

UM ALGORITMO DE ESCALONAMENTO DE HORÁRIOS PARA OS CURSOS DE INFORMÁTICA E CIÊNCIA DA COMPUTAÇÃO DO IFCE-CAMPUS ARACATI, BASEADO EM PREFERÊNCIAS DOCENTES

Lucas Porfírio da Silva *

Diego Rocha Lima**

RESUMO

A construção de horários escolares é uma tarefa difícil para a maior parte das instituições de ensino, pois possui vários tipos de restrições e objetivos que variam de acordo com as características de cada instituição. No IFCE Campus Aracati uma das restrições é quantos e quais dias o docente, preferencialmente, realiza suas atividades no *Campus*. A ideia é realizar todas as suas atividades presenciais em dias e turnos consecutivos. Neste trabalho criamos uma solução com base em algoritmo guloso e heurísticas de busca local para apresentar possíveis soluções a partir de um grupo de restrições duras e suaves. A principal restrição é a preferência dos professores por dias de trabalho e respectivas disciplinas. Ao final, temos um programa que monta algumas grades de horários viáveis e as apresenta de forma organizada em arquivos aos docentes, para apreciação e avaliação dos mesmos.

Palavras-chave: Escalonamento de horários, Busca local, Preferência docente.

ABSTRACT

The construction of school timetables is a hard task for most educational institutions, as it has several types of restrictions and objectives that vary in each institution. At IFCE campus Aracati one of the restrictions is the number of days that the teacher is in person on the campus. In this work we create a solution based on greedy algorithm and heuristics to present possible solutions from a group of hard and light constraints. The main restriction is the teachers preference for working days. At the end, we have a program that assembles some viable grids and presents them in an organized way in files to the teachers, for their appreciation and evaluation.

Keywords: *Timetable schedules, Greedy algorithm, Teacher preference.*

Data de submissão para publicação: 02 jun. 2021.

* Autor, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil.

** Orientador, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil

1 INTRODUÇÃO

À medida que as instituições públicas de ensino superior se desenvolvem, elas estão sujeitas a enfrentar problemas, podendo ser relacionados a orçamento, condições de trabalho, carga horária, equipe docente reduzida ao mínimo necessário para o funcionamento de seus cursos ou ainda, uma grade de horários válida e que ao mesmo tempo atenda a preferência dos docentes. Dentre estas restrições destaca-se a última. Tais condições requerem que as organizações melhorem a criação de horários visando reduzir os custos de manutenção e maximizar o uso dos recursos disponíveis, bem como garantir o bom funcionamento dos cursos, sem deixar de oferecer qualquer disciplina por falta de espaço, horário ou docente.

Neste trabalho aborda-se a melhoria nas condições de trabalho da equipe docente otimizando as horas em que estão presente na instituição e focando na opção de escolha de dias para realizar atividades no campus. Esse problema na literatura é conhecido como escalonamento de horários universitários (ABDELHALIM; KHAYAT, 2016). O problema de escalonamento de horários no cenário escolar/institucional, visa encontrar um método de alocação adequado para eventos em horários e salas predefinidos, em que todas as restrições do problema sejam satisfeitas.

O problema de escalonamento de horários já é complexo levando em consideração apenas as restrições citadas acima, porém a abordagem demonstrada nesse trabalho conta com restrições específicas como as preferências dos professores para os dias trabalhados no *campus* e disciplinas ministradas.

O problema de escalonamento de horários faz parte do conjunto NP-difícil de otimização combinatória híbrida (MAHIBA; DURAI, 2012)(ABBASZADEH; SAEEDVAND, 2014)(GARREY; JOHNSON, 2002). A tarefa de programar horários ocorre no início de cada semestre nas escolas e universidades, incluindo-se a alocação de eventos em uma certa quantidade de horários e salas fixa. Neste problema deve-se satisfazer restrições duras e suaves durante a alocação dos eventos de acordo com recursos disponíveis.

Nem sempre o escalonamento traz uma total satisfação das restrições suaves. Mas, quando é possível satisfazer as restrições suaves, também é possível aumentar e fomentar a qualidade das tabelas de horários geradas (REDL, 2004)(ASMUNI, 2008)(OBIT, 2010).

Restrições duras devem ser satisfeitas totalmente para gerar uma solução viável sem conflitos. Restrições suaves estão relacionadas à qualidade da solução que, quando maximizada, gera a melhor solução factível. Diferente das restrições duras, as restrições suaves podem ser quebradas sem comprometer a factibilidade da solução.

O problema no Instituto Federal do Ceará *campus* Aracati se dá quando os professores possuem dedicação exclusiva, porém residem em outras cidades, o que faz com que seja melhor aproveitar ao máximo os dias em que estão na cidade na qual trabalham. Considera-se ainda que alguns docentes dão aula em programas de pós-graduação ou em outros *campi* que ficam localizados em cidades diferentes.

No geral, os professores cumprem toda sua carga horária semanal de trabalho em 3 dias

consecutivos no *campus*. Não que isso seja uma regra mas, de acordo com o escalonamento de horários, isso acaba sendo viável.

O desafio é alinhar todos os eventos e recursos de modo a não infringir as restrições duras do algoritmo e garantir parte das restrições suaves. Dessa forma, o algoritmo possibilita a criação de cenários (grade de horários) diferentes para cada professor da instituição. Atualmente o campus faz uso de um *software* para criação da grade, mas que ao final algumas mudanças pontuais são feitas de forma manual com o intuito de garantir o melhor aproveitamento de dias e horários para cada professor no *campus*, de acordo com suas disciplinas.

O algoritmo aqui apresentado mostra de forma mais eficiente algumas opções de possíveis grades de horários do semestre atual, distribuindo-se de maneira eficiente os dias e horários do professor no *campus*. Fazendo assim com que os professores possam analisar mais de uma opção factível, trabalho que antes seria feito manualmente.

Este trabalho está dividido da seguinte forma. Na Seção 2 é apresenta-se a fundamentação teórica e na Seção 3 demonstra-se alguns trabalhos relacionados nos quais se apresenta-se outras abordagens para a solução do problema de escalonamento de horários. Já na Seção 4 apresenta-se a metodologia, em que descreve-se os métodos e heurísticas utilizadas em nosso algoritmo. Na Seção 5 estão os resultados obtidos com a implementação do algoritmo. Por fim a Seção 6 traz a conclusão e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O problema de escalonamento de horários é um problema de otimização combinatória NP-difícil (MAHIBA; DURAI, 2012). No problema abordado a qualidade da solução proposta é mensurada a partir da quantidade de restrições suaves respeitadas e a viabilidade da solução somente caso, nenhuma restrição dura seja infringida. (ASMUNI, 2008)

O problema de escalonamento de horários baseia-se em atribuir uma dupla ou tupla factível de atributos que otimize um conjunto de métricas e indicadores. Com o objetivo de minimizar as diferenças de tempo, maximizar a utilização de espaço e minimizar os custos relevantes ao uso de recursos. Assim, encontrar um algoritmo eficiente para esses problemas, especialmente quando se trata de instituições com grande quantidade de cursos, também não é tarefa fácil porém, boas soluções podem ser fornecidas quando o problema é melhor compreendido em termos de complexidade (ABDELHALIM; KHAYAT, 2016)..

Supõe-se que um semestre, em que tem que ser definido um horário para trinta disciplinas, com quatro aulas cada turno, por uma semana. Consideraremos uma semana com cinco dias de trabalho, com dez períodos de tempo de aulas todos os dias, e em que só existem três turmas. O espaço de pesquisa para a construção de um horário semanal é tão grande como:

$$(5 \times 10 \times 3)^{4 \times 30} \approx 1.35 \times 10^{261}$$

Esse resultado representa a quantidade aproximada de possibilidades com uma instancia menor do problema. o custo para gerar todas as saídas é alto. Essa situação hipotética é menor que a situação real que tratamos aqui.

Dentre os dados necessários para a solução desse problema temos:

- Eventos: Uma atividade com horário marcado que envolve professores, turma e estudantes;
- Horário: Que indica um intervalo de tempo. Por exemplo:09:00-10:00 da manhã;
- Pessoas: Agrupados de pessoas como professores e estudantes de cada turma;
- Restrições: Regras que se dividem entre duras e suaves.

Como visto no trabalho(BABAEI; KARIMPOUR; HADIDI, 2015).

As restrições duras são do tipo que “quebram” o sistema de escalonamento se não forem satisfeitas. Por exemplo, se um professor estiver alocado em dois eventos no mesmo horário. Pois não é possível que alguém ministre duas disciplinas no mesmo horário em locais diferentes. Restrições duras não devem ser quebradas de forma alguma, e as usadas no nosso trabalho são:

- Professores não podem ser alocados para duas disciplinas no mesmo horário;
- Dois professores não podem estar alocados na mesma turma no mesmo horário;
- Alocar as disciplinas selecionadas nos dias preferidos por cada professor;
- Professores não podem ser alocados para ministrar disciplinas no turno da manhã e da noite no mesmo dia ou os três turnos no mesmo dia ;
- Respeitando os dias impossíveis, que o professor não pode estar presente no *campus*

Já as restrições suaves fazem com que a solução se torne melhor, mas quando infringida não gera quebra no sistema, apenas geram horários de menor qualidade. Por exemplo, os horários de um professor são AB turno da tarde e CD turno da noite, pode ser considerado não tão bom devido ao tempo ocioso, mas ainda assim é um horário possível(ALGHAMDI et al., 2020). Já as restrições suaves são as que podem ser quebradas mas, quando respeitadas a solução é melhor. Às usadas nesse trabalho são:

- Os horários não devem ser alocados em horários que o professor indica como dias possíveis mas não preferíveis;
- Os horários não devem ser alocados em horários que o professor indica como dias preferíveis;
- reduzir o tempo ocioso do professor entre disciplinas.

No contexto do IFCE *campus* Aracati o problema se mostra de pequena escala no tamanho da instância, se comparado a outras universidades ou institutos. Mas se mostra de alta complexidade quando se trata da quantidade de restrições.

Pode-se facilmente observar que quão maior a instância, ou seja, quantidade de professores, salas, disciplinas e recursos, mais difícil fica para o algoritmo montar uma solução viável. Isso porque a quantidade de dimensões aumenta.

Os parâmetros selecionados para este trabalho incluem as preferências dos professores para dias de trabalho e dias nos quais não podem estar no campus, visando atribuir da melhor forma suas disciplinas e horários. Otimizar o tempo de permanência dos professores no campus, permite que a carga horária seja cumprida de forma integral nos dias em que estiver no campus.

3 TRABALHOS RELACIONADOS

O trabalho *A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA)* (ABDELHALIM; KHAYAT, 2016) é focada a minimização dos espaços vagos de tempo entre as aulas, a maximização da utilização dos recursos e minimização dos custos para manter cada recurso é utilizada a restrição de não haver duas aulas usando o mesmo recurso, a capacidade da sala e o número de estudantes matriculados. São usadas preferências do professor para os recursos como restrição leve.

O trabalho *Solving a Capacitated Exam Timetabling Problem Instance Using a Bi-objective NSGA-II* (LEITE et al., 2015) os autores resolvem uma instância de 80 cursos em 5 programas de graus diferentes focando em reduzir o tempo de geração dos horários da universidade substituindo a abordagem manual por uma automatizada, o que aumentou as inscrições de estudantes de outros cursos nas grades compartilhadas.

O trabalho *Multi-objective Evolutionary Algorithms for Resource Allocation Problems* (DATTA; DEB; FONSECA, 2007) utilizaram o algoritmo NSGA-II resolvendo duas instâncias diferentes com o mesmo algoritmo tendo duas instâncias e dois conjuntos de restrições distintos. Na primeira, busca-se minimizar a quantidade de espaços livres entre as aulas dos alunos e maximizar a quantidade de espaço entre as aulas de cada professor. Na segunda instância maximizar o retorno econômico, minimizar as emissões de carbono e minimizar a erosão do solo. Neste trabalho são utilizadas várias restrições duras.

No artigo *Solving University Timetabling Problems Using Advanced Genetic Algorithms* (KAZARLIS; PETRIDIS; FRAGKOU, 2005) foi utilizado um algoritmo genético para resolver o problema com um método de representação indireta, usando dois algoritmos, um para classificação da instância baseado no sistema de prioridades e outro para “escala” os horários. A solução foi aplicada em uma universidade real e supera em muito a solução feitas a mão.

No artigo *Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm* (DATTA; DEB; FONSECA, 2006) é um trabalho expositivo em que os autores focam em defender a utilização de algoritmos genéticos para geração da escala de horários para o contexto geral do problema e defende que em todas as universidades os problemas são os mesmos e que nas instituições de ensino o trabalho de gerar os horários se mostra uma atividade laboral.

Os trabalhos citados nessa seção ressaltam a importância de se encontrar um método de geração de horários que satisfaça diversos requisitos e restrições tanto dos docentes quanto dos discentes. Isso em um tempo de computação viável. Dependendo do método, a geração de um horário pode consumir bastante tempo e memória.

Os trabalhos que utilizam meta heurísticas ou modelos matemáticos costumam apresentar resultados custosos pois testam todas as possibilidades, ou seja, testam soluções impossíveis que poderiam ter sido descartadas após serem consideradas inviáveis.

Com base nessas informações, nosso algoritmos tenta trazer soluções viáveis e em um tempo de computacional satisfatório e sem problemas quanto ao uso de memória.

4 METODOLOGIA

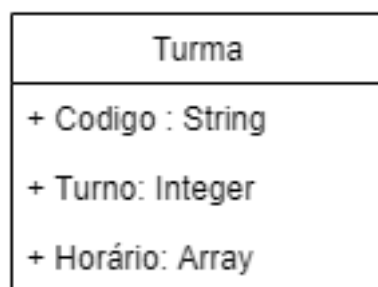
Esta seção explana-se a metodologia utilizada neste trabalho de forma a gerar soluções viáveis e satisfatórias. Descrevemos como estão organizados os dados, como foi implementado o algoritmo de acordo com todas as restrições e como são disponibilizados os resultados.

4.1 ORGANIZAÇÃO DE DADOS

Os dados do trabalho dividem-se em 3 partes: turmas, professores e disciplinas. Apresentaremos em detalhes nas subseções seguintes.

4.1.1 TURMAS

Figura 1 – modelo de dados da classe turma.



Fonte: Elaborada pelos autores (2021).

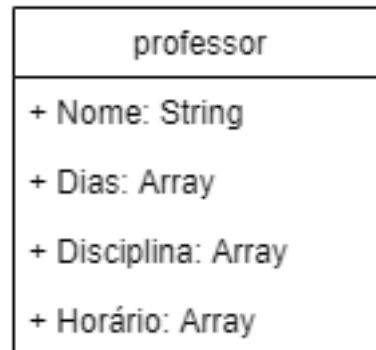
O Código na classe turma identifica uma turma, composto por sigla do curso e semestre da turma.

O turno representa o turno principal da turma

Já o horário um vetor com vinte posições, sendo dividido em 5 grupos de 4 posições cada, onde cada grupo representa um dia da semana subdividido nos horários diários A, B, C e D. Em cada posição do vetor horário da turma estão contidos o código da disciplina e o código professor, usado para realizar as possíveis trocas de horários.

4.1.2 PROFESSORES

Figura 2 – modelo de dados da classe professor.



Fonte: Elaborada pelos autores (2021).

Os dados da classe professor estão como mostra-se na Figura 2

Sendo o nome único para identificar o professor

O campo dias em formato de vetor com 5 posições em que classifica cada dia em:

- o tipo 1, prefere ser alocado nesse dia;
- o tipo 2, pode ser alocado nesse dia mas não prefere;
- o tipo 3, não pode ser alocado nesse dia.

Já o campo disciplina em formato de vetor com 5 posições em que armazena o código da disciplina que o professor vai lecionar no semestre, ordenado por prioridade definida pelo próprio professor.

Os horários dos professores dividem-se em 3 grupos de 20 posições onde cada seção representa um turno, e cada seção é sub-seccionada em 5 grupos de 4 posições onde cada subseção representa um dia da semana onde, cada subdivisão representa os horários diários A, B, C e D.

Assim para gerar equivalência no horário da turma e professor é necessária, dessa forma temos :

$$(turno \times 20) + (dia \times 4) + h = (\text{posição equivalente no horário o professor}).$$

Em que h a célula de horário A, B, C ou D de um dia de aula.

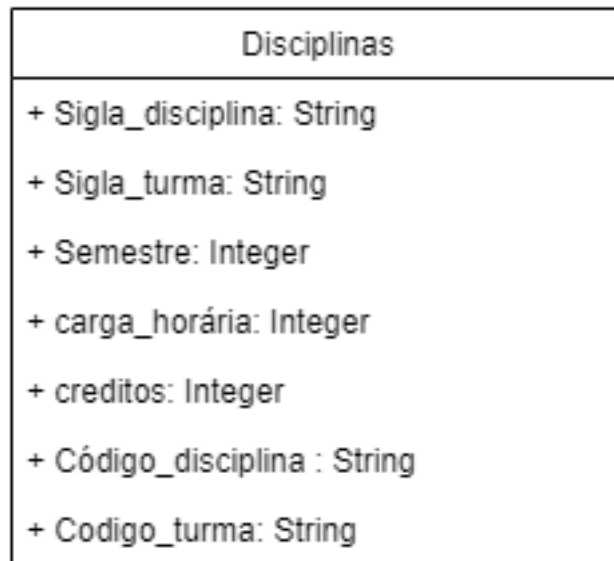
No horário do professor em cada posição estão contidos o código da disciplina e o índice da turma usado para realizar trocas.

4.1.3 DISCIPLINAS

Na figura 3 mostra o modelo da classe disciplina em que:

- sigla_disciplina: representa a sigla da disciplina em formato string;
- sigla_turma: representa a sigla da turma em formato string;

Figura 3 – modelo de dados da classe disciplina. Fonte: Autoral



Fonte: Elaborada pelos autores (2021).

- semestre: representa o semestre da turma em formato inteiro;
- carga_horária: representa a carga horária da disciplina por semestre em formato inteiro;
- créditos: representa a quantidade de células que essa disciplina deve ocupar em formato inteiro;
- código_disciplina: composto por sigla da disciplina carga horária e sigla da turma, gerando um identificador único em formato de string ;
- código_turma: composto por sigla da turma e semestre, para identificar uma turma em formato de string.

4.2 ALGORITMO PARA ALOCAÇÃO DE HORÁRIOS

Esta seção abrange os métodos e algoritmos utilizados para gerar soluções viáveis de acordo com as restrições dos professores.

Nesta trabalho utiliza-se os dados ordenados dos professores inicialmente, em que a ordem depende da quantidade de horas do professor e a condição de desempate é o professor que possuir a maior quantidade de disciplinas. Por exemplo, se temos 2 professores diferentes ambos com 120 horas, o professor que alocar mais disciplinas tem prioridade maior.

4.3 BUSCA LOCAL

Para a busca local, utilizam-se dois métodos que vamos chamar de *shift 1* e *shift 2*. As operações de *shifts* implementadas são realizadas quando um dos horários está livre, seja a turma

ou o professor, buscando a melhor possibilidade de troca. Todas as atribuições e trocas são realizadas em dias que o professor selecionou.

As trocas são selecionadas a partir do dia mais ocupado entre todos os professores, pois notamos que ao realizar trocas para dias mais ocupados resulta em melhores soluções.

A intenção dessa operação é reduzir a necessidade de troca a longo prazo, reduzindo-se o tempo ocioso no campus e maximizando-se a ocupação de horários nos dias selecionados.

4.3.1 *SHIFT 1*

Esse método recebe um professor, uma turma e um posição. Após isso realiza uma condicional caso for verdadeira, busca entre os horários do professor e turma da célula uma posição para realizar troca, caso encontrar essa posição e essa troca melhorar a saída, armazena os dados para troca e continua a busca.

Resumindo o método realiza uma troca no horário de um professor específico entre as turmas do mesmo, como visto na Figura 4 e no Algoritmo 1.

Figura 4 – Fluxograma *shift 1*.



Fonte: Elaborada pelos autores (2021).

4.3.2 *SHIFT 2*

Esse método recebe um professor, uma turma e um posição. Após isso realiza uma condicional caso for verdadeira busca entre os horários do professor da célula e no horário da turma uma posição para realizar troca, caso encontre essa posição e essa troca melhorar a saída, armazena os dados para troca e continua a busca.

Resumindo o método realiza uma troca dentro do horário de dois professores de uma turma específica, como visto nas Figuras 5 e 6 e no Algoritmo 2.

Algoritmo 1: Shift 1

```

if professor.horario[turno*20+pos] != "" and
turma.horario[pos] == "" then
  for dia2 in dias do
    if diasselecionados[dia2] == 1 then
      for h2 in horas do
        pos2 = dia2 × 4 + h2
        if professor.horario[turno*20+pos2] == "" then
          if turma2.horario[pos2] == "" then
            if ocupacaoatual > ocupacao[pos2] then
              armazena dados para troca

```

Figura 5 – fluxograma shift 2.



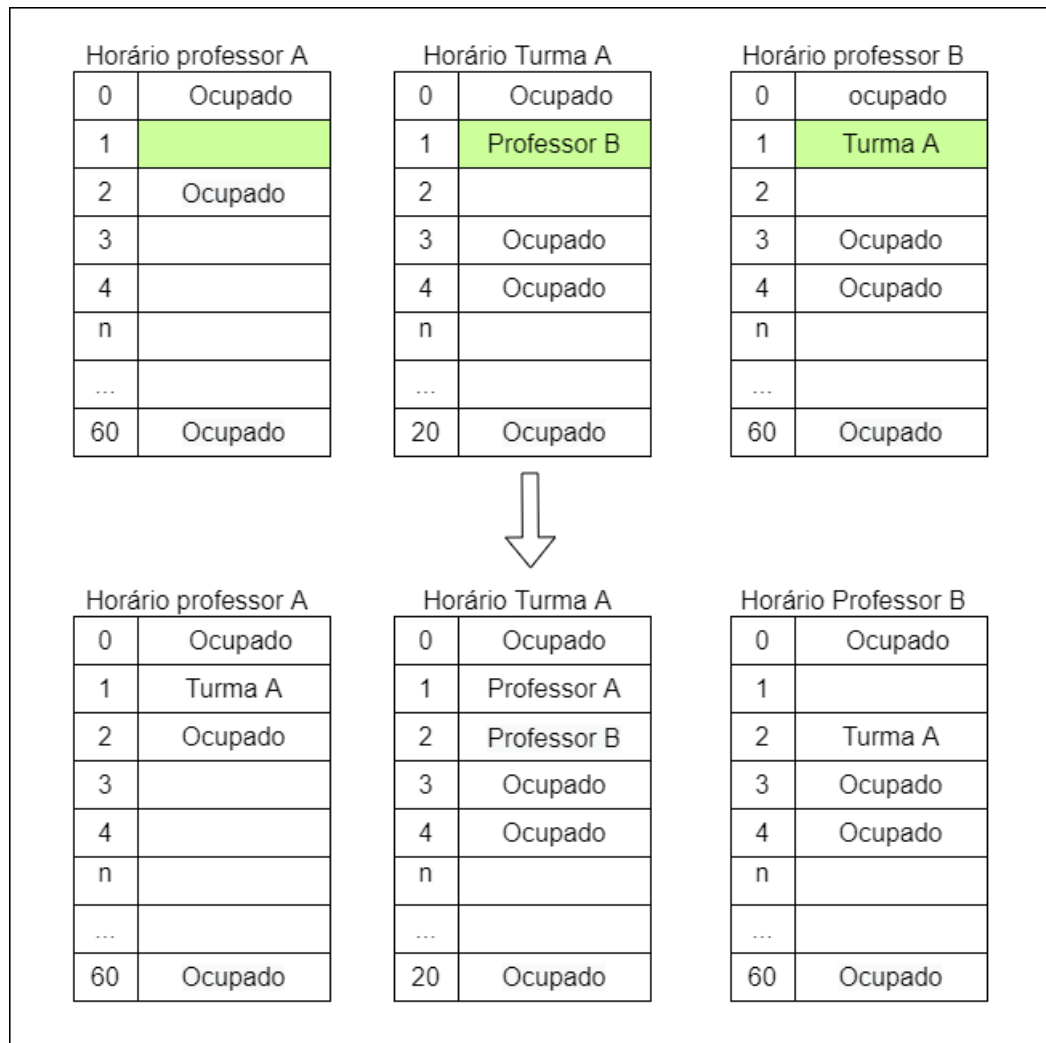
Fonte: Elaborada pelos autores (2021).

Algoritmo 2: shift 2

```

if professor.horario[turno*20+pos] == "" and
turma.horario[pos] != "" then
  for dia2 in dias do
    if professor2.diasselecionados[dia2] == 1 and professor.diasselecionados[dia2]
    == 1 then
      for h in horas do
        pos2 = dia2 × 4 + h
        if professor2.horario[turno*20+pos2] == "" and turma.horario[pos2]
        == "" then
          if ocupacaoatual > ocupacao[pos2] then
            armazena os dados para troca

```

Figura 6 – Representação do *shift* 2.

Fonte: Elaborada pelos autores (2021).

4.3.3 FUNÇÃO PRINCIPAL DO ALGORITMO

Inicialmente o algoritmo itera entre os horários do professor em dias de tipo 1. Na segunda fase executa busca local e na terceira fase itera nos horários do professor em dias de tipo 2, como ilustrado na Figura 7.

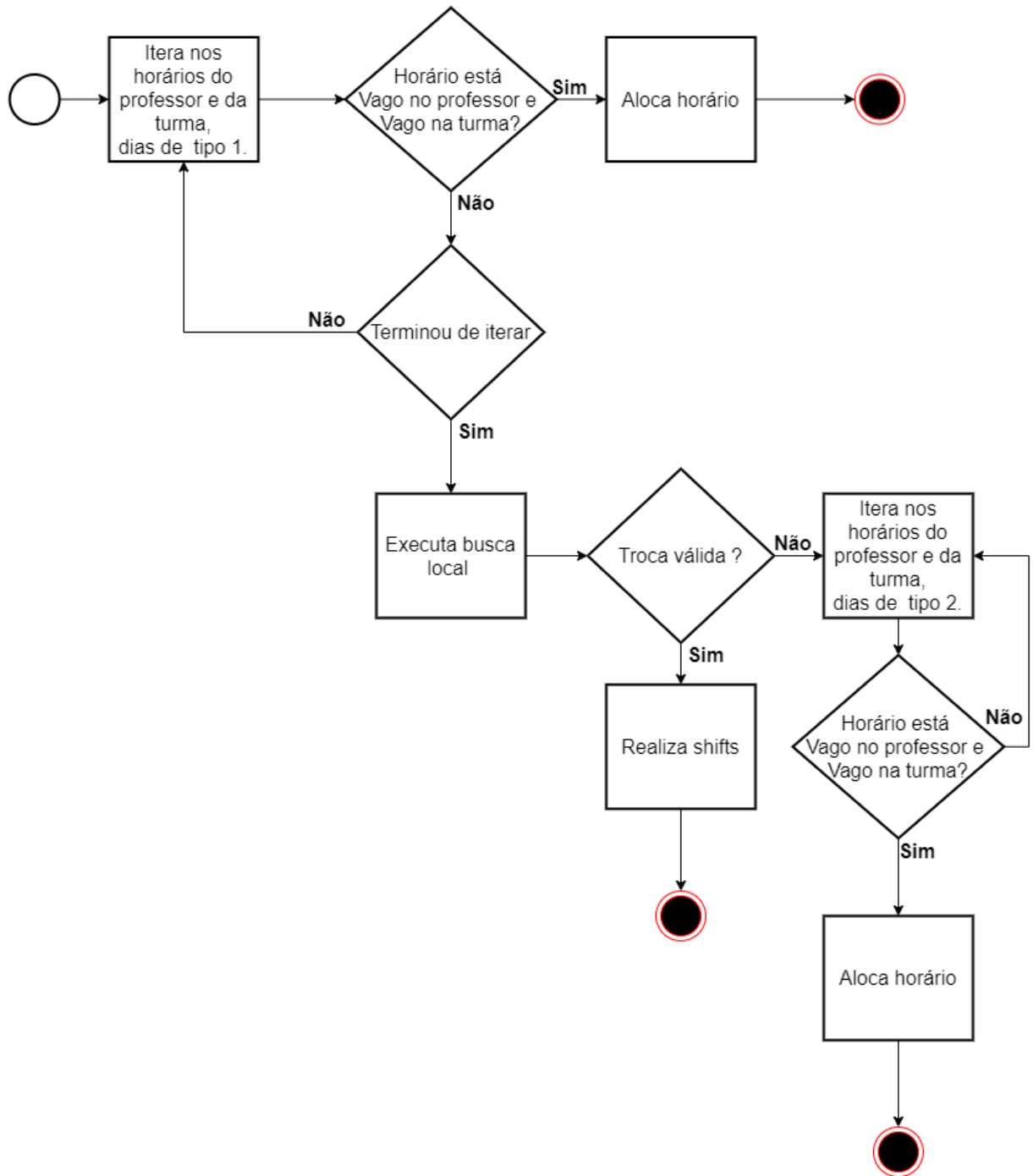
5 TESTES E RESULTADOS

A linguagem escolhida para a implementação foi o *python* por questão de conhecimento e para usar o *colab*¹, plataforma disseminada no processamento de dados.

Como resultado obteremos conjuntos de soluções satisfatórios que podem ser usados para a discussão entre coordenação e professores na escolha da grade de horários. As soluções/grades são geradas em um tempo de computação satisfatório, menos de 1 segundo. Assim as reuniões realizadas no início do semestre para definição dos horários podem ser otimizadas. Isso compa-

¹ <https://colab.research.google.com>

Figura 7 – Representação do algoritmo no geral.



Fonte: Elaborada pelos autores (2021).

Algoritmo 3: MAIN

```

dias=[0,1,2,3,4]
horas=[0,1,2,3]
ocupacaoatual = -1
for dia in dias do
    if professor.diasselecionados[dia] == 1 then
        for h in horas do
            pos = dia*4+h
            if professor.horario[turno*20+pos] == "" and
                turma.horario[pos] == "" then
                └ Atribui os dados nos campos necessários

for dia in dias do
    if professor.diasselecionados[dia] == 1 then
        for h in horas do
            pos = dia*4+h
            shift1()
            shift2()

if dados armazenados pelos shifts then
    └ Atribui os dados nos campos necessários

for dia in dias do
    if professor.diasselecionados[dia] == 0 then
        for h in horas do
            pos = dia*4+h
            if professor.horario[turno*20+pos] == "" and
                turma.horario[pos] == "" then
                └ Atribui os dados nos campos necessários

```

rando com o tempo que se leva para construir grades de todas as turmas e professores de maneira manual, ou ainda utilizando algum *software* proprietário.

Inicialmente, os testes foram realizados e apresentados ao eixo de Informação e Comunicação, comparando-se a grade utilizada no semestre letivo 2020.1, com algumas grades geradas pelo sistema. Vale salientar que esse semestre é anterior ao semestre que está em andamento em nosso campus, 2020.2. Essas informações estão contidas na tabela 1.

Após atestado o funcionamento e a qualidade da solução pretende-se expandir a instância para todos os eixos/cursos do *Campus*.

Professores são aqueles que ministram as disciplinas. Professores de eixo comum ministram disciplinas nesse eixo e nos outros eixos da instituição. Professores exclusivos do eixo ministram disciplinas específicas desse eixo. Disciplinas são parte do curso em semestres específicos. Turmas dividem-se em semestres de cada curso. Referente a Tabela 1

Na Figura 8 temos uma representação da grade de horários onde cada célula representa um horário, agrupado em pares que são separados pelo tempo de intervalo entre as aulas. Dessa forma, temos um horário exclusivo para cada turma representado por uma matriz [5 × 4]. Dessa

Tabela 1 – dados da instância de teste

Dado	Quantidade
Professores (total)	22
Professores (comum)	9
Professores (do eixo)	13
Disciplinas	43
Turmas	8
duração em horas(por semestre)	400

Fonte: Elaborada pelos autores (2021).

Figura 8 – Representação de horário por curso e turma.

Curso: Ciência da computação
Semestre: 1/tarde

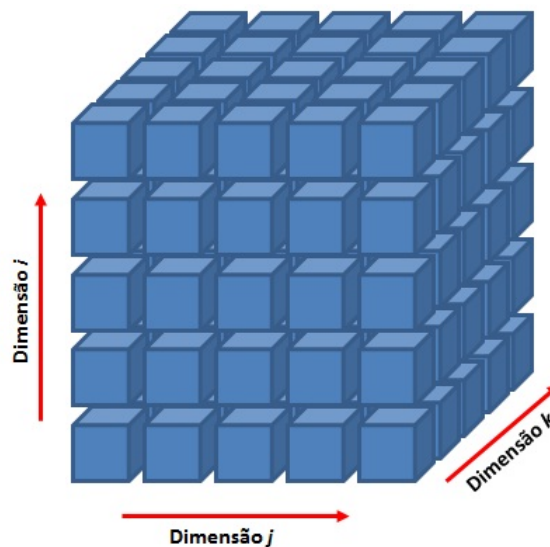
	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
A					
B					
C					
D					

Fonte: Elaborada pelos autores (2021).

forma facilita a identificação de possíveis trocas e alocações de horário, evitando as restrições duras da solução.

Com essa quantidade de professores, disciplinas e turmas, chegar em uma solução viável e que agrade à maioria dos professores pode ser trabalhoso e demorado. Mesmo assim estamos apenas na instância de teste que representa apenas uma parte dos cursos do campus.

Figura 9 – Representação dos horários em cada turno.



Fonte: Elaborada pelos autores (2021).

A figura 9 representa uma matriz de três dimensões [I × J × K]:

- J (eixo X) representa a quantidade de dias de aula;
- I (eixo Y) representa a quantidade máxima de eventos por turno;
- K (eixo Z) representa a quantidade de turmas alocadas.

Na instância de teste temos uma matriz $[5 \times 4 \times 11]$ pois a instância de teste considera apenas o eixo da informação e comunicação que possui 8 (oito) turmas no curso de Bacharelado em Ciência da Computação (BCC) e 3 (três) turmas Técnico em Informática (TI)

Relembrando que uma das restrições duras, mais óbvias, requer que nenhum professor esteja alocado para dois eventos no mesmo horário. A Figura 9 mostra que todos os pontos $[I,J,K]$ variando K e mantendo os mesmos J e I, sejam professores diferentes.

Na instância geral do problema temos uma matriz $[5 \times 4 \times 39]$ sendo que existem salas do campus que cumprem funções específicas como laboratórios de informática, química, de aquicultura e de hotelaria. Com usos específicos exclusivos de algumas disciplinas e cursos.

Ao final da execução, o programa converte os resultados em dois arquivos em *Portable Document Format* (PDF). Um com o horário individual de cada professor, como mostrado na Figura 10, e outro com os horários da cada turma, como mostrado na figura 11.

Nesta última, cada disciplina vem acompanhada do nome do respectivo docente, suprimido neste exemplo com uma tarja preta.

Para que outros horários sejam gerados, utilizaremos uma estratégia *multi-start* e alteraremos a ordem dos professores na base de dados, mantendo-os em seus respectivos dias e disciplinas.

Figura 10 – Representação dos horários de um professor.

X	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:15-08:15					
08:15-09:15					
09:25-10:25					
10:25-11:25					
13:15-14:15			Comp - (80H) - BCC	LP1 - (80H) - BCC	
14:15-15:15			Comp - (80H) - BCC	LP1 - (80H) - BCC	
15:25-16:25			Comp - (80H) - BCC	LP1 - (80H) - BCC	
16:25-17:25			Comp - (80H) - BCC	LP1 - (80H) - BCC	
18:30-19:20					ED - (80H) - BCC
19:20-20:10					ED - (80H) - BCC
20:20-21:10			TC - (80H) - BCC	ED - (80H) - BCC	TC - (80H) - BCC
21:10-22:00			TC - (80H) - BCC	ED - (80H) - BCC	TC - (80H) - BCC

Fonte: Elaborada pelos autores (2021).

Figura 11 – Representação dos horários de uma turma. Dados pessoais estão ocultos.

X	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
18:30-19:20	BD2 - (80H) - BCC - [REDACTED]	EP - (40H) - BCC - [REDACTED]	CN - (80H) - BCC - [REDACTED]	RC2 - (80H) - BCC - [REDACTED]	POO - (80H) - BCC - [REDACTED]
19:20-20:10	BD2 - (80H) - BCC - [REDACTED]	EP - (40H) - BCC - [REDACTED]	CN - (80H) - BCC - [REDACTED]	RC2 - (80H) - BCC - [REDACTED]	POO - (80H) - BCC - [REDACTED]
20:20-21:10	AA - (40H) - BCC - [REDACTED]	BD2 - (80H) - BCC - [REDACTED]	CN - (80H) - BCC - [REDACTED]	RC2 - (80H) - BCC - [REDACTED]	POO - (80H) - BCC - [REDACTED]
21:10-22:00	AA - (40H) - BCC - [REDACTED]	BD2 - (80H) - BCC - [REDACTED]	CN - (80H) - BCC - [REDACTED]	RC2 - (80H) - BCC - [REDACTED]	POO - (80H) - BCC - [REDACTED]

Fonte: Elaborada pelos autores (2021).

6 CONCLUSÕES E TRABALHOS FUTUROS

Com base em tudo que se demonstra neste artigo o trabalho foi considerado promissor pelos docentes. Foi possível atingir nossos objetivos, gerando grades de horários viáveis de acordo com as restrições dos professores quanto à sua disponibilidade de dias e preferência por disciplinas. Foi possível apresentar também nossos resultados aos professores do Eixo de Informação e Comunicação. E receber uma avaliação dos mesmos onde em sua maioria, consideraram que os horários montados são satisfatórios. Os mesmos ainda deram algumas sugestões para possíveis melhorias.

Assim, poderemos concluir que o trabalho pode trazer alguns benefícios ao eixo de Informação e Comunicação. E ao campus como um todo:

- Redução do tempo e quantidade de reuniões para geração dos horários (feito antes do início de cada semestre);
- Possibilidade dos professores avaliarem mais de um cenário para seus horários;
- Melhor aproveitamentos dos horários do professor no campus, nos dias de trabalho;
- Como as grades são geradas de forma instantânea, os professores também têm a possibilidade de verificar como ficariam seus horários se ministrassem outras disciplinas e em outros dias;

Como trabalhos futuros pretendemos:

- ampliar o uso do programa para toda a instituição.
- Tornar o sistema mais robusto, permitindo o uso em outros campus do IFCE e demais instituições de ensino.

Pretenderemos ainda adicionar mais restrições solicitadas pelos docentes como: não ter uma mesma disciplina alocadas em 4 horários consecutivos de maneira opcional, horário livre em algum dos turnos e adicionar as disciplinas optativas que podem ser ministradas no contra-turno.

Outro objetivo futuro é deixar a plataforma mais usual, de forma que os coordenadores possam usar e simular.

REFERÊNCIAS

- ABBASZADEH, M.; SAEEDVAND, S. A fast genetic algorithm for solving university scheduling problem. **IAES International Journal of Artificial Intelligence**, IAES Institute of Advanced Engineering and Science, v. 3, n. 1, p. 7–15, 2014.
- ABDELHALIM, E. A.; KHAYAT, G. A. E. A utilization-based genetic algorithm for solving the university timetabling problem (uga). **Alexandria Engineering Journal**, Elsevier, v. 55, n. 2, p. 1395–1409, 2016.
- ALGHAMDI, H. et al. A review of optimization algorithms for university timetable scheduling. **Engineering, Technology & Applied Science Research**, v. 10, n. 6, p. 6410–6417, 2020.
- ASMUNI, H. **Fuzzy methodologies for automated university timetabling solution construction and evaluation**. Tese (Doutorado) — University of Nottingham, 2008.
- BABAEI, H.; KARIMPOUR, J.; HADIDI, A. A survey of approaches for university course timetabling problem. **Computers & Industrial Engineering**, Elsevier, v. 86, p. 43–59, 2015.
- DATTA, D.; DEB, K.; FONSECA, C. M. Solving class timetabling problem of iit kanpur using multi-objective evolutionary algorithm. **KanGAL, Report**, v. 2006006, p. 1–10, 2006.
- DATTA, D.; DEB, K.; FONSECA, C. M. Multi-objective evolutionary algorithms for resource allocation problems. In: SPRINGER. **International Conference on Evolutionary Multi-Criterion Optimization**. [S.l.], 2007. p. 401–416.
- GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**. [S.l.]: wh freeman New York, 2002. v. 29.
- KAZARLIS, S.; PETRIDIS, V.; FRAGKOU, P. Solving university timetabling problems using advanced genetic algorithms. **GAs**, v. 2, n. 7, p. 8–12, 2005.
- LEITE, N. et al. Solving a capacitated exam timetabling problem instance using a bi-objective nsga-ii. In: **Computational Intelligence**. [S.l.]: Springer, 2015. p. 115–129.
- MAHIBA, A. A.; DURAI, C. A. D. Genetic algorithm with search bank strategies for university course timetabling problem. **Procedia Engineering**, Elsevier, v. 38, n. 0, p. 253–263, 2012.
- OBIT, J. H. **Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems**. Tese (Doutorado) — University of Nottingham, 2010.
- REDL, T. A. **A study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions**. [S.l.], 2004.