Relatório do Primeiro Trabalho
Disciplina de Otimização

Lucas Sulzbach

Dezembro de 2020

1 Introdução ao Problema

Enunciado 1.1 (Escalonamento de tarefas em máquinas) *Uma empresa tem um conjunto de tarefas (jobs) T para serem executadas em um determinado período (por exemplo um mês). Para a execução destas tarefas a empresa conta com um conjunto M de máquinas (próprias ou alugadas). Cada tarefa $t \in T$ consome uma quantidade de horas de cpu, h_t. Cada máquina $m \in M$ tem um custo de operação (aluguel) de c_m reais por hora, tem um tempo máximo (u_m) em horas que pode ser usada, e pode executar um subconjunto de tarefas $S_m \subseteq T$. Queremos minimizar os custos e executar todas as tarefas.*

2 Modelagem

Dado que para cada máquina existe um subconjunto do total de tarefas que ela pode executar, existe também, para cada tarefa, um subconjunto do total de máquinas que podem executá-la.

Definição 2.1 (T_m) *Para toda máquina $m \in M$ existe um conjunto $T_m \subseteq T$ tal que T_m é o conjunto das tarefas que podem ser executadas pela máquina m (Equivalente ao conjunto S_m definido em 1.1).*

Definição 2.2 (M_t) *Para toda tarefa $t \in T$ existe um conjunto $M_t \subseteq M$ tal que M_t é o conjunto das máquinas que podem executar a tarefa t.*

O problema consiste em descobrir como escalonar todas as tarefas de T nas máquinas M de maneira a minimizar o custo da locação das máquinas utilizadas. Existem diversas possíveis combinações do tipo tarefa t a ser executada $\times$ máquina m alocada para a execução da tarefa. Definimos a seguir uma combinação deste tipo como uma *Locação*.

Definição 2.3 (Locação de uma máquina para execução de uma tarefa) *Dadas uma tarefa $t \in T$ e uma máquina $m \in M$ tal que $t \in T_m$ e $m \in M_t$, definimos uma locação de m para a execução de t como uma tupla (m, t).*

Como um problema de escalonamento, as incógnitas a serem descobertas podem ser expressas como as medidas de tempo de cada locação.

Definição 2.4 (Tempo de locação) *Dada uma locação* (m, t)*, o tempo de locação, em horas, é denotado por* $x_{m,t}$*.*

2.1 Restrições

Sabendo o que representam as incógnitas que queremos descobrir, podemos começar a modelar as restrições do programa linear que resolve o problema. Nesta modelagem foram utilizadas três tipos de restrição. Verbalmente, elas podem ser expressas como:

1. Enquanto medidas de tempo, as variáveis podem assumir o valor nulo (no caso de uma máquina elegível para a execução de uma tarefa não ser alocada), porém nunca podem assumir valores negativos;

2. Os tempos máximos de uso de cada máquina devem ser respeitados;

3. A soma dos tempos de alocação de uma tarefa em cada máquina deve equivaler ao tempo total de execução da tarefa.

Para modelar a primeira categoria de restrições, escolhamos o conjunto das máquinas disponíveis M. Para cada máquina $m \in M$ há o conjunto de tarefas que ela pode executar T_m. Para cada máquina $m \in M$ e para cada uma de suas respectivas tarefas $t \in T_m$ temos uma locação (m, t), cujo tempo $x_{m,t}$ é maior ou igual a zero.

$$\forall m \in M, t \in T_m \; x_{m,t} >= 0 \tag{1}$$

Para a segunda parte, escolhamos novamente as máquinas de M e formemos as locações pareando-as com suas respectivas tarefas de T_m. Para cada máquina $m \in M$, a soma dos tempos de locação $x_{m,t}$ para cada tarefa $t \in T_m$ não deve exceder o tempo máximo de uso u_m fornecido no enunciado do problema.

$$\forall m \in M \sum_{t \in T_m} x_{m,t} <= u_m \tag{2}$$

Na última restrição, escolhamos desta vez o conjunto de tarefas T. Para cada tarefa $t \in T$ existe o conjunto de máquinas que podem executá-la M_t. A soma dos tempos de alocação da tarefa $x_{m,t}$ em cada máquina $m \in M_t$ deve equivaler ao tempo necessário para execução h_t fornecido no enunciado.

$$\forall t \in T \sum_{m \in M_t} x_{m,t} = h_t \tag{3}$$

2.2 Função objetivo

Escolhamos o conjunto de máquinas M. Para cada máquina $m \in M$ há o conjunto de tarefas que ela pode executar T_m. Para cada máquina $m \in M$ e para cada uma de suas respectivas tarefas $t \in T_m$ temos uma locação (m, t), cujo tempo, em horas, é denotado por $x_{m,t}$. Também para cada máquina m foi fornecido no enunciado um custo por hora c_m de uso da máquina. Obtemos o custo final de cada locação (m, t) multiplicando seu respectivo tempo $x_{m,t}$ pelo custo c_m. Para obter o custo total, basta somar os custos de cada locação de m, que por sua vez são obtidos através da soma dos custos finais de cada locação de m para cada tarefa t.

$$\sum_{m \in M} \sum_{t \in T_m} c_m \times x_{m,t} = ? \tag{4}$$

3 Implementação

O pacote *lp_scheduler* foi implementado em *Python 3*, fazendo uso da biblioteca *PuLP* para a resolução do programa linear em si. Não detalharemos aqui as rotinas para processamento da entrada e da saída do programa. Vamos focar apenas na parte da modelagem, encontrada no método *lp_scheduler.LP.model*.

O estado da aplicação no iníco da modelagem está descrito no diagrama de classes encontrado na Figura 1. A entrada foi processada, dando origem a uma lista de objetos *Task* que representa o conjunto T. Cada instância de *Machine* (ou cada $t \in T$) possui o atributo *runTime* que representa h_t. Foi criada também uma lista de objetos *Machine* que representa o conjunto M. Cada instância de *Machine* (ou cada $m \in M$) possui os atributos *cost* e *maxTime* que representam, respectivamente, c_m e u_m. Cada m também está associada a uma lista de *Tasks* representando T_m por meio do atributo *tasks*. Tanto T quanto M estão associadas a um *LP*.

Junto ao diagrama da Figura 1, o diagrama de interação da Figura 2 descreve exatamente como o código do projeto é uma implementação da modelagem descrita neste relatório.

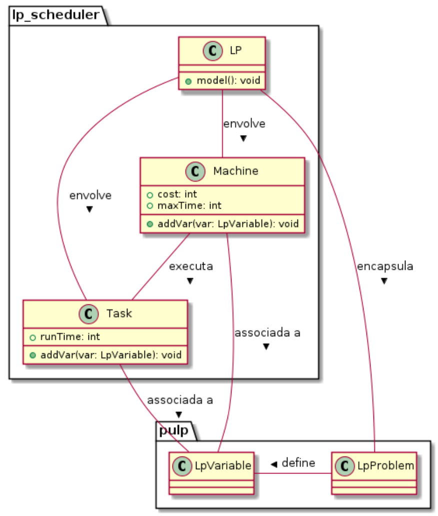

Figura 1: Diagrama de Classes de Visão para a operação *lp_scheduler.LP.model* (Desenvolvido com *PlantUML*)

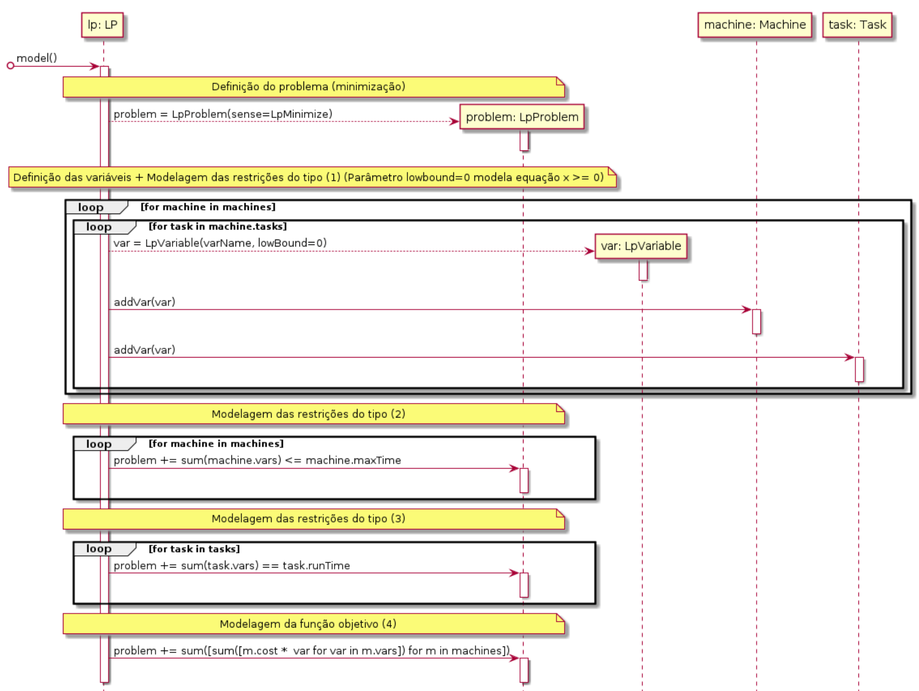

Figura 2: Diagrama de Interação para a operação *lp_scheduler.LP.model* (Desenvolvido com *PlantUML*)