

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS JATAÍ
BACHARELADO EM ENGENHARIA ELÉTRICA

KARLA NATHANE DOS SANTOS SOUZA

ELABORAÇÃO DE GRADE HORÁRIA DO CURSO TÉCNICO EM
ELETROTÉCNICA INTEGRADO AO ENSINO MÉDIO DO INSTITUTO FEDERAL
DE GOIÁS – CAMPUS JATAÍ UTILIZANDO ALGORITMO GENÉTICO

Jataí - GO

2022



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Karla Nathane dos Santos Souza

Matrícula: 20151020040121

Título do Trabalho: Elaboração de Grade Horária do Curso Técnico em Eletrotécnica Integrado ao Ensino Médio do Instituto Federal de Goiás – Câmpus Jataí Utilizando Algoritmo Genético

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. () Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
3. () Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- () O documento está sujeito a registro de patente.
() O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
() Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais incluídos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Jataí 15/08/22
Local Data

Karla Nathane S. Souza

Assinatura do Autor e/ou Detentor dos Direitos Autorais

KARLA NATHANE DOS SANTOS SOUZA

**ELABORAÇÃO DE GRADE HORÁRIA DO CURSO TÉCNICO EM
ELETROTÉCNICA INTEGRADO AO ENSINO MÉDIO DO INSTITUTO FEDERAL
DE GOIÁS – CAMPUS JATAÍ UTILIZANDO ALGORITMO GENÉTICO**

Trabalho de Conclusão de curso apresentado ao
Instituto Federal de Educação, Ciência e Tecnologia de
Goiás, Câmpus Jataí para obtenção do grau de Bacharel
em Engenharia Elétrica.

Orientadora: Profa. Dra. Jaqueline Oliveira Rezende.

Jataí - GO

2022

Dados Internacionais de Catalogação na Publicação na (CIP)

Souza, Karla Nathane dos Santos.

Elaboração de grade horária do curso técnico em Eletrotécnica integrado ao ensino médio do Instituto Federal de Goiás – Câmpus Jataí utilizando algoritmo genético [manuscrito] / Karla Nathane dos Santos Souza. -- Jataí: IFG, Coordenação do Curso de Bacharelado em Engenharia Elétrica, 2022.

47 f.; il.

Orientadora: Profa. Dra. Jaqueline Oliveira Rezende.

Bibliografias.

1. Algoritmo genético. 2. Computação evolutiva 3. Otimização de processos. I. Rezende, Jaqueline Oliveira. II. IFG, Câmpus Jataí. III. Título.

Ficha catalográfica elaborada pela Seção Téc.: Aquisição e Tratamento da Informação.
Bibliotecária - Rosy Cristina O. Barbosa - CRB 1/2380 – C. Jataí. Cód. F045/2022-1.



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS JATAÍ

Folha de Aprovação

KARLA NATHANE DOS SANTOS SOUZA

ELABORAÇÃO DE GRADE HORÁRIA DO CURSO TÉCNICO EM ELETROTÉCNICA INTEGRADO AO ENSINO MÉDIO DO INSTITUTO FEDERAL DE GOIÁS - CAMPUS JATAÍ UTILIZANDO ALGORITMO GENÉTICO

Trabalho de Conclusão de Curso apresentado a Coordenação do Curso de Bacharelado em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Câmpus Jataí, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia Elétrica.

Monografia defendida e aprovada, em 14 de julho de 2022, pela banca examinadora constituída pelos seguintes membros:

BANCA EXAMINADORA

Profa. Dra. Jaqueline Oliveira Rezende
Presidente da banca / Orientadora
Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Jataí

Prof. Dr. André Luiz Silva Pereira
Membro
Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Jataí

Prof. Dr. Marcelo Semensato
Membro
Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Jataí

Documento assinado eletronicamente por:

- **Andre Luiz Silva Pereira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 11/08/2022 14:17:15.
- **Marcelo Semensato**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 09/08/2022 13:51:03.
- **Jaqueline Oliveira Rezende**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 09/08/2022 11:12:03.

Este documento foi emitido pelo SUAP em 05/07/2022. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifg.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 300281

Código de Autenticação: d12fa23848



Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Av. Presidente Juscelino Kubitschek, nº 775, Residencial Flamboyant, JATAÍ / GO, CEP 75804-714
(64) 3605-0843 (ramal: 843)

*O que sabemos é uma gota, o que
ignoramos é um oceano.*

(Sir Isaac Newton)

RESUMO

O presente trabalho tem como objetivo principal o desenvolvimento de um código utilizando algoritmo genético para a geração da grade horária otimizada do Técnico em Eletrotécnica Integrado ao Ensino Médio do Instituto Federal de Goiás - Campus Jataí e que obedecem a restrições previamente impostas. O uso de algoritmo genético em problemas de otimização tem-se mostrado muito eficiente devido ao seu vasto campo de busca retornando sempre um ótimo resultado.

Baseado na biologia evolutiva, o método do algoritmo genético possui operadores genéticos que contribuem para a criação das populações, assim como na biologia o código passa por etapas como avaliação dos indivíduos, cruzamento, seleção natural e mutação genética, podendo assim obter uma maior variedade do campo de busca. O algoritmo genético consiste em um conjunto de técnicas já estabelecidas que são adequadas de acordo com o problema a ser resolvido.

A implementação deste projeto visa apresentar o modelo de resolução de problemas de otimização com a utilização de algoritmo genético. Foi apresentado a estrutura do algoritmo genético e demonstrado a construção do código, bem como os resultados obtidos, comprovando a efetividade do método descrito.

Palavras-chave: Algoritmo Genético. Computação Evolutiva. Otimização de Processos.

ABSTRACT

The present work has as main objective the development of a code used genetically for the generation of the timetable of the Technician in Electrotechnics Integrated to the High School of the Federal Institute of Goiás - Acampamento Jataí and that obey the previously implemented one. The use of genetic animals in optimization problems has proved to be very efficient due to its vast search field, always returning an excellent result.

Based on evolutionary biology, crossbreeding, natural selection and transformation, which can contribute to the creation of genetic populations, as well as evolutionary biology, crossbreeding, natural selection and transformation, which can thus obtain genetics. search field. Genetic risk is already a set according to the problem that is defined accordingly.

The implementation of this aims to present the model for solving an optimization project using genetic problems. A unique and proven structuring of the code construction was presented, as well as the presentation of the results of the described method.

Keywords: Genetic Algorithm. Evolutionary Computing. Process Optimization.

LISTA DE FIGURAS

Figura 1 – Estrutura do algoritmo genético	14
Figura 2 – Modelo de Roleta para seleção dos pais	16
Figura 3 – Cruzamento em um ponto de corte	17
Figura 4 – Cruzamento em dois pontos de corte	17
Figura 5 – Mutação aleatória	18
Figura 6 – Mutação por troca	18
Figura 7 – Indivíduo modelo	25
Figura 8 – Geração dos filhos para o primeiro ponto de corte	27
Figura 9 – Geração dos filhos para o segundo ponto de corte	27
Figura 10 – Mutação dos filhos nos cromossomos 2 e 5	28
Figura 11 – Evolução dos indivíduos ao longo de 350 gerações	29

LISTA DE TABELAS

Tabela 1 - Grade horária do curso de Bacharelado em Engenharia Elétrica em 2022/1	21
Tabela 2 – Representação numérica dos professores do curso de Técnico em Eletrotécnica ..	22
Tabela 3 – Horários preenchidos dos professores no curso de Engenharia Elétrica	23
Tabela 4 – Matriz de restrições dos professores.....	23
Tabela 5 - Vetores de restrição	24
Tabela 6 - Dias trabalhados por cada professor.....	31
Tabela 7 – Melhor indivíduo	32
Tabela 8 – Grade horária final.....	33

SUMÁRIO

1	INTRODUÇÃO	10
1.2	OBJETIVOS.....	11
1.3	JUSTIFICATIVA	11
1.4	ESTRUTURA DO TRABALHO	12
2	ALGORITMO GENÉTICO	13
2.1	Computação evolutiva	13
2.2	Evolução nos algoritmos evolutivos.....	13
2.3	Origem do algoritmo genético	13
2.4	Estrutura do algoritmo genético	14
2.4.1	Inicializar a população.....	15
2.4.2	Avaliação da aptidão	15
2.4.3	Seleção.....	15
2.4.4	Cruzamento.....	16
2.4.5	Mutação	17
2.4.6	Elitismo.....	18
2.4.7	Atualização da população	18
2.4.8	Critério de parada	19
3	DESENVOLVIMENTO	20
4	RESULTADOS.....	29
5	CONCLUSÕES	34
	REFERÊNCIAS BIBLIOGRÁFICAS	35
	APÊNDICE A – Código desenvolvido para geração da grade horária	36

1 INTRODUÇÃO

Computação Evolutiva é uma área de pesquisa da Inteligência Artificial, que tem como base conceitos da seleção natural para resolução de problemas de otimização. A computação evolutiva atua por meio de métodos e técnicas inspirados na evolução natural, tendo operadores genéticos para reproduzir os fenômenos de adaptação e seleção natural das espécies em códigos computacionais (Mitchell, 1997).

As técnicas difundidas pela computação evolutiva de acordo com Banzhaf (1998), são: Programação Evolutiva, Estratégias Evolutivas, Algoritmo Genético e Programação Genética. Na computação evolutiva um conjunto de passos é definido como base para seus métodos, que consiste na geração de uma população inicial que será modificada dentro de um ciclo evolutivo. Essa modificação acontece para que a cada nova geração se obtenha indivíduos melhores, através dos operadores genéticos. Além disso, conceitos da seleção natural são aplicados e os melhores indivíduos vão sendo aprimorados até que se tenha um resultado considerado muito bom.

Os métodos da computação evolutiva são amplamente utilizados para resolução de problemas de otimização, uma vez que sua base sempre será a mesma, apenas suas variáveis que serão adaptadas para o problema que se deseja solucionar. No método de algoritmo genético, de acordo com Holland (1975), tem-se uma junção de conceitos da Biologia Evolutiva com a Ciência da Computação, onde uma população inicial é gerada e submetida a operadores genéticos simulando os processos de seleção natural, cruzamento de indivíduos e mutação. A partir desses operadores as gerações vão sendo modificadas e a cada vez tem-se gerações melhoradas geneticamente e quando atingem um critério definido de parada retornam o valor que deve ser considerado como a melhor solução.

As grades horárias consistem na organização de arranjos entre professores e turmas, obedecendo restrições previamente estabelecidas. Quando se busca encontrar uma grade em que não tenha choque de horários para os professores e que eles sejam alocados para a menor quantidade de dias trabalhados por semana um problema de otimização pode ser gerado. Definindo quais métricas serão utilizadas para que se encontre uma boa solução para o problema proposto, uma função objetivo pode ser traçada e aplicando os conceitos de computação evolutiva, o código para resolução deste problema pode ser proposto.

1.2 OBJETIVOS

Objetivo Geral

Implementar um código de geração de grade horária automatizada para o curso Técnico em Eletrotécnica Integrado ao Ensino Médio do Instituto Federal de Goiás - Campus Jataí. Para isso, será aplicado o algoritmo genético para otimização da quantidade de dias da semana que cada professor deve atuar no mencionado curso.

Objetivos Específicos

Os objetivos específicos que contribuem para realização do objetivo geral são os seguintes:

- Estudo da utilização de algoritmo genético para otimização de processos;
- Desenvolvimento de um código com algoritmo genético para otimizar a geração da grade horária;
- Utilização de métodos de correção através da aleatoriedade para o caso das restrições não atendidas;
- Utilização dos operadores genéticos para um resultado otimizado;
- Apresentação e análise dos resultados obtidos.

1.3 JUSTIFICATIVA

Notou-se que atualmente a confecção da grade horária é feita completamente de forma manual e que muito tempo é gasto na confecção dela, principalmente para o curso Técnico em Eletrotécnica. Esse é um curso anual, mas possui elaboração semestral de sua grade horária porque ela é diretamente ligada à grade do curso de Bacharelado em Engenharia Elétrica, semestral, uma vez que muitos professores atuam em ambos os cursos. Assim, muitas restrições devem ser levadas em consideração para que se obtenha um horário viável.

A utilização de algoritmos genéticos tem se mostrado uma alternativa muito útil para problemas de otimização, visto que o algoritmo possui uma capacidade de processamento muito rápida e eficaz. Através da utilização de técnicas e operadores genéticos o algoritmo genético consegue obter um campo de busca de soluções muito grande e diversificado.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos. O presente capítulo trata de considerações gerais sobre o trabalho em nível introdutório, apresentando as considerações iniciais, a justificativa e os objetivos gerais e específicos.

No capítulo 2 são apresentadas informações gerais sobre o algoritmo genético, como surgiram, seu funcionamento e a estrutura básica do mesmo.

No capítulo 3 tem-se o desenvolvimento do código e a aplicação de operadores genéticos para convergência mais rápida e precisa.

No capítulo 4 é feita a apresentação dos resultados e do gráfico mostrando os melhores resultados por gerações.

O capítulo 5 finaliza o trabalho trazendo as considerações finais.

2 ALGORITMO GENÉTICO

2.1 Computação evolutiva

Surgida nos anos 1950 a computação evolutiva tem sido uma forma de solucionar problemas de otimização através de conceitos da evolução natural. Através de sua implementação problemas complexos de otimização podem ser solucionados com base em fenômenos da natureza (Michalewicz & Fogel, 2000).

Na computação evolutiva teve-se o desenvolvimento dos seguintes algoritmos: algoritmos genéticos, programação evolutiva, estratégias evolutivas, sistemas classificados e programação genética (Fogel, 1962).

A computação evolutiva utiliza de técnicas genéricas que podem ser adaptadas de acordo com o problema que se deseja solucionar, sendo utilizados como modelos computacionais adaptados para a aplicação utilizando dos operadores computacionais para se obter um campo de busca bastante amplo e diverso, apresentando uma solução mais eficiente (Mitchell, 1997).

2.2 Evolução nos algoritmos evolutivos

Nos algoritmos evolutivos, a evolução é feita utilizando os seguintes métodos: reprodução com herança genética, utilização de variáveis aleatórias e seleção natural para formação das novas gerações. Todos os algoritmos evolutivos possuem etapas base em comum, as quais são: reprodução, aplicação de variáveis aleatórias, competição e seleção, esses operadores genéticos serão implementados para melhor se adequar ao tipo de problema que se deseja resolver.

2.3 Origem do algoritmo genético

O pesquisador americano John Henry Holland foi quem começou a desenvolver as primeiras pesquisas em códigos computacionais com sistemas genéticos e foi então considerado o criador do algoritmo genético e o responsável pelo seu nome. Em 1975, ele publicou seu livro, intitulado “*Adaptation in Natural and Artificial Systems*” sendo a base inicial da implementação do algoritmo genético.

2.4 Estrutura do algoritmo genético

O funcionamento do algoritmo genético consiste em representar as possibilidades de solução do problema proposto através de indivíduos pertencentes a uma população. Os indivíduos gerados serão tratados através das técnicas de melhoramento genético para que uma nova população seja criada e então submetida a etapa de seleção natural. Logo o ciclo se repete até que uma condição de parada seja alcançada e o resultado mais satisfatório seja encontrado. Na Figura 1 pode-se observar o fluxograma da estrutura básica do algoritmo genético.

Figura 1 – Estrutura do algoritmo genético



Fonte: Adaptado de (Lobo, 2005)

2.4.1 Inicializar a população

A população é formada pelos indivíduos, os quais são cromossomos que contém a codificação que representa uma possível solução do problema proposto. Para inicializar a população os indivíduos são gerados de forma aleatória para garantir que a população seja bastante diversificada e o espaço de busca seja mais amplo. Caso alguma informação inicial sobre o problema seja conhecida, pode-se utilizar destes conhecimentos para inicializar a população. A população inicial pode ser gerada de diversas formas desde que respeite a aleatoriedade dos indivíduos.

2.4.2 Avaliação da aptidão

Determina quanto o indivíduo satisfaz o problema apresentado. De acordo com Linden (2006) através da função objetivo, tem-se a indicação da qualidade dos indivíduos analisados. Por isso, é fundamental que se tenha uma função objetivo adequada ao problema proposto, pois ela que definirá a eficiência do algoritmo identificando quais são as melhores soluções. Se a função não for adequada, o algoritmo poderá demorar um tempo maior para convergir para a solução e até mesmo não encontrar a solução mais eficiente.

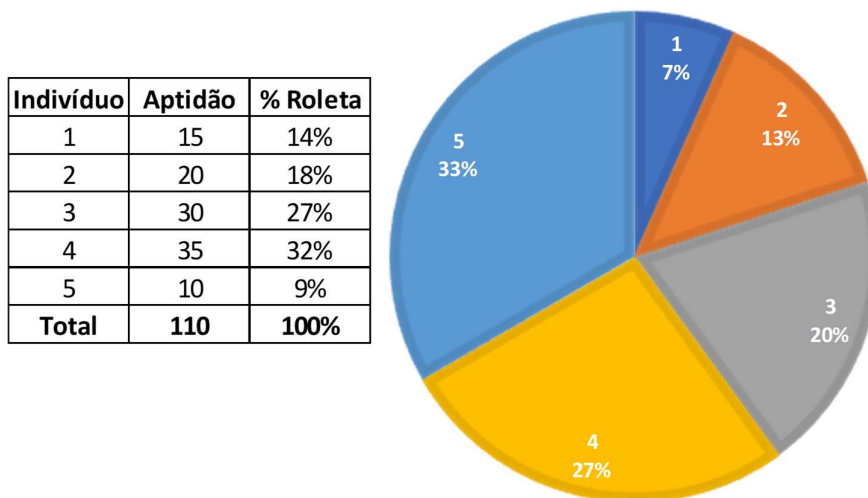
2.4.3 Seleção

A etapa de seleção consiste na escolha dos pares de indivíduos para ser feita a reprodução. Os indivíduos selecionados são chamados de pais. Existem diversos métodos para que seja realizada a seleção, o método mais comum é chamado Método de Seleção por Roleta.

Segundo Goldberg (1989), os indivíduos são representados como se estivessem em uma roleta, onde o espaço ocupado por cada indivíduo será proporcional ao seu valor de aptidão, conhecido através da etapa de avaliação da aptidão. Logo, os indivíduos que apresentarem maior aptidão terão espaços maiores na roleta, consequentemente os piores indivíduos terão espaços menores na roleta.

Para encontrar o valor correspondente a área ocupada na roleta, é realizado a somatória das aptidões da população total e para cada indivíduo é feito o cálculo da divisão do seu valor de aptidão pelo valor da aptidão total da população, encontrando qual o valor proporcional que ele ocupará. Para a escolha dos dois pais, números aleatórios serão gerados representando o giro da roleta, os indivíduos localizados nas posições indicadas pelos números gerados serão os pais selecionados para a geração da nova população. Na Figura 2 é apresentado um modelo de roleta.

Figura 2 – Modelo de Roleta para seleção dos pais



Fonte: A autora

Outro método amplamente utilizado é a seleção por torneio, proposto por Wetzel (1983), onde inicialmente são escolhidos de forma aleatória alguns indivíduos da população total formando uma subpopulação temporária. Em seguida, é aplicada a função objetivo aos integrantes da subpopulação temporária e o indivíduo escolhido é aquele que apresenta a melhor aptidão. Para a escolha dos indivíduos pais são realizados dois torneios, sendo cada torneio para escolha de um pai.

2.4.4 Cruzamento

Método utilizado na formação das novas possíveis soluções, também denominado reprodução e recombinação, consiste na combinação dos dois indivíduos selecionados gerando os dois filhos. O cruzamento só é realizado se um número gerado de forma aleatória variando de 0 a 1 for menor que a probabilidade pré-estabelecida.

Segundo Lobo (2005), os métodos mais utilizados são cruzamento em um ponto e cruzamento em dois pontos. No método de cruzamento em um ponto tem-se a troca de material genético dos indivíduos pais para a formação dos indivíduos filhos. Para este cruzamento um número aleatório é gerado para escolher a posição que ocorrerá a troca dos genes, a posição selecionada é denominada de ponto de corte do cruzamento. Os filhos serão gerados mantendo os genes iguais aos genes dos pais até o ponto de cruzamento e invertendo os genes restantes. Na Figura 3 tem-se o exemplo do cruzamento em um ponto.

Figura 3 – Cruzamento em um ponto de corte

Ponto Selecionado: 3

	1	2	3	4	5	6	7
Pai 1	1	0	1	1	0	1	0
Pai 2	0	0	1	1	1	1	0
Filho 1	1	0	1	1	1	1	0
Filho 2	0	0	1	1	0	1	0

Fonte: Adaptado de (Lobo, 2005)

O método de cruzamento em dois pontos, é similar ao cruzamento em um ponto, a diferença é que dois pontos serão selecionados de forma aleatória, assim os filhos terão os mesmos genes que os pais nas extremidades e no centro terão os genes dos pais invertidos. Na Figura 4 é apresentado o exemplo de cruzamento em dois pontos.

Figura 4 – Cruzamento em dois pontos de corte

Pontos Selecionados: 2 e 4

	1	2	3	4	5	6	7
Pai 1	1	0	1	1	0	1	0
Pai 2	0	0	1	1	1	1	0
Filho 1	1	0	1	1	0	1	0
Filho 2	0	0	1	1	1	1	0

Fonte: Adaptado de (Lobo, 2005)

2.4.5 Mutação

Na etapa de mutação um ou mais genes serão modificados aleatoriamente para que se tenha maior diversificação das novas populações. Uma taxa de mutação é definida, geralmente é uma taxa bem pequena, e para cada indivíduo filho um número aleatório é gerado, caso esse número seja menor que a taxa de mutação o indivíduo filho passará pela mutação genética.

De acordo com Goldberg (1989) os principais métodos de mutação são de mutação aleatória e de mutação por troca. O método de mutação aleatória consiste na geração de valores aleatórios que serão os genes dos indivíduos que sofrerão a mutação. Os genes selecionados

serão trocados por outros valores que também serão gerados de forma aleatória. Na figura 5 é apresentado o modelo de mutação aleatória.

Figura 5 – Mutação aleatória

Gene Selecionado: 5

1	2	3	4	5	6	7
5	8	3	4	9	7	2

1	2	3	4	5	6	7
5	8	3	4	1	7	2

Fonte: Adaptado de (Linden, 2006)

No método de mutação por troca, pares de genes serão gerados de forma aleatória e os genes correspondentes a esses valores serão trocados entre si. Na Figura 6 tem-se o modelo de mutação por troca.

Figura 6 – Mutação por troca

Genes Selecionados: 3 e 5

1	2	3	4	5	6	7
5	8	3	4	9	7	2

1	2	3	4	5	6	7
5	8	9	4	3	7	2

Fonte: Adaptado de (Linden, 2006)

2.4.6 Elitismo

Através do elitismo pode-se manter na próxima geração os melhores indivíduos da geração atual, evitando que as próximas populações sejam piores que a geração atual (Linden, 2006). Com a utilização dos operadores genéticos pode ocorrer do melhor indivíduo ser modificado e caso não utilize o elitismo uma solução boa já encontrada pode ser perdida. O elitismo não é uma etapa obrigatória para codificação do algoritmo genético, mas sua utilização faz com que o código se torne mais eficiente e sua execução seja feita de forma mais rápida.

2.4.7 Atualização da população

Nesta etapa, os indivíduos filhos gerados são inseridos na população substituindo os indivíduos pais. A substituição é feita para que se mantenha o tamanho da população fixo ao

longo das gerações, porque o algoritmo atua em um ambiente com recursos limitados assim como ocorre na natureza. Os indivíduos selecionados na etapa de elitismo também são integrados a nova população nessa fase, garantindo que a nova população não seja pior que a anterior (Linden, 2006).

2.4.8 Critério de parada

As etapas descritas anteriormente são efetuadas durante uma geração e são realizadas sucessivamente até atingir o critério de parada previamente estabelecido. Segundo Banzhaf (1998) os principais critérios de parada são: número máximo de gerações, limite de tempo de execução, convergência, limite de aptidão e valor da função objetivo.

3 DESENVOLVIMENTO

Para o desenvolvimento do projeto de geração de grade horária utilizando algoritmo genético, primeiramente foi definido que o *software* utilizado para programar seria o *Scilab*. Além disso, os dados seriam organizados e apresentados através de planilhas do *Microsoft Excel*.

A grade horária do curso Técnico em Eletrotécnica é dependente da grade horária do Bacharelado em Engenharia Elétrica, porque possuem alguns professores em comum. Foi realizado um levantamento de dados da grade atual com as coordenações dos cursos Técnico em Eletrotécnica e Bacharelado em Engenharia Elétrica do IFG referente ao primeiro semestre do ano de 2022.

Na Tabela 1 tem-se a grade horária do curso de Engenharia Elétrica para o primeiro semestre de 2022. A partir dela foram extraídos os dados dos professores que ministram disciplinas em ambos os cursos e em quais horários eles já estavam alocados.

Tabela 1 - Grade horária do curso de Bacharelado em Engenharia Elétrica em 2022/1

DIA	HORARIO	1º ENG S - 401	3º ENG S - 402	5º ENG S - 403	7º ENG S - 404	8º ENG	9º ENG S - 405
SEGUNDA	07:00 – 08:30	Introd. aos Sistemas Computacionais Aladir	Cálculo Diferencial e Integral II Naara	Instalações Elétricas Patrícia	Cálculo Numérico Fabio	----	Análise de Sistemas Elétricos Fernando P.
	08:45 – 10:15	Língua Portuguesa Rennika	Eletricidade e Magnetismo Substituto	Circuitos Elétricos II Marcelo	Fenômenos de Transporte Substituto	----	Subestação Patrícia
	10:30 – 12:00	Química Geral Carlos Cezar	Equações Diferenciais Fabio	Eletrônica Analógica I Lambert	Máquinas Elétricas II Marcelo	----	Proteção Dori
TERÇA	07:00 – 08:30		Eletricidade e Magnetismo Substituto	Conversão de Energia Camila Dias	Aclonamentos Elétricos André	----	Algoritmos Genéticos Jaqueline
	08:45 – 10:15	Cálculo Diferencial e Integral I Benilson	Eletrônica Digital Lambert	Conversão de Energia Camila Dias	Aclonamentos Elétricos André		Análise de Sistemas Elétricos Fernando P.
	10:30 – 12:00	Cálculo Diferencial e Integral I Benilson	Ondas, Ótica e Termodinâmica Dominike	Eletrônica Analógica I Lambert	Máquinas Elétricas II Marcelo		Análise de Sistemas Elétricos Fernando P.
QUARTA	07:00 – 08:30	Introd. aos Sistemas Computacionais Aladir	Equações Diferenciais Fabio	Instalações Elétricas Patrícia		Libras I Kamila	Processamento Digital de Sinais Camila Dias
	08:45 – 10:15	Língua Portuguesa Rennika	Eletrônica Digital Lambert	Circuitos Elétricos II Marcelo	Transformadores Patrícia	Libras II Kamila	Processamento Digital de Sinais Camila Dias
	10:30 – 12:00	Química Geral Carlos Cezar	Eletricidade e Magnetismo Substituto	Circuitos Elétricos II Marcelo	Cálculo Numérico Fabio	----	Subestação Patrícia
QUINTA	07:00 – 08:30	Fundamentos de Eletricidade Kenya	Cálculo Diferencial e Integral II Naara	Conversão de Energia Camila Dias	Eletrônica de Potência Lambert	----	Algoritmos Genéticos Jaqueline
	08:45 – 10:15	Geometria Analítica Carmenclia	Desenho Básico Rafael	Microprocessadores Kenya	Redes de Computadores Renato		Projeto Interdisciplinar Jaqueline
	10:30 – 12:00	Geometria Analítica Carmenclia	Desenho Básico Rafael	Microprocessadores Kenya	Fenômenos de Transporte Substituto	----	Proteção Dori
SEXTA	07:00 – 08:30	Introdução à Engenharia Kenya	Ondas, Ótica e Termodinâmica Dominike	Métodos e Técnicas de Programação Renato	Eletrônica de Potência Lambert	----	----
	08:45 – 10:15	Cálculo Diferencial e Integral I Benilson	Ondas, Ótica e Termodinâmica Dominike	Métodos e Técnicas de Programação Renato	Eletrônica de Potência Lambert	----	----
	10:30 – 12:00					----	----

Fonte: IFG (2022)

O curso Técnico em Eletrotécnica atualmente possui 35 professores. Para fazer a representação no código foi atribuído um valor numérico para cada professor como mostrado na Tabela 2.

Tabela 2 – Representação numérica dos professores do curso de Técnico em Eletrotécnica

Professores	Cod
SIRLENE	1
ALLINE	2
LUCIANA	3
WELLINGTON	4
DORI	5
LUCIANA BORGES	6
JAQUELINE	7
MANOEL	8
CARLOS CÉZAR	9
EUCLIDES	10
FERNANDO NUNES	11
THIAGO	12
LUCAS	13
ALEXANDRE	14
FAUSTO	15
ISABEL	16
ALYSSON	17
KÊNIA	18
EUCLIDES	19
ANGELITA	20
JAIRO	21
ANDREA	22
ISABEL	23
KARINE	24
ESTEFÂNIA	25
KENNYA	26
RONEY	27
JUAN	28
LAÍSSE	29
RENNIKA	30
Rita/Kamilla	31
LAMBERT	32
MARCO ANT.	33
HUGO	34
FERNANDO P	35

Fonte: A autora

Em seguida, observou-se quais professores também ministram disciplinas para o curso de Engenharia Elétrica. Tendo como base a grade atual do curso de engenharia foi criada uma matriz para esses professores, sinalizando em quais dias eles ministravam disciplinas para o curso de engenharia, conforme mostrado na Tabela 3.

Tabela 3 – Horários preenchidos dos professores no curso de Engenharia Elétrica

Segunda			Terça			Quarta			Quinta			Sexta		
1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula
		x									x			
			x						x	x				
		x		x	x		x		x			x	x	
									x	x	x	x		
x				x	x									
	x						x							
		x						x						

Fonte: A autora

A partir da tabela anterior foram criados vetores de restrição, na grade atual são 7 professores que ministram disciplinas em ambos os cursos, então cada linha representa um professor, cada sinalização na matriz foi substituída pelo código numérico do respectivo professor conforme mostra a Tabela 4.

Tabela 4 – Matriz de restrições dos professores

Segunda			Terça			Quarta			Quinta			Sexta		
1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula	1 Aula	2 Aula	3 Aula
		5									5			
			7						7	7				
		32		32	32		32		32			32	32	
									26	26	26	26		
35				35	35									
	30						30							
		9						9						

Fonte: A autora

Para inserção da matriz de restrição da Tabela 4 no código, ela foi comprimida para 3 vetores de restrição, que funcionarão como uma máscara para comparação com os indivíduos gerados. Cada vetor de restrição possui 25 posições, 5 posições para cada dia letivo da semana, representando todas as possibilidades de aula do curso Técnico Integrado em Eletrotécnica. Na Tabela 5 os vetores de restrição são apresentados.

Tabela 5 - Vetores de restrição

Aula	Rest1	Rest2	Rest3
1 Seg	0	0	35
2 Seg	0	0	30
3 Seg	32	9	5
4 Seg	0	0	0
5 Seg	0	0	0
1 Ter	0	0	7
2 Ter	32	0	35
3 Ter	32	0	35
4 Ter	0	0	0
5 Ter	0	0	0
1 Qua	0	0	0
2 Qua	32	0	30
3 Qua	0	9	0
4 Qua	0	0	0
5 Qua	0	0	0
1 Qui	32	26	7
2 Qui	0	26	7
3 Qui	0	26	5
4 Qui	0	0	0
5 Qui	0	0	0
1 Sex	32	26	0
2 Sex	32	0	0
3 Sex	0	0	0
4 Sex	0	0	0
5 Sex	0	0	0

Fonte: A autora

Definiu-se que o algoritmo deveria retornar o horário mais otimizado em relação à quantidade de dias trabalhados por professores e que o mesmo professor não poderia ser agendado para mais de uma turma no mesmo horário. Para a função objetivo identificou-se que a melhor forma de medir e comparar os dados seria através de funções estatísticas, então as funções que mais se adequaram ao problema foram as funções de média aritmética e de variância.

Para garantir que o resultado seja uma quantidade de dias bem próximas para os professores a função precisaria retornar o resultado para a menor variância. No *software Scilab* já existe uma função para o cálculo da variância, que foi então utilizada como sendo a função objetivo do código tendo como parâmetros os dias trabalhos por cada professor por indivíduo e a média de cada vetor. Após essa definição as funções foram usadas para mensurar os dias

trabalhados por cada professor, o algoritmo que retornasse o menor valor de variância seria o melhor resultado.

Para a criação dos indivíduos o modelo escolhido foi uma matriz de vetores para cada ano. O curso Técnico em Eletrotécnica possui aulas de segunda-feira à sexta-feira e disciplinas no período matutino e vespertino, tendo três aulas pela manhã e duas pela tarde, totalizando um máximo de cinco aulas por dia, mas podendo haver horários livres. Assim, cada vetor possui 25 posições para representar todos os horários possíveis. O exemplo do indivíduo é apresentado na Figura 7.

Figura 7 – Indivíduo modelo

DIA	AULA	1 ANO	2 ANO	3 ANO
SEG	1			
	2			
	3			
	1			
	2			
TER	1			
	2			
	3			
	1			
	2			
QUA	1			
	2			
	3			
	1			
	2			
QUI	1			
	2			
	3			
	1			
	2			
SEX	1			
	2			
	3			
	1			
	2			

Fonte: A autora

Para a geração das populações foram criados indivíduo e sub indivíduos. Como os valores não seriam iguais para cada ano porque alguns professores que ministram disciplinas para o primeiro ano não são os mesmos professores que ministram disciplinas no segundo e no

terceiro ano, a geração de novos indivíduos seria feita por uma matriz de vetores para cada ano. Dessa forma, os sub indivíduos seriam os vetores e o indivíduo seria a matriz.

No código desenvolvido no *software Scilab*, para a geração dos indivíduos foi criado um vetor com 25 posições para cada ano com os códigos numéricos referentes a cada professor que ministra disciplinas para o respectivo ano e com o valor zero para as posições restantes, que serão os horários livres. No *software Scilab* existe uma função que faz uma distribuição randômica de valores pré-definidos, usando essa função foi feita a geração dos sub indivíduos. Definiu-se que cada população gerada teria 30 indivíduos, obtendo populações com 90 sub indivíduos com 25 genes cada, para a geração máxima foi fixado o valor de 350.

Após a geração de cada população, os indivíduos gerados são submetidos há uma verificação das restrições de horários dos professores e de duplicidade de professores alocados no mesmo horário entre os anos. Caso aconteça uma dessas condições, de forma randômica é feita uma correção que novamente é verificada e assim sucessivamente até que o indivíduo seja considerado apto.

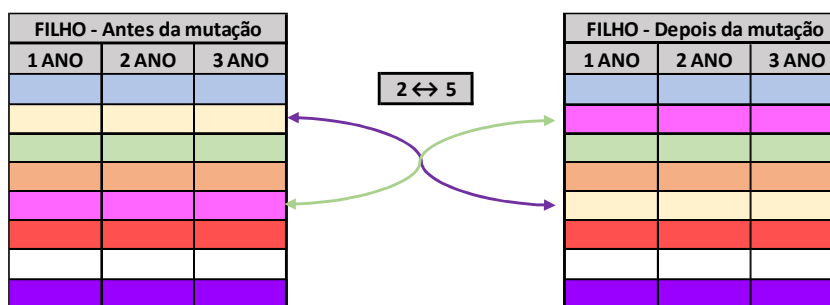
Assim que a população é considerada apta ela passa por uma função que faz a avaliação dos indivíduos encontrados, retornando à quantidade de dias trabalhados por professor. A partir dessa função é feito o cálculo da variância por indivíduo onde são selecionados os dois melhores indivíduos para utilizar a técnica de elitismo. Dessa forma, garantindo que a próxima população terá um resultado igual ou melhor ao da população atual.

Para a geração da próxima população é utilizado o método da roleta para a escolha dos pais da população atual. Em seguida, é realizada a etapa de cruzamento, onde uma taxa de cruzamento previamente definida será comparada com valores gerados de forma aleatória para cada dupla de pais caso o valor gerado seja menor que a taxa será feita a recombinação do código genético dos pais gerando os filhos, como o indivíduo é formado por três sub indivíduos foi definido que a combinação seria feita a partir de um ponto de corte escolhido aleatoriamente, podendo ser 1 ou 2.

Caso seja escolhido 1 para o ponto de corte, os filhos teriam os mesmos genes dos respectivos pais para o primeiro ano e os genes dos pais invertidos para o segundo e terceiro ano, conforme mostra a Figura 8.

comparado com o valor fixo da probabilidade de mutação, caso esse valor aleatório seja menor que o valor fixo o filho sofrerá mutação genética. Nessa etapa, serão gerados dois valores de forma aleatória representando quais genes serão mutados, trocando o valor de um pelo outro, conforme mostra a Figura 10.

Figura 10 – Mutação dos filhos nos cromossomos 2 e 5



Fonte: A autora

Após a mutação, será finalizada a nova população com a população dos filhos e serão também inseridos os indivíduos selecionados anteriormente pela técnica de elitismo. A nova população passa pela função de restrições para garantir que os indivíduos atendem aos requisitos definidos previamente. Essa é avaliada pela função objetivo onde um valor de variância é atribuído para cada novo indivíduo, os dois melhores indivíduos são selecionados novamente pelo elitismo e os melhores valores são armazenados em um vetor, o processo se repete até que as 350 gerações são criadas e avaliadas. Dessa forma, é apresentado o resultado final e um gráfico mostrando os melhores indivíduos por geração.

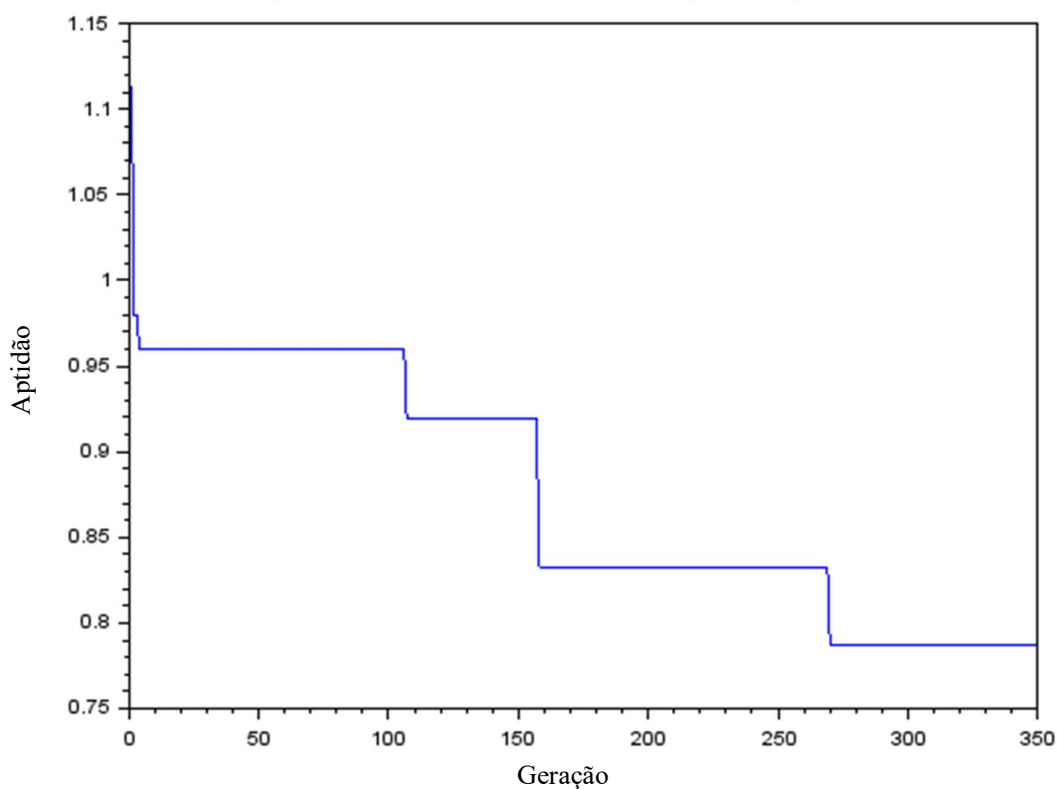
O resultado é um indivíduo com 3 sub indivíduos formado por 25 genes cada. Os sub indivíduos são inseridos em uma tabela do *Microsoft Excel*, com uma fórmula que procura por valores em tabelas e faz uma busca no nome do professor que cada valor representa, tendo assim o horário final do curso Técnico em Eletrotécnica. Para melhor visualização e comparação, o código armazena a melhor solução de cada geração em um vetor, e esse vetor é utilizado para geração dos gráficos dos melhores indivíduos através das gerações. Ao final da execução um vetor com a quantidade de dias trabalhados por professor também é apresentado. O código completo pode ser visualizado no Apêndice A.

4 RESULTADOS

Após realizadas execuções do código pode-se perceber que com 350 gerações garante-se o resultado mais otimizado. Os valores sempre convergem para o mesmo valor de variância a partir da 270ª geração, mas em algumas execuções o valor foi atingido após a 300ª geração.

Na Figura 5 tem a execução do código para um total de 350 gerações, encontrando o valor da solução próximo da 270ª geração.

Figura 11 – Evolução dos indivíduos ao longo de 350 gerações



Fonte: A autora

Através do gráfico pode-se observar a eficácia do elitismo, sua utilização faz com que o algoritmo preserve sua melhor solução ao longo das gerações, retornando apenas soluções melhores ou iguais ao melhor indivíduo da geração anterior. Com a utilização do elitismo o código converge para a melhor solução em um menor espaço de tempo tendo assim a otimização da sua execução.

O melhor valor retornado para a variância foi de 0,787, encontrado com uma taxa de 0,85 para probabilidade de cruzamento, uma taxa de 0,10 para probabilidade de mutação do

indivíduo e em aproximadamente 10 minutos de execução do código. Um dos requisitos utilizados para se obter a melhor otimização é que o resultado tenha a menor quantidade de dias trabalhados por professor. Escolheu-se como medida de otimização a variância, para que o resultado possa ter o menor valor de dias trabalhado por professor e de forma que não tenha valores vindos dos extremos, mas que cada professor possa ter uma quantidade média bem próxima de dias de trabalho durante a semana.

Foi retornado um vetor com a quantidade de dias trabalhados por cada professor, que foi referido pelo seu código numérico e através do *Excel* os dados foram tabelados conforme mostra a Tabela 6. No caso do professor Lambert, ele já possuía em sua grade de horários restritos todos os dias da semana preenchidos para o Bacharelado em Engenharia Elétrica, não tendo como resultar em um número menor de dias trabalhados para ele. Quanto aos outros professores, pode-se notar que os valores retornados são bem próximos.

Tabela 6 - Dias trabalhados por cada professor

Cod	Dias Trabalho	Prof
32	5	LAMBERT
7	3	JAQUELINE
15	3	FAUSTO
26	3	KENNYA
30	3	RENNIKA
5	2	DORI
6	2	LUCIANA BORGES
8	2	MANOEL
9	2	CARLOS CÉZAR
10	2	EUCLIDES
11	2	FERNANDO NUNES
13	2	LUCAS
16	2	ISABEL
17	2	ALYSSON
18	2	KÊNIA
25	2	ESTEFÂNIA
35	2	FERNANDO P
1	1	SIRLENE
2	1	ALLINE
3	1	LUCIANA
4	1	WELLINGTON
12	1	THIAGO
14	1	ALEXANDRE
19	1	EUCLIDES
20	1	ANGELITA
21	1	JAIRO
22	1	ANDREA
23	1	ISABEL
24	1	KARINE
27	1	RONEY
28	1	JUAN
29	1	LAÍSSE
31	1	Rita/Kamilla
33	1	MARCO ANT.
34	1	HUGO

Fonte: A autora

O melhor indivíduo, resultante do código desenvolvido, foi inserido no *Excel* conforme mostra a Tabela 7. Em seguida, esse foi convertido para a grade horária. A grade horária retornada é a mais otimizada obedecendo as restrições impostas.

Tabela 7 – Melhor indivíduo

1ano	2ano	3ano
5	27	7
3	17	7
10	22	35
5	0	31
16	0	18
15	8	30
2	13	33
1	19	17
1	23	28
6	35	0
4	11	26
13	11	34
13	15	10
0	11	0
9	11	0
14	25	29
14	21	11
11	20	25
7	18	0
5	6	0
12	15	16
15	25	26
8	24	0
0	15	0
0	26	32

Fonte: A autora

A grade horária finalizada é apresentada na Tabela 8. Com a análise dessa tabela, verifica-se a efetividade do código retornando uma grade sem coincidência de professores no mesmo horário e em turmas diferentes, com o menor número possível de dias trabalhados por professor e sem coincidir também com as restrições de horário previamente impostas que representam as aulas que esses professores ministram em outros cursos.

Tabela 8 – Grade horária final

Dia	Hora	1ano		2ano		3ano	
SEGUNDA	07:00	5	DORI	27	RONEY	7	JAQUELINE
	08:45	3	LUCIANA	17	ALYSSON	7	JAQUELINE
	10:15	10	EUCLIDES	22	ANDREA	35	FERNANDO P
	13:00	5	DORI	0	Livre	31	Rita/Kamilla
	14:45	16	ISABEL	0	Livre	18	KÊNIA
TERÇA	07:00	15	FAUSTO	8	MANOEL	30	RENNIKA
	08:45	2	ALLINE	13	LUCAS	33	MARCO ANT.
	10:15	1	SIRLENE	19	EUCLIDES	17	ALYSSON
	13:00	1	SIRLENE	23	ISABEL	28	JUAN
	14:45	6	LUCIANA BORGES	35	FERNANDO P	0	Livre
QUARTA	07:00	4	WELLINGTON	11	FERNANDO NUNES	26	KENNYA
	08:45	13	LUCAS	11	FERNANDO NUNES	34	HUGO
	10:15	13	LUCAS	15	FAUSTO	10	EUCLIDES
	13:00	0	Livre	11	FERNANDO NUNES	0	Livre
	14:45	9	CARLOS CÉZAR	11	FERNANDO NUNES	0	Livre
QUINTA	07:00	14	ALEXANDRE	25	ESTEFÂNIA	29	LAÍSSE
	08:45	14	ALEXANDRE	21	JAIR	11	FERNANDO NUNES
	10:15	11	FERNANDO NUNES	20	ANGELITA	25	ESTEFÂNIA
	13:00	7	JAQUELINE	18	KÊNIA	0	Livre
	14:45	5	DORI	6	LUCIANA BORGES	0	Livre
SEXTA	07:00	12	THIAGO	15	FAUSTO	16	ISABEL
	08:45	15	FAUSTO	25	ESTEFÂNIA	26	KENNYA
	10:15	8	MANOEL	24	KARINE	0	Livre
	13:00	0	Livre	15	FAUSTO	0	Livre
	14:45	0	Livre	26	KENNYA	32	LAMBERT

Fonte: A autora

5 CONCLUSÕES

Neste trabalho foi desenvolvida uma solução para a geração da grade horária do curso Técnico Integrado em Eletrotécnica do Instituto Federal de Goiás – Campus Jataí utilizando algoritmo genético. O algoritmo desenvolvido gera uma população inicial que é aprimorada de forma aleatória até se tornar apta as condições pré-determinadas. Após ser considerada apta, a população ainda é submetida a várias técnicas de alteração genética gerando novas populações, as quais também são validadas para serem consideradas aptas aumentando a possibilidade de encontrar o indivíduo ideal. Dessa forma, a solução mais viável pode ser encontrada em um vasto espaço de busca.

O resultado determinado pelo algoritmo genético desenvolvido obteve excelente desempenho, uma vez que encontrou uma boa solução, a qual obedece a todos os critérios pré-estabelecidos. O algoritmo converge em aproximadamente 270 gerações, levando um tempo consideravelmente rápido para encontrar a solução. Considerando que as grades horárias atualmente são feitas de forma manual e que a grade do curso Técnico Integrado em Eletrotécnica precisa ser refeita a cada novo semestre letivo, o código será de grande utilidade devido à rapidez e eficiência apresentados, gerando novas grades aptas em apenas 10 minutos de execução.

REFERÊNCIAS BIBLIOGRÁFICAS

BANZHAF, W.; NORDIN, P.; KELLER, R. E. & FRANCONI, F. D. **Genetic Programming: an introduction**. ISBN 155860510X. Morgan Kaufmann, 1998.

FOGEL, L. J. **Autonomous automata**. Industrial Research, vol. 4, pp. 14-19, 1962.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley, 1989.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. The University of Michigan Press, Ann Arbor, MI, 1975.

LINDEN, R. **Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional**. Brasport, 2006.

LOBO, E. L. M. **Uma solução do problema de horário escolar via algoritmo genético paralelo**. CEFET-MG, 2005.

MICHALEWICZ, Z. & FOGEL, D. B. **How to solve it: Modern Heuristics**. Springer-Verlag, 2000.

MITCHELL, M. **An introduction to genetic algorithms**. Cambridge: Mit Press. 1997. 207 p.

WETZEL, A. **Evaluation of the effectiveness of genetic algorithms in combinatorial optimization**. University of Pittsburg, 1983.

APÊNDICE A – Código desenvolvido para geração da grade horária

O código do algoritmo genético desenvolvido nesse trabalho foi implementado no *software Scilab* e pode ser visto a seguir.

```

clear
clc
//Funções
function p=auxrest(g, i, p)
    troca = grand(1, "uin", 1,gen)
    while troca <= g
        troca = grand(1, "uin", 1,gen)
    end
    aux1 = p(i,troca)
    p(i,troca) = p(i,g)
    p(i,g) = aux1
endfunction

function p=auxrest2(g, i, p)
    troca = grand(1, "uin", 1,gen)
    while troca == g
        troca = grand(1, "uin", 1,gen)
    end
    aux1 = p(i,troca)
    p(i,troca) = p(i,g)
    p(i,g) = aux1
endfunction

function [p1, p2, p3]=restricao(p1, p2, p3, r1, r2, r3)
    for i=1:ind
        for g=1:gen
            if r1(g)<>0 || r2(g)<>0 || r3(g)<>0
                if r1(g) == p1(i,g) || r2(g) == p1(i,g) || r3(g) == p1(i,g) & p1(i,g)<>0
                    p1=auxrest(g,i,p1)
                end
            end
        end
    end
endfunction

```

```

    end
    if r1(g) == p2(i,g) || r2(g) == p2(i,g) || r3(g) == p2(i,g) & p2(i,g) <> 0
        p2=auxrest(g,i,p2)
    end
    if r1(g) == p3(i,g) || r2(g) == p3(i,g) || r3(g) == p3(i,g) & p3(i,g) <> 0
        p3=auxrest(g,i,p3)
    end
end
end
end

for i=1:ind
    for g=1:gen
        if p1(i,g) <> 0
            if p1(i,g) == p2(i,g)
                p2=auxrest2(g,i,p2)
            end

            if p1(i,g) == p3(i,g)
                p3=auxrest2(g,i,p3)
            end
        end
        if p2(i,g) <> 0
            if p2(i,g) == p1(i,g)
                p1=auxrest2(g,i,p1)
            end

            if p2(i,g) == p3(i,g)
                p3=auxrest2(g,i,p3)
            end
        end
    end
end
endfunction

```

```

function f=fitf(p1, p2, p3, r1, r2, r3)
//35 professores
f=zeros(ind,prof)
for aux2=1:ind
    for aux=1:prof
        a=1
        aux3=0
        while a<=gen
            if p1(aux2,a)==aux
                aux3=ceil(a/5)
            end
            if p2(aux2,a)==aux
                aux3=ceil(a/5)
            end
            if p3(aux2,a)==aux
                aux3=ceil(a/5)
            end
            if r1(1,a)==aux
                aux3=ceil(a/5)
            end
            if r2(1,a)==aux
                aux3=ceil(a/5)
            end
            if r3(1,a)==aux
                aux3=ceil(a/5)
            end
            if (modulo(a,5)==0 & aux3>0)
                f(aux2,aux)=f(aux2,aux)+1
                aux3=0
            end
            a=a+1
        end
    end
end
end

```

```

end
endfunction

```

```

function roleta=Roleta(aval)
    indiv = size(aval)(1)
    nCruz = ceil(indiv / 2)-1
    roleta = zeros(nCruz, 2)
    avalTotal = sum(aval)
    aval = cumsum(aval ./ avalTotal)
    for i = 1:nCruz
        r = rand()
        aux = find(aval >= r)
        selecao1 = aux(1)
        while 1
            r = rand()
            aux = find(aval >= r)
            selecao2 = aux(1)
            if selecao1 ~= selecao2 then
                break
            end
        end
        roleta(i, :) = [selecao1, selecao2]
    end
endfunction

```

```

function [filho1, filho2, filho3]=Crossover(p1, p2, p3, selecao)
    filho1= zeros(ind,gen)
    filho2=zeros(ind,gen)
    filho3= zeros(ind,gen)
    nCruz = size(selecao)(1)
    for i = 1:nCruz
        r = rand()
        if r < pCruz
            ponto = grand(1,"uin", 1,2)

```

```

    if ponto == 1
        filho1(i,:)=p1(selecao(i,1),:)
        filho1(nCruz+i,:)=p1(selecao(i,2),:)
        filho2(i,:)=p2(selecao(i,2),:)
        filho2(nCruz+i,:)=p2(selecao(i,1),:)
        filho3(i,:)=p3(selecao(i,2),:)
        filho3(nCruz+i,:)=p3(selecao(i,1),:)
    end
    if ponto == 2
        filho1(i,:)=p1(selecao(i,1),:)
        filho1(nCruz+i,:)=p1(selecao(i,2),:)
        filho2(i,:)=p2(selecao(i,1),:)
        filho2(nCruz+i,:)=p2(selecao(i,2),:)
        filho3(i,:)=p3(selecao(i,2),:)
        filho3(nCruz+i,:)=p3(selecao(i,1),:)
    end
end
end
if r > pCruz
    filho1(i,:)=p1(selecao(i,1),:)
    filho1(nCruz+i,:)=p1(selecao(i,2),:)
    filho2(i,:)=p2(selecao(i,1),:)
    filho2(nCruz+i,:)=p2(selecao(i,2),:)
    filho3(i,:)=p3(selecao(i,1),:)
    filho3(nCruz+i,:)=p3(selecao(i,2),:)
end
end
endfunction

function [filho1, filho2, filho3]=Mutacao(filho1, filho2, filho3, pInd)
[indiv, genes] = size(filho1)
for i = 1:indiv-2
    r = rand()
    if r < pInd then
        aux1 = grand(1, "uin", 1, genes)

```

```

    aux2 = grand(1, "uin", 1, genes)
    aux3=filho1(i,aux1)
    filho1(i,aux1)=filho1(i,aux2)
    filho1(i,aux2)=aux3

    aux3=filho2(i,aux1)
    filho2(i,aux1)=filho2(i,aux2)
    filho2(i,aux2)=aux3

    aux3=filho3(i,aux1)
    filho3(i,aux1)=filho3(i,aux2)
    filho3(i,aux2)=aux3
end
end
endfunction

//Definição de constantes

grand("setsd", 5)
ind = 30
gen = 25
prof = 35
geracao = 1
gMax = 350

//Iniciando a população

//Vetores de professores e aulas livres por ano para ser a base da criação randômica dos
indivíduos
vet1 = [1 2 3 4 5 6 7 1 8 9 10 11 12 13 14 14 5 15 5 15 13 16 0 0 0]
vet2 = [11 13 17 15 11 18 15 6 11 11 8 19 20 21 35 22 23 15 24 25 25 26 27 0 0]
vet3 = [28 17 29 30 18 31 11 7 32 16 33 7 10 26 26 34 25 35 0 0 0 0 0 0 0]

//Inserção dos vetores de restrição

```

```

rest1 = [0 0 32 0 0 0 32 32 0 0 0 32 0 0 0 32 0 0 0 0 32 32 0 0 0]
rest2 = [0 0 9 0 0 0 0 0 0 0 0 0 9 0 0 26 26 26 0 0 26 0 0 0 0]
rest3 = [35 30 5 0 0 7 35 35 0 0 0 30 0 0 0 7 7 5 0 0 0 0 0 0]

//Gerando a população inicial através de permuta dos vetores criados para cada ano
pop1 = grand(ind, "prm", vet1)
pop2 = grand(ind, "prm", vet2)
pop3 = grand(ind, "prm", vet3)

disp("SubPopulação de 1 anos", pop1)
disp("SubPopulação de 2 anos", pop2)
disp("SubPopulação de 3 anos", pop3)
disp("Restrições:",rest1,rest2,rest3)

while 1
    w=0

    for aux5=1:ind
        for aux6=1:gen
            if pop1(aux5,aux6)>0
                w=isequal(pop1(aux5,aux6),pop2(aux5,aux6))+w
                w=isequal(pop1(aux5,aux6),pop3(aux5,aux6))+w
                w=isequal(pop1(aux5,aux6),rest1(1,aux6))+w
                w=isequal(pop1(aux5,aux6),rest2(1,aux6))+w
                w=isequal(pop1(aux5,aux6),rest3(1,aux6))+w
            end
            if pop2(aux5,aux6)>0
                w=isequal(pop2(aux5,aux6),pop3(aux5,aux6))+w
                w=isequal(pop2(aux5,aux6),rest1(1,aux6))+w
                w=isequal(pop2(aux5,aux6),rest2(1,aux6))+w
                w=isequal(pop2(aux5,aux6),rest3(1,aux6))+w
            end
            if pop3(aux5,aux6)>0
                w=isequal(pop3(aux5,aux6),rest1(1,aux6))+w
                w=isequal(pop3(aux5,aux6),rest2(1,aux6))+w
            end
        end
    end
end

```

```

        w=isequal(pop3(aux5,aux6),rest3(1,aux6))+w
    end
end
end
if w > 0
    [pop1,pop2,pop3]=restricao(pop1,pop2,pop3,rest1,rest2,rest3)
end
if w==0
    break
end
end
end

```

```

disp("População após Restrição:",pop1,pop2,pop3)
//Avaliar População
f1 = fitf(pop1,pop2,pop3,rest1,rest2,rest3)
//Fit da População
disp("fit",f1)
for aux7=1:ind
    md(aux7,1)=sum(f1(aux7,:))
end
md = md/prof
for a3=1:ind
    var(a3,1) = variance(f1(a3,:), 2, md(a3,1))
end
disp("Variância:",var)

```

```

//Elitismo selecionando os dois melhores indivíduos:

```

```

[melhorAv(1),n(1)] = min(var)
elitismo = var
[elit1, k1] = min(elitismo)
elitismo(k1,:) = max(elitismo)
[elit2, k2] = min(elitismo)

```

```

//Novas Populações

```

```

while(geracao < gMax) do

    //Seleção dos pais pelo método da Roleta
    selecao = Roleta(var)
    disp("Seleção",selecao)

    //Recombinação dos filhos
    fi1= zeros(ind,gen)
    fi2=zeros(ind,gen)
    fi3= zeros(ind,gen)
    pCruz=0.85
    [fi1,fi2,fi3] = Crossover(pop1,pop2,pop3, selecao)

    //Mutação
    pInd = 0.10
    disp("Probabilidade de mutação: ",pInd)
    [fi1,fi2,fi3] = Mutacao(fi1,fi2,fi3, pInd)

    //Integrando a nova população:
    fi1($,:) = pop1(k1,:)
    fi1($-1,:) = pop1(k2,:)
    fi2($,:) = pop2(k1,:)
    fi2($-1,:) = pop2(k2,:)
    fi3($,:) = pop3(k1,:)
    fi3($-1,:) = pop3(k2,:)

    //Nova população
    [pop1,pop2,pop3] = (fi1,fi2,fi3)
    disp(pop1,pop2,pop3)
    //Avaliar nova população
    while 1
        w=0
        for aux5=1:ind

```

```

for aux6=1:gen
    if pop1(aux5,aux6)>0
        w=isequal(pop1(aux5,aux6),pop2(aux5,aux6))+w
        w=isequal(pop1(aux5,aux6),pop3(aux5,aux6))+w
        w=isequal(pop1(aux5,aux6),rest1(1,aux6))+w
        w=isequal(pop1(aux5,aux6),rest2(1,aux6))+w
        w=isequal(pop1(aux5,aux6),rest3(1,aux6))+w
    end
    if pop2(aux5,aux6)>0
        w=isequal(pop2(aux5,aux6),pop3(aux5,aux6))+w
        w=isequal(pop2(aux5,aux6),rest1(1,aux6))+w
        w=isequal(pop2(aux5,aux6),rest2(1,aux6))+w
        w=isequal(pop2(aux5,aux6),rest3(1,aux6))+w
    end
    if pop3(aux5,aux6)>0
        w=isequal(pop3(aux5,aux6),rest1(1,aux6))+w
        w=isequal(pop3(aux5,aux6),rest2(1,aux6))+w
        w=isequal(pop3(aux5,aux6),rest3(1,aux6))+w
    end
end
end
if w > 0
    [pop1,pop2,pop3]=restricao(pop1,pop2,pop3,rest1,rest2,rest3)
end
if w==0
    break
end
end
geracao = geracao + 1

```

//Integrando a elite na nova população:

```

pop1($,:) = f1($,:)
pop1($-1,:) = f1($-1,:)
pop2($,:) = f2($,:)

```

```

pop2($-1,:) = fi2($-1,:);
pop3($,:) = fi3($,:);
pop3($-1,:) = fi3($-1,:);

//Avaliar População
f1 = fitf(pop1,pop2,pop3,rest1,rest2,rest3)

//Fit da População
disp("fit",f1)
for aux7=1:ind
    md(aux7,1)=sum(f1(aux7,:))
end
md = md/prof
for a3=1:ind
    var(a3,1) = variance(f1(a3,:), 2, md(a3,1))
end
disp("Variância:",var)

//Elitismo selecionando os dois melhores indivíduos:
elitismo = var
[elit1, k1] = min(elitismo)
elitismo(k1,:) = max(elitismo)
[elit2, k2] = min(elitismo)
[melhorAv(geracao),n(geracao)] = min(var)
end

//Resultados
disp("Melhores Avaliações", melhorAv)
Rf = min(melhorAv)
a = 1
for dd = 1:gMax
    if melhorAv(dd) == Rf
        Rfinal(a) = melhorAv(dd)
        aux(a) = dd
    end
end

```

```
        a = a+1
    end
end

x = aux($)
disp("Melhor Sequencia: ",pop1(n(x,:),:),pop2(n(x,:),:),pop3(n(x,:),:))
disp("Melhor Variancia: ",var(n(x,:)))
disp("Qtd Dias de Trabalho",f1(n(x,:)))
plot(melhorAv)
```