UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FRANCISCO SERGIO DE FREITAS FILHO

UMA ABORDAGEM PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES EM DISCIPLINAS UTILIZANDO PROGRAMAÇÃO LINEAR INTEIRA

QUIXADÁ – CEARÁ

2016

FRANCISCO SERGIO DE FREITAS FILHO

UMA ABORDAGEM PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES EM DISCIPLINAS UTILIZANDO PROGRAMAÇÃO LINEAR INTEIRA

Monografia apresentada no curso de Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Ciência da Computação. Área de concentração: Computação.

Orientador: Prof. Mrs. Lucas Ismaily Bezerra Freitas

Co-Orientador: Prof. Dr. Críston Pereira de Souza

QUIXADÁ – CEARÁ

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

F936a Freitas Filho, Francisco Sergio de.
Uma abordagem para o problema de alocação de professores em disciplinas utilizando programação linear inteira / Francisco Sergio de Freitas Filho. – 2016.
37 f. : il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2016.
Orientação: Prof. Me. Lucas Ismaily Bezerra Freitas.
Coorientação: Prof. Dr. Críston Pereira de Souza.

1. Universidade Federal do Ceará - Campus Quixadá. 2. Programação inteira. 3. Alocação. 4. Professores. 5. Otimização. I. Título.

CDD 004

FRANCISCO SERGIO DE FREITAS FILHO

UMA ABORDAGEM PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES EM DISCIPLINAS UTILIZANDO PROGRAMAÇÃO LINEAR INTEIRA

Monografia apresentada no curso de Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Ciência da Computação. Área de concentração: Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Mrs. Lucas Ismaily Bezerra Freitas (Orientador)
Campus Quixadá
Universidade Federal do Ceará – UFC

Prof. Dr. Críston Pereira de Souza (Co-Orientador)
Campus Quixadá
Universidade Federal do Ceará - UFC

Prof. Dr. Wladimir Araújo Tavares
Campus Quixadá
Universidade Federal do Ceará - UFC

Prof. Dr. Fábio Carlos Sousa Dias
Campus Quixadá
Universidade Federal do Ceará - UFC

A Deus.

Aos meus avós.

Aos meus pais, Sergio e Valéria.

AGRADECIMENTOS

A Deus por minha vida, família e amigos, e por permitir que tudo isso acontecesse.

À minha família pelos valores morais, apoio e excelente educação que me proporcionaram.

À minha namorada, Danielly, pelo companheirismo e pela paciência durante os momentos em que “foi abandonada” rs.

Agradeço a todos os professores do curso de ciência da computação pela dedicação e por me proporcionar o conhecimento.

Aos Professores Mrs. Lucas Ismaily Bezerra Freitas e Dr. Críston Pereira de Souza, pela excelente orientação e contribuição no desenvolvimento desse estudo.

Aos colegas da turma de graduação, pelas reflexões, críticas, ajudas e sugestões recebidas.

A todos que contribuíram de forma direta ou indireta para que isso acontecesse.

RESUMO

A atribuição de professores em disciplinas é uma tarefa que ocorre periodicamente em instituições de ensino. Devido a imensa quantidade de possíveis atribuições distintas de professores em disciplinas e restrições impostas que devem ser atendidas, esse processo de alocação é uma tarefa complexa e que, se realizada de forma manual, pode demandar muito tempo e esforço de quem a realiza. Este trabalho apresenta uma proposta de solução para o Problema de Alocação de Professores em Disciplinas, levando em consideração o processo adotado pela Universidade Federal do Ceará - Campus Quixadá e suas restrições. Além disso, são consideradas as preferências de professores por disciplinas. Para resolver o problema, foi formulado e implementado um modelo de programação linear inteira que atenda as restrições adotadas pela instituição e ainda maximize a preferência global dos professores por disciplinas a eles atribuídas. Ademais, a partir do modelo implementado, foram realizados e apresentados experimentos com dados aleatórios e com dados reais para análise do modelo.

Palavras-chave: Programação Linear Inteira. Alocação de Professores. Otimização.

ABSTRACT

The assignment of professors to disciplines is a task that occurs periodically in educational institutions. Due to the immense amount of possible different assignments of professors to disciplines and some constraints that must be met, this is a complex process and, if done manually, can require a lot of time and effort from the one who performs it. This study presents a proposal for a solution to the Timetabling Problem, taking into account the process adopted by Universidade Federal do Ceará - Campus Quixadá and its restrictions. Besides, the preferences of profesors by disciplines are considered. To solve the problem, an integer linear programming model was formulated, and implemented, that meets the restrictions adopted by the institution and also maximizes the professors' overall preference for the disciplines assigned to them. In addition, from the implemented model, experiments were performed with random and with real data for analysis of the model.

Keywords: Integer Linear Programming. Timetabling Problem. Optimization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

PL	Programação Linear
PLI	Programação Linear Inteira
PAPD	Problema de Alocação de Professores em Disciplinas
UFC - Quixadá	Universidade Federal do Ceará - Campus Quixadá

SUMÁRIO

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 (DODÓ, 2011). O problema de alocação de horários faz parte de uma área bastante explorada, conhecida como *Timetabling Problems*, e apresenta diversas formulações. Apesar de ser uma área muito estudada e explorada, os problemas contam com muitas particularidades que geralmente os tornam problemas NP-Díficeis.

É perceptível a exponencialidade do problema, uma vez que, para m professores e n disciplinas, tem-se m^n possibilidades, sendo possível notar o crescimento exponencial à medida que cresce o tamanho da instância. Em meio a tantas possibilidades de atribuições, 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 inviável e trabalhoso, sendo de extrema importância o desenvolvimento de ferramentas que possam automatizar ou, pelo menos, auxiliar nesse processo.

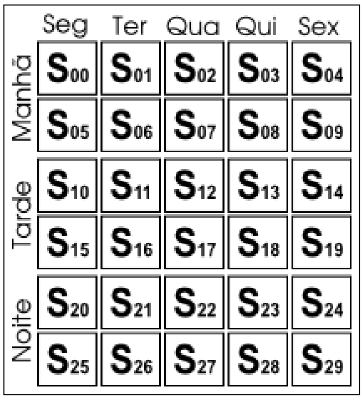
Este trabalho apresenta uma solução para o Problema de Alocação de Professores em Disciplinas (PAPD) para a UFC-Quixadá, com o objetivo de auxiliar no processo de elaboração de horários semestrais. Para isso, foi definido e implementado um modelo de programação linear inteira com base nos critérios requeridos pela instituição supracitada, apresentados na Seção 1.1.

Por ser uma área bastante estudada, é possível encontrar na literatura uma variedade de trabalhos com algumas semelhanças ao abordado nesse texto. Alguns trabalhos selecionados são apresentados na Seção 1.2. O restante deste estudo está dividido como segue. No Capítulo 2 são apresentados conceitos básicos que fundamentam a compreensão do texto. O Capítulo 3 apresenta o modelo de programação inteira proposto neste documento. Resultados obtidos e considerações sobre este trabalho são apresentados, respectivamente, nos Capítulos 4 e 5.

1.1 O Problema

O PAPD para a UFC-Quixadá, abordado neste documento, consiste na atribuição de professores em disciplinas, partindo do pressuposto que a programação dos *slots* das disciplinas já estão definidos. Nesse contexto, um ***slot*** é um espaço de tempo de duas horas de um turno e dia da semana, conforme mostrado na Figura 1. Por exemplo, o *slot* 00 (S_{00}) representa a primeira aula da manhã de segunda-feira.

Figura 1 – Representação visual dos *slots*.



Fonte: Elaborada pelo autor.

O trabalho foi elaborado levando em consideração os critérios adotados no processo de alocação na UFC-Quixadá. Desse modo, resolver o PAPD para a UFC-Quixadá é propor, dentre as inúmeras possibilidades, uma solução que maximize a preferência global dos docentes por disciplinas e satisfaça todas as restrições exigidas pela instituição supracitada, sendo elas:

- Um professor tem uma quantidade mínima e máxima de horas que deve trabalhar semanalmente;
- São definidos dois **grupos de horários**: segunda a quinta e terça a sexta, e todas as disciplinas atribuídas a um professor devem ter seus *slots* no mesmo grupo de horários;
- Os casais de professores devem ser alocados no mesmo grupo de horários;

- Se um professor for alocado para uma disciplina no último *slot* da noite, ele não pode ser alocado para uma disciplina do primeiro *slot* da manhã do dia seguinte;
- Um professor não pode ser alocados em duas disciplinas que compartilham *slots*;
- Um professor só pode ser alocados em, no máximo, dois turnos por dia;
- Uma disciplina tem uma quantidade determinada de professores distintos que devem ser alocados a ela;
- Existem *slots* destinados a seminários de professores, de modo que todo professor tem que estar disponível em pelo menos um desses *slots*;
- Existem *slots* destinados a reuniões, e 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.

1.2 Trabalhos Relacionados

Em (XAVIER et al., 2013) é mencionado a dificuldade de problemas de alocação e apresentada uma proposta de uma ferramenta auxiliar para a alocação de horários de um centro universitário no Espírito Santo. Xavier et al. (2013) desenvolveram um algoritmo baseado no modelo meta-heurístico de Pesquisa de Vizinhança Variável (*Variable Neighborhood Search*, VNS) e no método de busca local de Descida em Vizinhança Variável (*Variable Neighborhood Descent*, VND) e realizaram testes com dois cursos daquela instituição de ensino. O resultado obtido mostrou-se superior em tempo gasto e qualidade quando comparado ao processo manual, reduzindo o tempo em até 40 vezes. O problema abordado em (XAVIER et al., 2013) se assemelha ao problema abordado neste trabalho, ambos tratam de um problema de alocação em uma instituição de ensino visando auxiliar o processo de uma instituição. Porém, o artigo trata uma alocação mais geral, incluindo salas, disciplinas, horários e professores, diferentemente da proposta deste trabalho, que parte do pressuposto que as disciplinas já estão alocadas a seus horários.

Em (DERIS et al., 1999) é apresentada uma abordagem que mescla duas técnicas para a solução do problema de alocação de horários para universidades: algoritmos genéticos (AG) e *constraint-based reasoning* (CBR). Na implementação do algoritmo, as instâncias do problema são modeladas de modo a ser representada por um cromossomo. Então, inicialmente são gerados indivíduos de forma aleatória, passando por processos de mutação, *crossover* e avaliação (seguindo as técnicas de AG). Deris et al. (1999) utilizam a técnica CBR para fazer uma validação das soluções geradas pelo AG e modificá-las quando necessário, resultando em uma maior eficiência que uma abordagem de AG pura. Os autores ressaltam que a implementação desenvolvida reduziu bastante o espaço de busca, resultando em soluções melhores que uma implementação AG pura. Assim como o trabalho de Xavier et al. (2013), esse estudo aborda o problema de uma maneira geral, diferentemente da proposta deste documento, que se propõe a alocar apenas professores.

No estudo de (CIRINO; COSTA; SANTOS, 2013), é considerado o problema de alocação de salas através de um estudo de caso no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP). Os autores propõem um modelo de programação linear inteira que engloba todos os requisitos de qualidade exigidos pelo ICMC-USP. Os autores utilizaram o pacote IBM CPLEX 12.4 para resolver o modelo e compararam a solução encontrada com a solução manual utilizada pelo Instituto. Cirino, Costa e Santos (2013) apontam que o modelo matemático mostrou-se mais eficiente em tempo gasto e pôde ser utilizado como uma ferramenta auxiliar. Esse problema é similar, em partes, ao tratado neste documento, por se tratar de um problema de alocação, porém, o artigo aborda uma solução para a alocação de salas de aulas.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo contém os conceitos que fundamentam a compreensão do texto apresentado neste documento. As Seções 2.1 e 2.2 apresentam, respectivamente, conceitos básicos de programação linear e programação linear inteira, mencionando o que difere em problemas desses dois tipos. A Seção 2.3 apresenta uma visão geral da classe de problemas *Timetabling Problems* e dificuldades de problemas de alocação.

2.1 Programação Linear

Antes de falarmos de programação linear, introduziremos três conceitos importantes: função linear, desigualdade linear e igualdade linear. Winston (2004) afirma que uma função $f(x_1, x_2, ..., x_n)$ sobre $x_1, x_2, ..., x_n$ é uma **função linear** se e somente se para algum conjunto de constantes $c_1, c_2, ..., c_n$, $f(x_1, x_2, ..., x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n$. E para qualquer função linear $f(x_1, x_2, ..., x_n)$ e qualquer número b, as desigualdades $f(x_1, x_2, ..., x_n) \geq b$ e $f(x_1, x_2, ..., x_n) \leq b$ são **desigualdades lineares**, assim como $f(x_1, x_2, ..., x_n) = b$ é uma **igualdade linear** (CORMEN, 2009). Além disso, no decorrer desta Seção faz-se necessário a utilização de vetores para definir alguns conceitos. Por questões de notação, assume-se que todo vetor é um vetor coluna, e um vetor linha é um vetor transposto. Embora no texto seja usada a notação $a = (a_1, a_2, ..., a_n)$, a é um vetor coluna e a' é seu vetor transposto (linha). Casos especiais são explicitados.

De acordo com Cormen (2009), alguns problemas podem ser formulados de modo a maximizar ou minimizar um objetivo, levando em consideração que existem recursos limitados e restrições concorrentes que devem ser atendidas. Uma vez que é possível definir o abjetivo como uma função linear de determinadas variáveis e restrições sobre os recursos como igualdades ou desigualdades lineares sobre essas variáveis, pode-se dizer que temos um problema de **programação linear (PL)** que pode ser formulado através de um modelo de PL.

Considere o seguinte modelo de programação linear e alguns conceitos apresentados por Bertsimas e Tsitsiklis (1997):

Exemplo 1

$$
\begin{array}{lrcl}
\text{minimizar} & 2x_1 - x_2 + 4x_3 & & \\
\text{sujeito a} & x_1 + x_2 + x_4 & \leq & 2 \\
& 3x_2 - x_3 & = & 5 \\
& x_3 + x_4 & \geq & 3 \\
& x_1 & \geq & 0 \\
& x_3 & \leq & 0.
\end{array}
$$

No exemplo anterior, x_1, x_2, x_3 e x_4 são as **variáveis** do modelo, cujos valores devem ser escolhidos de modo a minimizar a função de custo linear $2x_1 - x_2 + 4x_3$, denominada **função objetivo**, e também respeitar o conjunto de igualdades e desigualdades lineares, denominadas **restrições** do problema. As desigualdades $x_1 \geq 0$ e $x_3 \leq 0$ são restrições de *não-negatividade* e *negatividade*, respectivamente. Quando uma variável não tem restrição de não-negatividade ou negatividade, ela é uma **variável livre**. Sejam $a = (a_1, a_2, a_3, a_4)$ e $c = (c_1, c_2, c_3, c_4)$ vetores de constantes, $x = (x_1, x_2, x_3, x_4)$ o vetor de **variáveis de decisão** e b um escalar, podemos escrever as restrições como $a'x \leq b$, $a'x \leq b$ ou $a'x \geq b$, e a função objetivo da forma $c'x$, onde $a'x = \sum_{i=0}^{4} a_i x_i$ e $c'x = \sum_{i=0}^{4} c_i x_i$. Tomando como exemplo a restrição $x_3 + x_4 \geq 3$, tem-se $a = (0,0,1,1)$ e $b = 3$. Já para a função objetivo, temos $c = (2,-1,4,0)$.

Um problema de PL pode ser descrito de forma generalizada. Desse modo, tem-se um vetor n-dimensional $c = (c_1, c_2, ..., c_n)$ de custos e deseja-se minimizar uma função linear de custo $c'x = \sum_{i=0}^{n} c_i x_i$ sobre todo vetor n-dimensional $x = (x_1, x_2, ..., x_n)$ que satisfaça um conjunto de restrições lineares. Sejam L, G e E conjuntos finitos de índices, de modo que para todo i pertencente a esses conjuntos, existe um vetor n-dimensional $a_i = (a_{i1}, a_{i2}, ..., a_{in})$ e um escalar b_i que darão origem a i-ésima restrição. Então, é possível escrever e modelar o problema da seguinte maneira:

$$
\begin{array}{lrcll}
\text{minimizar} & c'x & & & \\
\text{sujeito a} & a_i'x & \geq & b_i, & i \in G, \\
& a_i'x & \leq & b_i, & i \in L, \\
& a_i'x & = & b_i, & i \in E.
\end{array}
$$

O conjunto de todos os possíveis vetores x é denominado **espaço de busca**. Um vetor x que satisfaz todas as restrições de um problema é chamado de **solução viável**. Uma solução viável x^* é uma **solução ótima** se e somente se ela satisfaz a condição $c'x^* \leq c'x$ para toda solução viável x. Nesse caso, $c'x^*$ é denominado **custo ótimo**. Por outro lado, se para todo número real k existir uma solução viável x com custo menor que k, então dizemos que o custo ótimo é $-\infty$ ou **ilimitado**. Vale ressaltar que adotar o objetivo como minimização da função linear de custo para a notação generalizada não exclui problemas de maximização, pois maximizar $c'x$ é equivalente a minimizar $-c'x$.

Além da função objetivo, também é possível reescrever restrições mantendo a equivalência. Uma restrição de igualdade $a_i'x = b_i$ é equivalente a duas restrições de desigualdade $a_i'x \leq b_i$ e $a_i'x \geq b_i$. Restrições da forma $a_i'x \leq b_i$ podem ser escritas como $(-a_i')x \geq -b_i$. Desse modo, um problema de PL geral pode ser remodelado utilizando apenas restrições da forma $a_i'x \geq b_i$. Seja m o total de restrições, indexadas por $i \in \{1,2,...,m\}$, seja $b = (b_1,b_2,...,b_m)$ e A uma matriz de dimensões $m \times n$, tal que a i-ésima linha de A é a_i', as restrições de um problema PL geral podem ser escritas da forma $a_i'x \geq b_i$ e representadas matricialmente como $Ax \geq b$, conforme mostrado adiante.

$$\begin{array}{rrcl} \text{minimizar} & c'x & & \\ \text{sujeito a} & Ax & \geq & b_i. \end{array}$$

Uma vez definido um modelo de um problema de PL, existem técnicas capazes de resolver o problema em tempo polinomial (BERTSIMAS; TSITSIKLIS, 1997). Desse modo, uma vez que um problema é modelado, pode-se fazer o uso de ferramentas otimizadas que utilizam tais técnicas. Um exemplo de ferramenta bastante utilizada e mencionada na literatura é o *solver* CPLEX (IBM, 2016).

2.2 Programação Linear Inteira

Na Seção 2.1 foram apresentados problemas que podem ser formulados como um problema de programação linear. Em um problema de PL, as variáveis de decisão podem assumir quaisquer valores do conjunto dos números reais que atendam o objetivo do problema (maximização ou minimização) e satisfaçam suas restrições. Entretanto, em diversos problemas, existe a necessidade de que os valores das variáveis de decisão sejam números inteiros, por exemplo, quando essas representam quantidades de objetos. Quando, em um problema de programação linear, pelo menos uma das variáveis de decisão só pode assumir valores inteiros, tem-se um problema de **programação linear inteira (PLI)** ou, simplesmente, problema de

programação inteira. A restrição que restringe o domínio de uma variável para números inteiros é denominada **restrição de integralidade**. Quando algumas variáveis de um problema de PLI podem assumir valores contínuos, trata-se de um problema de **programação linear inteira mista (PLIM)** (BERTSIMAS; TSITSIKLIS, 1997). Em contrapartida, se todas as variáveis só podem assumir valores discretos, tem-se um problema de **programação linear inteira pura (PLIP)** (ALVES; DELGADO, 1997).

Alguns problemas podem ser bem específicos, de modo que as variáveis só assumem os valores 0 ou 1. Essa abordagem é muito utilizada quando existe a necessidade de representar valores *booleanos* para as variáveis. Quando as variáveis de um PLI são apenas variáveis binárias, trata-se um problema de **programação linear inteira binária (PLIB)** (BERTSIMAS; TSITSIKLIS, 1997).

À primeira vista, pode parecer que é possível resolver um problema de PLIP de forma eficiente, visto que existe algoritmo polinomial para resolver problemas de PL, e como são considerados apenas valores inteiros, há menos soluções a serem consideradas. Entretanto, isto é uma falsa impressão. Além disso, mesmo quando um problema de PLI tem um espaço de busca finito, não significa que o número de possíveis soluções a serem consideradas seja pequeno. Na verdade, restringir para valores discretos os valores das variáveis de decisão de um programa de PL dá outras características ao problema e pode impossibilitar o uso de técnicas utilizadas na resolução de problemas de programação linear, sendo necessárias a utilização de técnicas específicas (ALVES; DELGADO, 1997).

Encontrar uma solução para um PLI é NP-Difícil, ou seja, não se conhece um algoritmo eficiente com garantias de resolver um problema de programação linear inteira em tempo viável (BERTSIMAS; TSITSIKLIS, 1997), tornando-se inviável analisar todas as possíveis soluções. Desse modo, faz-se necessário a utilização de técnicas que consigam reduzir ao máximo a quantidade de possíveis soluções (ALVES; DELGADO, 1997).

2.3 *Timetabling Problem*

Definir horários com base em recursos e restrições é uma tarefa realizada em vários cenários do cotidiano. Em algumas instituições, por exemplo, existe a necessidade de alternar os funcionários que trabalham aos domingos, distribuir plantões, definir o local em que cada funcionário deve estar em determinado período de tempo, dentre outras necessidades. Ao realizar esse tipo de atividade, as instituições devem seguir uma série de restrições predefinidas com base em regimentos externos, como leis, e internos, como anseios da própria instituição. Além disso, as instituições podem requerer uma atribuição de horários que satisfaça algum(ns) objetivo(s), como reduzir custos. O problema exemplificado já é estudado há décadas e faz parte de uma classe de problemas desafiadores, conhecida como *Timetabling Problems*. Segundo Wren (1995), ***Timetabling Problem*** consiste em alocar, dado os recursos disponíveis, objetos em determinado tempo-espaço, sujeito a uma série de restrições e de modo a satisfazer, tanto quanto possível, um conjunto de objetivos desejados.

Devido a ampla definição, *Timetabling Problems* abrangem uma grande variedade de problemas e contam com algumas subdivisões. Entretanto, neste documento, considera-se apenas *Educational Timetabling*, que trata de problemas de alocação de horários em universidades e escolas de ensino médio, e corresponde a área de interesse deste trabalho. *Educational Timetabling* está presente em toda instituição de ensino ao redor do mundo e é um dos *Timetabling Problems* mais estudados do ponto de vista prático. Os problemas são muito difíceis e importantes para as instituições, e várias técnicas têm sido usadas para criar soluções viáveis de qualidade (KRISTIANSEN; STIDSEN, 2013).

Um grande número de variantes do problema tem sido proposto na literatura. Schaerf (1999), tomando como base o tipo de instituição de ensino e tipo de requisitos, classifica *Educational Timetabling* em três principais categorias:

- ***School Timetabling:*** agendamento semanal para as aulas de uma escola, evitando conflitos de horários;
- ***Course Timetabling:*** agendamento semanal para aulas de disciplinas de uma universidade, minimizando a sobreposição de aulas de disciplinas que têm estudantes em comum;
- ***Examination timetabling:*** agendamento para os exames de um conjunto de disciplinas universitárias, evitando o choque de exames de disciplinas com alunos em comum.

Schaerf (1999) relata que a categorização dos problemas não é restrita, no sentido que problemas podem ser especificados de modo que estejam em mais de uma categoria. Desse modo, Kristiansen e Stidsen (2013) menciona que a categorização dos problemas pode variar na literatura.

Carter e Laporte (1997) divide um *Timetabling Problem* em cinco subproblemas diferentes (em que nem todos são relevantes em algumas situações específicas):

- ***Course Timetabling:*** atribuir disciplinas em espaços de tempo, levando em consideração os recursos disponíveis, tais como número de salas e professores disponíveis;

- ***Class-Teacher Timetabling:*** processo de alocação mais comum em instituições de ensino médio, consiste em determinar, sem que haja conflitos, horários de aulas para disciplinas e seus professores, partindo do pressuposto que a alocação de professores em disciplinas e turmas já tenham sido realizadas;
- ***Student Scheduling:*** uma vez que a mesma disciplina pode ter diferentes sessões de horários disponíveis, consiste em atribuir alunos a sessões de suas disciplinas, sem sobreposição de horários e levando em consideração determinadas restrições como, por exemplo, capacidades das salas;
- ***Teacher Assignment:*** atribuir professores em disciplinas enquanto maximiza a função de preferência e satisfazendo todas restrições estabelecidas;
- ***Classroom Assignment:*** atribuição de salas por períodos de tempo considerando algumas restrições, tais como localização, tamanho e preferências.

Devido a grande quantidade de possibilidades e particularidades impostas, problemas de agendamentos de horários são geralmente problemas NP-Difíceis (CARTER; LAPORTE, 1997).

3 PROPOSTA DE MODELO PARA O PAPD PARA A UFC-QUIXADÁ

Neste Capítulo é apresentado o modelo de programação linear inteira para o Problema de Alocação de Professores em Disciplinas com base no objetivo e restrições apresentadas na Seção 1.1. Na Seção 3.1 são detalhados os dados de entrada para o problema. A Seções 3.2 e 3.3 apresentam, respectivamente, as variáveis utilizadas no modelo e sua formulação. Detalhes de implementação são apresentados na Seção 3.4, e uma solução do modelo é mostrada na Seção 3.5.

3.1 Entrada

O modelo possui os seguintes parâmetros de entrada:

- D é o conjunto de disciplinas ofertadas.
- P é o conjunto de professores.
- $S = [00, 29]$ 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 casais, onde cada casal $c \in C$ consiste em uma tupla $(c_1, c_2) \in P \times P$.
- $disciplinas : S \mapsto \mathscr{P}(D)$, 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.
- $d_{prof} : P \mapsto \mathscr{P}(D)$, 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

Essa Seção é destinada a apresentação da formulação matemática para o PAPD para a UFC-Quixadá. De início é mostrado a função objetiva e, posteriormente, são detalhadas todas as restrições.

A função objetivo visa encontrar a solução que maximize a preferência global dos professores por disciplinas.

Função Objetivo:

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

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

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

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

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

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

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.

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

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 seminários.

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

A restrição (9) assegura que um professor não será alocado para nenhuma disciplina em *slots* destinados a 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 os cônjuges estão alocados no mesmo grupo de horários.

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

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

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}.

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

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

A seguir apresentamos o modelo completo proposto neste estudo.

Função Objetivo:

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

Restrições:

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

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

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

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

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

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

$$\forall l \in \{0, 1, 2, 3, 4\}.$$

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

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

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

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

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

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

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

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

3.4 Implementação

O modelo foi implementado utilizando o ambiente de desenvolvimento do QT (QT, 2016) e a biblioteca CPLEX 12.6.1 (IBM, 2016), em sua configuração padrão, para a linguagem C++ (CPLUSPLUS, 2016). 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 (PYTHON, 2016).

Por conveniência, assume-se que para uma instância, 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.

3.5 Exemplo

Um exemplo de instância pode ser observada nos quadros 1, 2, 3 e 4. O Quadro 1 representa os professores e suas cargas horárias mínimas e máximas; o Quadro 2 representa as disciplinas e seus *slots* (conforme definido na Seção 1.1); o Quadro 3 representa a preferência dos professores por disciplinas; e o Quadro 4 representa os *slots* de seminários, as reuniões e os casais. Como é possível observar, o exemplo não contém seminários e reuniões. Para essa instância, o tempo de resposta foi de 0.00 segundos e o valor da função objetivo foi 28. A alocação proposta pelo modelo para essa instância é mostrada na Figura 2.

Quadro 1 – Carga horária dos professores para instância de exemplo.

Professor	Carga horária mínima	Carga horária máxima
Prof_1	0	10
Prof_2	0	10
Prof_3	0	10
Prof_4	0	10
Prof_5	0	10
Prof_6	2	10
Prof_7	2	10
Prof_8	0	10

Fonte: Elaborado pelo autor.

Quadro 2 – Distribuição de disciplinas para instância de exemplo.

Disciplina	Quantidade de professores	Slot 1	Slot 2	Slot 3
Disc_1	1	0	1	2
Disc_2	1	0	1	-
Disc_3	1	7	3	-
Disc_4	1	3	4	-
Disc_5	1	4	-	-
Disc_6	1	5	6	-
Disc_7	1	8	9	-
Disc_8	1	6	8	-

Fonte: Elaborado pelo autor.

Quadro 3 – Preferências dos professores por disciplinas para instância de exemplo.

-	Prof_1	Prof_2	Prof_3	Prof_4	Prof_5	Prof_6	Prof_7	Prof_8
Disc_1	6	6	-	8	-	-	-	-
Disc_2	2	3	-	-	-	-	1	-
Disc_3	-	-	5	-	-	-	-	2
Disc_4	-	-	-	5	1	-	-	-
Disc_5	-	-	-	-	3	-	-	-
Disc_6	-	-	-	-	-	5	-	-
Disc_7	-	-	-	-	-	-	2	1
Disc_8	-	2	-	-	-	-	-	-

Fonte: Elaborado pelo autor.

Quadro 4 – Informações complementares para instância de exemplo.

Seminários	Reuniões	Casais
-	-	(Prof_6, Prof_7)

Fonte: Elaborado pelo autor.

Figura 2 – Solução proposta pelo modelo para instância de exemplo.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB	Disc_1 - Prof_1 Disc_2 - Prof_7	Disc_1 - Prof_1 Disc_2 - Prof_7	Disc_1 - Prof_1	Disc_3 - Prof_3 Disc_4 - Prof_4	Disc_4 - Prof_4 Disc_5 - Prof_5
	CD	Disc_6 - Prof_6	Disc_8 - Prof_2 Disc_6 - Prof_6	Disc_3 - Prof_3	Disc_8 - Prof_2 Disc_7 - Prof_8	Disc_7 - Prof_8
TARDE	AB					
	CD					
NOITE	AB					
	CD					

Fonte: Elaborada pelo autor.

4 RESULTADOS EXPERIMENTAIS

Este Capítulo é destinado à apresentação de experimentos e resultados obtidos. Todos os experimentos foram realizados em uma única máquina que dispõe das seguintes configurações: processador Intel Core i5-3337U (4x1.80GHz), 5.7 GB de memória RAM e sistema operacional Ubuntu 14.04 LTS.

Na Seção 4.1 é apresentado um experimento utilizando dados reais. Experimentos e resultados com instâncias aleatórias são detalhados na Seção 4.2, na qual os resultados foram avaliados pelo tempo de execução. É importante ressaltar que utilizamos 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

Para a realização desse experimento, foi solicitado à coordenação do Campus, através dos professores orientadores, dados da UFC-Quixadá referente ao agendamento de aulas do semestre 2016.2, e um documento contendo a frequência em que cada professor ministrou cada disciplina. Após a obtenção dos dados, foi feito um pré-processamento, demandando em grande parte de trabalho manual, e foi determinada uma padronização dos dados. Após isso, os dados foram armazenados em um banco de dados no qual foram feitas consultas para montar a instância de teste. Nesse caso, foi feito uma estimativa da preferência dos professores por disciplinas utilizando a frequência em que os docentes lecionaram cada disciplina. Por exemplo, se um professor p lecionou a disciplina d cinco vezes, então $f_p(p,d) = 5$.

No total, são 108 disciplinas, e 55 professores e o tempo de resposta foi 0.07 segundos. Para analisar a qualidade da resposta, comparamos valor da função objetivo da solução proposta com o valor da função objetivo da alocação utilizada na instituição no semestre supracitado. Após o experimento, pudemos concluir que a alocação sugerida pelo modelo apresentou-se superior na ordem de 20% da alocação utilizada pela instituição, conforme mostrado na Tabela 1.

Tabela 1 – Comparação entre o valor da função objetivo da solução proposta pelo modelo e a alocação utilizada pela UFC-Quixadá no semestre 2016.2.

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

Fonte: Elaborada pelo autor.

4.2 Experimentos com Dados Aleatórios

A análise de viabilidade do tempo de resposta do modelo foi feita através de experimentos com instâncias aleatórias de diferentes tamanhos e com tempo limite de execução (*TIMEOUT*) de duas horas (7200 segundos). Os dados aleatórios foram gerados usando distribuição uniforme. Para o tamanho das instâncias aleatórias foi definido uma média de 2.8 disciplinas por professor, e as demais proporções foram baseadas na instância de dados reais. Foram utilizadas três tipos de disciplinas: com um *slot*, com dois *slots* e com três *slots*. O tipo de cada disciplina foi escolhido de forma aleatória, com 5% de chances de ter apenas um *slot*, 87% de ter dois *slots* e 8% de ter três *slots*. A escolha dos *slots* para cada disciplina foi precedida da escolha aleatória de um turno. Para cada professor foi sorteado um número dentre $0, 4$ e 8 para representar sua carga horária mínima e um número dentre $4, 8, 12$ e 14 para representar sua carga horária máxima, nos certificamos que a carga mínima era sempre menor que a máxima. Para escolha das preferências de cada professor, foram sorteadas entre 7% e 22% de disciplinas, e atribuídas preferências entre 1 e 10. Conforme definido na Seção 3.4, essas disciplinas também compõe o conjunto de disciplinas que ele está apto a lecionar. Foram sorteados dois ou três *slots* para representar os seminários e foram criadas dez reuniões, de modo que cada reunião ocorre em um *slot* e requer entre 3% e 20% dos professores. E, finalmente, assumimos que 10% da quantidade de professores são casais.

Para parte dos experimentos, foram criadas trinta e seis instâncias aleatórias seguindo o critério descrito anteriormente. Também foram criadas quarenta e seis instâncias seguindo o mesmo critério, porém, para esse conjunto de testes, cada professor tinha preferências atribuídas entre 10% e 30% das disciplinas. Na realização dos testes, foram criadas até três instâncias para cada quantidade de professores e disciplinas ou a quantidade necessária para atingir o *TIMEOUT* estabelecido. O tamanho de cada instância, o tempo gasto de cada experimento (medido em segundos) e os experimentos que resultaram em solução ótima de cada conjunto de testes podem ser observados nas Tabelas 2 e 3.

Tabela 2 – Experimentos atribuindo para cada professor entre 7% e 22% das disciplinas como preferenciais.

Professores	Disciplinas	Tempo exp. 1	Tempo exp. 2	Tempo exp. 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

Fonte: Elaborada pelo autor.

Tabela 3 – Experimentos atribuindo para cada professor entre 10% e 30% das disciplinas como preferenciais.

Professores	Disciplinas	Tempo exp. 1	Tempo exp. 2	Tempo exp. 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
500	1400	***TIMEOUT***	-	-	-

Fonte: Elaborada pelo autor.

Para o experimento 3 das instâncias com 50 professores e 140 disciplinas (linha 1) da Tabela 2, o modelo verificou em 0.02 segundos que não existia solução viável. Na mesma tabela, no experimento 3 da última linha, o programa excedeu o *TIMEOUT* e encontrou apenas soluções viáveis. Já para a Tabela 3, com exceção do experimento 1 apresentado na última linha, onde o modelo não encontrou nenhuma solução, todos os experimentos resultaram em soluções ótimas.

Após análise dos experimentos apresentados, há indícios que a quantidade de preferências atribuídas exerce impacto tanto no tempo de resposta, quanto na existência de uma solução para a instância, visto que o número de soluções viáveis podem aumentar a medida que essa quantidade aumenta. Além disso, o modelo apresentou bons resultados até mesmo para instâncias de tamanhos superiores as quantidades reais de professores e disciplinas da instituição referida neste trabalho.

5 CONCLUSÃO E CONSIDERAÇÕES FINAIS

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

Inicialmente fizemos um levantamento das restrições adotadas pela UFC-Quixadá, seguido pela formulação e implementação do modelo, e solicitação de dados referentes ao semestre letivo 2016.2 para realização de um experimento com dados reais. Então, foi feita uma comparação entre uma solução sugerida pelo modelo desenvolvido e a alocação empregada no semestre 2016.2 e uma análise da eficiência do tempo de resposta do modelo para instâncias de diferentes tamanhos. Levando em consideração o experimento com os dados obtidos da UFC-Quixadá, apresentado no Capítulo 4, Seção 4.1, a utilização deste modelo poderia contribuir significativamente com a qualidade da alocação de professores na referida instituição. Considerando os tamanhos das instâncias dos experimentos com dados aleatórios, apresentados no Capítulo 4, Seção 4.2, é possível notar que a utilização do modelo para instâncias reais da UFC-Quixadá pode apresentar um bom desempenho, visto que foi possível resolver o problema para instâncias de tamanhos superiores de forma eficiente. Desse modo, conseguimos cumprir com o objetivo traçado durante a realização deste trabalho.

Como trabalhos futuros, listamos algumas possibilidades:

- A solicitação, perante a coordenação do Campus, dos dados referente as preferências reais dos professores por disciplinas;
- Otimizar o modelo criando apenas as variáveis x e y necessárias;
- Formular uma função objetivo que leve em conta também o equilíbrio de carga horária entre os professores;
- Incluir restrições indicando possíveis horários indisponíveis de cada professor;
- Utilizar também preferências de professores por *slots*;
- O desenvolvimento de uma ferramenta que facilite o uso do modelo.

REFERÊNCIAS

ALVES, R.; DELGADO, C. Programação linear inteira. **Porto, Faculdade de Economia–Universidade do Porto. Apostila**, 1997.

BERTSIMAS, D.; TSITSIKLIS, J. N. **Introduction to linear optimization**. [S.l.]: Athena Scientific Belmont, MA, 1997. v. 6.

CARTER, M. W.; LAPORTE, G. Recent developments in practical course timetabling. In: SPRINGER. **International Conference on the Practice and Theory of Automated Timetabling**. [S.l.], 1997. p. 3–19.

CIRINO, R. B. Z.; COSTA, A. M.; SANTOS, M. O. Um modelo matemático para a resolução do problema de alocação de salas no instituto de ciências matemáticas e de computação da universidade de são paulo. 2013.

CORMEN, T. H. **Introduction to algorithms**. [S.l.]: MIT press, 2009.

CPLUSPLUS. 2016. Disponível em: <http://www.cplusplus.com/>. Acesso em: 26 Maio 2016.

DERIS, S.; OMATU, S.; OHTA, H.; SAAD, P. Incorporating constraint propagation in genetic algorithm for university timetable planning. **Engineering applications of artificial intelligence**, Elsevier, v. 12, n. 3, p. 241–253, 1999.

DODÓ, A. A. Aplicação da teoria dos jogos na resolução do problema de alocação de professores em disciplinas. 2011.

IBM. **IBM ILOG CPLEX Optimization Studio**. 2016. Disponível em: <http://www-03.ibm.com/software/products/pt/ibmilogcpleoptistud>. Acesso em: 19 Maio 2016.

KRISTIANSEN, S.; STIDSEN, T. R. **A comprehensive study of educational timetabling-a survey**. [S.l.], 2013.

PYTHON. 2016. Disponível em: <https://www.python.org/>. Acesso em: 19 Novembro 2016.

QT. 2016. Disponível em: <https://www.qt.io/>. Acesso em: 26 Maio 2016.

SCHAERF, A. A survey of automated timetabling. **Artificial intelligence review**, Springer, v. 13, n. 2, p. 87–127, 1999.

WINSTON, W. L. **Operations research: applications and algorithms**. [S.l.: s.n.], 2004. v. 3.

WREN, A. Scheduling, timetabling and rostering—a special relationship? In: SPRINGER. **International Conference on the Practice and Theory of Automated Timetabling**. [S.l.], 1995. p. 46–75.

XAVIER, B. M.; SILVA, A. D. da; VIANNA, D. S.; COSTA, H. G.; COELHO, W. B. Proposta de alocação de horários de professores e turmas em instituições de ensino superior utilizando uma heurística vns/vnd. 2013.

APÊNDICE A – Alocação proposta pelo modelo

O Apêndice A é destinado à apresentação da alocação proposta pelo modelo desenvolvido nesse estudo para a instância com dados reais da oferta de disciplinas referente ao semestre 2016.2 da UFC-Quixadá. Para uma melhor visualização, separamos as disciplinas de cada curso da instituição. As Figuras 3, 4, 5, 6, 7 e 8 apresentam, respectivamente, as atribuições de professores proposta pelo modelo para as disciplinas do curso de Ciência da Computação, Sistemas de Informação, Engenharia de Software, Engenharia da Computação, Design Digital e Redes de Computadores.

Figura 3 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Ciência da Computação.

–	–	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB	Pré-Cálculo (Prof_11) Aprend._de_Máquina (Prof_12) IHC (Prof_25) Cálculo (Prof_30) APS (Prof_39)	Aprend._de_Máquina (Prof_12) IHC (Prof_25) Cálculo (Prof_30) APS (Prof_39)	Alg._Linear (Prof_11) Arq._de_Computadores (Prof_12) SO (Prof_26) Teoria_da_Computação (Prof_43)	Visão_Computacional (Prof_13) PAA (Prof_14) Sist._Distribuídos (Prof_38) Mat._Discreta (Prof_55)	Visão_Computacional (Prof_13) PAA (Prof_14) Sist._Distribuídos (Prof_38) Mat._Discreta (Prof_55)
	CD	Alg._Linear (Prof_11) Arq._de_Computadores (Prof_12) SO (Prof_26) Teoria_da_Computação (Prof_43)	ED (Prof_22) IA (Prof_51) FBD (Prof_53)	ED (Prof_22) IA (Prof_51) FBD (Prof_53)	Eng._de_Software (Prof_18) POO (Prof_22) LIP (Prof_35)	Eng._de_Software (Prof_18) POO (Prof_22) LIP (Prof_35)
TARDE	AB	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28) FUP (Prof_50)	FUP (Prof_50)	
	CD	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28) FUP (Prof_50)		Teoria_dos_Grafos (Prof_35)	Teoria_dos_Grafos (Prof_35)

Fonte: Elaborada pelo autor.

Figura 4 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Sistemas de Informação.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB	Ger._Projeto_SW (Prof_9) Cálculo (Prof_20) APS (Prof_28)	Ger._Projeto_SW (Prof_9) Cálculo (Prof_20) APS (Prof_28)	Auditoria_Seg._SI (Prof_1) Lógica (Prof_15) TGS (Prof_23)	Mat._Discreta (Prof_41) Ger._Redes (Prof_46) Gestão_Inf._e_SI (Prof_52)	Mat._Discreta (Prof_41) Ger._Redes (Prof_46) Gestão_Inf._e_SI (Prof_52)
	CD	Auditoria_Seg._SI (Prof_1) Lógica (Prof_15) TGS (Prof_23)	Gestão_Proc._Neg. (Prof_23) Contab._e_Custos (Prof_33) POO (Prof_34) LIP (Prof_35)	Gestão_Proc._Neg. (Prof_23) Contab._e_Custos (Prof_33) POO (Prof_34) LIP (Prof_35)	Tóp._Especiais_III (Prof_37) Arq._de_Computadores (Prof_55)	Tóp._Especiais_III (Prof_37) Arq._de_Computadores (Prof_55)
TARDE	AB	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28)		Tóp._Av._BD (Prof_53)
	CD	NPI_Tarde (Prof_28)	NPI_Tarde (Prof_28)	Tóp._Av._BD (Prof_53)		

Fonte: Elaborada pelo autor.

Figura 5 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Engenharia de Software.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB	NPI_Manha (Prof_8)	NPI_Manha (Prof_8)	NPI_Manha (Prof_8) Sist._Multiagentes (Prof_39)	Des._Disp._Móveis (Prof_37)	Des._Disp._Móveis (Prof_37)
	CD	NPI_Manha (Prof_8) Sist._Multiagentes (Prof_39)	NPI_Manha (Prof_8) FUP (Prof_16)	FUP (Prof_16)	FUP (Prof_16) Manutenção_SW (Prof_19)	Manutenção_SW (Prof_19)
TARDE	AB	Qualidade_SW (Prof_10) Arq._de_Computadores (Prof_12) Lógica (Prof_43)	Qualidade_SW (Prof_10) Arq._de_Computadores (Prof_12) Lógica (Prof_43)	Prob._e_Estatística (Prof_5) Pr._Detalhado_SW (Prof_9) Arq._SW (Prof_19)	Prob._e_Estatística (Prof_5) Pr._Detalhado_SW (Prof_9) Arq._SW (Prof_19)	Redes_Sist._Dist. (Prof_7) Redes_Sociais (Prof_45) Mat._Básica (Prof_51)
	CD	IHC (Prof_25) Persistência (Prof_36) POO (Prof_39)	IHC (Prof_25) Persistência (Prof_36) POO (Prof_39)	Redes_Sist._Dist. (Prof_7) Redes_Sociais (Prof_45) Mat._Básica (Prof_51)	Ger._Projeto_SW (Prof_18) Intr._Proc._Req._SW (Prof_45) FBD (Prof_53) Mat._Discreta (Prof_55)	Ger._Projeto_SW (Prof_18) Intr._Proc._Req._SW (Prof_45) FBD (Prof_53) Mat._Discreta (Prof_55)
NOITE	AB	Inglês_I (Prof_54)	Inglês_I (Prof_54)	PPCT (Prof_52)		
	CD					

Fonte: Elaborada pelo autor.

Figura 6 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Engenharia da Computação.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB				Mat._Básica (Prof_51)	Mat._Básica (Prof_51)
	CD					
TARDE	AB	Circuitos_Digitais (Prof_20) Int._Arq._Comp. (Prof_29) Cálculo_III (Prof_30)	Circuitos_Digitais (Prof_20) Int._Arq._Comp. (Prof_29) Cálculo_III (Prof_30)	T.P.S._Embarcados_II (Prof_24) Cálculo_I (Prof_30)	T.P.S._Embarcados_II (Prof_24) Cálculo_I (Prof_30)	Mat._Discreta (Prof_14) ED (Prof_22) Arq._Comp._II (Prof_49)
	CD	Alg._Linear (Prof_6) SO_II (Prof_56)	Alg._Linear (Prof_6) SO_II (Prof_56)	Mat._Discreta (Prof_14) ED (Prof_22) Arq._Comp._II (Prof_49)	Eq._Diferenciais (Prof_49)	Eq._Diferenciais (Prof_49)
NOITE	AB	Pré-Cálculo_Noite (Prof_11)		FUP (Prof_41)	FUP (Prof_41)	
	CD		FUP (Prof_41)			

Fonte: Elaborada pelo autor.

Figura 7 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Design Digital.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB		DES._II (Prof_32)		Proj._Integrado_I (Prof_32) Proj._Integrado_II (Prof_44) Proj._Integrado_II (Prof_57)	Proj._Integrado_II (Prof_44) Proj._Integrado_II (Prof_57)
	CD		DES._II (Prof_32)		Prog._p/_Design (Prof_27) Proj._Integrado_I (Prof_32) Marketing (Prof_52)	Prog._p/_Design (Prof_27) Marketing (Prof_52)
TARDE	AB	Fotografia (Prof_17) Psic._e_Percep. (Prof_54)	Des._II (Prof_32) Ling._Marc._e_Script (Prof_57)	Direção_de_Arte (Prof_31) Hist._Design (Prof_44)	Com._Visual_II (Prof_31) Des._II (Prof_32)	Com._Visual_I (Prof_31) Com._Visual_I (Prof_44) Ling._Marc._e_Script (Prof_57)
	CD	Fotografia (Prof_17) Psic._e_Percep. (Prof_54)	Com._Visual_II (Prof_31) P._Design (Prof_57)	Direção_de_Arte (Prof_31) Hist._Design (Prof_44)	Avaliação_da_IHC (Prof_4) P._Design (Prof_57)	Avaliação_da_IHC (Prof_4) Com._Visual_I (Prof_31) Com._Visual_I (Prof_44)

Fonte: Elaborada pelo autor.

Figura 8 – Solução proposta pelo modelo para instância com dados reais: disciplinas do curso de Redes de Computadores.

--	--	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB					
	CD					
TARDE	AB					
	CD				PPCT (Prof_52)	
NOITE	AB	Redes (Prof_2) R._Alta_Velocidade (Prof_16) Ger._Projetos (Prof_36)	Gestão_TI_e_Com (Prof_1) Redes (Prof_2) R._Alta_Velocidade (Prof_16)	P._Integrado (Prof_26) POO (Prof_34) S._Distribuídos (Prof_38)	P._Integrado (Prof_26) POO (Prof_34) S._Distribuídos (Prof_38)	A._Desemp._Redes (Prof_7) Lab._Infra_Estr. (Prof_42) SO (Prof_49) T2_FUP (Prof_50)
	CD	Gestão_TI_e_Com (Prof_1) P._Scripts (Prof_29) Empreended. (Prof_33)	P._Scripts (Prof_29) Empreended. (Prof_33) Ger._Projetos (Prof_36)	A._Desemp._Redes (Prof_7) Ética (Prof_33) SO (Prof_49)	Prob._e_Estatística (Prof_5) RC_Móveis (Prof_7) Des._Web (Prof_27) T2_FUP (Prof_50)	Prob._e_Estatística (Prof_5) RC_Móveis (Prof_7) Des._Web (Prof_27) T2_FUP (Prof_50)

Fonte: Elaborada pelo autor.