

Adriano Yuji Sato de Vasconcelos

**Algoritmos genéticos para a resolução de problemas de
alocação de horários de aulas na UFGD**

Brasil

2021

Adriano Yuji Sato de Vasconcelos

Algoritmos genéticos para a resolução de problemas de alocação de horários de aulas na UFGD

Trabalho de conclusão de curso, objetivando o desenvolvimento e a proposta de uma ferramenta que soluciona o problema de *timetabling* para os cursos de Engenharia de Computação e Sistemas de Informação da Universidade Federal da Grande Dourados - UFGD.

Universidade Federal da Grande Dourados - UFGD
Faculdade de Ciências Exatas e Tecnologia - FACET
Engenharia de Computação Bacharelado

Orientador: Marcos Mansano Furlan

Brasil
2021



MINISTÉRIO DA EDUCAÇÃO
FUNDAÇÃO UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

AVALIAÇÃO DE PROJETO DE TCC OU TCC

TÍTULO DO TRABALHO: Algoritmos genéticos para a resolução de problemas de alocação de horários de aulas na UFGD

NOME DO(S) ESTUDANTE(S): Adriano Yuji Sato De Vasconcelos

ORIENTADOR(A): Marcos Mansano Furlan

DATA DA APRESENTAÇÃO: 12/05/2021

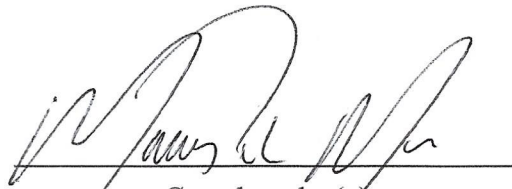
HORÁRIO: 09:00

AVALIAÇÃO DA COMISSÃO EXAMINADORA:

<input type="checkbox"/> Trabalho de Conclusão de Curso I	<input checked="" type="checkbox"/> Trabalho de Conclusão de Curso II
Presidente: Marcos Mansano Furlan Assinatura: <u><i>Marcos M. Furlan</i></u>	
Membro 1: Willian Paraguassu Amorim Assinatura: <u><i>Willian P. Amorim</i></u>	
Membro 2: Vanessa Munhoz Reina Bezerra Assinatura: <u><i>Vanessa R. Bezerra</i></u>	
Resultado:	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado condicionado a modificações (especificar no campo observações) <input type="checkbox"/> Reprovado
Observações:	
Eu, Adriano Yuji Sato De Vasconcelos, Orientador Marcos Mansano Furlan, e Coordenador do Curso Marcos Paulo Moro, declaramos para os devidos fins que estamos de acordo com a defesa do Trabalho de Conclusão de Curso, seja no formato remoto (Google Meet), e que a mesma será gravada.	

Adriano Yuji Sato De Vasconcelos

Adriano Yuji Sato De Vasconcelos



Coordenador(a)

Bacharelado em Engenharia da Computação

¹ Conforme **RESOLUÇÃO NÚMERO 106 de 29/06/2020 - Art. 5º** As defesas de Trabalho de Conclusão de Curso (TCC) deverão ocorrer por meio remoto (Google Meet, Skype, Zoom, entre outros) quando houver concordância, por escrito, entre a coordenação de curso, o docente orientador e o estudante.

Algoritmos genéticos para a resolução de problemas de alocação de horários de aulas na UFGD

Adriano Y. S. de Vasconcelos¹, Marcos M. Furlan¹

¹Faculdade de Ciências Exatas e Tecnologia (FACET)
Universidade Federal da Grande Dourados (UFGD)
Dourados – MS – Brasil.

adrianoyuji@gmail.com, marcosfurlan@ufgd.edu.br

Abstract. *Reconciling the employees agenda and the institution's attributions is a highly complex task. Methods and tools responsible to find the synchronism between resources and demand can be used to assist decision making. In this paper, it was developed a decision support tool using JavaScript and genetic algorithm to generate class schedule and teacher allocations in an optimized way. The studied problem considers two computer related course of Universidade Federal da Grande Dourados, a list of disciplines and teachers, in addition to institutional characteristics and restrictions. The developed computational tool was able to solve the problem addressed meeting the restrictions and preferences imposed, generating a satisfactory solution for the user.*

Resumo. *Conciliar a agenda dos colaboradores e as atribuições das instituições é uma tarefa de alta complexidade. Métodos e ferramentas responsáveis por encontrar o sincronismo entre recurso e demanda podem ser utilizados para auxiliar na tomada de decisão. Neste trabalho, desenvolveu-se uma ferramenta de apoio à tomada de decisão utilizando Javascript e algoritmo genético para gerar alocações de horários de aulas e de professores de forma otimizada. O problema estudado considera dois cursos da área de Computação da Universidade Federal da Grande Dourados, quadro de disciplinas e docentes, além de características e restrições institucionais. A ferramenta computacional desenvolvida foi capaz de solucionar o problema abordado, atendendo as restrições e preferências impostas, gerando uma solução satisfatória para o usuário.*

1. Introdução

Estruturar de forma manual um horário acadêmico que respeite as atribuições da instituição, satisfaça as restrições do problema e atenda às necessidades dos docentes, torna-se impraticável devido à complexidade do problema conforme ocorre o aumento das restrições dos individuais e da quantidade de docentes, disciplinas, horários e turmas.

O problema de alocação da grade de horários, também conhecido como *timetabling problem*, já foi amplamente revisado e estudado durante os anos e conta com diversas aplicações práticas estudadas na literatura, como por exemplo, o escalonamento de enfermeiros [Cheang et al. 2003], horários de aula em instituições de ensino [Burke et al. 1997] [Caldeira and Rosa 1997] e planejamento de transporte público [Lourenço et al. 2001].

O objetivo do *timetabling* é criar uma grade de horários que satisfaça dois tipos de restrições: as restrições rígidas (*hard constraints*) e as restrições flexíveis (*soft constraints*). As restrições rígidas consistem de regras que não devem ser violadas em nenhuma condição. Já o cumprimento das restrições flexíveis é opcional, ou seja, uma solução ideal obrigatoriamente deve respeitar todas as restrições rígidas, mas não necessariamente as restrições flexíveis. Entretanto, as restrições flexíveis servem como um diferencial benéfico para as soluções geradas, quanto mais restrições flexíveis forem atendidas, maior é a qualidade da solução.

Alguns exemplos comuns de uma restrição rígida seriam a impossibilidade de um professor ministrar mais de uma aula no mesmo dia e horário e a impossibilidade de uma turma possuir duas disciplinas no mesmo dia e horário. Exemplos comuns de restrições flexíveis podem ser vinculadas à preferências, como um professor específico possui determinado horário como mais apropriado para ministrar aulas, mas não possui nenhuma impossibilidade de lecionar em outro horário.

No decorrer do estudo dos trabalhos da literatura, ficou evidente que não é conveniente, em termos computacionais, o uso de programação determinística para solucionar o *timetabling* devido ao elevado tempo computacional demandado para a resolução dos problemas. Muitos estudos são realizados para desenvolver métodos para auxiliar na descoberta de uma solução de alta qualidade, visto que não existe apenas um tipo de solução para o problema [Burke et al. 1997].

Diversos problemas computacionais são solucionados inspirados na natureza, observando cada indivíduo para chegar a um resultado coletivo, isto é, como se define uma meta-heurística evolutiva ou bioinspirada. O algoritmo genético tem embasamento na teoria evolucionária de Charles Darwin, na qual os seres sofrem mutações e recombinações que os modificam com o passar das gerações. Neste contexto, o processo de seleção natural auxilia na evolução das espécies impondo limitantes de recursos e obstáculos aos quais apenas os mais aptos irão sobreviver, este é um método bio-inspirado desenvolvido por John Henry Holland em 1975.

Com o uso do algoritmo genético, este trabalho tem como objetivo desenvolver uma ferramenta que gere soluções não-determinísticas para o problema de alocação de horários da Universidade Federal da Grande Dourados (UFGD) que atenderá as restrições específicas da universidade e auxiliará a coordenação dos cursos a montarem uma grade acadêmica.

Na Seção 2 faremos uma breve apresentação dos trabalhos da literatura que já abordaram o problema do *timetabling*. A Seção 3 apresenta as características do problema em questão, enquanto a Seção 4 apresenta a estrutura e descrição de um algoritmo genético e a Seção 5 apresenta os métodos utilizados para a geração de uma solução. Posteriormente serão apresentados os testes e resultados obtidos durante o desenvolvimento da ferramenta de apoio a decisão, na Seção 6. Por fim, a Seção 7 trás as conclusões e trabalhos futuros.

2. Revisão da literatura

Nos últimos 50 anos, muito já foi discutido sobre o problema de *timetabling*, e a tendência é que o assunto continue com o mesmo índice de popularidade por mais anos que virão.

Um fator que colabora com a constante popularidade do problema são as inovações dos modelos educacionais, que estão constantemente sofrendo mudanças em sua forma de operação e seus requisitos [Cunha 2016], exigindo sempre uma refatoração das soluções existentes.

Existem diversas variações deste problema no campo educacional, como o problema de alocação de horários para escolas (*school timetabling*), alocação de horários para cursos (*course timetabling*), alocação de horários para exames (*exam timetabling problems*) [Schaerf 1999] e o problema de alocação de horários para cursos de uma universidade (*University Course Timetabling Problem*) (UCTP).

O tópico sobre a utilização de uma solução computacional para solucionar o problema de *timetabling* vem sendo estudado desde ao menos a década de 1960. Em [Appleby et al. 1961] são apresentados os primeiros métodos para geração de horários, sendo eles: a geração de uma combinação aleatória dos horários e a geração de todas as combinações de horários possíveis por meio de força bruta. Ficou evidente que os métodos anteriores não são eficientes, então foi apresentado também um método que envolve uma abordagem heurística, inserindo regras que impactam na geração das soluções, que apesar de exigirem mais tempo para apresentar um resultado, são mais próximas do desejável.

Com o passar dos anos seguintes, mais métodos para a geração de soluções foram apresentados, como em [Gotlieb 1963, Csimá and Gotlieb 1964] no qual foi apresentado um método que ao invés de tentar gerar uma solução que atenda uma semana inteira de horários, o problema foi afinado para apenas um dia. Reduzindo drasticamente a complexidade do problema e da solução.

No trabalho [Almond 1966], foi apresentada uma solução que não só atende restrições rígidas, mas também tenta satisfazer restrições consideradas flexíveis, onde caso uma delas falhe, o algoritmo efetua manobras na tentativa de continuar a procurar por uma solução.

Em [Holland 1975] o método de algoritmo genético foi formalmente introduzido, o mesmo tem como objetivo utilizar mecanismos importantes de seleção natural. O algoritmo genético possui uma característica que através de métodos internos bem definidos, dificulta o processo de se limitar a máximos locais. Em [Golberg 1989] foram apresentados novos métodos e estratégias que otimizam os processos específicos do Algoritmo Genético.

Vários trabalhos da literatura abordaram o problema de *timetabling* associado ao método de Algoritmo Genético, dentre eles podemos citar [Coloni et al. 1992] que obteve sucesso na geração de uma tabela de horários que atendesse os requisitos de sua instituição. Trabalhos como [Souza et al. 2002] utilizam de outros procedimentos como a busca Tabu para otimização do processo de geração da solução.

O trabalho [Babaei et al. 2015] analisou e comparou 66 diferentes tipos de algoritmos variando entre métodos meta-heurísticos como o Algoritmo Genético, Algoritmo Memético, métodos híbridos, métodos de lógica difusa (*fuzzy approach*), entre outros. E indicaram que o algoritmo genético está entre os métodos mais lentos na geração de uma solução factível, o que é esperado visto que o mesmo efetua cálculos custosos para refinar a solução não-determinística. Contudo, o método mostrou-se eficiente na geração de

resultados de qualidade.

3. Descrição do problema

Neste trabalho aborda-se a alocação de horários da Universidade Federal da Grande Dourados, MS. Para ser mais preciso, trabalharemos com dados dos cursos de Engenharia de Computação e Sistema de Informação. Levando em consideração que os diferentes cursos possuem diferentes características como, por exemplo, o curso de Engenharia de Computação ocorre em tempo integral enquanto o curso de Sistema de Informação ocorre no período noturno, além de possuírem diferentes tipos de disciplinas e professores que podem ser compartilhados ou exclusivos.

Para que uma solução gerada seja considerada viável, ela deve atender um grupo de restrições. Estas restrições funcionam como regras e servem para orientar o algoritmo genético a avaliar uma solução e determinar se uma solução gerada é válida ou não. Estas restrições são divididas em dois tipos: restrições rígidas e restrições flexíveis.

3.1. Restrições rígidas

As restrições rígidas são consideradas obrigatórias. Uma solução viável deve atender todas, sem exceção, caso qualquer uma não seja atendida, esta solução não é válida. As restrições rígidas estabelecidas para atender as necessidades da Universidade Federal da Grande Dourados (UFGD) foram:

1. Não permitir que um mesmo professor seja alocado em diferentes turmas no mesmo horário;
2. Não alocar professores que informaram indisponibilidade em determinados horários;
3. A carga horária semanal de um professor deve ser no mínimo 8 horas e no máximo 20 horas;
4. Disciplinas devem respeitar a carga horária imposta no Projeto Pedagógico do Curso;
5. As aulas de uma mesma disciplina devem estar agrupadas em um mesmo período;
6. Não permitir a alocação de diferentes professores para uma mesma disciplina.

3.2. Restrições flexíveis

As restrições flexíveis são restrições opcionais e a sua satisfação não é obrigatória para que uma solução seja viável, contudo, quanto mais restrições flexíveis uma solução atender, mais satisfatória ela é. As restrições estabelecidas foram:

1. Alocação de professores em horários que informaram preferência;
2. Alocação de professores em disciplinas que informaram preferência;
3. As aulas de uma mesma disciplina podem estar agrupadas em um mesmo dia;
4. A distribuição de carga horária deve ser equilibrada entre os professores.

A restrição rígida 5 e a restrição flexível 3, apesar de semelhantes priorizam atributos diferentes. A primeira obriga que as aulas de uma disciplina estejam em um mesmo período, por exemplo, todas as aulas devem estar alocadas no período da manhã, caso uma aula se encontre no período da tarde, e outra se encontre no período da manhã, a solução não pode ser considerada factível. Já a restrição flexível indica que, caso as aulas de uma mesma disciplina estejam agrupadas em um mesmo dia, a qualidade da solução é maior. Por exemplo, todas as aulas de uma mesma disciplina estão alocadas na segunda-feira, isto indica que a solução é mais favorável.

4. Algoritmo Genético

Algoritmo genético é um método de busca baseado nos princípios da seleção natural [Fraser 1957, Bremermann 1958, Holland 1975]. O algoritmo genético, diferente dos métodos tradicionais determinísticos de otimização, utilizam meios probabilísticos para a convergência do resultado, trabalham com parâmetros codificados e avaliam cada indivíduo isoladamente, possuindo um paralelismo implícito. O algoritmo genético é altamente dependente da codificação do indivíduo para o determinado problema, sendo necessário um estudo particular de cada caso.

Um algoritmo genético tradicional é composto dos seguintes passos: Inicialização, avaliação (*fitness*), seleção, cruzamento (*crossover*) e mutação. Todas etapas são repetidas por um número de n gerações. A cada ciclo de geração, mais próximo do ideal a solução se torna, logo, quanto maior for o número de gerações, mais refinada e factível a solução gerada é.

4.1. Estrutura de um algoritmo genético

O Algoritmo 1 apresenta o pseudocódigo do algoritmo genético. No qual a função principal recebe como argumento de entrada:

- n que representa a quantidade máxima de gerações que serão efetuadas;
- i que representa a quantidade de indivíduos da população;
- P_c que refere à probabilidade de cruzamento que dois indivíduos selecionados possa sofrer;
- P_m que indica a probabilidade de um novo individuo sofrer mutação após o cruzamento.

Algoritmo 1: Estrutura Algoritmo Genético

Entrada: (n, i, P_c, P_m)

início

$t \leftarrow 0$;

 população[t] \leftarrow inicializaPopulação(i) (Seção 5.2);

 Avaliação(população);

repita

$t \leftarrow t + 1$;

$q \leftarrow 0$;

repita

$q \leftarrow q + 2$;

 pais \leftarrow Seleção(população[$t-1$]) (Seção 5.3.1);

 filhos \leftarrow Cruzamento(pais, P_c) (Seção 5.3.2);

 Mutaçao(filhos, P_m) (Seção 5.3.3);

 Avaliação(filhos)(Seção 5.4);

 população[t] \leftarrow filhos;

até $q \geq i$;

até $t = n$;

fim

E possui variáveis onde:

- t representa a quantidade atual de gerações executadas;
- q representa a quantidade de indivíduos geradas na geração atual t , neste caso, dois novos indivíduos são gerados a cada cruzamento.

5. Desenvolvimento

5.1. Representação de uma solução

Para este problema, um indivíduo é representado por y quadros de horários semanais, onde y representa o número de turmas em um curso. Este quadro de horário é representado por uma matriz S_{ytdh} , onde t representa o período do dia (matutino, vespertino e noturno), e d representa um dia da semana e h um horário, c_{ps} é um valor de uma célula da matriz, onde p é um professor e s é uma disciplina.

O valor de uma célula da matriz pode ser representada por dois estados, um estado vazio, onde não foi atribuído nenhum professor e nenhuma disciplina, e o estado em que professor e disciplina foram atribuídos à célula.

5.2. Geração da população inicial

Na literatura, o método mais popular encontrado para inicialização da população é a geração aleatória dos indivíduos [Von Zuben et al. 2000]. Entretanto com um conhecimento prévio do problema é possível usufruir de heurísticas para eliminar a possibilidade da existência de alguns grupos de indivíduos inválidos que potencialmente retardariam a evolução da população durante a etapa de cruzamento.

Para nosso problema a geração da população ocorre de forma híbrida, utiliza tanto de operadores aleatórios quanto operadores heurísticos:

- O operador aleatório é utilizado para sortear um período, um dia e um horário para alocar uma disciplina, os professores também são selecionados e alocados aleatoriamente, porém um operador heurístico prioriza professores que demonstraram interesse em lecionar dada disciplina.
- Há também fatores determinísticos que influenciam na geração, todas as disciplinas selecionadas serão alocadas em suas respectivas turmas e também terão sua carga horária corretamente distribuída durante a criação do indivíduo. Garantindo que as restrições Rígidas 4 e 5 da Seção 3 sempre serão atendidas. Acelerando a criação de indivíduos mais próximos da região de factibilidade.

5.3. Operadores Genéticos

5.3.1. Operadores de Seleção

Após uma população inicial definida, é necessário um método que será responsável por selecionar estes indivíduos para reproduzir e gerar novos descendentes. A ideia inicial é escolher os indivíduos com um *fitness* elevado que garantirão descendentes que estejam mais próximo da solução desejada.

Entretanto, caso aplique-se uma seleção extremamente elitista, selecionaremos apenas os melhores indivíduos da população, reduzindo drasticamente a diversidade que é fundamental para geração de futuras variações da população [Costa and DALLA BRUNA 2002]. Por outro lado, também não podemos aplicar uma

seleção muito abrangente, pois poderemos selecionar os indivíduos com um *fitness* muito baixo, causando retardo na evolução da população ou, até mesmo, convergência por deriva.

A literatura apresenta diversos métodos de seleção, um muito popular é o Método da Roleta, apresentado por [Golberg 1989], que consiste em criar uma roleta onde cada indivíduo da população recebe um segmento proporcional a seu *fitness*, ou seja quanto melhor for sua avaliação, maior será sua chance de ser selecionado para o cruzamento.

Para o problema abordado, utilizaremos outro método muito popular conhecido como Seleção por Torneio. Ele é capaz de garantir que a seleção dos indivíduos possua um alto nível de variabilidade. Este método seleciona aleatoriamente k membros da população (sendo k um valor pré determinado), dos quais o indivíduo com o *fitness* mais alto é selecionado para o cruzamento. Para cada processo de cruzamento dois torneios são feitos para que dois indivíduos sejam selecionados.

A Figura 1 apresenta um diagrama de como a Seleção por Torneio funciona para $k = 3$. Nesta seleção os indivíduos x, y, z são selecionados aleatoriamente da população, e o cromossomo melhor avaliado através da função de *fitness* é selecionado pelo método.

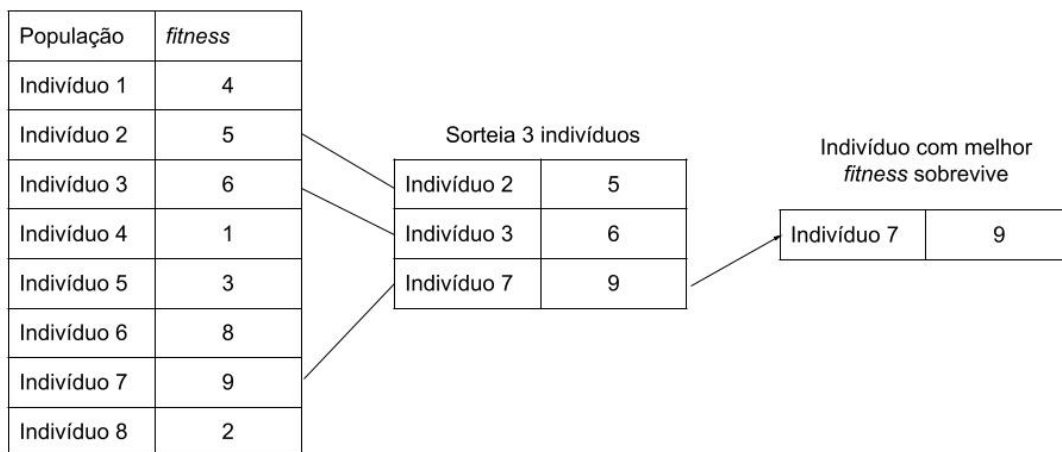


Figura 1. Ilustração do diagrama do operador de seleção por torneio ($k=3$)

5.3.2. Operadores de Cruzamento

O cruzamento é um operador utilizado nos indivíduos escolhidos pelo processo de seleção, ele promove a troca de informações entre os mesmos para que novos indivíduos que carregam essa mistura de genes sejam gerados. Este operador é afetado pela taxa de cruzamento P_c .

Os métodos mais comumente utilizados foram introduzidos por [Holland 1975], sendo eles: o Cruzamento de um ponto, o Cruzamento de dois pontos e o Cruzamento Uniforme. Estes métodos consistem em definir um ou mais pontos de corte nos genes dos indivíduos selecionados e efetuar a recombinação das informações.

Devido à natureza do nosso problema, o uso de qualquer um dos métodos citados anteriormente causa inconsistência nas restrições rígidas, o que basicamente inviabilizou o uso de qualquer uma delas. Felizmente, [Goldberg et al. 1985] sugeriram o uso de um outro método conhecido como *Partially-Mapped Crossover (PMX)* para problemas cuja ordem ou quantidade dos genes influenciam diretamente no *fitness* do indivíduo.

O funcionamento do *PMX* pode ser descrito da seguinte forma. Primeiro, são selecionados dois pontos aleatoriamente ao longo dos genes dos pais. Suponhamos que os pontos de cortes são selecionados entre o primeiro e o segundo dia e o segundo ponto entre o terceiro e o quarto dia do quadro de horários semanal como mostrado na Figura 2, onde os pontos de corte são representados pelas linhas mais espessas.

QUADRO 1	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 1	Horário 5	Horário 9	Horário 13	Horário 17
08:10h	Horário 2	Horário 6	Horário 10	Horário 14	Horário 18
09:15h	Horário 3	Horário 7	Horário 11	Horário 15	Horário 19
10:05h	Horário 4	Horário 8	Horário 12	Horário 16	Horário 20

QUADRO 2	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 20	Horário 16	Horário 12	Horário 8	Horário 4
08:10h	Horário 19	Horário 15	Horário 11	Horário 7	Horário 3
09:15h	Horário 18	Horário 14	Horário 10	Horário 6	Horário 2
10:05h	Horário 17	Horário 13	Horário 9	Horário 5	Horário 1

Figura 2. Imagem ilustrativa dos pontos de corte selecionados pelo *PMX*.

Os campos entre os cortes dos pontos são chamados de seção de mapeamento. No exemplo citado, o primeiro mapeamento será $5 \leftrightarrow 16$, pois durante o cruzamento, a célula horário 5 do Quadro 1 corresponde à célula horário 16 do Quadro 2, criando esta relação. Continuando, são gerados os relacionamentos: $6 \leftrightarrow 15$, $7 \leftrightarrow 14$, $8 \leftrightarrow 13$, $9 \leftrightarrow 12$, $10 \leftrightarrow 11$, $11 \leftrightarrow 10$ e $12 \leftrightarrow 9$. Em seguida o método copia o conteúdo entre os pontos de corte do primeiro quadro para o segundo quadro e o conteúdo do segundo quadro é transferido para o primeiro.

Após efetuarmos o cruzamento, percebemos que existem horários repetidos nas tabelas, como por exemplo no Quadro 1 da Figura 3, existem duas ocorrências do horário 13 até o horário 16 e também não há ocorrência dos horário 5 até o horário 8, o mesmo tipo de problema acontece com o Quadro 2 da Figura 3. Este tipo de acontecimento desprezeta a restrição rígida que obriga uma solução a garantir a carga horária imposta no Projeto Pedagógico do Curso.

Logo, o *PMX* efetua a legalização das colunas que se encontram fora da área do corte. Para isto, utilizaremos o mapeamento de dados que foi efetuado no primeiro passo.

QUADRO 1	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 1	<u>Horário 16</u>	Horário 12	<u>Horário 13</u>	Horário 17
08:10h	Horário 2	<u>Horário 15</u>	Horário 11	<u>Horário 14</u>	Horário 18
09:15h	Horário 3	<u>Horário 14</u>	Horário 10	<u>Horário 15</u>	Horário 19
10:05h	Horário 4	<u>Horário 13</u>	Horário 9	<u>Horário 16</u>	Horário 20



QUADRO 2	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 20	<u>Horário 5</u>	Horário 9	<u>Horário 8</u>	Horário 4
08:10h	Horário 19	<u>Horário 6</u>	Horário 10	<u>Horário 7</u>	Horário 3
09:15h	Horário 18	<u>Horário 7</u>	Horário 11	<u>Horário 6</u>	Horário 2
10:05h	Horário 17	<u>Horário 8</u>	Horário 12	<u>Horário 5</u>	Horário 1

Figura 3. Imagem ilustrativa do cruzamento de genes pelo *PMX*, as células sublinhadas indicam sua duplicação.

Então para ambos os quadros, sabemos que $5 \leftrightarrow 16$, então buscaremos a ocorrência do horário 5 e trocaremos para o horário 16, e vice versa. Sabemos que $6 \leftrightarrow 15$ então efetuaremos a substituição dos valores nas tabelas. Após efetuar a legalização para todos os genes da parte externa do corte, finalmente teremos os novos indivíduos que carregam os dados de ambos os pais e que serão inseridos na população, como visto na Figura 4.

QUADRO 1	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 1	Horário 16	Horário 12	Horário 8	Horário 17
08:10h	Horário 2	Horário 15	Horário 11	Horário 7	Horário 18
09:15h	Horário 3	Horário 14	Horário 10	Horário 6	Horário 19
10:05h	Horário 4	Horário 13	Horário 9	Horário 5	Horário 20

QUADRO 2	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Horário 20	Horário 5	Horário 9	Horário 13	Horário 4
08:10h	Horário 19	Horário 6	Horário 10	Horário 14	Horário 3
09:15h	Horário 18	Horário 7	Horário 11	Horário 15	Horário 2
10:05h	Horário 17	Horário 8	Horário 12	Horário 16	Horário 1

Figura 4. Imagem ilustrativa dos novos indivíduos legalizados pelo *PMX*.

5.3.3. Operadores de Mutação

Os operadores de mutação são fundamentais para adicionarem e manterem a diversidade genética da população. Graças a mutação, é possível explorar um espaço muito maior de soluções. Em momentos que o processo sofre uma estagnação, caindo em um máximo local por exemplo, este método permite que novas características inexistentes anteriormente surjam. Este operador é afetado pela taxa de mutação P_m .

Como a natureza deste método envolve aleatoriedade, não é possível garantir que a alteração seja benéfica. Caso o resultado seja positivo, ele permanecerá e será herdado nas próximas gerações, caso seja negativo, provavelmente será eliminado através da seleção ou será corrigido no cruzamento.

Dois tipos de operadores de mutação foram usados. O primeiro envolve exclusivamente fatores aleatórios, enquanto o segundo envolve fatores mais heurísticos para gerar resultados benéficos para o indivíduo.

O primeiro fator de mutação, seleciona aleatoriamente duas células do quadro de horários do indivíduo e troca o conteúdo de ambas. Como citamos anteriormente, este operador é aleatório, então não podemos garantir que o resultado desta troca de posições será benéfica. Podemos observar o comportamento desta mutação na Figura 5.



Figura 5. Diagrama da mutação de troca aleatória de células.

O segundo fator de mutação implementado, possui um comportamento que irá beneficiar o indivíduo caso ele seja cumprido. Este comportamento afeta positivamente a restrição rígida que obriga aulas de uma mesma disciplina serem agrupadas em um mesmo período. Este fator procura no quadro de horários se existe em um mesmo dia e período o agrupamento de pelo menos 75% da carga horária de uma disciplina. Caso encontre, o processo irá procurar na matriz onde está localizada a célula que contém o

restante da carga horária da disciplina e irá agrupá-la com os 75% que foram inicialmente identificados. A Figura 6 apresenta uma ilustração de como este método funciona.

QUADRO 1	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Disciplina 1		Disciplina 2		Disciplina 2
8:10h	Disciplina 1	Disciplina 3			Disciplina 3
9:15h	Disciplina 2			Disciplina 1	Disciplina 2
10:05h	Disciplina 1	Disciplina 3		Disciplina 3	

QUADRO 1	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:20h	Disciplina 1		Disciplina 2		Disciplina 2
8:10h	Disciplina 1	Disciplina 3			Disciplina 3
9:15h	Disciplina 1			Disciplina 2	Disciplina 2
10:05h	Disciplina 1	Disciplina 3		Disciplina 3	

Figura 6. Diagrama da mutação de agrupamento de disciplinas.

5.4. Função *fitness*

A função de *fitness* calcula para todos os indivíduos de uma população o quão factível eles são e qual será a capacidade relativa de cada um para sobreviver durante a fase de seleção. Essa função é específica para cada problema e geralmente não segue nenhum padrão de cálculo.

Todos os indivíduos começam com uma pontuação igual a 0, e conforme a avaliação das restrições, o indivíduo é punido ou recompensado. A punição é baseada no nível da restrição. Caso seja uma restrição rígida, a pontuação deduzida ou recompensada é alta, para restrições flexíveis, como a satisfação não é obrigatória, não há penalidade e a recompensa é baixa. A Tabela 1 contém uma lista de todas as restrições rígidas citadas na Seção 3 e a Tabela 2 contém as restrições flexíveis, ambas tabelas possuem suas devidas penalidades e recompensas. As numerações seguem as mesmas definidas anteriormente na seção supracitada.

6. Testes e Resultados

Os docentes de ambos os cursos de Engenharia de Computação e Sistemas de Informação preencheram um formulário virtual demonstrando interesse ou não em ministrar as disciplinas presentes nos projetos pedagógicos e declarando indisponibilidade ou preferência em determinados horários. O uso destes dados serviu para que os resultados gerados representassem com maior precisão as situações reais que ocorrem durante a criação das tabelas de horários. Procurando atender as necessidades determinadas pela coordenação dos cursos, mas também levando em consideração as preferências e indisponibilidades dos professores.

Tabela 1. Punições e recompensas das restrições rígidas.

Restrição	Penalidade	Recompensa
1	-10	+10
2	-50	0
3	-10	0
4	-10	0
5	-5	+5
6	-50	0

Tabela 2. Punições e recompensas das restrições flexíveis.

Restrição	Penalidade	Recompensa
1	0	+5
2	0	+5
3	0	+5
4	0	+5

Foram utilizadas 3 diferentes instâncias para execução dos testes:

1. Inclui somente o curso de Engenharia de Computação, possui 40 disciplinas, 20 docentes, 5 turmas por semestre e as aulas são ministradas no período integral.
2. Inclui somente o curso de Sistemas de Informação, possui 30 disciplinas, 20 docentes, 4 turmas por semestre e as aulas são ministradas no período noturno.
3. A terceira instância é uma combinação da primeira e da segunda instância, como os cursos possuem disciplinas em comum, foram somados um total de 50 disciplinas, e a grande maioria dos professores também são compartilhados entre ambos os cursos, totalizando 22 professores.

Para uma melhor otimização de tempo e da qualidade dos resultados, foram realizados testes com diferentes valores para as variáveis do algoritmo genético, utilizando valores acima e abaixo dos valores determinados inicialmente. O ajuste dos valores otimizados para os parâmetros foi executado de forma iterativa e considerando a seguinte ordem: quantidade de gerações, tamanho da população, taxa de cruzamento e taxa de mutação. Foram feitas 5 execuções do método para cada conjunto de parâmetros e cada instância, utilizando as mesmas entradas de dados, para uma melhor precisão dos resultados apresentados. As Figuras 7, 8, 9 e 10 apresentam os resultados dos testes, nas quais cada ponto no gráfico representa *fitness*, no eixo vertical, e tempo computacional, no eixo horizontal, de uma execução completa do algoritmo genético proposto.

Os testes foram realizados em um Intel Core i5 1.60GHz *Quad Core* e 12GB de memória RAM no navegador Google Chrome. A ferramenta foi desenvolvida com o uso da linguagem de programação JavaScript¹, utilizando a biblioteca ReactJS² para o desenvolvimento da interface web do usuário. Para o desenvolvimento da API REST, foi utilizado JavaScript e NodeJS³, utilizando o banco de dados não-relacional (NoSQL)

¹ Acessível em: <https://www.javascript.com/>

² Acessível em: <https://pt-br.reactjs.org/>

³ Acessível em: <https://nodejs.org/en/>

MongoDB⁴ para armazenamento das informações cadastradas pelo usuário.

Observando a Figura 7, foi efetuado a alteração no valor da variável de quantidade de gerações (n), utilizando os valores 100, 250 e 400. Podemos notar na instância 1 que não houve um avanço significativo na qualidade dos resultados quando comparamos $n = 250$ com $n = 400$, indicando que o algoritmo genético não encontra melhorias para as soluções a partir da 250ª geração. O mesmo acontece para a instância 2, indicando que o resultado atinge seu limite antes da 100ª geração. Para a instância 3, notamos uma melhoria nos resultados para $n = 400$ comparado ao valor $n = 250$, mas não são resultados significativos quando levamos em consideração que a execução leva praticamente o dobro do tempo.

Teste valor n (quantidade de gerações)

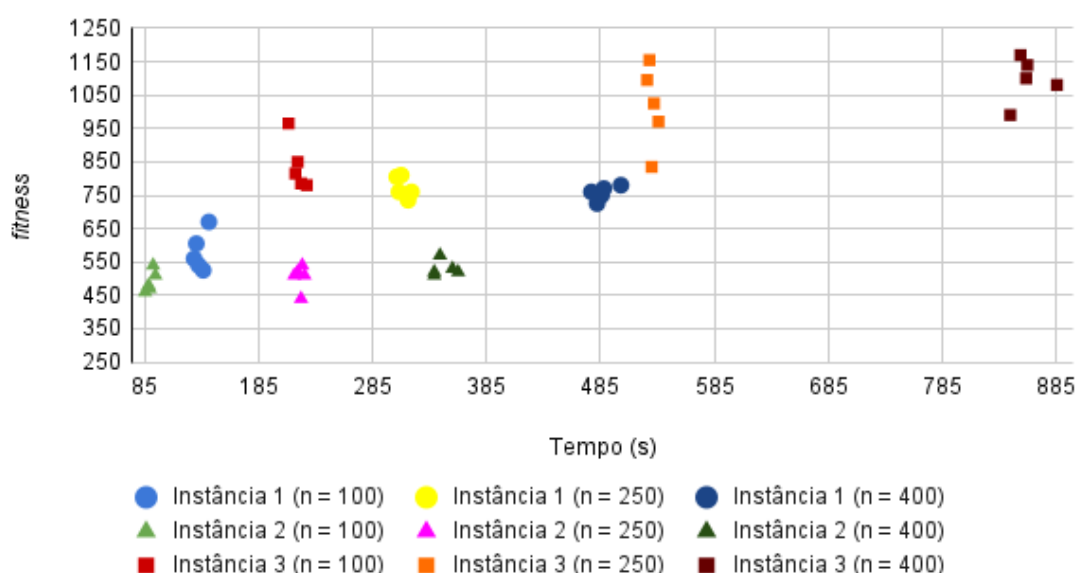


Figura 7. Gráfico dos resultados dos testes da da variável n .

Analisando a Figura 8, o valor da variável de população (i) foi alterado, utilizando os valores 50, 100 e 200. Os resultados obtidos foram muito similares com os resultados obtidos no teste anterior. Para a instância 1 e 3, não houve uma melhoria significativa nos resultados de $i = 100$ para $i = 200$ que justifique o dobro do tempo de execução. Para a instância 2, o resultado atingiu o limite no *fitness* mesmo variando os valores de i devido a sua menor quantidade de dados. Desta forma, podemos perceber que instâncias maiores demandam mais indivíduos na população para chegar a resultados de alta qualidade. Isso é esperado, pois com mais disciplinas, docentes e horários para a alocação, a quantidade de soluções aumenta. Dessa forma, mais indivíduos ajudam no processo de mapeamento do espaço de soluções. No entanto, isso gera um aumento significativo no tempo computacional necessário para a resolução da alocação.

Na Figura 9, a variável alterada foi a taxa de cruzamento (P_c), utilizando os valores 65%, 85% e 95%. Ao analisarmos os resultados, percebemos que para as instâncias 1

⁴Acessível em: <https://www.mongodb.com/>

Teste valor i (população)

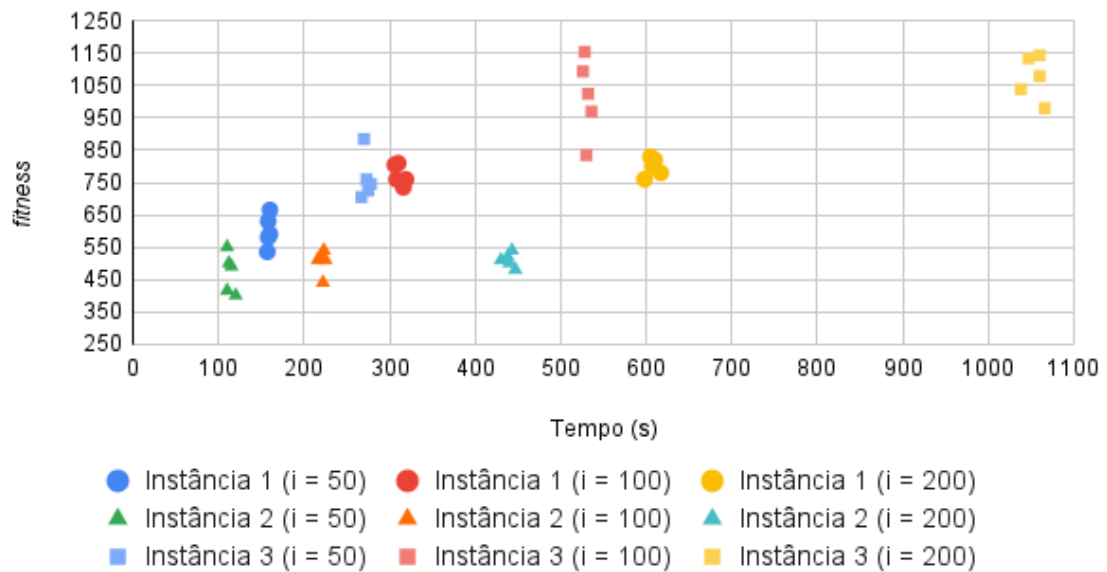


Figura 8. Gráfico dos resultados dos testes da variável i .

e 2, aumentar ou diminuir a chance de cruzamento não impacta na qualidade das soluções, gerando aumento somente no tempo de execução. Mas para a instância 3, podemos notar que ao diminuir a chance de cruzamento para 65%, houve uma redução considerável no *fitness* dos resultados das execuções. Isso indica que o operador de cruzamento aumenta sua importância para instâncias maiores. Além disso, verificamos que 85% aparenta ser um valor adequado, pois gera uma taxa vantajosa de mistura das soluções, com tempo computacional reduzido em relação a taxa de 95%.

Por fim, na Figura 10, a variável alterada foi a taxa de mutação (P_m), utilizando os valores 0%, 10%, 20% e 40%. Pode-se notar que os resultados gerados para $P_m = 0\%$ são significativamente inferiores aos resultados gerados para $P_m > 0\%$, indicando que a mutação possui um papel importante na evolução da qualidade da população durante a execução do algoritmo genético. Os resultados gerados com as diferentes taxas de mutação $P_m > 0\%$, não apresentaram distinção significativa para as instâncias testadas, indicando que a mutação não precisa ocorrer com alta frequência durante os processos do algoritmo genético.

Através dos resultados obtidos nos testes de otimização, os parâmetros citados na Seção 4.1 foram definidos para a execução dos testes: a quantidade máxima de gerações foi definida em 250 ($n = 250$), a quantidade de indivíduos da população foi limitada à 100 ($i = 100$), a taxa de cruzamento (P_c) foi definida em 85% e a taxa de mutação (P_m) foi definida em 20%.

Tendo as variáveis do algoritmo genético definidas, os resultados dos testes serão apresentados através de gráficos que estão presentes nas Figuras 11, 12 e 13, onde cada ponto no gráfico representa o melhor indivíduo da geração. Fazendo uma análise prévia dos resultados, algumas execuções se mostraram muito similares, causando a

Teste valor P_c (Taxa de cruzamento)

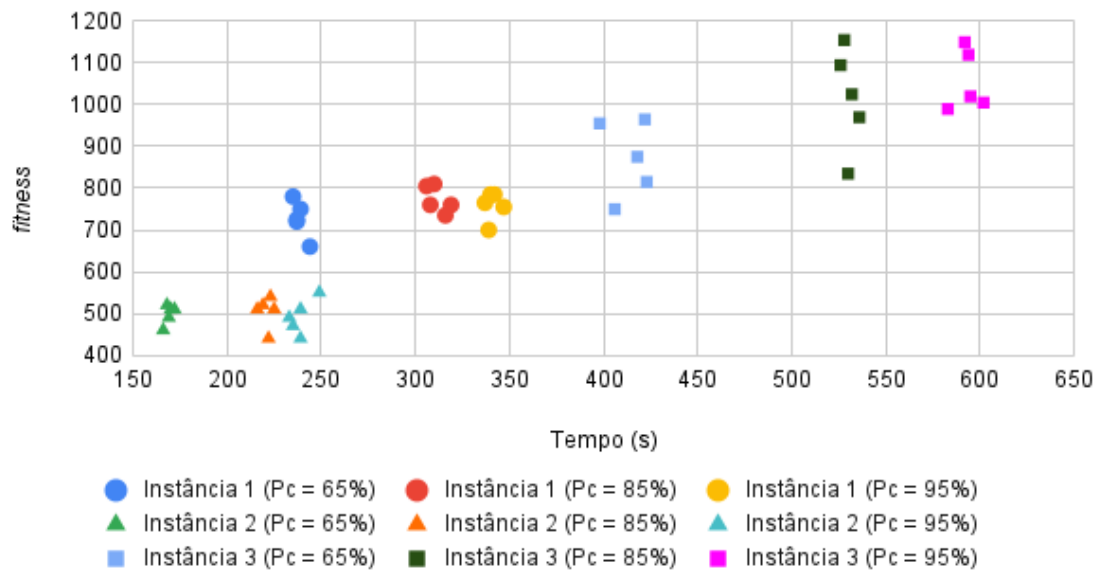


Figura 9. Gráfico dos resultados dos testes da variável P_c .

Teste valor P_m (Taxa de mutação)

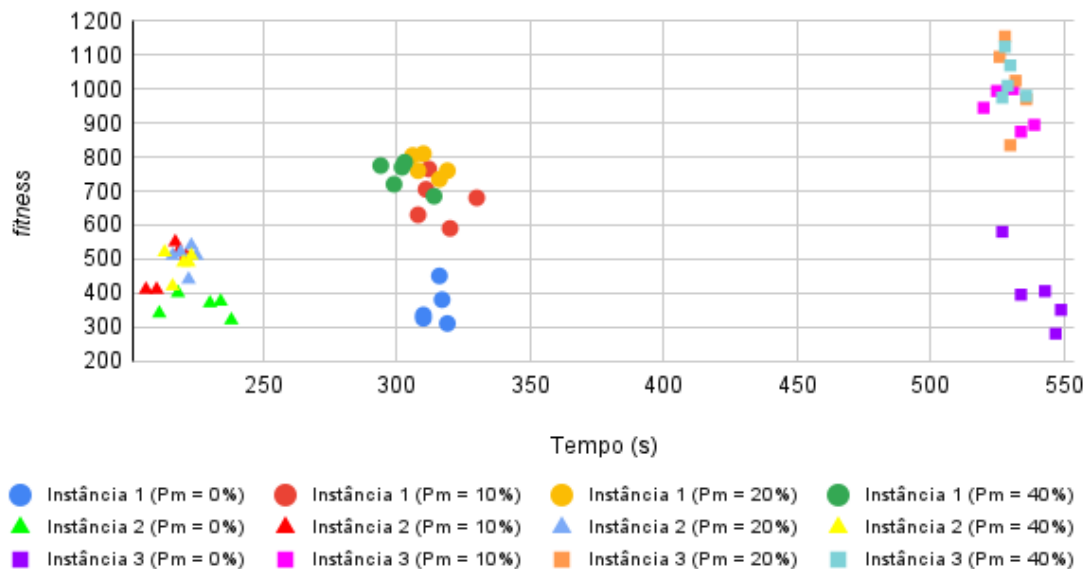


Figura 10. Gráfico dos resultados dos testes da variável P_m .

sobreposição de parte das linhas, entretanto isto tornou evidente a robustez do método para os parâmetros definidos.

Podemos visualizar a evolução das populações geradas durante os testes da primeira instância observando o gráfico da Figura 11. O processo de execução do algoritmo demorou em média 6 minutos. Notamos que, devido a maior quantidade de disciplinas e

horários, houve uma evolução gradual dos indivíduos até a 170^a geração, onde a curva de evolução começou a desacelerar.

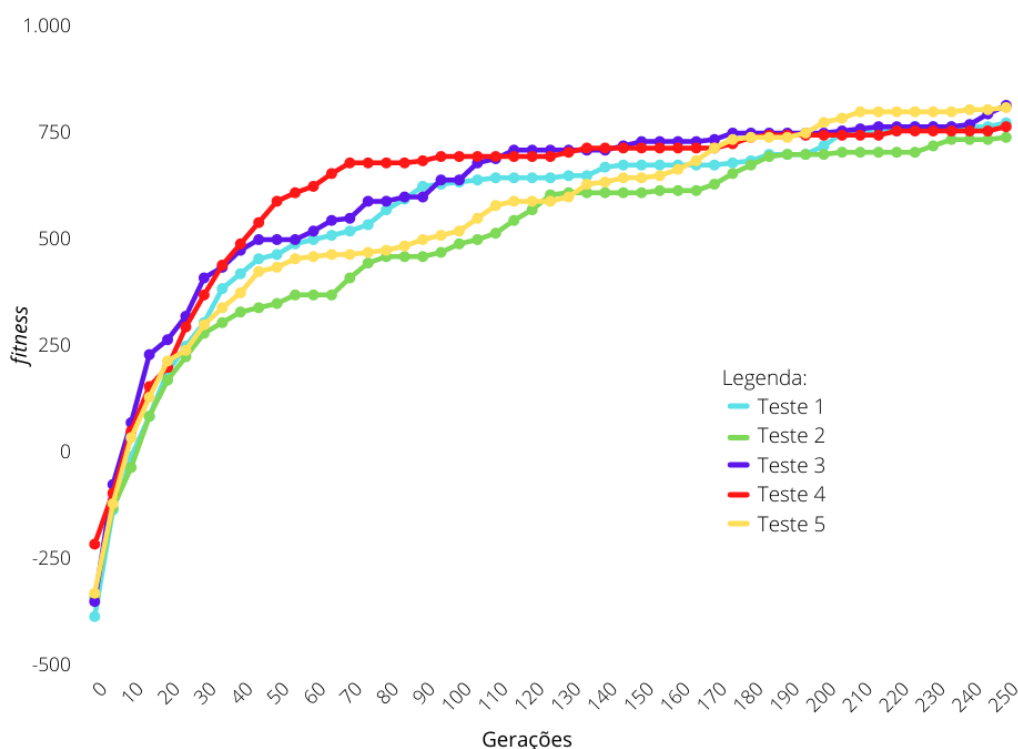


Figura 11. Gráfico da evolução da população da primeira instância.

Observando os dados de geração da segunda instância no gráfico da Figura 12. O processo de execução do algoritmo demorou em média 4 minutos, visto que possui apenas um período de aula, houve uma redução na quantidade de iterações do algoritmo. É possível notar uma estagnação na evolução do melhor indivíduo que ocorrem na maior parte dos casos de testes na faixa da 85^a geração, o que indica que para cursos com apenas um período, o valor da variável n pode ser definido em valores inferiores ao usado nos testes.

A terceira instância de testes levou em média 9 minutos para ser executada, visto que efetua o cálculo de 2 cursos ao mesmo tempo. Podemos notar no gráfico da Figura 13, que a geração inicial da população possui uma pontuação *fitness* inicial muito baixa (altamente penalizada), mas com o passar das gerações é possível notar que a evolução do melhor indivíduo é constante, e que ele continuaria a evoluir se não fosse limitada pelo número de gerações.

Os resultados obtidos para as instâncias 1 e 2 podem ser considerados factíveis, não atendendo apenas a restrição rígida 3, visto que a quantidade de disciplinas ofertadas por apenas um curso não é suficiente para atender a demanda de carga horária mínima de todos os professores. Por outro lado, para ($n = 250$), os resultados da terceira instância de testes demonstraram potencial para continuar evoluindo, uma vez que diversas restrições ainda estavam pendentes. Podemos concluir com estes resultados que, quanto maior for

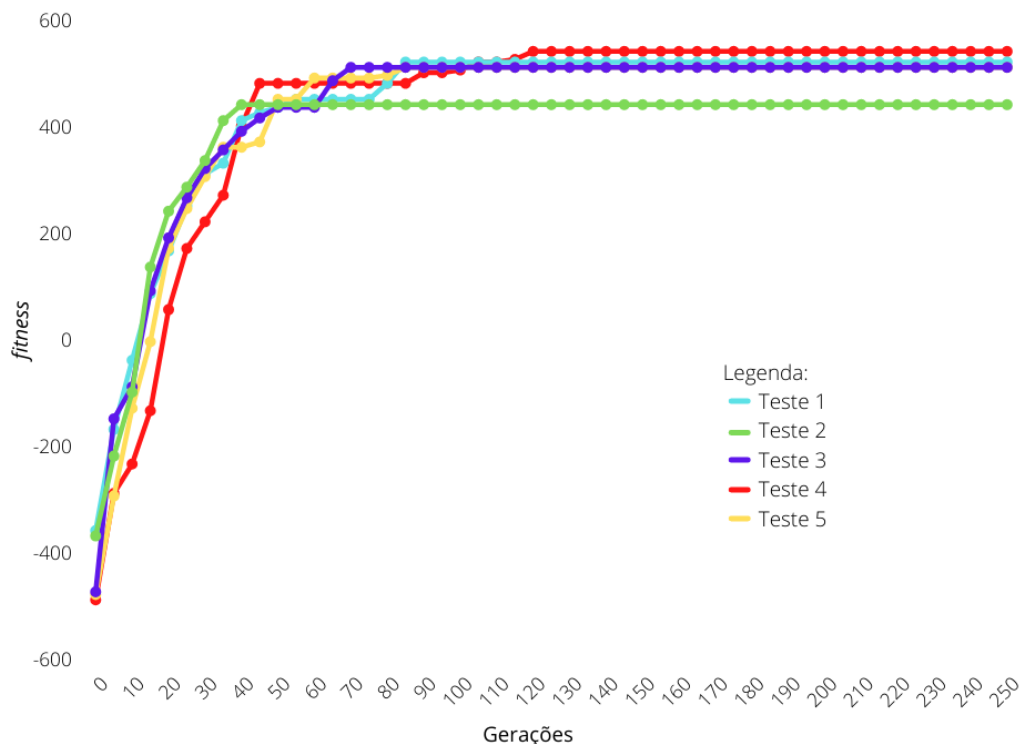


Figura 12. Gráfico da evolução da população da segunda instância.

sua quantidade de cursos, disciplinas, e períodos, maior deve ser seu número de gerações a ser executado para a obtenção de uma solução factível.

Um exemplo de resultado de cada instância foi anexado neste documento, sendo que: o Anexo A representa uma solução da instância 1, o Anexo B representa uma solução da instância 2, e o Anexo C, uma solução da instância 3.

7. Conclusões e trabalhos futuros

Apesar de existirem diversos estudos e ferramentas que tentam solucionar o problema de *UCTP*, é de extrema dificuldade dos usuários encontrarem uma que satisfaça todas as suas exigências. Por esse motivo, foi desenvolvido uma ferramenta de apoio a decisão que irá auxiliar a coordenação dos cursos que ainda passam por um processo completamente manual na geração das grades de horários.

Esta ferramenta foi desenvolvida com JavaScript para ser executada em navegadores. Além de efetuar a geração das grades dos horários através do algoritmo genético implementado, ela possui uma interface que permite ao usuário cadastrar cursos, professores e disciplinas que são armazenados em um banco de dados através de um servidor RESTful remoto. Antes de efetuar a geração, a aplicação permite ao usuário definir as disciplinas cujo horário foi informado por outra coordenação. A aplicação foi dividida em duas partes, o *frontend*⁵ e o *backend*⁶, o código fonte de ambas as partes foram hos-

⁵ Acessível em: <https://github.com/adrianoyuji/ufgd-timetabling>

⁶ Acessível em: <https://github.com/adrianoyuji/timetabling-api>

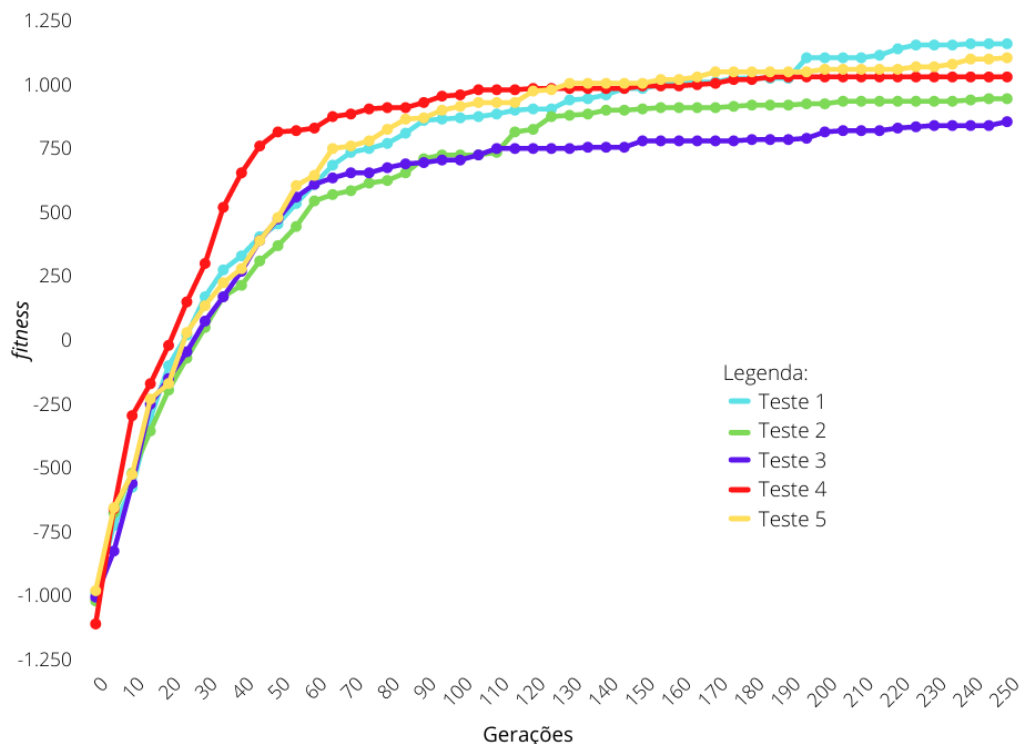


Figura 13. Gráfico da evolução da população da terceira instância.

pedados na plataforma de controle de versão github⁷.

Ao analisar os resultados obtidos, é possível avaliá-los como satisfatórios, mas é da natureza de algoritmos não-determinísticos gerarem resultados que possam não estar 100% de acordo com os requisitos implantados. A decisão final sempre será do coordenador do curso, do corpo docente e do conselho diretor da unidade acadêmica, e os resultados gerados servem apenas como ferramenta de apoio nesta decisão.

Como trabalhos futuros, podemos incluir a implementação de novas restrições que são específicas dos cursos da UFGD, a implantação de disciplinas que atendem múltiplos cursos de forma simultânea, o desenvolvimento de uma aplicação para *smartphones* que possa coletar as informações dos docentes a cada semestre, com ajuste de preferência e disponibilidade de horário. Permitir que a coordenação possa ajustar a solução em tempo de execução (sempre que o algoritmo estagnar), fixando disciplinas em determinados horários ou com professores específicos. Estes trabalhos refinariam ainda mais a solução gerada, além de prover mais conveniência na hora da coleta das informações de cada professor e da aplicação das soluções.

Referências

Almond, M. (1966). An algorithm for constructing university timetables. *The Computer Journal*, 8(4):331–340.

⁷Acessível em: <https://github.com/>

- Appleby, J., Blake, D., and Newman, E. (1961). Techniques for producing school timetables on a computer and their application to other scheduling problems. *The Computer Journal*, 3(4):237–245.
- Babaei, H., Karimpour, J., and Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59.
- Bremermann, H. J. (1958). *The evolution of intelligence: The nervous system as a model of its environment*. University of Washington, Department of Mathematics.
- Burke, E., Jackson, K., Kingston, J. H., and Weare, R. (1997). Automated university timetabling: The state of the art. *The computer journal*, 40(9):565–571.
- Caldeira, J. and Rosa, A. C. (1997). School timetabling using genetic search. *Practice and Theory of Automated Timetabling, Toronto*.
- Cheang, B., Li, H., Lim, A., and Rodrigues, B. (2003). Nurse rostering problems—a bibliographic survey. *European journal of operational research*, 151(3):447–460.
- Coloni, A., Dorigo, M., and Maniezzo, V. (1992). A genetic algorithm to solve the timetable problem. *Politecnico di Milano, Milan, Italy TR*, pages 90–060.
- Costa, E. O. and DALLA BRUNA, M. (2002). Resolução de timetabling utilizando evolução cooperativa. *Universidade Federal do Paraná*.
- Csima, J. and Gotlieb, C. (1964). Tests on a computer method for constructing school timetables. *Communications of the ACM*, 7(3):160–163.
- Cunha, M. I. (2016). Inovações na educação superior: impactos na prática pedagógica e nos saberes da docência. *Em Aberto*, 29(97).
- Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers ii. effects of linkage on rates of advance under selection. *Australian Journal of Biological Sciences*, 10(4):492–500.
- Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989(102):36.
- Goldberg, D. E., Lingle, R., et al. (1985). Alleles, loci, and the traveling salesman problem. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 154, pages 154–159. Lawrence Erlbaum Hillsdale, NJ.
- Gotlieb, C. (1963). The construction of class-teacher timetables. In *IFIP congress*, volume 62, pages 73–77.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*.
- Lourenço, H. R., Paixao, J. P., and Portugal, R. (2001). *The crew-scheduling module in the gist system*. Universitat Pompeu Fabra Portugal.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, 13(2):87–127.
- Souza, M. J. F., Costa, F., and Guimarães, I. (2002). Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas. *Computer*.

Von Zuben, F. J. et al. (2000). Contribuições a solução de problemas de escalonamento pela aplicação conjunta de computação evolutiva e otimização com restrições.

Anexo A

Primeiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	Livre
08:10	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	DESENHO POR COMPUTADOR Felipe José Carbone	Livre
09:15	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	DESENHO POR COMPUTADOR Felipe José Carbone	Livre
10:05	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	Livre
13:20	Livre	Livre	ELETRÔNICA DIGITAL Murilo Táparo	LABORATÓRIO DE DESENHO POR COMPUTADOR Felipe José Carbone	Livre
14:10	Livre	Livre	Livre	Livre	Livre
15:15	LABORATÓRIO DE ELETRÔNICA DIGITAL Carlos Elias Arminio Zampieri	Livre	Livre	LABORATÓRIO DE DESENHO POR COMPUTADOR Felipe José Carbone	Livre
16:05	LABORATÓRIO DE ELETRÔNICA DIGITAL Carlos Elias Arminio Zampieri	Livre	ELETRÔNICA DIGITAL Murilo Táparo	Livre	Livre

Segundo Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Anderson Bessa da Costa	Livre	Livre
08:10	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Anderson Bessa da Costa	Livre	Livre
09:15	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Anderson Bessa da Costa	Livre	Livre
10:05	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Anderson Bessa da Costa	Livre	Livre
13:20	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO II Anderson Bessa da Costa	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre
14:10	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO II Anderson Bessa da Costa	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre
15:15	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO II Anderson Bessa da Costa	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre
16:05	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO II Anderson Bessa da Costa	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre

Terceiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	BANCO DE DADOS I Silvana Morita Melo	Livre	ANÁLISE DE ALGORITMOS Joinville Batista Junior
08:10	Livre	Livre	Livre	Livre	ANÁLISE DE ALGORITMOS Joinville Batista Junior
09:15	Livre	Livre	Livre	Livre	ANÁLISE DE ALGORITMOS Joinville Batista Junior
10:05	Livre	Livre	Livre	Livre	ANÁLISE DE ALGORITMOS Joinville Batista Junior
13:20	Livre	Livre	LABORATÓRIO DE BANCO DE DADOS I Silvana Morita Melo	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan
14:10	Livre	Livre	Livre	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan
15:15	Livre	Livre	LABORATÓRIO DE BANCO DE DADOS I Silvana Morita Melo	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan
16:05	Livre	BANCO DE DADOS I Silvana Morita Melo	Livre	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan

Quarto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	SISTEMAS OPERACIONAIS I Felipe José Carbone	DESENVOLVIMENTO DE APLICAÇÕES WEB Felipe José Carbone	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	Livre	Livre
08:10	SISTEMAS OPERACIONAIS I Felipe José Carbone	DESENVOLVIMENTO DE APLICAÇÕES WEB Felipe José Carbone	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	Livre	Livre
09:15	SISTEMAS OPERACIONAIS I Felipe José Carbone	DESENVOLVIMENTO DE APLICAÇÕES WEB Felipe José Carbone	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	Livre	Livre
10:05	SISTEMAS OPERACIONAIS I Felipe José Carbone	DESENVOLVIMENTO DE APLICAÇÕES WEB Felipe José Carbone	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	Livre	Livre
13:20	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfírio da Silva Sacchi	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre
14:10	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfírio da Silva Sacchi	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre
15:15	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfírio da Silva Sacchi	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre
16:05	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfírio da Silva Sacchi	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre

Quinto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre
08:10	Livre	Livre	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre
09:15	Livre	Livre	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre
10:05	Livre	Livre	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre
13:20	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre
14:10	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre
15:15	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre
16:05	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre

Anexo B

Primeiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO I Wellington Lima dos Santos	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Carla Adriana Brarvinski Zanchet
19:50	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO I Wellington Lima dos Santos	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Carla Adriana Brarvinski Zanchet
20:50	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO I Wellington Lima dos Santos	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Carla Adriana Brarvinski Zanchet
21:40	Livre	Livre	LABORATÓRIO DE PROGRAMAÇÃO I Wellington Lima dos Santos	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Carla Adriana Brarvinski Zanchet

Segundo Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES	LINGUAGEM DE PROGRAMAÇÃO I	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO	Livre	ALGORITMOS E ESTRUTURAS DE DADOS III
20:50	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Rodrigo Porfirio da Silva Sacchi	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	Livre	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos
21:40	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Rodrigo Porfirio da Silva Sacchi	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	Livre	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos

Terceiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE BANCO DE DADOS II Everton Castelão Tetila	ENGENHARIA DE SOFTWARE I Evanise Araujo Caldas Ruiz	INTRODUÇÃO À COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	REDES DE COMPUTADORES Felipe José Carbone
19:50	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	ENGENHARIA DE SOFTWARE I Evanise Araujo Caldas Ruiz	INTRODUÇÃO À COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	REDES DE COMPUTADORES Felipe José Carbone
19:50	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	ENGENHARIA DE SOFTWARE I Evanise Araujo Caldas Ruiz	INTRODUÇÃO À COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	REDES DE COMPUTADORES Felipe José Carbone
20:50	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE BANCO DE DADOS II Everton Castelão Tetila	ENGENHARIA DE SOFTWARE I Evanise Araujo Caldas Ruiz	BANCO DE DADOS II Everton Castelão Tetila	REDES DE COMPUTADORES Felipe José Carbone
21:40	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	ENGENHARIA DE SOFTWARE I Evanise Araujo Caldas Ruiz	BANCO DE DADOS II Everton Castelão Tetila	REDES DE COMPUTADORES Felipe José Carbone

Quarto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	Livre
19:50	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	Livre
20:50	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	Livre
21:40	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	Livre

Anexo C

Primeiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	Livre
08:10	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	Livre	Livre	LABORATÓRIO DE ELETRÔNICA DIGITAL Carlos Elias Arminio Zampieri
09:15	Livre	Livre	Livre	Livre	Livre
10:05	Livre	Livre	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	Livre
13:20	LABORATÓRIO DE DESENHO POR COMPUTADOR Felipe José Carbone	Livre	ELETRÔNICA DIGITAL Murilo Táparo	DESENHO POR COMPUTADOR Felipe José Carbone	Livre
14:10	Livre	Livre	LABORATÓRIO DE ELETRÔNICA DIGITAL Carlos Elias Arminio Zampieri	Livre	Livre
15:15	LABORATÓRIO DE DESENHO POR COMPUTADOR Felipe José Carbone	TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO Rosenilda Marques da Silva Felipe	Livre	DESENHO POR COMPUTADOR Felipe José Carbone	Livre
16:05	Livre	Livre	ELETRÔNICA DIGITAL Murilo Táparo	Livre	Livre

Segundo Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	LABORATÓRIO DE PROGRAMAÇÃO II Rodrigo Porfirio da Silva Sacchi	Livre	Livre	Livre	Livre
08:10	Livre	Livre	Livre	Livre	Livre
09:15	LABORATÓRIO DE PROGRAMAÇÃO II Rodrigo Porfirio da Silva Sacchi	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Rodrigo Porfirio da Silva Sacchi	Livre
10:05	Livre	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre	ALGORITMOS E ESTRUTURAS DE DADOS II Rodrigo Porfirio da Silva Sacchi	Livre
13:20	LABORATÓRIO DE PROGRAMAÇÃO II Rodrigo Porfirio da Silva Sacchi	Livre	Livre	Livre	Livre
14:10	LABORATÓRIO DE PROGRAMAÇÃO II Rodrigo Porfirio da Silva Sacchi	Livre	Livre	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	ALGORITMOS E ESTRUTURAS DE DADOS II Rodrigo Porfirio da Silva Sacchi
15:15	Livre	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	ALGORITMOS E ESTRUTURAS DE DADOS II Rodrigo Porfirio da Silva Sacchi	Livre	Livre
16:05	Livre	FUNDAMENTOS DE TEORIA DA COMPUTAÇÃO Rodrigo Yoshikawa Oeiras	Livre	Livre	Livre

Terceiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	ANÁLISE DE ALGORITMOS Carlos Elias Arminio Zampieri	ANÁLISE DE ALGORITMOS Carlos Elias Arminio Zampieri	LABORATÓRIO DE BANCO DE DADOS I Silvana Morita Melo	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan	Livre
08:10	Livre	Livre	Livre	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan	Livre
09:15	Livre	Livre	Livre	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri
10:05	Livre	Livre	Livre	LINGUAGENS FORMAIS E AUTÔMATOS Marcos Mansano Furlan	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri
13:20	ANÁLISE DE ALGORITMOS Carlos Elias Arminio Zampieri	BANCO DE DADOS I Silvana Morita Melo	LABORATÓRIO DE BANCO DE DADOS I Silvana Morita Melo	BANCO DE DADOS I Silvana Morita Melo	Livre
14:10	Livre	Livre	Livre	Livre	Livre
15:15	Livre	Livre	Livre	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri	TÓPICOS AVANÇADOS EM COMPUTAÇÃO I Carlos Elias Arminio Zampieri
16:05	ANÁLISE DE ALGORITMOS Carlos Elias Arminio Zampieri	Livre	Livre	Livre	Livre

Quarto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	Livre	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre
08:10	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	Livre	Livre	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	Livre
09:15	Livre	Livre	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	Livre	Livre
10:05	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	TECNOLOGIA E COMUNICAÇÃO DE DADOS Marcos Paulo Moro	Livre	Livre	Livre
13:20	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfirio da Silva Sacchi	DESENVOLVIMENTO DE APLICAÇÕES WEB Joinville Batista Junior	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	SISTEMAS OPERACIONAIS I Felipe José Carbone
14:10	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfirio da Silva Sacchi	DESENVOLVIMENTO DE APLICAÇÕES WEB Joinville Batista Junior	Livre	SISTEMAS OPERACIONAIS I Felipe José Carbone
15:15	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfirio da Silva Sacchi	DESENVOLVIMENTO DE APLICAÇÕES WEB Joinville Batista Junior	Livre	SISTEMAS OPERACIONAIS I Felipe José Carbone
16:05	CIRCUITOS ELETRÔNICOS Rodrigo Yoshikawa Oeiras	TÓPICOS AVANÇADOS EM COMPUTAÇÃO III Rodrigo Porfirio da Silva Sacchi	DESENVOLVIMENTO DE APLICAÇÕES WEB Joinville Batista Junior	MICROCONTROLADORES E APLICAÇÕES Marcos Mansano Furlan	SISTEMAS OPERACIONAIS I Felipe José Carbone

Quinto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
07:20	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre	Livre	Livre
08:10	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre
09:15	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre	Livre	Livre
10:05	Livre	VERIFICAÇÃO, VALIDAÇÃO E TESTES DE SOFTWARE Silvana Morita Melo	Livre	Livre	Livre
13:20	Livre	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre
14:10	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre	Livre
15:15	Livre	Livre	COMPUTAÇÃO E SOCIEDADE Silvana Morita Melo	Livre	Livre
16:05	INTERAÇÃO HUMANO-COMPUTADOR Evanise Araujo Caldas Ruiz	Livre	Livre	Livre	Livre

Primeiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	Livre	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Rosenilda Marques da Silva Felipe	LABORATÓRIO DE PROGRAMAÇÃO I Murilo Táparo
19:50	Livre	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Rosenilda Marques da Silva Felipe	LABORATÓRIO DE PROGRAMAÇÃO I Murilo Táparo
20:50	Livre	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Rosenilda Marques da Silva Felipe	LABORATÓRIO DE PROGRAMAÇÃO I Murilo Táparo
21:40	Livre	Livre	Livre	ALGORITMOS E ESTRUTURAS DE DADOS I Rosenilda Marques da Silva Felipe	LABORATÓRIO DE PROGRAMAÇÃO I Murilo Táparo

Segundo Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Marcos Mansano Furlan	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	Livre
19:50	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Marcos Mansano Furlan	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	Livre
20:50	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Marcos Mansano Furlan	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	Livre
21:40	ALGORITMOS E ESTRUTURAS DE DADOS III Wellington Lima dos Santos	FUNDAMENTOS DE SISTEMAS DE INFORMAÇÃO Felipe José Carbone	ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES Marcos Mansano Furlan	LINGUAGEM DE PROGRAMAÇÃO I Anderson Bessa da Costa	Livre

Terceiro Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	BANCO DE DADOS II Everton Castelhão Tetila	REDES DE COMPUTADORES Felipe José Carbone	INTRODUÇÃO À COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	ENGENHARIA DE SOFTWARE I Felipe José Carbone
19:50	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	REDES DE COMPUTADORES Felipe José Carbone	INTRODUÇÃO À COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	ENGENHARIA DE SOFTWARE I Felipe José Carbone
20:50	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	LABORATÓRIO DE COMPUTAÇÃO GRÁFICA Adailton José Alves da Cruz	REDES DE COMPUTADORES Felipe José Carbone	LABORATÓRIO DE BANCO DE DADOS II Everton Castelhão Tetila	ENGENHARIA DE SOFTWARE I Felipe José Carbone
21:40	LINGUAGEM DE PROGRAMAÇÃO III Joinville Batista Junior	BANCO DE DADOS II Everton Castelhão Tetila	REDES DE COMPUTADORES Felipe José Carbone	LABORATÓRIO DE BANCO DE DADOS II Everton Castelhão Tetila	ENGENHARIA DE SOFTWARE I Felipe José Carbone

Quarto Ano

Horário	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
19:00	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	Livre	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe
19:50	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	Livre	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe
20:50	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	Livre	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe
21:40	GERÊNCIA DE PROJETOS Felipe José Carbone	GERÊNCIA DE REDES DE COMPUTADORES Carlos Elias Arminio Zampieri	Livre	DESENVOLVIMENTO DE PROJETO DE SISTEMAS COMPUTACIONAIS II Felipe José Carbone	METODOLOGIA CIENTÍFICA EM CIÊNCIA DA COMPUTAÇÃO Rosenilda Marques da Silva Felipe