Modelo PLI para Alocação de Professores da UFC-Quixadá

Francisco Sergio de Freitas Filho
Universidade Federal do Ceará (UFC)
Av. José de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará 63902-580
sergiofilhoce07@gmail.com

Lucas Ismaily Bezerra Freitas
Universidade Federal do Ceará (UFC)
Av. José de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará 63902-580
ismailybf@ufc.br

Críston Pereira de Souza
Universidade Federal do Ceará (UFC)
Av. José de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará 63902-580
criston@ufc.br

RESUMO

A atribuição de professores em disciplinas é uma tarefa complexa, principalmente quando existe muito compartilhamento de professores entre cursos, como é o caso da UFC-Quixadá. Neste trabalho apresentamos uma formulação em programação linear inteira que busca maximizar a soma das preferências dos professores, e atende um conjunto de restrições próprias da instituição. Testamos o modelo utilizando dados reais da alocação do segundo semestre de 2016, obtendo solução ótima em 0,07 segundos, e com qualidade cerca de 20% maior que a alocação adotada neste semestre (feita manualmente). Além disso, construímos instâncias aleatórias respeitando as mesmas proporções da instância real, e concluímos que é possível encontrar soluções ótimas para instâncias cerca de 4 vezes maiores que nossa atual alocação. Além disso, verificamos que é possível resolver instâncias maiores aumentando o número de disciplinas que cada professor podem lecionar.

PALAVRAS CHAVE. Alocação de Professores e Disciplinas. Programação Linear Inteira.

ABSTRACT

The assignment of teachers in disciplines is a complex task, especially when there are many teachers shared between courses, as is the case of UFC-Quixadá. In this work we present an integer programming formulation that seeks to maximize the sum of the preferences of the teachers, and attend a set of restrictions of the institution. We tested the model using real data from the second half of 2016, obtaining an optimal solution in 0,07 seconds, and with quality about 20% greater than the allocation adopted this semester (done manually). In addition, we construct random instances respecting the same proportions of the real instance, and we conclude that it is possible to find optimal solutions for instances that are almost 4 times larger than our current allocation. In addition, we find that it is possible to solve larger instances by increasing the number of courses that each teacher can teach.

KEYWORDS. Timetabling Problem. Integer Linear Programming.

1. Introdução

As instituições de ensino se deparam periodicamente com dois problemas em comum: o problema de alocação de disciplinas em horários e o de alocação de professores em disciplinas, seguindo uma séries de restrições em ambos os problemas. Além disso, sabe-se que os professores podem ter preferências por determinadas disciplinas e dias da semana para ministrar suas aulas, e que recursos, como salas e materiais, podem ser limitados e compartilhados. Portanto, devem ser bem gerenciados para que não hajam conflitos. Essa categoria de problema de alocação faz parte de uma área bastante explorada, conhecida como *Timetabling Problems*, e apresenta diversas formulações [Pillay, 2014]. Na literatura, o problema foi abordado de diversas formas, alguns exemplos são: algoritmos genéticos [Alves et al., 2017], programação por restrição [Kaewchanid e Wangmaeteekul, 2016] e programação inteira [Kristiansen et al., 2015], [Bakır e Aksop, 2008], [Daskalaki et al., 2004]. Apesar de ser uma área muito estudada e explorada, os problemas contam com muitas particularidades que geralmente os tornam problemas NP-Difíceis [Carter e Laporte, 1997].

Uma solução para o problema deve atender a todas as restrições de viabilidade adotadas pela instituição e considerar as preferências dos professores por determinadas disciplinas e dias da semana. Essa atividade, quando realizada de forma manual, demanda muito tempo e trabalho de seus realizadores, tornando-se inapropriada em função do crescimento das instituições de ensino. Por sua vez, soluções computacionais tem ganhos na qualidade de resposta, além de reduzir o número de profissionais envolvidos e tempo gasto no processo manual [Xavier et al., 2013].

No Campus da Universidade Federal do Ceará em Quixadá (UFC-Quixadá), a direção do Campus, juntamente com as coordenações de cursos, se reúnem semestralmente com o intuito de definir a grade de horários semanais de cada disciplina e professores responsáveis por ministrá-las. Com o surgimento de novos cursos e, como consequência, maior número de disciplinas e professores, o processo de alocação de forma manual torna-se cada vez mais trabalhoso, sendo de extrema importância o desenvolvimento de ferramentas que possam automatizar ou, pelo menos, auxiliar nesse processo. Atualmente o Campus contém seis cursos de graduação e sessenta e dois professores.

Este trabalho apresenta uma solução para o Problema de Alocação de Professores em Disciplinas (PAPD) para a UFC-Quixadá, com o objetivo de auxiliar no processo de elaboração de horários semestrais. Para isso, foi definido e implementado um modelo de programação linear inteira com base nos critérios requeridos pela UFC-Quixadá.

O restante deste artigo está dividido como segue. A Seção 2 apresenta a definição do problema e as restrições adotadas. A Seção 3 apresenta o modelo de programação inteira proposto para o problema. Resultados obtidos e considerações sobre este trabalho são apresentados, respectivamente, nas Seções 4 e 5.

2. Definição do Problema

Neste trabalho consideramos o PAPD para o cenário da UFC-Quixadá, e partimos do pressuposto que a alocação de disciplinas em horários é pré-definida. Nas definições a seguir utilizamos o conceito de *slot*, que consiste em um intervalo de duas horas onde ocorre uma aula. As aulas são realizadas de segunda-feira a sexta-feira, nos turnos da manhã, tarde e noite, e com dois *slots* por turno (Figura 2). Além disso, as restrições abaixo devem ser atendidas:

- Um professor tem uma quantidade mínima e máxima de horas que deve trabalhar semanalmente;
- São definidos dois grupos de horários: segunda-feira a quinta-feira e terça-feira a sexta-feira, e todas as disciplinas atribuídas a um professor devem ter seus *slots* no mesmo grupo de horários. Ou seja, todo professor tem a segunda-feira ou sexta-feira sem disciplinas alocadas; permitindo assim, por exemplo, que ele possa participar de reuniões na sede da Universidade em Fortaleza-CE;

Figura 1: *Slots* semanais.

		Seg	Ter	Qua	Qui	Sex
Manhã		S_{00}	S_{01}	S_{02}	S_{03}	S_{04}
		S_{05}	S_{06}	S_{07}	S_{08}	S_{09}
Tarde		S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
		S_{15}	S_{16}	S_{17}	S_{18}	S_{19}
Noite		S_{20}	S_{21}	S_{22}	S_{23}	S_{24}
		S_{25}	S_{26}	S_{27}	S_{28}	S_{29}

- Alguns pares de professores podem solicitar alocação em um mesmo grupo de horários, ou seja, que ambos sejam alocados no início da semana ou ambos alocados no final da semana. Isso ocorre principalmente entre casais de professores;
- Se um professor for alocado para uma disciplina no último *slot* da noite, ele não deve ser alocado para uma disciplina do primeiro *slot* da manhã do dia seguinte;
- Um professor não pode ser alocado em duas disciplinas que compartilham *slots*;
- Um professor só pode ser alocado em no máximo dois turnos por dia;
- Uma disciplina tem uma quantidade determinada de professores distintos que devem ser alocados a ela;
- As reuniões envolvendo todos os professores (seminários) ocorrem nos *slots* S_{07} e S_{12}. Portanto, todo professor deve ter pelo menos um destes *slots* livre, ou seja, não pode ter aulas alocadas em ambos os *slots*;
- Existem *slots* destinados às reuniões de Colegiado de Curso, Núcleo Docente Estruturante, Conselho de Campus, e outros grupos. Cada reunião requer a disponibilidade dos professores que a compõe. Desse modo, dado uma reunião, os professores que a compõe não podem ser alocados para disciplinas que compartilham *slots* com a reunião;
- Um professor só pode ser atribuído a disciplinas que está apto a lecionar.

3. Modelo em Programação Linear Inteira

Descrevemos a seguir a entrada, as variáveis e as restrições do modelo em Programação Linear Inteira para PAPD da UFC-Quixadá.

3.1. Entrada

O modelo possui os seguintes parâmetros de entrada:

- D é o conjunto de disciplinas ofertadas;
- P é o conjunto de professores;
- $S = \{00, 01, 02, \ldots, 29\}$ é o conjunto de *slots*;
- $S_s \subset S$ é o conjunto de *slots* alocados para seminários;
- R é o conjunto de reuniões, onde cada reunião $r \in R$ possui dois conjuntos R_p e R_s que representam, respectivamente, o conjunto de professores e o conjunto de *slots* da reunião r;

- C é o conjunto de pares de professores que devem ser alocados no mesmo grupo de horário, onde cada $c \in C$ consiste em uma tupla $(c_1, c_2) \in P \times P$;
- $disciplinas : S \mapsto \mathcal{P}(D)$, tal que $disciplinas(s) = D_s$ é o conjunto de disciplinas atribuídos ao *slot* s;
- $c_{max} : P \mapsto \mathbb{N}$, tal que $c_{max}(p) = c_p^+$ é a quantidade máxima de *slots* que podem ser atribuídos ao professor p;
- $c_{min} : P \mapsto \mathbb{N}$, tal que $c_{min}(p) = c_p^-$ é a quantidade mínima de *slots* que podem ser atribuídos ao professor p;
- $f_p : P \times D \mapsto \mathbb{N}$, tal que $f_p(p, d) = p_{pd}$ é o valor de preferência do professor p pela disciplina d;
- $c_{prof} : D \mapsto \mathbb{N}$, tal que $c_{prof}(d) = n_d$ é a quantidade de professores que devem ser alocados para a disciplina d;
- $d_{prof} : P \mapsto \mathcal{P}(D)$, tal que $d_{prof}(p) = D_p$ é o conjunto de disciplinas que podem ser atribuídas ao professor p.

3.2. Variáveis

O modelo desenvolvido possui dois tipos de variáveis:

- x_{ps}: 1 se o professor p for atribuído a alguma disciplina no *slot* s, e 0 caso contrário.
- y_{pd}: 1 se o professor p for atribuído a disciplina d, e 0 caso contrário.

3.3. Formulação

A função objetivo visa encontrar uma solução que maximize a preferência global dos professores por disciplinas, ou seja,

$$\max \sum_{p \in P} \sum_{d \in D} y_{pd} \cdot p_{pd}$$

As restrições (1) e (2) garantem que os limites de cargas horárias dos professores sejam respeitados,

$$\sum_{s \in S} x_{ps} \geq c_p^-, \ \forall p \in P. \tag{1}$$

$$\sum_{s \in S} x_{ps} \leq c_p^+, \ \forall p \in P. \tag{2}$$

A restrição (3) garante que um professor seja alocado em apenas um dos grupos de horários,

$$x_{pi} + x_{pj} \leq 1, \ \forall p \in P, \ \forall i \in \{0, 5, \ldots, 25\}, \ \forall j \in \{4, 9, \ldots, 29\}. \tag{3}$$

A restrição (4) assegura que um professor não seja alocado para uma aula no último *slot* da noite e uma aula no primeiro *slot* da manhã do dia seguinte,

$$x_{p(25+j)} + x_{p(1+j)} \leq 1, \ \forall p \in P, j \in \{0, 1, 2, 3\}. \tag{4}$$

A restrição (5) garante que um professor só pode ser alocado em uma disciplina por *slot*, e que um professor alocado para uma disciplina será alocado para todos os *slots* dessa disciplina,

$$\sum_{d \in D_s} y_{pd} = x_{ps}, \ \forall p \in P, s \in S. \tag{5}$$

A restrição (6) assegura que um professor só pode ser alocado em no máximo dois turnos diários,

$$x_{p(i+l)} + x_{p(j+l)} + x_{p(k+l)} \leq 2,\ \forall p \in P, \forall i \in \{0,5\}, \forall j \in \{10,15\}, \forall k \in \{20,25\}, \forall l \in \{0,1,2,3,4\}. \tag{6}$$

A restrição (7) diz que quantidade de professores atribuídos a uma disciplina é igual a quantidade de professores solicitada para a mesma,

$$\sum_{p \in P} y_{pd} = n_d,\ \forall d \in D. \tag{7}$$

A restrição (8) certifica que todo professor não será atribuído a nenhuma disciplina em pelo menos um dos *slots* reservados para as reuniões com todos os professores,

$$\sum_{s \in S_s} x_{ps} \leq |S_s| - 1,\ \forall p \in P. \tag{8}$$

A restrição (9) assegura que um professor não será alocado para nenhuma disciplina em *slots* destinados às reuniões ao qual ele faz parte,

$$x_{ps} = 0,\ \forall r \in R,\ \forall p \in R_p,\ \forall s \in R_s. \tag{9}$$

As restrições (10) e (11) garantem que cada par de professores em C estão alocados no mesmo grupo de horários,

$$x_{c_1\,i} + x_{c_2\,j} \leq 1,\ \forall (c_1, c_2) \in C,\ \forall i \in \{0,5,\ldots,25\},\ \forall j \in \{4,9,\ldots,29\}. \tag{10}$$

$$x_{c_2\,i} + x_{c_1\,j} \leq 1,\ \forall (c_1, c_2) \in C,\ \forall i \in \{0,5,\ldots,25\},\ \forall j \in \{4,9,\ldots,29\}. \tag{11}$$

A restrição (12) assegura que um professor só é atribuído a uma disciplina que ele é apto a lecionar,

$$y_{pd} = 0,\ \forall p \in P,\ \forall d \in D - D_p. \tag{12}$$

E as restrições (13) e (14) são, respectivamente, restrições de integralidade das variáveis y_{pd} e x_{ps},

$$y_{pd} \in \{0,1\},\ \forall p \in P,\ \forall d \in D. \tag{13}$$

$$x_{ps} \in \{0,1\},\ \forall p \in P,\ \forall d \in D. \tag{14}$$

4. Resultados Experimentais

O modelo foi implementado utilizando o ambiente de desenvolvimento do QT e a biblioteca CPLEX 12.6.1, em sua configuração padrão para a linguagem C++. Essa biblioteca é fornecida pela IBM e possibilita modelagem e resolução de problemas de programação linear e programação inteira. Para o processamento dos dados que compõe a entrada, foi utilizada a linguagem Python 2.7.6. Todos os experimentos foram realizados em uma única máquina que dispõe da seguinte configuração: processador Intel Core i5-3337U (4x1.80GHz), 5.7 GB de memória RAM e sistema operacional Ubuntu 14.04 LTS.

Por conveniência, assume-se que um professor está apto a lecionar disciplinas que ele tem alguma preferência explícita. Desse modo, o número de disciplinas que um professor tem preferência também é a quantidade de disciplinas que ele está apto a lecionar. Também vale ressaltar que o modelo considera ofertas distintas da mesma disciplina como disciplinas distintas.

Na Seção 4.1 é apresentado um experimento utilizando dados reais. Experimentos e resultados com instâncias aleatórias são detalhados na Seção 4.2. É importante ressaltar que foi utilizado o tempo total de resposta dado pelo CPLEX (já considerando a soma do processamento de cada núcleo e pré-processamento) em todos os experimentos.

4.1. Experimento com Dados Reais

Neste experimento utilizamos dados reais referentes à alocação de disciplinas para o segundo semestre de 2016. Para construir as preferências dos professores, utilizamos o número de vezes que cada professor ministrou cada disciplina. Por exemplo, se um professor p lecionou a disciplina d cinco vezes, então $f_p(p, d) = 5$.

Nesta instância temos 55 professores e 108 disciplinas. O tempo de resposta do modelo implementado foi de 0,07 segundos. Para analisar a qualidade da resposta, foi comparado o valor da função objetivo da solução proposta com o valor da função objetivo da alocação utilizada na instituição no segundo semestre de 2016 (construída manualmente). Conforme observado na Tabela 1, a solução fornecida pelo modelo é cerca de 20% melhor que a alocação manual.

Tabela 1: Comparação entre os valores da função objetivo da solução proposta pelo modelo e da alocação utilizada pela UFC-Quixadá no segundo semestre de 2016.

Alocação	Valor da função objetivo
Alocação sugerida pelo modelo	475
Alocação empregada na UFC-Quixadá em 2016.2	394

4.2. Experimentos com Dados Aleatórios

Como o modelo produziu solução ótima em apenas 0,07 segundos para a instância real, produzimos instâncias aleatórias para avaliar o tamanho máximo da instância que permita encontrar solução viável dentro do tempo limite de duas horas (7200 segundos). As instâncias aleatórias respeitam as proporções da instância real: (i) 2,8 disciplinas por professor, (ii) 5% das disciplinas com um *slot*, 87% com dois *slots*, e 8% com três *slots*, (iii) carga horária mínima no conjunto $\{0, 4, 8\}$, e carga horária máxima no conjunto $\{4, 8, 12, 14\}$ (garantindo mínima menor ou igual à máxima), (iv) entre 7% e 22% das disciplinas podem ser lecionadas por um professor, com preferências variando de 1 até 10, (v) dois ou três *slots* escolhidos para as reuniões com todos os professores, (vi) dez reuniões, cada uma contendo de 3% a 20% dos professores, e (vii) 10% dos professores são divididos em pares, sendo que cada par pertence a um mesmo grupo de horário. Todos os sorteios foram feitos utilizando distribuição uniforme.

As instâncias foram divididas em dois conjuntos de testes: no Conjunto 1 (36 instâncias) cada professor pode ensinar entre 7% e 22% das disciplinas, e no Conjunto 2 (45 instâncias) cada professor pode ensinar entre 10% e 30% das disciplinas. Os resultados para os Conjuntos 1 e 2 estão nas tabelas 2 e 3, respectivamente.

No Conjunto 1, para a terceira instância com 50 professores e 140 disciplinas, o modelo verificou em 0,02 segundos que não existia solução viável. Na terceira instância com 210 professores e 588 disciplinas, o modelo excedeu o tempo limite de duas horas e encontrou solução viável com GAP (diferença relativa ao limite inferior fornecido pela relaxação linear) de $0,02\%$. Para o Conjunto 2, o modelo obteve solução ótima para todas as instâncias testadas.

Concluímos que é possível resolver instâncias maiores quando aumentamos o número de disciplinas que os professores podem ensinar. Ou seja, quando aumentamos as opções de disciplinas dos professores aumentamos também o espaço de busca, mas isso facilita a obtenção de soluções viáveis, permitindo assim podar mais rapidamente a árvore de enumeração do *branch-and-bound*. Mesmo para o caso com menos opções de disciplinas por professor, foi possível encontrar solução ótima para instâncias com 180 professores e 504 disciplinas, o que é cerca de 4 vezes maior que nossa instância real.

5. Conclusão e Considerações Finais

Este trabalho teve como objetivo automatizar o processo de alocação dos docentes em disciplinas da Universidade Federal do Ceará - Campus Quixadá. Para isso, foi desenvolvido um modelo de programação linear inteira que maximiza a preferência global de professores por disciplinas e atende a todas as restrições impostas pela instituição.

Tabela 2: Conjunto de Teste 1 - atribuindo para cada professor entre 7% e 22% das disciplinas como preferenciais.

Professores	Disciplinas	Tempo Inst. 1	Tempo Inst. 2	Tempo Inst. 3	Resultado ótimo
50	140	2.05	0.98	0.02	1 e 2
60	168	2.19	9.13	4.56	1, 2 e 3
70	196	7.23	19.16	8.29	1, 2 e 3
80	224	30.90	58.96	156.57	1, 2 e 3
90	252	30.48	130.36	13.17	1, 2 e 3
100	280	32.60	27.79	12.99	1, 2 e 3
110	308	171.11	58.39	13.47	1, 2 e 3
120	336	137.44	269.33	37.13	1, 2 e 3
130	364	222.70	563.37	776.05	1, 2 e 3
150	420	2028.12	358.45	723.56	1, 2 e 3
180	504	6834.10	2450.96	1175.44	1, 2 e 3
210	588	2109.01	1977.06	***TIMEOUT***	1 e 2

Tabela 3: Conjunto de Teste 2 - atribuindo para cada professor entre 10% e 30% das disciplinas como preferenciais.

Professores	Disciplinas	Tempo Inst. 1	Tempo Inst. 2	Tempo Inst. 3	Resultado ótimo
50	140	39.94	11.54	29.88	1, 2 e 3
60	168	9.14	33.28	85.89	1, 2 e 3
70	196	18.03	51.35	112.59	1, 2 e 3
80	224	20.06	119.40	139.76	1, 2 e 3
90	252	126.70	1870.87	194.81	1, 2 e 3
100	280	448.36	197.60	429.34	1, 2 e 3
110	308	360.27	153.75	37.51	1, 2 e 3
120	336	181.98	82.78	372.77	1, 2 e 3
130	364	207.44	2174.54	376.26	1, 2 e 3
150	420	764.29	2450.22	491.09	1, 2 e 3
180	504	1305.46	72.69	2675.35	1, 2 e 3
210	588	274.58	479.39	150.02	1, 2 e 3
250	700	1134.46	718.85	337.74	1, 2 e 3
300	840	725.26	1589.44	893.46	1, 2 e 3
400	1120	6365.12	2270.93	608.20	1, 2 e 3

Comparamos a solução fornecida pelo modelo com a alocação manual realizada no segundo semestre de 2016, e concluímos que o modelo é capaz de melhorar em cerca 20% as preferências do professores. Realizamos também experimentos com instâncias aleatórias que respeitam as mesmas proporções da instância real, e concluímos que o modelo é capaz de fornecer solução ótima para instâncias com 180 professores e 504 disciplinas, o que é bem maior que os 55 professores e 108 disciplinas da nossa última alocação. Neste último experimento, percebemos também que é possível resolver instâncias maiores aumentando o número de disciplinas que podem ser lecionadas por professor.

Referências

Alves, S. S., Oliveira, S. A., e Neto, A. R. R. (2017). A recursive genetic algorithm-based approach for educational timetabling problems. In *Designing with Computational Intelligence*, p. 161–175. Springer.

Bakır, M. A. e Aksop, C. (2008). A 0-1 integer programming approach to a university timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 37(1):41–55.

Carter, M. W. e Laporte, G. (1997). Recent developments in practical course timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, p. 3–19. Springer.

Daskalaki, S., Birbas, T., e Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135.

Kaewchanid, P. e Wangmaeteekul, P. (2016). Solving a teaching assistant timetabling using constraint based approach. In *2016 International Conference on Computational Intelligence and Applications (ICCIA)*, p. 56–60.

Kristiansen, S., Sørensen, M., e Stidsen, T. R. (2015). Integer programming for the generalized high school timetabling problem. *Journal of Scheduling*, 18(4):377–392.

Pillay, N. (2014). A survey of school timetabling research. *Annals of Operations Research*, 218(1): 261–293.

Xavier, B. M., da Silva, A. D., Vianna, D. S., Costa, H. G., e Coelho, W. B. (2013). Proposta de alocação de horários de professores e turmas em instituições de ensino superior utilizando uma heurística vns/vnd.