GOODS DELIVERY LOGISTICS

Diogo Fonte - up202004175
João Tedesco - up202000416
Sofia Moura - up201907201

Descrição do problema

O objetivo principal deste trabalho é implementar uma plataforma digital de gestão de uma empresa de logística urbana, com o objetivo de tornar a sua operação o mais eficiente possível. Para que tal seja possível, vão ser exploradas diferentes formas de distribuir os vários pedidos pelos estafetas registados na plataforma, no caso dos pedidos normais, assim como a sequência de entregas dos pedidos expresso.

Cenário 1

otimização do número de estafetas

Formalização

Dados de entrada

- Armazém: caracterizado por listas de estafetas e encomendas como atributos privados, que serão ordenados e posteriormente percorridos.
- Estafetas: caracterizados por uma lista de encomendas inicialmente vazia, volume e peso máximo que podem carregar, volume e peso atual ocupado pelas encomendas.
- Encomendas: caracterizadas pelo seu volume e peso.

Dados de saída

- Estafetas: com encomendas atríbuidas

Função objetivo

- Atribuir o máximo número de encomendas ao número mínimo de estafetas.

Algoritmos relevantes

Para resolver o problema decidiu-se utilizar o algoritmo greedy mais conhecido por "bin packing" para distribuir as diferentes encomendas por cada estafeta necessário.

Primeiramente ordenaram-se as encomendas, por ordem crescente, de acordo com o seu valor de peso ou volume ocupado (conforme qual dos dois fosse o máximo entre ambos), já que desta forma minimiza-se o volume e peso transportados e consequentemente maximiza-se o número de encomendas transportadas.

Como se optou por usar a abordagem de "first fit decreasing" do algoritmo, de seguida iterou-se a lista de estafetas, distribuindo o máximo possível de encomendas, um a um.

Apesar de não fazer parte do algoritmo "bin packing", pois o normal seria ter estafetas com capacidades iguais, é importante referir que se ordenou também a lista de estafetas, por ordem decrescente, pela soma do valor de peso e volume que podiam transportar, uma vez que desta forma os estafetas com menor capacidade de transporte seriam deixados para último caso.

Análise de complexidade

Complexidade Temporal

- 2 std::list::sort -> O(n*log(n) + e*log(e))
- 2 ciclos identados for para iteração -> O(e*n)
  - std::list::erase -> O(1)

Final: O(n*log(n) + e*log(e) + e*n*1) = O(e*n)

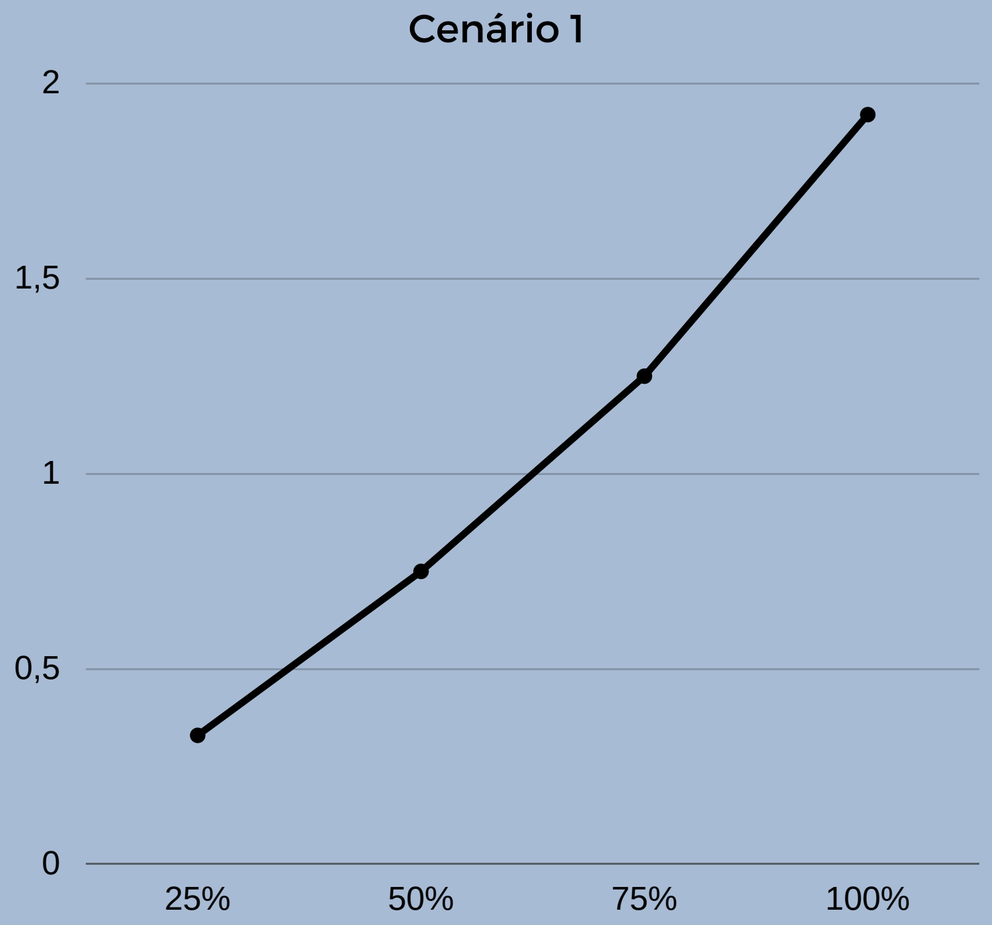
Complexidade Espacial

- std::list para representar a lista de encomendas -> O(n)
- std::list para representar a lista de estafetas -> O(e)

Final: O(n+e)

(sendo "n" e "e" os tamanhos das listas de encomendas e estafetas)

Resultados da avaliação empírica

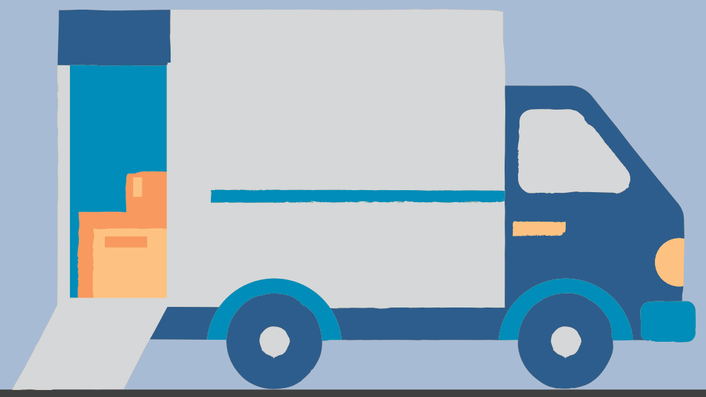


O eixo das abscissas representa a percentagem de items usados a partir dos ficheiros originais e o eixo das ordenadas, o tempo de execução decorrido em milisegundos.

Conclusão: a partir da análise do gráfico, conclui-se que a evolução da complexidade temporal aumenta linearmente.

Cenário 2

otimização do lucro da empresa



Formalização

Dados de entrada

- Armazém: caracterizado por listas de estafetas e encomendas como atributos privados, que serão ordenados e posteriormente percorridos.
- Estafetas: caracterizados por uma lista de encomendas inicialmente vazia, volume e peso máximo que podem carregar, volume e peso atual ocupado pelas encomendas.
- Encomendas: caracterizadas pelo seu volume e peso.

Dados de saída

- Lucro da empresa naquele dia com as entregas e estafetas selecionadas(os).

Função objetivo

- Obter o máximo de lucro possível para a empresa, tendo em conta os estafetas disponíveis e as encomendas a serem entregues.

Algoritmos relevantes

Para resolver o problema decidiu-se utilizar o algoritmo de programação dinâmica mais conhecido por "multidimensional knapsack problem" para distribuir as diferentes encomendas por cada estafeta necessário, otimizando o lucro da empresa.

Primeiramente ordenaram-se os estafetas, por ordem decrescente de acordo com o custo por volume+peso, calculado por (peso+volume)/custo.

De seguida iterou-se a lista de estafetas e usou-se o algoritmo "multidimensional knapsack problem" para encontrar o melhor subconjunto de encomendas para o estafeta selecionado.
O somatório de recompensas desse subconjunto menos o custo do estafeta é o lucro para o caso específico. Se for positivo retira-se o subconjunto da lista de encomendas, coloca-se no estafeta selecionado e depois é feita a mesma operação para o próximo estafeta. Caso seja negativo a função termina e retorna o valor calculado até ao momento.

Análise de complexidade

Complexidade Temporal

- 1 std::list::sort -> O(n*log(n))
- 1 ciclo for para iterar
  os estafetas -> O(e) :
  - 3 ciclos identados for para iteração
    -> O(n*e.maxPeso*e.maxVolume)
  - 1 ciclo for para iterar
    as encomendas-> O(n)

Final: O(e*(n*e.maxPeso*e.maxVolume+n))

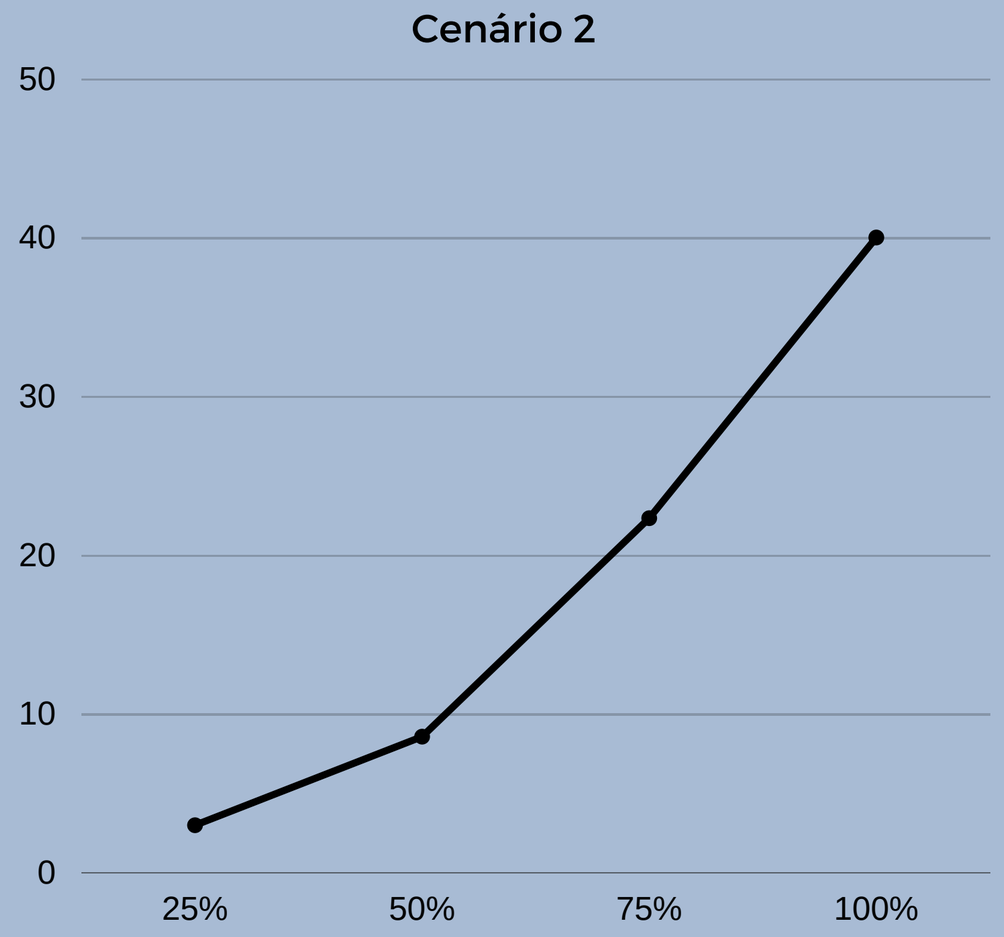
Complexidade Espacial

- std::list para representar a lista de encomendas -> O(n)
- std::list para representar a lista de estafetas -> O(e)
- std::vector para a função auxiliar multidimensionalKnapsack

Final: O(n+e+e*(n*e.maxPeso*e.maxVolume))

(sendo "n" e "e" os tamanhos das listas de encomendas e estafetas)

Resultados da avaliação empírica

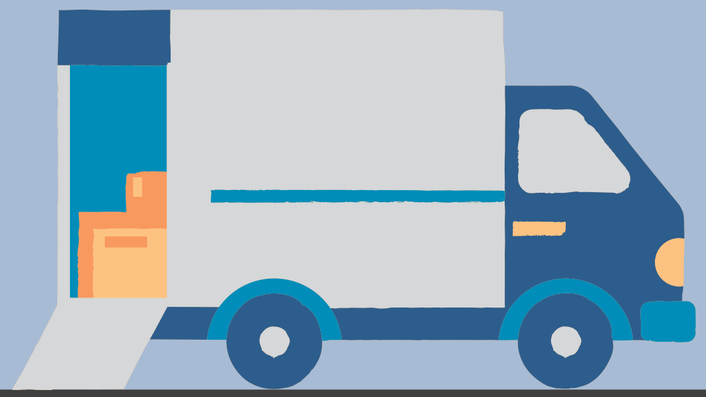


O eixo das abscissas representa a percentagem de items usados a partir dos ficheiros originais e o eixo das ordenadas, o tempo de execução decorrido em segundos.

Conclusão: a partir da análise do gráfico, conclui-se que a evolução da complexidade temporal aumenta exponencialmente.

Cenário 3

otimização das entregas expresso



Formalização

Dados de entrada

- Armazém: caracterizado por uma lista de encomendas como atributo privado, que será ordenada por ordem crescente de duração de entrega e posteriormente percorrida.
- Encomendas: caracterizadas pelo seu volume, peso e duração de entrega.

Dados de saída

- Número de encomendas expresso realizadas no dia.
- Encomendas expresso selecionadas.

Função objetivo

- Realizar o máximo de entregas de encomendas expresso no dia.

Algoritmos relevantes

Para resolver o problema decidiu-se utilizar uma variação do algoritmo greedy que resolve o problema de escalonamento de atividades, para escolher as diferentes encomendas expresso de forma a que se consiga maximizar o número de encomendas entregues no dia em questão.

Primeiramente ordenaram-se as encomendas, por ordem crescente de duração de entrega

Posto isto iterou-se a lista de encomendas expresso e foi-se adicionando à lista de encomendas expresso, enquanto havia tempo disponível.

Ao priorizar as encomendas de menor duração, foi possível maximizar o número de encomendas expresso entregues, tal como pretendido.

Análise de complexidade

Complexidade Temporal

- std::list::sort -> O(n*log(n))
- ciclo for para iteração -> O(n)
- std::list::erase no for -> O(1)

Final: O(n*log(n) + n*1) = O(n*log(n))

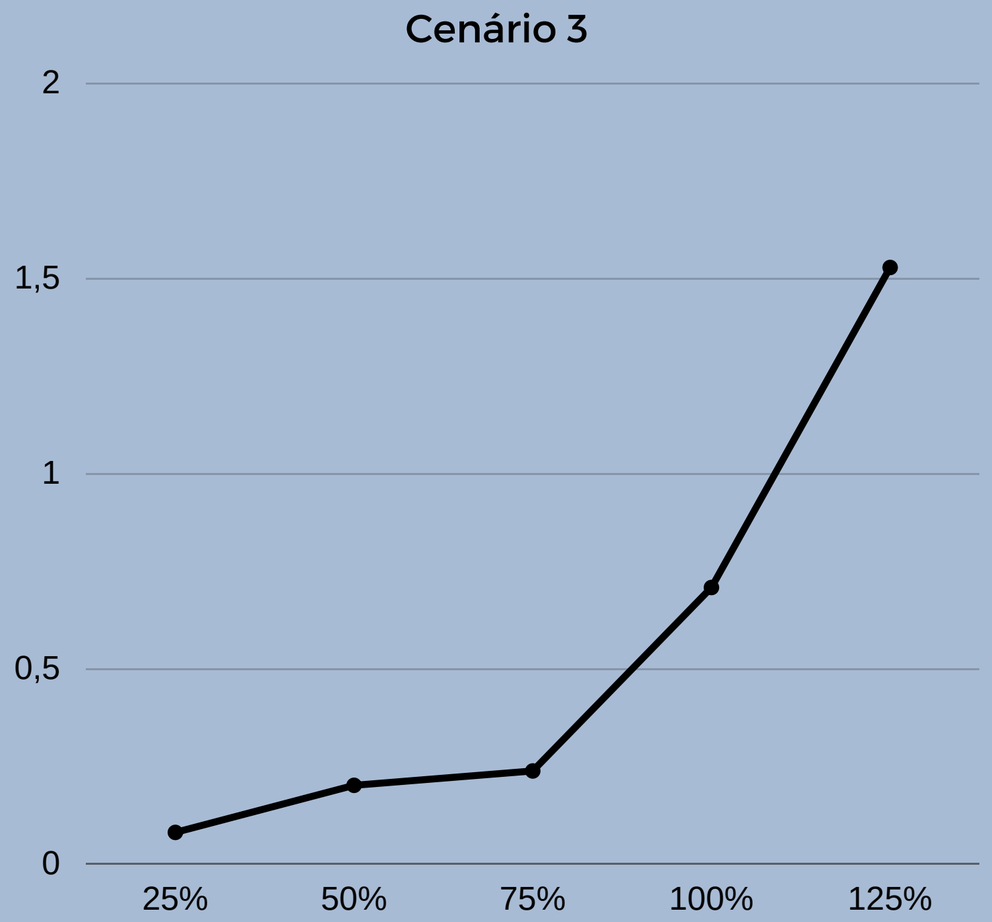
Complexidade Espacial

- std::list para representar a lista de encomendas expresso -> O(n)

Final: O(n)

(sendo "n" o tamanho da lista de encomendas expresso)

Resultados da avaliação empírica



O eixo das abscissas representa a percentagem de items usados a partir dos ficheiros originais e o eixo das ordenadas, o tempo de execução decorrido em milisegundos.

Conclusão: a partir da análise do gráfico, conclui-se que a evolução da complexidade temporal aumenta exponencialmente.

Funcionalidade extra

- Medir a eficiência da operação da empresa, em termos do quociente entre o número de pedidos efetivamente entregues e o número de pedidos recebidos num dia.
- Pode ser feita consoante o cenário (1, 2 ou 3) escolhido para otimização.

Algoritmo de destaque

O algoritmo que se decidiu destacar foi o "Bin Packing" ou Problema do Empacotamento. Decidiu-se utilizar este algoritmo para resolver o problema do cenário 1, uma vez que tal como o algoritmo original, para o qual se tem objetos de diferentes volumes que devem ser guardados num número finito de recipientes de forma a minimizar o número de recipientes utilizados, também o cenário descreve um conjunto de encomendas (objetos) de volumes diferentes que precisam de ser atribuídas a um conjunto de estafetas (recipientes).

Principais dificuldades encontradas

- definir o melhor critério de comparação de capacidade global entre estafetas e entre encomendas

Esforço de cada elemento

As tarefas foram divididas por cada elemento do grupo de forma equilibrada e todos se esforçaram igualmente