 399, Valor: 1214  
ID: E4, Quantidade: 385, Valor: 1333  
ID: E3, Quantidade: 410, Valor: 1565  
Tempo total de execução: 0.0267018 s

Melhor lucro: 40348  
Energia sobrando: 38 MW  
Lances selecionados:  
ID: E25, Quantidade: 476, Valor: 2480  
ID: E24, Quantidade: 589, Valor: 3164  
ID: E23, Quantidade: 279, Valor: 1311  
ID: E22, Quantidade: 363, Valor: 2007  
ID: E21, Quantidade: 467, Valor: 2038  
ID: E20, Quantidade: 358, Valor: 1847  
ID: E18, Quantidade: 487, Valor: 2617  
ID: E17, Quantidade: 483, Valor: 2828  
ID: E16, Quantidade: 240, Valor: 1234  
ID: E15, Quantidade: 385, Valor: 2023  
ID: E14, Quantidade: 346, Valor: 1552  
ID: E13, Quantidade: 213, Valor: 1115  
ID: E11, Quantidade: 310, Valor: 1381  
ID: E10, Quantidade: 495, Valor: 2259  
ID: E9, Quantidade: 558, Valor: 2983  
ID: E7, Quantidade: 232, Valor: 1209  
ID: E6, Quantidade: 297, Valor: 1499  
ID: E4, Quantidade: 433, Valor: 2327  
ID: E3, Quantidade: 240, Valor: 1210  
ID: E2, Quantidade: 398, Valor: 1768  
ID: E1, Quantidade: 313, Valor: 1496  
Tempo total de execução: 0.026057301 s

Resultado da execução para o primeiro e segundo grupo de empresas, respectivamente.

# Análise geral

Analisando os resultados, os algoritmos de Backtracking e de Programação Dinâmica obtiveram os mesmos resultados (26725 dinheiros/MW e 40348 dinheiros/MW respectivamente). O tempo de execução dos algoritmos foi a única diferença entre os dois métodos. No backtracking, o primeiro grupo de empresas, foi executado em 0,38 segundos, e o segundo em 0,47 segundos. Por outro lado, o de programação dinâmica, foi executado relativamente mais rápido, com 0,0267 segundos para o primeiro grupo de empresas e o segundo, novamente, em 0,0260 segundos. Resultado curioso, já que o segundo grupo de empresas possuíam valores de lances maiores.

Já nos algoritmos gulosos, para o primeiro conjunto de empresas, obtivemos os mesmos valores de lucro que obtivemos nos métodos de backtracking e de programação dinâmica, 26725 dinheiros. Para o segundo grupo das empresas, obtivemos valores diferentes para a execução dos dois métodos do algoritmo guloso. Caso vendêssemos a energia visando o melhor custo benefício, venderíamos uma maior quantidade de energia (39271 dinheiros/MW) do que se fôssemos vender pelo maior valor (38673 dinheiros/MW). Talvez por causa do método de poda escolhido.

Tais resultados são condizentes com a proposta de cada algoritmo, em que o de Programação Dinâmica performou melhor que os outros, de maneira relativamente rápida, e encontrou a solução ótima, enquanto que, o de backtracking, curiosamente, encontrou a solução ótima, mas demorando muito mais tempo do que o de Programação Dinâmica. Já o algoritmo guloso, seria o “pior” método para encontrar uma solução, pelo menos, para valores grandes, apesar de ser relativamente rápido de ser executado.

# Especificações das máquinas utilizadas

**Dono:** Bernardo Oliveira Pires

**Processador:** Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz

**RAM instalada:** 16 GB (utilizável: 15,9 GB)

**Tipo de sistema:** Windows 11 Home Single Language

**Dono:** Nathan Gonçalves de Oliveira

**Processador:** AMD Ryzen 5 5500U 6-Core Processor 2.10 GHz

**RAM instalada:** 8 GB (utilizável: 6,82 GB)

**Tipo de sistema:** Windows 11

**Dono:** Vitor de Souza Xavier

**Processador:** Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz 2.40 GHz

**RAM instalada:** 8 GB RAM

**Tipo de sistema:** Windows 10