Aluno: Lucas Block Villatore

Descrição do problema

Escalonamento de uso no tempo

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 maisde um destes usos podem ser alugados num mesmo dia para uma mesmam´aquina, se a soma dos tempos for menor que as 9 horas do expediente.Cada cliente pede quanto tempo, em minutos, vai usar uma m´aquina. Essetempo deve estar entre 0 e 540 minutos.

A empresa tem maquinas. Ao receber um conjunto de pedidos, o gerente da empresa precisa escalonar em qual maquina e em qual dia cada uso vai ser feito.

Considere que a demanda (pedidos) eé dada por um conjunto de **n** pares(ni, ti), onde **ni** é o numero de pedidos de tempo ti, com 1 ≤ i ≤ n.

Queremos minimizar o número de dias necessario para atender aos pedidos da demanda

Modelagem

Nesse problema, precisamos gerar dinamicamente as variáveis das equações.

Para descobrir essas variáveis, precisamos listar todas as possibilidades de escalonamento de tempo de máquinas de acordo com a entrada dada, para esse problema, pegamos somente as possibilidades no qual o tempo ocioso das máquinas seja menor que o menor tempo solicitado de um pedido. Com essas variáveis, conseguimos relacionar os custos e as restrições de todos os pedidos em relação ao tempo disponível de máquina.

O custo está relacionado com a quantidade de possibilidades, temos que o custo é:

$$C = \sum_{i=1}^{|P|} xi$$

onde |P| é a quantidade de possibilidades.

Extraímos as equações de restrições a partir das varíaveis.
Para cada pedido, formalizamos uma restrição na seguinte fórmula:

$$n1 * x1 + n2 * x2 + \ldots + nm * xm >= ni$$

Implementação

O trabalho é dividido em quatro partes

- Leitura
- Geração de variáveis
- Geração de equações
- Retorno das equações

Leitura:
Parte trivial do trabalho, somente leitura dos campos

Geração de variáveis:
A partir dos campos lidos na etapa anterior, o algoritmo permuta combinações utilizando recursão para listar as melhores possibilidades de dividir o custo do escalonamento de tempo entre as máquinas disponíveis.

Exemplo:
A entrada abaixo vai me gerar a seguinte permutação de horários

```
$ python3 main.py | lp_solve

3 4
10 200
5 330
10 420
8 500
```

P1: 2 * x1
P2: 1 * x1 + 1 * x2
P3: 1 * x3
P4: 1 * x4

Cada possibilidade é uma maneira de distribuir o tempo das máquinas entre os valores das entradas

Geração de equações:
A partir das variáveis geradas no passo anterior, conseguimos montar nossas equações de restrições

O algoritmo, irá gerar N equações, sendo que N é o número de pedidos.
Para cada possibilidade de permutação de pedidos, verifico se existe pedido existe na possibilidade do dia, caso exista eu adiciono o valor da possibilidade na equação de restrição.

Por exemplo, utilizando a entrada do passo anterior, uma das equações que o algoritmo me retornará é:

c4: 2x1 + 1x2 + 0x3 + 0x4 >= 10;
(Porque no P1 existe 2 * x1 e em P2 existe também o 1 * x1)

Retorno das equações

Parte trivial do trabalho, somente retorno para o terminal as funções com a formatação de entrada para o lp_solve

Exemplos de entrada e saída

Entrada:

```
$ python3 main.py | lp_solve

3 4
10 200
5 330
10 420
8 500
```

Saída redirecionada para o lp_solve:

```
min : x1  + x2  + x3  + x4 ;
c1: 0x1 + 0x2 + 0x3 + 1x4 >= 8;
c2: 0x1 + 0x2 + 1x3 + 0x4 >= 10;
c3: 0x1 + 1x2 + 0x3 + 0x4 >= 5;
c4: 2x1 + 1x2 + 0x3 + 0x4 >= 10;
x1  >= 0;
x2  >= 0;
x3  >= 0;
x4  >= 0;
```

Retorno do lp_solve:

```
Value of objective function: 25.50000000

Actual values of the variables:
x1                            2.5
x2                              5
x3                             10
x4                              8
```

Como executar

No terminal executar

- Para mostrar as equações e restrições no terminal

```
$ python3 main.py
```

- Para redirecionar as equações e restrições para o lp_solve

```
$ python3 main.py | lp_solve
```