

Logística Urbana para Entrega de Mercadorias

Plataforma Digital

Guilherme Sequeira, Pedro Ramalho, Tomás Pacheco

Faculdade de Engenharia
Universidade do Porto

Desenho de Algoritmos, 2021/2022, 2º semestre

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Descrição do problema

O objetivo do projeto consiste em implementar a plataforma de uma empresa de logística urbana, de modo a tornar a sua operação o mais eficiente possível. Considera-se então a implementação de 3 cenários diferentes, cada um com o seu propósito.

A seguir apresenta-se uma explicação formal e detalhada sobre cada um destes cenários, bem como as análises temporais da sua implementação.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas**
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Formalização do problema

Descrição

Dado um conjunto de estafetas E , de tamanho m , cada um com volume máximo V_e e peso máximo W_e , e um conjunto de pedidos P , de tamanho n , cada um com volume v_p e peso w_p , pretende-se maximizar o número de pedidos entregues, minimizando o número de estafetas contratados.

Seja I_p a variável que representa a inclusão da entrega $p \in P$, C_e a variável que representa a contratação do estafeta $e \in E$, e x_{ep} a variável que determina se o pedido p é entregue pelo estafeta e .

1 Objetivo

$$\text{maximizar } \sum_{p=1}^n I_p$$

$$\text{minimizar } \sum_{e=1}^m C_e$$

2 Restrições

- $w_p, W_e, v_p, V_e \in \mathbb{Z}^+$
- $C, I, x \in \{0, 1\}$
- $W_e \geq \sum_{p=1}^n w_p x_{ep}, V_e \geq \sum_{p=1}^n v_p x_{ep}$

Descrição do algoritmo

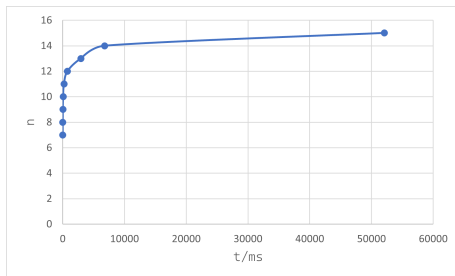
Em baixo encontra-se pseudocódigo para o algoritmo usado no cenário 1:

```
1 sort(estafetas, ordem decrescente, por valor)
2 sort(pedidos, ordem decrescente, por valor)
3
4 for pedido in pedidos:
5     for estafeta in estafetas:
6         if pedido fits in estafeta:
7             estafeta.add_pedido(pedido)
8             break
```

Análise da complexidade

Relembrando que m indica o número de estafetas e n indica o número de pedidos, o algoritmo anterior possui complexidade:

$$\mathcal{O}(n \log(n) + m \log(m) + mn)$$



($n = 7, t = 4, e = 54$) ($n = 8, t = 8, e = 107$) ($n = 9, t = 21, e = 215$)

($n = 10, t = 62, e = 422$) ($n = 11, t = 198, e = 853$) ($n = 12, t = 766, e = 1703$)

($n = 13, t = 2927, e = 3402$) ($n = 14, t = 6795, e = 6795$) ($n = 15, t = 52077, e = 13576$)

Observação: para um valor $n = k$ haverá 2^k estafetas e 9×2^k pedidos.

Problema

É necessário atribuir um critério de prioridade aos estafetas e aos pedidos. Como obter tal critério?

Inicialmente, começou-se por chamar a este critério de "valor". Assim, cada estafeta e pedido possui um valor associado. A primeira iteração do algoritmo definia o valor de um estafeta e pedido da seguinte forma:

$$val_e = \min(w_e/w_t, v_e/v_t) \qquad val_p = \min(w_p/w_t, v_p/v_t),$$

onde w_t e v_t representam o peso e volume ocupado por todos os pedidos, respetivamente.

Desta forma, o algoritmo seria capaz de ajustar a prioridade dada aos estafetas selecionados de acordo com a necessidade de peso ou volume mais predominante nos pedidos. Após várias iterações observou-se que $val_e = w_e + v_e$ e $val_p = w_p + v_p$ obteve os melhores resultados.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros**
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Formalização do problema

Descrição

Dado um conjunto de estafetas E , de tamanho m , cada um com volume máximo V_e , peso máximo W_e e custo C_e , e um conjunto de pedidos P , de tamanho n , cada um com volume v_p , peso w_p e recompensa c_p , pretende-se maximizar o número de pedidos entregues de forma a maximizar os lucros da empresa.

Seja I_p a variável que representa a inclusão da entrega $p \in P$, H_e a variável que representa a contratação do estafeta $e \in E$, e x_{ep} a variável que determina se o pedido p é entregue pelo estafeta e .

1 Objetivo

$$\text{maximizar } \sum_{p=1}^n I_p \qquad \text{maximizar } \sum_{p=1}^n c_p I_p - \sum_{e=1}^m C_e H_e$$

2 Restrições

- $w_p, W_e, v_p, V_e, c_p, C_e \in \mathbb{Z}^+$
- $I, H, x \in \{0, 1\}$
- $W_e \geq \sum_{p=1}^n w_p x_{ep}, V_e \geq \sum_{p=1}^n v_p x_{ep}$

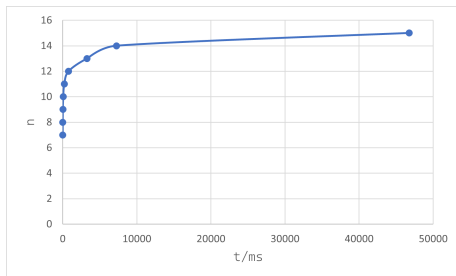
Em baixo encontra-se pseudocódigo para o algoritmo usado no cenário 2:

```
1 sort(estafetas, ordem decrescente, por custo)
2 sort(pedidos, ordem decrescente, por custo)
3
4 for pedido in pedidos:
5     for estafeta in estafetas:
6         if pedido fits in estafeta:
7             estafeta.add_pedido(pedido)
8             break
```

Análise da complexidade

Relembrando que m indica o número de estafetas e n indica o número de pedidos, o algoritmo anterior possui complexidade:

$$\mathcal{O}(n \log(n) + m \log(m) + mn)$$



($n = 7$, $t = 3$, $L = 378472$) ($n = 8$, $t = 8$, $L = 766746$) ($n = 9$, $t = 19$, $L = 1415487$)

($n = 10$, $t = 70$, $L = 2951155$) ($n = 11$, $t = 217$, $L = 5963478$) ($n = 12$, $t = 795$, $L = 12009077$)

($n = 13$, $t = 3285$, $L = 23843787$) ($n = 14$, $t = 7240$, $L = 48277534$) ($n = 15$, $t = 46744$, $L = 96521460$)

Observação: para um valor $n = k$ haverá 2^k estafetas e 9×2^k pedidos.

Problema

É necessário atribuir um critério de prioridade aos estafetas e aos pedidos. Como obter tal critério?

Inicialmente, começou-se por chamar a este critério de "custo". Assim, cada estafeta e pedido possui um custo associado. A primeira iteração do algoritmo definia o custo de um estafeta e pedido da seguinte forma:

$$C_e = (W_e + V_e)/R_e \qquad c_p = r_p/(w_p + v_p),$$

onde R_e e r_p representam o custo de um estafeta e a recompensa de um pedido, respetivamente.

Desta forma, o algoritmo daria prioridade aos estafetas que possuíssem uma capacidade de armazenamento maior, com um custo menor, e aos pedidos com uma recompensa maior, que ocupassem menos espaço. Após várias iterações observou-se que $c_p = w_p + v_p$ obteve os melhores resultados.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega**
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Descrição

Dado um conjunto de pedidos *expresso* P , de tamanho n , cada um com tempo de entrega t_p , pretende-se maximizar o número de pedidos entregues num só dia, sabendo que a entrega dos pedidos só pode ocorrer entre as 09:00h e 17:00 e só pode ser entregue um único pedido de cada vez.

Seja I_p a variável que representa a inclusão da entrega $p \in P$.

① Objetivo

$$\text{maximizar } \sum_{p=1}^n I_p$$

② Restrições

- $I \in \{0, 1\}$
- $\sum_{p=1}^n I_p t_p \leq 8 \times 60 \times 60$

Descrição dos algoritmos

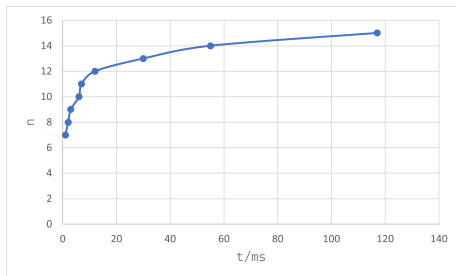
Em baixo encontra-se pseudocódigo para o algoritmo do cenário 3:

```
1 TEMPO_LIMITE = 8 * 60 * 60
2 tempo_total = 0
3 i = 0
4
5 sort(pedidos, ordem crescente, por tempo de entrega)
6
7 while true:
8     entregar(pedido[i])
9     tempo_total += pedido[i].tempo
10
11     if tempo_total > TEMPO_LIMITE:
12         tempo_total -= pedido[i].tempo
13         retirar(pedido[i])
14         break
15
16     i++
```

Análise da complexidade

Relembrando que n indica o número de pedidos, o algoritmo anterior possui complexidade:

$\mathcal{O}(n \log(n))$, no pior dos casos



($n = 7, t = 1, p = 172$) ($n = 8, t = 2, p = 202$) ($n = 9, t = 3, p = 229$)

($n = 10, t = 6, p = 254$) ($n = 11, t = 7, p = 269$) ($n = 12, t = 12, p = 278$)

($n = 13, t = 30, p = 284$) ($n = 14, t = 55, p = 286$) ($n = 15, t = 117, p = 288$)

Observação: para um valor $n = k$ haverá 2^k estafetas e 9×2^k pedidos.

A estratégia usada consiste em ordenar os pedidos por ordem crescente de duração, o que garante a solução ótima. É uma abordagem *greedy* ao problema que atinge os resultados esperados.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo**
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências

Os cenários 1 e 2 possuem soluções semelhantes - implementam um algoritmo de *first-fit* em que os estafetas e pedidos foram sujeitos a uma heurística de ordenação de modo a maximizar a eficiência do mesmo. O critério de prioridade, em cada cenário, provém de uma avaliação empírica resultante de diversas iterações de cada um dos algoritmos, pelo que sofreu várias alterações ao longo da realização do projeto.

Realça-se deste modo a obtenção de uma solução, para os primeiros cenários, que tenta emular o comportamento de um algoritmo *first-fit decreasing* num contexto multi-dimensional.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo**
- 7 Outras observações
- 8 Referências

Durante a realização do projeto surgiram diversas dificuldades, pelo que se salientam as seguintes:

- encontrar um critério de prioridade para os estafetas e pedidos, nos cenários 1 e 2
- o enquadramento de algoritmos conhecidos (*bin-packing*, *0-1 knapsack*) num contexto a 2 e 3 dimensões
- o estudo fidedigno e preciso da complexidade, quer temporal quer espacial, dos algoritmos implementados

Cada um dos elementos do grupo contribuiu equitativamente para a realização do projeto.

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações**
- 8 Referências

Como foi gerado o *dataset*?

Foram gerados um total de 9 ficheiros para estafetas e 9 ficheiros para entregas. Estes dados foram obtidos da seguinte forma:

- 1 definiu-se o número de estafetas numa potência de 2
- 2 decidiu-se que o número de entregas seria 9 vezes o número de estafetas, de modo a ser fiel ao *dataset* original
- 3 geraram-se valores aleatórios para cada um dos atributos que compõem um estafeta e pedido (volume, peso, custo, recompensa, tempo...)
- 4 escreveram-se os dados em ficheiros com a mesma formatação do *dataset* original

Caso o utilizador queira usar o seu próprio dataset deverá garantir que o ficheiro dos estafetas e pedidos possuem exatamente os nomes *assistants.txt* e *deliveries.txt*, respetivamente. Deverá ainda garantir que a primeira linha de cada ficheiro contém o número de linhas com informação a ser processada pela plataforma.

Os critérios de prioridade escolhidos para os cenários 1 e 2 derivam essencialmente de um processo de tentativa e erro. No entanto, podem existir formas de melhorar os resultados obtidos para estes algoritmos, pelo que se salientam as seguintes:

- 1 um critério de ordenação que melhor se adaptasse à natureza bi ou tridimensional do problema, ou seja, que desse igual importância às variáveis envolvidas sem as tratar de forma intercambiável, como é o caso do volume e do peso
- 2 uma melhor atribuição de pedidos a um estafeta, que tenha em conta o contexto global do cenário, de modo a aprimorar os resultados obtidos
- 3 experimentação com outras tácticas de inserção de pedidos em estafetas, como por exemplo *knapsack* binário bidimensional

Conteúdos

- 1 Descrição do problema
- 2 Cenário 1 - minimização de estafetas
- 3 Cenário 2 - maximização dos lucros
- 4 Cenário 3 - minimização do tempo de entrega
- 5 Destaque de algoritmo
- 6 Principais dificuldades, esforço do grupo
- 7 Outras observações
- 8 Referências**

Referências:

<https://www.sciencedirect.com/science/article/abs/pii/S0167637703000579>

<https://www.sciencedirect.com/science/article/pii/S1877050913003980>