Trabalho 1 - Desenho de Algoritmos

Grupo 11

José Isidro
Matilde Silva
Tomás Torres

Descrição do problema

Neste trabalho, dado um dataset exemplo, foi-nos atribuída a tarefa de otimizar diversos aspectos de uma empresa de logística urbana. Estes problemas, modelados através dos clássicos problemas combinatórios bin packing, 0-1 knapsack, job scheduling, respetivamente, são:

- Cenário 1: otimização do número de estafetas
- Cenário 2: otimização do lucro da empresa
- Cenário 3: otimização das entregas expresso

Cenário 1 - Formalização

Dados:

$P => \{ p1, \dots , pn \}$ - Vetor de n encomendas

$E => \{ e1, \dots , em \}$ - Vetor de m estafetas

<u>Função Objetivo:</u> $Min(\sum em)$

Sujeito a,

$\sum pn_{peso} <= em_{pesomax}$ para $\forall \in m,n$ $\mathbb{N}$ e pn está atribuído a em

$\sum pn_{vol} <= em_{volmax}$ para $\forall \in m,n$ $\mathbb{N}$ e pn está atribuído a em

Cenário 1 - Descrição de Algoritmos relevantes

Ao analisar o problema, a solução a qual conseguimos aplicar e mais ótima foi através da aplicação de *Greedy Algorithms*. Esta abordagem apresentou ser bastante eficiente pois, organizando as estafetas em ordem decrescente e os pedidos em ordem crescente, foi possível distribuí-los visando maximizar a quantidade de pedidos em uma mesma estafeta, consequentemente minimizando a quantidade de estafetas a serem utilizadas.

Cenário 1 - Análise da Complexidade

Neste cenário, são usadas as estruturas *MaxHeap* e *MinHeap* para organizar as encomendas(MinHeap) e os estafetas(MaxHeap), garantindo extrair o valor mínimo e o valor máximo, respetivamente, com complexidades temporais *O(log n)* e complexidades espaciais *O(1).*

O algoritmo contém um ciclo a percorrer os estafetas enquanto os extrai da *MaxHeap* com complexidade temporal, dentro do qual tem outro ciclo que percorre as encomendas, enquanto as extrai da *MinHeap*.

Portanto, para este cenário, a complexidade temporal esperada seria:

$$T(n) = E\ [\ E \log(E) + P \log(P)\ ] \Rightarrow EP \log(P)$$

E a complexidade espacial esperada seria:

$$S(n) = E \log(E)\ [\ O(1) + P \log(P)\ ] \Rightarrow EP \log(P)$$

Cenário 1 - Avaliação Empírica

Após a devida testagem, com os datasets fornecidos, o tempo máximo foi de, aproximadamente, 0,0075s. Este total, quando dividido pelo tamanho do dataset, é exponencialmente pequeno.

O algoritmo conseguiu chegar a uma solução onde, distribuindo todos os 450 pedidos, conseguiu o fazer dentre 27 estafetas, sem sobrepor o peso e o volume dos mesmos, os maximizando.

Cenário 2 - Formalização

Dados:

$P \Rightarrow \{ p_1, \ldots, p_n \}$ - Vetor de pedidos/encomendas

$E \Rightarrow \{ e_1, \ldots, e_m \}$ - Vetor de estafetas

<u>Função Objetivo:</u> $Max(\sum p_{n\,recompensa} - \sum e_{m\,custo})$

Sujeito a,

$\sum p_{n\,peso} <= e_{m\,pesomax}$ para $\forall \in m,n$ $\mathbb{N}$ e pn está atribuído a em

$\sum p_{n\,vol} <= e_{m\,volmax}$ para $\forall \in m,n$ $\mathbb{N}$ e pn está atribuído a em

Cenário 2 - Descrição de Algoritmos relevantes

Ao analisar o problema, a solução que consideramos mais ótima é alcançada através da utilização de *Greedy Algorithms*. Esta abordagem provou ser a mais eficiente, uma vez que para <u>maximizar o lucro total da empresa</u>, será necessário maximizar o lucro numa viagem, isto é, a diferença entre a recompensa total das encomendas e o custo do estafeta. Desta forma, se cada viagem tiver o maior lucro possível, no final do dia, a empresa terá o objetivo cumprido.

Cenário 2 - Análise da Complexidade

Neste cenário, são usadas as estruturas *MaxHeap* e *MinHeap* para organizar as encomendas e os estafetas, garantindo extrair o valor máximo e o valor mínimo, no melhor e no pior caso, respetivamente, com complexidades temporais *O(log n)* e complexidades espaciais *O(1)*.

O algoritmo contém um ciclo a percorrer os estafetas enquanto os extrai da *MinHeap* com complexidade temporal, dentro do qual tem outro ciclo que percorre as encomendas, enquanto as extrai da *MaxHeap*.

Portanto, para este cenário, a complexidade temporal esperada seria:

$$E\,[\,E \log(E) + P \log(P)\,] \Rightarrow EP \log(P)$$

E a complexidade espacial esperada seria:

$$E \log(E)\,[\,O(1) + P \log(P)\,] \Rightarrow EP \log(P)$$

Cenário 2 - Avaliação Empírica

Após a devida testagem, com os datasets fornecidos, o tempo máximo demorado para o programa correr foi ~0.008s. Este total, quando dividido pelo tamanho do dataset, é exponencialmente pequeno.

O algoritmo conseguiu chegar a uma solução ótima com um lucro de 203593, utilizando 16 estafetas e completando 335 entregas.

Cenário 3 - Formalização

Dados:

P => { p1, … , pn } - Vetor de Encomendas Expresso

<u>Função Objetivo:</u> Min(∑ tempoentrega)

Sujeito a,

∑ tempoentrega < 28800 (número de segundos das 9:00 às 17:00)

Cenário 3 - Descrição de Algoritmos relevantes

Após analisado o problema, a solução mais aparente é a utilização de *Greedy Algorithms*. Tal abordagem provaria mais eficiente, visto que para minimizar o tempo médio de entregas, será necessário minimizar o tempo de cada entrega individual feita num dia.

O uso de um *Greedy Algorithm* garante que cada escolha feita é ótima, não só numa determinada instância, mas também para o resultado final.

Cenário 3 - Análise da Complexidade

Neste cenário, a estrutura *MinHeap* provou ser a mais adequada, já que garante extrair o valor mínimo, no pior caso, em complexidade temporal *O(log n)*.

Não obstante, a criação do *MinHeap* requer todos os elementos do dataset, de modo a organizá-los pelo tempo de entrega.

Portanto, para este cenário, a complexidade temporal esperada seria
O(log n + n) = O(n). A complexidade espacial é O(1).

Cenário 3 - Avaliação Empírica

Após vários testes, com datasets de tamanho variável, o tempo máximo demorado para o programa correr foi ~0.007s. Este total, quando dividido pelo tamanho n do dataset, era exponencialmente pequeno.

Algoritmo de Destaque

O nosso foco neste trabalho foi cumprir os requerimentos da forma mais correta possível, portanto, ao invés de investir grande parte do nosso esforço num único algoritmo, decidimos dar igual atenção aos 3 cenários, maximizando, assim, os resultados positivos.

Dificuldades e Esforço dos Elementos

Uma das grandes dificuldades encontradas neste trabalho foi a constante necessidade de adaptação. Como o nosso objetivo foi resolver cada cenário da forma mais eficiente possível, muitas das vezes, tornou-se imperativo a adequação ao problema em mão, e o ponderar das soluções.

Esforço e contribuição de cada elemento:

Tomás Torres - Cenário 1
José Isidro - Cenário 2
Matilde Silva - Cenário 3

Obrigada pela atenção!