



Universidade de Brasília
Departamento de Estatística

Otimização na Alocação de Docentes
Uma Proposta de Ferramenta Computacional para o Departamento de
Estatística da UnB

Karl Peixoto

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2025

Karl Peixoto

**Otimização na Alocação de Docentes
Uma Proposta de Ferramenta Computacional para o Departamento de
Estatística da UnB**

Orientador(a): Eduardo Monteiro

Relatório apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

**Brasília
2025**

incluir dedicatória (opcional)

Agradecimentos

- Escrever os agradecimentos...

Resumo

A alocação de encargos didáticos em instituições de ensino superior é um problema clássico de otimização combinatória, conhecido como Timetabling, caracterizado pela alta complexidade e pela necessidade de satisfazer diversas restrições administrativas e preferências pessoais. No Departamento de Estatística da Universidade de Brasília (UnB), esse processo demanda esforço manual significativo a cada semestre. Este trabalho propõe a automatização e otimização desse processo através do desenvolvimento de um Sistema de Apoio à Decisão (SAD). O problema foi modelado matematicamente e solucionado através da comparação de diferentes abordagens, incluindo Programação Linear Inteira (PLI) e meta-heurísticas como Algoritmos Genéticos (AG) e Otimização por Colônia de Formigas (ACO). Para garantir a aplicabilidade prática dos modelos estudados, foi desenvolvida uma aplicação web em Python, projetada para ser intuitiva e de fácil manutenção. A ferramenta permite ao gestor configurar parâmetros, definir alocações fixas e visualizar os resultados de forma interativa, oferecendo transparência sobre o cumprimento de restrições e o atendimento às preferências docentes. Os resultados computacionais demonstraram que **finalizar trecho**. O sistema final foi entregue como legado ao departamento, proporcionando uma ferramenta robusta para agilizar o planejamento acadêmico e reduzir conflitos na distribuição de disciplinas.

Palavras-chaves:

Otimização Combinatória; Timetabling; Pesquisa Operacional; Meta-heurísticas;
Alocação de Docentes;

Lista de Tabelas

1	Limites de busca dos parâmetros do ACO	24
2	Limites de busca dos parâmetros do AG	26
3	Distribuição das preferências por disciplina	34
4	Distribuição das preferências por docente	36
5	Valores selecionados para os parâmetros do ACO	40
6	Valores selecionados para os parâmetros do AG	44
7	Resultado das simulações de comparação	44
8	Resumo organizado das execuções do AG (Apêndice).	60

Lista de Figuras

1	Gráfico de Dispersão 3D - Parâmetros LHS do ACO	26
2	Gráfico de Dispersão 3D - Parâmetros LHS do AG	28
3	Distribuição dos dados após perturbações aleatórias	31
4	Histograma de pontuação relativa e tempo até convergência ACO	37
5	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do ACO	38
6	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do ACO	39
7	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG	40
8	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG	41
9	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG	42
10	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG	43
11	Boxplots do tempo de execução total e tempo até convergência dos algoritmos	45
12	Gráficos de dispersão entre o tempo até convergência e as perturbações aplicadas	46
13	Gráficos de dispersão entre a pontuação relativa e as perturbações aplicadas	47
14	Captura de tela da interface inicial - Aba de Docentes	50
15	Captura de tela da interface inicial - Aba de Disciplinas	51
16	Captura de tela de execução	52
17	Captura de tela do resultado	53
18	Detalhes das preferências do docente	53
19	Detalhes dos interessados na disciplina	53

Sumário

1 Introdução	8
2 Referencial Teórico	9
3 Metodologia	10
3.1 Programação Linear Inteira	10
3.1.1 Características da PLI	10
3.1.2 Vantagens e Desvantagens	11
3.1.3 Aplicação ao problema	11
3.1.4 Componentes do Modelo	11
3.1.5 Função Objetivo	12
3.1.6 Restrições do Modelo	13
3.2 Algoritmos Genéticos	14
3.2.1 Representação e População	14
3.2.2 Função de Avaliação	15
3.2.3 Operadores Genéticos	15
3.2.4 Estrutura Geral do Algoritmo	15
3.2.5 Critérios de Parada	15
3.2.6 Vantagens e Desvantagens	16
3.2.7 Aplicação de Algoritmos Genéticos à Alocação de Docentes	16
3.2.8 Representação do Indivíduo (Cromossomo)	16
3.2.9 Função de Avaliação (Fitness Function)	17
3.2.10 Operadores Genéticos Aplicados	18
3.3 Algoritmos de Colônias de Formigas	19
3.3.1 Princípios Fundamentais	19
3.3.2 Componentes do ACO	20
3.3.3 Procedimento Geral do ACO	20

3.4 Aplicação de Algoritmos de Colônia de Formigas (ACO)	21
3.4.1 Estrutura do Problema e Representação	21
3.4.2 Componentes do Modelo ACO	21
3.5 Delineamento Experimental	23
3.5.1 Exploração do Espaço de Parâmetros	24
3.5.2 Cenários de Teste e Perturbação dos Dados	28
3.5.3 Métricas de Avaliação e Ambiente Computacional	31
4 Resultados	33
4.1 Caracterização do Cenário de Estudo	33
4.1.1 Análise da Procura por Disciplinas	34
4.1.2 Conclusão	37
4.2 Análise e Calibração do Algoritmo de Colônia de Formigas	37
4.2.1 Análise de sensibilidade dos hiperparâmetros	37
4.2.2 Análise de Convergência	39
4.3 Análise e Calibração do Algoritmo Genético	40
4.3.1 Análise de sensibilidade dos hiperparâmetros	40
4.3.2 Análise de Convergência	43
4.4 Comparação entre os algoritmos	44
5 Conclusão	48
6 Implementação e Ferramenta Desenvolvida	49
6.1 Inicialização da Plataforma	49
6.2 Interface	49
Referências	54
Apêndice	55
A Códigos de Perturbação	55
A.1 Código de simulação de horários	55
A.2 Código de simulação de preferências	57
A.3 Código de simulação de máximo de disciplinas	59

B Tabela Hiperparâmetros Ajuste - AGs	59
C nome apêndice 2	61
Anexo	62
A nome anexo 1	62

1 Introdução

A alocação de professores a disciplinas é uma atividade fundamental para o funcionamento eficiente de instituições de ensino superior. Esse processo, embora rotineiro, apresenta elevada complexidade devido à necessidade de conciliar diversas restrições e preferências, tais como carga horária dos docentes, compatibilidade com as disciplinas, restrições curriculares e disponibilidade de salas e horários. Além disso, é desejável que a satisfação dos professores seja considerada, equilibrando a distribuição das atividades e contribuindo para um ambiente de trabalho mais justo e eficiente.

Esse problema está inserido no contexto mais amplo dos problemas de *timetabling* e *scheduling*, os quais vêm sendo extensivamente estudados desde as décadas de 1960 e 1970, especialmente com o avanço da Pesquisa Operacional e da Inteligência Artificial. Segundo Burke e Petrovic (2002), o problema de alocação de aulas está entre os mais desafiadores na área de otimização combinatória, e a literatura especializada sugere uma ampla gama de abordagens, incluindo Programação Linear Inteira, Algoritmos Genéticos, Busca Tabu, Colônias de Formigas e Algoritmos Híbridos.

No contexto brasileiro, a maioria das alocações é feita manualmente ou com o uso de planilhas, o que limita a eficiência do processo. Este projeto propõe o desenvolvimento de uma ferramenta computacional de apoio à tomada de decisão para o Departamento de Estatística da Universidade de Brasília, que automatize a alocação de professores considerando restrições operacionais e preferências individuais. A solução será baseada em algoritmos de otimização, com comparação entre diferentes abordagens clássicas e metaheurísticas.

2 Referencial Teórico

Aqui você deve escrever a Revisão teórica ou incluir na Metodologia.

3 Metodologia

A proposta inicial para o estudo é a comparação da utilização de 3 algoritmos de otimização diferentes, sendo um deles exato (Programação Linear Inteira) e os outros dois sendo heurísticos, ou seja, que selecionam a melhor opção dentro das apresentadas, mas não garantem a convergência para a melhor solução global.

Os métodos propostos seguem com uma breve descrição abaixo:

3.1 Programação Linear Inteira

A Programação Linear Inteira (PLI) é uma técnica de otimização utilizada para resolver problemas nos quais as variáveis de decisão assumem apenas valores inteiros, geralmente binários (0 ou 1). Trata-se de um caso particular da Programação Linear (PL), no qual se impõe restrições de integralidade às variáveis.

A estrutura geral de um problema de PLI pode ser expressa da seguinte forma:

$$\text{Minimizar (ou Maximizar)} \quad c^\top x \quad (3.1.1)$$

$$\text{sujeito a} \quad Ax \leq b \quad (3.1.2)$$

$$x \in \mathbb{Z}^n \quad (3.1.3)$$

onde:

- x é o vetor de variáveis de decisão;
- c é o vetor de coeficientes da função objetivo;
- A é a matriz de coeficientes das restrições;
- b é o vetor de constantes do lado direito das restrições;
- \mathbb{Z}^n denota que as variáveis são inteiras.

3.1.1 Características da PLI

Problemas de PLI são comuns em contextos nos quais decisões discretas precisam ser tomadas, como alocação de recursos, escalonamento, roteamento e planejamento.

Uma classe frequente de variáveis são as binárias, indicadoras de escolha ou ativação de determinada alternativa:

$$x_i \in \{0, 1\}$$

Esses modelos permitem a formulação de restrições complexas, tais como exclusões mútuas, limites mínimos/máximos, prioridades e preferências, por meio de combinações lineares das variáveis.

3.1.2 Vantagens e Desvantagens

A principal vantagem da PLI é a obtenção de soluções exatas e interpretáveis. No entanto, a escalabilidade pode ser uma limitação, especialmente quando o número de variáveis e restrições cresce, o que pode demandar grande poder computacional.

3.1.3 Aplicação ao problema

A alocação de docentes às disciplinas é um problema clássico de otimização combinatória que pode ser modelado por meio da PLI. A modelagem a seguir tem como propósito encontrar uma distribuição de disciplinas que maximize a satisfação do corpo docente, representada por suas preferências de alocação, ao mesmo tempo que respeita as restrições operacionais de conflito de horário entre as disciplinas e limite de carga horária máxima.

3.1.4 Componentes do Modelo

Para a construção do modelo matemático, definem-se os seguintes componentes, baseados nos dados de entrada e nas decisões a serem tomadas.

Conjuntos e Índices.

- $p \in P$: Representa cada professor do departamento disponível para lecionar disciplinas.
- $d \in D$: Representa cada disciplina ofertada pelo departamento em um dado semestre.

Parâmetros.

Os parâmetros constituem os dados de entrada do modelo, extraídos de fontes como o questionário de preferências e os sistemas acadêmicos da universidade.

- $pref_{pd}$: Valor numérico da preferência do professor p em lecionar a disciplina d .
- ch_d : Carga horária, em horas por semestre, da disciplina d .
- $CH_{max,p}$: Carga horária máxima, em horas por semestre, que pode ser atribuída ao professor p .
- W : Coeficiente de penalidade, um valor escalar utilizado para realizar penalizações sobre alocações indesejadas.
- $Cf_{d_i d_j} \in \{0, 1\}$: Parâmetro binário que assume valor 1 se as disciplinas d_i e d_j são ofertadas em horários conflitantes, e 0 caso contrário.

Variáveis de Decisão.

As variáveis de decisão representam as escolhas que o modelo deve realizar.

- $x_{pd} \in \{0, 1\}$: Variável binária que assume valor 1 se o professor p for alocado para a disciplina d , e 0 caso contrário.
- $y_p \in \mathbb{Z}^+$: Variável inteira auxiliar que quantifica o número de alocações de baixo interesse para o professor p .

3.1.5 Função Objetivo

A função objetivo, apresentada na Equação 3.1.4, é formulada para maximizar o escore total de satisfação dos docentes, ao mesmo tempo que minimiza a ocorrência de alocações consideradas indesejáveis. O termo à esquerda maximiza a soma das preferências das alocações efetivadas. O termo à direita introduz uma penalidade, ponderada pelo coeficiente W , para cada alocação de baixo interesse realizada.

$$\text{Maximizar } Z = \left(\sum_{p \in P} \sum_{d \in D} pref_{pd} \cdot x_{pd} \right) - W \cdot \left(\sum_{p \in P} y_p \right) \quad (3.1.4)$$

A magnitude do coeficiente W é crucial para o comportamento do modelo. Um valor adequadamente alto garante que o otimizador evite violações das restrições brandas, a menos que seja estritamente necessário para se obter uma solução viável.

A abordagem proposta utiliza W como um único valor que penaliza apenas o número de violações de restrições brandas de forma igual. No entanto, é possível propor a criação de W_1, W_2, \dots, W_n , que apresentem pesos diferentes para violação de cada conjunto de restrições brandas. E também é possível propor W como um vetor, que tenha um valor diferente para cada professor, possibilitando a criação de uma hierarquia entre as preferências que favoreça professores que tenham outros encargos no departamento além das disciplinas, ou até mesmo utilizar isso para fazer correções ao longo do tempo e garantir que professores que foram menos favorecidos em semestres anteriores tenham suas preferências atendidas.

3.1.6 Restrições do Modelo

As restrições definem o espaço de soluções viáveis, garantindo que apenas alocações válidas sejam consideradas. Elas são divididas em restrições duras (*hard constraints*), que devem ser obrigatoriamente satisfeitas, e brandas (*soft constraints*), que são desejáveis, mas podem ser violadas a um custo.

Restrições Duras (Hard Constraints).

- Garantia de Oferta de Disciplinas:** Cada disciplina do conjunto de oferta D deve ser atribuída a exatamente um professor. Esta restrição, definida na Equação 3.1.5, assegura que toda a oferta de disciplinas do semestre seja coberta, e que nenhuma disciplina seja alocada para dois professores diferentes.

$$\sum_{p \in P} x_{pd} = 1, \quad \forall d \in D \quad (3.1.5)$$

- Limite de Carga Horária Docente:** A soma das cargas horárias das disciplinas alocadas a um professor não pode exceder sua capacidade máxima de trabalho, conforme a Equação 3.1.6.

$$\sum_{d \in D} ch_d \cdot x_{pd} \leq CH_{max,p}, \quad \forall p \in P \quad (3.1.6)$$

- Restrição de Conflito de Horário:** Para garantir que um docente não seja

alocado a duas disciplinas que ocorrem simultaneamente, a seguinte restrição é imposta para cada professor p e para cada par de disciplinas (d_i, d_j) com conflito de horário.

$$x_{pd_i} + x_{pd_j} \leq 1, \quad \forall p \in P, \forall (d_i, d_j) \text{ tal que } Cf_{d_id_j} = 1 \quad (3.1.7)$$

Restrições Brandas (Soft Constraints).

3. **Minimização de Alocações de Baixo Interesse:** O modelo deve evitar, sempre que possível, alocar um professor a uma disciplina pela qual ele demonstrou baixo interesse (e.g., $pref_{pd} = 1$). A Equação 3.1.8 conecta a variável de alocação x_{pd} à variável de penalidade y_p .

$$\sum_{d \in D | pref_{pd}=1} x_{pd} \leq y_p, \quad \forall p \in P \quad (3.1.8)$$

Se nenhuma disciplina de baixo interesse for alocada ao professor p , o somatório à esquerda será zero, permitindo que y_p seja zero e nenhuma penalidade seja incorrida. Contudo, se uma ou mais dessas disciplinas forem alocadas, y_p será forçado a ser no mínimo 1, ativando a penalidade W na função objetivo.

3.2 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são técnicas de otimização e busca inspiradas nos princípios da seleção natural e da genética biológica, propostos inicialmente em 1975 e adaptado em Holland (1992). São amplamente utilizados para resolver problemas complexos, especialmente aqueles com espaços de solução grandes, não lineares e com múltiplos ótimos locais.

3.2.1 Representação e População

A primeira etapa na construção de um AG é a definição da representação dos indivíduos, que codificam possíveis soluções do problema. Essa codificação pode ser feita por vetores binários, inteiros, reais ou estruturas mais complexas, dependendo do problema.

Uma *população inicial* é então gerada aleatoriamente, composta por N indivíduos (soluções candidatas).

3.2.2 Função de Avaliação

Cada indivíduo é avaliado por meio de uma *função de aptidão* (*fitness function*), que quantifica a qualidade da solução em relação ao objetivo do problema. Essa função guia o processo evolutivo, favorecendo soluções melhores ao longo das gerações.

3.2.3 Operadores Genéticos

A evolução da população ocorre por meio de operadores genéticos:

- **Seleção:** seleciona os indivíduos mais aptos para reprodução. Métodos comuns incluem roleta, torneio e seleção por ranking.
- **Crossover (recombinação):** combina pares de indivíduos selecionados para gerar novos indivíduos, simulando reprodução biológica. O cruzamento pode ocorrer em um ou mais pontos da estrutura do cromossomo.
- **Mutação:** introduz variações aleatórias em indivíduos da nova geração, alterando aleatoriamente partes da solução com baixa probabilidade, promovendo diversidade.

3.2.4 Estrutura Geral do Algoritmo

1. Inicializar a população com N indivíduos aleatórios.
2. Avaliar a aptidão de cada indivíduo.
3. Enquanto o critério de parada não for satisfeito:
 - (a) Selecionar indivíduos com base na aptidão.
 - (b) Aplicar operadores de crossover e mutação para gerar nova população.
 - (c) Avaliar a nova população.
 - (d) Atualizar a população (com ou sem elitismo).

3.2.5 Critérios de Parada

Alguns critérios comuns para interromper o algoritmo incluem:

- Número fixo de gerações.

- Convergência da população.
- Tempo de execução máximo.
- Ausência de melhoria após várias gerações.

3.2.6 Vantagens e Desvantagens

AGs são particularmente eficazes em problemas com espaços de busca complexos e funções objetivo não deriváveis ou descontínuas. Sua principal vantagem é a capacidade de explorar amplamente o espaço de soluções e escapar de ótimos locais.

Por outro lado, os AGs não garantem a obtenção da solução ótima global, e sua performance pode ser sensível à configuração dos parâmetros, como taxa de crossover, taxa de mutação e tamanho da população.

3.2.7 Aplicação de Algoritmos Genéticos à Alocação de Docentes

Como alternativa à abordagem exata da PLI, propõe-se a implementação de um Algoritmo Genético (AG), uma meta-heurística inspirada nos princípios da evolução natural. O AG é particularmente eficaz na exploração de espaços de solução vastos e complexos, como o da alocação de docentes, sendo capaz de encontrar soluções de alta qualidade em tempo computacional razoável. A seguir, detalha-se a adaptação dos componentes do AG para este problema.

3.2.8 Representação do Indivíduo (Cromossomo)

No contexto do AG, uma solução candidata é denominada *indivíduo* ou *cromossomo*. Para o problema de alocação, cada indivíduo representará uma grade de alocação completa e viável. A representação escolhida é a de um **vetor de alocação direta**, denotado por C .

- A estrutura do cromossomo C é um vetor de inteiros cujo comprimento é igual ao número total de disciplinas ofertadas, $|D|$.
- O índice do vetor, d , corresponde a uma disciplina específica do conjunto D (para $d \in \{0, 1, \dots, |D| - 1\}$).
- O valor contido na posição d do vetor, $C[d]$, é o identificador do professor $p \in P$ alocado para lecionar a disciplina d .

Um exemplo de cromossomo para um conjunto hipotético de 4 disciplinas seria: $C = [p_3, p_7, p_2, p_3]$, o que significa que o Professor 3 foi alocado para as disciplinas 0 e 3, o Professor 7 para a disciplina 1, e o Professor 2 para a disciplina 2. Esta representação garante, por construção, que cada disciplina seja alocada a exatamente um professor, satisfazendo nativamente a restrição da alocação única.

3.2.9 Função de Avaliação (Fitness Function)

A função de avaliação, ou *fitness function*, é o componente que mede a qualidade de um indivíduo (solução), guiando o processo de seleção. Uma pontuação de fitness mais alta indica uma solução melhor. A função proposta, $f(C)$, equilibra a maximização da preferência docente com a penalização de violações de restrições de carga horária.

A função de fitness para um cromossomo C é definida pela Equação 3.2.1:

$$f(C) = \left(\sum_{d=0}^{|D|-1} pref_{C[d],d} \right) - \left(W_{ch} \cdot \sum_{p \in P} \text{Excesso}_p(C) \right) \quad (3.2.1)$$

Os componentes da função são:

- **Escore de Preferência:** O primeiro termo, $\sum pref_{C[d],d}$, é a soma direta das preferências de cada alocação presente no cromossomo C , análogo ao objetivo principal do modelo de PLI.
- **Penalidade por Carga Horária:** O segundo termo penaliza o excesso de carga horária. W_{ch} é um coeficiente de penalidade escalar, e $\text{Excesso}_p(C)$ é o total de créditos que excedem a carga horária máxima do professor p na solução C , calculado conforme a Equação 3.2.2.

$$\text{Excesso}_p(C) = \max \left(0, \left(\sum_{d|C[d]=p} ch_d \right) - CH_{max,p} \right) \quad (3.2.2)$$

- **Penalidade por Conflito de Horário:** O terceiro termo, de forma análoga, penaliza conflitos de horário. W_{hor} é um coeficiente de penalidade específico para esta violação, e $\text{Cf}_p(C)$ calcula o número de conflitos de horário para o professor p na solução C .

3.2.10 Operadores Genéticos Aplicados

Os operadores genéticos são os mecanismos que promovem a evolução da população de soluções a cada geração.

Seleção.

Existem diversas formas de selecionar os indivíduos que seguirão para a próxima etapa, sendo os 3 principais: Seleção por Roleta, Seleção Por Ranking e Seleção por Torneio. A seleção por roleta possui a implementação mais simples, em que cada indivíduo é avaliado e a sua probabilidade de seleção é proporcional ao seu escore, no entanto essa forma de seleção apresenta um alto risco de convergência prematura, pois soluções razoáveis podem acabar dominando o processo de seleção nas primeiras iterações, acabando com a diversidade da solução. A seleção por Ranking apresenta uma correção da seleção por Roleta, utilizando a classificação de cada solução para determinar a sua probabilidade de seleção, o que reduz a dominância das soluções inicialmente melhores, mas introduz um grande aumento na complexidade computacional do problema, que requer que a cada rodada todos os indivíduos sejam avaliados e ordenados. A terceira alternativa apresenta uma convergência mais cuidadosa, o que a torna mais lenta, mas que resolve o problema de forma satisfatória, e portanto foi a forma de seleção adotada para o problema. O método de seleção por Torneio consiste em:

1. Sortear aleatoriamente k indivíduos da população (um "torneio" de tamanho k , $k=3$ por exemplo).
2. Avaliar o fitness dos k indivíduos selecionados.
3. O indivíduo com o maior valor de fitness é selecionado para reprodução, e eles retornam para a população.
4. Repetir o processo até que o número desejado de pais seja selecionado.

Este método é eficiente e tende a evitar a convergência prematura que pode ocorrer em outros métodos, como o da roleta.

Crossover (Reprodução).

O operador de crossover combina o material genético de dois pais (P_1 e P_2) para gerar um ou mais descendentes (O_1, O_2). Será utilizado o **crossover de um ponto**.

1. Um ponto de corte aleatório, k , é sorteado ao longo do comprimento do cromossomo (onde $1 \leq k < |D|$).
2. O primeiro descendente (O_1) é formado pela combinação dos primeiros k genes de P_1 com os genes restantes de P_2 .
3. O segundo descendente (O_2) é formado pela combinação dos primeiros k genes de P_2 com os genes restantes de P_1 .

Exemplo: Se $P_1 = [p_A, p_B | p_C, p_D]$ e $P_2 = [p_E, p_F | p_G, p_H]$, com corte em $k = 2$, os descendentes seriam $O_1 = [p_A, p_B, p_G, p_H]$ e $O_2 = [p_E, p_F, p_C, p_D]$.

Mutação.

A mutação introduz diversidade na população, alterando aleatoriamente um ou mais genes de um indivíduo com uma baixa probabilidade, μ (taxa de mutação). Será aplicada a **mutação de gene único**:

1. Para cada gene (posição d) em um cromossomo recém-gerado, um número aleatório é sorteado.
2. Se este número for menor que a taxa de mutação μ , o gene sofre uma mutação.
3. A mutação consiste em substituir o professor atualmente alocado, $C[d] = p$, por um novo professor, p' , sorteado aleatoriamente do conjunto de todos os professores P .

3.3 Algoritmos de Colônias de Formigas

Os Algoritmos de Colônia de Formigas (Ant Colony Optimization - ACO) são meta-heurísticas inspiradas no comportamento de formigas reais na busca por caminhos entre a colônia e fontes de alimento. Propostos inicialmente por Dorigo e Gambardella (1997), os ACOs têm sido aplicados com sucesso em diversos problemas de otimização combinatória.

3.3.1 Princípios Fundamentais

A ideia central do ACO é simular o processo de construção de soluções por agentes artificiais (*formigas*), que exploram o espaço de soluções de um problema, depositando e seguindo trilhas de feromônio que indicam a qualidade das soluções encontradas.

3.3.2 Componentes do ACO

- **Representação do Problema:** O problema é modelado como um grafo onde os nós representam componentes da solução e as arestas representam possíveis transições entre esses componentes.
- **Trilha de Feromônio (τ):** Cada aresta do grafo possui um valor de feromônio que influencia a probabilidade de ser escolhida pelas formigas.
- **Heurística (η):** Informação adicional sobre o problema que pode guiar as formigas na construção de soluções mais promissoras.
- **Função de Transição:** A probabilidade p_{ij}^k de uma formiga k se mover do nó i para o nó j é dada por:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta}$$

onde α e β são parâmetros que controlam a influência do feromônio e da heurística, respectivamente, e \mathcal{N}_i^k é o conjunto de nós vizinhos de i ainda não visitados pela formiga k .

- **Atualização de Feromônio:** Após todas as formigas construírem suas soluções, os valores de feromônio são atualizados para reforçar boas soluções e evaporar feromônios antigos:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$$

onde ρ é a taxa de evaporação e $\Delta \tau_{ij}^k$ é a quantidade de feromônio depositada pela formiga k na aresta (i, j) .

3.3.3 Procedimento Geral do ACO

1. Inicializar os parâmetros do algoritmo e os valores iniciais de feromônio.
2. Para cada iteração:
 - (a) Cada formiga constrói uma solução completa baseada nas probabilidades de transição.
 - (b) Avaliar a qualidade das soluções construídas.
 - (c) Atualizar os valores de feromônio com base nas soluções encontradas.

3. Repetir até que um critério de parada seja satisfeito (número máximo de iterações, tempo limite, convergência, etc.).

3.4 Aplicação de Algoritmos de Colônia de Formigas (ACO)

No problema de alocação de docentes, as "formigas" serão agentes computacionais que construirão soluções completas. Cada passo está associado à uma disciplina, e o caminho formado representa a seleção de professores que ela fez. E o "feromônio" representará a qualidade aprendida das decisões de alocação.

3.4.1 Estrutura do Problema e Representação

Conforme a análise de abordagens, o problema será modelado segundo a estrutura de **Construção Sequencial**. Neste modelo, a construção de uma solução (uma grade de alocação completa) é decomposta em uma sequência de $|D|$ decisões, uma para cada disciplina a ser alocada. Cada formiga artificial do algoritmo executará este processo construtivo para gerar uma solução candidata a cada iteração.

3.4.2 Componentes do Modelo ACO

A seguir, são detalhados os componentes fundamentais do ACO e sua adaptação para o problema em questão.

Construção da Solução pelas Formigas.

Cada formiga m constrói uma solução completa S_m , que é um conjunto de pares (disciplina, professor). O processo é iterativo:

1. Para cada disciplina $d_k \in D$, onde $k = 1, \dots, |D|$, a formiga deve escolher um professor p do conjunto de professores P .
2. A escolha é probabilística e guiada pela trilha de feromônio e pela informação heurística.
3. **Construção Consciente das Restrições:** Antes de uma formiga escolher um professor para a disciplina d_k , ela avalia a viabilidade daquela alocação. Um professor p só é considerado um candidato elegível se a adição da carga horária ch_{d_k}

não exceder sua carga horária máxima ($CH_{max,p}$), considerando as alocações que a mesma formiga já realizou nos passos anteriores. Isso garante que toda solução gerada seja, por construção, viável no que tange à carga horária.

Informação Heurística (η).

A informação heurística, ou atratividade *a priori*, de se alocar o professor p para a disciplina d , é uma informação local que auxilia na decisão da formiga. Para este problema, a preferência do docente pela disciplina é uma excelente medida heurística. Assim, define-se:

$$\eta_{dp} = pref_{pd} + \epsilon \quad (3.4.1)$$

Onde $pref_{pd}$ é a preferência do professor p pela disciplina d . Adiciona-se uma pequena constante $\epsilon > 0$ (e.g., $\epsilon = 0.1$) para evitar que uma preferência nula anule a probabilidade de escolha, permitindo uma exploração mínima do espaço de soluções.

Trilha de Feromônio (τ).

A trilha de feromônio τ_{dp} está associada a cada possível alocação do par (d, p) . Este valor representa a "memória" coletiva da colônia, indicando o quanto desejável foi, em iterações passadas, alocar o professor p para a disciplina d como parte de uma solução global de alta qualidade. No início do algoritmo, todas as trilhas de feromônio são iniciadas com um valor pequeno e constante.

Regra de Transição Probabilística.

No passo k da construção, ao decidir qual professor alocar para a disciplina d_k , uma formiga m escolhe o professor p dentre um conjunto de professores viáveis \mathcal{N}_k^m com base na probabilidade definida pela Equação 3.4.2.

$$p(p|d_k) = \begin{cases} \frac{[\tau_{d_k p}]^\alpha \cdot [\eta_{d_k p}]^\beta}{\sum_{j \in \mathcal{N}_k^m} [\tau_{d_k j}]^\alpha \cdot [\eta_{d_k j}]^\beta} & \text{se } p \in \mathcal{N}_k^m \\ 0 & \text{caso contrário} \end{cases} \quad (3.4.2)$$

Onde:

- \mathcal{N}_k^m é o conjunto de professores elegíveis para a disciplina d_k na visão da formiga m (i.e., aqueles cuja alocação não viola as restrições de carga horária nem as de conflito de horário).

- α é um parâmetro que controla a influência da trilha de feromônio (a experiência passada da colônia).
- β é um parâmetro que controla a influência da informação heurística (a preferência do professor).

Atualização de Feromônio.

Após todas as formigas terem construído suas soluções, as trilhas de feromônio são atualizadas. Este processo tem duas etapas: evaporação e depósito.

1. **Evaporação:** Uma fração ρ (taxa de evaporação, $0 < \rho \leq 1$) do feromônio é evaporada de todas as trilhas. Isso permite o ”esquecimento” de soluções ruins e incentiva a exploração de novas partes do espaço de busca.

$$\tau_{dp} \leftarrow (1 - \rho) \cdot \tau_{dp}, \quad \forall (d, p) \in D \times P \quad (3.4.3)$$

2. **Depósito:** As formigas que construíram as melhores soluções depositam uma quantidade de feromônio nas alocações que utilizaram. O valor depositado, $\Delta\tau_{dp}$, é proporcional à qualidade (fitness) da solução.

$$\Delta\tau_{dp}^m = \begin{cases} Q/f(S_m) & \text{se a formiga } m \text{ alocou } (d, p) \\ 0 & \text{caso contrário} \end{cases} \quad (3.4.4)$$

Onde $f(S_m)$ é a função de fitness da solução S_m da formiga m (pode-se utilizar a mesma função da Equação 3.2.1), e Q é uma constante de desempenho. A atualização completa é dada por: $\tau_{dp} \leftarrow \tau_{dp} + \sum_m \Delta\tau_{dp}^m$.

3.5 Delineamento Experimental

Para fazer a otimização dos parâmetros dos algoritmos e validar o seu desempenho, foi elaborado um esquema experimental completo. Esses experimentos tem como objetivo avaliar a qualidade e consistência de cada um dos algoritmos, a sensibilidade dos modelos para variações nos seus hiperparâmetros e a eficiência computacional em diferentes cenários.

3.5.1 Exploração do Espaço de Parâmetros

O algoritmo de colônia de formigas possui 5 parâmetros principais: número de gerações, número de formigas, Alfa, Beta e Taxa de Evaporação. O algoritmo genético possui 6 parâmetros: número de gerações, tamanho da população, taxa de crossover, mutação, elitismo e tamanho do torneio de seleção. Devido à alta dimensionalidade dos dados, a navegação pelo espaço de possibilidades não é trivial, portanto, para poder explorar o espaço de busca de forma eficiente e uniforme, utilizou-se o método de amostragem por Hipercubo Latino (*Latin Hypercube Sampling - LHS*).

O LHS é uma técnica de amostragem estratificada que divide o intervalo de cada parâmetro em N subintervalos de mesma probabilidade, garantindo que amostras sejam selecionadas a partir de combinações dos subintervalos sem sobreposição, ou seja, cada subintervalo de um parâmetro gera apenas 1 amostra que é formada combinando com valores gerados dos subintervalos de outras posições dos outros parâmetros. Isso permite uma cobertura muito mais uniforme do espaço de parâmetros do que a amostragem aleatória simples, e sendo muito mais parcimoniosa e eficiente do que o *Grid Search* que testa todas as combinações de valores.

Para esse estudo, foram geradas 1000 configurações distintas (amostras), onde cada configuração representa um conjunto único de hiperparâmetros para o AG e o ACO. Isso permite analisar com as diferentes combinações influenciam a convergência e a qualidade final da alocação. Os intervalos definidos para exploração seguem apresentados abaixo:

Tabela 1: Limites de busca dos parâmetros do ACO

Parâmetro	Limite Inferior	Limite Superior
Nº de Formigas	10	100
Alfa	0.1	5
Beta	1	10
Taxa de Evaporação	0.01	0.6

O número de formigas utilizado na busca ajuda a aumentar o número de possibilidades que são exploradas ao mesmo tempo, é ideal aumentar para problemas com muitos mínimos locais para utilizar a diversidade para evitá-los, no entanto, o aumento do nº de formigas faz com que o número de operações realizadas a cada iteração aumente de forma acelerada, portanto precisamos localizar o menor número de formigas que consiga

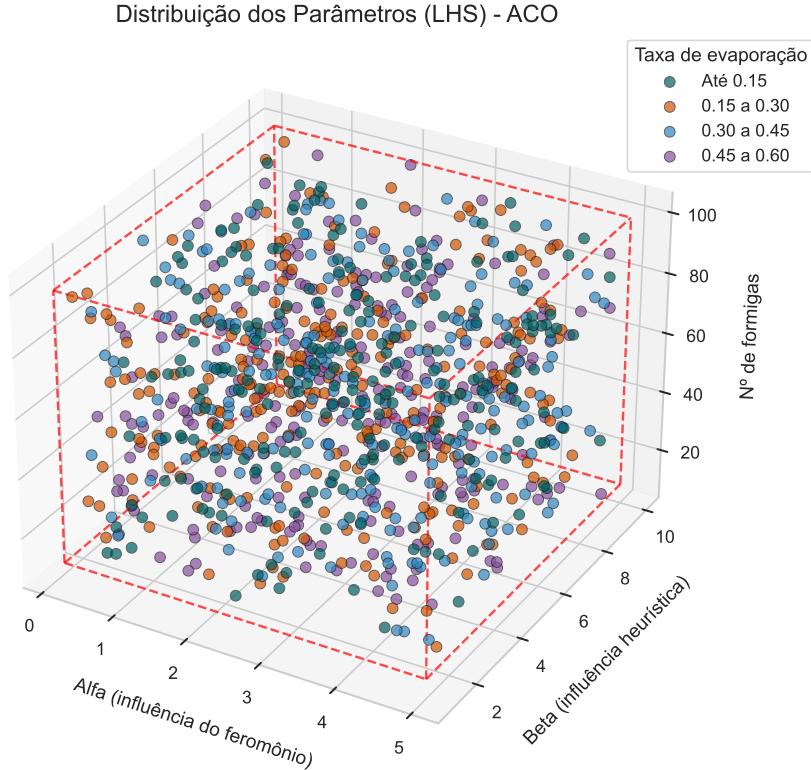
resolver o problema em questão.

Os parâmetros alfa e beta controlam o comportamento individual de cada formiga. Com um valor alfa baixo, a formiga não usa a informação do aprendizado coletivo e só realiza decisões individuais, no entanto, quando alfa é muito elevado as formigas não exploram individualmente e convergem rapidamente para o primeiro mínimo local que encontrarem. Esse intervalo abrange os 2 extremos que podem ser explorados, para encontrar um ponto de equilíbrio ideal no meio. O parâmetro beta reforça a heurística, ou seja, o conhecimento prévio sobre o problema utilizado para direcionar as soluções, nesse caso a heurística é implementada através do sorteio de professores com probabilidade proporcional à preferência para lecionar a disciplina, apenas quando não excederem a carga horária máxima ou tiverem algum conflito. Então ter um valor elevado para a o beta é essencial para que as formigas construam soluções factíveis e já direcionadas, mas o parâmetro não pode ser tão elevado para não impedir o aprendizado coletivo da melhor solução dentre as possíveis.

E a taxa de evaporação controla a velocidade de aprendizado da colônia, caso o valor seja muito baixo, o conhecimento global vai se acumulando lentamente, e quando é muito alto faz com que o esquecimento seja muito acelerado e o algoritmo se torna quase uma busca aleatória, por isso é necessário balancear para ter conhecimento suficiente para direcionar para as melhores soluções, mas esquecer rápido o suficiente para não ficar preso nas escolhas ruins.

Segue uma representação visual da exploração de espaços feita pelo LHS:

Figura 1: Gráfico de Dispersão 3D - Parâmetros LHS do ACO



Com o gráfico, podemos ver que há uma distribuição homogênea que ocupa os espaços de exploração de forma uniforme. Permitindo com que a análise de sensibilidade explore as melhores regiões de configuração dos parâmetros.

E para explorar as configurações do AG, as seguintes regiões foram exploradas:

Tabela 2: Limites de busca dos parâmetros do AG

Parâmetro	Limite Inferior	Limite Superior
Tamano da População	30	300
Taxa de Crossover	0.5	0.95
Taxa de Mutação	0	0.3
Tamanho do Torneio	3	15
Proporção de Elitismo	0	0.4

O tamanho da população no algoritmo genético também é essencial, no entanto, as iterações desse algoritmo são mais simples e rápidas, por isso podemos utilizar valores maiores. O objetivo é o mesmo do ACO, garantir diversidade com o menor custo computacional possível. Em geral populações com menos de 30 indivíduos terão uma con-

vergênci a muito prematura, e um tamanho populacional acima de 300 torna o algoritmo computacionalmente caro.

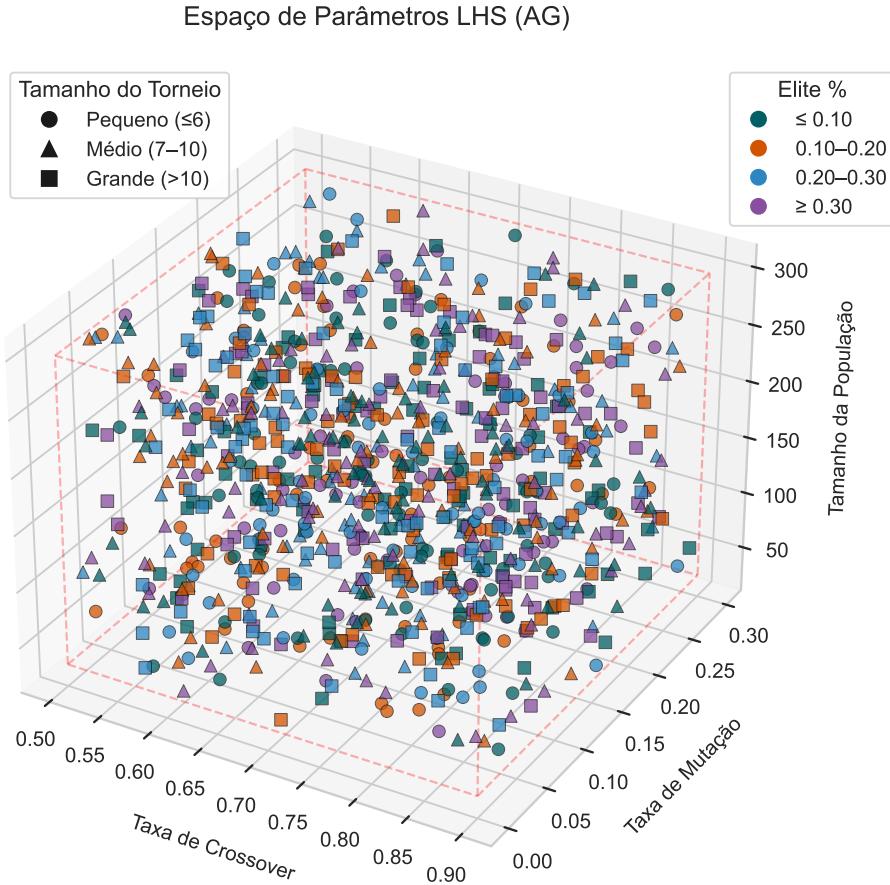
A taxa de crossover é essencial que seja alta para caracterizar os algoritmos genéticos, a grande vantagem do método é a combinação de blocos das boas soluções, portanto a taxa de cruzamento deve ser alta. Caso seja abaixo de 0.5, o algoritmo começa a perder a característica de recombinação e passa a depender da sobrevivência e mutação.

A taxa de mutação é muito importante para acrescentar diversidade às soluções, permitindo escapar dos mínimos locais, e pode ser a única forma de transformar alguma solução inválida em válida, no entanto, se for muito elevada o algoritmo se torna uma busca aleatória, e as mutações começam a piorar as boas soluções que estão convergindo.

O tamanho do torneio controla a pressão seletiva, quando um torneio é feito com poucos indivíduos há mais chance que um indivíduo ruim sobreviva para as próximas etapas, preservando a diversidade e atrasando a convergência. Já torneios muito grandes podem convergir muito rapidamente para um mínimo local.

E a proporção de elitismo auxilia também a acelerar a convergência, ideal para quando as soluções geradas são boas, mas pode acabar piorando ao prejudicar a diversificação das soluções e favorecer apenas as com melhores pontuações locais.

Figura 2: Gráfico de Dispersão 3D - Parâmetros LHS do AG



Podemos observar que o espaço combinado das variáveis fica completamente preenchido e combinado, apresentando as condições necessárias para a exploração e otimização dos parâmetros.

3.5.2 Cenários de Teste e Perturbação dos Dados

Os dados de referência desse trabalho são os de preferência e disciplinas ofertadas no semestre 25.1, mas para evitar que o ajuste dos parâmetros seja feito de forma viesada para esse cenário específico, precisamos introduzir uma variabilidade nos dados, para que os parâmetros sejam otimizados para lidar com cenários variados. No entanto, apesar de haver variação na disponibilidade dos professores, suas preferências pelas disciplinas e nos horários ao longo do semestre, as mudanças também não são completamente radicais, há pouca variabilidade nos dados, em grande parte pela própria natureza. E uma das maiores dificuldades desse trabalho foi simular esse comportamento aleatório.

Para simular essa variabilidade, cada uma das 1000 simulações foi executada sobre uma instância de dados levemente modificada a partir dos dados originais. As perturbações aplicadas foram:

- **Conflitos de Horários entre as disciplinas**

- **Justificativa:** A restrição de conflito de horários é uma das mais importantes e que não pode ser violada em nenhum cenário, portanto variar as disciplinas que têm conflito entre si é essencial para ter algoritmos robustos.
- **Metodologia:** A proporção de disciplinas que terão seu horário alterado é obtido partindo de um $P \sim \mathcal{N}(0.2, 0.2)$ limitada no 0 e no 1. A partir dos índices das disciplinas que serão alteradas, é sorteado um novo horário para ela baseada no tipo de disciplina (Regular ou de Serviço) e na carga horária dela, para que só sejam feitas alterações compatíveis com as práticas do departamento. E com base nos novos horários, a matriz de conflitos é recalculada.
- **Métricas:** As medidas utilizadas para avaliar a mudança em relação aos dados originais são: número de conflitos e o número de disciplinas alteradas.

A função elaborada pode ser verificada no apêndice: A.1.

- **Preferências dos Docentes pelas Disciplinas**

- **Justificativa:** A distribuição de preferência dos docentes pelas disciplinas altera completamente o cenário de otimização. No cenário atual, cada disciplina em geral possui um número reduzido de professores que tem preferência por ela, facilitando a otimização que deve apenas escolher um dos poucos professores que tem interesse e que não possui conflito de horário, portanto ao introduzir aleatoriedade nesse perfil de escolha, os algoritmos terão que se adaptar para lidar com cenários de matérias mais disputadas entre os professores.
- **Metodologia:** A alteração nos dados de preferências é feita da seguinte forma: é amostrado um $P \sim \mathcal{N}(0.1, 0.1)$, que define a proporção de preferências do total (23 professores x 47 turmas) que será alterada. Essa variação é feita re-sorteando cada uma das preferências pelos valores possíveis, com as probabilidades iguais à frequência delas no dataframe original, acrescidas de uma perturbação aleatória com distribuição $\mathcal{N}(\delta, \delta^2)$, para que vários tipos de cenários diferentes sejam gerados.
- **Métricas:** A principal métrica observada é o número médio de professores interessados em cada disciplina. Ou seja, para cada turma é calculado o número

de professores que colocaram preferência 3 para a turma específica, depois é tirada a média entre todas as turmas de uma mesma disciplina, e dividimos a média de interesses na disciplina pelo número de turmas, para obter o número de professores interessados por turma para cada disciplina. E por último, calculamos a média dessa coluna obtendo um número que representa o número médio de professores por turma que tem interesse em cada disciplina.

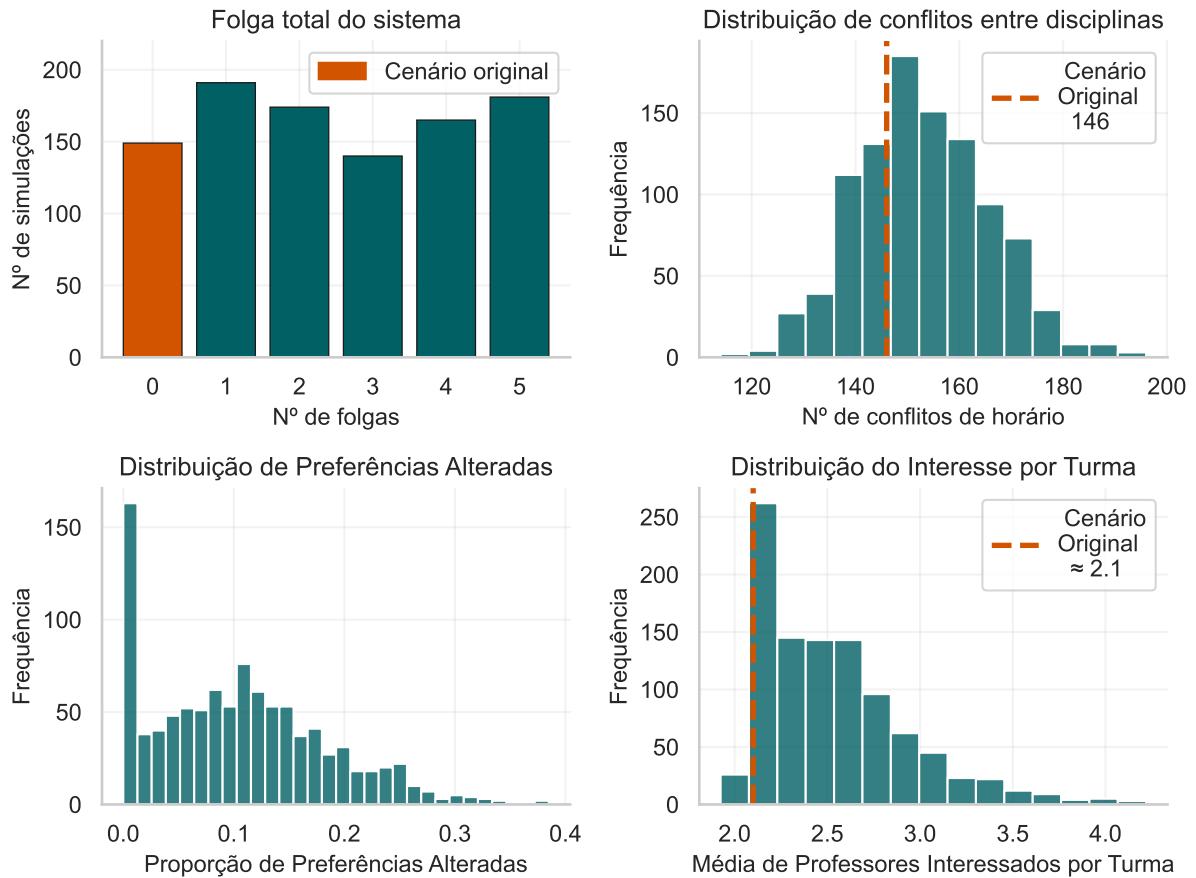
A função elaborada pode ser verificada no apêndice: A.2.

• Disponibilidade dos professores

- **Justificativa:** A disponibilidade dos professores define a folga do sistema, se a soma das disponibilidades dos professores for maior do que o número de turmas, há folga no sistema e os algoritmos podem fazer trocas de forma mais simples sem se preocupar com penalização, mas quando há pouca ou nenhuma folga as trocas devem ser muito mais cuidadosas, pois podem piorar muito uma solução.
- **Metodologia:** Como os dados base foram definidos para que não houvesse folga no sistema a implementação ficou mais simples. O número de folgas que serão adicionadas é sorteado de 0 a 6, e para cada folga adicionada no sistema é acrescentada mais 1 disciplina no limite de algum dos docentes que está com máximo definido como 1.
- **Métricas:** A única métrica observada é o número de folgas adicionadas, que é o número de folgas do sistema. Para que seja possível estudar como ele se comporta para diferentes configurações. *A função elaborada pode ser verificada no apêndice: A.3.*

Segue abaixo o resultado das perturbações utilizadas na simulação para análise de sensibilidade dos parâmetros:

Figura 3: Distribuição dos dados após perturbações aleatórias



O gráfico 3 mostra o resultado das modificações feitas, introduzindo uma aleatoriedade que varia levemente o perfil dos dados ao redor do cenário original que temos, mas o seu principal ganho é a variação na própria configuração de preferências e horários que ajuda a selecionar os parâmetros que funcionem bem para configurações mais amplas dos dados.

3.5.3 Métricas de Avaliação e Ambiente Computacional

Os algoritmos foram avaliados com base em 3 métricas principais:

1. **Qualidade da Solução:** Soma ponderada do número de preferências atendidas;
2. **Tempo Até Convergência:** Mensurado utilizando o *CPU Time* (tempo de processador), para isolar o consumo de recursos do algoritmo de interferências do sistema operacional;

3. **Convergência:** Análise do número de gerações necessárias para atingir a melhor solução ou estabilizar.

Os experimentos foram realizados utilizando a linguagem Python (Versão 3.13.0) e as bibliotecas PuLP (para o modelo exato PLI) e NumPy/Pandas (para a manipulação de dados e implementação das meta-heurísticas).

4 Resultados

4.1 Caracterização do Cenário de Estudo

Para a definição do problema e avaliação dos resultados, iremos considerar os dados da consulta de preferência realizada com os professores para o semestre 2025.1 e a oferta disponível no SIGAA para o mesmo semestre.

O questionário de consulta da oferta é organizado da seguinte forma:

	1	2	3	Em princípio, não tenho interesse.
Análise de Regressão Linear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Computação em Estatística 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delineamento e Análise de Experimentos 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estatística Exploratória	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Inferência Estatística	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Introdução à Probabilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Métodos Estatísticos 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Métodos Estatísticos 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Técnicas de Amostragem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
TCC1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Em que cada professor avalia todas as matérias ofertadas pelo departamento de acordo com a sua preferência. Para este trabalho, a divisão dos pontos da preferência foi reorganizada, de forma que: 0 representa o não interesse, e o 3 foi considerado como preferência máxima.

E comparando os dados da oferta com a pesquisa por interesse, apenas 23 dos 28 professores haviam respondido o questionário. Dos 5 que não responderam, 1 ocupava a posição de coordenador, então foi adicionado ao arquivo de preferências com preferência

3 pela disciplina TCC1 e 0 nas demais, e os demais professores foram considerados como professores substitutos, contratados para preencher as disciplinas de serviço do departamento, portanto ficando com preferência 3 pelas turmas de serviço e 0 pelas demais.

4.1.1 Análise da Procura por Disciplinas

Para analisar a preferência dos professores pelas disciplinas, foi feita uma contagem de quantas vezes cada disciplina obteve cada pontuação de preferência pelos professores, e o seguinte resultado foi obtido:

Tabela 3: Distribuição das preferências por disciplina

Disciplina	Nº de Turmas	Pref 0	Pref 1	Pref 2	Pref 3
Analise de dados categorizados	1	25	2	0	1
Analise de regressao linear	1	21	2	2	3
Analise de series temporais	1	24	2	0	2
Analise de sobrevivencia	1	25	1	0	2
Bioestatistica	2	1	13	5	9
Computacao em estatistica 1	2	21	3	2	2
Computacao em estatistica 2: python	1	25	0	1	2
Computacao em estatistica 2: r	1	24	2	2	0
Delineamento e analise de experimentos	1	25	2	0	1
Demografia	1	27	1	0	0
Estatistica aplicada	10	1	12	4	11
Estatistica computacional	1	23	2	1	2
Estatistica exploratoria	1	21	5	0	2
Inferencia bayesiana	1	26	1	0	1
Inferencia estatistica	1	21	2	1	4
Introducao a probabilidade	2	13	5	2	8
Laboratorio de estatistica 1	1	26	0	2	0
Metodos estatisticos 1	1	19	3	4	2
Metodos estatisticos 2	1	20	4	2	2
Modelos lineares generalizados	1	25	1	0	2
Praticas de extensao em estatistica 2	1	27	0	1	0
Probabilidade e estatistica	11	1	8	2	17
Processos estocasticos	1	20	1	3	4
Tecnicas de amostragem	1	26	1	0	1
Trabalho de conclusao de curso 1	1	27	0	0	1

Podemos observar pela tabela 3 que, de forma geral, as disciplinas possuem poucos professores com interesse. Com exceção das disciplinas de serviço do departamento (Probabilidade e Estatística, Bioestatística e Estatística Aplicada) que praticamente não possuem nenhuma indicação de não interesse em lecionar. Além disso, outra tendência que pode ser observada é que as disciplinas iniciais do curso também possuem um interesse um pouco mais elevado do que as demais, e conforme a especificidade da disciplina aumenta, o número de professores com interesse diminui. Mostrando que no departamento há uma alta especialização nas áreas de cada disciplina e existem poucas opções com interesse em cada disciplina. Além disso, se destacam algumas disciplinas obrigatórias do currículo com pouco ou nenhum interesse pelos professores, como: Demografia, Análise de Dados Categorizados, Inferência Bayesiana e Técnicas de Amostragem.

Quase todas as disciplinas possuem um número de professores com alto interesse maior ou igual ao número de turmas ofertadas, criando a possibilidade da construção de alocações que cada turma seja lecionada por um professor que possui alto interesse, gerando pontuações mais altas na avaliação. O valor máximo teórico de pontuação seria de 141 (47×3) que seria obtido se todas as disciplinas fossem alocadas para professores com preferência nível 3. Mas como há 4 disciplinas que não possuem preferência 3, e uma delas possui apenas preferência 1, o valor máximo que pode ser obtido é de 136 pontos. No entanto, esse cenário pode não ocorrer na prática por conflitos de horários entre disciplinas com pouco interesse, selecionadas pelos mesmos docentes.

Podemos também analisar quantas vezes cada docente marcou alguma disciplina com uma preferência específica:

Tabela 4: Distribuição das preferências por docente

Docente	Pref 0	Pref 1	Pref 2	Pref 3
docente_01	23	10	11	3
docente_02	20	2	15	10
docente_03	24	12	0	11
docente_04	19	2	14	12
docente_05	21	12	1	13
docente_06	22	11	1	13
docente_07	11	5	4	27
docente_08	14	27	5	1
docente_09	21	12	0	14
docente_10	24	0	0	23
docente_11	18	0	14	15
docente_12	19	15	10	3
docente_13	24	0	0	23
docente_14	23	23	0	1
docente_15	18	12	0	17
docente_16	24	0	0	23
docente_17	24	0	0	23
docente_18	13	20	2	12
docente_19	15	18	10	4
docente_20	46	0	0	1
docente_21	21	12	0	14
docente_22	19	14	3	11
docente_23	17	11	3	16
docente_24	24	0	0	23
docente_25	5	25	4	13
docente_26	16	14	2	15
docente_27	21	12	0	14
docente_28	22	13	0	12

As inserções feitas na tabela de preferências foram os docentes: 20 (coordenador), 13, 16, 17 e 24 (os professores substitutos).

A Tabela 4 conduz a análise para o perfil dos docentes. Observa-se uma variabilidade significativa no comportamento. Por exemplo, o Docente 12 manifestou desinteresse por 41 das 46 disciplinas, não indicando preferência máxima por nenhuma, o que o caracteriza como um perfil de alocação potencialmente mais restrito. Em contrapartida, os

Docentes 10 e 13 demonstram grande engajamento ou um leque de interesses mais amplo, marcando 27 e 32 disciplinas com o maior nível de preferência, respectivamente. Essa heterogeneidade no perfil docente é um fator chave de complexidade que a ferramenta de otimização deverá gerenciar.

4.1.2 Conclusão

No cenário considerado temos 28 docentes e 47 disciplinas (considerando turmas de uma mesma disciplina como disciplinas diferentes), e trabalhamos com a restrição de que cada docente deve lecionar no máximo 2 disciplinas. Portanto temos uma capacidade total de 56 alocações para uma demanda de 47 turmas, resultando em uma folga teórica de apenas 9 posições.

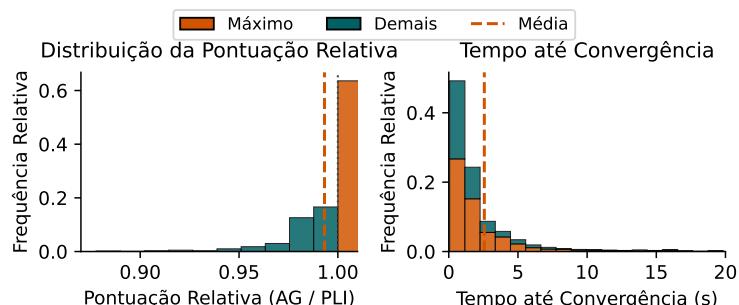
4.2 Análise e Calibração do Algoritmo de Colônia de Formigas

O objetivo do estudo de sensibilidade dos parâmetros é avaliar quais escolhas para os parâmetros vão gerar o melhor resultado de forma consistente, e irão realizar isso com o menor custo computacional. Para avaliar os resultados foi utilizada a pontuação relativa do algoritmo em relação à PLI, variando de 0 a 1. E para avaliar a eficiência computacional foi utilizado o tempo até a convergência do algoritmo.

4.2.1 Análise de sensibilidade dos hiperparâmetros

O resultado das simulações com perturbações e variando os parâmetros do ACO, seguem apresentados nos gráficos abaixo:

Figura 4: Histograma de pontuação relativa e tempo até convergência ACO



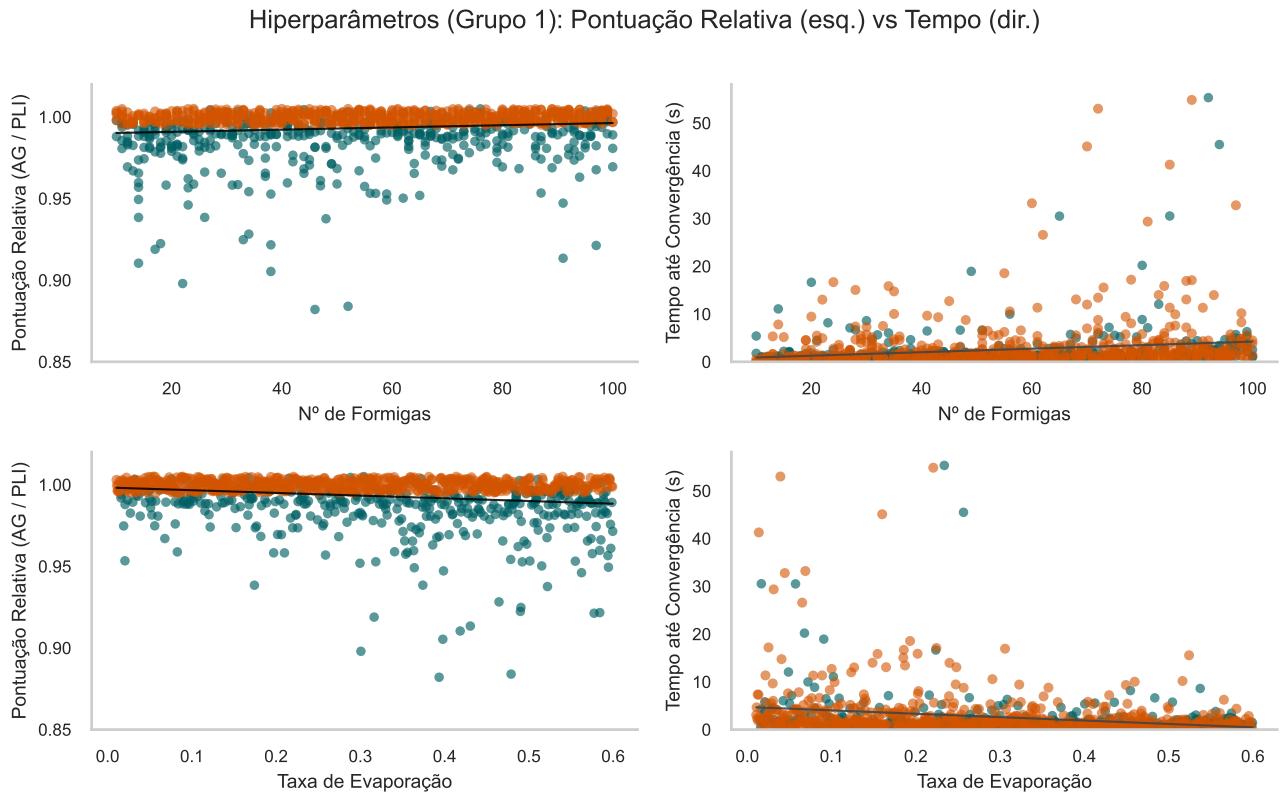
Das 1000 simulações, 636 apresentaram o escore máximo, e com mais de 80%

das amostras obtendo um escore relativo acima de 99%. E podemos observar que os tempos de convergência do algoritmo foram extremamente baixos, com apenas 41 das 1000 simulações obtendo um tempo de convergência acima de 10 segundos.

Para analisar a sensibilidade dos hiperparâmetros no valor da função objetivo, os seguintes gráficos foram construídos:

Foram elaborados gráficos de dispersão entre cada hiperparâmetro e a pontuação relativa, e ao lado a comparação com o tempo até convergência. As simulações que tiveram uma pontuação máxima foram marcadas de laranja para facilitar a análise. E uma perturbação aleatória foi adicionada no eixo Y para facilitar a visualização dos pontos que estão na mesma região.

Figura 5: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do ACO

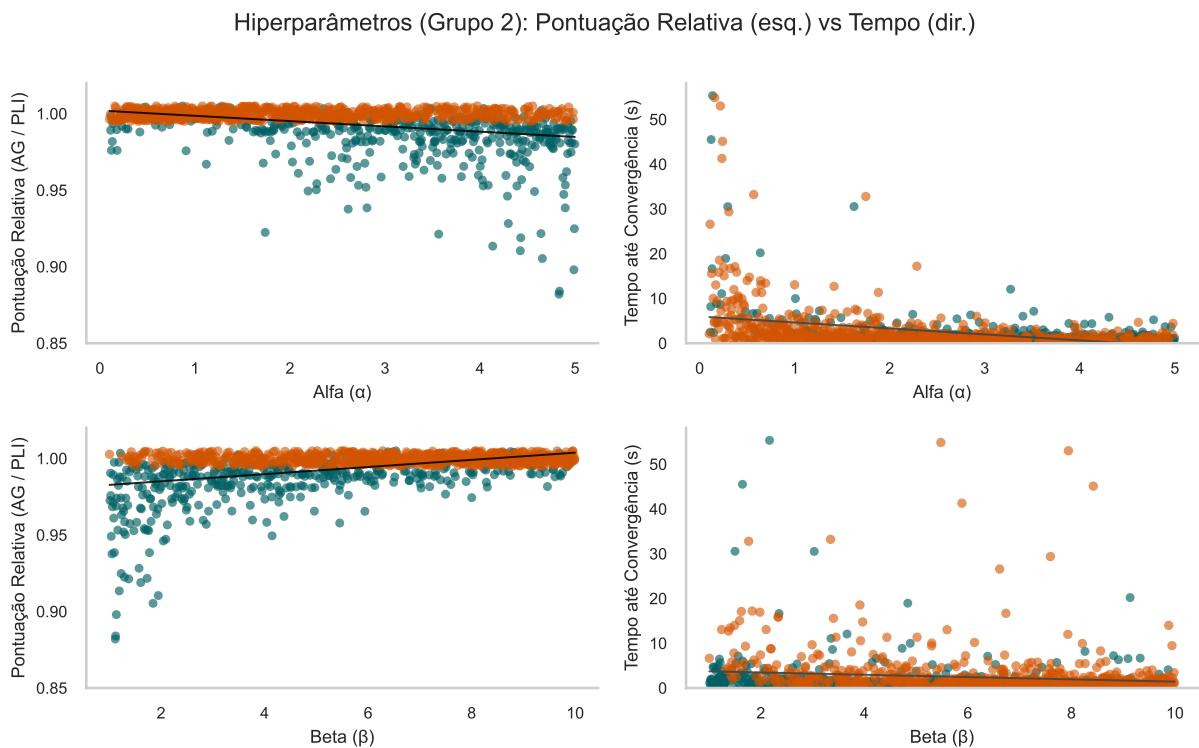


A partir da análise do gráfico 5 podemos observar que não há uma relação muito forte entre número de formigas e pontuação, com as falhas se apresentando distribuídas aleatória ao longo do intervalo. E no tempo computacional, o resultado foi de acordo com o esperado, que o aumento no número de formigas faz com que o número de operações a cada geração aumente também, mas não há a necessidade do ganho de robustez aumentando

o tamanho da população. Podendo portanto utilizaremos 30 formigas para as próximas etapas.

A taxa de evaporação apresenta uma relação negativa fraca com a pontuação, com taxas de evaporação mais baixas apresentando uma variabilidade menor na pontuação, além de uma menor proporção de falhas. E no tempo computacional, o efeito parece se inverter um pouco, mas a relação é fraca pelos bons resultados da grande maioria dos pontos. Podendo então nos permitir fixar a taxa de evaporação em 0.1.

Figura 6: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do ACO



Podemos observar que os parâmetros alfa e beta possuem um efeito muito maior no resultado da performance do algoritmo. Há indícios fortes de que conforme o alfa aumenta, os resultados pioram drasticamente. E com beta há a tendência oposta. O que significa que o

alfa = 1.5 beta = 8

4.2.2 Análise de Convergência

Após as análises, os seguintes valores foram fixados para os parâmetros do ACO:

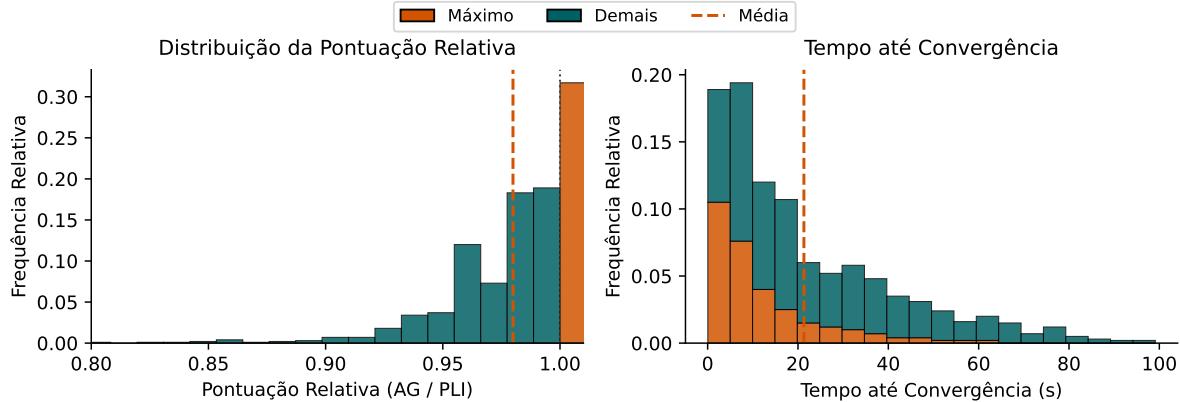
Tabela 5: Valores selecionados para os parâmetros do ACO

Parâmetro	Valor
Número de Gerações	100
Número de Formigas	30
Taxa de Evaporação	0.1
Alfa	1.5
Beta	8

4.3 Análise e Calibração do Algoritmo Genético

4.3.1 Análise de sensibilidade dos hiperparâmetros

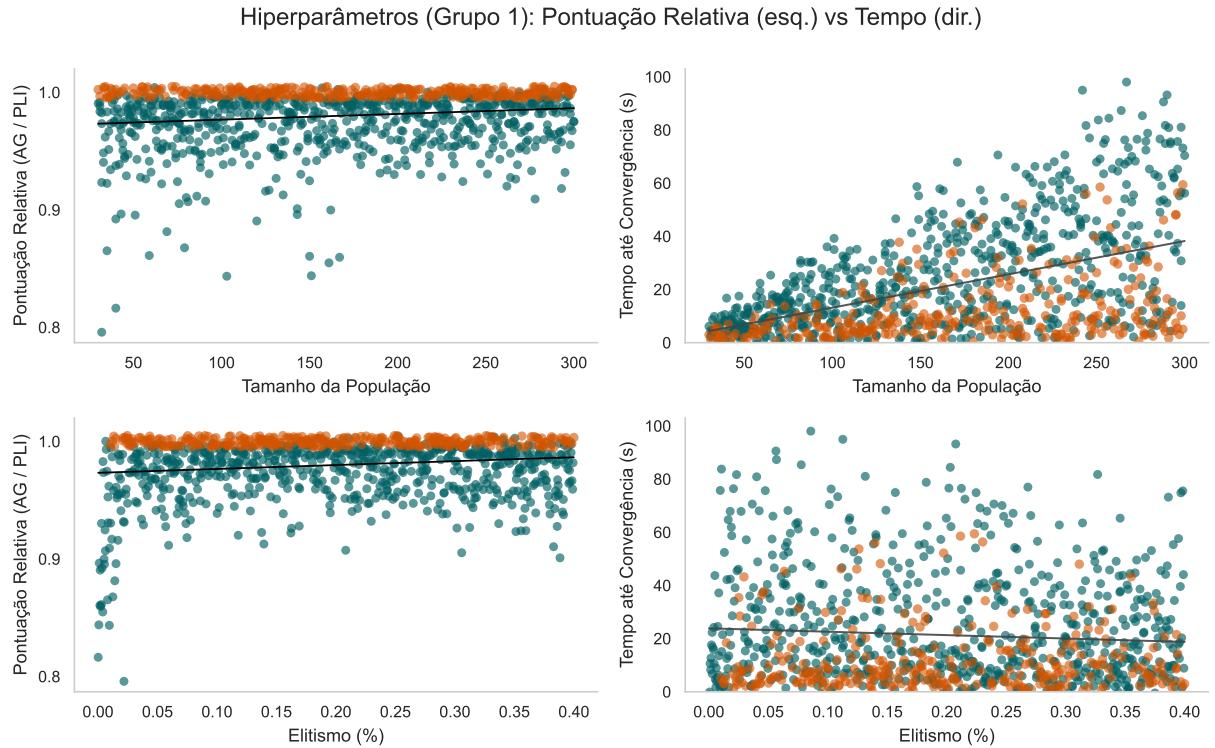
Figura 7: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG



Analizando os resultados do AG, podemos observar que a convergência do algoritmo não foi tão forte e expressiva quanto a do ACO, no entanto, os resultados ainda são bem promissores e apenas indicam que o algoritmo é mais sensível para a escolha dos parâmetros do que o ACO, ou que a escolha dos intervalos foi mais abrangente, mas ainda indica que há o potencial de se obter performances tão boas quanto as do ACO.

Observando os gráficos em Figura 7 podemos observar que mais de 30% das simulações convergiram para o valor máximo, e as demais se concentram em regiões próximas, com mais de 50% obtendo um valor acima de 99% do valor máximo. E analisando o tempo de convergência, podemos observar que as simulações que convergiram para o valor máximo apresentam uma distribuição mais concentrada nos menores tempos, enquanto as que não convergiram se apresentam mais distribuídas ao longo do intervalo.

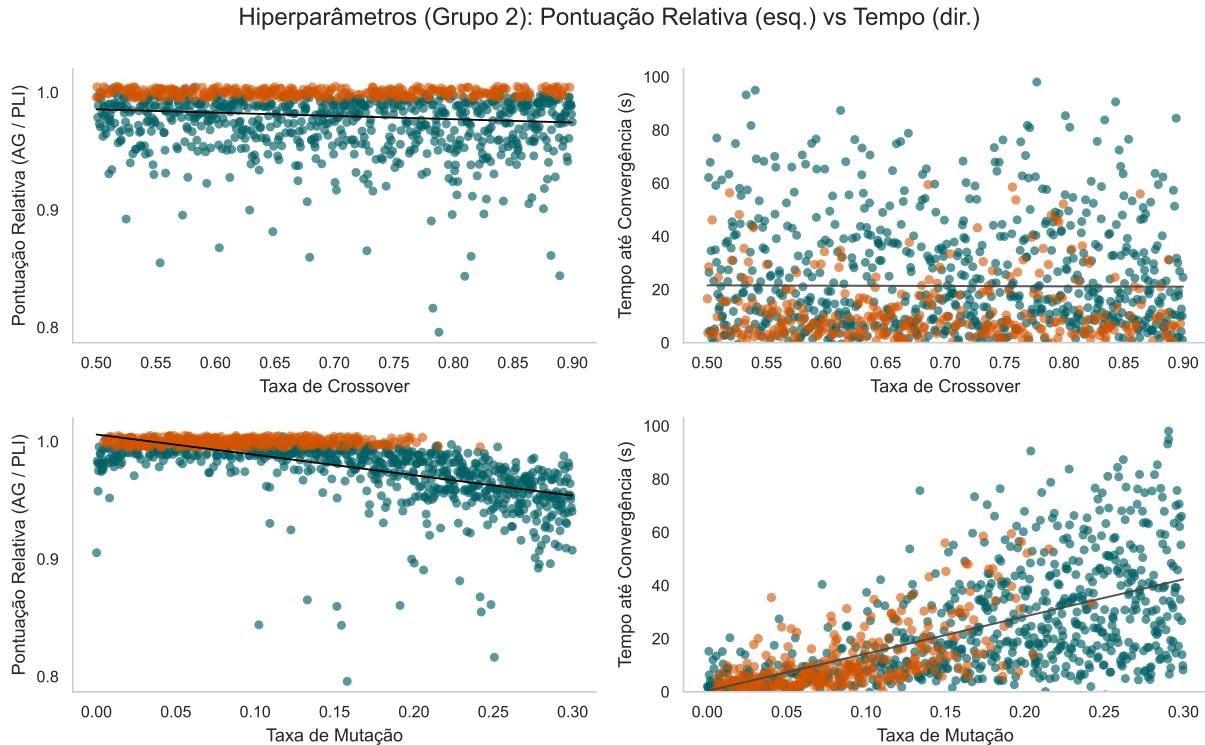
Figura 8: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG



tamanho = 100 elitismo = 15% Analisando os gráficos na Figura 8 podemos ver que a convergência se torna levemente mais estável com o aumento do tamanho da população, mas os resultados durante todo o intervalo apresenta bons resultados. O tempo de convergência apresenta o comportamento esperado, diretamente proporcional ao tamanho da população devido ao aumento no número de operações realizadas. Analisando os resultados com as populações de 90 a 110 indivíduos temos 70 amostras, com uma pontuação média de 98% da pontuação máxima, e 75% delas apresentando uma pontuação relativa acima de 97%, apresentando uma estabilidade alta, com um tempo médio de 13.3 segundos até a convergência, fixando, portanto o tamanho da população em 100 indivíduos.

Na analise do percentual de elitismo podemos observar que os únicos resultados abaixo de 90% da pontuação máxima são obtidos com valores abaixo de 5% de elitismo, e o tempo até a convergência do algoritmo se distribui de maneira uniforme ao longo do intervalo, mas diminuindo a concentração de convergência máxima para valores acima de 30%. Portanto será utilizado um valor de 15% de elitismo.

Figura 9: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG

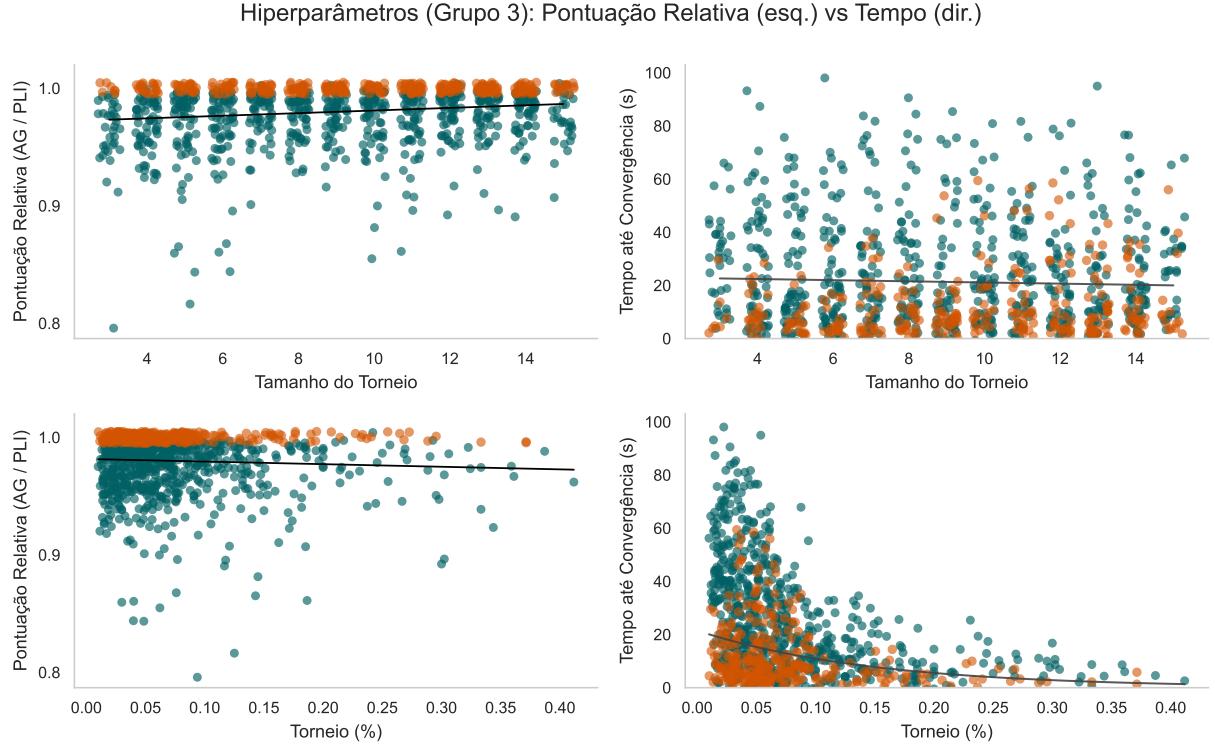


crossover = .7 mutacao = .5

Observando os gráficos da figura 9, podemos ver que a taxa de crossover apresenta uma pequena influência negativa na performance do algoritmo, com um leve aumento na instabilidade da convergência para valores acima de 75%, enquanto o tempo se apresenta de forma uniforme. Fixaremos então o valor de 70% para essa taxa.

A taxa de mutação apresenta a relação mais expressiva com a performance do algoritmo, com 47% das simulações que não convergiram se apresentando em simulações com valores acima de 20% para taxa de mutação, e com apenas 4 simulações com valor máximo nessa região. Além da piora na pontuação, o tempo até a convergência apresenta um grande aumento em média, e na sua variação, tornando o algoritmo mais imprevisível também. Portanto será fixada uma taxa de mutação de 5%.

Figura 10: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG



torneio = 10

Analizando os gráficos da figura 10, podemos observar que não há uma relação forte entre o tamanho do torneio e a pontuação ou o tempo até a convergência, e analisando o percentual do tamanho do torneio em relação a população os resultados parecem indicar tendências de aumento no tempo até a convergência para percentuais menores. No entanto, como o tamanho do torneio foi fixado a partir de um intervalo fixo, e não percentual da população, os resultados de demora no tempo de convergência se devem às simulações com um tamanho da população muito alto, que resultam em um percentual baixo do tamanho do torneio, confundindo as piorias causadas pelas populações grandes com os resultados das simulações que tiveram um percentual do grupo de torneio. Portanto, o valor utilizado para o tamanho do torneio será de 10 indivíduos.

4.3.2 Análise de Convergência

Após as análises, os seguintes valores foram selecionados para os parâmetros do algoritmo genético:

Tabela 6: Valores selecionados para os parâmetros do AG

Parâmetro	Valor
Número de Gerações	350
Número de Indivíduos	100
Taxa de Elitismo	0.15
Taxa de CrossOver	0.7
Taxa de Mutação	0.05
Tamanho do Torneio	10

4.4 Comparação entre os algoritmos

Para comparar a performance entre os 3 algoritmos e selecionar o que obteve a melhor performance, foi realizada uma nova simulação de 100 cenários diferentes, com o objetivo de analisar se as escolhas realizadas dos parâmetros obtém os mesmos resultados da etapa de ajuste.

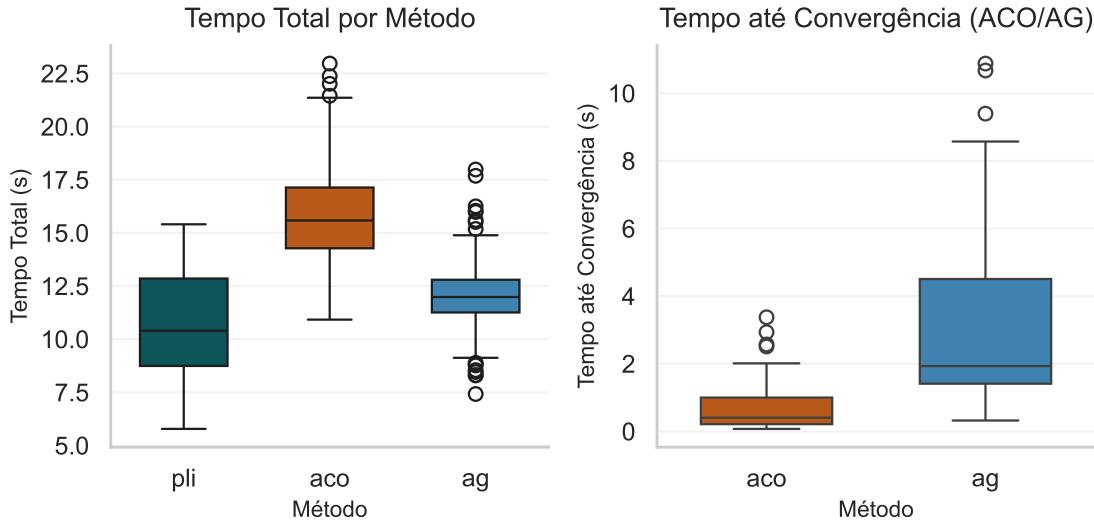
Os resultados obtidos a partir das simulações seguem apresentados na tabela e nos gráficos abaixo:

Tabela 7: Resultado das simulações de comparação

Alg	Nº Max	$\mu_{pontuacao}$	Média	Desvio Padrão	Média	Desvio Padrão
			Tempo	Tempo	Tempo	Tempo
PLI	-	-	10.64	0.67	10.64	0.67
AG	82	0.998	3.22	2.52	12.04	2.52
ACO	100	1	0.69	2.45	15.73	2.87

Para analisar se a boa performance é constante ao longo de diversos cenários, os seguintes gráficos foram construídos:

Figura 11: Boxplots do tempo de execução total e tempo até convergência dos algoritmos



Pela análise da tabela, podemos ver que o ACO teve uma performance bem robusta, com as 100 simulações atingindo o resultado máximo, e com o menor tempo de convergência médio. No entanto, o tempo médio total foi o maior entre os 3, indicando que a convergência acontece extremamente rápido para a maioria das simulações, mas o tempo gasto para percorrer um número de gerações alto o suficiente para ter convergências extremamente estáveis é o mais alto dentre os 3. O Algoritmo Genético atingiu o valor máximo em apenas 82 simulações, mas ainda obteve um resultado extremamente alto e satisfatório, com tempo médio até a convergência próximo do ACO, e tempo médio total um pouco abaixo. E o PLI foi o algoritmo mais estável no tempo, apresentando a menor variação entre os 3.

Figura 12: Gráficos de dispersão entre o tempo até convergência e as perturbações aplicadas

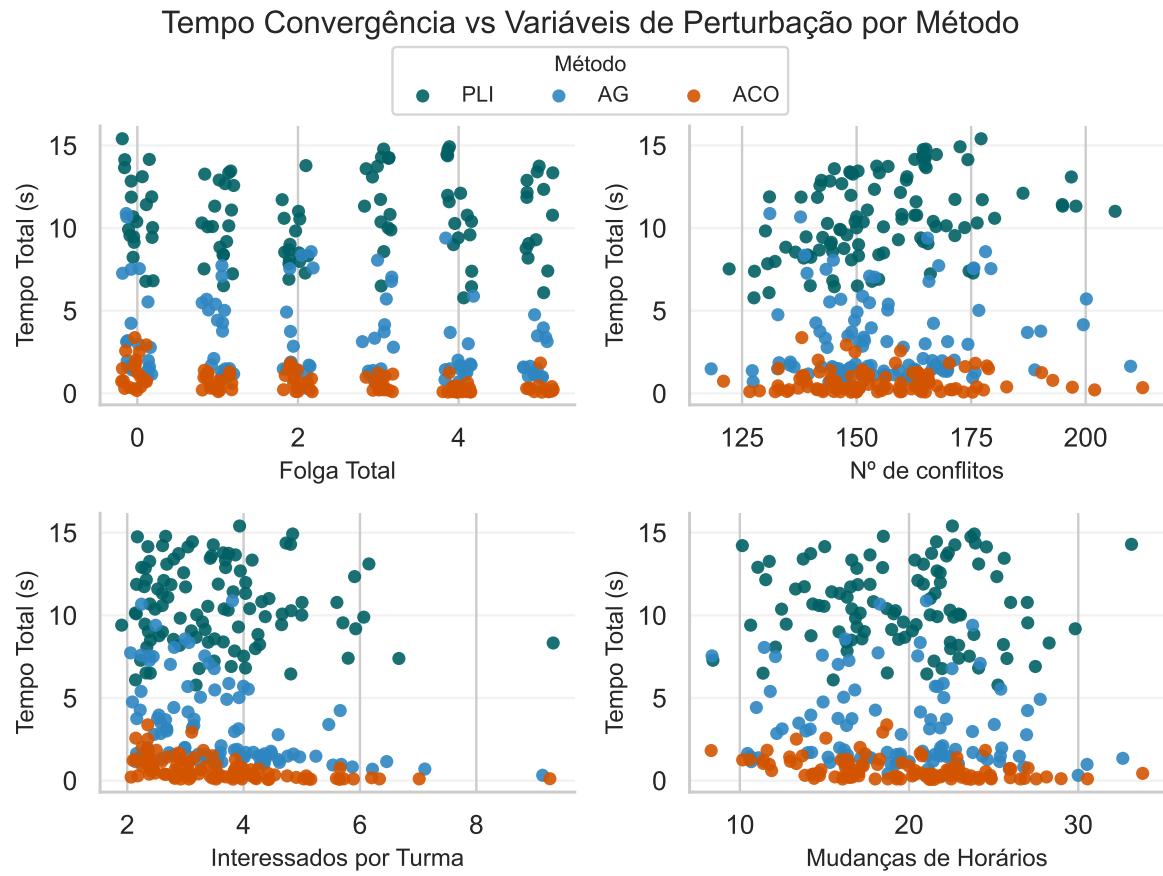
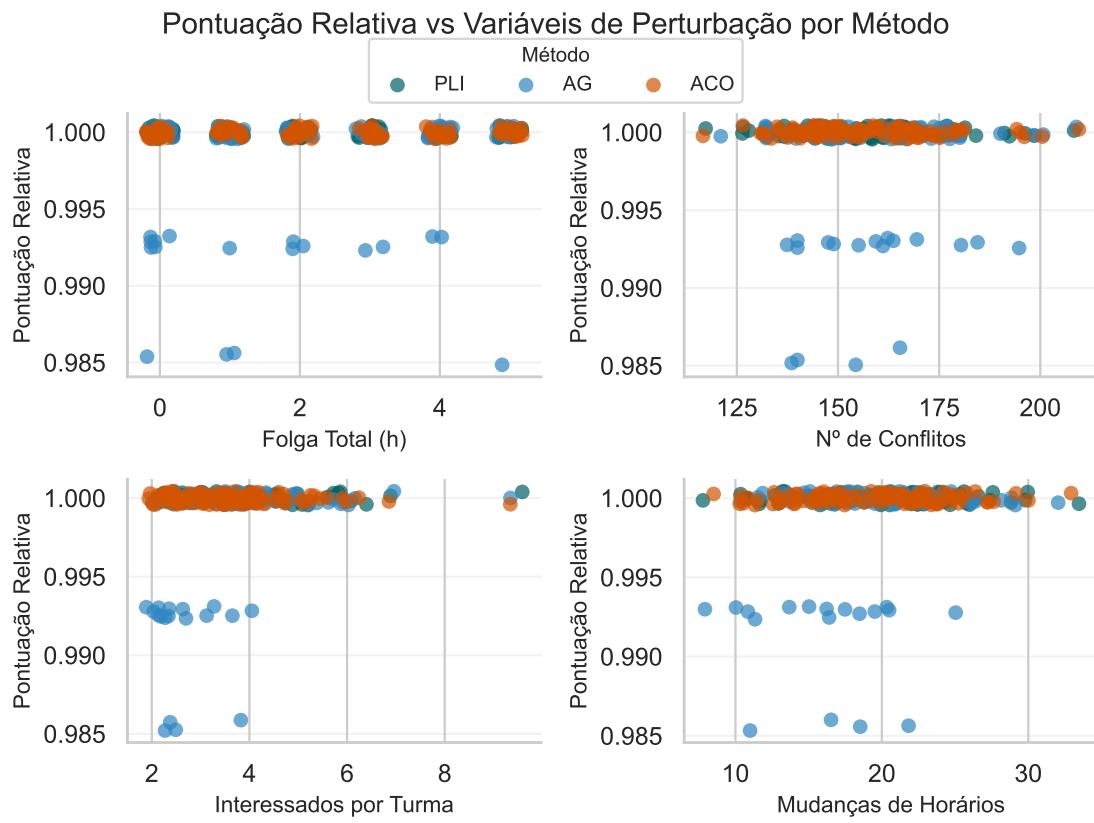


Figura 13: Gráficos de dispersão entre a pontuação relativa e as perturbações aplicadas



5 Conclusão

Aqui você deve escrever a conclusão ou considerações finais.

6 Implementação e Ferramenta Desenvolvida

Com o objetivo de fornecer uma solução usável e robusta para o departamento, foi desenvolvido um ambiente web que integra os algoritmos de otimização implementados. A ferramenta foi projetada com foco na simplicidade de uso e manutenção, permitindo que futuras gerações de coordenadores utilizem o sistema sem necessidade de conhecimento profundo sobre o funcionamento.

A ferramenta foi desenvolvida utilizando Python, com a estrutura visual construída com a biblioteca Flask.

6.1 Inicialização da Plataforma

A gestão dos dados de entrada será feito por planilhas em excel padronizadas, sendo elas:

- Disciplinas: Uma lista com as disciplinas que serão ofertadas no semestre, com um identificador único para cada combinação de disciplina-turma, o nome da disciplina, o código da turma, a carga horária em horas, o tipo de disciplina, e o código do horário;
- Docentes: Uma lista com os docentes disponíveis para o semestre, que contenha: um identificador único para o docente, o nome do docente e o número máximo de disciplinas que ele poderá lecionar no semestre;
- Preferências: Uma lista corrida com cada combinação de disciplina e docente, com a preferência do docente pela disciplina.

Após a configuração dos dados na estrutura correta de diretório, o usuário final deverá executar o arquivo python, e utilizar um navegador para acessar o ambiente local.

6.2 Interface

Ao inicializar a plataforma, a primeira tela apresenta os dados que estão sendo utilizados, para a conferência de que todos os docentes e disciplinas foram carregados da forma correta:

Figura 14: Captura de tela da interface inicial - Aba de Docentes

A interface é uma interface web com uma barra superior preta contendo os links 'Dados Iniciais', 'Execução' e 'Resultado'. Abaixo disso, uma seção intitulada 'Dados Iniciais' mostra o texto 'Professores e Disciplinas carregados para referência.' Com dois botões lado a lado: 'Professores' (em destaque) e 'Disciplinas'. Abaixo, uma seção intitulada 'Professores e Preferências' exibe uma tabela com os seguintes dados:

ID	Nome	Carga Máx.	Nº Sem Preferência	Nº Pref. 1	Nº Pref. 2	Nº Pref. 3
docente_01		200	23	10	11	3
docente_02		200	20	2	15	10
docente_03		200	24	12	0	11
docente_04		200	19	2	14	12

O nome dos docentes foi ocultado para manter a privacidade. Logo após o nome e o ID aparecem as informações do número de disciplinas que ele marcou com cada nível de preferência.

E na aba ao lado, é possível ver as disciplinas que foram carregadas:

Figura 15: Captura de tela da interface inicial - Aba de Disciplinas

A captura de tela mostra a interface inicial da ferramenta, com uma barra superior preta contendo os links "Dados Iniciais", "Execução" e "Resultado". A seção principal é intitulada "Dados Iniciais" e contém o sub-título "Professores e Disciplinas carregados para referência.". Abaixo, há uma barra com dois botões: "Professores" (em azul) e "Disciplinas" (destacado em branco). A seção "Disciplinas Ofertadas" exibe uma tabela com as seguintes informações:

ID Disciplina	Nome	Carga Horária	Código Horário	Código Turma
EST0001_01	COMPUTACAO EM ESTATISTICA 1	30	6M34	1
EST0001_02	COMPUTACAO EM ESTATISTICA 1	30	3T45	2
EST0004_01	ESTATISTICA COMPUTACIONAL	60	3T2345	1
EST0005_01	INFERENCIA BAYESIANA	60	35M34	1
EST0011_01	MODELOS LINEARES GENERALIZADOS	60	24M34	1
EST0017_01	METODOS ESTATISTICOS 2	60	24T23	1
EST0019_01	ESTATISTICA APLICADA	90	246M12	1
EST0019_02	ESTATISTICA APLICADA	90	246M12	2
EST0019_03	ESTATISTICA APLICADA	90	246M12	3
EST0019_04	ESTATISTICA APLICADA	90	246M34	4
EST0019_05	ESTATISTICA APLICADA	90	246M34	5

Após a conferência dos dados, o usuário deverá navegar para a aba de "Execução" que apresenta as seguintes opções:

Figura 16: Captura de tela de execução

Dados Iniciais Execução Resultado

Executar Otimização

Selecionar Algoritmos:

PLI ACO AG

Seed (opcional)

42

Parâmetros PLI

Peso Penalidade (W)

4

Parâmetros ACO

Nº Formigas	Nº Gerações	Alfa	Beta	Taxa Evaporação
10	30	1.5	8	0.15

Alocações Fixas (opcional):

Professor	Disciplina	Remover
docente_20 - LEANDRO TAVARES CORREIA	EST0081_01 - TRABALHO DE CONCLUSAO DE CURSC	Remover

[Adicionar Alocação](#)

Selecione pares Professor-Disciplina. Serão aplicados como fixos na solução.

Executar

Nessa aba, o usuário poderá selecionar o algoritmo que quer utilizar, configurar uma semente, e poderá modificar os parâmetros de inicialização dos algoritmos, que como padrão serão os algoritmos selecionados ao longo das análises.

Além disso, o usuário terá a opção de fixar alocações selecionando o docente e a disciplina-turma que irá lecionar. É possível pré-fixar qualquer número de alocações, e os 3 algoritmos inicializam sem as disciplinas já alocadas, e com os professores com a carga horária já comprometida.

Após a seleção, é só selecionar o botão de "Executar", e após o período de carregamento, o resultado final aparecerá da seguinte forma:

Figura 17: Captura de tela do resultado

Dados Iniciais		Execução	Resultado
Resultado da Otimização			
Escore: 132 ● 43 ● 1 ● 1 ● 2 ● 1 Total: 47			
Professor	Disciplina	Pref.	Cód. Horário
docente_01	TECNICAS DE AMOSTRAGEM EST0036_01	3	356M12
LEANDRO TAVARES CORREIA docente_20	TRABALHO DE CONCLUSAO DE CURSO 1 EST0081_01	3	6M1234
docente_04	PROBABILIDADE E ESTATISTICA EST0023_03	3	46M12
docente_05	PROBABILIDADE E ESTATISTICA EST0023_04	3	24M34
docente_06	ESTATISTICA APLICADA EST0019_07	3	246N12
docente_07	INFERENCIA BAYESIANA EST0005_01	3	35M34
docente_08	METODOS ESTATISTICOS 1 EST0070_01	3	24M34

Na tela do resultado, o usuário poderá ver na parte superior: a pontuação daquela alocação específica, o número de preferências 3,2,1 e 0 atendidas, representadas pelas cores verde, amarelo, laranja e vermelho respectivamente. E ao lado poderá ver o número de alocações que foram pré selecionadas, e o total de disciplinas que foram consideradas.

No canto superior direito, o usuário pode selecionar um botão para baixar o resultado em um arquivo Excel. E poderá analisar na parte de baixo a lista completa das alocações realizadas.

E para facilitar a navegação e conferência dos resultados, ao passar o cursor em cima do nome de algum docente será possível ver as disciplinas que ele foi alocado, e logo abaixo a lista das disciplinas que ele manifestou alto interesse (Figura 18), e ao mover o cursor para o nome de alguma disciplina será possível ver o nome dos professores que tinham interesse de lecionar ela (Figura 19).

Figura 18: Detalhes das preferências do docente

JOSE AUGUSTO FIORUCCI docente_18	PROBABILIDAD EST0023_01
Disciplinas alocadas	
JOSE AUGUS docente_18	- PROBABILIDADE E ESTATISTICA - COMPUTACAO EM ESTATISTICA 2: PYTHON
Preferência 3 (alto interesse)	
docente_19	- PROBABILIDADE E ESTATISTICA - COMPUTACAO EM ESTATISTICA 2: PYTHON

Figura 19: Detalhes dos interessados na disciplina

TRABALHO DE CONCLUSAO DE CURSO 1 EST0081_01	Preferência 3 (alto interesse) - LEANDRO TAVARES CORREIA EST0069_01
INTRODUCAO A	

Os códigos necessários para funcionamento do arquivo, tratamento dos dados, e explicações mais detalhadas de uso e manutenção ficarão disponíveis em um repositório público na plataforma github.

Referências

- BURKE, E. K.; PETROVIC, S. Recent research directions in automated timetabling. *European journal of operational research*, Elsevier, v. 140, n. 2, p. 266–280, 2002.
- DORIGO, M.; GAMBARDELLA, L. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 53–66, 1997.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: MIT press, 1992.

Apêndice

A Códigos de Perturbação

A.1 Código de simulação de horários

```
def perturbar_horarios(df_disciplinas, seed=None):
    """
    Perturba os horários das disciplinas selecionando aleatoriamente novas
    → opções
    com base na distribuição de frequência existente para cada par (Carga
    → Horária, Tipo).
    """
```

A função:

1. Calcula a distribuição de probabilidade dos horários atuais.
2. Sorteia uma proporção variável de disciplinas para alterar.
3. Para cada disciplina selecionada, escolhe um novo horário compatível, adicionando um ruído estocástico às probabilidades originais para
→ garantir variabilidade.
4. Recalcula a matriz de conflitos e retorna o saldo (delta) em relação
→ ao estado original.

Parameters:

df_disciplinas : pd.DataFrame

DataFrame contendo as disciplinas com colunas 'horario',
→ 'carga_horaria' e 'tipo_disciplina'.

seed : int, optional

Seed para reproduzibilidade do gerador de números aleatórios.

Returns:

tuple

(metricas: dict, matriz_conflitos_perturbada: pd.DataFrame)
Onde 'metricas' contém o número de mudanças e o saldo de conflitos
→ gerados/resolvidos.

"""

```
if seed is not None:
    rng.seed(seed)

df_result = df_disciplinas.copy()
#Calcula as probabilidades de cada horario baseado na carga horaria e
↪ tipo de disciplina
probs = df_result[['carga_horaria', 'tipo_disciplina',
↪ 'horario']].value_counts(normalize=True)
#Seleciona aleatoriamente uma proporção de disciplinas para alterar o
↪ horario
proporcao_alteracoes = min(max(rng.normal(0.2, .2, 1 ), 0), 1)
indices_escolhidos =
↪ df_result.sample(frac=float(proporcao_alteracoes)).index

for indice in indices_escolhidos:
    try:
        i = df_result.loc[indice]
        #Seleciona as opcoes possiveis para o tipo de disciplina e carga
        ↪ horaria disponivel
        opcoes = probs[i['carga_horaria']][i['tipo_disciplina']]
        #Calcula as probabilidades de cada horario e adiciona um ruido
        ↪ normal
        probabilidades = normalizar_probabilidades(opcoes.values +
        ↪ rng.normal(loc=0, scale=.003, size=len(opcoes.values)))
        #Escolhe um novo horario baseado nas probabilidades
        df_result.at[indice, 'horario'] = rng.choice(opcoes.index,
        ↪ size=1, p = probabilidades)
    except:
        continue
#Nº de conflitos na matriz original é 146
n_mudancas_horarios = len(indices_escolhidos)
matriz_conflitos = criar_matriz_conflitos(df_result)

#Saldo de conflitos
n_conflitos = matriz_conflitos.sum().sum()/2
metricas = {
    'n_mudancas_horarios': n_mudancas_horarios,
    'n_conflitos': n_conflitos
}
return metricas, matriz_conflitos
```

A.2 Código de simulação de preferências

```
def perturbar_preferencias(df_preferencias, df_disciplinas, seed=None):
    """
    Altera as preferencias forma aleatória.

    Parameters:
    -----
    df_preferencias : pd.DataFrame
        DataFrame contendo pelo menos a coluna 'preferencia'
    proporcao_perturbacao : float
        Proporção de preferências a serem alteradas (0 a 1)
    seed : int, optional
        Seed para reproduzibilidade

    Returns:
    -----
    pd.DataFrame
        DataFrame com as preferências perturbadas
    """

    if seed is not None:
        rng.seed(seed)

    #Calcula o número de turmas por disciplina
    turmas_disc = df_disciplinas[['id_disciplina', 'disciplina']]\
        .merge(df_disciplinas.groupby('disciplina').agg({'id_disciplina':'c'
        ↳ ount'}).rename(columns={'id_disciplina':'n_turmas'}).reset_index()
        ↳ x(), on='disciplina', how='left')\
        .copy()

    #Cria o dataframe que será alterado
    df_result = df_preferencias.copy()
    proporcao_perturbacao = min(max(rng.normal(.15, .1), 0), 1)
    opcoes = [0, 1, 2, 3]
    prob_original = [0.43, 0.21, 0.07, 0.29]
    probabilidades = normalizar_probabilidades(prob_original +
        ↳ rng.normal(loc=0, scale=.09, size=4))
    n_alteracoes = int(len(df_result) * proporcao_perturbacao)
```

```
try:
    escolhas = rng.choice(opcoes,
                          size=n_alteracoes,
                          replace=True,
                          p=probabilidades)
except:
    escolhas = rng.choice(opcoes,
                          size=n_alteracoes,
                          replace=True,
                          p=prob_original)

indices_aleatorios = df_result.sample(n = n_alteracoes).index
df_result.loc[indices_aleatorios, 'preferencia'] = escolhas

#Calcula métricas de variacao de preferências
#Calcula numero de professores interessados por turma
analise = df_result[df_result['preferencia']
                     ==3].groupby('id_disciplina').agg({'id_docente':'count'}).reset_index()
analise.rename(columns={'id_docente':'n_professores_interessados'})
#Adiciona numero de turmas por disciplina
analise = analise.merge(turmas_disc, on='id_disciplina', how='outer')
#Calcula a media de professores interessados por disciplina
analise = analise.groupby(['disciplina']).agg({'id_disciplina':'first',
                                               'n_professores_interessados':'mean',
                                               'n_turmas':'first'}).reset_index()
#Calcula interessados por turma
analise['interessados_por_turma'] = analise['n_professores_interessados']/
                                     analise['n_turmas']
#Calcula global de interessados por turma
n_interessados_por_turma = analise['interessados_por_turma'].mean()
metricas = {
    'proporcao_perturbacao': proporcao_perturbacao,
    'n_interessados_por_turma': n_interessados_por_turma
}

return metricas, df_result
```

A.3 Código de simulação de máximo de disciplinas

```
def perturbar_max_disciplinas(df_professores, seed=None):
    """
    Aumenta aleatoriamente o número máximo de disciplinas para professores
    que têm max_disciplinas = 1.

    Parameters:
    -----
    df_professores : pd.DataFrame
        DataFrame contendo informações dos professores com coluna
        'max_disciplinas'
    seed : int, optional
        Seed para reproduzibilidade

    Returns:
    -----
    tuple
        (folga_ganha, df_result) onde:
        - folga_ganha: número de professores que ganharam folga
        - df_result: DataFrame com max_disciplinas atualizado
    """
    if seed is not None:
        rng.seed(seed)

    df_result = df_professores.copy()
    folga_ganha = rng.randint(0, 6)
    indices_aleatorios = df_result[df_result['max_disciplinas'] ==
                                    1].sample(folga_ganha).index
    df_result.loc[indices_aleatorios, 'max_disciplinas'] += 1
    metricas = {
        'folga_ganha': folga_ganha
    }
    return metricas, df_result
```

B Tabela Hiperparâmetros Ajuste - AGs

Tabela 8: Resumo organizado das execuções do AG (Apêndice).

Simulacao	Populacao	Torneio	Elite	Taxa_Crossover	Taxa_Mutacao	Fitness_Final	Tempo_Seg
1	412	5	0	0.499	0.085	136	6.486
2	402	3	0	0.839	0.238	132	14.868
3	311	5	4	0.778	0.292	136	13.405
4	199	5	5	0.883	0.210	136	6.345
5	118	2	0	0.962	0.088	130	1.842
6	219	6	4	0.415	0.111	136	4.134
7	462	9	4	0.598	0.099	136	8.149
8	258	4	1	0.903	0.018	136	1.504
9	453	3	4	0.803	0.283	136	19.541
10	476	7	5	0.992	0.150	136	11.658
11	396	2	1	0.742	0.091	134	6.505
12	270	6	0	0.913	0.286	135	13.024
13	300	9	2	0.865	0.008	136	1.354
14	433	7	3	0.409	0.242	136	25.612
15	87	3	2	0.830	0.154	136	5.419
16	23	5	5	0.554	0.143	136	1.121
17	26	4	4	0.920	0.277	136	2.463
18	499	7	2	0.480	0.003	136	4.773
19	195	5	1	0.816	0.045	136	4.867
20	423	9	5	0.530	0.078	136	15.735
21	206	9	0	0.606	0.060	136	6.217
22	246	8	5	0.475	0.129	136	13.462
23	95	5	5	0.489	0.257	136	9.251
24	108	6	0	0.617	0.056	136	3.088
25	399	2	3	0.810	0.141	135	23.911
26	340	5	2	0.734	0.225	136	30.086
27	131	5	4	0.634	0.274	136	13.477
28	323	9	0	0.964	0.122	136	17.191
29	389	7	1	0.847	0.034	136	8.182
30	238	4	3	0.674	0.161	136	15.705
31	481	6	1	0.929	0.228	136	43.943
32	241	3	0	0.614	0.062	136	7.642
33	136	6	1	0.719	0.290	135	15.626
34	336	7	2	0.449	0.244	136	32.235
35	263	8	2	0.589	0.027	136	4.910
36	333	5	1	0.625	0.031	136	6.685
37	123	7	1	0.533	0.298	135	14.352
38	486	6	2	0.746	0.270	136	51.379
39	377	4	2	0.567	0.135	136	21.628
40	386	5	1	0.630	0.136	136	22.205
41	113	2	4	0.568	0.215	135	9.513
42	82	8	1	0.951	0.235	136	7.686
43	325	7	4	0.649	0.132	136	18.245
44	427	9	3	0.447	0.117	136	21.669
45	371	4	0	0.403	0.083	136	14.240
46	142	4	1	0.754	0.253	135	13.995
47	53	4	4	0.979	0.004	136	0.523
48	437	9	2	0.517	0.049	136	11.572
49	290	6	1	0.713	0.021	136	4.728

C nome apêndice 2

Anexo

A nome anexo 1

texto...