Relatório - Trabalho 1
Otimização

Douglas Affonso Clementino

2021

INTRODUÇÃO

Uma empresa aluga máquinas (para uso remoto) sob demanda de seus clientes. A única restrição é que as máquinas só podem ser usadas durante um mesmo dia de trabalho (expediente de 8h às 17h). Possivelmente mais de um destes usos podem ser alugados num mesmo dia para uma mesma máquina, se a soma dos tempos for menor que as 9 horas do expediente. Cada cliente pede quanto tempo, em minutos, vai usar uma máquina. Esse tempo deve estar entre 0 e 540 minutos.

A empresa tem m máquinas. Ao receber um conjunto de pedidos, o gerente da empresa precisa escalonar em qual máquina e em qual dia cada uso vai ser feito. Considere que a demanda (pedidos) é dada por um conjunto de pares (n_i, t_i), onde n_ie o número de pedidos de tempo, com $1 \leq i \leq n$. Queremos minimizar o número de dias necessário para atender aos pedidos da demanda.

Busca-se então desenvolver uma solução que efetue a modelagem do problema como uma entrada válida ao resolvedor de problemas lineares *lp_solve*(BERKELAAR; EIKLAND; NOTEBAERT, 2021).

Este problema pode ser reinterpretado como *"Cutting Paper Rolls"* apresentado em (MATOUEK; GäRTNER, 2006), e esta é a forma na qual o problema será abordado.

1 MODELAGEM

Dada a relação com *"Cutting Paper Rolls"*, o problema consiste em definir os possíveis padrões de divisão em que cada dia-máquina pode ser dividido e, a partir daí, definir em qual quantidade cada um desses padrões deve ser executado de forma a comprir todas as demandas.

1.1 DEFINIÇÕES

1.1.1 Quantidade de máquinas (m)

Quantidade de máquinas disponíveis para operar diariamente.

1.1.2 Quantidade de Classes (n)

Quantidade de classe de pedidos solicitados (diferentes tempos de execução).

1.1.3 Máximo hora-máquina diária (T)

Tempo limite de execução diária (540 minutos, como indicado em Introdução).

1.1.4 Tempo de Classe i (t_i)

Tempo de pedido i, com $1 \leq i \leq n$.

1.1.5 Quantidade de Solicitações por Classe i (n_i)

Quantidade de solicitações de tempo t_i, com $1 \leq i \leq n$.

1.1.6 Tupla *quantidade-tempo* por Classe de pedido ((n_i, t_i))

Par quantidade (n_i) e tempo (ti) de pedidos, com $1 \leq i \leq n$.

1.1.7 Conjunto de Possíveis Padrões (P)

Conjunto de possíveis padrões de dia-maquina.

1.1.8 Conjunto de Coeficiestes de Padrão Válido (p)

Conjunto de coeficientes a_i que representa um dos padrões de P, com $|p| = n$.

1.1.9 Variavel Relacionada a Cada padrão $p \in P$ (x_j)

Variável que representa a quantidade de vezes da aplicação do padrão p_j com $j \in [1..|P|]$.

1.2 RESTRIÇÕES

Dadas as definições, a primeira etapa para a solução do problema consiste em definir padrões de $p \in P$. Cada padrão p será um conjunto de elementos a_i de forma que o produto com os tempos t_i que seja maximal em relação a T da forma:

$$p = \{a_i | mínimo(t_1, t_2, ..., t_n) < \sum_{i=1}^{n} a_i \times t_i \leq T\}$$

$$P = Todos\ os\ possíveis\ p$$

Onde $p \in P$ será equivalente a 1 dia-máquina de trabalho, com a_i sendo a quantidade de vezes um processo de tempo t_i será executado na aplicação do padrão p. Importante observar que o tempo de cada padrão é inferiormente limitado pelo menor t_i possível, ou seja $mínimo(t_1, t_2, ..., t_n)$.

Assim, é possível atribuir uma variável x_j relacionada a cada padrão p_j, sendo a variável que representará a quantidade de aplicações do padrão p_j em busca de atender cada pedido q_i. Dessa forma as restrições serão dadas em função a quantidade de vezes que cada padrão contribui para a atender a cada requisições q_i, cada qual na seguinte forma:

$$\sum_{j=1}^{|P|} a_{ji} \times x_j \geq q_i, 1 \leq i \leq n$$

Além disso, uma vez que a aplicação de cada padrão pode ser nula ou positiva, não é possível atribuir valores negativos qualquer x_j, logo as seguintes restrições também são restrições válidas:

$$x_j \geq 0, j \in [1..|P|]$$

1.3 FUNÇÃO OBJETIVO

Dado o problema e as especificações, temos que cada dia-máquina é representado por um padrão $p \in P$, que por sua vez tem sua quantidade de aplicações representada por x_j. Então, como queremos minimizar a quantidade de dias-máquina é o mesmo que dizer que quer-se minimizar a quantidade de padrões aplicados, ou seja, o valor das variáveis x_j. Logo, temos como função objetivo:

$$min \sum_{j=1}^{|P|} x_j$$

2 IMPLEMENTAÇÃO

O programa em questão foi desenvolvido em *Python 3*, sem a utilização de bibliotecas relevantes ao processo de resolução do problema, o programa principal assim como as funções utilizadas podem ser encontradas no módulo *lp_tempo*. O exemplo de entrada utilizado para as exemplificações pode ser encontrado em 'exemplos/programa/exemplo_trabalho_540.txt'. Quanto aos demais exemplos, estes descrevem outras instâncias de problemas *"Cutting Paper Rolls"*, e podem ser testados ao setar a variável de compilação MAXIMO para o valor que consta no nome do exemplo (make MAXIMO=X, assim como MAXIMO = 540 para 'exemplo_trabalho_540.txt').

2.1 LEITURA DOS DADOS

A função responsável pela leitura de dados é *constroiEstruturas()*, que definirá o dicionário *problema*, que conterá:

- **tempo_maximo:** Tempo, em minuto de execução sequencial máxima diária por máquina (540 minutos, como enunciado).
- **qtn_maquinas:** Quantidade de máquinas disponíveis para execução de processos m.
- **qtn_tempos:** Quantidade de classes de pedidos n.
- **pedidos:** Lista de tuplas contendo os pares (n_i, t_i), ou seja, quantidade e o tempo de cada classe.

Além de efetuar a leitura e a criação da estrutura *'problema'*, algumas verificações são feitas, como o tipo dos dados (strings contendo inteiros), quantidade de classes de pedido informadas e recebidas, se valor de tempos não é negativo ou extrapola o tempo máximo, entre outros.

2.1.1 Exemplo:

Para a entrada:

3 4

10 200

5 330

10 420

8 500

O dicioário *'problema'* terá a seguinte forma:

{

tempo_maximo: 540

qtn_maquinas: 3,

qtn_tempos: 4,

pedidos: [(10, 200), (5, 330), (10, 420), (8, 500)],

}

2.2 CRIAÇÃO DE PADRÕES

A partir das especificações em *'problema'*, a função *gerarMatrizPadroes()*, efetuará a geração de uma 'matriz_padroes' cujas linhas representam os padrões e as colunas os valores correspondentes à composição dos padrões (coeficientes a_{ji}, com $0 \leq j \leq |P| - 1$ e $0 \leq i \leq n - 1$).

Para gerar a matriz, sendo *tempo_maximo* o limite superior de minutos-máquina/dia (540), tempos a lista de tempos t_i, com $1 \leq i \leq n$, e menor_tempo sendo $mínimo(\{t_1, t_2, ..., t_n\})$:

Algoritmo 1: gerarMatrizPadroes(tempo_maximo, tempos[ ], menor_tempo)

```
Result: Matriz com padrões (linhas) seus e coeficientes (colunas)
foreach bolsa em bolsas do
    remanescente := tempo_maximo - somaElementos(bolsa);
    if menor_tempo > remanescente ≥ 0 then
        Novo padrão maximal encontrado, adicioná-lo a matriz_padroes;
        Remover bolsa bolsas;
    end
end
bolsas_tmp := ∅ ;
while bolsas <> ∅ do
    foreach bolsa em bolsas do
        foreach tempo em tempos do
            remanescente := tempo_maximo - (somaElementos(bolsa) + tempo);
            if remanescente ≥ menor_tempo then
                bolsa ainda pode ser preenchida, inserir (bolsa ∪ {tempo}) em
                 bolsas_tmp;
            end
            if menor_tempo > remanescente ≥ 0 then
                Novo padrão maximal encontrado, adicioná-lo a matriz_padroes;
            end
        end
    end
    bolsas := bolsas_tmp;
    bolsas_tmp := ∅;
end
```

2.2.1 Exemplo:

Para o mesmo exemplo utilizado anteriormente, após passar por este processo teremos a seguintes entradas *tempo_maximo*, *tempos*, *menor_tempo*, e *matriz_padroes* como saida:

tempo_maximo: 540,

tempos: [200, 330, 420, 500],

menor_tempo: 200,

matriz_padroes: $\begin{bmatrix} [0,0,1,0] \\ [0,0,0,1] \\ [2,0,0,0] \\ [1,1,0,0] \end{bmatrix}$

2.3 SAÍDA EM FORMATO LP

Nesta última etapa, a partir da leitura dos dados em *problema* e da *matriz_padrões*, será construído e impresso a formulação do problema a ser resolvido em formato *lp* do programa *lp_solve*(BERKELAAR; EIKLAND; NOTEBAERT, 2021).

A função objetivo deve refletir o modelo proposto em Função Objetivo, logo, será definida uma lista de variáveis x_j, $j[0..|P|-1]$, ou seja, a quantidade de linhas de *matriz_padroes*. Sendo assim será impresso: :

$$\text{min: x0 + x1 + x2 +... + xj ;}$$

Quanto às restrições, verifica-se para cada pedido (n_i, t_i), $1 \leq i \leq n$, quais os padrões (linhas de *matriz_padroes*) este pedido é contemplado. Imprimindo-se assim o somatório das variáveis do padrão que contemplam o pedido (x_j) multiplicadas pelo coeficiente atrelado. Tal expressão deverá ser maior ou igual a quantidade solicitada de cada pedido, dessa forma, cada uma das restrições terá o formato:

$$c_{0i} \times x0 + c_{1i} \times x1 + ... + c_{ji} \times xj \geq q_i;$$
$$\text{,com } 0 \leq j \leq |P| - 1 \text{ e } 0 \leq i \leq n - 1$$

Sendo c_{ji} o valor registrado na matriz *matriz_padroes* em linha i e coluna j. Os valores cujo coeficiente seja 0 são ignorados na impressão pois, de fato, não interfere para a restrição. Quanto às restrições de $xj \geq 0, \forall j \in [0 \,..\, (|P|-1)]$ não são necessárias de ser informadas, já que o programa supõe que todas as variáveis apresentadas possuam valores positivos.

2.3.1 Exemplo:

Para o exemplo anterior, teremos a seguinte saída:

min: x0 + x1 + x2 + x3 ;

2 x2 + 1 x3 >= 10 ;

1 x3 >= 5 ;

1 x0 >= 10 ;

1 x1 >= 8 ;

Referências

BERKELAAR, M.; EIKLAND, K.; NOTEBAERT, P. *LP file format*. 2021. [Acessado em 21 Junho 2021]. Disponível em: <http://lpsolve.sourceforge.net/5.5/lp-format.htm>. Citado 2 vezes nas páginas 1 e 5.

MATOUEK, J.; GäRTNER, B. *Understanding and Using Linear Programming (Universitext)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 3540306978. Citado na página 1.