UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JHONATA ADAM SILVA MATIAS

UM MODELO DE PROGRAMAÇÃO INTEIRA PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES DA UFC-QUIXADÁ

QUIXADÁ – CEARÁ

2016

JHONATA ADAM SILVA MATIAS

UM MODELO DE PROGRAMAÇÃO INTEIRA PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES DA UFC-QUIXADÁ

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. Me. 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)

M38m Matias, Jhonata Adam Silva.
Um Modelo de Programação Inteira Para o Problema de Alocação de Professores da UFC-Quixadá / Jhonata Adam Silva Matias. – 2016.
34 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. Alocação de Professores. 2. Alocação de Disciplinas. 3. Programação Inteira. 4. UFC-Quixadá. I. Título.
CDD 004

JHONATA ADAM SILVA MATIAS

UM MODELO DE PROGRAMAÇÃO INTEIRA PARA O PROBLEMA DE ALOCAÇÃO DE PROFESSORES DA UFC-QUIXADÁ

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. Me. 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. Me. Arthur Rodrigues Araruna
Campus Quixadá
Universidade Federal do Ceará - UFC

Prof. Me. Paulo Henrique Macedo de Araujo
Campus Quixadá
Universidade Federal do Ceará - UFC

AGRADECIMENTOS

Aos meu familiares, especialmente à minha mãe Joana, que sempre me deu o apoio necessário para continuar e concluir minha graduação.

Aos Professores Lucas Ismaily e Críston Souza, pela paciência em me fazer as devida correções e pela generosidade em compartilhar seus conhecimentos.

À minha namorada Joice, pelo carinho, apoio emocional e por estar sempre presente.

À minha amiga Camila, pelos conselhos e por desconstruir por diversas vezes a minha rotina.

Aos professores participantes da banca examinadora Arthur Rodrigues Araruna e Paulo Henrique Macedo de Araujo, pelo tempo e pelas sugestões.

Aos meus colegas de turma, por toda a ajuda durante os anos de graduação.

“O amor é sábio, o ódio é tolo.”

(Bertrand Russell)

RESUMO

As universidades lidam frequentemente com o problema de alocação de professores e disciplinas. Neste trabalho abordamos esse problema buscando maximizar a preferencia geral dos porfessores por disciplinas e minimizar o choque de horário entre disciplinas em que mais alunos possam se matricular. O objetivo desse trabalho é desenvolver um modelo de Programação Inteira capaz de resolver o problema de alocação de professores e disciplinas aplicado ao Campus da Universidade Federal do Ceará em Quixadá. É apresentado o passo a passo da construção do modelo de Programação Inteira baseado nas restrições de alocação praticadas no Campus. A partir da implementação do modelo foram realizados experimentos para instâncias produzidas com as ofertas de disciplinas do Campus do semestre de 2016.2. Os experimentos mostraram que o modelo não é efetivo na alocação de professores e disciplinas para instâncias com 100% dos dados a serem alocados atualmente no Campus.

Palavras-chave: Alocação de Professores. Alocação de Disciplinas. Programação Inteira. UFC-Quixadá.

ABSTRACT

Universities often face the problem of professors and courses scheduling. In this work, we address this problem by maximizing the general preference of professors for courses and minimizing time overlap on courses in which more students may enroll. The goal of this work is to develop a Integer Programming Model able to solving the problem of professors and courses scheduling to be applied on Quixadá Campus of the Federal University of Ceará. The construction of the Integer Programming model is presented step by step based on the Campus scheduling constraints. Using offered courses of the Campus in the semester of 2016.2, we perform experiments from the implementation of the model. The experiments showed that the model is not effective to the problem of professors and courses scheduling for instances that contain 100% of the data to be allocated currently on Campus.

Palavras-chave: Professors Scheduling. Courses Scheduling. Integer Programming. UFC-Quixadá.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PAPD	Problema de Alocação de Professores em Disciplinas
UFC-Quixadá	Universidade Federal do Ceará em Quixadá
PL	Programação Linear
PI	Programação Inteira
PIM	Programação Inteira Mista
PIB	Programação Inteira Binária

SUMÁRIO

1 INTRODUÇÃO

Antes do início de um semestre letivo, as universidades realizam uma série de tarefas a fim de se preparar para as atividades de um novo semestre. Entre essas tarefas se encontra a alocação de professores em disciplinas, que lida com o problema de alocar professores em disciplinas e disciplinas nos horários de aula respeitando um conjunto de restrições. O problema se torna difícil de resolver quando lidamos com um grande volume de dados. Isso se dá por alguns motivos, que vão desde a quantidade de restrições envolvidas até às preferências concorrentes dos professores. Doravante, chamaremos esse problema de **PAPD (problema de alocação de professores em disciplinas)**.

No campus da Universidade Federal do Ceará em Quixadá (UFC-Quixadá), a direção do campus e os coordenadores de curso participam da definição da grade de horários acadêmicos. Nela está definida toda a alocação do campus, tanto de professores em disciplinas quanto de disciplinas na grade de horários. Atualmente as ofertas de disciplinas são definidas por cada curso. São definidos de forma manual a alocação dos professores a cada oferta, e de forma automatizada a alocação das disciplinas nos horários de aula. Essa alocação é feita respeitando um conjunto de restrições que definem como os professores e as disciplinas devem ser alocados.

Dentro do contexto de uma Universidade, conseguimos identificar três grupos de pessoas que são diretamente afetadas pelo PAPD, seja pela alocação gerada ou pelo método utilizado para fazer a alocação. São eles: os professores, os alunos e os responsáveis por gerar a alocação. Os professores têm suas preferências quanto às disciplinas que vão ministrar. Os alunos, por sua vez, são diretamente afetados por uma alocação que reduza suas opções de matrícula. Quanto aos responsáveis pela alocação, esses lidam com um volume de trabalho que cresce de forma exponencial em função da quantidade de dados envolvidos na alocação. Automatizar a resolução do PAPD, maximizando a preferência geral dos professores e as opções de matrícula dos alunos, é vantajoso para todas as pessoas afetadas pelo problema.

Uma larga variedade de problemas de alocação voltados à escolas de ensino médio e universidades são encontrados na literatura, bem como diversas abordagens são propostas para resolver o problema, dentre elas: Algoritmos Genéticos, Satisfação de Restrições, Fluxo em Redes e Programação Inteira, entre outras (SCHAERF, 1999). Neste trabalho, modelamos o PAPD como um problema de Programação Inteira.

Um problema de Programação Inteira consiste em um conjunto de variáveis que compõem uma função objetivo e um conjunto de equações ou inequações lineares, que são as

restrições do problema. Resolver um problema de Programação Inteira consiste em encontrar uma valor inteiro para as variáveis que satisfaça todas as restrições e minimize (ou maximize, dependendo da natureza do problema) o valor da função objetivo.

Este trabalho teve como objetivo geral produzir um modelo de Programação Inteira capaz de resolver o PAPD aplicado à UFC-Quixadá. Para tanto, identificamos as restrições que são utilizadas no processo de alocação. A partir das restrições levantadas, modelamos o PAPD da UFC-Quixadá como um problema de Programação Inteira e realizamos alguns testes implementando o modelo e verificando o tempo que ele leva para resolver o problema para instâncias de diferentes tamanhos.

No trabalho de Dodó (2011), é proposta uma forma de modelar o PAPD a partir dos conceitos da Teoria dos Jogos. Esse trabalho também leva em conta as restrições da alocação da UFC-Quixadá de forma que uma solução do modelo proposto no trabalho satisfaz essas restrições considerando a preferência por disciplinas dos professores. Assim como no trabalho de Dodó (2011), nosso trabalho leva em conta as restrições da alocação da UFC-Quixadá para produzir um modelo que maximiza a preferência por disciplinas dos professores. A diferença entre os dois trabalhos se encontra na abordagem utilizada para resolver o problema e na maximização das opções de matrícula que realizamos no nosso trabalho.

Lach e Lübbecke (2012) apresentam um modelo de Programação Inteira baseado no conjunto de restrições das instâncias da *2nd International Timetabling Competition* (ITC, 2007). Esse modelo resolve as instâncias em dois passos: primeiro associa disciplinas com seus horários e depois associa esses horários às salas de aula. O que é importante ressaltar da abordagem é que ela trata o problema com dois tipos de restrições: *Hard* e *Soft*. Restrições *Hard* devem ser respeitadas incondicionalmente; por outro lado, restrições *Soft* devem ser satisfeitas tanto quanto possível. Lach e Lübbecke (2012), assim como no nosso trabalho, trabalham com Programação Inteira para resolver o problema de alocação em universidades. Os dois trabalhos se diferem na forma como resolvem problema, pois o nosso trabalho não divide em passos o processo de resolução. Além disso, os dois trabalhos lidam com um conjunto de restrições diferentes.

Esse trabalho está dividido em cinco capítulos. No Capítulo 2 é apresentada a fundamentação teórica. O Capítulo 3 apresenta o modelo de Programação Inteira desenvolvido para o PAPD da UFC-Quixadá. O Capítulo 4 apresenta a avaliação feita sobre o modelo desenvolvido. E por fim, no Capítulo 5, são feitas as considerações finais e as indicações de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos fundamentais para a compreensão deste trabalho. A Seção 4.1 apresenta os conceitos de Programação Linear. A Seção 4.2 descreve Programação Inteira, evidenciando o que difere um modelo de Programação Inteira de um modelo de Programação Linear. A Seção 4.3 apresenta uma visão geral dos problemas de *Educational Timetabling*. Para detalhes além dos apresentados a seguir, indicamos: Schrijver (1998), Bertsimas e Tsitsiklis (1997) e Hillier e Lieberman (2010).

2.1 Programação Linear

Um modelo de **Programação Linear (PL)** é constituído por um conjunto de variáveis. Essas variáveis compõem uma função linear objetivo, além de um conjunto de equações e inequações lineares que representam as restrições do modelo. Solucionar um problema de PL é encontrar um valor para cada uma das variáveis que respeite todas as restrições e minimize o custo da função objetivo (BERTSIMAS; TSITSIKLIS, 1997).

O problema e os conceitos apresentados adiante foram retirados de Bertsimas e Tsitsiklis (1997).

Considere o seguinte problema de PL:

Exemplo 1.

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

No exemplo acima, x_1, x_2, x_3 e x_4 são as variáveis do modelo. São compostas por elas a **função objetivo** $2x_1 - x_2 + 4x_3$, e as **restrições** $x_1 + x_2 + x_4 \leq 2$, $3x_2 - x_3 = 5$, $x_3 + x_4 \geq 3$, $x_1 \geq 0$ e $x_3 \leq 0$. As duas últimas restrições são **restrições de integridade** das variáveis x_1 e x_3.

Essas restrições têm como intuito estabelecer os limites (superiores ou inferiores) das variáveis. As demais variáveis que não possuem restrições de integridade, são chamadas **variáveis livres** e podem assumir qualquer valor pertencente ao conjunto dos números reais.

As restrições representadas por inequações ou equações podem assumir os formatos $ax^T \leq b$, $ax^T \geq b$ ou $ax^T = b$, onde $a = (a_1, a_2, ..., a_n)$ e $x = (x_1, x_2, ..., x_n)$ são **vetores linha**, e b é um **escalar** qualquer (mais à frente veremos uma forma mais geral de representar todas as restrições). Se considerarmos a segunda restrição do exemplo acima, temos $a = (0, 3, -1, 0)$ e $b = 5$. A função objetivo tem a forma cx^T, onde $c = (c_1, c_2, ..., c_n)$ e $x = (x_1, x_2, ..., x_n)$. As expressões cx^T da função objetivo e ax^T das restrições podem ser escritas como $\sum_{i=1}^{n} c_i x_i$ e $\sum_{i=1}^{n} a_i x_i$, respectivamente.

As variáveis envolvidas no modelo são chamadas **variáveis de decisão**. Uma valoração para as variáveis de decisão que satisfaça todas as restrições é chamada **solução viável**. O conjunto de todas as soluções viáveis é chamado **conjunto viável** ou **conjunto de possibilidades**. Dado um conjunto de soluções viáveis de um problema de PL qualquer, uma solução viável pertencente a esse conjunto que tenha menor valor segundo a função objetivo é chamada **solução viável ótima** ou simplesmente **solução ótima**. Quando a natureza do problema é de maximização, pode-se tratar a função de maximização utilizando seu valor oposto e transformando-a em uma função de minimização equivalente. Note que maximizar cx^T equivale a minimizar $-cx^T$.

Um problema de PL pode ter sua representação generalizada da seguinte forma: considere as restrições representadas por equações ($ax^T = b$). Podemos reescrevê-las como um par de inequações $ax^T \leq b$ e $ax^T \geq b$. Restrições com o formato $ax^T \leq b$, podem ser reescritas como $(-a)x^T \geq -b$ e restrições de integridade podem ser escritas como inequações, onde a tem apenas um valor não nulo. Podemos também representar todos coeficientes de todas as restrições como uma única matriz A de dimensões $m \times n$, onde cada linha é um vetor de coeficientes de uma única restrição. As variáveis de decisão seriam um vetor $x = (x_1, x_2, ..., x_n)$ e b representaria o vetor $(b_1, b_2, ..., b_m)$.

Portanto, podemos generalizar um problema de PL escrevendo-o da seguinte forma:

minimizar

$$cx^T$$

sujeito a

$$Ax^T \geq b^T$$

Reescrevendo as restrições do Exemplo 1, temos:

$$\begin{pmatrix} -1 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geq \begin{pmatrix} -2 \\ 5 \\ -5 \\ 3 \\ 0 \\ 0 \end{pmatrix}$$

E para a função objetivo:

$$\begin{pmatrix} 2 & -1 & 4 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Como ilustração, considere o seguinte exemplo de Hillier e Lieberman (2010):

Exemplo 2. Uma determinada empresa fabrica produtos de vidro, entre os quais janelas e portas de vidro. A empresa possui três fábricas industriais. As esquadrias de alumínio e ferragens são feitas na Fábrica 1, as esquadrias de madeira são produzidas na Fábrica 2 e, finalmente, a Fábrica 3 produz o vidro e monta os produtos.

A empresa decidiu que produtos não rentáveis estão sendo descontinuados, liberando capacidade produtiva para o lançamento de dois novos produtos com grande potencial de vendas:

Produto 1: Uma porta de vidro de 2,5 m com esquadria de alumínio;

Produto 2: Uma janela duplamente adornada com esquadrias de madeira de $1,20 \times 1,80m$.

O Produto 1 requer parte da capacidade produtiva das Fábricas 1 e 3, mas nenhuma da Fábrica 2. O Produto 2 precisa apenas das Fábricas 2 e 3. A empresa poderia vender tanto quanto fosse possível produzir desses produtos por essas fábricas. Entretanto, pelo fato de ambos

os produtos poderem estar competindo pela mesma capacidade produtiva na Fábrica 3, não está claro qual mix dos dois produtos seria o mais lucrativo. O problema então é definido da seguinte forma:

Determinar quais devem ser as taxas de produção para ambos os produtos de modo a maximizar o lucro total, sujeito às restrições impostas pelas capacidades produtivas limitadas disponíveis nas três fábricas. (Cada produto será fabricado em lotes de 20, de forma que a taxa de produção é definida como o número de lotes produzidos por semana.) É permitida qualquer combinação de taxas de produção que satisfaça essas restrições, inclusive não produzir nada de um produto e o máximo possível do outro.

Obtendo estimativas razoáveis, os dados coletados foram os seguintes:

- A Fábrica 1 leva uma hora para produzir um lote do produto 1 e tem quatro horas de produção disponíveis por semana.
- A Fábrica 2 leva duas horas para produzir um lote do produto 2 e tem doze horas de produção disponíveis por semana.
- A Fábrica 3 leva três horas para produzir um lote do produto 1, duas horas para produzir um lote do produto dois e tem dezoito horas de produção disponíveis por semana.
- O lucro por lote obtido pelo produto 1 é de U$ 3.000 e o lucro por lote do produto 2 é U$ 5.000.

Para formular o modelo matemático (PL) para esse problema, façamos:

$x_1 =$ número de lotes do produto 1 produzido semanalmente

$x_2 =$ número de lotes do produto 2 produzido semanalmente

$Z =$ lucro total por semana (em milhares de dólares) obtido pela produção desses dois produtos

Portanto, x_1 e x_2 são as variáveis de decisão para o modelo. Usando-se as informações de lucro obtemos:

$$Z = 3x_1 + 5x_2$$

O objetivo é escolher os valores de x_1 e x_2 de forma a maximizar $Z = 3x_1 + 5x_2$, sujeito às restrições impostas em seus valores por limitações de capacidade de produção disponível nas três fábricas. As informações obtidas indicam que cada lote de produto 1 fabricado por semana usa uma hora de tempo de produção por semana na Fábrica 1, ao passo que estão disponíveis somente quatro horas semanais. Essa restrição é expressa

matematicamente pela inequação $x_1 \leq 4$. Similarmente, a Fábrica 2 impõe a restrição $2x_2 \leq 12$. O número de horas de produção usado semanalmente na Fábrica 3 escolhendo-se x_1 e x_2 como as taxas de produção dos novos produtos seria $3x_1 + 2x_2$. Portanto, a declaração matemática da restrição da Fábrica 3 é $3x_1 + 2x_2 \leq 18$. Finalmente, já que as taxas de produção não podem ser negativas, é necessário restringir as variáveis de decisão para serem não-negativas: $x_1 \geq 0$ e $x_2 \geq 0$.

Em suma, na linguagem matemática da PL, o problema é escolher os valores de x_1 e x_2 de forma a

$$\begin{aligned} &\text{maximizar} \\ &\qquad Z = 3x_1 + 5x_2 \\ &\text{sujeito a} \\ &\qquad x_1 \leq 4 \\ &\qquad 2x_2 \leq 12 \\ &\qquad 3x_1 + 2x_2 \leq 18 \\ &\qquad x_1 \geq 0 \\ &\qquad x_2 \geq 0 \end{aligned}$$

Dado um modelo de PL de um determinado problema, utiliza-se *solvers* como o *CPLEX* (IBM, 2016) para obter uma solução para o problema. Portanto, a principal dificuldade em trabalhar com PL é produzir um modelo para o problema, visto que uma solução para o mesmo pode ser encontrada em tempo polinomial (LUENBERGER; YE, 1984).

2.2 Programação Inteira

Na seção anterior foram apresentadas as variáveis de decisão e o seu papel na modelagem dos problemas de PL. Elas podem assumir qualquer valor pertencente ao conjunto dos números reais desde de que esse valor respeite as restrições do modelo. Porém, em muitos problemas do mundo real, as variáveis de decisão só fazem sentido se assumirem valores inteiros. Quando modelamos um problema de PL onde todas as variáveis de decisão têm a restrição de só assumirem valores inteiros, chamamos o modelo gerado de um modelo de **Programação Linear Inteira** ou simplesmente **Programação Inteira (PI)**. Também são encontrados problemas em

que só parte das variáveis de decisão necessitam ser inteiras. Esses são chamados problemas de **Programação Inteira Mista (PIM)**.

Há casos mais específicos de PI onde as variáveis representam decisões binárias. Por exemplo: $x_i = 1$, se a decisão i for *sim* e $x_i = 0$, se a decisão i for *não*. Modelos de PI que tenham variáveis de decisão binárias são também chamados modelos de **Programação Inteira Binária (PIB)**.

Considere o Exemplo 2, apresentado na seção anterior. Se for adicionada a restrição de que só serão contabilizados lotes inteiros, teríamos que x_1 e x_2 só poderiam assumir valores inteiros não negativos. O problema inicialmente proposto como um problema de PL se tornaria um problema de PI.

Problemas de PI puros (que só possuem variáveis inteiras) com uma região de soluções viáveis limitada têm a garantia de possuírem apenas um número finito de soluções viáveis. Isso pode dar a falsa impressão de que problemas de PI são fáceis de resolver, porém isso não é verdade (HILLIER; LIEBERMAN, 2010). Schrijver (1998) demonstra que resolver um problema de PI em sua versão de decisão é um problema NP-completo.

2.3 *Educational Timetabling*

Educational Timetabling trata de problemas de alocação envolvendo universidades e escolas de ensino médio. Kingston (2013) divide esses problemas em três classes de problemas, são eles: *school timetabling*, *course timetabling* e *examination timetabling*. *School timetabling* se refere a problemas de alocação em escolas de ensino médio. Os outros dois, *course timetabling* e *examination timetabling*, tratam de problemas de alocação de horários de disciplinas e provas em universidades. Problemas de *Educational Timetabling* são tipicamente problemas NP-Difíceis (LEWIS, 2008). Adiante será dada um explicação mais precisa sobre cada uma dessas classes de problemas.

2.3.1 School Timetabling

No problema de ***school timetabling***, a alocação é dada para ciclos semanais ou quinzenais. Os horários são particionados em períodos de mesmo tamanho. O problema também tem a característica de que os estudantes estão agrupados em classes (KINGSTON, 2004). A forma tradicional do problema consiste na atribuição de aulas a períodos de tal forma que nenhum

professor ou classe se envolva em mais de uma aula ao mesmo tempo (ŠLECHTA, 2004).

2.3.2 Course Timetabling

Course timetabling é um problema de alocação voltado para as universidades. Nesse problema, um conjunto de disciplinas devem ser associadas a uma grade de horários e a um conjunto salas de aula. Além disso, determinadas restrições devem ser respeitadas: um professor não pode estar associado a mais de uma disciplina em um mesmo período; uma sala de aula não pode receber dois cursos ao mesmo tempo; disciplinas de um mesmo curso não podem compartilhar um período, etc. Se for impossível satisfazer todas as restrições, o número de restrições violadas deve ser minimizado (LACH; LÜBBECKE, 2012).

Para Schaerf (1999), ***course timetabling*** envolve a alocação semanal de aulas de uma universidade, de forma que minimize os choques de horários entre cursos que tenham alunos em comum.

Diferente de *school timetabling*, *course timetabling* não agrupa os alunos em classes. O problema nessa forma leva em conta as escolhas individuais dos alunos quanto as disciplinas a serem cursadas (KINGSTON, 2004).

Este trabalho trata de um problema similar aos de *course timetabling*, onde são levadas em conta problemas encontrados no processo de alocação da UFC-Quixadá. Por esse motivo, o problema abordado aqui tem algumas diferenças com relação aos de *course timetabling*. Um exemplo é a alocação em salas de aulas, que não é levada em conta neste trabalho. São consideradas aqui apenas as alocações de professores em disciplinas, e de disciplinas em horários.

2.3.3 Examination Timetabling

Examination timetabling é definido como sendo a atribuição de um conjunto de exames (ou provas) a um número limitado de horários sujeito a um conjunto de restrições (YANG; PETROVIC, 2004). Essa a alocação deve evitar choque de horários entre exames de cursos que tenham alunos em comum (SCHAERF, 1999).

Assim como *course timetabling*, esse é um problema encontrado nas universidades. A característica principal que difere ambos, é a restrição quanto aos choques de horários entre disciplinas ou exames que tenham alunos em comum. *course timetabling* tenta minimizar esses choques de horários, enquanto que *examination timetabling* deve proibi-los.

3 MODELO PARA O PAPD DA UFC-QUIXADÁ

A construção de um modelo de PI para o problema de alocação de professores da UFC-Quixadá demanda o conhecimento de algumas informações relevantes no domínio do problema. Dessa forma, foi feito um levantamento de como é realizada atualmente a alocação na UFC-Quixadá, e chegamos às informações que são importantes para a alocação dos professores e das disciplinas.

A alocação é feita sobre uma grade de horários semanal. Cada elemento dessa grade representa um horário disponível para alocação ao qual chamamos de ***slot***. Um *slot* tem o valor de dois créditos (2 horas), ou seja, uma disciplinas de quatro créditos demanda dois *slots* para ser alocada. Um único dia agrupa seis *slots*, os quais estão divididos nos três turnos de um dia (manhã, tarde e noite). Um turno de um dia agrupa dois *slots*, o *slot* AB e o *slot* CD do turno.

Neste trabalho, mapeamos os *slots* para valores inteiros, dessa forma os *slots* da segunda-feira são representados pelos valores: 1, 2, 3, 4, 5 e 6; os da terça-feira: 7, 8, 9, 10, 11 e 12; e assim por diante. A tabela abaixo mostra de forma completa como os *slots* estão organizados na grade de horários.

Figura 1 – Grade de horários e seus respectivos *slots*.

--		SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
MANHÃ	AB	Slot 01	Slot 07	Slot 13	Slot 19	Slot 25
	CD	Slot 02	Slot 08	Slot 14	Slot 20	Slot 26
TARDE	AB	Slot 03	Slot 09	Slot 15	Slot 21	Slot 27
	CD	Slot 04	Slot 10	Slot 16	Slot 22	Slot 28
NOITE	AB	Slot 05	Slot 11	Slot 17	Slot 23	Slot 29
	CD	Slot 06	Slot 12	Slot 18	Slot 24	Slot 30

Fonte – Elaborada pelo autor

Para os professores, há uma carga horária máxima e mínima para cumprir. Essa carga horária pode ser diferente para cada professor, já que alguns professores realizam atividades como coordenação de curso, que permitem redução de carga horária. O número de disciplinas que um professor deve ser associado depende então de sua carga horário máxima e mínima e do número de créditos das disciplinas.

Há também restrições que definem os horários e os dias aos quais os professores não devem ser alocados:

- professores não devem ser alocados nos três turnos de um mesmo dia (essa restrição pode ser quebrada desde que autorizado por escrito pelo professor. Mantivemos obrigatória para todos os professores em nossa implementação);
- professores não devem ser alocados no último horário de um dia e no primeiro horário do dia seguinte;
- professores devem estar livres no horário CD da manhã ou no AB da tarde de quarta (para os encontros denominados "seminários");
- um professor deve ter a segunda-feira ou a sexta-feira live;
- o dia livre (segunda-feira ou sexta-feira) é o mesmo para casais de professores;
- professores não devem ser alocados a nenhuma disciplina nos horários das reuniões que participam.

A UFC-Quixadá conta atualmente com seis cursos de graduação. Todo semestre os cursos ofertam um conjunto de disciplinas, de forma que as disciplinas a serem alocadas em um semestre são as ofertadas pelos cursos. As disciplinas estão agrupadas nesses cursos de forma que algumas delas são compartilhadas entre cursos e outras não. Embora essas disciplinas sejam compartilhadas, cada oferta é específica de um curso.

A maioria dos cursos são de horário integral, ou seja, utilizam os turnos da manhã e da tarde para alocar suas disciplinas. Porém cada curso opta por um horário preferencial, onde a maioria das disciplinas obrigatórias são alocadas. Em alguns casos, a soma de créditos das disciplinas obrigatórias de um semestre de um cursos excede a quantidade de créditos de um turno (10 créditos). Para resolver isso os cursos optam por alocar os créditos excedentes em outro turno, já que é imprescindível que disciplinas obrigatórias de um mesmo semestre não choquem horário. Para representar isso em nosso modelo definimos um horário primário e um horário secundário para cada curso. O horário primário é onde são alocadas as disciplinas obrigatórias, e o secundário é onde são alocadas as disciplinas obrigatórias excedentes. As disciplinas optativas são livres para serem alocadas em qualquer horário.

Assim como os professores, as disciplinas têm seu conjunto de restrições:

- uma disciplina tem um número fixo de professores que devem ser associados a ela;
- disciplinas associadas a um mesmo professor não compartilham *slots*;
- disciplinas obrigatórias de um mesmo semestre de um mesmo curso não devem compartilhar *slots*;

- o número de *slots* em que uma disciplina deve ser alocada é igual à metade de seus créditos;
- uma disciplina pode ter seus horários prefixados;
- uma disciplina deve compartilhar *slots* com um conjunto prefixado de disciplinas.

O modelo de PI proposto neste trabalho foi então construído a partir das restrições do problema. O modelo representa as entidades a serem alocadas da seguinte forma: variáveis binárias representam a associação professores/disciplinas e disciplinas/horários (ou *slots*, como chamaremos neste trabalho). Por exemplo: o valor de uma variável z_{pi} é 1 se o professor p está associado à disciplina i, e 0 caso contrário. Dessa forma, estão sendo mapeados professores, disciplinas e *slots* para variáveis binárias. Essas compõem as equações e inequações lineares das restrições do modelo.

Definimos que há **choque de horários** entre duas disciplinas quando elas compartilham pelo menos um *slot*.

A seguir estão as variáveis, parâmetros e conjuntos presentes no modelo. Em seguida, serão apresentadas as restrições e a função objetivo.

Conjuntos:

- P - professores;
- D - disciplinas;
- D_p - disciplinas do professor p;
- C - cursos;
- K_c - semestres de um curso c;
- D_{kc}^{ob} - disciplinas obrigatórias de um semestre k de um curso c;
- D_c^{op} - disciplinas optativas de um curso c;
- S - *slots*;
- S_{pri_c} - *slots* do turno primário do curso c;
- S_{sec_c} - *slots* do turno secundário do curso c;
- T_i - disciplinas às quais a disciplina i deve compartilhar *slots*;
- H_i - *slots* prefixados da disciplina i;
- R - reuniões;
- P_r - professores da reunião r;
- S_r - *slots* da reunião r;
- A - casais de professores;

- G_m - grupos de *slots* de tamanho m.

Parâmetros:

- max_p - número máximo de créditos do professor p;
- min_p - número mínimo de créditos do professor p;
- a_{ij} - número de alunos que podem cursar as disciplinas i e j;
- b_{pi} - nível de preferência do professor p pela disciplina i;
- crd_i - número de créditos da disciplina i;
- n_i - número de professores na disciplina i;

Variáveis:

- x_{ij} - 1 se as disciplinas i e j compartilham *slot*, 0 caso contrário;
- y_{is} - 1 se a disciplina i está alocada no *slot* s, 0 caso contrário
- w_{ps} - 1 se o professor p está associado ao *slot* s, 0 caso contrário;
- z_{pi} - 1 se o professor p está associado à disciplina i, 0 caso contrário;
- u_{kc} - 1 se todas as disciplinas obrigatórias do semestre k do curso c estão alocadas no turno primário do próprio curso;
- v_p - 1 se o professor p não dá aula na sexta-feira, 0 se não dá aula na segunda-feira.

A função objetivo define qual alocação é melhor. Para isso ela leva em conta a preferência geral dos professores e os choques de horários das disciplinas. O somatório $\sum a_{ij} * x_{ij}$ representa, na função objetivo, a soma dos alunos que podem se matricular em disciplinas que compartilham *slots*. O somatório $\sum b_{pi} * z_{pi}$ representa a soma das preferências dos professores. É importante notar que a preferência de um professor por uma disciplinas só é contabilizada se ele estiver associado a essa disciplinas.

É utilizado na função objetivo um parâmetro $\alpha \in [0,1]$. Esse valor balanceia os pesos dos dois lados da função objetivo. Por exemplo, quando $\alpha = 0$, é considerado na função objetivo apenas a preferência dos professores, e ignorado o número de alunos que podem fazer pares de disciplinas que compartilham *slots*. Por outro lado, se $\alpha = 1$, é considerado apenas o número de alunos nos choques de horários. Assim, para $\alpha = 0.5$, é dado peso igual para ambos.

Função Objetivo:

$$\min \alpha * \sum a_{ij} * x_{ij} + (\alpha - 1) * \sum b_{pi} * z_{pi}$$

As Restrições (3.1) e (3.2) definem, respectivamente, a carga horária máxima e

mínima de cada professor.

$$\sum_{i \in D} crd_i * z_{pi} \leq max_p, \forall p \in P \tag{3.1}$$

$$\sum_{i \in D} crd_i * z_{pi} \geq min_p, \forall p \in P \tag{3.2}$$

A Restrição (3.3) impede que professores estejam alocados nos três turnos de um mesmo dia. Apresentaremos essa restrição em sua forma não linear para facilitar o entendimento do modelo, embora seja possível torná-la linear.

$$\sum_{q \in \{0,2,4\}} max(w_{p(s+q)}, w_{p(s+q+1)}) \leq 2, \forall s \in \{1,7,13,19,25\}, \forall p \in P \tag{3.3}$$

A Restrição (3.4) define que professores não devem ser alocados no último horário de um dia e no primeiro do dia seguinte.

$$w_{ps} + w_{p(s+1)} \leq 1, \forall s \in \{6,12,18,24\}, \forall p \in P \tag{3.4}$$

A Restrição (3.5) se refere aos seminários, em que os professores devem estar livres no horário CD da manhã ou no AB da tarde de quarta.

$$w_{p14} + w_{p15} \leq 1, \forall p \in P \tag{3.5}$$

As Restrições (3.6) e (3.7) definem que um professor deve ter a segunda-feira ou a sexta-feira livre.

$$\sum_{s \in \{1,2,\ldots,6\}} w_{ps} \leq 6 * v_p, \forall p \in P \tag{3.6}$$

$$\sum_{s \in \{25,26,\ldots,30\}} w_{ps} \leq 6 * (1 - v_p), \forall p \in P \tag{3.7}$$

A Restrição (3.8) se refere aos casais, e diz que o dia livre (segunda-feira ou sexta-feira) é o mesmo para casais de professores.

$$v_{p_1} = v_{p_2}, \forall \{p_1, p_2\} \in A \tag{3.8}$$

A Restrição (3.9) fixa as eventuais disciplinas predeterminadas de um professor.

$$z_{pi} = 1, \forall i \in D_p, \forall p \in P \tag{3.9}$$

A Restrição (3.10) diz que professores não devem ser alocados a nenhum disciplina nos horários das reuniões que participam.

$$w_{ps} = 0, \forall s \in S_r, \forall p \in P_r, \forall r \in R \tag{3.10}$$

A Restrição (3.11) define o número de professores associados a uma disciplina.

$$\sum_{p \in P} z_{pi} = n_i, \forall i \in D \tag{3.11}$$

A Restrição (3.12) impede que disciplinas associadas a um professor partilhem *slots*.

$$z_{pi} + z_{pj} + x_{ij} \leq 2, \forall i, j \in D, \forall p \in P \tag{3.12}$$

A Restrição (3.13) impede que disciplinas obrigatórias de um mesmo semestre e de um mesmo curso partilhem *slots*.

$$x_{ij} = 0, \forall i, j \in D_{kc}^{ob}, \forall k \in K_c, \forall c \in C \tag{3.13}$$

A Restrição (3.14) diz que se duas disciplinas partilham pelo menos um *slot*, então elas têm choque de horários.

$$y_{is} + y_{js} \leq x_{ij} + 1, \forall s \in S, \forall i, j \in D \tag{3.14}$$

As Restrições (3.15), (3.16) e (3.17) referem-se às disciplinas obrigatórias. Elas dizem que se as cargas horárias das disciplinas obrigatórias de um semestre de um curso são superiores à carga horária do turno primário desse curso, então essas disciplinas devem ser alocadas no horário primário e secundário do curso, de forma que o turno primário esteja completamente preenchido; caso contrário, todas as disciplinas devem ser alocadas no horário primário do curso. Note que na Restrição (3.16), quando $u_{ck} = 0$, temos que $(1 - u_{kc}) * 10 = 10$, o que garante que o turno primário do curso c esteja preenchido por disciplinas obrigatórias do semestre k do mesmo curso. Já na Restrição (3.17), quando $u_{ck} = 0$, $(1 - u_{kc}) * (-20 + \sum_{i \in D_{kc}^{ob}} crd_i)$ é igual ao número de créditos excedentes, o que garante que o excedente de créditos do semestre seja alocado no turno secundário do curso.

$$2 * \sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{pri_c}} y_{is} \geq u_{kc} * \sum_{i \in D_{kc}^{ob}} crd_i, \forall k \in K_c, \forall c \in C \tag{3.15}$$

$$\sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{pri_c}} y_{is} \geq (1 - u_{kc}) * 10, \forall k \in K_c, \forall c \in C \tag{3.16}$$

$$2 * \sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{sec_c}} y_{is} \geq (1 - u_{kc}) * (-20 + \sum_{i \in D_{kc}^{ob}} crd_i), \forall k \in K_c, \forall c \in C \tag{3.17}$$

A Restrição (3.18) diz que o número de *slots* que uma disciplina deve ser alocada é igual a metade de seus créditos. Essa restrição além de garantir que todas as disciplinas serão

alocadas, também garante que serão alocadas à quantidade adequada de *slots*.

$$2 * \sum_{s \in S} y_{is} = crd_i, \forall i \in D \tag{3.18}$$

A Restrição (3.19) fixa o número de *slots* de um professor em proporção ao número de *slots* das disciplinas que ele está associado.

$$2 * \sum_{s \in S} w_{ps} = \sum_{i \in D} crd_i * z_{pi}, \forall p \in P \tag{3.19}$$

A Restrição (3.20) diz que se um professor está associado a uma disciplina, ele também está associado aos *slots* dessa disciplina. A garantia de que essa restrição funciona é dada pelas Restrições (3.18) e (3.19).

$$y_{is} + z_{pi} \leq w_{ps} + 1, \forall p \in P, \forall s \in S, \forall i \in D \tag{3.20}$$

A Restrição (3.21) define que uma disciplina deve compartilhar *slots* com um conjunto prefixado de disciplinas.

$$x_{ij} = 1, \forall j \in T_i, \forall i \in D \tag{3.21}$$

A Restrição (3.22) fixa os eventuais horários predeterminados de uma disciplina.

$$y_{is} = 1, \forall s \in H_i, \forall i \in D \tag{3.22}$$

A Restrição (3.23) define os grupos de *slots* aos quais as disciplinas devem ser alocadas. Essa Restrição tem a função de reduzir os espaço de busca eliminando a simetria nas possibilidades de alocação das disciplinas. É importante notar que para que ela funcione adequadamente não deve haver intersecção entre os grupos de *slots* de mesmo tamanho.

$$y_{is_1} = y_{is_2}, \forall s_1, s_2 \in g, \forall g \in G_{(crd_i/2)}, \forall i \in D \tag{3.23}$$

As demais restrições são de integridade.

$$a_{ij} \in \mathbb{N} \tag{3.24}$$

$$b_{pi} \in \mathbb{R} \tag{3.25}$$

$$max_p \in \mathbb{N} \tag{3.26}$$

$$min_p \in \mathbb{N} \tag{3.27}$$

$$crd_i \in \mathbb{N} \tag{3.28}$$

$$n_i \in \mathbb{N} \tag{3.29}$$

$$x_{ij} \in \{0, 1\} \tag{3.30}$$

$$y_{is} \in \{0,1\} \tag{3.31}$$

$$z_{pi} \in \{0,1\} \tag{3.32}$$

$$w_{ps} \in \{0,1\} \tag{3.33}$$

$$u_{kc} \in \{0,1\} \tag{3.34}$$

$$v_p \in \{0,1\} \tag{3.35}$$

Abaixo está uma apresentação completa do modelo desenvolvido:

Função Objetivo:

$$\min \alpha * \sum a_{ij} * x_{ij} + (\alpha - 1) * \sum b_{pi} * z_{pi}$$

Restrições:

$$\sum_{i \in D} crd_i * z_{pi} \leq max_p, \forall p \in P \tag{3.1}$$

$$\sum_{i \in D} crd_i * z_{pi} \geq min_p, \forall p \in P \tag{3.2}$$

$$\sum_{q \in \{0,2,4\}} max(w_{p(s+q)}, w_{p(s+q+1)}) \leq 2, \forall s \in \{1,7,13,19,25\}, \forall p \in P \tag{3.3}$$

$$w_{ps} + w_{p(s+1)} \leq 1, \forall s \in \{6,12,18,24\}, \forall p \in P \tag{3.4}$$

$$w_{p14} + w_{p15} \leq 1, \forall p \in P \tag{3.5}$$

$$\sum_{s \in \{1,2,\ldots,6\}} w_{ps} \leq 6 * v_p, \forall p \in P \tag{3.6}$$

$$\sum_{s \in \{25,26,\ldots,30\}} w_{ps} \leq 6 * (1 - v_p), \forall p \in P \tag{3.7}$$

$$v_{p_1} = v_{p_2}, \forall \{p_1, p_2\} \in A \tag{3.8}$$

$$z_{pi} = 1, \forall i \in D_p, \forall p \in P \tag{3.9}$$

$$w_{ps} = 0, \forall s \in S_r, \forall p \in P_r, \forall r \in R \tag{3.10}$$

$$\sum_{p \in P} z_{pi} = n_i, \forall i \in D \tag{3.11}$$

$$z_{pi} + z_{pj} + x_{ij} \leq 2, \forall i, j \in D, \forall p \in P \tag{3.12}$$

$$x_{ij} = 0, \forall i, j \in D_{kc}^{ob}, \forall k \in K_c, \forall c \in C \tag{3.13}$$

$$y_{is} + y_{js} \leq x_{ij} + 1, \forall s \in S, \forall i, j \in D \tag{3.14}$$

$$2 * \sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{pri_c}} y_{is} \geq u_{kc} * \sum_{i \in D_{kc}^{ob}} crd_i, \forall k \in K_c, \forall c \in C \tag{3.15}$$

$$\sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{pri_c}} y_{is} \geq (1 - u_{kc}) * 10, \forall k \in K_c, \forall c \in C \tag{3.16}$$

$$2 * \sum_{i \in D_{kc}^{ob}} \sum_{s \in S_{sec_c}} y_{is} \geq (1 - u_{kc}) * (-20 + \sum_{i \in D_{kc}^{ob}} crd_i), \forall k \in K_c, \forall c \in C \tag{3.17}$$

$$2 * \sum_{s \in S} y_{is} = crd_i, \forall i \in D \tag{3.18}$$

$$2 * \sum_{s \in S} w_{ps} = \sum_{i \in D} crd_i * z_{pi}, \forall p \in P \tag{3.19}$$

$$y_{is} + z_{pi} \leq w_{ps} + 1, \forall p \in P, \forall s \in S, \forall i \in D \tag{3.20}$$

$$x_{ij} = 1, \forall j \in T_i, \forall i \in D \tag{3.21}$$

$$y_{is} = 1, \forall s \in H_i, \forall i \in D \tag{3.22}$$

$$y_{is_1} = y_{is_2}, \forall s_1, s_2, \forall g \in G_{(crd_i/2)}, \forall i \in D \tag{3.23}$$

$$a_{ij} \in \mathbb{N} \tag{3.24}$$

$$b_{pi} \in \mathbb{R} \tag{3.25}$$

$$max_p \in \mathbb{N} \tag{3.26}$$

$$min_p \in \mathbb{N} \tag{3.27}$$

$$crd_i \in \mathbb{N} \tag{3.28}$$

$$n_i \in \mathbb{N} \tag{3.29}$$

$$x_{ij} \in \{0,1\} \tag{3.30}$$

$$y_{is} \in \{0,1\} \tag{3.31}$$

$$z_{pi} \in \{0,1\} \tag{3.32}$$

$$w_{ps} \in \{0,1\} \tag{3.33}$$

$$u_{kc} \in \{0,1\} \tag{3.34}$$

$$v_p \in \{0,1\} \tag{3.35}$$

4 RESULTADOS

Implementamos o modelo produzido utilizando a linguagem de programação *C++11* (CPLUSPLUS, 2016) em conjunto com o *QT 5.2.1* (QT, 2016) e o *solver CPLEX 12.6.1* da *IBM* (IBM, 2016) afim de avaliar o desempenho do modelo para instâncias de diferentes tamanhos. Escolhemos a linguagem *C++* para preparar as instâncias para o *CPLEX* por conta do seu alto desempenho. Utilizamos as bibliotecas de acesso aos dados do *QT*, por apresentarem facilidade de uso. O *CPLEX* foi utilizado por se tratar de um *solver* robusto, que realiza de forma automática a conversão de restrições não lineares para restrições lineares e dá suporte a execução paralela (*threads*), ambos recursos utilizados nos experimentos deste trabalho.

Todos os experimentos foram realizados em um computador com processador *Intel(R) Xeon(R) CPU E31240 @ 3.30GHz*, 8 GB de memória RAM e com 8 GB na partição de *swap*, executando o sistema operacional Ubuntu 14.04.5 LTS.

Os testes foram realizados sobre as ofertas de disciplinas da UFC-Quixadá do semestre 2016.2. Os dados contam com 108 ofertas de disciplinas divididas em 6 cursos. São ao todo 58 professores e a preferência de um professor por uma disciplina foi dada pelo número de vezes que esse professor ministrou essa disciplina no campus. O número de alunos nos choques de horários das disciplinas é definido de forma aleatória, já que não foi possível conseguir essas informações para os testes. Primeiramente é atribuído a cada disciplina um valor entre 10 e 50 (incluindo os extremos) referente ao número de alunos que podem cursar cada disciplina. Depois, para cada par (a,b) de disciplinas, sorteamos um valor entre 0 e o mínimo do número de alunos de a e b para definir a interseção de alunos entre as duas disciplinas. O valor definido para a interseção passa a ser então o número de alunos que podem cursar as duas disciplinas.

A partir do conjunto de dados obtido, utilizando a linguagem de programação *Python 2.7* (PYTHON, 2016) para automatizar o processamento dos dados e o banco de dados *SQLite 3.11* (SQLITE, 2016) para armazená-los, ambos escolhidos pela facilidade de uso. Preparamos aleatoriamente instâncias com diferentes proporções dos dados reais, ou seja, instâncias com 10%, 20%, ..., 90% e 100% dos professores e disciplinas escolhidos de forma aleatória (exemplo: 10% dos professores e 10% das disciplinas). Aplicamos essas instâncias para valores de $\alpha \in \{0, 0.5, 1\}$, e definimos um *timeout* de 10 horas.

É importante lembrar que as alocações produzidas com $\alpha = 1$ não levam em consideração a preferência dos professores e são possivelmente alocações indesejáveis e não aplicáveis à UFC-Quixadá. Os testes realizados para esse valor têm intuito apenas de avaliar o

desempenho do modelo quando são desconsideradas as preferências por disciplinas dos professores.

A Tabela 1 apresenta o resultado dos experimentos para todas as instâncias. A primeira coluna se refere ao nome das instâncias, a segunda coluna ao valor α aplicado sobre as instâncias, a terceira ao tempo total de execução em segundos (*real time* + *sync time* + *wait time* + *root node processing*, fornecidos pelo *CPLEX*), e a quarta coluna apresenta o resultado obtido para cada teste.

Tabela 1 – Tabela de Experimentos.

Instância	α	Tempo (segundos)	Resultado
10%	0	1.46	Ótimo
	0.5	0.30	Ótimo
	1	0.25	Ótimo
20%	0	3.61	Ótimo
	0.5	1943.68	Ótimo
	1	1353.89	Ótimo
30%	0	13.51	Ótimo
	0.5	1297.13	Solução viável (Excedeu a memória)
	1	1701.42	Solução viável (Excedeu a memória)
40%	0	43.79	Ótimo
	0.5	2255.91	Solução viável (Excedeu a memória)
	1	2043.66	Solução viável (Excedeu a memória)
50%	0	195.00	Ótimo
	0.5	3405.17	Solução viável (Excedeu a memória)
	1	8915.71	Solução viável (Excedeu a memória)
60%	0	170.55	Ótimo
	0.5	8008.68	Solução viável (Excedeu a memória)
	1	20669.59	Solução viável (Excedeu a memória)
70%	0	375.21	Ótimo
	0.5	9585.77	Solução viável (Excedeu a memória)
	1	6608.70	Nenhuma solução (Excedeu a memória)
80%	0	3386.09	Ótimo
	0.5	27010.67	Solução viável (Excedeu a memória)
	1	9018.20	Nenhuma solução (Excedeu a memória)
90%	0	7369.57	Nenhuma solução (Excedeu a memória)
	0.5	10763.45	Nenhuma solução (Excedeu a memória)
	1	10153.70	Nenhuma solução (Excedeu a memória)
100%	0	4611.18	Nenhuma solução (Excedeu a memória)
	0.5	36001.29	Solução viável (Timeout)
	1	21974.03	Nenhuma solução (Excedeu a memória)

Fonte – Produzido pelo autor

Na Tabela 1 podemos perceber que quando $\alpha = 0$, a implementação obteve soluções ótimas para as instâncias de tamanho até 80% da oferta total do semestre 2016.2, mas para instâncias de maior tamanho, nenhuma solução foi encontrada. Para $\alpha = 0.5$, nenhuma solução ótima foi encontrada para as instâncias com mais de 20% dos dados, mas obtivemos pelo menos uma solução viável para todas as instâncias com exceção da de 90%. Para $\alpha = 1$, nenhuma solução ótima foi encontrada para as instâncias com mais de 20%, e nas instâncias com mais de 60% dos dados, não foi encontrada nenhuma solução viável.

Quanto às alocações geradas, nas instâncias menores, ocorrem muitos casos em que professores são associados à disciplinas que estão fora da sua área de estudo, como por exemplo, um professor de otimização associado à disciplina de fotografia. Isso é esperado, pois sendo os professores e disciplinas escolhidos de forma aleatória para compor a instância, não temos a garantia de haver uma proporção adequada entre os professores de uma área de estudo e as disciplinas dessa mesma área. É importante lembrar, que essas instâncias tem o intuito apenas de avaliar o desempenho do modelo. Na solução viável encontrada para a instância com 100% dos dados, a alocação de professores em disciplinas foi quase em sua totalidade semelhante às alocações que são aplicadas atualmente no campus. Apesar disso houveram algumas alocações aparentemente indesejáveis entre professores e disciplinas (professores alocados à disciplinas às quais não estão haptos a lecionar). Considerando que a solução encontrada não é a solução ótima, e que o programa parou por exceder o tempo limite de execução, isso já era esperado para essa situação.

Como ilustração, a Figura 2 apresenta a alocação gerada para as disciplinas do quarto semestre do curso de Ciência da Computação para a instância com 100% dos dados e $\alpha = 0.5$.

Figura 2 – Alocação das disciplinas do quarto semestre do curso de Ciência da Computação

CIENCIA DA COMPUTACAO - 4 Semestre						
--		SEGUNDA	TERA	QUARTA	QUINTA	SEXTA
MANH	AB	Alg. Linear (Prof_11)	Alg. Linear (Prof_11)	LIP (Prof_35)	PAA (Prof_14)	PAA (Prof_14)
	CD	LIP (Prof_35)	FBD (Prof_53)	FBD (Prof_53)	APS (Prof_39)	APS (Prof_39)
TARDE	AB					
	CD					
NOITE	AB					
	CD					

Fonte – Elaborada pelo autor

5 CONSIDERAÇÕES FINAIS

Neste trabalho, desenvolvemos um modelo de PI para o problema de alocação de professores e disciplinas da UFC-Quixadá. O modelo desenvolvido busca maximizar a preferência geral dos professores e minimizar o número de alunos que podem cursar pares de disciplinas com choque de horário.

Inicialmente fizemos um levantamento das restrições de alocação aplicadas atualmente na UFC-Quixadá. Modelamos essas restrições como um problema de PI, buscando tratar a maximização das preferências dos professores e a minimização dos choques de horários na função objetivo do modelo. Implementamos o modelo e realizamos testes convertendo as ofertas das disciplinas da UFC-Quixadá de 2016.2 como entrada para a implementação.

A partir dos testes realizados, ficou claro que a abordagem proposta por este trabalho não é efetiva para solucionar o problema de alocação da UFC-Quixadá. Mesmo obtendo uma solução viável para 100% dos dados quando $\alpha = 0.5$, não há garantia de que as possíveis soluções viáveis encontradas pela implementação, sejam melhores do que as já produzida atualmente no Campus. Além disso, através do resultado para a instância com 90% dos dados e com valor $\alpha = 0.5$, podemos ver que não há garantia nem de que encontremos uma solução viável antes que haja estouro de memória no processo de busca mesmo para instâncias menores.

Podemos concluir que é necessário buscar uma estratégia mais eficiente se quisermos realizar a alocação de professores e disciplinas para uma quantidade de dados de tamanho igual à que é alocada na UFC-Quixadá atualmente.

Como trabalhos futuros, propomos os seguintes:

- Observando os experimentos, percebemos que para instâncias menores (de 10% e 20%) o modelo se mostrou eficaz para encontrar uma solução ótima. Baseado nisso, uma possível abordagem para resolver o problema pode ser a aplicação de um modelo de PI para fazer a alocação das disciplinas de cada curso separadamente. Essa abordagem pode se mostrar efetiva, apesar de não garantir uma solução ótima para o problema de alocação completa do campus.
- Comparar nosso método de alocação com outros métodos propostos na literatura. Existem diversas abordagens para problema de alocação que utilizam Inteligência Artificial. Talvez seja interessante a implementação de algum desses métodos realizando uma comparação com o nosso.
- Dado que a alocação de professores na UFC-Quixadá já é feita de forma

automatizada, algo interessante seria o desenvolvimento de um sistema com foco na facilidade de uso para usuários não especialistas no assunto. Isso eliminaria a necessidade de uma pessoa especialista realizando a tarefa de preparação dos dados, restando para essa pessoa apenas a tarefa de supervisionar todo o processo.

REFERÊNCIAS

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

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

DODÓ, A. A. **Aplicação da Teoria dos Jogos na Resolução do Problema de Alocação de Professores em Disciplinas**. [S.l.: s.n.], 2011.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. [S.l.]: McGraw Hill, 2010.

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

ITC. **International Timetabling Competition**. 2007. Disponível em: <http://www.cs.qub.ac.uk/itc2007/>. Acesso em: 26 Maio 2016.

KINGSTON, J. H. A tiling algorithm for high school timetabling. In: **Practice and Theory of Automated Timetabling V**. [S.l.]: Springer, 2004. p. 208–225.

KINGSTON, J. H. Educational timetabling. In: **Automated Scheduling and Planning**. [S.l.]: Springer, 2013. p. 91–108.

LACH, G.; LÜBBECKE, M. E. Curriculum based course timetabling: new solutions to udine benchmark instances. **Annals of Operations Research**, Springer, v. 194, n. 1, p. 255–272, 2012.

LEWIS, R. A survey of metaheuristic-based techniques for university timetabling problems. **OR Spectrum**, v. 30, n. 1, p. 167–190, 2008. ISSN 1436-6304. Disponível em: <http://dx.doi.org/10.1007/s00291-007-0097-0>.

LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**. [S.l.]: Springer, 1984. v. 2.

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.

SCHRIJVER, A. **Theory of linear and integer programming**. [S.l.]: John Wiley & Sons, 1998.

ŠLECHTA, P. Decomposition and parallelization of multi-resource timetabling problems. In: **Practice and Theory of Automated Timetabling V**. [S.l.]: Springer, 2004. p. 177–189.

SQLITE. 2016. Disponível em: <https://www.sqlite.org/>. Acesso em: 26 Maio 2016.

YANG, Y.; PETROVIC, S. A novel similarity measure for heuristic selection in examination timetabling. In: **Practice and Theory of Automated Timetabling V**. [S.l.]: Springer, 2004. p. 247–269.