Logística Urbana para Entrega de Mercadorias

Desenho de Algoritmos

Grupo 37

Guilherme Moreira- up202007036 (2LEIC11)
Isabel Amaral- up202006677 (2LEIC13)
Milena Gouveia- up202008862 (2LEIC13)

Descrição do problema

O objetivo deste trabalho era a criação de uma plataforma eletrónica de *crowdsourcing* personalizada às necessidades de uma empresa que realiza entrega de mercadorias em zonas urbanas, procurando otimizar vários cenários com que a empresa se depara no dia a dia.

Essencialmente, são realizados dois tipos de serviços de entrega:

- Normal: com taxa de envio, peso e volume associados. Estes são feitos com recurso à subcontratação de estafetas que cobram um valor pelo serviço e que usam as suas próprias viaturas, também estas com volume e peso máximo.
- Expresso: com tempo estimado de entrega (ida+volta) e são feitas por uma viatura da própria empresa que transporta um pedido de cada vez, independentemente do seu volume ou peso.

Cenário 1 - Otimização do número de estafetas

Objetivo: Maximizar as entregas e minimizar os estafetas, sabendo que estes têm uma certa capacidade (volume e peso máximo).

Estratégia utilizada: Greedy, Bin Packing

Formalização do problema:

Dados de entrada:

- D = {D1, D2, D3, ..., Dn}, conjunto de estafetas (drivers) cada um com volume Vd e peso Wd
- P = {P1, P2, P3, ..., Pn}, conjunto de encomendas (packages) de entrega normal cada uma com volume Vp e peso Wp

Output: Nd, nº total de estafetas a serem utilizados

Cenário 1 - Otimização do número de estafetas

Descrição da solução:

- Foram considerados 3 critérios para a ordenação das encomendas e estafetas:
  - peso, descendente
  - volume, descendente
  - peso+volume, descendente
- aplicou-se o algoritmo descrito de seguida considerando cada um dos critérios de ordenação e, no final, escolheu-se aquele com melhores resultados (menor nº de estafetas)
- para cada encomenda a ser entregue, percorre-se a lista (ordenada) de estafetas disponíveis e atribui-se a encomenda ao primeiro estafeta com capacidade para a transportar, isto é, cujo peso e volume da encomenda juntamente com o peso e volume que o estafeta já se encontra a transportar não excedam os limites máximos da viatura do estafeta

Cenário 1 - Otimização do número de estafetas

Complexidade temporal: O(n*d), n é o nº de encomendas e d é o nº de estafetas

- ordenação: O(n*log n) - utilizando o método sort definido na classe list da STL e a nossa própria função de comparação

método alocatePackages: O(n*d)

- iteração pela lista de encomendas: O(n)

para cada encomenda, encontrar um estafeta com capacidade para a transportar:

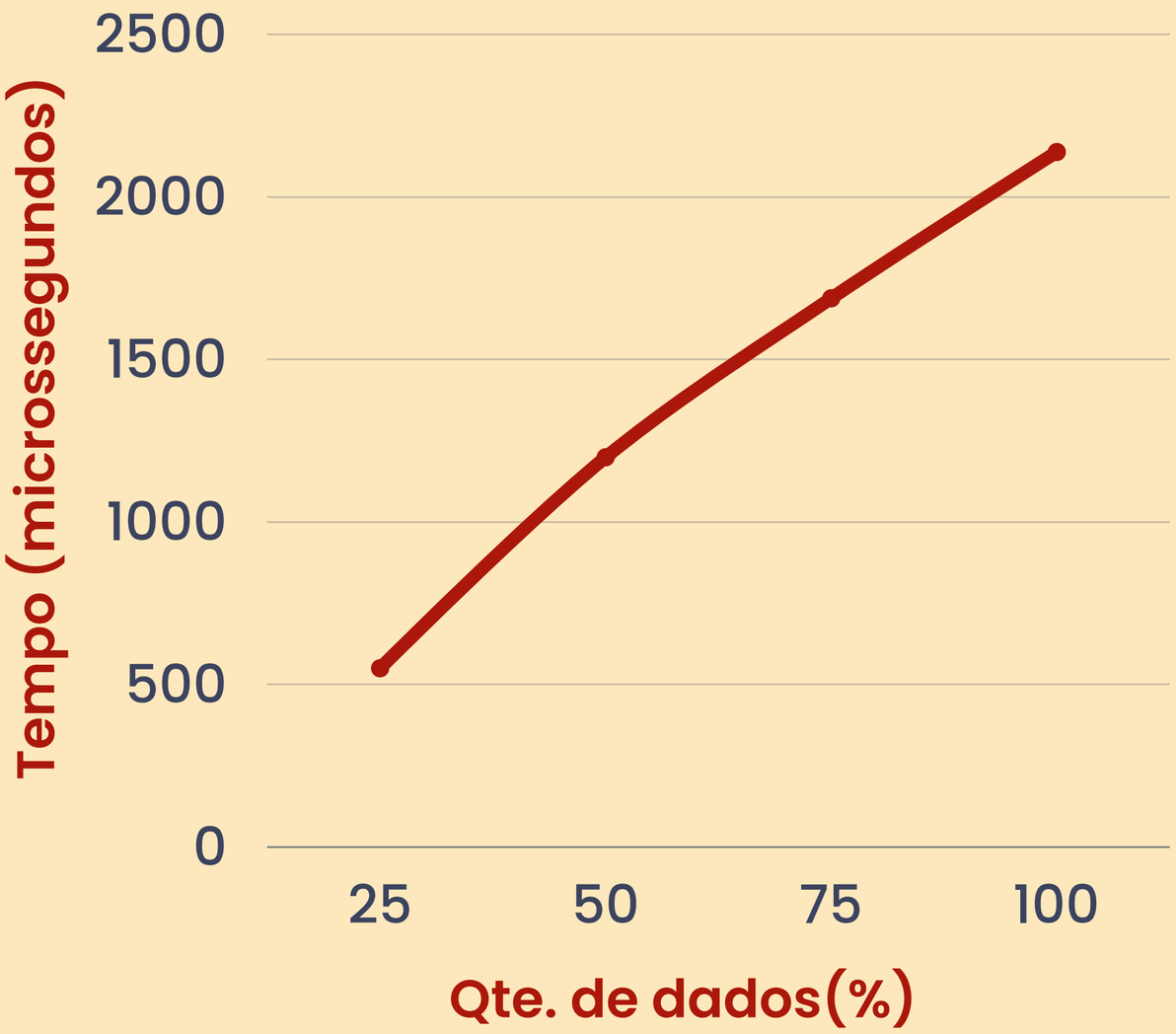
- iteração pela lista de estafetas: O(d) - nos casos em que for necessário usar todos os estafetas (pior caso)

Complexidade espacial: O(n+d), n é o nº de encomendas e d é o nº de etsfetas

Todas as variáveis ocupam espaço constante com excessão das seguintes variáveis auxiliares:

- lista de encomendas a entregar - tamanho n
- lista de estafetas a ser utilizados - tamanho d, no pior caso

Cenário 1 - Análise Empírica



Com recurso ao gráfico ao lado, pode-se verificar que para diferentes tamanhos de input, o tempo de execução aumenta **linearmente**, tal foi como referido.

Cenário 2 – Otimização do lucro da empresa

Objetivo: Maximizar o lucro da empresa e as encomendas a entregar, tendo em conta que a empresa recebe uma receita por cada entrega e que os estafetas cobram um valor pelo número total de pedidos entregues num dia.

Estratégia utilizada: Greedy

Formalização do problema:

Dados de entrada:

- D = {D1, D2, D3, ..., Dn}, conjunto de estafetas (drivers) cada um com volume Vd, peso Wd e custo Cd
- P = {P1, P2, P3, ..., Pn}, conjunto de encomendas (packages) de entrega normal cada uma com volume Vp, peso Wp e recompensa Rp

Output: L, lucro que a companhia terá após as entregas

Cenário 2 - Otimização do lucro da empresa

Descrição da solução:

- ordenação das pedidos e dos estafetas por ordem descendente e ascendente da receita/custo, respetivamente
- para cada estafeta, alocar o máximo de encomendas possíveis
- no final de cada iteração, escolher o estafeta que garante maior lucro, eliminá-lo da lista de estafetas disponíveis e eliminar as encomendas que lhe foram alocadas da lista de encomendas a entregar
- repetir este processo até que não sobrem encomendas ou até se começar a ter um lucro negativo
- se no final houver pedidos por entregar, recorre-se ao algoritmo usado no cenário 1, pois este tem o principal objetivo de maximizar o número de entregas. Contudo, só se usa estes valores se o lucro for maior que o obtido anteriormente

Cenário 2 - Otimização do lucro da empresa

Complexidade temporal: O(n*d), n é o nº de encomendas e d é o nº de estafetas

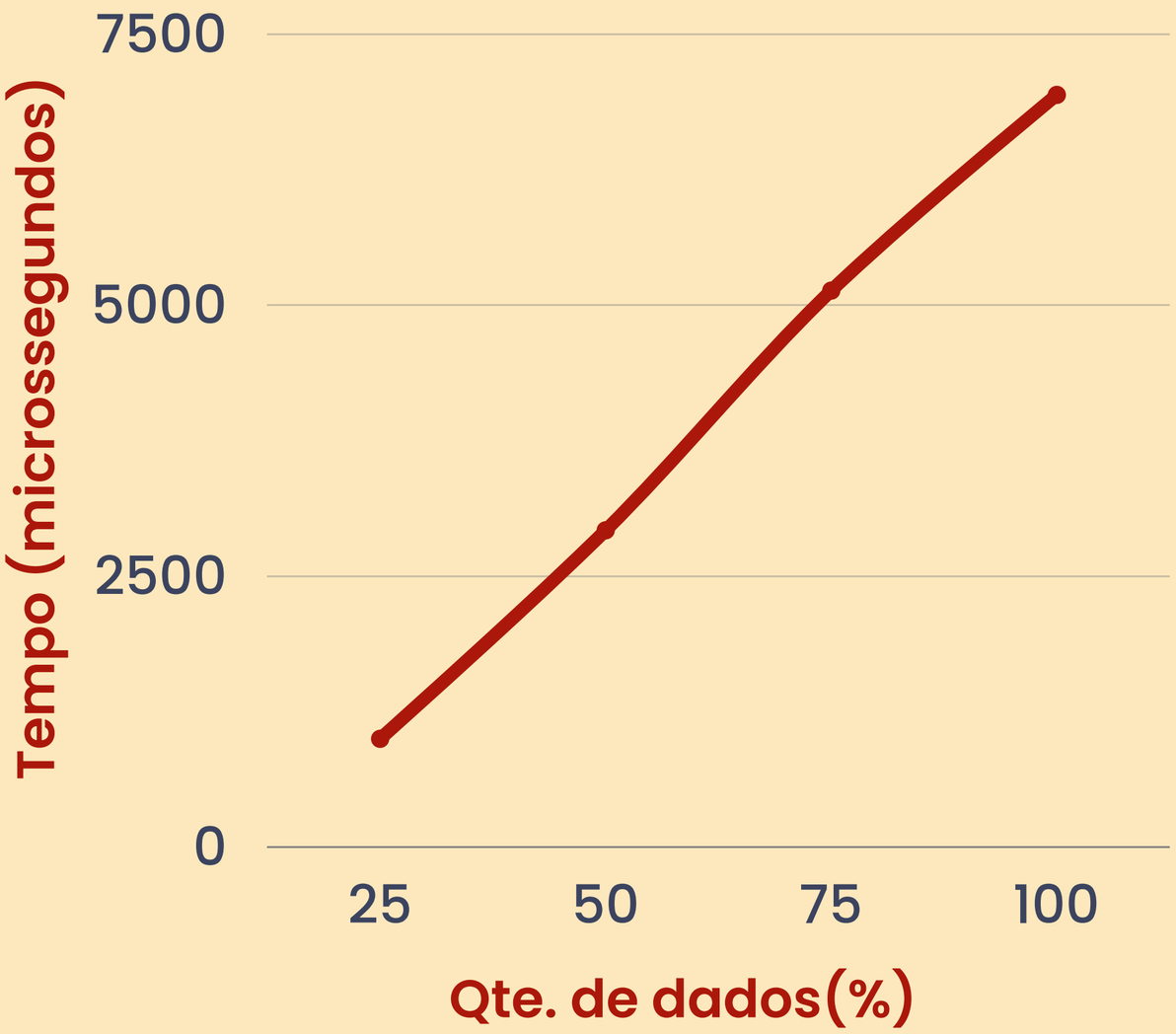
- ordenação: O(n*log n) - utilizando o método sort definido na classe list da STL e a nossa própria função de comparação
  enquanto houver encomendas para entregar: O(n), no pior caso
- iteração pela lista de estafetas: O(d) - no caso em que for necessário usar todos os estafetas
  para cada estafeta, alocar o máximo de encomendas:
- iteração pela lista de encomendas: O(1) - nº de encomendas selecionadas para cada estafeta será muito inferior ao nº total de encomendas (no caso médio)

Complexidade espacial: O(n+d), n é o nº de encomendas e d é o nº de etsfetas

Todas as variáveis ocupam espaço constante com excessão das seguintes variáveis auxiliares:

- lista de encomendas a entregar - tamanho n
- lista de estafetas a ser utilizados - tamanho d, no pior caso

Cenário 2 - Análise Empírica



O gráfico ao lado permite-nos perceber que, de acordo a dimensão do input, a determinação da solução deste cenário tem um tempo que cresce **linearmente** tal como referido.

Cenário 3 – Otimização das entregas expresso

Objetivo: Minimizar o tempo médio previsto das entregas expresso a serem realizadas num dia, isto é, maximizar o número de encomendas expresso a ser entregues durante o horário comercial da companhia (das 9h às 17h).

Estratégia utilizada: Greedy

Formalização do problema:

Dados de entrada:

- P = {P1, P2, P3, ..., Pn}, conjunto de encomendas (packages) de entrega normal cada uma com recompensa de entrega Cp

Output: Np, nº de encomendas expresso que será possível entregar

Cenário 3 – Otimização das entregas expresso

Descrição da solução:

- ordenação das encomendas por tempo estimado de entrega ascendente (da que demora menos tempo a entregar para a que demora mais)
- considerar que o tempo total que pode ser gasto a entregar este tipo de entregas corresponde a 8 horas (duração do horário comercial)
- selecionar o maior número de encomendas do início da lista (encomendas de menor tempo de entrega) cuja soma dos tempos de duração não excede as 8 horas

Cenário 3 – Otimização das entregas expresso

Complexidade temporal: O(n*log n), n é o nº de encomendas

- ordenação: O(n*log n) - utilizando o método sort definido na classe list da STL e a nossa própria função de comparação
- iteração pela lista de encomendas expresso: O(n) - nos casos em que for possível entregar todas as encomendas (pior caso)

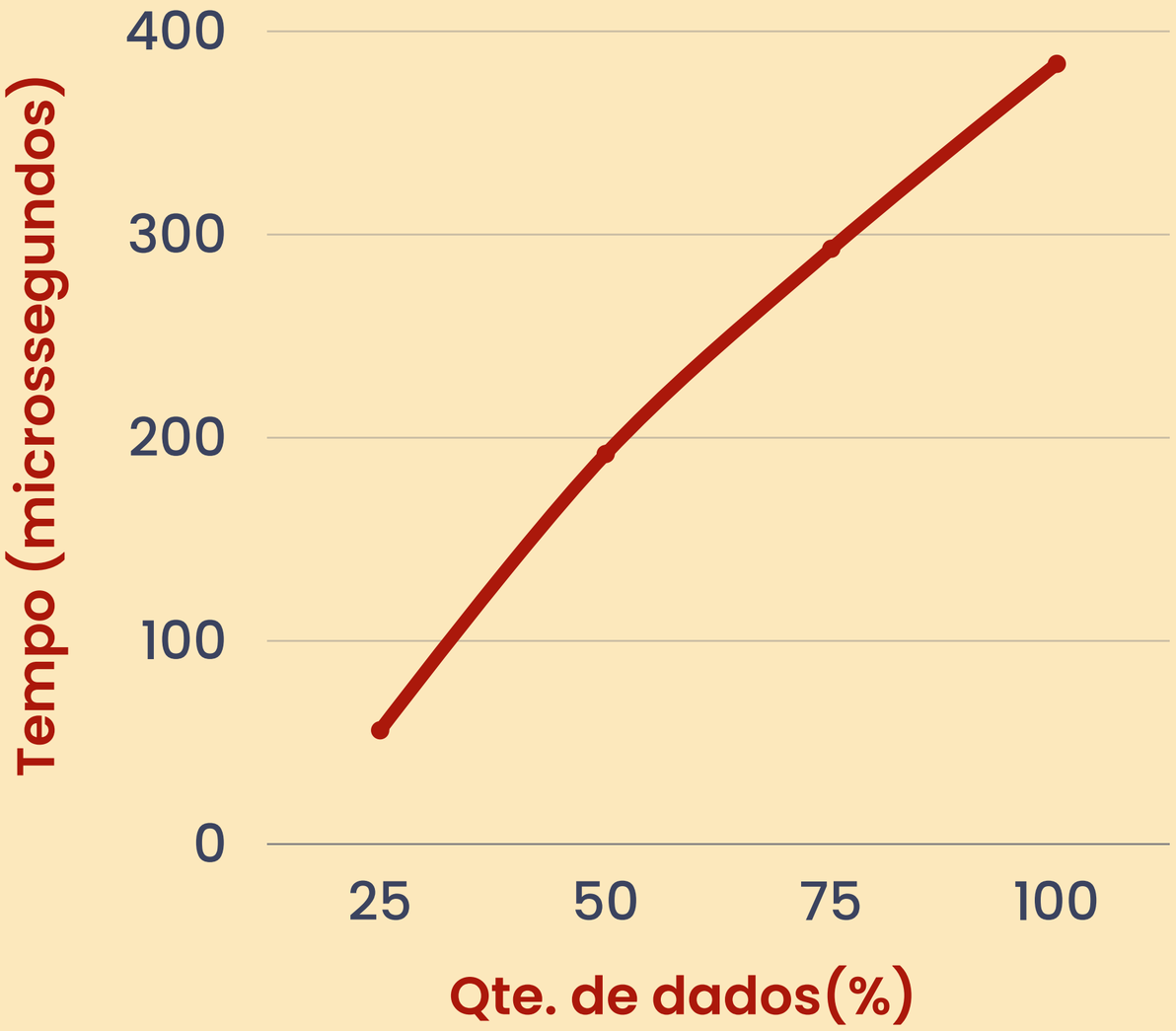
Complexidade espacial: O(n), n é o nº de encomendas

Além dos dados de entrada:

- lista de encomendas a entregar - tamanho n

Todas as variáveis usadas ocupam espaço constante, com exceção da lista onde as encomendas a ser entregues são guardadas. No pior caso, caso em que todas as encomendas são entregues, essa lista terá tamanho n

Cenário 3 - Análise Empírica



O seguinte gráfico, aparenta aproximar-se de um gráfico **n*logn**, o que está de acordo com a complexidade temporal mencionada anteriormente.

Funcionalidade extra

Decidimos incluir a possibilidade de transferir os pedidos não entregues para o dia seguinte passando a priorizar estes.

Isto foi feito adicionando um novo atributo, prioridade, aos objetos que representam encomendas, tanto normais como expresso. No dia em que seria suposto entregar estes pedidos (no momento em que são lidos os ficheiros de criação de encomendas), a prioridade de todos os pedidos é 1. Caso não seja possível a sua entrega, ao fim do primeiro dia a prioridade é incrementada, passando a 2, e ao fim de outro dia em que não seja novamente possível a sua entrega passará a 3...

Em qualquer um dos cenários quando as encomendas são ordenadas (usando o critério escolhido para esse cenário), a prioridade passa também a ser levada em consideração e, deste modo, encomendas com maior prioridade ficam colocadas no início da lista.

Esforço e principais dificuldades

As nossas principais dificuldades foram, no geral, o planeamento do melhor algoritmo a usar nos cenários 1 e 2 de modo a levar em consideração todas as restrições dadas no enunciado.

No geral, o trabalho foi dividido pelos elementos do grupo de forma mais ou menos equitativa. Cada elemento ficou responsável pela implementação de um cenário e algumas das classes que foram usadas como base. Contudo, o planeamento foi feito em conjunto e, durante todo o processo, todos os elementos do grupo se mostraram disponíveis para ajudar e esclarecer as dúvidas que foram surgindo ao longo do trabalho.