

1º Trabalho de Grupo – Desenho de Algoritmos

G73

André Lima – 202008169

Guilherme Almeida – 202006137

Mariana Lobão – 202004260

Descrição do Problema

Uma empresa de distribuição de encomendas pretende melhorar a eficácia das suas entregas, tornando mais eficiente a distribuição de estafetas, maximizando o lucro, entre outros. O nosso objetivo será ir de encontro as essas expectativas, procurando uma solução adequada para os seguintes cenários:

1. Maximizar o número de encomendas entregues, minimizando o número de estafetas selecionados
2. Maximizar o lucro, maximizando também as encomendas entregues por dia
3. Minimizar o tempo médio de entrega das encomendas por dia

Cenário 1 | Formalização

- Variáveis de entrada:

N_E - Número de encomendas

N_C - Número de carrinhas

$VolMax[j]$ - Volume máximo da carrinha j

$PesoMax[j]$ - Peso máximo da carrinha j

$Vol[i]$ - Volume da encomenda i

$Peso[i]$ - Peso da encomenda i

$Tempo[i]$ - Tempo estimado de entrega da encomenda i

$CarrinhaSelecionada[j]$ - 1 se a carrinha j foi selecionada, 0 senão

$Colocada[i, j]$ - 1 se a encomenda i foi colocada na carrinha j , 0 senão

Cenário 1 | Formalização

- Minimizar: $K = \sum_{j=1}^{N_C} CarrinhaSelecionada[j]$
- Sujeito a:
 - $\sum_{i=1}^{N_E} Peso[i] * Colocada[i, j] \leq PesoMax[j], \forall j \in \{1, \dots, N_C\}$
 - $\sum_{i=1}^{N_E} Vol[i] * Colocada[i, j] \leq VolMax[j], \forall j \in \{1, \dots, N_C\}$
 - $\sum_{i=1}^{N_E} Tempo[i] * Colocada[i, j] \leq (8 * 3600), \forall j \in \{1, \dots, N_C\}$
 - $\sum_{j=1}^{N_C} Colocada[i, j] \leq 1, \forall i \in \{1, \dots, N_E\}$

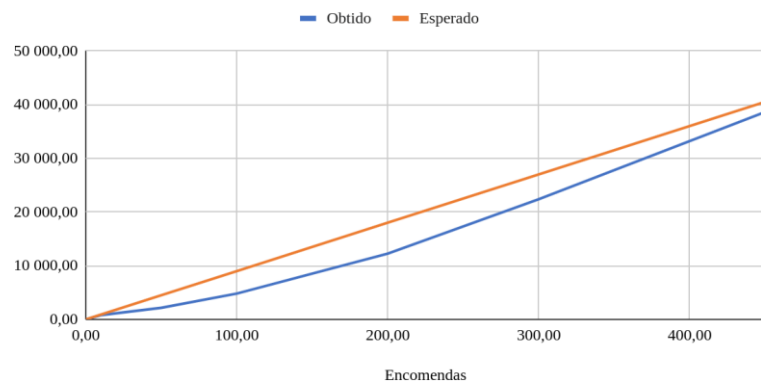
Cenário 1 | Algoritmos Relevantes

- Algoritmo utilizado: Online First Fit Bin-Packing
- Objetivo: Preencher o menor número de carrinhas com o maior número de encomendas
- Prós e contras: Por meio de uma solução ótima local, tenta-se encontrar uma solução ótima geral (algoritmo greedy), o que pode ou não acontecer. No nosso caso, o algoritmo encontra-se entre a solução ótima e 1.7 vezes esse valor, além de ser um algoritmo eficiente em termos temporais.
- Outras opções: Offline Bin-Packing
- Análise da complexidade:
 - Temporal: $O(C * E)$
 - Espacial: $O(C + E)$

Cenário 1 | Avaliação Empírica

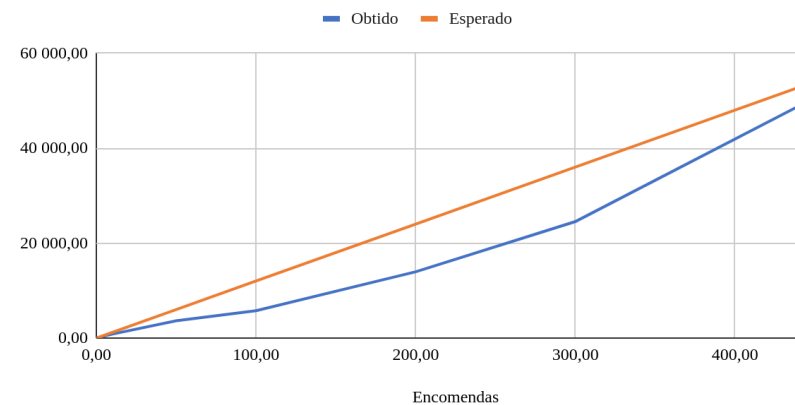
Cenário 1: $O(|C| |E|)$

10 carrinhas



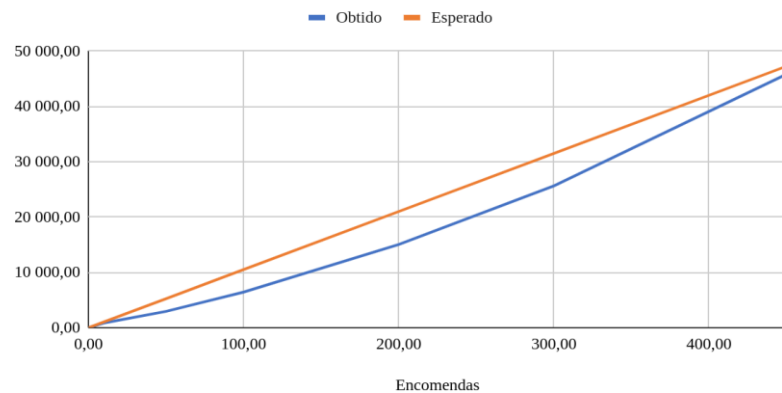
Cenário 1: $O(|C| |E|)$

20 carrinhas



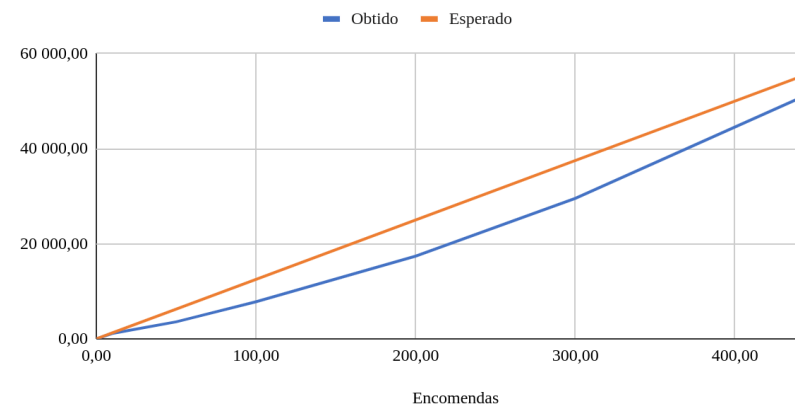
Cenário 1: $O(|C| |E|)$

30 carrinhas



Cenário 1: $O(|C| |E|)$

50 carrinhas



Cenário 2 | Formalização

- Variáveis de entrada:

N_E - Número de encomendas

N_C - Número de carrinhas

$VolMax[j]$ - Volume máximo da carrinha j

$PesoMax[j]$ - Peso máximo da carrinha j

$Vol[i]$ - Volume da encomenda i

$Peso[i]$ - Peso da encomenda i

$Tempo[i]$ - Tempo estimado de entrega da encomenda i

$CarrinhaSelecionada[j]$ - 1 se a carrinha j foi selecionada, 0 senão

$Colocada[i, j]$ - 1 se a encomenda i foi colocada na carrinha j , 0 senão

$Recompensa[i]$ - Recompensa da encomenda i

$Custo[j]$ - Custo da carrinha j

Cenário 2 | Formalização

- Maximizar:

$$R = \sum_{j=1}^{N_C} CarrinhaSelecionada[j] * \left(\left(\sum_{i=1}^{N_E} Colocada[i, j] * Recompensa[i] \right) - Custo[j] \right)$$

- Sujeito a:

$$\sum_{i=1}^{N_E} Peso[i] * Colocada[i, j] \leq PesoMax[j], \forall j \in \{1, \dots, N_C\}$$

$$\sum_{i=1}^{N_E} Vol[i] * Colocada[i, j] \leq VolMax[j], \forall j \in \{1, \dots, N_C\}$$

$$\sum_{i=1}^{N_E} Tempo[i] * Colocada[i, j] \leq (8 * 3600), \forall j \in \{1, \dots, N_C\}$$

$$\sum_{j=1}^{N_C} Colocada[i, j] \leq 1, \forall i \in \{1, \dots, N_E\}$$

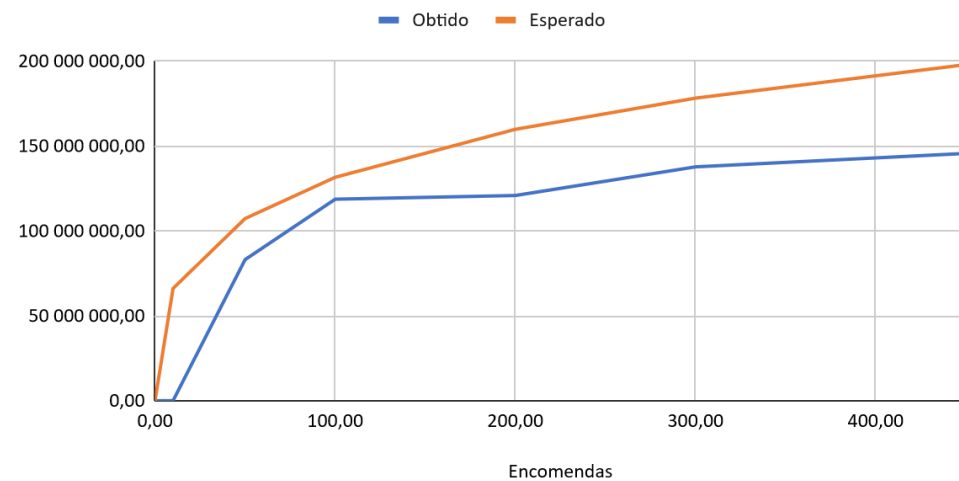
Cenário 2 | Algoritmos Relevantes

- Algoritmo utilizado:
 - Para cada carrinha, selecionar aleatoriamente 20 encomendas.
 - Dessas encomendas, colocar as encomendas na carrinha que maximizam o aumento da recompensa.
 - Repetir o processo até mais nenhuma encomenda for selecionada.
 - Após percorrer todas as carrinhas, para cada encomenda sem carrinha atribuída ou atribuída a uma carrinha com lucro negativo, atribuí-la à carrinha com maior lucro com capacidade para a encomenda.
 - Retirar as carrinhas com lucro não negativo e as respetivas encomendas.
 - Repetir o processo até não serem retiradas nenhuma encomendas em 3 iterações consecutivas
- Objetivo: Maximizar o lucro na entrega das encomendas
- Prós e contras: Algoritmo demora algum tempo a ser executado, no entanto, fornece uma solução aceitável
- Outras opções: First Fit Bin Packing com as carrinhas ordenadas por ordem crescente de custo e as encomendas ordenadas por ordem decrescente de recompensa
- Análise da complexidade:
 - Temporal: $O(\log(C)^2 * \log(E)^2 + C * \log(C))$
 - Espacial: $O(C * E)$

Cenário 2 | Avaliação Empírica

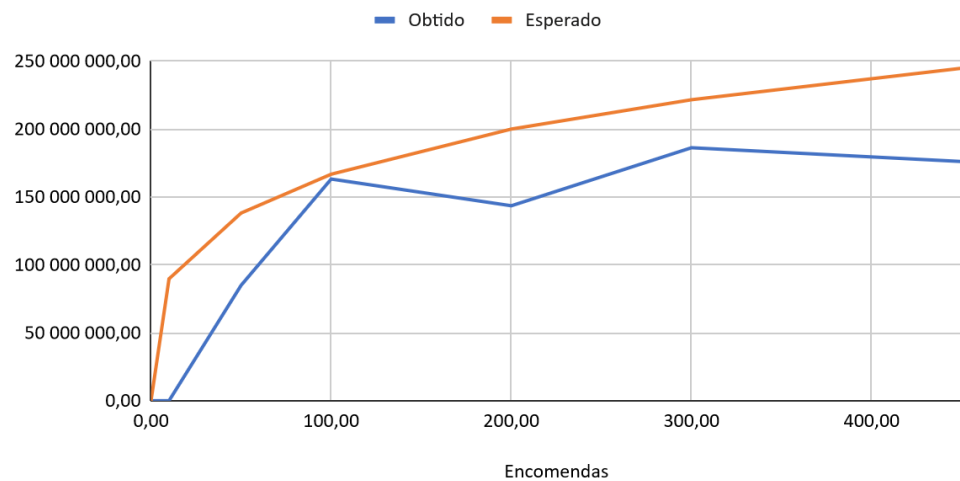
Cenário 2: $O((\log |C|)^2 (\log |E|)^2 + |C| \log |C|)$

30 carrinhas



Cenário 2: $O((\log |C|)^2 (\log |E|)^2 + |C| \log |C|)$

40 carrinhas



Cenário 3 | Formalização

- Variáveis de entrada:

N_E - Número de encomendas

$Tempo[i]$ - Tempo estimado de entrega da encomenda i

$EntregaSelecionada[i]$ - 1 se a encomenda i foi selecionada, 0 senão

Cenário 3 | Formalização

- Minimizar:
$$T = \frac{\sum_{i=1}^{N_E} EntregaSelecionada[i] * i * Tempo[i]}{\sum_{i=1}^{N_E} EntregaSelecionada[i]}$$
- Sujeito a:
$$\sum_{i=1}^{N_E} EntregaSelecionada[i] * Tempo[i] \leq (8 * 3600)$$

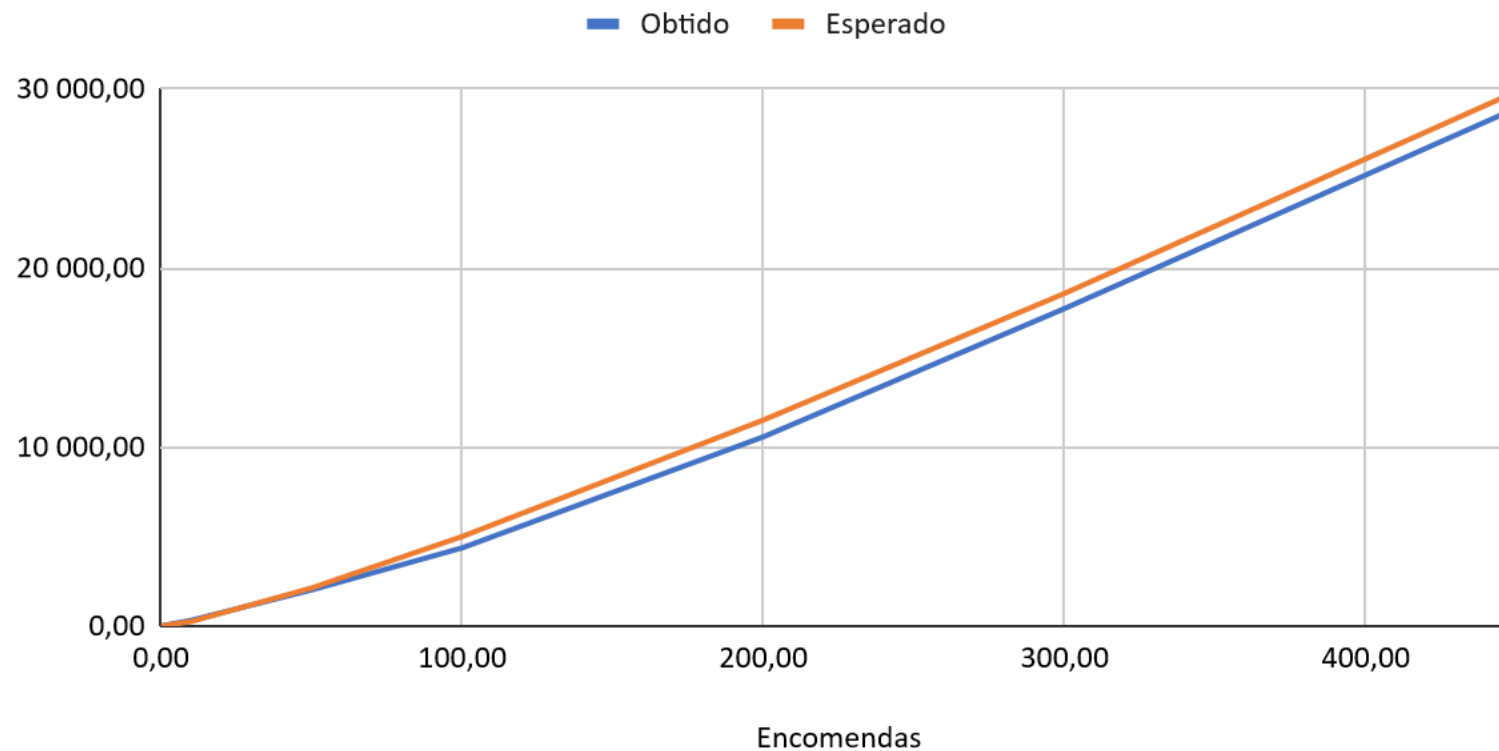
Cenário 3 | Algoritmos Relevantes

- Algoritmo utilizado: Job Scheduling
- Objetivo: Alocar primeiro as encomendas mais rápidas de modo a minimizar o tempo médio de fim de entrega.
- Prós e contras: É um algoritmo simples e rápido, em que basta ordenar as entregas com base no tempo decrescente e alocá-las.
- Análise da complexidade:
 - Temporal: $O(E * \log(E))$
 - Espacial: $O(E)$

Cenário 3 | Avaliação Empírica

Cenário 3: $O(|E| \log |E|)$

50 carrinhas



Funcionalidade Extra

- Medimos a eficiência operacional da empresa com cada algoritmo, igual ao quociente entre o número de pedidos efetivamente entregues e o número de encomendas, para cada dia de trabalho.

Destaque de Funcionalidade

- Conseguimos implementar uma boa solução para o cenário 2, tanto a nível de memória, como a nível de execução temporal, do ponto de vista de uma empresa.

Exemplos de Execução

```
> /home/marinevas/uni/DA/logistics/build/logistics
✓ Please choose the file to read drivers data from: carrinhas.txt
? Please choose the file to read deliveries data from: encomendas.txt
```

Choose one of the following scenarios:

```
[1] [Scenario 1] Deliver the largest number of orders while minimizing the number of drivers used
[2] [Scenario 2] Maximize the profit
[3] [Scenario 3] Maximize the number of express deliveries per day of work
[4] Exit
? Your option [1 - 4]:
```

```
===== Statistics =====
Fulfilled Deliveries      Unfulfilled Deliveries      Total Deliveries
-----
450                        0                            450
Selected Drivers          Unselected Drivers          Total Drivers
-----
24                         26                           50
Efficiency: 100.00%
Press ENTER to continue...

```

```
===== Statistics =====
Fulfilled Deliveries      Unfulfilled Deliveries      Total Deliveries
-----
449                        1                            450
Average ending time: 589.27s
Efficiency: 99.78%
Press ENTER to continue...

```

Dificuldades e Participação

- Dificuldades:
 - Encontrar um algoritmo eficiente e com uma boa aproximação com um consumo de memória aceitável para o cenário 2
- O trabalho foi desenvolvido por todos os elementos, em sessões síncronas, pelo que concordamos que todos demonstramos igual empenho no mesmo