Logística Urbana para Entrega de Mercadorias

G116

Turma 01

28/04/2022

- Eduardo Luís Tronjo Ramos
- Emanuel Silva Gestosa

Descrição do problema

Pretende-se implementar algoritmos para ajudar a gestão de uma empresa de logística urbana, com especial atenção aos cenários:

- Minimizar o número de estafetas para a entrega de todos os pedidos ou do maior número de pedidos, num dia.
- Maximizar o lucro da empresa para a entrega de todos os pedidos ou do maior número de pedidos, num dia.
- Minimizar o tempo médio previsto das entregas expresso a serem realizadas, num dia.

Cenário 1 - Formalização

- Dados:
  - p_1, ..., p_n - peso das encomendas 1 a n
  - v_1, ..., v_n - volume das encomendas 1 a n
  - P_1, ..., P_n - capacidade de peso dos estafetas 1 a n
  - V_1, ..., V_n - capacidade de volume dos estafetas 1 a n
- Objetivo : Minimizar $\sum_i^n X_i$
- Restrições:
  - $\sum_i^n p_i x_{ij} <= P_j, \forall j$ in {1, ..., n}
  - $\sum_i^n v_i x_{ij} <= V_j, \forall j$ in {1, ..., n}
  - $x_{ij} \in \{0, 1\}$
  - $X_i \in \{0, 1\}$

  onde $X_i = 1$ se o estafeta i está a ser usado e $x_{ij} = 1$ se a encomenda i está a ser entregue pelo estafeta j

Cenário 1 - Descrição de algoritmos relevantes

De modo a resolver este problema de bin packing bidimensional foi usado o algoritmo best-fit-decreasing:

- Ordenar encomendas da maior para a menor (peso * volume)
- Mantém uma lista de estafetas usados, inicialmente vazia
- Para cada encomenda, escolher o estafeta da lista com menor capacidade disponível que consiga ainda transportar a encomenda (se existir)
  - Se existir, usar o estafeta para entregar a encomenda
  - Caso contrário, tentar adicionar um novo estafeta à lista, com a maior capacidade possível e que consiga transportar a encomenda

Cenário 1 - Análise de complexidade

Seja n o número de encomendas e A(n) o número de estafetas usados,

- Complexidade temporal:
  $\mathcal{O}(n \log n)$
- Complexidade espacial:
  $\mathcal{O}(A(n))$

Cenário 1 - Resultados da avaliação empírica

Para efeitos de teste, foi corrido o programa 5 vezes para cada uma das situações descritas e calculada a média dos resultados.

- Dataset original: 450 encomendas
  $ time ./DA_Projeto 1
  6.4ms
- 900 encomendas
  $ time ./DA_Projeto 1
  15.2ms

$15.2/6.4 \simeq \frac{900 \log(900)}{450 \log(450)}$
Verifica-se que os resultados obtidos estão de acordo com o esperado.

Cenário 2 - Formalização

- Dados:
  - p_1, ..., p_n - peso das encomendas 1 a n
  - v_1, ..., v_n - volume das encomendas 1 a n
  - r_1, ..., r_n - recompensa das encomendas 1 a n
  - P_1, ..., P_n - capacidade de peso dos estafetas 1 a n
  - V_1, ..., V_n - capacidade de volume dos
  - C_1, ..., C_n - custo de transporte dos estafetas 1 a n
- Objetivo : Maximizar $\sum_i^n r_i x_i - \sum_i^n C_i X_i$
- Restrições:
  - $\sum_i^n p_i x_{ij} <= P_j, \forall j$ in {1, ..., n}
  - $\sum_i^n v_i x_{ij} <= V_j, \forall j$ in {1, ..., n}
  - $x_i \in \{0, 1\}$
  - $x_{ij} \in \{0, 1\}$
  - $X_i \in \{0, 1\}$

  onde $X_i = 1$ se o estafeta i está a ser usado, $x_i = 1$ se a encomenda i for entregue e $x_{ij} = 1$ se a encomenda i está a ser entregue pelo estafeta j

Cenário 2 - Descrição de algoritmos relevantes

De modo a resolver este problema de knapsack bidimensional, foi usado o seguinte algorítmo:

- Ordena-se os estafetas do mais rentável para o menos rentável
- Para cada estafeta, resolve-se o problema knapsack 0-1 com as encomendas que ainda não foram entregues
  - Usando programação dinâmica, determinar qual a combinação de encomendas que maximiza o lucro.
  - Repetir até que acabem os estafetas, encomendas, ou até os estafetas deixarem de ser rentáveis.

Cenário 2 - Análise de complexidade

Seja n o número de encomendas, e o número de estafetas, p o peso máximo médio de cada estafeta e v o volume máximo médio de cada estafeta,

- Complexidade temporal:
  $\mathcal{O}(n \log n + e \log e + e * n * (p + v))$
- Complexidade espacial:
  $\mathcal{O}(n * (p + v))$

Cenário 2 - Resultados da avaliação empírica

Para efeitos de teste, foi corrido o programa 5 vezes para cada uma das situações descritas e calculada a média dos resultados. Os esfetas são os mesmos para ambas as situações.

- Dataset original: 450 encomendas
  $ time ./DA_Projeto 2
  1.619s
- 900 encomendas
  $ time ./DA_Projeto 2
  4.53s

Os resultados obtidos encontram-se acima do esperado, portanto achamos que a complexidade temporal que calculamos não se encontra totalmente correta.

Cenário 3 - Formalização

- Dados:
  - d_1, ..., d_n - duração de entrega das encomendas de 1 a n
  - t - tempo disponível para entrega de encomendas
- Objetivo : Maximizar $\sum_i^n x_i$
- Restrições:
  - $x_i \in \{0, 1\}$
  - $d_i, t \in \mathbb{N}$

onde $x_i = 1$ se a encomenda i for escolhida para entrega

Cenário 3 - Descrição de algoritmos relevantes

De modo a resolver este problema de job scheduling, foi usado um algorítmo greedy:

- Ordernar as encomendas por ordem ascendente de duração
- Para cada entrega, testar se o tempo disponível para entregas é maior ou igual à duração de entrega
  - Caso seja, escolher a encomenda para entrega e subtrair a duração ao tempo disponível
  - Caso não seja, o algorítmo termina

Cenário 3 - Análise de complexidade

Seja n o número de encomendas,

- Complexidade temporal:
  $\mathcal{O}(n \log n)$
- Complexidade espacial:
  $\mathcal{O}(1)$

Cenário 3 - Resultados da avaliação empírica

Para efeitos de teste, foi corrido o programa 5 vezes para cada uma das situações descritas e calculada a média dos resultados.

- Dataset original: 450 encomendas
  $ time ./DA_Projeto 3
  2.6ms
- 900 encomendas
  $ time ./DA_Projeto 3
  4.2ms

$4.2/2.6 < \frac{900 \log(900)}{450 \log(450)}$

Verifica-se que os resultados obtidos estão ligeiramente abaixo do que era esperado. Estes erros provavelmente devem-se ao facto de a amostra usada ser demasiado pequena.

Destaque de algoritmo

Decidímos destacar o algoritmo usado na resolução do primeiro cenário. Na nossa primeira abordagem, começamos por usar o algoritmo first-fit. Como não ficamos satisfeitos pesquisamos por algoritmos melhores, passando a usar best-fit e mais tarde best-fit-decreasing, o que melhorou significativamente a minimização de estafetas usados, mantendo as complexidades temporais e espaciais inalteradas.

Exemplos de execução

Exemplos de execução com o dataset original:

- Cenário 1:
  $ ./DA_Projeto 1
  Foram entregues 450/450 encomendas e foram usados 21/50 estafetas.
  Lucro total: 177289.
- Cenário 2:
  $ ./DA_Projeto 2
  Foram entregues 349/450 encomendas e foram usados 16/50 estafetas.
  Lucro total: 217470.
- Cenário 3:
  $ ./DA_Projeto 3
  Foram entregues 124/450 encomendas com um tempo médio de 231 segundos.
  Tempo restante: 148 segundos.
  Lucro total: 147842.

Dificuldades e auto-avaliação

Ao longo da realização do trabalho deparamo-nos com dificuldades na resolução do segundo cenário.

O facto de se tratar de um problema de knapsack 0-1 bidimensional, dificultou a pesquisa sobre possíveis abordagens por ser menos comum do que o knapsack clássico, tornando assim mais demorosa a resolução deste problema.

Autoavaliação:

- Eduardo Luís Tronjo Ramos: 50%
- Emanuel Silva Gestosa: 50%