Estudo e modelagem de soluções de otimização discreta para grades horárias

Aluno: Nilo Cesar Teixeira

Orientador: Prof. Alfredo Goldman

Instituto de Matemática e Estatística - IME/USP

nilo.teixeira@gmail.com

1. Introdução

Em problemas de otimização discreta, busca-se uma solução x^* em um conjunto discreto F que otimiza (maximiza ou minimiza) uma função objetivo $c(x)$, definida para todo $x \in F$ [1].

Uma maneira sistemática de modelar uma ampla classe de tais problemas é expressá-los como problemas de programação linear inteira, e resolvê-los através de alguma biblioteca ou software (denominado "*solver*"), após descrever o problema na linguagem ou interface fornecida pelo "*solver*".

Neste trabalho, tal metodologia foi aplicada ao problema de alocação de recursos operacionais de acordo com uma grade horária, problema recorrente em instituições de ensino, onde os docentes representam o recurso, e a grade é composta pelos horários de aulas.

2. Conceitos

2.1 Álgebra linear

Um vetor $y \in \Re^n$ é uma **combinação linear** de vetores $x_i \in \Re^n,\ i \in \{1, ..., k\},\ k \in \mathbb{N}$, se existem escalares $\alpha_i \in \Re$ tais que $y = \sum_{i=1}^{k} \alpha_i x_i$.

Uma combinação linear é chamada de:

- **afim**, se $\sum_{i=1}^{k} \alpha_i = 1$;
- **cônica** (*não confundir com côncava*), se $\alpha_i \geq 0$;
- **convexa**, se for afim e cônica.

O **fecho linear** (ou "*span*", ou **subespaço linear**) é o conjunto formado por todas as possíveis combinações lineares de tais vetores. Analogamente, define-se os termos **subespaço afim**, **cone**, e **fecho convexo**, para os itens acima [2].

Um subconjunto $P \subseteq \Re^n$ é denominado **poliedro** se

$$P = \{x \in \Re^n \mid Ax \leq b,\ A \in \Re^{m\,x\,n},\ b \in \Re^m\}$$

, e é denotado por $P(A, b)$.

Um poliedro da forma $\{x \in \Re^n \mid a'x \leq \alpha,\ \alpha \in \Re,\ a \in \Re^n\}$ é denominado **semi-espaço**. Logo, segue que $P(A, b)$ é uma intersecção de semi-espaços.

2.2 Programação linear

Define-se então um problema de programação linear PL como:

$$PL = \{\max c'x \mid x \in P(A, b),\ c \in \Re^n,\ x \geq 0\}$$

E um problema de **programação linear inteira** PLI como:

$$PLI = \{PL \mid x \in \mathbb{Z}^n\}$$

Denominamos $c'x$ a **função objetivo**, e x, **variável de decisão** do PL. O **fecho inteiro** de $P(A, b)$ equivale ao seu fecho convexo para $x \in PLI$.

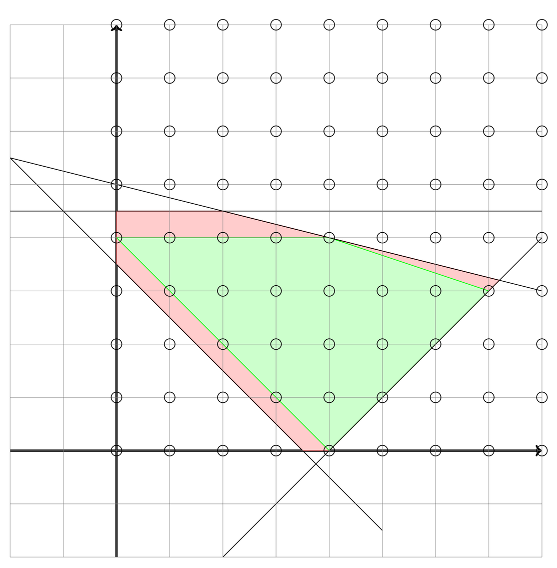

Figura 1: *Poliedro (em vermelho) e seu fecho inteiro (em verde)*

2.3 Teoria dos Grafos

Um **digrafo** $DG(V, A)$ consiste de um conjunto finito não-vazio V de elementos chamados vértices e um conjunto A de **pares ordenados** de elementos distintos de V, chamados arestas. Um **grafo** $G(V, A') \supseteq DG(V, A)$ pode ser modelado a partir de um digrafo, construindo para cada aresta $a \in A$, uma aresta a^* no sentido contrário, e adicionando este **par** de arestas ao conjunto A'.

Um **emparelhamento** em um grafo é um conjunto de arestas $X \subseteq A$ onde nenhum par de arestas possui vértices em comum. Um emparelhamento é **maximal** se, ao acrescentar qualquer outra aresta de A em X, este conjunto deixa de ser um emparelhamento, e **máximo** se, para qualquer outro emparelhamento $X' \neq X$, $|X'| \leq |X|$.

3. Escolha do *solver*

Inicialmente, a modelagem deste trabalho foi formulada em **Lingo** [4], pois foi o primeiro *solver* com que tomei contato que dispunha de uma linguagem de modelagem e que resolvia problemas de programação inteira.

Uma alternativa *open-source*, com desempenho equivalente e com uma linguagem de modelagem superior, foi encontrada após algum tempo: o **GNU Linear Programming Kit** (glpk) [3], que substituiu o Lingo neste trabalho.

A título de exemplo, apresentamos um modelo que encontra, se existente, uma ocorrência de um emparelhamento máximo em um grafo.

4. Modelando um problema linear inteiro

4.1 Modelando restrições

Uma variável de decisão $x_k \in \{0, 1\}$, pode compôr restrições no modelo de diversas formas. Quando ela representa a ocorrência ou não de um evento, temos comumente alguns padrões de modelagem [1]:

- No máximo um entre os n eventos ocorre:
$$\sum_{k=1}^{n} x_k \leq 1$$
- Ou ambos eventos ocorrem (e.g. para $n = 2$), ou nenhum ocorre:
$$x_1 = x_2$$
- Se um evento x_1 ocorre, o evento x_2 necessariamente ocorre:
$$x_1 \leq x_2$$

4.2 Exemplo: Emparelhamento máximo

```
#Vértices
set V;

#Arestas
param A {i in V, j in V}, binary;

#Variável de decisão
var X {v1 in V, v2 in V}, binary;

#Função objetivo
maximize obj: sum {v1 in V, v2 in V} X[v1,
  v2] * A[v1,v2];

## Restrições ##
#Cada aresta deve ter duas pontas distintas
check {v in V} : A[v, v] = 0;

#Modelando grafo a partir de um digrafo
check {v1 in V, v2 in V} : A[v1,v2] = A[v2,
  v1];

#Arestas duplas, ou nenhuma aresta
s.t. r1 {v1 in V, v2 in V} : X[v1,v2] * A[
  v1,v2] = X[v2,v1] * A[v2,v1];

#Modelando emparelhamento
s.t. r2 {v1 in V} : sum {v2 in V} (X[v1,v2]
   * A[v1,v2] + X[v2,v1] * A[v2,v1]) <= 2;

#Resolve o problema linear
solve;

#Mostra resultado
display X;

data;

#Definição de parâmetros
set V := V1 V2 V3 V4;

param A : V1 V2 V3 V4 :=
       V1 0  1  0  0
       V2 1  0  1  0
       V3 0  1  0  1
       V4 0  0  1  0;

#Resposta para esta instância:
  {{1,2},{3,4}}
end;
```

5. Modelando uma grade horária

Na modelagem proposta, temos a seguinte função objetivo:

$$f(H, Peso) = H^{m,\,sl}_{hr,\,d} \ * \ Peso^{m,\,sl}_{hr,\,d}$$

$$(H^{m,\,sl}_{hr,\,d} \in \{0, 1\}, Peso^{m,\,sl}_{hr,\,d} \in \mathbb{Z})$$

Os índices representam o seguinte:

- **m** $\in MATERIA$: Matéria lecionada;
- **sl** $\in SALA$: Sala de aula;
- **hr** $\in HORARIO$: Horário da aula;
- **d** $\in DIA$: Dia da semana.

Assim, formulamos o seguinte PLI, sujeito a restrições de viabilidade:

$$\max \sum_{hr,\,d}^{m,\,sl} H^{m,\,sl}_{hr,\,d} \ * \ Peso^{m,\,sl}_{hr,\,d}$$

Assim, sugerindo $Pesos$ atrelados às variáveis de decisão $H^{m,\,sl}_{hr,\,d}$, podemos indicar preferências na grade horária.

6. Modelando restrições

No modelo, existem restrições de viabilidade, e restrições de escolha das possíveis combinações dos índices. Observemos um exemplo:

6.1 Restrição de viabilidade

1. **Restrição 1:** Uma sala não pode ter duas disciplinas lecionadas ao mesmo tempo

   para cada $sl \in SALA$
   para cada $hr \in HORARIO$
   para cada $d \in DIA$

$$\sum_m H^{m,\,sl}_{hr,\,d} \leq 1$$

7. Conclusão

A modelagem proposta supõe alguns requisitos limitantes, como o fato de não poder haver mais de um professor da mesma matéria (o que será exemplificado e demonstrado na apresentação), mas considerando a evolução da pesquisa até chegar em uma modelagem que descrevesse e resolvesse o problema, mesmo com tais limitações, já foi para o autor um motivo de contentamento com o resultado do trabalho.

Referências

[1] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusetts, USA, 3ª impressão, 1997.

[2] Carlos Eduardo Ferreira and Yoshiko Wakabayashi. *Combinatória Poliédrica e Planos de Corte Faciais.* UNICAMP, Instituto de Computação, Campinas, SP, Brasil, 1996.

[3] Andrew Makhorin. Glpk (gnu linear programming kit). http://www.gnu.org/s/glpk/, 2011. [Online; acessado em 01/Nov/2011].

[4] Lindo Systems. Lingo optimization modeling software. http://www.lindo.com/, 2011. [Online; acessado em 01/Set/2011].