

UNIVERSIDADE DE PASSO FUNDO
Programa de Pós-Graduação em
Computação Aplicada

Dissertação de Mestrado

**SOLUÇÃO DE TIMETABLING PARA
UNIVERSIDADE COM FOCO NA
OTIMIZAÇÃO PARA AVALIAÇÃO DOS
CURSOS PELO ENADE**

MATHEUS FELIPE FABIAN



**UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

**SOLUÇÃO DE TIMETABLING PARA
UNIVERSIDADE COM FOCO NA
OTIMIZAÇÃO PARA AVALIAÇÃO DOS
CURSOS PELO ENADE**

Matheus Felipe Fabian

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Marcelo Trindade Rebonatto

Passo Fundo
2024

CIP – Catalogação na Publicação

F118s Fabian, Matheus Felipe
Solução de *Timetabling* para universidade com foco na
otimização para avaliação dos cursos pelo Enade [recurso
eletrônico] / Matheus Felipe Fabian. – 2024
2 MB ; PDF.

Orientador: Prof. Marcelo Trindade Rebonatto.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2024.

1. Algoritmos genéticos. 2. Otimização combinatória.
3. Universidades e faculdades - Corpo docente. 4. Sistema
Nacional de Avaliação da Educação Superior (Brasil).
I. Rebonatto, Marcelo Trindade, orientador. II. Título.


CDU: 004

Catalogação: Bibliotecária Jucelei Rodrigues Domingues - CRB 10/1569


ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO

MATHEUS FELIPE FABIAN


Aos dezenove dias do mês de março do ano de dois mil e vinte e quatro, às quatorze horas, realizou-se, de forma on-line, por meio de videoconferência e presencial no Auditório da UPF ONLINE (Prédio D1). A sessão pública de defesa do Trabalho de Conclusão de Curso “SOLUÇÃO DE TIMETABLING PARA UNIVERSIDADE COM FOCO NA OTIMIZAÇÃO PARA AVALIAÇÃO DOS CURSOS PELO ENADE”, de autoria de Matheus Felipe Fabian, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos professores doutores Edson Luiz Padoin, Marcelo Trindade Rebonatto e Rafael Rieder. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato **APROVADO**. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.

Documento assinado digitalmente
 **MARCELO TRINDADE REBONATTO**
Data: 26/03/2024 10:15:51-0300
Verifique em <https://validar.iti.gov.br>


Prof. Dr. Marcelo Trindade Rebonatto
- UPF-
Presidente da Banca Examinadora
(Orientador)

Documento assinado digitalmente
 **RAFAEL RIEDER**
Data: 26/03/2024 15:41:32-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Rafael Rieder
- UPF-
(Avaliador Interno)

Documento assinado digitalmente
 **EDSON LUIZ PADOIN**
Data: 01/04/2024 21:57:14-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Edson Luiz Padoin
- UNIJUÍ-
(Avaliador Externo)

Documento assinado digitalmente
 **CARLOS AMARAL HOLBIG**
Data: 26/03/2024 16:47:10-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Carlos Amaral Hölbig - UPF
(Coordenador do PPGCA)

SOLUÇÃO DE TIMETABLING PARA UNIVERSIDADE COM FOCO NA OTIMIZAÇÃO PARA AVALIAÇÃO DOS CURSOS PELO ENADE

RESUMO

O *Curriculum-Based Course Timetabling* é um problema de agendamento de disciplinas, professores e cursos em períodos. Como há uma grande variedade de soluções, implementar uma grade horária sem gerar conflitos torna-se uma tarefa trabalhosa. Algoritmos de otimização são comumente utilizados para resolver esse problema, gerando soluções viáveis e possibilitando modelar o problema segundo restrições. Isso pode ser utilizado pela instituição para a solução se adaptar às necessidades específicas. Sendo assim, é possível construir um algoritmo que se adapte ao Sistema Nacional de Avaliação feito nas instituições de ensino superior do Brasil. Essas avaliações tem como um de seus fatores para o cálculo da nota de um curso a composição do corpo docente, mais especificamente, a quantidade de professores com titulação mestre e doutor e a quantidade de professores com regime de trabalho parcial ou integral. Este trabalho apresenta a modificação do algoritmo genético gerador de grade horária da UPF, destinada a otimizar a nota do corpo docente conforme a avaliação efetuada pelo Sistema Nacional de Avaliação da Educação Superior. Através da mudança no cálculo da função de avaliação e adicionando novas *constraints* com pesos definidos mediante testes preliminares, foi possível direcionar as soluções para alocarem maior quantidade de professores que cumprem os insumos avaliados. A otimização foi aplicada aos 18 cursos que participaram do Enade no ano de 2023. Os resultados foram positivos, com 17 cursos alcançando um aumento estatisticamente significativo na alocação de professores em pelo menos um dos três insumos otimizados, 16 cursos tendo um aumento na alocação de doutores e de professores com regime de trabalho desejados com diferença estatisticamente significativa em relação ao algoritmo original, e três cursos aumentaram de forma estatisticamente significativa a alocação de mestres, sendo que 11 cursos já possuíam 100% de mestres alocados no algoritmo original. Ainda vale ressaltar que dois cursos conseguiram preencher em 100% o número de professores alocados que atendem a todos os requisitos impostos pela avaliação.

Palavras-Chave: CB-CTT; Algoritmo Genético; Otimização; CPC; SINAES.

TIMETABLING SOLUTION FOR UNIVERSITY WITH FOCUS ON OPTIMIZATION FOR THE COURSES EVALUATION BY ENADE

ABSTRACT

The Curriculum-Based Course Timetabling is a scheduling problem that allocate disciplines, professors and courses in time periods. There are a variety of possibilities for allocate each professor to a discipline at some time period, so, implementing a timetable without generating any conflict becomes a hard task. Optimization algorithms are commonly used to solve this problem, generating viable solutions and making it possible to model the problem according to constraints. This can be used by the institution to adapt the solution to specific needs. Then, it is possible to build an algorithm that adapts to the evaluation system of higher education institutions in Brazil. These evaluations have as one of their factors for calculating the grade of a course the composition of the teaching staff, more specifically, the number of professors with master's and doctorate degrees and the number of professors with partial or full-time work. This paper modified the genetic algorithm used by UPF to optimize the teaching staff's grade according to the assessment applied by SINAES. By changing the calculation of the algorithm's evaluation function and adding new constraints with weights defined through preliminary tests, it was possible to direct the solutions to allocate a greater number of professors that fit in the considered factors. The optimization was applied to the 18 courses that participated in Enade in the year 2023. The results were positive, with 17 courses achieving a statistically significant increase in the allocation of professors in at least one of the three optimized factors, 16 courses having an increase in the allocation of doctors and professors with the desired work regime with a statistically significant difference in relation to the original algorithm, and three courses increased the allocation of masters in a statistically significant way, with 11 courses already having 100% masters allocated in the original algorithm. It is also worth highlighting that two courses managed to fill 100% the number of allocated professor who fit all the requirements imposed by the assessment.

Keywords: CB-CTT; Genetic Algorithm; Optimization; CPC; SINAES.

LISTA DE FIGURAS

Figura 1	– Pseudocódigo de Algoritmo Genético adaptado de [33].	17
Figura 2	– Na parte superior cruzamento de ponto único e embaixo cruzamento de 2 pontos [32].	21
Figura 3	– Representação de uma ótima local e ótima global de uma função [35].	22
Figura 4	– Representação dos cromossomos, adaptado de [11].	27
Figura 5	– Fluxo do AG com as modificações inseridas.	32
Figura 6	– Fluxo do AG com as modificações inseridas.	34
Figura 7	– Modelagem do banco de dados para tabela Enade.	35
Figura 8	– Média da proporção de professores doutores para cada configuração de peso.	42
Figura 9	– Média da proporção de professores mestres para cada configuração de peso.	42
Figura 10	– Média da proporção de professores com regime de trabalho parcial ou integral para cada configuração de peso.	43
Figura 11	– Média do número de violações para cada configuração de peso.	44
Figura 12	– Grade horária gerada pelo sispad para o curso de arquitetura e urbanismo do terceiro semestre [57].	73

LISTA DE TABELAS

Tabela 1	–	Composição da nota do CPC [20]	16
Tabela 2	–	Trabalhos relacionados: operadores genéticos e funcionalidades .	24
Tabela 3	–	Trabalhos relacionados: funcionalidades adicionais	25
Tabela 4	–	Representação dos <i>timeslots</i> distribuídos em uma semana	26
Tabela 5	–	<i>Constraints</i> do AG original com seus pesos e limites	29
Tabela 6	–	Conjunto de Pesos para testes iniciais	41
Tabela 7	–	Resultados dos Testes para definição de pesos	45
Tabela 8	–	Resultados dos Testes para definição de pesos para <i>clash</i> e não compartilhado	46
Tabela 9	–	<i>Constraints</i> atuais com seus pesos e limites.....	47
Tabela 10	–	Estatística dos resultados globais.....	48
Tabela 11	–	Estatística dos resultados globais para <i>clash</i> e disciplinas não compartilhadas	48
12		Estatística dos resultados finais por curso	49
Tabela 13	–	Teste de Levene das configurações de pesos testadas.....	66
Tabela 14	–	Teste de Levene dos resultados para definição de pesos.....	67
Tabela 15	–	Teste de Shapiro-Wilk dos resultados para definição de pesos.....	68
Tabela 16	–	Teste de Levene dos resultados finais global	69
Tabela 17	–	Teste de Levene dos resultados finais para <i>clash</i> e não comparti- lhado	70
18		Teste de Levene dos resultados finais para cada curso.....	71

LISTA DE SIGLAS

AG – Algoritmo Genético

CB-CTT – Problema de Alocação de Curso Baseado em Currículo

CI – Conceito Institucional

CPA – Comissão Própria de Avaliação

CPC – Conceito Preliminar de Cursos

EAD – Educação à Distância

ENADE – Exame Nacional de Desempenho dos Estudantes

IDD – Indicador de Diferença entre os Desempenhos Observado e Esperado

IES – Instituição de Ensino Superior

IGC – Índice Geral de Cursos

INEP – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

INPI – Instituto Nacional da Propriedade Industrial

MEC – Ministério da Educação

SINAES – Sistema Nacional de Educação Superior

UPF – Universidade de Passo Fundo

SUMÁRIO

1	INTRODUÇÃO	11
2	REVISÃO DE LITERATURA	14
2.1	AVALIAÇÃO DA EDUCAÇÃO SUPERIOR	14
2.2	ALGORITMOS GENÉTICOS	17
2.2.1	Representação das Soluções	18
2.2.2	Seleção	19
2.2.3	Cruzamento	20
2.2.4	Mutação	21
2.2.5	Aplicações	22
2.3	<i>CURRICULUM-BASED COURSE TIMETABLING PROBLEM</i>	23
2.4	TRABALHOS RELACIONADOS	24
2.5	AG ATUAL: UTILIZADO NA UPF	26
2.5.1	Inicialização	27
2.5.2	Avaliação	28
2.5.3	Crossover e Elitismo	30
2.5.4	Mutação	31
3	MATERIAIS E MÉTODOS	32
3.1	INTRODUÇÃO	32
3.2	BANCO DE DADOS	35
3.3	NOVAS <i>CONSTRAINTS</i> E FUNÇÃO AVALIAÇÃO	36
3.4	METODOLOGIA PARA EXECUÇÃO E ANÁLISE DOS RESULTADOS	37
4	RESULTADOS	40
4.1	CURSOS UTILIZADOS NOS RESULTADOS	40
4.2	DEFINIÇÃO DOS PESOS DAS <i>CONSTRAINTS</i>	41

4.3	RESULTADOS GLOBAIS SUMÁRIO	47
4.4	RESULTADOS POR CURSO	49
4.4.1	Estatística descritiva por curso	52
4.4.2	Análise de variações significativas por curso	53
4.4.3	Análise geral dos resultados obtidos	55
4.5	PUBLICAÇÕES	56
5	CONCLUSÃO	57
	Referências	59
	APÊNDICE A – Teste de Levene para resultados de definição de pesos.....	66
	APÊNDICE B – Teste de Levene para resultados de definição de pesos para quantidade de Clashes e Disciplinas Não Compartilhadas	67
	APÊNDICE C – Teste de Shapiro-Wilk para resultados dos testes de configurações de pesos - Clash e Disciplinas Não Compartilhadas	68
	APÊNDICE D – Teste de Levene para resultados finais - global.....	69
	APÊNDICE E – Teste de Levene para resultados finais - clash e não compartilhado	70
	APÊNDICE F – Teste de Levene para resultados finais.....	71
	ANEXO A – Distribuição gerada pelo SISPAD para um nível de um curso.....	73

1. INTRODUÇÃO

O Problema de Alocação de Curso baseado em Currículo (CB-CTT) é um problema de escalonamento de horários de disciplinas no qual o objetivo é atribuí-las semanalmente a períodos (*timeslots*) e professores, de tal forma que uma série de restrições (*constraints*) sejam cumpridas [1]. As restrições impostas podem ser do tipo *hard* ou *soft*. Uma solução ótima para o problema se considera quando as restrições do tipo *hard* são todas satisfeitas enquanto as do tipo *soft* são minimizadas. Geralmente, as principais restrições do tipo *hard* são limitar um professor para não ministrar duas disciplinas que ocorrem ao mesmo tempo, e limitar disciplinas do mesmo currículo para que não sejam agendadas no mesmo dia e horário. As restrições do tipo *soft* podem ser número mínimo de dias úteis e número máximo de horas de um professor [2].

Este problema é do tipo não polinomial, classificado como NP-Completo [3], e possui uma alta complexidade computacional. Além disso, se aplicado a um número grande de cursos, níveis e disciplinas, o problema pode se tornar ainda mais complicado [4]. Por este motivo, a construção manual de um horário é uma tarefa extremamente difícil, que exige um demorado esforço [5], consumindo vários dias ou semanas de trabalho [6] e com forte probabilidade de que não se satisfaçam todas as *constraints* [5]. Além disso, horários mal elaborados não são apenas inconvenientes, mas podem resultar em perdas significativas de tempo, esforço e dinheiro [7].

Tendo isso em vista, automatizar esse processo acaba sendo de grande valor para uma instituição de ensino, pois torna a grade horária mais eficiente, além de economizar tempo. Com esse intuito, algoritmos de busca de otimização tais como os algoritmos evolucionários são frequentemente utilizados para resolver esse tipo de problema [8], pois conseguem gerar soluções viáveis, sem consumir um tempo polinomial de uma solução exata, com grande eficácia.

Dessa maneira, Instituições de Ensino Superior (IES) podem se beneficiar gerando grades horárias melhores e de maneira mais simples, respeitando a maioria das *constraints*. Além disso, cada IES possui características diferentes, tornando algumas necessidades para resolver o problema de geração da grade horária distintas uma da outra. Uma grade gerada por um algoritmo consegue atender restrições específicas que podem surgir em uma IES [9].

A Universidade de Passo Fundo (UPF) possui um sistema gerador de grade horária desenvolvido com um algoritmo genético (AG). Esse sistema possui algumas especificidades que não são encontradas em outras soluções propostas na literatura [10, 11, 12, 13, 5, 6, 14, 15, 8, 7, 16, 4, 17], como disciplinas compartilhadas, créditos não presenciais e disciplinas fora da grade horária. Algumas das principais *hard constraints* deste AG que

podem ser citadas são disciplinas compartilhadas, um grupo de disciplinas que devem ocorrer no mesmo horário com o mesmo professor; e *clashes*, que significam que um professor não pode ministrar duas disciplinas no mesmo intervalo de tempo, com a exceção das disciplinas compartilhadas. Juntamente, existem as *constraints* do tipo *soft* que controlam a carga horária diária e total de professor, além de controlar qual disciplina cada professor tem prioridade para ministrar e também o número de turnos diferentes que uma turma pode ter em um dia.

No Brasil, o Sistema Nacional de Avaliação de Ensino Superior (SINAES) é o órgão responsável por avaliar as instituições e cursos de ensino superior brasileiras [18]. Essa avaliação é feita com a realização do Exame Nacional de Desempenho dos Estudantes (Enade), que ocorre a cada três anos [19] para cada curso. Junto com a prova, também são computadas outras avaliações, entre elas, o cálculo do Conceito Preliminar de Cursos (CPC) que considera, entre outros elementos, o corpo docente de cada curso, tendo insumos como a titulação e o regime de trabalho dos professores [20].

Na literatura existem soluções de *timetabling* para as IES e instituições de ensino em geral. Muitas delas trabalham com otimização do corpo docente [4, 17, 21], mas sempre em funcionalidades conhecidas e também já implementadas no AG da UPF, como, por exemplo, prioridade de professores para certas disciplinas, indisponibilidade de dias para um professor e controle de horas. Porém, não foram encontradas soluções de *timetabling* para uma IES que considere a avaliação da Instituição, nem no Brasil, com a avaliação realizada pelo SINAES, nem em outra parte do mundo com outro tipo de avaliação similar. Dessa maneira, podemos dizer que essa otimização pode ser considerada um potencial existente não utilizado por essas instituições.

Os atributos utilizados nessa otimização são a proporção de doutores, a proporção de mestres e o regime de trabalho do corpo docente de cada curso que será avaliado. Além disso, como a avaliação ocorre em anos diferentes para cada curso, essa otimização foi aplicada somente aos cursos que participam do Enade no ano indicado, ou seja, a cada ano em que o algoritmo for executado ele irá trabalhar para otimizar os insumos de titulação e regime de trabalho do corpo docente em cursos diferentes.

Nesse contexto, foram realizadas mudanças na estrutura do algoritmo incluindo novas *constraints* que foram criadas em volta de uma análise dos fatores utilizados para o cálculo da nota do CPC. Essas *constraints* passaram por testes para definir seus pesos. Primeiramente foram analisadas algumas combinações de pesos diferentes, sendo definidos os pesos finais através de uma série de execuções utilizadas para obter resultados e compará-los de maneira estatística. Após essa definição, outra bateria de execuções foi realizada a fim de obter os resultados finais, que também foram analisados de forma estatística, comparando as soluções obtidas no algoritmo original com as soluções obtidas no algoritmo modificado.

O objetivo deste trabalho é adaptar o AG da UPF que resolve o CB-CTT, para que ele consiga priorizar a alocação de professores que possuam os atributos necessários para que a nota do Corpo Docente no cálculo do CPC para os cursos que realizam o Enade no ano corrente aumente. Consequentemente, com as alterações, é esperado que a nota do conceito dos cursos otimizados aumente na avaliação realizada, melhorando também a avaliação geral da IES.

Este trabalho está dividido em seis capítulos, sendo o primeiro esta introdução. O segundo capítulo apresenta a revisão de literatura abordando os principais conceitos para entender o desenvolvimento do projeto, sendo eles a avaliação da educação superior, algoritmos genéticos e o CB-CTT. Além disso, este capítulo ainda aborda os trabalhos relacionados e explica a estrutura e funcionamento da aplicação existente na UPF antes das modificações realizadas neste trabalho. O terceiro capítulo, materiais e métodos, descreve o planejamento e desenvolvimento do projeto, mostrando todas as partes que precisaram ser alteradas no AG. O quarto capítulo trás os resultados obtidos após a execução do algoritmo com as mudanças já realizadas fazendo uma comparação com os resultados do algoritmo original. Por fim, o quinto capítulo mostra a conclusão.

Decorrente deste trabalho, além da dissertação e da modificação no algoritmo utilizado para a geração de grade horária da UPF, foram escritos dois artigos: um que foi submetido em 2023 e que aguarda a decisão final; e outro em 2024 que está em processo de submissão. Também foi realizado registro do *software* desenvolvido no Instituto Nacional da Propriedade Industrial (INPI). O algoritmo já está gerando grades horárias otimizadas para a avaliação realizada pelo SINAES na UPF.

2. REVISÃO DE LITERATURA

Neste capítulo são abordados os principais conceitos envolvidos no desenvolvimento deste trabalho. Primeiramente é descrito sobre a avaliação da educação superior no Brasil, com ênfase nos indicadores de avaliação dos cursos de graduação. Após, são introduzidos conceitos básicos sobre algoritmos genéticos, como sua estrutura geral e algumas de suas aplicações. Em uma nova seção é apresentado o problema de geração de grade horária nas instituições de ensino superior, explicando sua definição, assim como abordagens utilizadas. Por fim são mostrados os trabalhos relacionados.

2.1 AVALIAÇÃO DA EDUCAÇÃO SUPERIOR

O Sistema Nacional de Avaliações da Educação Superior (SINAES) é o órgão responsável pela regulação e avaliação da educação superior no Brasil [18]. Ele foi criado através da Lei nº 10.861, de 14 de abril de 2004, e de acordo com a mesma, tem por finalidades a melhoria da qualidade da educação superior, orientação da expansão da sua oferta, aumento permanente da sua eficácia institucional e efetividade acadêmica e social e a promoção do aprofundamento dos compromissos e responsabilidades sociais das instituições de educação superior [22]. O processo para a avaliação leva em consideração aspectos como ensino, pesquisa, extensão, responsabilidade social, gestão da instituição e corpo docente [23]. A avaliação é composta pela Avaliação Institucional da Instituição de Ensino Superior (IES), Avaliação do Desempenho dos Estudantes e pela Avaliação dos Cursos de Graduação [18].

A Avaliação Institucional utiliza-se de dois processos, sendo eles a autoavaliação conduzida pela Comissão Própria de Avaliação (CPA) de cada instituição e a avaliação externa, realizada pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e pelo Ministério da Educação (MEC) [18]. A autoavaliação visa analisar o funcionamento da Instituição por meio da verificação da coerência entre o que está previsto no Plano de Desenvolvimento Institucional e as Políticas Institucionais. A avaliação externa *in loco* é utilizada para gerar o Conceito Institucional (CI) [24]. Esse conceito é calculado para verificar se a IES está apta para que os estudantes consigam se desenvolver no meio acadêmico [25]. O conceito institucional pode variar de 1 a 5 na qual a nota 3 é a mínima exigida, a qual considera a qualidade da IES como satisfatória para o funcionamento [26].

A Avaliação do Desempenho dos Estudantes é feita pela aplicação das provas do Exame Nacional de Desempenho de Estudantes (Enade) além da análise dos resultados do questionário do estudante [24]. O Enade é um exame aplicado pelo INEP desde 2004. Nele avalia-se o rendimento dos estudantes concluintes dos cursos de graduação, com

questões de formação geral, medindo o desenvolvimento de competências e habilidades necessárias e o nível de atualização dos estudantes com relação à realidade brasileira e mundial, e questões de formação específica que medem o conhecimento em conteúdos previstos nas diretrizes curriculares dos cursos [19]. O questionário do estudante tem o objetivo coletar informações que permitam caracterizar o perfil dos estudantes e o contexto de seus processos formativos [27].

A Avaliação dos Cursos é feita através da medição de dois índices: o Conceito Preliminar de Curso (CPC) e o Índice Geral de Cursos (IGC). O IGC é obtido através da média ponderada das notas dos cursos de graduação e pós-graduação de cada instituição, sintetizando num único indicador, a qualidade de todos os cursos de graduação, mestrado e doutorado da mesma instituição de ensino, sendo divulgado anualmente, pelo MEC [18]. Este índice é divulgado após o resultado do Enade [28].

O CPC, por sua vez é um indicador de qualidade que, a partir de alguns pontos, consegue avaliar a qualidade dos cursos de graduação. O conceito se baseia em oito componentes agrupados em quatro dimensões. Na nota técnica do cálculo do CPC [20], esses componentes são indicados a partir dos seguintes itens:

- Desempenho dos Estudantes: Mensurado a partir das notas dos estudantes concluintes no Exame Nacional de Desempenho dos Estudantes (Enade);
- Valor agregado pelo processo formativo oferecido pelo curso: Mensurado a partir dos valores do Indicador de Diferença entre os Desempenhos Observado e Esperado (IDD);
- Percepção Discente sobre as Condições do Processo Formativo: Mensurado através da aplicação do questionário do estudante. Coleta informações da organização didático-pedagógica, infraestrutura e instalações físicas da instituição, além das oportunidades oferecidas para ampliação da formação acadêmica de um estudante.
- Corpo Docente: Baseado em informações obtidas a partir do Censo da Educação Superior, referente ao ano de aplicação do Enade, sobre a titulação e o regime de trabalho dos docentes vinculados aos cursos avaliados.

A nota da titulação do corpo docente para o cálculo do CPC é obtida a partir da proporção de mestres e doutores sobre a quantidade total de professores no curso. A nota dos professores pelo seu regime de trabalho é calculada de forma semelhante, a partir da proporção de docentes com regime de trabalho parcial ou integral (não horistas), sobre a quantidade total de professores no curso. A Equação 1 adaptada de [20] demonstra uma fórmula genérica para o cálculo das notas de titulação e regime de trabalho.

$$P_c = \frac{N_c}{T_c} \quad (1)$$

onde:

- P_c é a proporção de docentes do curso de graduação c que atendem aos requisitos impostos pelo cálculo;
- N_c é o número de docentes do curso de graduação c que atingem aos requisitos impostos pelo cálculo;
- T_c é o número total de docentes vinculados ao curso de graduação c .

Para o cálculo de proporção de mestres, por exemplo, divide-se o número de professores com titulação igual ou superior a mestre pelo número de professores total do curso. De maneira análoga aplica-se para a proporção de doutores, dividindo o número de professores com grau de doutor pelo número de professores total do curso e finalmente, para o cálculo de regime de trabalho será dividido o número de professores com regime parcial ou integral pelo número de professores total do curso.

Todos os resultados, tanto dos cálculos para proporção de mestres e doutores quanto para o regime de trabalho, passam por uma padronização para que a nota final fique em uma escala de zero (0) a cinco (5) [20]. Após a padronização de todos os elementos que compõe o cálculo do CPC, o conceito então é formado de acordo com o peso de cada uma das dimensões avaliadas (Tabela 1).

Tabela 1. Composição da nota do CPC [20]

Dimensão	Componente	Peso	
Desempenho dos Estudantes	Nota dos Concluintes no Enade (NC)	20.00%	
Valor agregado pelo processo formativo oferecido pelo curso	Nota do Indicador de Diferença entre os Desempenhos Observado e Esperado (NIDD)	35.00%	
Corpo Docente	Nota de Proporção de Mestres (NM)	7,5%	30%
	Nota de Proporção de Doutores (ND)	15%	
	Nota de Regime de Trabalho (NR)	7,5%	
Percepção Discente sobre as Condições do Processo Formativo	Nota referente à organização didático-pedagógica (NO)	7,5%	15%
	Nota referente à infraestrutura e Instalações Físicas (NF)	5,0%	
	Nota referente às oportunidades de ampliação da formação acadêmica e profissional (NA)	2,5%	

Como mostrado na Tabela 1, o peso do corpo docente para a geração da nota final do CPC é de 30%, segundo maior peso dentre as 4 dimensões avaliadas, tendo inclusive maior peso que a nota do Enade realizada pelos estudantes concluintes do curso. Também é possível ver que dentro da dimensão do corpo docente o peso para a proporção de doutores é o dobro do peso para regime de trabalho e proporção de mestres. Além do mais, como um doutor, pela forma de cálculo, também é um mestre, um professor doutor acaba sendo muito importante para o aumento do conceito calculado, já que ele conta duas vezes em componentes que somados chegam a 22,5% dos 30% de peso totais para o cálculo da nota.

2.2 ALGORITMOS GENÉTICOS

Os algoritmos genéticos (AGs) foram inventados por John Holland na década de 1960. O objetivo original de Holland não era projetar algoritmos para resolver problemas específicos, mas sim estudar o fenômeno da adaptação como ocorre na natureza e desenvolver maneiras pelas quais os mecanismos de adaptação natural pudessem ser importados em sistemas de computador [29].

Os AGs são um tipo de algoritmo, dentre os algoritmos evolucionários. De acordo com Back [30] eles são baseados na teoria da evolução, proposta por Charles Darwin, que propõe a seleção natural como uma mudança adaptativa das espécies, favorecendo a sobrevivência dos indivíduos mais adaptados ao ambiente, os quais passam seus genes para as próximas gerações.

Os AGs em geral possuem um certo número de etapas básicas, que são seguidas para gerar uma solução. Eles funcionam de forma a iniciar primeiramente uma população de teste, de forma aleatória e uma função que avalia a partir de alguns critérios, o *fitness* ou aptidão de cada indivíduo [31]. Essa função de aptidão é muito importante para saber quais indivíduos devem ser levados à próxima geração, uma vez que os indivíduos com maior avaliação geralmente são escolhidos para gerar descendentes [32]. O processo descrito é iterado diversas vezes até que se obtenha uma solução desejável para o problema. A Figura 1 adaptada de [33] destaca as etapas básicas do funcionamento de um algoritmo genético.

```
1: Initialize population with n individuals
2: Evaluate the n individuals
3: repeat
4:   Generate new individuals from the current generation
5:   Evaluate the new generated individuals;
6:   Replacing current generation individuals with the new generation
7: until criteria satisfied
8: Output Best individual generated
```

Figura 1. Pseudocódigo de Algoritmo Genético adaptado de [33].

No meio do processamento, antes de passar pela função de avaliação, a etapa de gerar novos indivíduos ainda pode ser dividida em outras partes, pois o algoritmo passa por operações genéticas que modificam a geração. Essas modificações servem para que os indivíduos consigam evoluir e gerar melhores soluções. A forma mais simples de um algoritmo genético envolve três tipos de operadores: seleção, cruzamento e mutação [29].

2.2.1 Representação das Soluções

Para iniciar a construção de um algoritmo genético um dos primeiros passos é a definição de como a sua solução final será representada. Essa etapa serve para que o mundo real seja vinculado com o algoritmo genético. Dessa forma é necessário que haja a abstração de algumas características da solução do mundo real, criando um contexto de problema bem definido e tangível dentro do qual possíveis soluções possam existir, ser manipuladas por um computador, avaliadas e armazenadas [32].

Além disso, a eficiência de uma representação também está relacionada aos operadores de busca aplicados nessa representação. De fato, ao definir uma representação, deve-se ter em mente como a solução será avaliada e como os operadores de busca trabalharão. Muitas representações alternativas podem existir para um determinado problema, mas uma representação viável deve ter algumas características [33].

Primeiramente, a representação escolhida para o problema, deve possuir completude, ou seja, todas as soluções que podem ser atribuídas ao problema devem ter uma maneira de serem representadas. Além disso, todas as soluções devem possuir um caminho de busca a partir de qualquer outra solução no espaço de busca. Por último, a solução deve ser de fácil manipulação pelos operadores de busca [33].

Existem algumas representações clássicas que são comumente usadas para resolver uma grande variedade de problemas de otimização. De acordo com sua estrutura, existem duas classes principais de representações, as lineares e as não lineares [33].

As representações lineares podem ser vistas como sequências de símbolos de determinado alfabeto [33]. A maioria dos AGs usam cadeias de bits de ordem e comprimento fixo para codificar soluções [29]. Esse tipo de solução pode ser utilizado onde as variáveis de decisão apontam a presença ou ausência de um elemento ou uma decisão, que pode ser do tipo sim ou não [33].

Essa combinação de características podem ser, em um problema combinatorial, a escolha de se determinado item fará ou não parte da solução. Um exemplo clássico de algoritmo que pode utilizar uma solução binária é o problema da mochila binária [32]. A representação desse problema poderia ser uma cadeia binária, onde o índice do elemento representaria um item e o seu conteúdo representaria se ele será utilizado na resolução (valor 1) ou não (valor 0) [32].

Entretanto, alguns problemas podem necessitar mais do que apenas 0 ou 1 para a solução. As codificações binárias não são naturais e muitos problemas são difíceis de se manipular [29]. É o caso de problemas de atribuição. Suponha que um conjunto de k tarefas seja atribuído a m agentes para maximizar o lucro total. Uma tarefa pode ser atribuída a qualquer agente. Uma codificação clássica para esta classe de problemas pode ser baseada em um vetor inteiros onde cada índice representa uma tarefa e o valor representa o

agente atribuído. Muitos problemas de otimização da vida real, como alocação de recursos, podem ser reduzidos a problemas de atribuição [33].

Outro tipo de problema que pode ser reduzido a uma sequência de inteiros, são problemas de permutação. Muitos problemas de sequenciamento, planejamento e roteamento são considerados como problemas de permutação [33]. A diferença para os outros problemas é que enquanto uma representação binária, ou inteira simples, permite que os números ocorram mais de uma vez, uma permutação não permite. Um problema clássico que pode ser resolvido dessa forma é o problema do caixeiro viajante [32]. O objetivo do problema é descobrir a ordem de cidades a se visitar, a qual possui o mínimo de custo para se locomover entre as cidades. Como o desejo é saber a ordem das cidades, uma maneira de se apresentar a solução para o problema é uma sequência de inteiros, onde cada número inteiro representa uma cidade [32].

Por fim, ainda existem uma variedade de problemas de otimização em ciências da engenharia que consistem em encontrar os melhores parâmetros no projeto de um determinado componente. Esta classe de problemas é conhecida como problemas de otimização de parâmetros. Alguns parâmetros podem estar associados a valores reais enquanto outros estão associados a valores discretos. Assim, uma solução pode ser representada por um vetor de valores mistos, os quais podem ser do tipo real ou discreto [33].

Entretanto, alguns problemas que exigem estruturas complexas, não se encaixam em nenhuma das soluções apresentadas. Nesses casos, a melhor forma para encontrar uma representação para eles, é partir para representações não lineares. Uma das representações mais utilizadas são as árvores. Elas geralmente são utilizadas em problemas para otimização estruturada hierárquica [33]. Elas têm várias vantagens, incluindo o fato de permitirem que o espaço de busca seja aberto. Porém, é preciso ter o cuidado para que as árvores não cresçam de forma descontrolada, impedindo a formação de soluções candidatas mais estruturadas e hierárquicas [29].

2.2.2 Seleção

Este operador seleciona quais cromossomos na população serão utilizados para cruzamento. Também define quantos descendentes cada cromossomo selecionado gerará [33]. O principal propósito da seleção é enfatizar os indivíduos mais aptos na população na esperança de que seus descendentes obtenham um *fitness* ainda maior, ou seja, quanto mais adaptado estiver um indivíduo de acordo com a função de avaliação, mais vezes ele será selecionado para se reproduzir [29].

Contudo, os piores indivíduos não devem ser descartados da seleção, pois podem produzir algum material genético que seja eficiente [33]. Quando é realizada uma seleção muito forte, significa que indivíduos altamente aptos abaixo do ideal assumirão o controle

da população, reduzindo a diversidade necessária para mudanças e progresso. Entretanto, uma seleção muito fraca resultará em uma evolução muito lenta [29].

Há vários métodos de seleção aplicados em AGs. Uma abordagem comum é a seleção proporcional por *fitness*, onde a probabilidade de um indivíduo qualquer ser selecionado para gerar descendentes depende de seu valor absoluto de aptidão comparado aos valores absolutos de aptidão do resto da população. Esse é um exemplo de seleção que pode se tornar muito forte, fazendo com que indivíduos proeminentes tomem conta de toda a população muito rapidamente. Isso tende a direcionar o processo de busca e torna menos provável que o algoritmo faça uma busca completa no espaço de soluções possíveis, onde podem existir melhores soluções [32].

Vendo essa desvantagem nesse tipo de seleção, outros métodos tentam contornar esses problemas fazendo com que a probabilidade não seja proporcionalmente de acordo com o *fitness* [32]. Um exemplo disso é a seleção por ranqueamento. Nesse tipo de seleção os indivíduos não são escolhidos pelo seu valor absoluto de aptidão, mas sim através do seu ranking. Os indivíduos são ordenados por ordem de seu *fitness* e a chance de cada um ser escolhido será por sua posição no ranking e não pelo seu real valor de aptidão [30].

Outro tipo de seleção que tenta minimizar os pontos fracos da seleção proporcional é a seleção por torneio. A seleção por torneio consiste em selecionar aleatoriamente k indivíduos. Um torneio é então aplicado aos k membros do grupo para selecionar o melhor [33]. Quanto maior a quantidade de indivíduos no torneio, maior a chance desses indivíduos escolhidos possuírem um *fitness* acima da média da população [32].

2.2.3 Cruzamento

O papel dos operadores de cruzamento é herdar algumas características dos dois pais para gerar os descendentes [33]. O cruzamento de ponto único é a forma mais simples: uma única posição de cruzamento é escolhida aleatoriamente e as partes de dois pais após a posição de cruzamento são trocadas para formar dois descendentes [29]. Uma representação do cruzamento em ponto único é mostrada na Figura 2A. Essa técnica pode ser expandida para múltiplos pontos como é mostrado na Figura 2B. O cruzamento de n pontos seleciona mais de um ponto de cruzamento para criar os descendentes [34]. Os dois descendentes são gerados alternando os genes dos pais de acordo com as duas divisões geradas, ou seja, se os dois indivíduos são ABCDE e FGHIJ e serem divididos em dois pontos A|BCD|E e F|GHI|J, os filhos gerados serão A|GHI|E e F|BCD|J [33].

Como é possível ver na Figura 2, tanto a técnica de cruzamento com um ponto quanto a técnica com dois pontos, funcionam de forma semelhante seguindo o mesmo padrão. Dessa forma é possível aplicar essa técnica dividindo os cromossomos em quantos

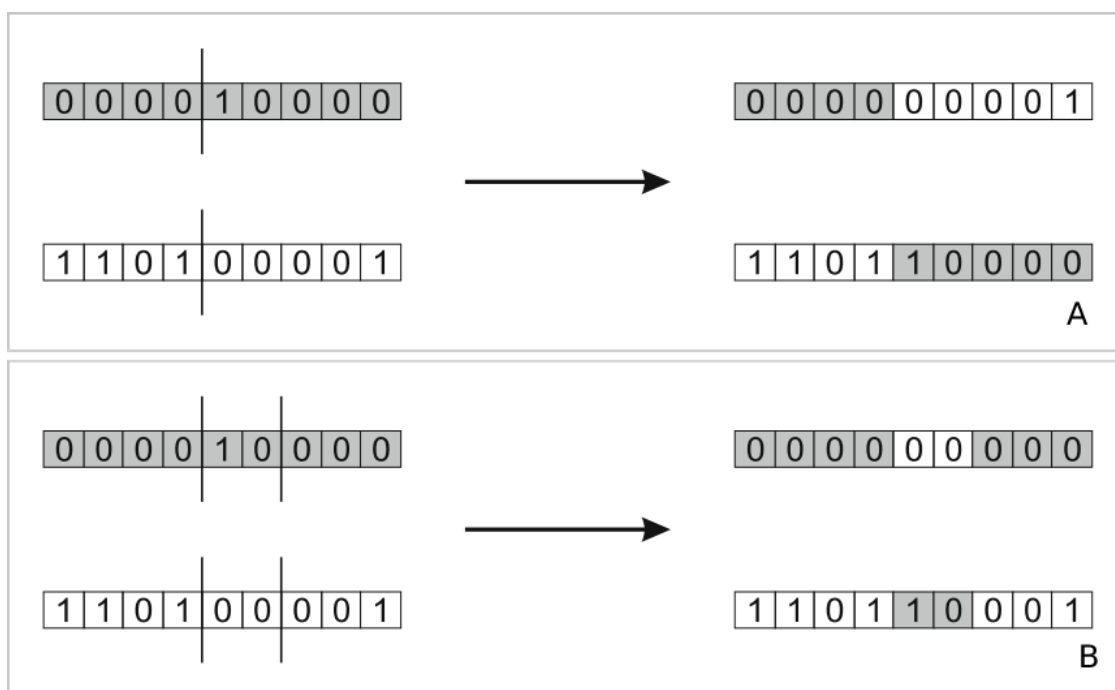


Figura 2. Na parte superior cruzamento de ponto único e embaixo cruzamento de 2 pontos [32].

pontos forem necessários. A aplicação continuará a mesma, alternando os genes do primeiro para o segundo pai a cada divisão.

Além desses dois tipos, outro cruzamento muito utilizado é o cruzamento uniforme, o qual fornece a uniformidade na combinação dos genes de ambos os pais [34]. Diferentemente dos dois métodos abordados anteriormente, que dividem os pais em pequenas seções contíguas de genes, este método trabalha com cada um dos genes independentemente [32]. Ele realiza as operações sorteando um número real aleatório entre 0 e 1. Este número sorteado é que definirá se aquele gene específico será proveniente do primeiro ou do segundo pai [34]. O processo é repetido para cada gene, até gerar um novo cromossomo completo que será o primeiro descendente. O segundo descendente gerado é o mapeamento inverso do primeiro [32].

2.2.4 Mutação

Utilizar apenas o operador de cruzamento para produzir uma nova geração pode fazer com que o AG fique estagnado em uma solução que não é a melhor possível [33]. Isso é muito comum quando se está lidando com problemas multimodais. Estes são problemas em que há um número de pontos que são melhores do que todas as suas soluções vizinhas. Esses pontos são chamados ótimas locais e o ponto mais alto deles é denominado como ótimo global (Figura 3). Um problema em que existe apenas um ótimo local é conhecido como unimodal [32].

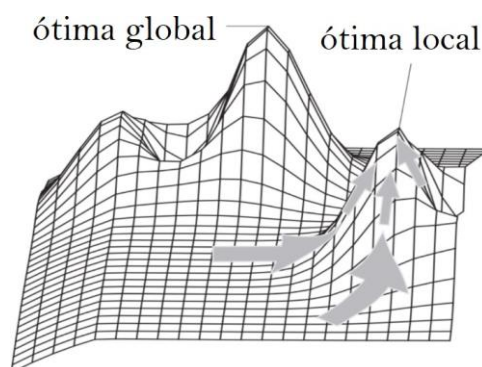


Figura 3. Representação de uma ótina local e ótina global de uma função [35]

Como é possível ver, a Figura 3 mostra uma matriz, que representa o conjunto de soluções possíveis para determinado problema. O relevo presente nesta matriz representa o espaço de busca. A solução boa é a solução naquele local. Desse modo, quando um algoritmo genético encontra o ponto que está definido como *local optimum*, analisando que todos os pontos próximos ao local tem uma solução considerada pior, o algoritmo pode ficar estagnado neste ponto não encontrando outros pontos em lugares diversos da matriz, que poderiam possuir soluções melhores.

O operador de mutação é muito utilizado para minimizar este problema, encorajando diversidade na população ao fazer alterações na população de indivíduos já existentes [33]. O operador de mutação aplica as mudanças aleatoriamente em um ou mais genes do indivíduo para gerar seu descendente, assim, criando novas soluções adaptativas e evitando as ótimas locais [33]. A operação de mutação gera a descendência alterando aleatoriamente um ou vários bits de um indivíduo. Um descendente pode então possuir características diferentes de seus pais e evitar a busca local no espaço de busca, aumentando a probabilidade de encontrar o ótimo global [36].

2.2.5 Aplicações

Os AGs e algoritmos evolucionários em geral, assim como várias outras meta-heurísticas possuem uma grande aplicabilidade em problemas reais de otimização em diversas áreas onde o escopo de solução é grande para se encontrar a melhor solução com um método iterativo comum. Entre essas áreas, é comum encontrar problemas na área científica e industrial [33].

A área de otimização possui diversos problemas em que AGs são utilizados [37, 38, 39]. Um problema de otimização muito comum de ser encontrado é o problema de escalonamento. Esse problema é um processo de tomada de decisão complexo que é usado regularmente em muitas indústrias de manufatura e serviços [40]. O conceito principal de um problema de escalonamento pode ser descrito como um problema em que um

número de tarefas precisa ser atribuído a um conjunto de recursos ou máquinas a um custo total mínimo [41]. Na literatura, diversos problemas são encontrados utilizando AGs para resolvê-los [42, 43, 44].

Uma subclasse de problemas de alocação é o problema de geração de grade horárias (*timetabling*). Esse problema consiste em atribuir um número de eventos (exames, cursos, reuniões) em um número limitado de intervalos de tempo [45]. Esse problema também possui diversas aplicações com abordagens em AGs [12, 13, 4].

2.3 CURRICULUM-BASED COURSE TIMETABLING PROBLEM

O *Curriculum-Based Course Timetabling* (CB-CTT) é um problema de atribuição de disciplinas, onde é necessário alocar professores para ministrar determinada disciplina em determinado horário, considerando restrições exigidas pela instituição [4]. Esse problema é imprescindível para as instituições de ensino superior e automatizá-lo é de grande importância visto sua alta complexidade, já que é um problema considerado NP-Completo [3].

A geração da grade horária em uma instituição, por vezes é realizada de forma manual, tornando-se um processo demorado e sem garantias de que as suas restrições sejam completamente resolvidas. As restrições (*constraints*) de um problema de *timetabling* podem ser de dois tipos, *hard* ou *soft*. O objetivo é satisfazer as *hard constraints* e minimizar as falhas nas *soft constraints* [46].

Segundo Bettinelli *et al.* os *hard constraints* básicos de um CB-CTT são [2]:

1. Um professor não pode ministrar duas ou mais disciplinas ao mesmo tempo (*clash*);
2. Um professor não pode ministrar uma disciplina em um horário que não possui disponibilidade;
3. Todos oferecimentos de disciplinas devem ser alocados e atribuídos à períodos semanais distintos.

Ainda segundo Bettinelli *et al.*, algumas *soft constraints* podem ser [2]:

1. Selecionar um professor preferencial para uma disciplina;
2. Completar o número de horas de um professor de acordo com seu regime de trabalho;
3. Minimizar o número de dias na semana com aula aos alunos de um curso.

Essas *constraints* são importantes para que se consiga medir a eficiência de uma solução gerada pelo algoritmo, pois elas representam uma avaliação determinando se o

requisito é válido ou não para a solução [32]. Com o uso de algoritmos de busca estocásticos, tais como os algoritmos evolucionários, essas restrições são utilizadas como forma de modelar o problema com o uso de funções objetivo [32]. Esse processo de satisfazer as restrições resulta numa boa solução aproximada do problemas sem consumir um tempo polinomial de execução que as soluções exatas demandariam.

2.4 TRABALHOS RELACIONADOS

O problema da geração de grade horária vem sendo estudado há muito tempo e sendo desenvolvido com diversas abordagens como as meta-heurísticas [46], programação inteira [21] e lógica difusa [47]. Entre as meta-heurísticas, colônias artificiais de abelhas [48, 49], colônias de formigas [50], algoritmos meméticos [14, 45], otimização por enxame de partículas [51, 52], algoritmos híbridos e algoritmos genéticos [17, 4, 13, 16], são métodos normalmente encontrados na literatura. De acordo com [46], AGs estão entre as meta-heurísticas mais populares para a resolução de problemas de *timetabling* [53, 8, 15].

A Tabela 2 traz algumas soluções que foram desenvolvidas com a abordagem de AGs. Nela é possível ver quais tipos de operadores genéticos e funcionalidades foram utilizados em cada uma das soluções propostas.

Tabela 2. Trabalhos relacionados: operadores genéticos e funcionalidades

Trabalho	Inicialização	Seleção	Recombinação	Mutação	Elitismo	Ano
[10]	Aleatória	Roleta	PMX	Swap	Sim	2003
[11]	Aleatória	Roleta	Ponto único	Swap	Não	2008
[12]	Heurística	Torneio	-	Swap	Sim	2010
[13]	Heurística	Truncamento	-	Dirigida	Sim	2010
[5]	Heurística	Torneio	Ponto único	Troca de heurística	Não	2010
[6]	Aleatória	Torneio, Rank	Múltiplos	Troca aleatória	Sim	2011
[14]	GRASP	Truncamento	Múltiplos	Troca aleatória	Sim	2011
[15]	Busca local	Torneio	Uniforme	Neighborhood move	Sim	2014
[8]	Busca local	Torneio	Uniforme	Não especificado	Sim	2015
[7]	Heurística	Truncamento	Utilization	Targeted	Sim	2016
[16]	Aleatória	Aleatória	OX	Swap	Não	2017
[4]	Heurística	Roleta	Ponto múltiplo	Múltiplos	Sim	2020
[17]	Aleatória	Roleta	Ponto Único	Troca Aleatória	Não	2021
[54]	Busca Local	Estado Estacionário	Ponto Único	Múltiplos	Não	2022
[55]	Aleatória	Torneio	Ponto Único	Troca Aleatória	Não	2023

Na inicialização, Abdelhalim e Khayat [7] introduziram ao AG heurísticas para geração de boas soluções iniciais para a resolução do problema. Gozali *et al.* [4] utilizam um método guloso para inicialização. Em sua pesquisa, Nguyen *et al.* [14] utiliza GRASP, uma combinação de solução aleatória gulosa com busca local iterativa [56]. Outros métodos que utilizam busca local na inicialização são encontrados em [8, 15]. Ainda alguns trabalhos realizaram uma inicialização aleatória como em [17]. Ainda Borges [10] utiliza uma abordagem aleatória porém com controle de restrições de carga horária semanal de disciplinas e

professores além da não existência de colisão de horários de professores para os diferentes períodos, de forma semelhante ao que foi desenvolvido no AG atual da UPF.

Nos operadores genéticos várias abordagens diferentes foram utilizadas como a *truncation selection*, uma estratégia de seleção que consiste em ordenar indivíduos pelo *fitness* e selecionar alguma proporção dos melhores utilizada por [7, 14]. No cruzamento, uma versão modificada do OX (*order*) *crossover* é utilizada em [16], sendo esse um operador de permutação que recombina indivíduos, trocando um intervalo de alelos consecutivos e preenchendo os valores restantes na ordem em que apareceram anteriormente no pai. A exploração do operador de recombinação (um ponto, dois pontos ou uniforme) pode ser encontrada em [14, 6]. Para a mutação, a maioria dos trabalhos utiliza algum tipo de mutação por swap [12, 11]. Gozali *et al.* [4] utiliza três métodos distintos: *Moving*, que move uma disciplina que viole uma restrição para um *timeslot* não utilizado, *swapping* trocando duas disciplinas que violem alguma restrição e *comparing*, trocando uma disciplina que viola uma restrição para outro *timeslot* aleatório. Já Suyanto [13] utiliza mutação dirigida, que altera disciplinas com algum tipo de conflito para horários diferentes.

Além dos operadores genéticos, foi realizada uma comparação entre os trabalhos selecionados, para saber se algum deles implementa alguma funcionalidade das que já existem no algoritmo genético da UPF, além também da funcionalidade implementada neste trabalho, que é a otimização do corpo docente para a avaliação dos cursos realizada pelo SINAES. Como a avaliação dos cursos do SINAES é algo muito específico apenas para universidades do Brasil, foi considerada qualquer tipo de otimização para melhora de avaliação de uma IES. A comparação é mostrada na Tabela 3.

Tabela 3. Trabalhos relacionados: funcionalidades adicionais

Trabalho	Disciplinas compartilhadas	Disciplinas EAD ou fora de horário	Múltiplas turmas	Fechamento de grade sem professor	Otimização para avaliação da IES
[10]	Não	Não	Não	Não	Não
[11]	Não	Não	Não	Não	Não
[12]	Não	Não	Não	Não	Não
[13]	Não	Não	Não	Não	Não
[5]	Não	Não	Não	Não	Não
[6]	Não	Não	Não	Não	Não
[14]	Não	Não	Não	Não	Não
[15]	Não	Não	Não	Não	Não
[8]	Não	Não	Não	Não	Não
[7]	Sim	Não	Não	Não	Não
[16]	Não	Não	Não	Não	Não
[4]	Não	Não	Não	Não	Não
[17]	Não	Não	Não	Não	Não
[54]	Não	Não	Não	Não	Não
[55]	Não	Não	Não	Não	Não

Dentre os trabalhos selecionados, nenhum possuía disciplinas na modalidade de educação à distância (EAD) ou que não devam ser consideradas na grade final do curso. Além disso, não foram encontradas implementações de múltiplas turmas, nem mesmo a possibilidade de fechar a grade da disciplina sem um professor. Apenas o trabalho descrito

em [7] considera a ocorrência de disciplinas compartilhadas, porém de forma superficial. Em relação à otimização para avaliação da IES, também não foi encontrado em nenhum trabalho.

Isso mostra que o AG proposto pela universidade já propõe inovações nesse problema, que suprem algumas necessidades que podem ser encontradas dentro do contexto de uma universidade. Além disso, a implementação proposta neste trabalho traz mais uma funcionalidade não encontrada em literatura, promovendo a adaptação do algoritmo para otimizar a avaliação de uma IES no Brasil com a adaptação referente ao cálculo da nota do corpo docente.

2.5 AG ATUAL: UTILIZADO NA UPF

O algoritmo está escrito em linguagem C++, com banco de dados MySQL (MariaDB). Nesta base há 17 tabelas utilizadas pelo algoritmo genético, sendo 16 tabelas para entrada de dados, e uma para armazenar os resultados obtidos pela solução. A saída é gerada em uma tabela chamada *timeslotresult*, a qual são armazenados todos os *timeslots* ou horários da grade disponíveis, com o professor e a disciplina alocados nele. Esses *slots* são compostos em 64 partes, as quais representam cada dia de segunda a sexta que contém quatro períodos pela manhã, à tarde e à noite. Além disso, há quatro períodos no sábado de manhã totalizando 64 períodos semanais. A Tabela 4 mostra cada um dos horários de cada dia da semana e o seu respectivo *timeslot* representado.

Tabela 4. Representação dos *timeslots* distribuídos em uma semana

Timeslot						
Dia	1	2	3	4	5	6
	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira	Sábado
08:00	0	4	8	12	16	20
08:50	1	5	9	13	17	21
09:55	2	6	10	14	18	22
10:45	3	7	11	15	19	23
14:00	24	28	32	36	40	
14:50	25	29	33	37	41	
15:55	26	30	34	38	42	
16:40	27	31	35	39	43	
19:20	44	48	52	56	60	
20:50	45	49	53	57	61	
21:05	46	50	54	58	62	
21:50	47	51	55	59	63	

Além do código em C++ existe um *script* escrito em linguagem python que deve ser executado antes da execução do algoritmo para validar se a base de dados está correta. Para a visualização da solução ainda existe um sistema *web* [57] que recebe os dados

gerados da solução e projeta a visualização de forma gráfica, com possibilidade de ajustes manuais (Anexo A).

A estrutura de um indivíduo está organizada como uma lista representando um indivíduo da população, ou um cromossomo em genética. Cada posição dessa lista representa um curso e possui uma estrutura em segunda dimensão na qual representa um alelo em genética, armazenando cada nível do curso. Por fim, cada nível de um curso ainda possui outra dimensão que representa os horários em que cada disciplina será alocada. Cada um desses períodos de horário é chamado de *slot* e sua estrutura é uma tupla do tipo <professor, disciplina>, ou seja, é uma estrutura que vincula um professor juntamente com uma disciplina. Uma representação do cromossomo é apresentada na Figura 4.

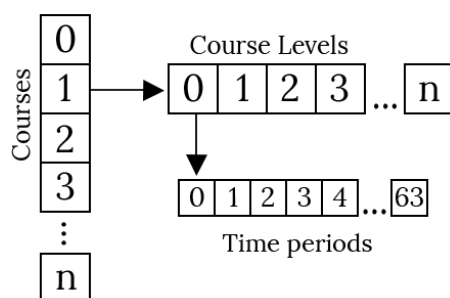


Figura 4. Representação dos cromossomos, adaptado de [11]

O código do algoritmo genético está dividido em funções, uma para inicialização e mais uma para cada operador genético implementado, sendo eles a mutação e cruzamento, além de uma função de avaliação, para calcular o *fitness* da população.

2.5.1 Inicialização

Na inicialização são alocados os professores e as disciplinas em cada *slot*. Para isto a alocação é feita seguindo uma ordem. Primeiro os cursos são iterados e então dentro de cada curso todos os oferecimentos das disciplinas deste curso são alocados, um após o outro. As alocações são realizadas de forma aleatória, porém diferente de [17], nem todos os valores são inicializados dessa forma. Na escolha de professores são controlados apenas os possíveis conflitos, ou seja, se uma disciplina possuir mais de um grupo, todos ocorrendo no mesmo horário um professor não será alocado de forma a ministrar mais de um grupo desta disciplina.

De forma similar a Gozali *et al.* [4], uma porcentagem da população possui inicialização por heurística, onde o professor selecionado, tende a ser o professor com a maior diferença de horas entre as já alocadas e as que deveriam ser alocadas entre todos os professores disponíveis para aquela disciplina. Além disso, para disciplinas compartilhadas, foram adotadas heurísticas para que os compartilhamentos sejam alocados de forma mais otimizada. Desse modo, as disciplinas compartilhadas são alocadas todas no mesmo horá-

rio com o mesmo professor, de forma que na primeira ocorrência da disciplina é decidido o professor e o horário em que ela ocorre e as demais ocorrências apenas mantêm o que foi escolhido. As heurísticas são aplicadas apenas em parte da população, pois embora a qualidade média das soluções melhore com o uso da inicialização heurística, é a diversidade de soluções aleatórias que leva a população às melhores soluções [58].

Os *slots* são escolhidos de forma similar aos professores, com o cuidado de que não haja conflito de horários entre as aulas de um professor. Se um conflito for encontrado ele aloca em outro *slot* disponível.

2.5.2 Avaliação

Na avaliação é calculado o *fitness* de cada indivíduo gerado. Esse cálculo é feito através de uma fórmula, que usa com base as *constraints* definidas para o algoritmo. Cada *constraint* possui um peso definido. As *hard constraints* possuem maior peso, pois elas precisam ter uma prioridade a serem resolvidas se comparadas com as *soft constraints*. A fórmula para o cálculo do *fitness* do indivíduo é representada na Equação 2. Ela se baseia na atribuição de pesos para cada *constraint* [2]. O uso desse tipo de funções de avaliação com pesos é uma estratégia utilizada em [45, 14]. Com o objetivo de evitar a inversão de prioridades, um limite é definido para cada uma das *constraints* empregadas de forma similar a [7, 16].

$$f_i = \sum_{i=1}^n w_i \times \frac{l_i - v_i}{l_i} \quad (2)$$

onde:

- f_i : Valor do *fitness* do indivíduo
- n : Número de *constraints* presentes no cálculo
- w_i : Peso da i -ésima *constraint*
- l_i : Limite da i -ésima *constraint*
- v_i : Número de violações da i -ésima *constraint*

Na Equação 2, cada restrição contribui individualmente com um valor para o *fitness*, sendo que este valor pode variar de zero (0) (l_i violações) a w_i (não ocorrem violações). Quanto maior o valor de f_i , mais adaptado ao ambiente é o indivíduo, isto é, melhor é a solução.

O peso de uma *constraint* é definido anteriormente à execução do algoritmo. Esse valor fica armazenado no banco de dados. Eles são definidos de acordo com o objetivo

final do algoritmo, ou seja, se uma *constraint* for de maior importância para a solução, logo ela terá um peso maior.

O número de violações, é calculado durante a execução do algoritmo. Quando uma solução é gerada e alguma parte dela não respeita o que é desejado em uma solução ideal, o algoritmo penaliza essa solução com uma violação. Por exemplo, se um professor é alocado para ministrar duas disciplinas não compartilhadas ao mesmo tempo, isso é uma violação do tipo *clash*, portanto essa *constraint* será contabilizada com uma violação.

O limite de uma *constraint* é definido de forma igual ao peso da *constraint*, antes da execução do algoritmo. Esse limite é atribuído como um controle do número de violações. Quanto mais próximo do limite o número de violações chegar, menor será o *fitness* do indivíduo. Se não houver violações, o *fitness* do indivíduo será incrementado em um valor igual ao peso da *constraint*. Esse valor vai diminuindo de acordo com o número de violações. Se o número de violações chegar no valor do limite, não é acrescentado nenhum valor ao *fitness* do indivíduo. Caso ocorra um número de violações superior ao limite, o agregado de valor daquela *constraint* fica *negativo*.

Na Tabela 5 é possível ver cada uma das *constraints* implementadas atualmente com seu peso e seu limite.

Tabela 5. *Constraints* do AG original com seus pesos e limites

<i>Constraint</i>	Peso	Limite	Tipo
Oferecimento	500000.00	20.00	<i>Hard</i>
Não Compartilhado	300000.00	2000.00	<i>Hard</i>
Clash	200000.00	1600.00	<i>Hard</i>
Indisponibilidade	100000.00	350.00	<i>Hard</i>
Max Horas	10000.00	200.00	<i>Soft</i>
Horas Dia	10000.00	400.00	<i>Soft</i>
Prioridade	1000.00	200.00	<i>Soft</i>
Horas Professor	200.00	4000.00	<i>Soft</i>
Turnos Excesso	200.00	20.00	<i>Soft</i>
Excesso Dias	20.00	200.00	<i>Soft</i>

A *constraint* de peso mais alto é a *Oferecimento*. Ela guarda a diferença entre o número de oferecimentos de um nível que um curso possui e os oferecimentos especificados e os oferecimentos alocados. Uma violação é contada para esta *constraint* sempre que um nível de um curso não tiver alocado não oferta o número de créditos de disciplinas que deveriam ser oferecidos para o AG alocar na grade horária. Os conflitos gerados por essa *constraint* devem ser resolvidos de forma manual antes da execução do algoritmo.

A *constraint Não Compartilhado* aponta o número de vezes em que uma disciplina compartilhada está oferecida em horários ou professores diferentes do que o esperado, ou seja, adiciona uma violação para cada crédito de disciplina fora do horário ou com professor diferente do resto do compartilhamento. Essa *constraint* tenta evitar que uma disciplina compartilhada fique alocada em um horário, e outra disciplina de mesmo compartilhamento

fique alocada em outro horário. A *constraint Clash* mostra se um professor está alocado um número de vezes em mais de uma disciplina não compartilhada no mesmo horário. Para cada *timeslot* onde um professor se encontra alocado em duas disciplinas que não são compartilhadas, uma violação é contada.

Os professores possuem cinco *constraints* para fazer controle sobre as suas alocações. A primeira é a *Indisponibilidade*, que conta uma violação para cada *timeslot* onde um professor está alocado para no qual está especificado que estaria indisponível. A *constraint Horas Dia* conta a soma do número de horas de excesso dos professores em um dia (8 horas). Por exemplo, um professor alocado para 10 horas em um dia vai somar 2 a essa violação. A outra *constraint* é a *Prioridade*, que contabiliza as violações de acordo com a soma das diferenças da prioridade do professor alocado para uma disciplina e da prioridade do professor de maior prioridade que poderia estar dando a disciplina. A *constraint Horas Professor* conta a diferença total entre horas definidas e horas alocadas para professores. A violação para esta *constraint* e as horas alocadas forem menores que as definidas é somada através do resultado absoluto da multiplicação entre a diferença de horas alocadas e definidas para o professor pelo valor do peso da hora do professor, que está definido no banco de dados. Se as horas alocadas forem maiores que as definidas, a violação somada será a diferença de horas elevado na potência de valor igual ao peso da hora do professor mais 2. Por último a *constraint Max Horas* é o número de diferença total entre horas alocadas e horas desejadas. Por exemplo, se forem desejadas 32 horas para um professor, a violação para esta *constraint* será o valor das horas alocadas menos 32 para professores com mais de 32 horas alocadas.

As outras duas *constraints* são a *Excesso Turnos* e *Excesso Dias*. A *constraint Excesso Turnos* adiciona uma violação para cada turno em excesso para cada dia (máximo 2). Por exemplo, um curso integral com um dia tendo aula em 3 turnos soma 1 a essa violação, e a *constraint Excesso Dias* busca minimizar para ter menos dias de aula na semana. Conta uma violação para cada dia a mais de aula além do mínimo necessário para um curso nível de acordo com o número de créditos oferecidos. Sendo assim, se o número de créditos totais do curso nível cabem em 3 dias da semana, se forem alocadas disciplinas em 5 dias diferentes na semana para esse curso nível, 2 violações serão contabilizadas.

2.5.3 Crossover e Elitismo

O cruzamento aplicado entre os pais dos indivíduos é feito com um ponto, similar a [5, 17], ou seja, o indivíduo novo é dividido em duas partes, a primeira sendo do pai e a segunda da mãe. O primeiro filho recebe a primeira parte dos genes do pai e a segunda parte dos genes da mãe. Para a geração do segundo filho ocorre o processo inverso.

O cruzamento em dois pontos o qual foi um dos métodos usados por [14] e o cruzamento uniforme usado em [8] também foram testados para ser uma opção para este

AG, porém a aplicação do cruzamento em cursos com disciplinas compartilhadas gerou indivíduos com uma média de menor aptidão do que seus pais, uma vez que a possibilidade de desfazer um compartilhamento é alta. Dessa forma, foi aplicado o cruzamento em um ponto apenas em níveis de cursos que não possuem disciplinas compartilhadas.

Nem todos os indivíduos da geração seguinte são formados por *crossover*. Esta solução também utiliza de elitismo, de forma semelhante a [4, 6]. Dessa forma é possível selecionar alguns dos melhores indivíduos para a próxima geração sem alteração em seu gene, prevenindo que boas soluções sejam perdidas durante a evolução.

2.5.4 Mutação

A mutação é utilizada para manter a diversidade da população através de pequenas alterações aleatórias realizadas em alguns indivíduos. Existem vários tipos de mutação que podem ser vistos na literatura, como pode ser visto na Tabela 2. Neste trabalho foi usada a mutação por *swap*, assim como encontrado em [10, 11, 12].

Na mutação implementada para o AG, existe a possibilidade de serem trocados professores ou horários das disciplinas alocadas. No caso dos professores, eles podem ser trocados por qualquer outro professor que está apto a ministrar a disciplina. No caso dos horários, uma disciplina que está em um horário, pode passar a ser ministrada em algum outro horário que pode estar vazio ou não. No caso do novo horário já estiver ocupado, é feito uma troca de horários entre as duas disciplinas. Os horários que podem ser utilizados também são respeitados de acordo com o turno em que a disciplina precisa ocorrer.

Uma dificuldade extra encontrada na mutação é o controle que deve ser tomado ao mutar uma disciplina compartilhada. As disciplinas compartilhadas são disciplinas que devem ocorrer no mesmo horário e com o mesmo professor. Com a mutação totalmente aleatória, uma das disciplinas que estão em um compartilhamento pode acabar trocando de horário ou de professor, desfazendo o compartilhamento e reduzindo o *fitness* da população. É necessário que as disciplinas compartilhadas sejam selecionadas para a mutação separadamente das outras disciplinas. Dessa forma quando uma mutação ocorrer em uma disciplina compartilhada, ela também acontece em todas as outras disciplinas do mesmo compartilhamento.

3. MATERIAIS E MÉTODOS

Neste capítulo serão abordados os elementos criados e modificados no AG da UPF durante a execução deste trabalho. Também é mostrado o fluxo do AG, e como foi realizada toda a análise para obter os resultados finais.

3.1 INTRODUÇÃO

O propósito realizado neste trabalho é a otimização da solução gerada pelo AG da UPF. Os indivíduos produzidos a cada geração devem se adaptar de tal forma a aumentar a nota do CPC, através da melhoria na avaliação sobre o corpo docente de cada curso da Universidade presente na prova do Enade do referido ano. Para isso é necessário que a função de avaliação dos indivíduos considere os insumos de titulação e regime de trabalho. Dessa forma a tendência do algoritmo é de produzir indivíduos que priorizem a alocação de professores com as titulações de Doutor ou Mestre e que tenham regime de trabalho parcial ou integral nos cursos que irão participar do Enade no ano.

Em AGs, esse tópico de adaptação a uma nova característica necessária é realizado com base no ajuste da função de avaliação, considerando novas *constraints* que irão balizar a adaptação do indivíduo. A adição de *constraints* deve ser implementada com cuidado, para não afetar negativamente outras tão ou mais importantes [59].

A Figura 5 mostra um fluxo com uma visão macro de todas as etapas realizadas no andamento do trabalho. Os quadrados brancos representam cada etapa do projeto e as notas em cinza descrevem em alguns pontos o que foi desenvolvido na etapa.

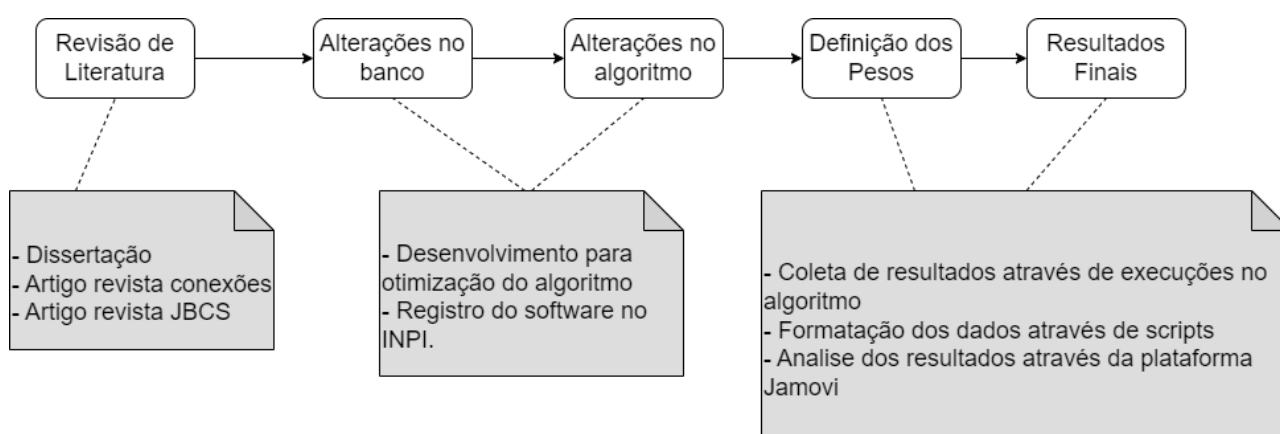


Figura 5. Fluxo do AG com as modificações inseridas

Para a realização deste trabalho, inicialmente foi realizada uma análise do atual estado do sistema existente, além de um estudo inicial do código para entender o funcionamento do algoritmo e obter o conhecimento necessário para o desenvolvimento da solução

pretendida. Desse modo foi possível fazer uma estruturação de quais locais precisariam de mudanças para que se atingisse o objetivo, ou seja, produzir grades horárias que resultem num conceito maior na nota do CPC nos cursos que realizam Enade no ano corrente.

Após o levantamento inicial de informações e estudos, foi realizada a modelagem da solução para o problema, adicionando modificações necessárias na base de dados, além da criação de novas *constraints* para que o algoritmo conseguisse controlar a evolução de acordo com os novos insumos. Os pesos e limites de cada uma das novas *constraints* foram avaliados, assim como das *constraints* já existentes.

Na sequência das modificações, foi adicionada uma entrada do ano no início da execução do AG. No entanto, a principal mudança realizada no algoritmo foi na função de avaliação, a fim de que considere novas *constraints*.

Para a validação das modificações realizadas, foi necessário testar o algoritmo, e ajustar os pesos das novas *constraints*. Para isso, foi definida uma metodologia de testes, com base em repetições. Após um conjunto de testes preliminares (piloto) e análise dos resultados iniciais, foram definidos os pesos das novas *constraints*. Por fim, foram realizadas as execuções finais para obtenção dos resultados, que foram analisados de forma estatística para que se comprovasse que a otimização foi de fato eficiente.

Convém salientar que nem todos os passos foram realizados sequencialmente um após o outro. A revisão de literatura foi realizada juntamente com outras etapas. Ela serviu como base para a construção da dissertação e também para a escrita dos dois artigos. As alterações no banco de dados e no algoritmo igualmente ocorreram de forma paralela, pois na medida em que foram adicionados campos novos nas tabelas, eles precisaram ser mapeados no algoritmo. Esses dois passos além de servirem como base para a coleta de resultados, também resultaram em um registro de software no INPI. Já as outras etapas foram realizadas na ordem em que o fluxo se encontra, pois todas elas dependem uma da outra.

As etapas de definição de pesos e resultados finais, apesar de terem ocorrido uma em sequência da outra, compartilham o mesmo fluxo geral, pois tanto para a definição de pesos quanto para geração dos resultados finais, o algoritmo precisou ser executado com as modificações, gerando resultados que foram comparados de forma estatística com execuções do algoritmo original. Ambas as etapas tiveram seus dados coletados através de *scripts* criados na linguagem *Python* e as análises foram realizadas através do software jamovi¹.

Um fluxo detalhado do AG com as modificações inseridas é apresentado na Figura 6. Nele é possível ver as mudanças adicionadas que estão numeradas e com fundo cinza.

O fluxo começa na etapa "Início da Execução", onde o AG importa os dados do banco. Esta etapa já conta com uma modificação realizada no trabalho, que é a adição de

¹Jamovi - <https://www.jamovi.org/>

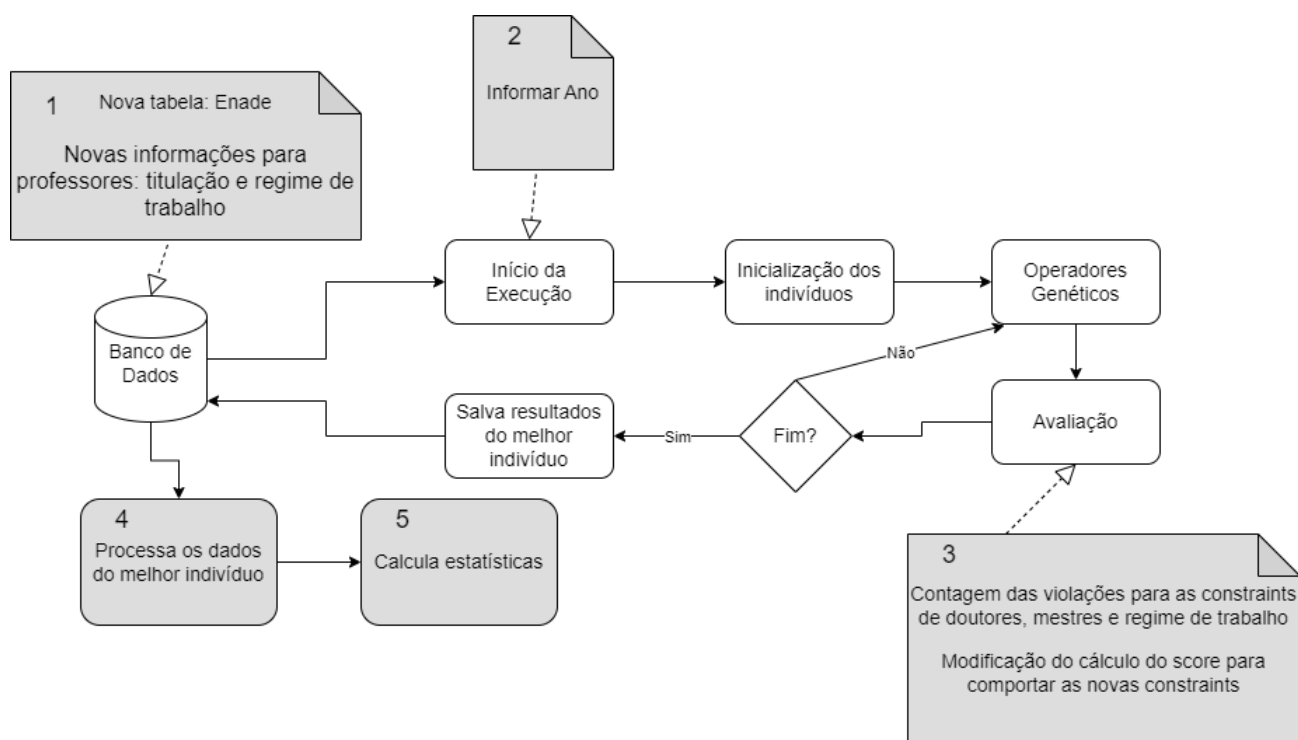


Figura 6. Fluxo do AG com as modificações inseridas

tabelas para entrada no banco de dados (destaque 1 na Figura 6). Já na etapa de Início da execução, é possível ver que a modificação de informar o ano foi inserida nessa etapa (destaque 2 na Figura 6).

Durante a execução, a etapa de inicialização dos indivíduos e aplicação dos operadores genéticos permanecem inalteradas. O fluxo segue até a função de avaliação, onde estão as principais mudanças implementadas no algoritmo com a integração das novas *constraints* no cálculo do *score* (destaque 3 na Figura 6). Ao final, os dados dos melhores indivíduos são salvos no banco de dados.

Ainda no fluxo da Figura 6, há dois passos que ocorrem após a execução e salvamento da solução no banco de dados. Eles não estão integrados a execução normal do algoritmo, mas foram essenciais na execução deste trabalho. Esses passos foram adicionados para verificar e validar se os resultados obtidos atingiram aos objetivos do trabalho. Com as novas *constraints* funcionando dentro do algoritmo, era necessário realizar testes para avaliar o impacto delas nas gerações.

Nesse contexto, o destaque 4 na Figura 6 foi desenvolvido, a fim de extrair os dados do melhor indivíduo gerado após uma execução. Esta tarefa foi realizada com um script em Python, especificamente criado para a função. A partir dos dados extraídos, o destaque 5 na Figura 6 representa a análise estatística dos resultados obtidos.

Nas próximas seções são descritos com mais detalhes cada uma das mudanças realizadas neste trabalho. Passando primeiramente pelas mudanças no banco de dados,

depois as mudanças realizadas no código do AG e por fim como foi realizado o processo de análise estatística e definição dos pesos das *constraints*.

3.2 BANCO DE DADOS

Foi analisada a estrutura em que se encontrava o banco de dados do AG para ver como as novas informações poderiam ser armazenadas para os professores e cursos. Primeiramente foi criada uma nova tabela chamada Enade para que se armazenem os cursos que realizarão o Enade em cada ano. Visto que alguns cursos possuem variações como, por exemplo, bacharel/licenciatura ou turnos diferentes, também há uma tabela chamada "Grupo Curso" que faz com que esses cursos possam ser trabalhados conjuntamente. É a partir dessa tabela que os cursos que participarão do Enade são referenciados. Portanto, a tabela Enade possui a coluna "grupo curso" juntamente com a coluna de "ano", que indica em que ano cada grupo curso irá realizar o Enade (Figura 7). Dessa forma é possível saber quais cursos serão utilizados na avaliação com as novas *constraints* a cada ano. Atualmente no banco de dados do AG existem um total de 81 grupos curso cadastrados. Desses 81, apenas 18 deles farão parte do Enade no ano de 2023, portanto serão esses os cursos analisados na validação das mudanças deste trabalho (Seção 4.1).

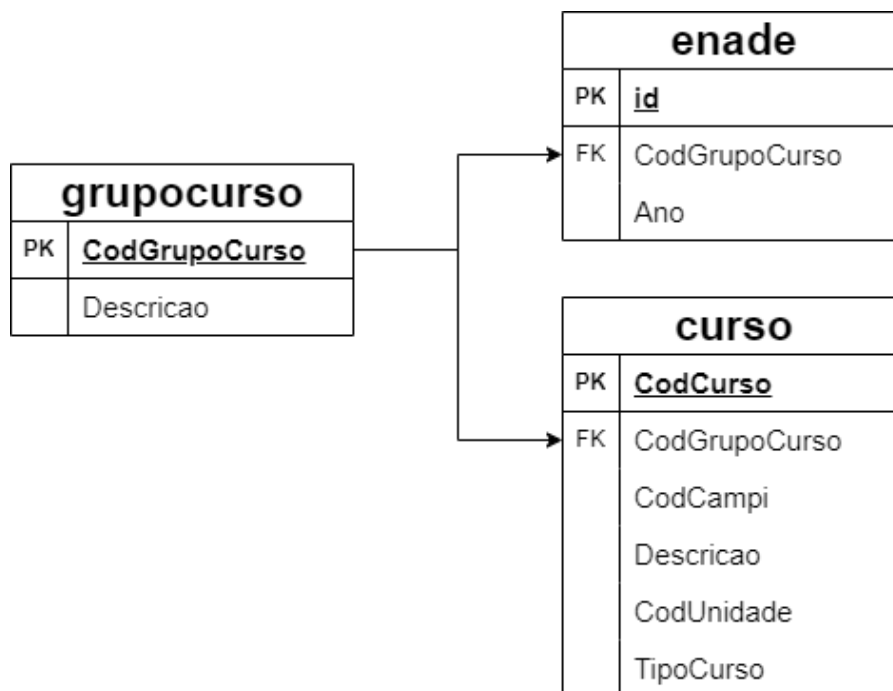


Figura 7. Modelagem do banco de dados para tabela Enade

Outra tabela que precisou ser modificada foi a de professores. Como os insumos para otimização são titulação e regime de trabalho de um professor, essas informações precisavam estar cadastradas no banco de dados. Apenas o campo de titulação já existia. Esse campo armazena uma letra para indicar se o professor é graduado (G), especialista

(E), mestre (M) ou doutor (D). Para indicação do regime de trabalho foi necessário criar uma nova coluna, indicando se o professor possui regime de trabalho horista, parcial ou integral. Esse campo foi adicionado como um inteiro no qual, que pode ser 0 (Indefinido), 1 (Integral), 2 (Parcial), 3 (Horista).

3.3 NOVAS CONSTRAINTS E FUNÇÃO AVALIAÇÃO

A função de avaliação é a função que calcula o *fitness* do indivíduo. É nela que se direciona o algoritmo para que ele gere soluções esperadas, ou seja, melhor adaptadas às características desejadas. Nesta função foram adicionadas ao cálculo do *fitness* novas *constraints*. Como a fórmula do cálculo do *fitness* é baseada através das violações, também precisaram ser encontradas (contadas) as violações para cada uma dessas *constraints*.

Como são três insumos adicionados a nota do CPC, foram definidas três novas *constraints*:

- Titulação Mestre;
- Titulação Doutor;
- Regime de Trabalho.

O requisito Titulação poderia ser realizado com apenas 1 *constraint*, porém como o peso da titulação de Mestre é diferente da de Doutor no cálculo do CPC, não faria sentido unir as duas em apenas uma. Além disso, com duas *constraints* separadas, é possível ter mais flexibilidade quanto às mudanças de pesos futuras.

Na função de avaliação, os indivíduos da população são percorridos a fim de verificar a sua adaptação. No caso das três novas *constraints* deve-se percorrer cada indivíduo para verificar os professores alocados em cada um dos cursos em que o Enade será aplicado no ano, sendo assim quando um professor não atinge um requisito *constraint*, uma violação é adicionada.

Para a contagem de violações de titulação (mestre ou doutores), toda a vez que um professor no curso tivesse titulação de graduação ou especialista, uma violação é adicionada a contagem de mestres, e outra para contagem de doutores. Quando um professor tiver titulação mestre, ele apenas conta uma violação para a contagem de doutores, e por fim um professor doutor não conta violação de titulação. Também é importante frisar que um professor que for alocado para duas ou mais disciplinas no mesmo curso não irá contar como duas violações, já que para o cálculo da nota do Enade é feito uma proporção através de todos os professores atuantes no curso, e não uma contagem de turmas ministradas por professor.

Para a contagem de violações de regime de trabalho foi realizado uma comparação semelhante a da titulação. Quando um professor tiver regime de trabalho registrado como indefinido ou horista, uma violação será adicionada a contagem da *constraint* de regime de trabalho. Se um professor tiver regime parcial ou regime integral, nenhuma violação será contada. Com o número de violações contados, eles foram adicionados na fórmula do cálculo do *fitness* para juntamente com o pesos e limites das novas *constraints* para realizar o cálculo do novo score do indivíduo.

O processo de definição dos pesos das *constraints* deve ser realizado considerando diversos fatores, como por exemplo, qual o impacto de cada um dos insumos no cálculo do CPC. Além disso, mesmo que a solução deste trabalho busque a aumentar a nota do curso na avaliação do Enade, outras restrições inerentes ao problema devem ser respeitadas, sob pena da solução encontrada ser inviável ou impraticável. É necessário que os indivíduos continuem sendo gerados adaptados para as outras necessidades já estabelecidas na Universidade. Apesar das *constraints* existentes fazerem esse papel, a adição de novos atributos a serem considerados na avaliação das soluções, pode trazer efeitos colaterais nos resultados do algoritmo. Tal fato ocorre quando o algoritmo escolher indivíduos com base em algumas restrições em detrimento de outras. Ou seja, é possível que com uma priorização com os insumos do Enade, o resultado para algumas outras *constraints* seja impactado, significativamente a menor. Um exemplo simples é o fato de alocar para todas as disciplinas de um curso o mesmo professor, com titulação de Doutor e Regime de Trabalho Integral. Os insumos do Enade para o curso serão os máximos possíveis, mas a *constraint clash* teria uma enorme quantidade de violações, inviabilizando a solução. É necessário portanto, ajustar os pesos das novas restrições de forma a otimizar ao máximo o ganho na nota do CPC, porém gerando o mínimo de efeitos colaterais em outras, principalmente as *hards* (*clashes* e não compartilhados).

A definição dos pesos de cada uma das novas *constraints* foi realizada em conjunto com uma revisão dos pesos das atuais e com base num teste piloto, para verificar o impacto do peso das novas *constraints* na solução. Esse processo está descrito na seção 4.2. A definição levou em conta uma metodologia de execução, coleta e análise estatística de resultado, definido na seção 3.4.

3.4 METODOLOGIA PARA EXECUÇÃO E ANÁLISE DOS RESULTADOS

Para a validação dos resultados obtidos com este trabalho é necessário que a execução do algoritmo continue gerando soluções aptas à resolução do problema, porém priorizando os insumos usados no cálculo do CPC para os cursos com Enade no ano. Uma vez que o AG da UPF possui forte influência pseudo-aleatória, considerar os resultados de apenas uma execução pode representar um caso de "sorte" ou "azar" e com certeza não representa maior parte das execuções.

Dessa forma, foi adotada uma metodologia de usar conjuntos de execuções, com os mesmos valores de entrada e analisar as médias dos resultados produzidos. Além disso, um processo de análise estatística dos resultados é utilizado, a fim de verificar se houve realmente ganhos ou prejuízos com determinadas escolhas de parâmetros.

Este processo foi utilizado em dois momentos no trabalho:

- definição dos pesos das *constraints*: 10 execuções;
- obtenção dos resultados finais: 32 execuções;

Os resultados de cada um dos conjuntos de execuções foram comparados com um conjunto de execuções do algoritmo original, ou seja, sem a adição das modificações realizadas. É a partir dessa comparação que as análises estatísticas são realizadas.

As comparações foram realizadas com base nos resultados obtidos das soluções para os insumos e principais *hard constraints*. Para isso, foi desenvolvido um *script* em *Python* que se conecta ao banco de dados, extrai os dados dos professores alocados em cada disciplina dos cursos através de uma *query* em SQL e formata esses dados para contabilizar os professores de cada curso. Esse processo gera um arquivo no formato JSON que mostra a proporção de professores alocados para cada um dos insumos em cada curso otimizado.

Após todas as execuções finalizadas, o resultado era um agregado de arquivos JSON no mesmo formato, com a quantidade de professores alocados em cada uma das execuções. O próximo passo foi juntar essas informações, para isso um novo *script* foi criado, que pega as informações dos arquivos JSON e gera um único arquivo no formato *xlsx*, uma tabela com todos os resultados de cada uma das execuções.

Essa tabela final foi utilizada como entrada de dados para o *software* Jamovi, que faz os cálculos estatísticos, representados pelo destaque identificado como **5** na Figura 6, utilizadas para se obter os resultados finais. Tanto as estatísticas descritivas como média e desvio padrão, quanto a análise estatística através do teste t de *Student* e o teste de Levene, além de todos os testes de normalidade e de variâncias realizados, foram calculados através do Jamovi.

Os testes foram planejados considerando a fórmula de cálculo do Enade para gerar o conceito do curso para o corpo docente. A fórmula utilizada é de proporção, entre professores que satisfazem os critérios exigidos (Equação 1). Calculou-se a porcentagem de professores de um curso com titulação doutorado e a porcentagem de professores com titulação mestre em relação a quantidade total de professores alocados em cada curso que participa do Enade. De mesma forma, para o regime de trabalho, foi calculada a porcentagem dos professores alocados do curso que possuem regime de trabalho parcial ou integral em relação a quantidade total de professores alocados.

Para a análise dos resultados foram criadas tabelas mostrando a média, mediana e desvio padrão da porcentagem de todas as execuções do algoritmo, comparando os resultados das execuções realizadas com as mudanças das execuções realizadas sem as mudanças do AG. Para que se tivesse uma garantia quanto aos resultados obtidos, para todas as comparações foram realizados testes de comparação que na estatística consegue descrever se a mudança da proporção de professores alocados para um determinado grupo de execuções foi estatisticamente significativa ou foi mero acaso.

Além disso, antes da realização do teste de comparação, foi realizado o teste de Levene para verificar a variância das amostras. O teste de Levene é necessário pois a fórmula para calcular a diferença entre as proporções é distinta caso as amostras sejam reprovadas no teste. Todos os testes de Levene realizados neste trabalho se encontram nos apêndices.

Apenas para os casos onde a amostra de execuções era pequena, foi necessário além do teste de Levene, realizar o teste de Shapiro-Wilk para verificar a normalidade das amostras. Nestes casos, se a amostra for reprovada no teste de normalidade, o teste utilizado para verificar a significância da diferença das proporções, foi o teste U de Man-Witney. Para todos os outros casos, tanto para amostras pequenas que não foram reprovadas no teste de Shapiro-Wilk, quanto para todos os casos onde amostras grandes foram utilizadas, o teste para calcular a diferença foi o teste T de Student.

4. RESULTADOS

Este capítulo apresenta os resultados da pesquisa, fornecendo uma análise detalhada dos dados obtidos durante as execuções. O principal objetivo é analisar se faz sentido para a universidade utilizar a modificação realizada para otimizar a alocação de professores em relação aos insumos de titulação e regime de trabalho nos cursos que realizam o Enade. Inicialmente foram realizadas execuções a fim de buscar os melhores pesos para as três novas *constraints* implementadas. Em seguida, são apresentados os resultados globais com a média de todos os 18 cursos que participam do Enade no ano de 2023. Após isso, procede-se uma análise individual curso a curso para concluir em quais deles a mudança foi realmente efetiva. Os resultados apresentados seguiram a metodologia de coleta e análise dos resultados, apresentada em 3.4.

4.1 CURSOS UTILIZADOS NOS RESULTADOS

Para realização dos testes foram utilizados dados reais da UPF, com todos os professores e disciplinas do segundo semestre do ano de 2023. Os cursos que foram analisados nos resultados foram os 18 cursos que participaram do Enade no ano de 2023:

- AGRONOMIA
- ARQUITETURA E URBANISMO
- ENFERMAGEM
- ENGENHARIA MECÂNICA
- ENGENHARIA ELÉTRICA
- ENGENHARIA CIVIL
- ENGENHARIA DE ALIMENTOS
- FISIOTERAPIA
- MEDICINA VETERINÁRIA
- MEDICINA
- NUTRIÇÃO
- ODONTOLOGIA
- FARMÁCIA
- FONOAUDIOLOGIA
- ESTÉTICA E COSMÉTICA CST
- ENGENHARIA DE PRODUÇÃO
- ENGENHARIA QUÍMICA
- ENGENHARIA AMBIENTAL E SANITÁRIA

Apenas os cursos relacionados acima tiveram a quantidade de violações das novas *constraints* contabilizados. Para os demais cursos da UPF, foram geradas grades horárias para verificar se mesmo com as modificações não haveriam problemas relacionados as modificações.

4.2 DEFINIÇÃO DOS PESOS DAS *CONSTRAINTS*

A ideia inicial para definir os pesos das novas *constraints* foi analisar qual a importância dessa otimização em relação às outras necessidades na resolução da grade horária. Também é necessário levar em consideração qual o grau de otimização seria bom para a universidade, além de cuidar qual é o peso máximo que pode ser atribuído, considerando a quantidade de professores existentes, sem afetar de forma significativa as *constraints* já existentes. Um peso grande (próximo ao das *hard constraints*) poderia acarretar em um efeito colateral em outras *constraints*, tendo como consequência, por exemplo, o aumento de *clashes* e disciplinas não compartilhadas.

Por conta disso, foi decidido que o peso deveria ser maior que as *soft constraints* já existentes, porém mais baixo que as *hard constraints*. Ainda assim, como são três novas *constraints*, o peso entre elas também podem variar. Como no conceito para a nota do cálculo do CPC a proporção de doutores tem um peso que é o dobro do peso para a proporção de mestres e de regime de trabalho [28], optou-se por tentar seguir esse padrão. Porém, para ter uma melhor precisão com o pesos, foram realizados alguns testes para verificar como o algoritmo se comportava. Os testes realizados consistiram em executar o algoritmo com os pesos atuais sem acrescentar as novas *constraints* e executar o algoritmo com as novas *constraints*, em configurações de pesos distintos (Tabela 6), comparando seus resultados.

Tabela 6. Conjunto de Pesos para testes iniciais

Configuração de Pesos	Peso Doutores	Peso Mestres	Peso Regime de Trabalho
Configuração 1	40000	20000	20000
Configuração 2	35000	35000	35000
Configuração 3	50000	15000	25000
Configuração 4	50000	25000	25000
Configuração 5	35000	15000	20000

Cada um dos testes, assim como a execução com o algoritmo original foram realizados com 10 execuções do algoritmo, com população de 1000 indivíduos e 1000 gerações para evolução. Os resultados foram calculados de forma global, ou seja, a média de cada *constraint* em todos os 18 cursos com Enade em conjunto, totalizando 180 proporções para cada um dos insumos. Também foram calculadas as médias das violações das *constraints clash* e não compartilhado, pois dentre as quatro *hard* existentes, a de oferecimento não é afetada durante a evolução, e a de indisponibilidade não é tão crucial em relação as demais

constraints e em geral todas as violações da mesma são sanadas durante as execuções. As médias de violações dessas duas *hard constraints* foram computadas em relação à todos os cursos da universidade, ou seja, como foram 10 execuções, são apenas 10 valores computados em cada *constraint*. Os resultados para a média das proporção de doutores, mestres e regime de trabalho são apresentados respectivamente nas Figuras 8, 9 e 10.

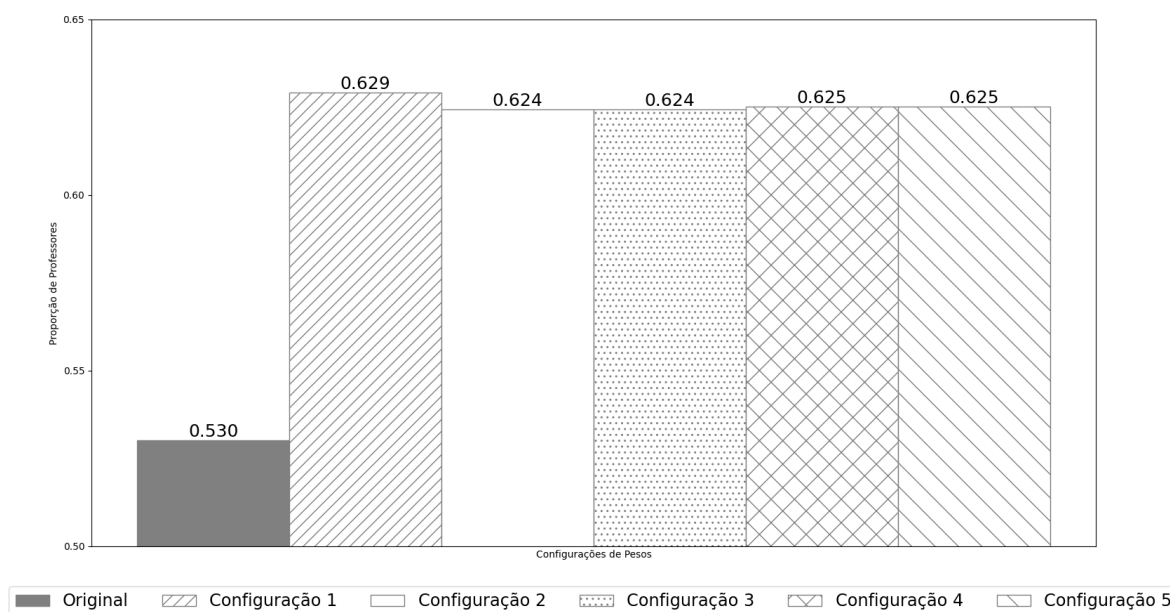


Figura 8. Média da proporção de professores doutores para cada configuração de peso

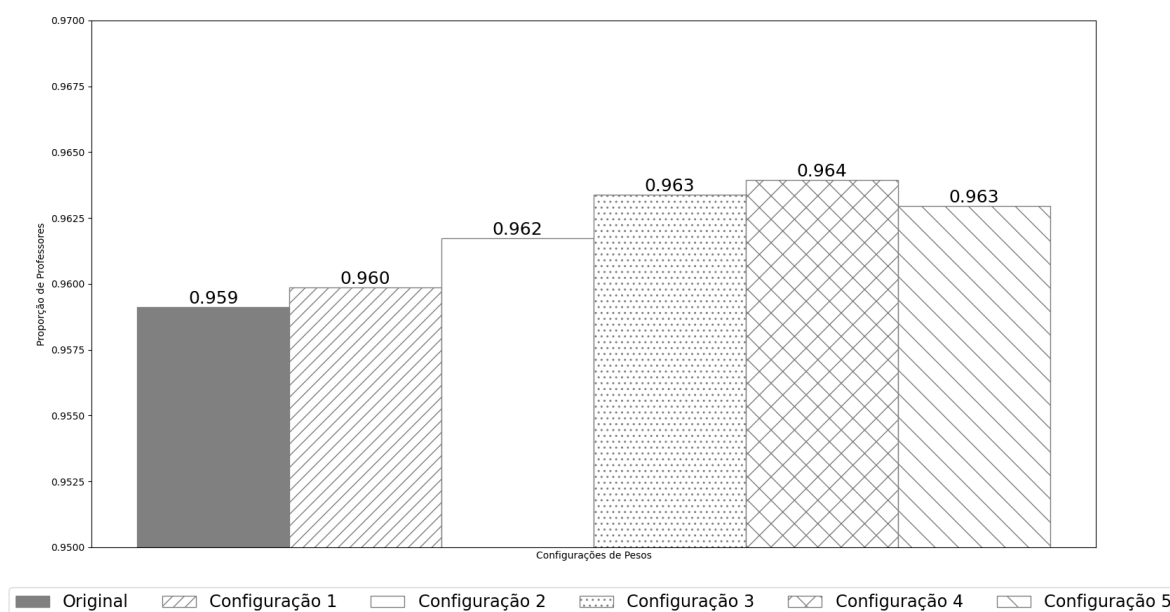


Figura 9. Média da proporção de professores mestres para cada configuração de peso

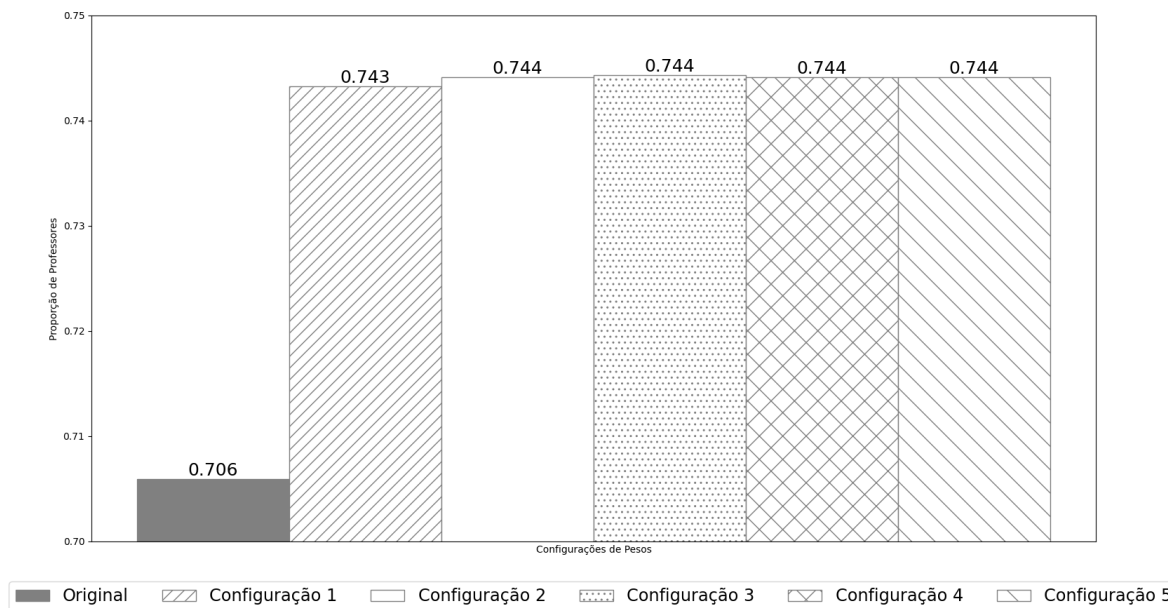


Figura 10. Média da proporção de professores com regime de trabalho parcial ou integral para cada configuração de peso

Pode se constatar que todos os conjuntos de teste obtiveram uma média melhor tanto nas titulações quanto na parte de regime de trabalho em relação às execuções do algoritmo original. Com os gráficos da Figura 8 e a Figura 10, é possível ver que a diferença nas proporções de doutores e regime de trabalho é maior entre o resultado original e as configurações de peso testadas. A proporção de mestres apresentada no gráfico da Figura 9, apesar da execução original ter a menor média, não possui uma diferença tão expressiva. Também vale ressaltar que a diferença entre cada um dos conjuntos de teste não foi grande, sendo apenas na terceira casa decimal para todas as médias analisadas.

No gráfico da Figura 11 são mostradas a média do número de *clashes* e disciplinas não compartilhadas para as 10 execuções de cada uma das configurações de pesos. Convém lembrar que os resultados são relativos a todos os cursos gerados e não apenas dos 18 relacionados para realizar o Enade.

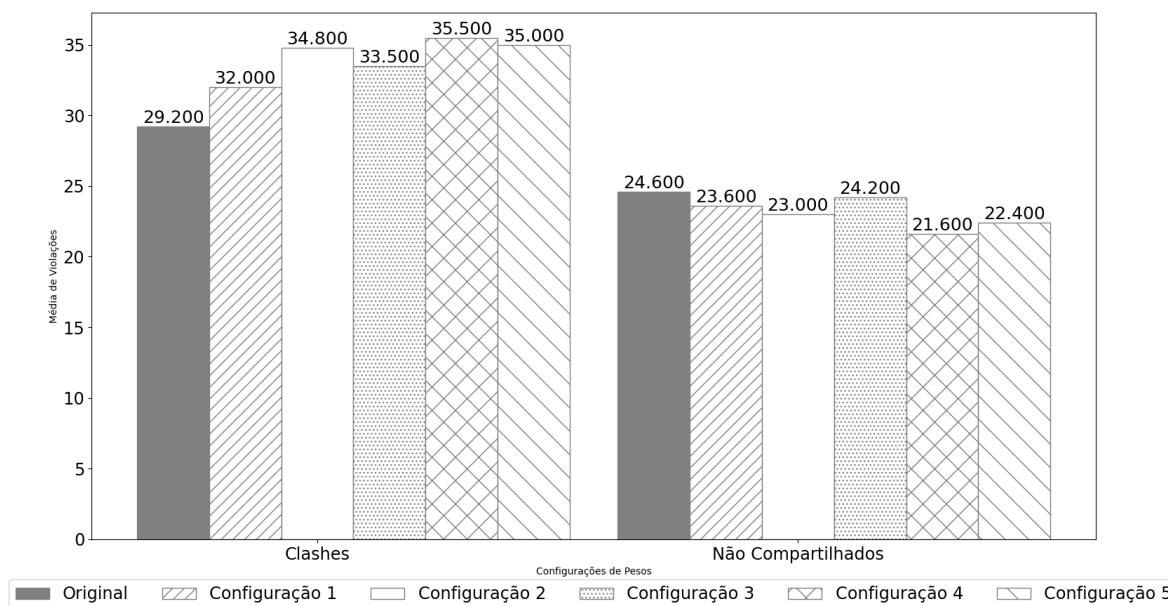


Figura 11. Média do número de violações para cada configuração de peso

Ao analisar os dados apresentados na Figura 11, pode-se observar que todos os conjuntos de teste aumentaram o número de *clashes*, mas diminuíram o número de disciplinas não compartilhadas em relação as execuções do algoritmo original. A diferença entre cada um dos conjuntos de teste não foi expressiva, sendo a maior delas, se compararmos a Configuração 1 com a Configuração 4 é de uma média de 3,5 *slots* com *lashes*, que é menos de um dia e turno (uma 4a. pela manhã por exemplo possui quatro *slots*). Há uma diferença de 2,6 *slots* onde disciplinas compartilhadas não mantiveram o mesmo horário, comparando a Configuração 3 com a Configuração 4.

Para a decisão de qual conjunto de pesos será utilizado para as execuções finais, foi realizada um análise estatística para cada configuração em relação ao original, para verificar se as alterações foram ou não estatisticamente significativas. Para a obtenção dos resultados foi realizado o Teste t de *Student* para cada configuração, comparando as 10 execuções com as 10 execuções originais. Cada uma das execuções gerou 18 valores (um para cada curso), então foram comparados um total de 180 valores em ambas execuções (original e modificado). Além disso foi realizado o teste de Levene para verificar a variação das amostras, pois a fórmula do teste t de *Student* é diferente se os dados tiverem homogeneidade na variância. Os resultados para o teste de variância estão contidos no Apêndice A. Como o total de 180 valores é considerado uma amostra grande, não foi necessário realizar o teste de normalidade para este caso [60].

Os resultados para o teste t de *Student*, assim como demais estatísticas descritivas são apresentadas na Tabela 7. O valor p do teste t de *Student* representa a significância do resultado, ou seja, a probabilidade de que a diferença observada entre as médias de dois grupos seja mero acaso. O nível de significância de 0,05 é frequentemente adotado

para determinar se os resultados são estatisticamente significativos. Deste modo, para este trabalho também se assumiu o nível de significância de 0,05 para avaliar a diferença observada entre os grupos. Isso significa que se o valor de p obtido no teste for menor que 0,05, os resultados são considerados estatisticamente significativos.

Outro valor que deve-se observar é o valor da coluna estatística, que mostra o quanto foi a diferença entre as amostras. Os valores negativos representam que o valor da proporção analisada foi maior na configuração de peso estabelecida do que no resultado original. Se for positivo, então o valor da proporção original foi maior.

Tabela 7. Resultados dos Testes para definição de pesos

Configuração de Pesos	Média	Mediana	DP	Estatística	p	
Prop. Dr.	Original	0,530	0,521	0,1603 ⁺	-	-
	Config. 1	0,629	0,618	0,2083 ⁺	-5,057	< ,001
	Config. 2	0,624	0,611	0,2173 ⁺	-4,680	< ,001
	Config. 3	0,963	1,000	0,0781 ⁺	-0,512	< ,001
	Config. 4	0,625	0,611	0,2171 ⁺	-4,728	< ,001
	Config. 5	0,625	0,613	0,2176 ⁺	-4,715	< ,001
Prop. Mr	Original	0,559	1,000	0,0796	-	-
	Config. 1	0,960	1,000	0,0793	-0,087	0,930
	Config. 2	0,962	1,000	0,0782	-0,312	0,755
	Config. 3	0,963	1,000	0,0781	-0,512	0,609
	Config. 4	0,964	1,000	0,0766	-0,585	0,559
	Config. 5	0,963	1,000	0,0781	-0,462	0,644
Prop. Reg. trab.	Original	0,706	0,743	0,1505	-	-
	Config. 1	0,743	0,722	0,1438	-2,378	0,018
	Config. 2	0,744	0,737	0,1504	-2,382	0,018
	Config. 3	0,744	0,730	0,1510	-2,386	0,018
	Config. 4	0,744	0,737	0,1507	-2,377	0,018
	Config. 5	0,744	0,743	0,1505	-2,377	0,018

Nota: Diferença estatisticamente significativa para $p < 0,05$. As médias de cada uma das configurações foram comparadas com a média original. ⁺Homogeneidade das variâncias rejeitada pelo teste de Levene.

Como é possível perceber na Tabela 7, todos os valores de p para o teste t nas proporções de doutores e regime de de trabalho foram menores do que 0,05, o que indica que para todas as configurações de pesos testadas, a proporção de professores com titulação de doutor e professores com regime de trabalho parcial ou integral possuíram um aumento considerado estatisticamente significativo.

Com relação à proporção de mestres, apesar de haver aumento na proporção em todas as configurações testadas, em nenhuma ocorreu um aumento considerado estatisticamente significativo. Isso pode ter ocorrido pois 83% dos professores da base de dados utilizada já possuem titulação mestre ou superior. Como mostrado no gráfico da Figura 9, a execução do algoritmo original já demonstra uma proporção maior que 95%, portanto, mesmo com a otimização as proporções de professores mestres alocados não conseguiriam ir muito além.

No que diz respeito às violações nas *constraints Clash* e Não Compartilhado, também foi realizado o teste t de *Student* para verificar se a mudança com os novos pesos era estatisticamente significativa. Para esse teste, como cada execução gera apenas um valor de *clash* e um de disciplinas não compartilhadas, foi analisado um total de 10 valores originais e 10 modificados, o que é considerado uma amostra pequena. Por conta disso, além do teste de Levene para verificar a homogeneidade das variâncias (Apêndice B), também foi necessário realizar o teste de Shapiro-Wilk para ver se as distribuições seguem um padrão de normalidade (Apêndice C). Caso reprovado no teste de normalidade, então o teste estatístico utilizado para verificar se o resultado é estatisticamente significativo é o teste U de Man-Whitney, caso contrário, o teste utilizado é o teste t de *Student*. A Tabela 8 mostra os resultados para as violações.

Percebe-se que o aumento nos *clashes* não foi considerado estatisticamente significativo para as configurações 1, 2 e 3. Houve também uma redução na quantidade de disciplinas não compartilhadas em todas as configurações testadas. Não se sabe o motivo da redução das violações nesta *constraint*. Para isso seria necessário fazer outra análise em relação a esses resultados, investigando quais disciplinas que podem estar sendo beneficiadas com essas mudanças e porque houve essa diminuição. Apesar da diminuição nas disciplinas não compartilhadas não ter sido considerada estatisticamente significativa em nenhum dos testes, foi um ótimo resultado, visto que, com a adição das novas *constraints* o esperado era que houvesse um pequeno aumento nas outras.

Tabela 8. Resultados dos Testes para definição de pesos para *clash* e não compartilhado

Configuração de Pesos	Média	Mediana	DP	Estatística	<i>p</i>	
Clash	Original	29,2	30,5	4,71	-	-
	Config. 1	32,0*	31,5	4,59	-1,146	0,195
	Config. 2	34,8*	33,0	8,66	-1,796	0,094
	Config. 3	33,5*	34,0	6,90	-1,628	0,123
	Config. 4	35,5*	36,0	5,87	-2,650	0,017
	Config. 5	35,0*	33,0	6,32	-2,330	0,033
Não Compartilhado	Original	24,6	24,5	2,12	-	-
	Config. 1	23,6**\$	22,0	5,64	27,0	0,085
	Config. 2	23,0*	22,0	4,64	0,991	0,340
	Config. 3	24,2**\$	23,0	3,79	0,291	0,775
	Config. 4	21,6**\$	22,0	6,57	32,0	0,182
	Config. 5	22,4*	21,0	4,43	1,420	0,180

Nota: *Teste t de *Student* ou **teste de Man-Whitney para amostras independentes. Diferença estatisticamente significativa para $p < 0,05$. As médias de cada uma das configurações foram comparadas com a média original. \$Distribuição de normalidade rejeitada pelo teste de Shapiro-Wilk. +Homogeneidade das variâncias rejeitada pelo teste de Levene.

Após realizados os testes globais com cada uma das configurações, foi elaborado um processo para a definição dos pesos finais. Como a Configuração 4 e Configuração 5 obtiveram um aumento estatisticamente significativo em número de violações do tipo *clash*

elas foram descartadas. Em seguida, comparando os resultados das proporções entre as Configurações 1, 2 e 3 foi decidido utilizar os pesos da Configuração 1.

Apesar dos resultados serem bastante próximos, a proporção de doutores na Configuração 1 foi um pouco maior que as demais neste teste e este é o maior peso para a nota do cálculo do CPC. Além disso, também foi a configuração que obteve a menor média de *clashes*. Salienta-se também, que nesta configuração os pesos de regime de trabalho e titulação de mestres possuem o mesmo valor, da mesma forma que são usados os pesos do conceito para o cálculo do CPC. A Tabela 9 exhibe todas as *constraints* já incluindo as novas e com os pesos definidos para serem utilizados no AG.

Tabela 9. *Constraints* atuais com seus pesos e limites

Constraint	Peso	Limite	Tipo
Oferecimento	500000,00	20,00	<i>Hard</i>
Não Compartilhado	300000,00	2000,00	<i>Hard</i>
Clash	200000,00	1600,00	<i>Hard</i>
Indisponibilidade	100000,00	350,00	<i>Hard</i>
Titulação Doutor	40000,00	200,00	Medium
Titulação Mestre	20000,00	200,00	Medium
Regime Trabalho	20000,00	400,00	Medium
Max Horas	10000,00	200,00	<i>Soft</i>
Horas Dia	10000,00	400,00	<i>Soft</i>
Prioridade	1000,00	200,00	<i>Soft</i>
Horas Professor	200,00	4000,00	<i>Soft</i>
Turnos Excesso	200,00	20,00	<i>Soft</i>
Excesso Dias	20,00	200,00	<i>Soft</i>

A ideia da criação de *constraints* de peso intermediário em relação aos já existentes fez com que se adicionasse uma nova categoria para identificá-los neste trabalho. Portanto, apesar da literatura dividir apenas em *hard* e *soft*, foi criado para este AG uma definição de *medium* para abranger essas novas *constraints*. Essa definição foi realizada para destacar a importância dessas *constraints* em relação às outras *soft constraints* existentes e também mostra que, apesar destas *constraints* não serem do tipo *hard*, ou seja, não serem essenciais para uma solução ser considerada viável neste problema, elas tem um impacto grande para definir se a solução é eficiente no ponto de vista da universidade.

4.3 RESULTADOS GLOBAIS

Com o objetivo de avaliar as novas *constraints* e pesos definidos, foram realizados testes envolvendo a configuração definidas com base na Tabela 9, considerando os 18 cursos

relacionados. Os resultados obtidos através das execuções foram positivos em relação às execuções originais. De forma semelhante aos testes da Seção 4.2, o aumento ocorreu em todos os insumos, porém performou melhor para titulação de doutores e regime de trabalho do que para mestres. A Tabela Tabela 10 contém os resultados destes testes. No Apêndice D estão os resultados do teste de Levene para a amostra global da proporção doutores, mestres e regime de trabalho.

Tabela 10. Estatística dos resultados globais

Insumo	Tipo	Média	Mediana	DP	Estatística	<i>p</i>
Doutores	Original	0,528	0,516	0,1738 ⁺	-8,333	< ,001
	Modificado	0,624	0,611	0,217		
Mestres	Original	0,963	1	0,0777	-0,181	0,857
	Modificado	0,963	1	0,0773		
Regime de Trabalho	Original	0,701	0,680	0,1634	-4,444	< ,001
	Modificado	0,743	0,722	0,1505		

Nota: Diferença estatisticamente significativa para $p < 0,05$. ⁺Homogeneidade das variâncias rejeitada pelo teste de Levene.

Como se pode ver na Tabela 10, a média das execuções do algoritmo original e também modificado são iguais até a terceira casa decimal para a proporção de mestres. Porém, como os outros dois insumos tiveram resultados melhores e a proporção de mestres não piorou, os impactos da otimização ainda serão benéficos no geral. Isso somado ao fato de que a universidade conta com uma grande quantidade de professores mestres e a proporção alocada nesse insumo já era grande (96%) mesmo antes da otimização.

Com a análise estatística realizada através do teste *t* de *Student* podemos confirmar esses resultados. É possível observar que a proporção de professores doutores alocados e a proporção de professores com regime de trabalho parcial ou integral, obtiveram um aumento considerado estatisticamente significativo (menor que 0,05), com destaque para a proporção de doutores que obteve na estatística *t*, que calcula a diferença entre as distribuições, um valor de 8,33. Quanto maior esse valor, maior a evidência contra a hipótese nula. Isso significa que há maior evidência de que há uma diferença significativa. Isso pode ser considerado um ótimo cenário principalmente devido ao fato de que para o cálculo da nota do CPC a proporção de doutores tem o dobro do peso em relação aos outros dois insumos. Porém, antes de seguir com as análises, há a necessidade de verificar como se comportaram as principais *hard constraints* (Tabela 11).

Tabela 11. Estatística dos resultados globais para *clash* e disciplinas não compartilhadas

Insumo	Tipo	Média	Mediana	DP	Estatística	<i>p</i>
Clash	Original	31,3	31,0	4,56	-0,973	0,335
	Modificado	32,8	31,5	7,44		
Não Compartilhado	Original	23,0	24,0	1,50	0,359	0,721
	Modificado	22,6	23,0	4,49		

Nota: Diferença estatisticamente significativa para $p < 0,05$.

Com relação às violações das *hard constraints*, analisando os dados disponíveis na Tabela 11 (Teste de Levene exibido no no Apêndice E), percebe-se que o aumento em número de *clashes* não resultou uma diferença estatística significativa, além de que houve uma diminuição no número de disciplinas não compartilhadas, o que pode ser considerado bom. Apesar da diminuição nas disciplinas não compartilhadas não ser estatisticamente significativo, o esperado no início do trabalho era um leve aumento. Se pode concluir então que o efeito gerado das novas *constraints* com os pesos selecionados não foi prejudicial para o desempenho do AG, considerando as principais *hard constraints* que já existiam.

4.4 RESULTADOS POR CURSO

Além da média geral da distribuição, como a avaliação feita para o CPC é individualizada por cursos, é importante saber o impacto dessas mudanças em cada um desses cursos que são avaliados. Para isso, foi extraído da solução a quantidade de professores alocados nos cursos que participam do Enade em 2023. Os resultados foram analisados através de uma visão estatística, com dados de média, mediana e desvio padrão, além de realizado o teste t de *Student* (Tabela 12) para constatar a significância da diferença entre os dados originais e modificados. O teste de Levene para as amostras de cada curso são exibidos no Apêndice F.

Tabela 12: Estatística dos resultados finais por curso

Curso	Insumos	Grupos	Média	Mediana	DP	Estatística	p
Agronomia	D _r	Original	0,791	0,778	0,0270	-7,45	< ,001
		Modificado	0,814	0,815	0,0257+		
	M _Δ	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R _T	Original	0,834	0,852	0,0188	-3,29	0,002
		Modificado	0,832	0,815	0,0188		
Arquitetura e Urbanismo	D _r	Original	0,498	0,500	0,0251	-1,953	0,055
		Modificado	0,508	0,500	0,0169		
	M _Δ	Original	0,914	0,903	0,0155	0,141	0,888
		Modificado	0,914	0,900	0,0164		
	R _T	Original	0,729	0,729	0,0255	-8,071	< ,001
		Modificado	0,776	0,767	0,0145+		
Enfermagem	D _r	Original	0,512	0,524	0,0252	-9,47	< ,001
		Modificado	0,565	0,556	0,02081		
	M _Δ	Original	0,952	0,952	0,00102	7,52	< ,001
		Modificado	0,949	0,950	0,00224+		

Continua na próxima página

Tabela 12 – Continuação da Página Anterior

Curso	Insumos	Grupos	Média	Mediana	DP	Estatística	p
	R.T.	Original	0,641	0,636	0,03126	-7,28	< ,001
		Modificado	0,696	0,700	0,02839		
Eng. de Alimentos	Dr.	Original	0,602	0,600	0,1400	-16,10	< ,001
		Modificado	1,000	1,000	0,000 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,967	1,000	0,779	-2,38	< ,001
		Modificado	1,000	1,000	0,000 ⁺		
Eng. Ambiental e Sanitária	Dr.	Original	0,670	0,667	0,1168	-15,99	< ,001
		Modificado	1,000	1,000	0,000 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,973	1,000	0,0642	-2,39	< ,001
		Modificado	1,000	1,000	0,000 ⁺		
Eng. Civil	Dr.	Original	0,435	0,437	0,0383	-11,08	< ,001
		Modificado	0,518	0,522	0,01771 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,823	0,822	0,0295	-8,33	< ,001
		Modificado	0,866	0,864	0,00339 ⁺		
Eng. Elétrica	Dr.	Original	0,450	0,450	0,0803	-10,56	< ,001
		Modificado	0,600	0,600	0,000 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,438	0,400	0,0609	-5,80	< ,001
		Modificado	0,500	0,500	0,0000 ⁺		
Eng. Mecânica	Dr.	Original	0,590	0,592	0,0406	-9,92	< ,001
		Modificado	0,663	0,667	0,00978 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,671	0,673	0,0335	-5,25	< ,001
		Modificado	0,703	0,708	0,00726 ⁺		
Eng. de Produção	Dr.	Original	0,455	0,444	0,0608	-13,69	< ,001
		Modificado	0,606	0,611	0,0140 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,667	0,667	0,0365	-7,32	< ,001

Continua na próxima página

Tabela 12 – Continuação da Página Anterior

Curso	Insumos	Grupos	Média	Mediana	DP	Estatística	p
Eng. Química	Dr	Modificado	0,718	0,722	0,0151 ⁺		
		Original	0,595	0,591	0,0417	-14,19	< ,001
	Ma	Modificado	0,700	0,700	0,00 ⁺		
		Original	1,000	1,000	0,00	-	-
	R T	Original	0,874	0,864	0,0232	-5,11	< ,001
		Modificado	0,900	0,900	0,00 ⁺		
Estética e Cosmética	Dr	Original	0,207	0,200	0,0213	1,79	0,078
		Modificado	0,200	0,200	0,000 ⁺		
	Me.	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	R.T.	Original	0,337	0,300	0,0468	-7,59	< ,001
		Modificado	0,400	0,400	0,000 ⁺		
Farmácia	Dr	Original	0,807	0,810	0,0137	-7,90	< ,001
		Modificado	0,855	0,857	0,00345 ⁺		
	Ma	Original	0,996	1,000	0,0137	-1,79	0,078
		Modificado	1,000	1,000	0,0000 ⁺		
	R T	Original	0,765	0,762	0,0410	-5,06	< ,001
		Modificado	0,803	0,810	0,01175 ⁺		
Fisioterapia	Dr	Original	0,381	0,381	0,01304	-10,40	< ,001
		Modificado	0,418	0,429	0,01222		
	Ma	Original	0,952	0,952	0,01222	-2,51	0,015
		Modificado	0,953	0,952	0,01222 ⁺		
	R T	Original	0,622	0,619	0,020	-13,88	< ,001
		Modificado	0,699	0,714	0,01690		
Fonoaudiologia	Dr	Original	0,361	0,366	0,0403	-8,10	< ,001
		Modificado	0,425	0,438	0,0181 ⁺		
	Ma	Original	0,899	0,875	0,0331	-1,87	0,067
		Modificado	0,914	0,933	0,0306		
	R T	Original	0,578	0,563	0,0310	-6,51	< ,001
		Modificado	0,616	0,625	0,0121 ⁺		
Medicina	Dr	Original	0,253	0,255	0,00302	-0,777	0,440
		Modificado	0,254	0,255	0,00550		
	Ma	Original	0,675	0,673	0,01337	0,336	0,738
		Modificado	0,673	0,673	0,01142		
	R T	Original	0,654	0,655	0,01337	-0,283	0,778
		Modificado	0,655	0,655	0,01265		

Continua na próxima página

Tabela 12 – Continuação da Página Anterior

Curso	Insumos	Grupos	Média	Mediana	DP	Estatística	p
Medicina Veterinária	Dr	Original	0,740	0,731	0,01722	-9,45	< ,001
		Modificado	0,769	0,769	0,00163 ⁺		
	M _A	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	P T	Original	0,653	0,654	0,00245	n	1 0000
		Modificado	0,653	0,654	0,00245 ⁺		
Nutrição	Dr	Original	0,511	0,500	0,03039	-14,608	< ,001
		Modificado	0,615	0,625	0,01679 ⁺		
	M _A	Original	0,938	0,938	0,00210	-0,547	0,586
		Modificado	0,939	0,938	0,00168		
	P T	Original	0,632	0,625	0,01102	-12,342	< ,001
		Modificado	0,738	0,750	0,02015 ⁺		
Odontologia	Dr	Original	0,644	0,643	0,02711	-11,40	< ,001
		Modificado	0,710	0,714	0,0177		
	M _A	Original	1,000	1,000	0,000	-	-
		Modificado	1,000	1,000	0,000		
	P T	Original	0,788	0,793	0,0129	-6,47	< ,001
		Modificado	0,810	0,815	0,0147		

Nota: Diferença estatisticamente significativa para $p < 0,05$. ⁺Homogeneidade das variâncias rejeitada pelo teste de Levene.

Com base na Tabela 12 é possível perceber que, assim como na distribuição global, a distribuição por cursos também teve um aumento geral nas médias dos três indicadores analisados.

4.4.1 Estatística descritiva por curso

Houve um aumento na média para proporção de doutores em 17 dos 18 cursos analisados, sendo que dois deles atingiram 100% neste insumo (Engenharia de Alimentos e Engenharia Ambiental e Sanitária). No curso onde este insumo não aumentou a diferença foi na 3a. casa decimal.

Quatro cursos tiveram aumento na proporção de mestres (Fisioterapia, Nutrição, Farmácia Fonoaudiologia), enquanto 11 cursos tinham 100% de professores mestres alocados antes da otimização, todos eles mantiveram esse valor. Já o curso de Farmácia, que tinha 99,6% dos professores mestres alocados em média antes da otimização, conseguiu chegar a 100% após a otimização. Além disso, dois cursos apresentaram uma piora em relação à média da proporção neste insumo, cujas alterações foram menores que 0,003.

Com relação ao regime de trabalho, 17 dos 18 cursos tiveram uma média maior após a otimização e apenas o curso de Medicina Veterinária manteve a média igual até a terceira casa decimal nas duas execuções. Além disso, assim como para proporção de doutores os cursos de Engenharia de Alimentos e Engenharia Ambiental e Sanitária atingiram 100% dos professores alocados neste insumo.

Nos três casos de cursos onde a média de um dos insumos não aumentou, houve um aumento (Engenharia Mecânica, Enfermagem e Medicina) houve aumento ou manutenção de um dos insumos no maior valor possível (100 %).

O destaque fica iminente nos cursos de Engenharia de Alimentos e Engenharia Ambiental e Sanitária que conseguiram, após a otimização, maximizar o número de professores (100%) para todos os três insumos. No geral, também é importante salientar que o desvio padrão diminuiu nas execuções do algoritmo otimizado, o que significa que houve menos variação entre essas execuções e que na maioria delas a otimização foi relevante. Vale lembrar também que alguns cursos podem não ter professores suficientes para uma maior/melhor otimização.

Com os valores apresentados na estatística descritiva, já é possível perceber que as distribuições geradas com as novas *constraints* atuando tendem a ter um valor maior nas proporções em cada um dos insumos analisados. Porém, apenas com aumento da média, não é possível dizer estatisticamente se o aumento foi ao acaso ou foi realmente por uma melhora real no algoritmo.

4.4.2 Análise de variações significativas por curso

Para saber se o aumento ou diminuição na proporção de professores do curso foi estatisticamente significativo, basta olhar o valor da coluna p que mostra o valor da significância do teste t de *Student* aplicado. Se esse valor foi menor do que 0,05 é porque a mudança foi estatisticamente significativa. Para saber se o resultado foi uma melhora ou piora em relação às execuções originais, é necessário olhar para a coluna estatística. Se o valor for negativo, é porque houve um aumento nos indicadores, se o valor foi positivo houve uma piora. O valor da estatística também consegue mensurar o tamanho da diferença entre cada execução. Quanto maior for o valor absoluto, maior a diferença entre as execuções.

Nos resultados obtidos pelo teste t de *Student* foi revelado uma melhora significativa na proporção de doutores em 16 dos 18 cursos analisados, o que pode ser considerado um ótimo resultado neste insumo. Os dois cursos que não tiveram aumento estatisticamente significativo foram os cursos de Medicina e Estética. O curso de Medicina obteve um aumento e o curso de Estética obteve uma piora nesses valores, ambos não considerados significativos. Metade dos cursos ultrapassaram uma estatística t de valor -10,00 na proporção de doutores em relação aos resultados originais e quatro deles se destacaram.

Foram os cursos de Engenharia Química (-14,19), Nutrição (-14,60), Engenharia Ambiental e Sanitária (-15,99) e Engenharia de Alimentos (-16,10).

Em relação à proporção de mestres, como é um insumo que já tinha uma proporção próxima a 100% antes da otimização, muitos cursos não obtiveram variação e não foi possível medir a estatística t. Dos 7 cursos calculados, 4 obtiveram um aumento no indicador e 3 obtiveram diminuição. Dos 4 que obtiveram aumento, apenas o curso de Fisioterapia obteve um aumento significativo. Já dos 3 que pioraram, um teve uma diminuição que destoou completamente do resto dos resultados. O curso de Enfermagem piorou obtendo um valor de 7,52 no cálculo da estatística para a relação de mestres, uma piora considerada estatisticamente significativa.

No que diz respeito ao regime de trabalho, os cursos que mais se destacaram foram Fisioterapia e Nutrição que obtiveram aumentos com valor da estatística t ultrapassando -10,00 nesse insumo. Também vale mencionar o aumento nos cursos como engenharia de produção (-7,32), Arquitetura e Urbanismo (-8,07) e Engenharia Civil (-8,33). Diferentemente dos outros dois insumos citados, onde houve casos de cursos que performaram pior em relação às execuções do algoritmo original, o pior caso para regime de trabalho foi no curso de Medicina Veterinária, o qual não mudou a proporção.

Ao analisar individualmente, alguns cursos se destacaram positivamente em seus resultados. O curso de Nutrição pode ter sido o que obteve o melhor resultado da base, pois além de ser o terceiro curso que mais otimizou a proporção de doutores, foi o segundo que mais otimizou o insumo de regime de trabalho (-12,34). Vale destacar também o curso de Fisioterapia, que obteve o melhor aumento na proporção de professores com regime parcial ou integral (-13,88) e também conseguiu uma melhora com valor da estatística chegando a -10,00 no insumo de doutores.

Alguns cursos, apesar de obterem resultados ruins em um insumo, compensaram com aumentos em outros insumos. É o caso de Enfermagem, que apesar do resultado ruim para mestres, ainda conseguiu um bom resultado pois os outros dois insumos aumentaram significativamente e tiveram aumento valor da estatística t chegando a -7,28 para regime de trabalho e -9,47 para doutores.

Já o curso de Estética, apesar da piora na proporção de doutores, obteve um aumento significativo, com a estatística t chegando a quase -8,00 nos insumos relacionados à proporção de professores com regime parcial ou integral. Na parte de mestres, assim como alguns outros cursos não foi possível obter o valor da estatística t pois não houve variação em relação aos resultados originais e gerados após as mudanças.

Por fim, também é necessário mencionar o curso de Medicina, que não obteve mudança considerada estatisticamente significativa em nenhum dos três insumos analisados. Além do pequeno aumento na proporção de doutores houve uma leve piora no indicador de mestres, e um aumento no indicador de regime de trabalho. Há, ainda outros cursos que apesar de não terem se destacado em nenhum dos insumos, no geral apresentaram

um resultado muito bom, como é o caso de Engenharia Mecânica, Engenharia Elétrica, Odontologia e Fonoaudiologia.

4.4.3 Análise geral dos resultados obtidos

No geral, praticamente todos os cursos conseguiram se beneficiar das mudanças realizadas no algoritmo e os resultados podem ser considerados muito bons. Se observarmos ainda a Tabela 12, é possível ver porque alguns cursos não obtiveram um aumento tão grande. Muitos cursos já possuíam 100% dos professores alocados em alguns insumos antes da otimização, principalmente se tratando de mestres. Nesse contexto, quando se olha a estatística t parece que não obtiveram um bom resultado, mas olhando a porcentagem de cada execução, percebemos que provavelmente eles apenas otimizaram até a capacidade total de professores.

Isso mostra que a funcionalidade implementada possui impacto positivo na organização de grade horária da universidade, fazendo com que esta consiga aumentar o conceito em praticamente todos os cursos. Certamente, a implementação dessa nova funcionalidade no AG mostra sua eficácia ao otimizar de forma significativa a geração da grade horária com os mesmos recursos disponíveis. Estes resultados podem aumentar o CPC dos cursos, e consequentemente o IGC da Instituição.

Além disso, vale ressaltar que a introdução dessa funcionalidade não implica em custos adicionais para a universidade. Ao contrário, sua implementação inteligente permite que a instituição opere de maneira mais eficiente, maximizando os recursos existentes. Isso fortalece financeiramente a universidade, trazendo oportunidades para investimento em outras áreas.

Dessa forma, a nova funcionalidade não apenas beneficia a organização da grade horária, mas também contribui na gestão financeira. Por fim, o resultado positivo deste trabalho é evidente, não se tratando apenas da melhoria do conceito de diversos cursos, mas também no prestígio da universidade, que demonstra sua capacidade de inovar e se adaptar às necessidades.

4.5 PUBLICAÇÕES

A geração de conhecimento é de extrema importância para a sociedade e, neste contexto, foram desenvolvidas uma série de iniciativas que visam promover o conhecimento gerado. Essas iniciativas representam um esforço contínuo para difundir o conhecimento gerado, contribuindo assim para o avanço não apenas da universidade, mas também para a comunidade acadêmica em geral.

Em primeiro lugar, o próprio projeto em si já representa uma significativa difusão de conhecimento. Uma nova versão do *software* foi criada, ampliando ainda mais seu potencial benefício para a universidade. Ele teve seu registro protocolado no INPI com o número BR 512024000459-9. Além disso o *software* desenvolvido também deverá passar pelas mãos de novos desenvolvedores que certamente continuarão contribuindo para que o algoritmo se aprimore de acordo com as necessidades da universidade.

Além do desenvolvimento do *software*, também foram produzidos artigos para divulgar as descobertas com a comunidade acadêmica. Um desses artigos foi submetido à revista Conexões – Ciência e Tecnologia, que é um periódico de divulgação científica e tecnológica do Instituto Federal do Ceará (IFCE) com o título "*A genetic algorithm for the curriculum-based course timetabling with shared subjects*" e outro será submetido ao *Journal of the Brazilian Computer Society (JBACS)* com o título "*A Curriculum-Based Course Timetabling with a focus on optimizing the ENADE grade for teaching staff*", ambos detalhando as inovações alcançadas no âmbito do projeto. Além disso, um terceiro artigo foi submetido ao evento EATI 2023 com o título "Gerção de Timetabling considerando avaliação do ENADE", onde também foi apresentado e gravado. A gravação da apresentação e o artigo foram disponibilizados nos anais do evento.

5. CONCLUSÃO

Este trabalho teve como objetivo modificar o algoritmo genético gerador da grade horária da universidade para otimizar a solução gerada de forma a priorizar a alocação de professores que respeitavam os insumos de titulação (mestre e doutor) e regime de trabalho (parcial ou integral). Tal necessidade se deu ao fato de que no Brasil as universidades são submetidas a avaliações e a avaliação do CPC leva em consideração esses fatores para a geração de um conceito final. Desta forma, ao priorizar a alocação destes professores, a nota do curso tende a aumentar.

Essa mudança, além de trazer mais um recurso útil para a grade gerada para a universidade, mostra que uma solução automatizada para o problema do *timetabling* é de grande eficácia para as instituições de ensino superior em geral. Ela é capaz de gerar uma grade horária viável de forma simples e rápida, e permite customizar a solução para direcioná-la a atender certos pontos específicos que cada instituição pode necessitar. O AG da UPF que até então possuía recursos não encontrados em outras soluções como disciplinas compartilhadas, múltiplas turmas e disciplinas EAD agora também tem uma abordagem priorizando a alocação de professores nos cursos que participam do Enade.

A otimização foi projetada para ocorrer apenas em cursos que participam do Enade. Dessa forma, como a avaliação ocorre em cursos diferentes a cada ano, o algoritmo também irá otimizar cursos diferentes. Para que isso acontecesse e o algoritmo acompanhasse essa mudança de cursos a cada ano, foi adicionado uma modificação no início da execução para informar o ano em questão, além de uma tabela para controlar quais cursos serão otimizados a cada ano.

Após a aplicação das modificações e análise dos resultados, pode-se dizer que a otimização proposta atendeu as expectativas de melhora, pois praticamente todos os cursos que foram otimizados conseguiram aumentar a proporção de professores alocados que se encaixavam nos insumos utilizados para calcular a nota do curso. Além disso, alguns cursos conseguiram chegar a 100% da otimização, ou seja, todos os professores se encaixavam nos três quesitos analisados. Os efeitos colaterais à mudança não foram significativos, aumentando minimamente a quantidade de *clashes* e até diminuindo o número de disciplinas não compartilhadas da solução.

Como resultado as soluções otimizadas escalam maior quantidade de professores com titulação de mestre e doutor, ou com regime de trabalho parcial ou integral para lecionar disciplinas nos cursos que realizam o Enade a cada ano. Desta forma, os cursos da universidade conseguem obter um conceito maior na avaliação do SINAES, sem a necessidade da universidade aumentar o consumo de recursos (contratação de novos docentes).

Embora este projeto tenha fornecido resultados interessantes, para trabalhos futuros, alguns ajustes ou novas abordagens podem ser aplicadas para otimizar ainda mais a nota gerada para o CPC de cada curso. Primeiramente, mais configurações de pesos distintos podem ser testadas, variando de forma mais agressiva o peso das novas *constraints*, além de também alterar o peso de outras *constraints*. Também podem ser feitas otimizações a partir da inicialização focando os cursos que passarão pela avaliação a alocar primeiramente os professores com maior titulação ou regime de trabalho específico. Também podem realizados estudos em volta das violações para entender o porquê ocorreu a diminuição nas disciplinas compartilhadas após as mudanças realizadas neste trabalho.

Não somente se limitando com foco na avaliação do Enade, o AG da UPF também pode acrescentar outras funcionalidades ainda não existentes e que são muito comuns em vários trabalhos na literatura. Por exemplo, expandir a solução para o espaço físico, trazendo conceitos de salas de aula, prédios e laboratórios utilizados. Dessa maneira também se pode controlar de forma mais fácil não somente quando, mas também onde irá ocorrer cada disciplina.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MANSOUR, N.; EL-JAZZAR, H. Curriculum based course timetabling. In: *2013 Ninth International Conference on Natural Computation (ICNC)*. Shenyang, China: IEEE, 2013. p. 787–792. ISBN 978-1-4673-4714-3.
- [2] BETTINELLI, A. et al. An overview of curriculum-based course timetabling. *TOP*, Springer Berlin Heidelberg, v. 23, n. 2, p. 313–349, jul 2015. ISSN 1134-5764. Disponível em: <<http://dx.doi.org/10.1007/s11750-015-0366-z>> <http://link.springer.com/10.1007/s11750-015-0366-z>>.
- [3] COOPER, T. B.; KINGSTON, J. H. The complexity of timetable construction problems. In: BURKE, E.; ROSS, P. (Ed.). *Practice and Theory of Automated Timetabling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 281–295. ISBN 978-3-540-70682-3.
- [4] GOZALI, A. A. et al. Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. *IEEJ Transactions on Electrical and Electronic Engineering*, v. 15, n. 3, p. 389–400, mar 2020. ISSN 1931-4973. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1002/tee.23067>>.
- [5] ELS, R.; PILLAY, N. An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem. In: *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*. IEEE, 2010. p. 460–466. ISBN 978-1-4244-7377-9. Disponível em: <<http://ieeexplore.ieee.org/document/5716340/>>.
- [6] PHUC, N. B.; KHANG, N. T. T. M.; NUONG, T. T. H. A New Hybrid GA-Bees Algorithm for a Real-world University Timetabling Problem. In: *2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation*. IEEE, 2011. p. 321–326. ISBN 978-1-4577-1151-0. Disponível em: <<http://ieeexplore.ieee.org/document/6131772/>>.
- [7] ABDELHALIM, E. A.; El Khayat, G. A. A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA). *Alexandria Engineering Journal*, Faculty of Engineering, Alexandria University, v. 55, n. 2, p. 1395–1409, jun 2016. ISSN 11100168. Disponível em: <<http://dx.doi.org/10.1016/j.aej.2016.02.017>> <https://linkinghub.elsevier.com/retrieve/pii/S1110016816000703>>.
- [8] JAENGCHUEA, S.; LOHPETCH, D. A hybrid genetic algorithm with local search and tabu search approaches for solving the post enrolment based course timetabling problem: Outperforming guided search genetic algorithm. In: *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE,

2015. v. 463, p. 29–34. ISBN 978-1-4673-7863-5. ISSN 21945357. Disponível em: <<http://ieeexplore.ieee.org/document/7408907/>>.

- [9] HOSNY, M. Metaheuristic Approaches for Solving University Timetabling Problems: A Review and Case Studies from Middle Eastern Universities. In: ROCHA, Á.; SERRHINI, M. (Ed.). Cham: Springer International Publishing, 2019, (Smart Innovation, Systems and Technologies, v. 111). p. 10–20. ISBN 978-3-030-03576-1. Disponível em: <<http://link.springer.com/10.1007/978-3-030-03577-8> http://link.springer.com/10.1007/978-3-030-03577-8_2>.
- [10] BORGES, S. K. *RESOLUÇÃO DE TIMETABLING UTILIZANDO ALGORITMOS GENÉTICOS E EVOLUÇÃO COOPERATIVA*. 103 p. Dissertação (Master's thesis) — Universidade Federal do Paraná, 2003.
- [11] FUCILINI, T. P. et al. Timetabling com algoritmos genéticos: resultados, restrições e exploração do paralelismo. *HÍFEN*, v. 32, n. 62, 2008. ISSN 0103-1155.
- [12] RAGHAVJEE, R.; PILLAY, N. An informed genetic algorithm for the high school timetabling problem. In: *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '10*. New York, New York, USA: ACM Press, 2010. p. 408–412. ISBN 9781605589503. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1899503.1899555>>.
- [13] SUYANTO. An Informed Genetic Algorithm for University Course and Student Timetabling Problems. In: RUTKOWSKI, L. et al. (Ed.). *Artificial Intelligence and Soft Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 229–236. ISBN 978-3-642-13232-2. Disponível em: <http://link.springer.com/10.1007/978-3-642-13232-2_28>.
- [14] NGUYEN, K. et al. Memetic algorithm for a university course timetabling problem. In: TAN, H. (Ed.). *Informatics in Control, Automation and Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 67–71. ISBN 978-3-642-25899-2.
- [15] BADONI, R. P.; GUPTA, D.; MISHRA, P. A new hybrid algorithm for university course timetabling problem using events based on groupings of students. *Computers & Industrial Engineering*, Elsevier Ltd, v. 78, p. 12–25, dec 2014. ISSN 03608352. Disponível em: <<http://dx.doi.org/10.1016/j.cie.2014.09.020> <https://linkinghub.elsevier.com/retrieve/pii/S0360835214002873>>.
- [16] ALVES, S. S. A.; OLIVEIRA, S. A. F.; NETO, A. R. R. A recursive genetic algorithm-based approach for educational timetabling problems. In: _____. *Designing with Computational Intelligence*. Cham: Springer International Publishing, 2017. p. 161–175. ISBN 978-3-319-44735-3. Disponível em: <https://doi.org/10.1007/978-3-319-44735-3_9>.

- [17] ALNOWAINI, G.; ALJOMAI, A. A. Genetic Algorithm For Solving University Course Timetabling Problem Using Dynamic Chromosomes. In: *2021 International Conference of Technology, Science and Administration (ICTSA)*. IEEE, 2021. p. 1–6. ISBN 978-1-6654-2862-0. Disponível em: <<https://ieeexplore.ieee.org/document/9406539/>>.
- [18] GUERRA, M.; CAVALCANTI, L. *AVALIAÇÃO DA EDUCAÇÃO SUPERIOR NO BRASIL: DAS PRIMEIRAS REGULACOES ATÉ O SISTEMA NACIONAL DE AVALIAÇÃO DA EDUCAÇÃO SUPERIOR (SINAES)*. UFPB, 2020. ISBN 978-65-5942-033-9. Disponível em: <<http://www.editora.ufpb.br/sistema/press5/index.php/UFPB/catalog/view/240/786/6755-1>>.
- [19] BRASIL. *Exame Nacional de Desempenho dos Estudantes (Enade)*. 2022. Disponível em: <<https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/enade>>. Acesso em: Nov. 2022.
- [20] QUINTILIO, R. et al. *NOTA TÉCNICA Nº 58/2020/CGCQES/DAES*. 2019. Disponível em: <https://download.inep.gov.br/educacao_superior/enade/notas_tecnicas/2019/NOTA_TECNICA_N_58-2020_CGCQES-DAES_Metodologia_de_calculo_do_CPC_2019.pdf>. Acesso em: May. 2024.
- [21] COLAJANNI, G.; DANIELE, P. A new model for curriculum-based university course timetabling. *Optimization Letters*, Springer Berlin Heidelberg, may 2020. ISSN 1862-4472. Disponível em: <<http://link.springer.com/10.1007/s11590-020-01588-x>>.
- [22] BRASIL. Lei nº 10.861, de 14 de abril de 2004. institui o sistema nacional de avaliação da educação superior – sinaes e dá outras providências. *Diário Oficial [da] República Federativa do Brasil*, Brasília, DF, 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/lei/l10.861.htm>.
- [23] MEC. *Sistema Nacional de Avaliação da Educação Superior (Sinaes)*. 2022. Disponível em: <<http://portal.mec.gov.br/sinaes/apresentacao>>.
- [24] RIPPEL, P. d. O.; PINTO, R. S. *AVALIAÇÃO DOS CURSOS DE GRADUAÇÃO DA UNIVERSIDADE FEDERAL DE PELOTAS: UMA ANÁLISE DO PERÍODO PÓS-REUNI*. 2017. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/179298/101_00693%20-%20ok.pdf?sequence=1isAllowed=y>.
- [25] Universidade Federal de Goiás. *Conceito Institucional*. 2024. Disponível em: <<https://cpa.secplan.ufg.br/p/48078-conceito-institucional>>. Acesso em: May, 2024.
- [26] PINTO, R. S.; MELLO, S. P. T. d.; MELO, P. A. Meta-avaliação: uma década do processo de avaliação institucional do sinaes. *Avaliação: Revista da Avaliação da Educação*

Superior (Campinas), Publicação da Rede de Avaliação Institucional da Educação Superior (RAIES), da Universidade Estadual de Campinas (UNICAMP) e da Universidade de Sorocaba (UNISO)., v. 21, n. 1, p. 89–108, Mar 2016. ISSN 1414-4077. Disponível em: <<https://doi.org/10.1590/S1414-40772016000100005>>.

- [27] BRASIL. *Questionário do Estudante*. 2022. Disponível em: <<https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/enade/questionario-do-estudante>>. Acesso em: Nov. 2022.
- [28] QUINTILIO, R. *NOTA TÉCNICA Nº 20/2019/CGCQES/DAES*. 2019. Disponível em: <https://download.inep.gov.br/educacao_superior/enade/notas_tecnicas/2019/nota_tecnica_n20-2019_CGCQES-DAES_calculo_NF_Enade.pdf>. Acesso em: May. 2024.
- [29] MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998. ISBN 0262631857.
- [30] BÄCK, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996. ISBN 9780195099713. Disponível em: <<https://doi.org/10.1093/oso/9780195099713.001.0001>>.
- [31] KOZA, J. R.; KEANE, M. A.; STREETER, M. J. Evolving Inventions. *Scientific American*, v. 288, n. 2, p. 52–59, feb 2003. ISSN 0036-8733. Disponível em: <<https://www.scientificamerican.com/article/evolving-inventions>>.
- [32] EIBEN, A.; SMITH, J. *Introduction to Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. 1–294 p. (Natural Computing Series). ISSN 1619-7127. ISBN 978-3-662-44873-1. Disponível em: <www.springer.com/series/0Apapers2://publication/uuid/F189C4FB-EB51-43FE-B6FC-17BAF1BF36C8 <http://link.springer.com/10.1007/978-3-662-44874-8>>.
- [33] TALBI, E. *Metaheuristics: From Design to Implementation*. Wiley, 2009. (Wiley Series on Parallel and Distributed Computing). ISBN 9780470496909. Disponível em: <<https://books.google.com.br/books?id=SIsa6zi5XV8C>>.
- [34] A.J., U.; P.D., S. CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW. *ICTACT Journal on Soft Computing*, v. 06, n. 01, p. 1083–1092, oct 2015. ISSN 09766561. Disponível em: <<http://ictactjournals.in/ArticleDetails.aspx?id=2109>>.
- [35] WEISE, T. *Global Optimization Algorithms - Theory and Application*. Second. Self-Published, 2009. Disponível em: <<https://api.semanticscholar.org/CorpusID:3079252>>.
- [36] HONG, T.-P.; WANG, H.-S. A dynamic mutation genetic algorithm. 1996 *IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence*

and Systems (Cat. No.96CH35929), v. 3, p. 2000–2005 vol.3, 1996. Disponível em: <<https://api.semanticscholar.org/CorpusID:61491991>>.

- [37] Lü, X. et al. Energy management of hybrid electric vehicles: A review of energy optimization of fuel cell hybrid power system based on genetic algorithm. *Energy Conversion and Management*, v. 205, p. 112474, 2020. ISSN 0196-8904. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0196890420300108>>.
- [38] LIANG, H. et al. An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system. *Mechanical Systems and Signal Processing*, v. 142, p. 106708, 2020. ISSN 0888-3270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0888327020300947>>.
- [39] BHOLA, J.; SONI, S.; CHEEMA, G. K. Genetic algorithm based optimized leach protocol for energy efficient wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, Springer Berlin Heidelberg, v. 11, n. 3, p. 1281–1288, 2020. ISSN 18685145. Disponível em: <<https://doi.org/10.1007/s12652-019-01382-3>>.
- [40] PINEDO, M. L. *Scheduling: Theory, Algorithms, and Systems*. 4th. ed. Springer New York, 2012. 676 p. Originally published by Prentice-Hall. ISBN 978-1-4614-2361-4 (eBook), 978-1-4899-9043-3 (Softcover). Disponível em: <<https://doi.org/10.1007/978-1-4614-2361-4>>.
- [41] ERNST, A.; JIANG, H.; KRISHNAMOORTHY, M. Exact solutions to task allocation problems. *Management Science*, v. 52, p. 1634–1646, 10 2006.
- [42] ABREU, L. R. et al. A genetic algorithm for scheduling open shops with sequence-dependent setup times. *Computers Operations Research*, v. 113, p. 104793, 2020. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054819302357>>.
- [43] REKHA, P. M.; DAKSHAYINI, M. Efficient task allocation approach using genetic algorithm for cloud environment. 2019.
- [44] PAL, S.; KHAN, I.; MAITI, M. K. A heuristic approach to solve multidimensional assignment problem. In: *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*. Dhanbad, India: IEEE, 2018. p. 1–7. ISBN 9781538630402,.
- [45] AHANDANI, M. A.; BAGHMISHEH, M. T. V. Memetic algorithms for solving university course timetabling problem. In: *2011 1st International eConference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2011. p. 40–44. ISBN 978-1-4673-5713-5. Disponível em: <<http://ieeexplore.ieee.org/document/6413321/>>.

- [46] BASHAB, A. et al. A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms. *Neural Computing and Applications*, Springer London, v. 32, n. 23, p. 17397–17432, dec 2020. ISSN 0941-0643. Disponível em: <<https://doi.org/10.1007/s00521-020-05110-3> <https://link.springer.com/10.1007/s00521-020-05110-3>>.
- [47] BABAEI, H.; KARIMPOUR, J.; HADIDI, A. A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, v. 86, p. 43–59, aug 2015. ISSN 03608352. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0360835214003714>>.
- [48] BOLAJI, A. L. et al. A modified artificial bee colony algorithm for post-enrolment course timetabling. In: TAN, Y.; SHI, Y.; MO, H. (Ed.). *Advances in Swarm Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. v. 7928 LNCS, p. 377–386. ISBN 978-3-642-38703-6. ISSN 03029743. Disponível em: <http://link.springer.com/10.1007/978-3-642-38703-6_45>.
- [49] ALZAQEBAH, M.; ABDULLAH, S. An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling. *Journal of Scheduling*, v. 17, n. 3, p. 249–262, jun 2014. ISSN 1094-6136. Disponível em: <<http://link.springer.com/10.1007/s10951-013-0352-y>>.
- [50] SAKAL, J.; FIELDSEND, J. E.; KEEDWELL, E. Learning assignment order in an ant colony optimiser for the university course timetabling problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. New York, NY, USA: ACM, 2021. v. 1, n. 1, p. 77–78. ISBN 9781450383516. Disponível em: <<https://dl.acm.org/doi/10.1145/3449726.3459534>>.
- [51] Larabi Marie-Sainte, S. A survey of Particle Swarm Optimization techniques for solving university Examination Timetabling Problem. *Artificial Intelligence Review*, Springer Netherlands, v. 44, n. 4, p. 537–546, dec 2015. ISSN 0269-2821. Disponível em: <<http://link.springer.com/10.1007/s10462-015-9437-7>>.
- [52] KANO, H.; CHEN, S. Particle swarm optimization with transition probability for timetabling problems. In: TOMASSINI, M. et al. (Ed.). *Adaptive and Natural Computing Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 256–265. ISBN 978-3-642-37213-1. Disponível em: <http://link.springer.com/10.1007/978-3-642-37213-1_27>.
- [53] ALOMARI, K. et al. A New Optimization on Harmony Search Algorithm for Exam Timetabling System. *Journal of Information & Knowledge Management*, v. 19, n. 01, p. 2040009, mar 2020. ISSN 0219-6492. Disponível em: <<https://www.worldscientific.com/doi/abs/10.1142/S0219649220400092>>.

- [54] WONG, C. H.; GOH, S. L.; LIKOH, J. A genetic algorithm for the real-world university course timetabling problem. In: *2022 IEEE 18th International Colloquium on Signal Processing Applications (CSPA)*. Selangor, Malaysia: IEEE, 2022. p. 46–50.
- [55] DUNKE, F.; NICKEL, S. A matheuristic for customized multi-level multi-criteria university timetabling. *Annals of Operations Research*, Apr 2023. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/s10479-023-05325-2>>.
- [56] FESTA, P.; RESENDE, M. G. C. GRASP. In: *Handbook of Heuristics*. Cham: Springer International Publishing, 2016. p. 1–24. Disponível em: <http://link.springer.com/10.1007/978-3-319-07153-4_23-1>.
- [57] UPF. *Sispad UPF*. 2022. Disponível em: <sispad.upf.br>. Acesso em: 2022-10-20.
- [58] SURRY, P. D.; RADCLIFFE, N. J. Inoculation to initialise evolutionary search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 1143, p. 269–285, 1996. ISSN 16113349.
- [59] LACH, G.; LÜBBECKE, M. E. Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*, v. 194, n. 1, p. 255–272, Apr 2012. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/s10479-010-0700-7>>.
- [60] GHASEMI, A.; ZAHEDIASL, S. Normality tests for statistical analysis: A guide for non-statisticians. *International Journal of Endocrinology and Metabolism, Briefland*, v. 10, n. 2, p. 486–489, dez. 2012. ISSN 1726-9148. Disponível em: <<http://dx.doi.org/10.5812/ijem.3505>>.

APÊNDICE A – TESTE DE LEVENE PARA RESULTADOS DE DEFINIÇÃO DE PESOS

Tabela 13. Teste de Levene das configurações de pesos testadas

Configuração de Pesos		F	gl	gl2	p
Prop. Dr.	Config. 1	7,98083	1	358	0,005
	Config. 2	9,7886	1	358	0,002
	Config. 3	9,95985	1	358	0,002
	Config. 4	9,5852	1	358	0,002
	Config. 5	9,92248	1	358	0,002
Prop. Mr	Config. 1	0,00477	1	358	0,945
	Config. 2	0,0286	1	358	0,866
	Config. 3	0,03085	1	358	0,861
	Config. 4	0,0865	1	358	0,769
	Config. 5	0,00816	1	358	0,928
Prop. Reg. trab.	Config. 1	0,40376	1	358	0,526
	Config. 2	0,0693	1	358	0,793
	Config. 3	0,00350	1	358	0,953
	Config. 4	0,0223	1	358	0,881
	Config. 5	0,02805	1	358	0,867

Nota: Homogeneidade das variâncias rejeitada pelo teste de Levene para $p < 0,05$

**APÊNDICE B – TESTE DE LEVENE PARA RESULTADOS DE
DEFINIÇÃO DE PESOS PARA QUANTIDADE DE CLASHES E
DISCIPLINAS NÃO COMPARTILHADAS**

Tabela 14. Teste de Levene dos resultados para definição de pesos

Configuração de Pesos	F	gl	gl2	p	
Clash	Config. 1	0,00139	1	18	0,971
	Config. 2	2,93	1	18	0,104
	Config. 3	0,713	1	18	0,410
	Config. 4	0,538	1	18	0,473
	Config. 5	1,91	1	18	0,184
Não Compartilhado	Config. 1	2,54889	1	18	0,128
	Config. 2	4,03	1	18	0,060
	Config. 3	1,060	1	18	0,317
	Config. 4	3,000	1	18	0,100
	Config. 5	3,06	1	18	0,097

Nota: Homogeneidade das variâncias rejeitada pelo teste de Levene para $p < 0,05$

**APÊNDICE C – TESTE DE SHAPIRO-WILK PARA RESULTADOS DOS
TESTES DE CONFIGURAÇÕES DE PESOS - CLASH E DISCIPLINAS
NÃO COMPARTILHADAS**

Tabela 15. Teste de Shapiro-Wilk dos resultados para definição de pesos

Configuração de Pesos		W	p
Clash	Config. 1	0,943	0,269
	Config. 2	0,978	0,898
	Config. 3	0,971	0,781
	Config. 4	0,972	0,788
	Config. 5	0,936	0,200
Não Compartilhado	Config. 1	0,757	< ,001
	Config. 2	0,956	0,459
	Config. 3	0,891	0,028
	Config. 4	0,819	0,002
	Config. 5	0,932	0,167

Nota: Normalidade rejeitada pelo teste de Shapiro-Wilk para $p < 0,05$

APÊNDICE D – TESTE DE LEVENE PARA RESULTADOS FINAIS - GLOBAL

Tabela 16. Teste de Levene dos resultados finais global

Insumo	F	gl	gl2	p
Doutores	17,8728	1	1150	< ,001
Mestres	0,0406	1	1150	0,840
Regime de Trabalho	2,1889	1	1150	0,139

Nota: Homogeneidade das variâncias rejeitada pelo teste de Levene para $p < 0,05$

APÊNDICE E – TESTE DE LEVENE PARA RESULTADOS FINAIS - CLASH E NÃO COMPARTILHADO

Tabela 17. Teste de Levene dos resultados finais para clash e não compartilhado

Constraint	F	gl	gl2	p
Clash	1,6282	1	62	0,207
Não Compartilhado	0,0115	1	62	0,915

Nota: Homogeneidade das variâncias rejeitada pelo teste de Levene para $p < 0,05$

APÊNDICE F – TESTE DE LEVENE PARA RESULTADOS FINAIS

Tabela 18: Teste de Levene dos resultados finais para cada curso

Curso	Insumo	F	gl	gl2	<i>p</i>
Agronomia	Prop. Dr	4,4604	1	62	0,039
	Prop. Mr	-	-	-	-
	Reg. Trab	0,0247	1	62	0,876
Arquitetura e urbanismo	Prop. Dr	1,63	1	62	0,207
	Prop. Mr	2,51	1	62	0,118
	Reg. Trab	15,79	1	62	< ,001
Enfermagem	Prop. Dr	0,570	1	62	0,453
	Prop. Mr	6,366	1	62	0,014
	Reg. Trab	0,558	1	62	0,458
Eng. de Alimentos	Prop. Dr	23,0	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	33,8	1	62	< ,001
Eng. Ambiental e Sanitaria	Prop. Dr	23,7	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	34,1	1	62	< ,001
Eng. Civil	Prop. Dr	17,0	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	42,9	1	62	< ,001
Eng. Elétrica	Prop. Dr	96,2	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	153,7	1	62	< ,001
Eng. Mecânica	Prop. Dr	60,3	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	30,3	1	62	< ,001
Eng. de Produção	Prop. Dr	29,79	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	8,44	1	62	0,005
Eng. Química	Prop. Dr	63,8	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	138,3	1	62	< ,001
Estética e Cosmética CST	Prop. Dr	16,0	1	62	< ,001

Continua na próxima página

Tabela 18 – Continuação da Página Anterior

Curso	Insumo	F	gl	gl2	<i>p</i>
	Prop. Mr	-	-	-	-
	Reg. Trab	363,6	1	62	< ,001
Farmácia	Prop. Dr	35,4	1	62	< ,001
	Prop. Mr	15,9	1	62	< ,001
	Reg. Trab	42,0	1	62	< ,001
Fisioterapia	Prop. Dr	0,854	1	62	0,359
	Prop. Mr	4,978	1	62	0,029
	Reg. Trab	0,380	1	62	0,540
Fonoaudiologia	Prop. Dr	20,43	1	62	< ,001
	Prop. Mr	3,13	1	62	0,082
	Reg. Trab	37,41	1	62	< ,001
Medicina	Prop. Dr	0,0778	1	62	0,781
	Prop. Mr	1,3576	1	62	0,248
	Reg. Trab	0,1822	1	62	0,671
Medicina veterinária	Prop. Dr	79,6	1	62	< ,001
	Prop. Mr	-	-	-	-
	Reg. Trab	2,01e-32	1	62	< ,001
Nutrição	Prop. Dr	13,068	1	62	< ,001
	Prop. Mr	0,310	1	62	0,580
	Reg. Trab	11,785	1	62	0,001
Odontologia	Prop. Dr	3,76	1	62	0,057
	Prop. Mr	-	-	-	-
	Reg. Trab	1,03	1	62	0,313

Nota: Homogeneidade das variâncias rejeitada pelo teste de Levene para $p < 0,05$.

ANEXO A – DISTRIBUIÇÃO GERADA PELO SISPAD PARA UM NÍVEL DE UM CURSO

ARQUITETURA E URBANISMO - CAMPUS I - PASSO FUNDO						
Campus: CAMPUS I - PASSO FUNDO, Nível: III-6483, Turno: Noite + Sábado, Turma: 23, C.Custo: 6483						
Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00 - 08:50						[A] SS
08:50 - 09:40	X	X	X	X	X	
09:55 - 10:45						
10:45 - 11:35						
14:00 - 14:50						
14:50 - 15:40	X	X	X	X	X	
15:55 - 16:45						
16:45 - 17:35						
19:20 - 20:05	B SS	D SS	E SS	F SS	G SS	H SS
20:05 - 20:50					I SS	K SS
21:05 - 21:50	C SS				J SS	L SS
21:50 - 22:35						N SS
						O SS
						[M] SS

#	Disciplina	EAD	Grupo	Cr(A)	Professor	Observações
A	ARQ167- HISTÓRIA E TEORIA DO URBANISMO I (2T)	80%	T1	2T	[A] SS	Grupo presencial. Compartilhada.
B	CIV196- TOPOGRAFIA DIGITAL E GEORREFERENCIAMENTO (2T2P)		T1	2T	[B] SS	
C	CIV196- TOPOGRAFIA DIGITAL E GEORREFERENCIAMENTO (2T2P)		P1	2P	[C] SS	
D	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P1	4P	[D] SS	
E	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P2	4P	[E] SS	
F	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P3	4P	[F] SS	
G	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P1	2P	[G] SS	
H	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P2	2P	[H] SS	
I	ARQ221- PROJETO ARQUITETÔNICO INTEGRADO III (6P)		P3	2P	[I] SS	
J	ARQ166- CONFORTO AMBIENTAL II: ILUMINAÇÃO NA ARQUITETURA (2T)		T1	2T	[J] SS	
K	ARQ168- REPRESENTAÇÃO E EXPRESSÃO APLICADA À ARQUITETURA E URBANISMO III (2P)		P1	2P	[K] SS	
L	ARQ168- REPRESENTAÇÃO E EXPRESSÃO APLICADA À ARQUITETURA E URBANISMO III (2P)		P2	2P	[L] SS	
M	ARQ169- COMPOSIÇÃO E CRÍTICA DE ARQUITETURA I (2T)	80%	T1	2T	[M] SS	Grupo presencial. Compartilhada.
N	ARQ220- HISTÓRIA E TEORIA DA ARQUITETURA II (2T)	80%	T1	2T	[N] SS	Grupo presencial.
O	ARQ222- INFORMÁTICA APLICADA À ARQUITETURA E URBANISMO II (2P)		P1	2P	[O] SS	

Figura 12. Grade horária gerada pelo sispad para o curso de arquitetura e urbanismo do terceiro semestre [57]