



**Universidade de Brasília
Departamento de Estatística**

**Meta-heurísticas e Métodos Exatos no Problema de Alocação de Docentes:
Algoritmos Genéticos, Colônia de Formigas e PLI no Departamento de
Estatística da UnB**

Karl Peixoto

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**

Karl Peixoto

**Meta-heurísticas e Métodos Exatos no Problema de Alocação de Docentes:
Algoritmos Genéticos, Colônia de Formigas e PLI no Departamento de
Estatística da UnB**

Orientador(a): Eduardo Monteiro de Castro Gomes

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**

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 a escala dos dados do problema e a inclusão de adaptações heurísticas nos algoritmos permitiram que o problema fosse resolvido de forma satisfatória e eficiente por todos os algoritmos.

Como contribuição prática, desenvolveu-se um Sistema de Apoio à Decisão disponibilizado para uso da comunidade acadêmica (disponível em: https://github.com/karl-peixoto/projeto_alocador (PEIXOTO, 2025)).

Palavras-chave: Otimização Combinatória; *Timetabling*; Pesquisa Operacional; Meta-heurísticas; Alocação de Docentes;

Agradecimentos

Agradeço à Universidade de Brasília (UnB) e ao Departamento de Estatística pela excelência no ensino, pela estrutura disponibilizada e pelas oportunidades de crescimento oferecidas ao longo da graduação.

Ao meu orientador, Prof. Eduardo, pelos direcionamentos precisos, pelo suporte fundamental na consolidação do código e do relatório, e pelas valiosas sugestões que aprimoraram a ferramenta desenvolvida neste trabalho.

Aos meus amigos do trabalho, Mariane, José Eduardo e Helton, que me acompanharam durante quase todo o curso. Agradeço imensamente pela compreensão nas épocas de alta demanda e pela flexibilidade que me permitiu conciliar as atividades acadêmicas com a rotina profissional. Um agradecimento especial ao Victor, pelos ensinamentos de programação, que foram essenciais para o desenvolvimento deste trabalho.

Aos meus pais, Renata e Yuri, meu eterno reconhecimento. Obrigado pelo apoio incondicional, pelo cuidado e preocupação constante com a minha saúde e bem estar. Agradeço por me proporcionarem todas as condições para ter chegado até aqui.

Aos meus irmãos, Luca e Diego, pelo convívio, pela parceria em casa e, principalmente pela paciência comigo nos momentos de maior intensidade e dedicação ao trabalho.

À minha madrinha, Roberta, pelo incentivo decisivo na escolha do curso e pelos vários cafés que orientaram os meus estudos. E à minha avó, Mariene, que me presenteou com o computador que viabilizou a realização de todas as atividades acadêmicas no início do curso. Estendo minha gratidão a toda a minha família, pelo apoio constante e pela compreensão nos momentos de ausência.

À minha namorada, Beatriz, pelo carinho, compreensão, paciência e pelo esforço genuíno em entender os problemas que eu compartilhava. Obrigado por ter me ouvido falar incansavelmente sobre este trabalho, sobre os diversos planos, e sobre as dificuldades que encontrei no caminho.

Por fim, aos meus amigos de fora do meio acadêmico, Tang, André e Giulia, que me acompanharam em toda a trajetória. Obrigado por me ouvirem elogiar e reclamar do curso, por lidarem com minhas ausências e por compreenderem as inúmeras vezes que eu disse que estava sem tempo nos finais de semestre. O apoio de vocês nos momentos bons e ruins foi indispensável.

Lista de Tabelas

1	Limites de busca dos parâmetros do ACO	25
2	Limites de busca dos parâmetros do AG	26
3	Distribuição das preferências por disciplina	40
4	Distribuição das preferências por docente	42
5	Valores selecionados para os parâmetros do ACO	46
6	Valores selecionados para os parâmetros do AG	50
7	Resultado das simulações de comparação	50

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	Captura de tela da interface inicial - Aba de Docentes	34
5	Captura de tela da interface inicial - Aba de Disciplinas	35
6	Captura de tela de execução	36
7	Captura de tela do resultado	37
8	Detalhes das preferências do docente	37
9	Detalhes dos interessados na disciplina	37
10	Histograma de pontuação relativa e tempo até convergência ACO	43
11	Gráficos de dispersão entre tamanho da população e taxa de evaporação pela pontuação e tempo de convergência do ACO	44
12	Gráficos de dispersão entre Alfa e Beta pela pontuação e tempo de con- vergência do ACO	45
13	Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG	46
14	Gráficos de dispersão entre o tamanho da população e percentual de eli- tismo pela pontuação e tempo de convergência do AG	47
15	Gráficos de dispersão entre as taxas de <i>crossover</i> e mutação pela pontuação e tempo de convergência do AG	48
16	Gráficos de dispersão entre tamanho absoluto e relativo do torneio pela pontuação e tempo de convergência do AG	49
17	Boxplots do tempo de execução total e tempo até convergência dos algoritmos	51
18	Gráficos de dispersão entre o tempo até convergência e as perturbações aplicadas	52
19	Gráficos de dispersão entre a pontuação relativa e as perturbações aplicadas	53

Sumário

1 Introdução	8
2 Referencial Teórico	9
2.1 Programação Linear Inteira	9
2.1.1 Características da PLI	9
2.1.2 Vantagens e Desvantagens	10
2.2 Algoritmos Genéticos	10
2.2.1 Representação e População	10
2.2.2 Função de Avaliação	10
2.2.3 Operadores Genéticos	10
2.2.4 Estrutura Geral do Algoritmo	11
2.2.5 Critérios de Parada	11
2.2.6 Vantagens e Desvantagens	12
2.3 Algoritmo de Colônia de Formiga	12
2.3.1 Princípios Fundamentais	12
2.3.2 Componentes do ACO	12
2.3.3 Procedimento Geral do ACO	13
3 Metodologia	14
3.1 Aplicação da Programação Linear Inteira	14
3.1.1 Componentes do Modelo	14
3.1.2 Função Objetivo	15
3.1.3 Restrições do Modelo	16
3.1.4 Otimização	17
3.2 Aplicação do Algoritmo Genético	17
3.2.1 Representação do Indivíduo (Cromossomo)	17
3.2.2 Função de Avaliação (<i>Fitness Function</i>)	18
3.2.3 Operadores Genéticos Aplicados	19

3.3 Aplicação do Algoritmo de Colônia de Formigas	21
3.3.1 Estrutura do Problema e Representação	21
3.3.2 Componentes do Modelo ACO	21
3.4 Delineamento Experimental	24
3.4.1 Exploração do Espaço de Parâmetros	24
3.4.2 Cenários de Teste e Perturbação dos Dados	28
3.4.3 Métricas de Avaliação e Ambiente Computacional	31
4 Ferramenta Desenvolvida	33
4.1 Inicialização da Plataforma	33
4.2 Interface	33
5 Resultados	38
5.1 Caracterização do Cenário de Estudo	38
5.1.1 Análise da Procura por Disciplinas	39
5.2 Análise e Calibração do Algoritmo de Colônia de Formigas	43
5.2.1 Análise de sensibilidade dos hiperparâmetros	43
5.2.2 Configuração dos Parâmetros	45
5.3 Análise e Calibração do Algoritmo Genético	46
5.3.1 Análise de sensibilidade dos hiperparâmetros	46
5.3.2 Configuração dos Parâmetros	49
5.4 Comparação entre os algoritmos	50
6 Conclusão	55
Referências	56

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. Para resolver esse problema, este projeto apresenta no capítulo 2 uma revisão teórica das abordagens utilizadas, no capítulo 3 uma descrição de como as abordagens foram adaptadas para o problema, no capítulo 4 uma proposta de ferramenta de apoio, automatizando o processo de alocação dos docentes considerando restrições operacionais, preferências individuais e decisões prévias do coordenador. A solução implementa e facilita o uso dos três algoritmos propostos, permitindo a escolha dos parâmetros dos modelos ou a utilização baseada nos resultados encontrados no capítulo 5.

2 Referencial Teórico

2.1 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:

$$\text{Minimizar (ou Maximizar)} \quad c^\top x \quad (2.1.1)$$

$$\text{sujeito a} \quad Ax \leq b \quad (2.1.2)$$

$$x \in \mathbb{Z}^n \quad (2.1.3)$$

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.

2.1.1 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 ex-

clusões mútuas, limites mínimos/máximos, prioridades e preferências, por meio de combinações lineares das variáveis.

2.1.2 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.

2.2 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.

2.2.1 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).

2.2.2 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.

2.2.3 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.

2.2.4 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).

2.2.5 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.

O critério de parada utilizado para essa implementação foi o número fixo de gerações.

2.2.6 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.

2.3 Algoritmo de Colônia de Formiga

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.

2.3.1 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.

2.3.2 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) .

2.3.3 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 utilizando uma função adaptada para o problema.
 - (c) Atualizar os valores de feromônio com base nas soluções, de forma a favorecer as soluções que resolvam o problema de forma satisfatória.
3. Repetir até que um critério de parada seja satisfeito (número máximo de iterações, tempo limite, convergência, etc.).

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 Aplicação da Programação Linear Inteira

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.

3.1.1 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 número de disciplinas, 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 .

3.1.2 Função Objetivo

A função objetivo, apresentada na equação 3.1.1, 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:

$$\text{Maximizar } Z = \left(\sum_{p \in P} \sum_{d \in D} pref_{pd} \cdot x_{pd} \right) - W \cdot \left(\sum_{p \in P} y_p \right) \quad (3.1.1)$$

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.

3.1.3 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*).

1. **Garantia de Oferta de Disciplinas:** Cada disciplina do conjunto de oferta D deve ser atribuída a exatamente um professor. Esta restrição, definida na equação 3.1.2, assegura que toda a oferta de disciplinas do semestre seja coberta, e que nenhuma disciplina seja alocada para dois professores diferentes

$$\sum_{p \in P} x_{pd} = 1, \quad \forall d \in D \quad (3.1.2)$$

2. **Limite de Carga Horária Docente:** O número de disciplinas alocadas a um professor não pode exceder a sua capacidade máxima de trabalho, conforme a equação 3.1.3

$$\sum_{d \in D} x_{pd} \leq CH_{max,p}, \quad \forall p \in P \quad (3.1.3)$$

3. **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.

$$x_{pd_i} + x_{pd_j} \leq 1, \quad \forall p \in P, \forall (d_i, d_j) \text{ tal que } Cf_{d_i d_j} = 1 \quad (3.1.4)$$

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 3.1.5 conecta a variável de alocação x_{pd} à variável de penalidade y_p

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

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.

3.1.4 Otimização

Para a resolução do modelo matemático proposto, não foi necessária a implementação manual de algoritmos de otimização exata. A solução computacional foi obtida utilizando a biblioteca *PuLP* (Mitchell, Stuart & Kean, Anita (2009)), desenvolvida em Python. Essa ferramenta atua como uma interface de modelagem que permite descrever a função objetivo e as restrições em código Python e utiliza *solvers* otimizados para a resolução, garantindo eficiência no processo de busca pela solução ótima global sem a necessidade da implementação completa do algoritmo.

3.2 Aplicação do Algoritmo Genético

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.

3.2.1 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.

3.2.2 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.

Na implementação adotada, optou-se por uma abordagem unificada de penalização, onde um único coeficiente pondera todas as violações de restrições duras (carga horária e conflitos), simplificando a calibração dos parâmetros. A função de *fitness* para um cromossomo C é definida pela equação 3.2.1:

$$f(C) = \left(\sum_{d=0}^{|D|-1} pref_{C[d],d} \right) - W \cdot \left(\sum_{p \in P} Excesso_p(C) + Conf_p(C) \right) \quad (3.2.1)$$

Os componentes da função são:

- **Escore de Preferência:** O primeiro termo, $\sum pref_{C[d],d}$, representa a soma direta das preferências de cada alocação presente no cromossomo C . Este termo busca maximizar a satisfação docente.
- **Penalidade Unificada (W):** O coeficiente escalar W (fator de penalidade) é aplicado à soma de todas as violações encontradas na solução. Este valor deve ser suficientemente alto para garantir que soluções inviáveis tenham *fitness* muito inferior às soluções viáveis.
- **Violação de Carga Horária ($\sum Excesso_p(C)$):** Representa o somatório de horas que excedem o limite individual de cada professor p . O excesso é calculado como $\max(0, CargaAtual_p - CH_{max,p})$.

- **Violação de Conflitos ($Conf_p(C)$):** Representa o número total de pares de disciplinas alocadas ao mesmo professor que possuem conflito de horário (i.e., ocorrem simultaneamente). A verificação é realizada através da matriz de conflitos pré-calculada.

3.2.3 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 score, 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). A implementação de mutação foi adaptada para realizar mutações inteligentes e se adaptar às regras do problema.

Utilizamos uma mutação adaptativa, quando o professor selecionado para lecionar a disciplina já está ocupado com a sua capacidade máxima, há uma troca entre os professores para não tornar a solução inviável, caso o professor ainda tenha capacidade a troca é feita apenas no gene sorteado.

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 , com probabilidade de seleção proporcional à preferência do professor pela disciplina do gene.

3.3 Aplicação do Algoritmo de Colônia de Formigas

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.

3.3.1 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 permutando aleatoriamente a ordem de visita das disciplinas para gerar uma solução candidata a cada iteração.

3.3.2 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 ($CH_{max,p}$) e se as disciplinas que o professor já foi alocado não possuem conflito de horário com as novas disciplinas, considerando

as alocações que a mesma formiga já realizou nos passos anteriores. Isso garante que toda solução gerada seja, por construção, viável.

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 medida intuitiva para a implementação da heurística. Assim, define-se:

$$\eta_{dp} = pref_{pd} + \epsilon \quad (3.3.1)$$

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 quão 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 inicializadas 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 3.3.2:

$$p(p|d_k) = \begin{cases} \frac{[\tau_{d_k p}]^\alpha \cdot [\eta_{d_k p}]^\beta}{\sum_{j \in \mathcal{N}_k^m} [\tau_{d_k j}]^\alpha \cdot [\eta_{d_k j}]^\beta} & \text{se } p \in \mathcal{N}_k^m \\ 0 & \text{caso contrário} \end{cases} \quad (3.3.2)$$

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.

$$\tau_{dp} \leftarrow (1 - \rho) \cdot \tau_{dp}, \quad \forall (d, p) \in D \times P \quad (3.3.3)$$

2. **Depósito:** Diferente da abordagem clássica onde todas as formigas depositam feromônio, a implementação adotada utiliza uma estratégia elitista por iteração. Apenas a formiga que construiu a melhor solução da geração atual (S_{ib}) realiza o depósito de feromônio nas alocações que compõem sua solução. A quantidade de feromônio depositada, $\Delta\tau$, é calculada dinamicamente em função da diferença de qualidade entre a solução atual e a melhor solução global encontrada até o momento (S_{gb}), conforme a equação 3.3.4:

$$\Delta\tau_{dp} = \begin{cases} \frac{1}{1+(f(S_{gb})-f(S_{ib}))} & \text{se } (d, p) \in S_{ib} \\ 0 & \text{caso contrário} \end{cases} \quad (3.3.4)$$

Onde $f(S)$ representa o valor da função objetivo (*fitness*). Essa abordagem intensifica a exploração nas vizinhanças das soluções promissoras, acelerando a convergência. O denominador atua como um fator de normalização, garantindo que o depósito máximo (valor 1.0) ocorra apenas quando a melhor formiga da geração iguala ou supera o recorde histórico de qualidade da otimização.

3.4 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.

3.4.1 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**).

Segundo McKay, Beckman e Conover (2000), que introduziram o método, o LHS oferece uma estimativa da média e da variância da saída do modelo com significativamente menos variabilidade do que a amostragem aleatória simples, sendo particularmente superior em simulações computacionais onde o custo de avaliação é alto. Além disso, Santner et al. (2003) destacam que o LHS é uma das classes de delineamentos mais eficazes para experimentos computacionais (*Space-Filling Designs*), pois projeta amostras que preencham o espaço de forma uniforme, maximizando a informação obtida com um número limitado de simulações.

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

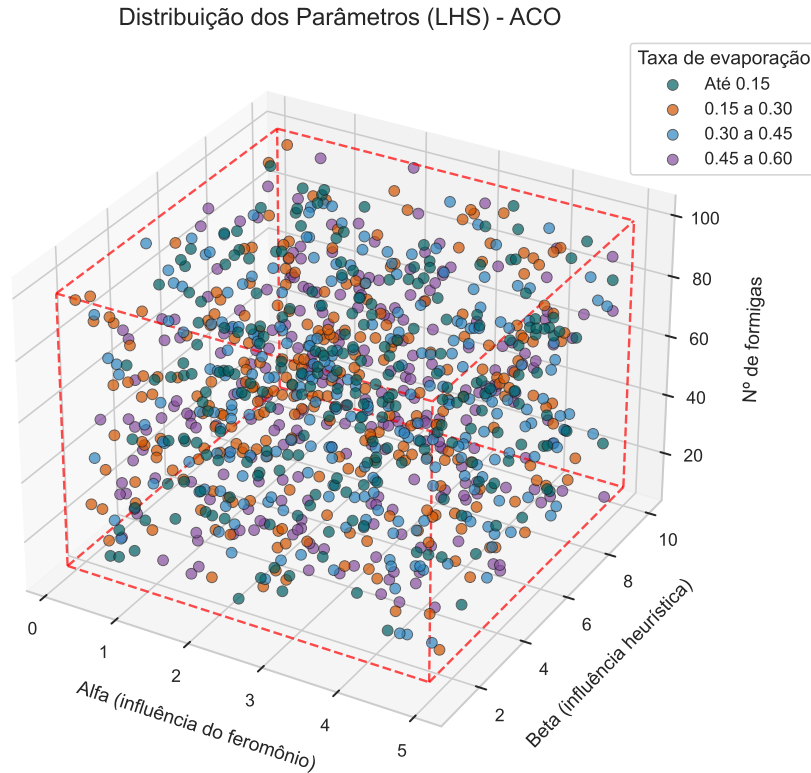
O número de formigas utilizado na busca define o número de possibilidades que são exploradas ao mesmo tempo. Aumentar o número de formigas ajuda a lidar com problemas com muitos mínimos locais, utilizando a diversidade para evitá-los. No entanto, o aumento do número de formigas faz com que o número de operações realizadas a cada iteração aumente de forma exponencial, portanto devemos minimizar o custo computacional sem prejudicar a qualidade da solução.

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. Portanto, selecionar um valor elevado para beta é essencial para que as formigas construam soluções factíveis e direcionadas, mas o parâmetro não pode ser tão elevado para não impedir o aprendizado coletivo da melhor solução dentre as possíveis.

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. Portanto, é necessário balancear para ter conhecimento suficiente para direcionar para as melhores soluções, mas esquecer rápido o suficiente para escapar dos mínimos locais das soluções iniciais.

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
Tamanho da População	30	300
Taxa de <i>Crossover</i>	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-

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

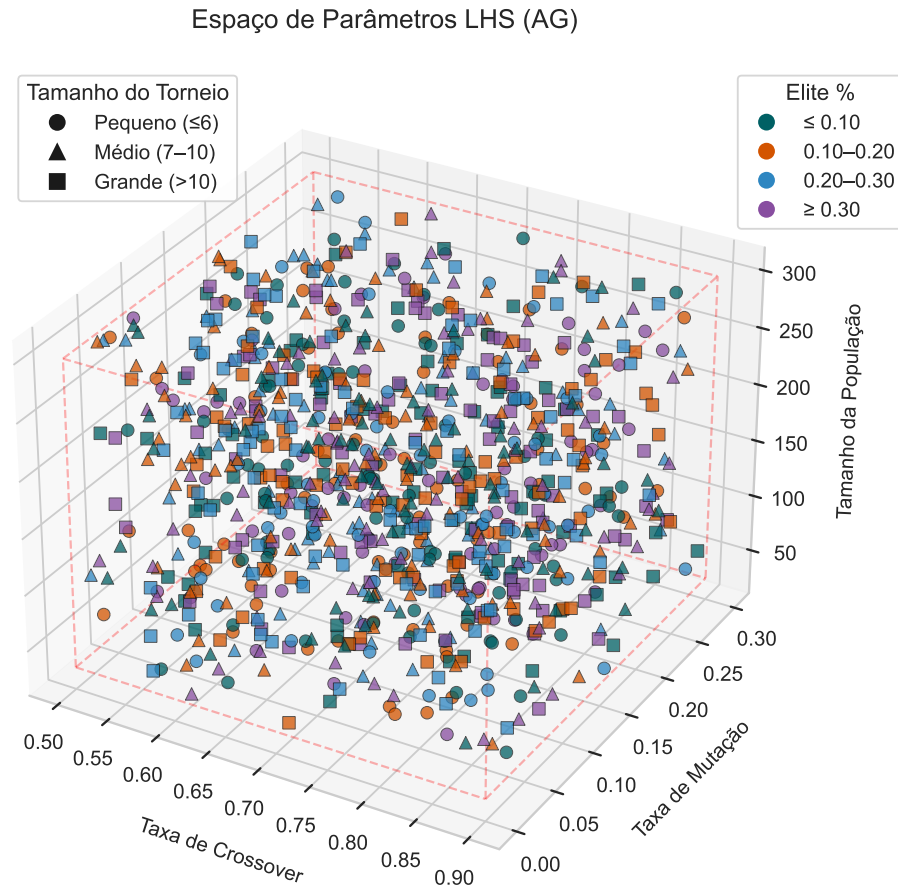
A taxa de *crossover* é essencial que seja alta para caracterizar os algoritmos genéticos, a grande vantagem do método é a combinação de blocos das boas soluções, portanto a taxa de cruzamento deve ser alta. Caso seja abaixo de 0.5, o algoritmo começa a perder a característica de recombinação e passa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

3.4.2 Cenários de Teste e Perturbação dos Dados

Os dados de referência desse trabalho são os de preferência e disciplinas ofertadas no primeiro semestre de 2025, com o objetivo de 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 não são completamente radicais, há pouca variabilidade nos dados, em grande parte pela própria natureza.

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 partir 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.

- **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 feito 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 $\mathcal{N}(\iota, \cdot/\nabla)$, 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.

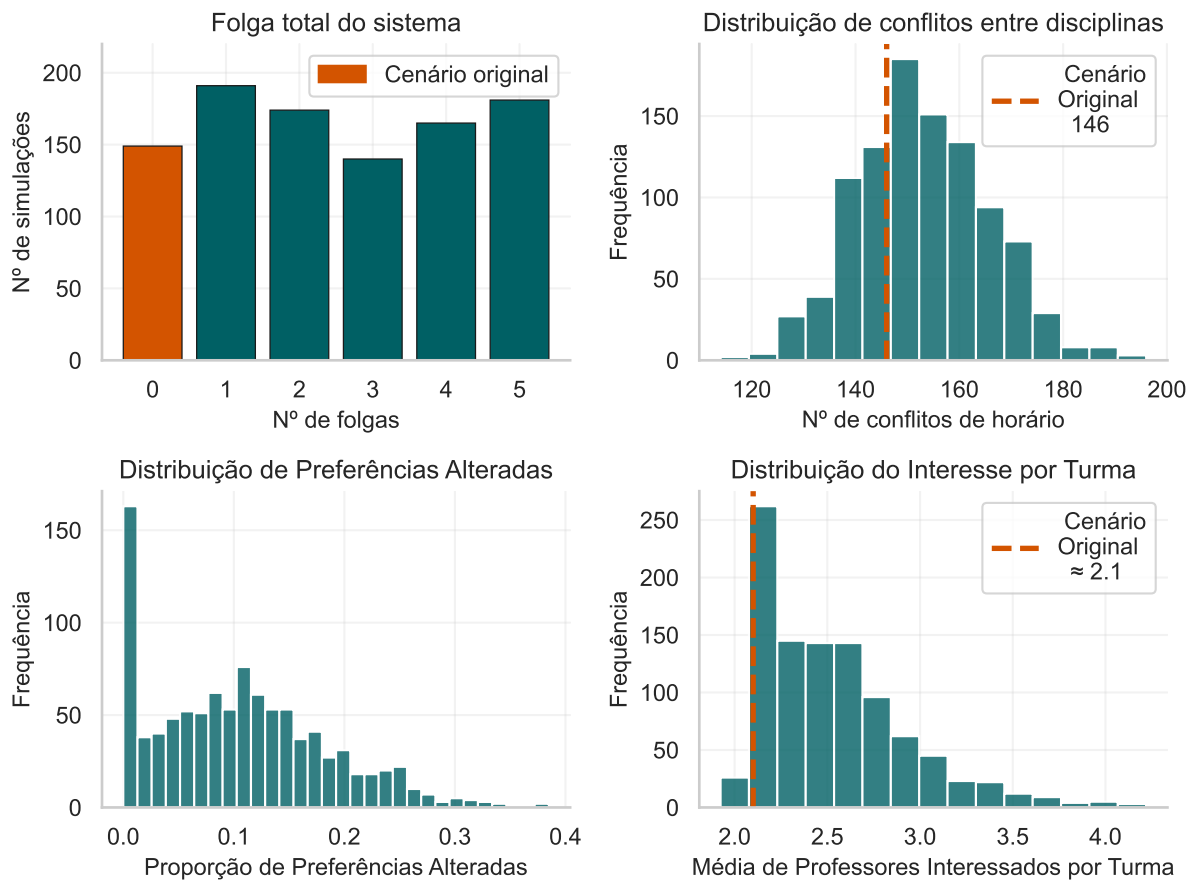
A função elaborada pode ser verificada no apêndice: ??.

- **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 5, 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 seguir é apresentado o resultado das perturbações utilizadas na simulação para análise de sensibilidade dos parâmetros.

Figura 3: Distribuição dos dados após perturbações aleatórias



A Figura 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.

3.4.3 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 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.

4.1 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.

4.2 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 4: Captura de tela da interface inicial - Aba de Docentes




Dados Iniciais Execução Resultado						
Dados Iniciais						
Professores e Disciplinas carregados para referência.						
Professores Disciplinas						
Professores e Preferências						
ID	Nome	Carga Máx.	N° Sem Preferência	N° Pref. 1	N° Pref. 2	N° 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 5: Captura de tela da interface inicial - Aba de Disciplinas



Dados Iniciais **Execução** **Resultado**

Dados Iniciais

Professores e Disciplinas carregados para referência.

[Professores](#) **Disciplinas**

Disciplinas Ofertadas

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 6: Captura de tela de execução

Dados Iniciais Execução Resultado

Executar Otimização

Selecione 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	
docente_20 - [REDACTED] ▾	EST0081_01 - TRABALHO DE CONCLUSAO DE CURSC ▾	<button>Remover</button>

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 ira 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 7: Captura de tela do resultado



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 8), 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 9).

Figura 8: Detalhes das preferências do docente

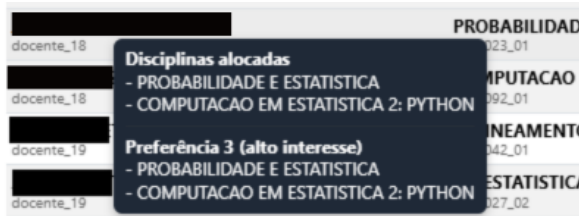
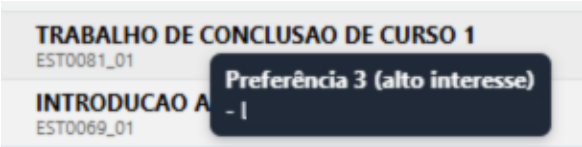


Figura 9: Detalhes dos interessados na disciplina



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.

5 Resultados

Para entender o contexto atual da dificuldade do problema para o departamento e garantir o funcionamento robusto da ferramenta, os seguintes estudos foram conduzidos.

5.1 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 primeiro semestre de 2025 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.

5.1.1 Análise da Procura por Disciplinas

Para analisar a preferência dos professores pelas disciplinas, a Tabela 3 apresenta uma contagem de quantas vezes cada disciplina obteve cada pontuação de preferência pelos professores, apresentada abaixo:

Tabela 3: Distribuição das preferências por disciplina

Disciplina	Nº de Turmas	Pref 0	Pref 1	Pref 2	Pref 3
Análise de dados categorizados	1	25	2	0	1
Análise de regressão linear	1	21	2	2	3
Análise de séries temporais	1	24	2	0	2
Análise de sobrevivência	1	25	1	0	2
Bioestatística	2	1	13	5	9
Computação em estatística 1	2	21	3	2	2
Computação em estatística 2: python	1	25	0	1	2
Computação em estatística 2: r	1	24	2	2	0
Delineamento e análise de experimentos	1	25	2	0	1
Demografia	1	27	1	0	0
Estatística aplicada	10	1	12	4	11
Estatística computacional	1	23	2	1	2
Estatística exploratória	1	21	5	0	2
Inferência bayesiana	1	26	1	0	1
Inferência estatística	1	21	2	1	4
Introdução à probabilidade	2	13	5	2	8
Laboratório de estatística 1	1	26	0	2	0
Métodos estatísticos 1	1	19	3	4	2
Métodos estatísticos 2	1	20	4	2	2
Modelos lineares generalizados	1	25	1	0	2
Práticas de extensão em estatística 2	1	27	0	1	0
Probabilidade e estatística	11	1	8	2	17
Processos estocásticos	1	20	1	3	4
Técnicas de amostragem	1	26	1	0	1
Trabalho de conclusão 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 apenas o coordenador do departamento pode optar por não selecionar. 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 máxima*

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 20 manifestou desinteresse por 46 das 47 disciplinas, indicando preferência máxima por apenas uma, o que o caracteriza como um perfil de alocação potencialmente mais restrito. Em contrapartida, o

Docente 7 demonstrou grande engajamento ou um leque de interesses mais amplo, marcando 27 disciplinas com o maior nível de preferência. Essa heterogeneidade no perfil docente é um fator chave de complexidade que a ferramenta de otimização deverá gerenciar.

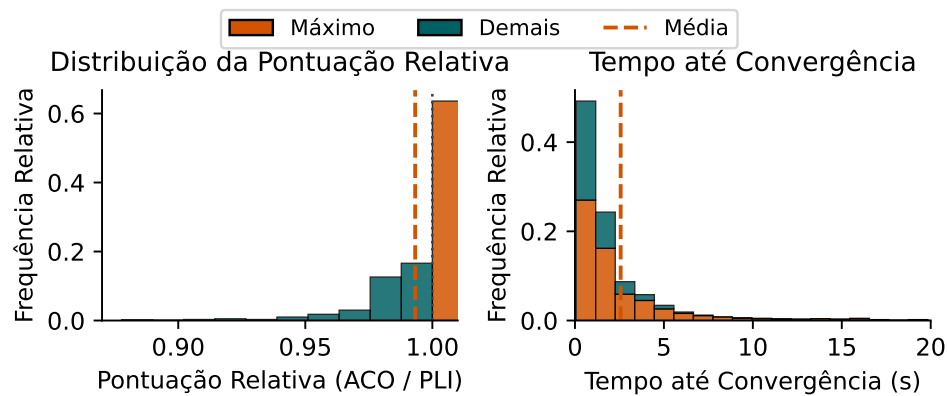
5.2 Análise e Calibração do Algoritmo de Colônia de Formigas

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.

5.2.1 Análise de sensibilidade dos hiperparâmetros

O resultado das simulações com perturbações e variando os parâmetros do ACO seguem apresentados nos gráficos a seguir.

Figura 10: Histograma de pontuação relativa e tempo até convergência ACO



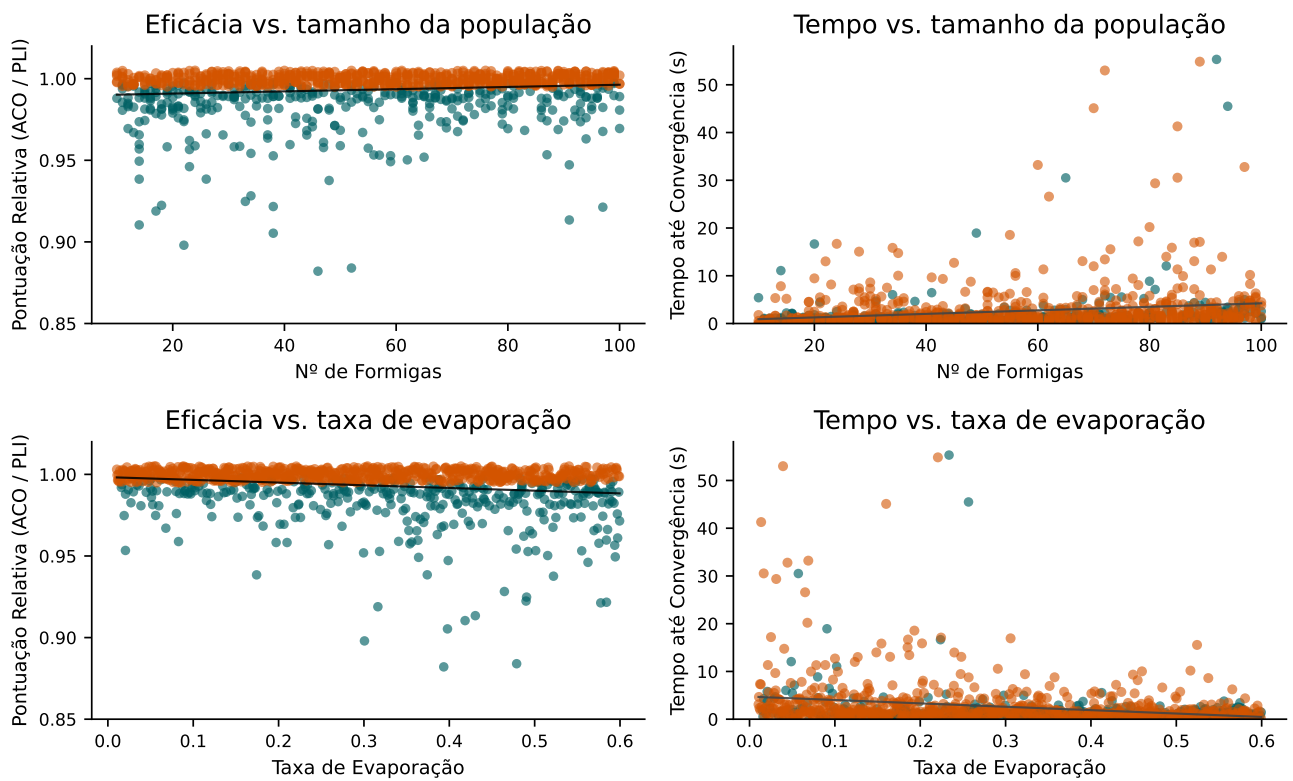
Das 1000 simulações, 636 apresentaram o escore máximo, e com mais de 80% das amostras obtendo um escore relativo acima de 99%. E podemos observar na Figura 10 que os tempos de convergência do algoritmo foram extremamente baixos, com apenas 41 das 1000 simulações obtendo um tempo de convergência acima 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. Para evidenciar a correlação entre as variáveis, foram incluídas linhas de tendência calculadas via regressão linear simples (ajuste polinomial de 1º grau). As retas representam a tendência central dos resultados, indicando como a variação de cada hiperparâmetro impacta a eficácia e o tempo de convergência do algoritmo.

Figura 11: Gráficos de dispersão entre tamanho da população e taxa de evaporação pela pontuação e tempo de convergência do ACO

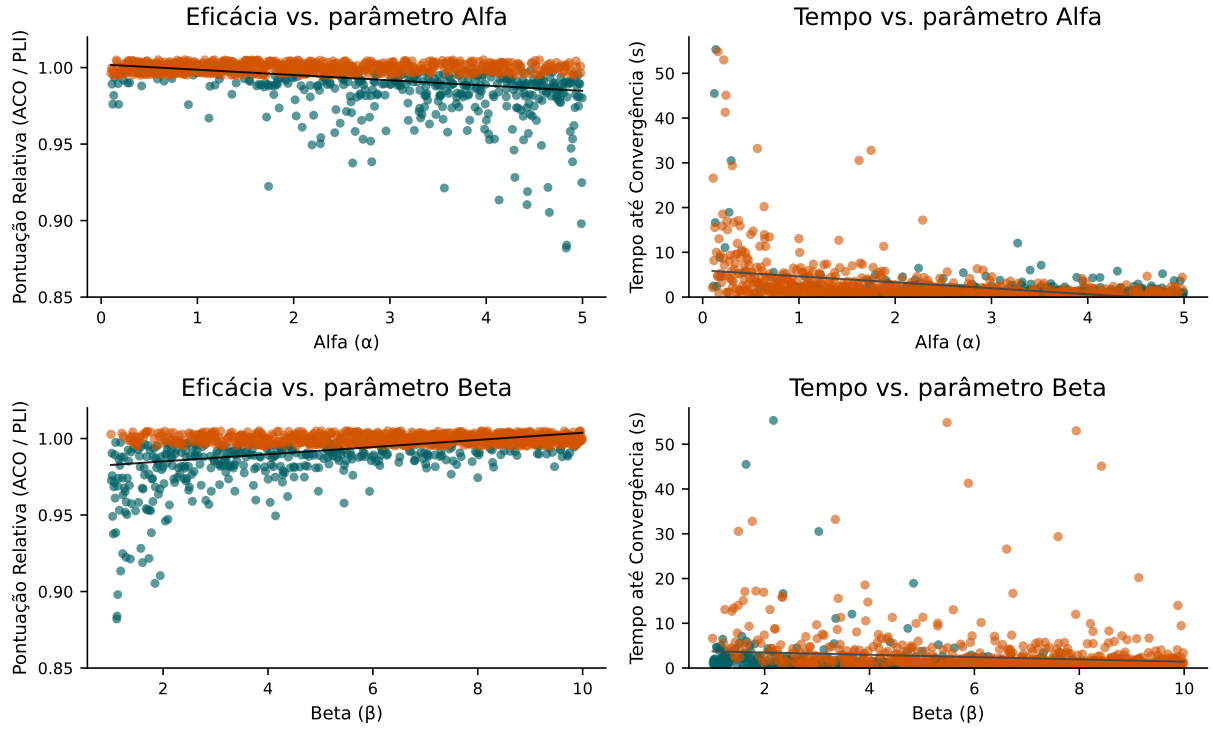


A partir da análise da Figura 11 podemos observar que não há uma relação forte entre número de formigas e pontuação, com as falhas se apresentando distribuídas de forma aleatória ao longo do intervalo. Com relação ao tempo computacional, o resultado se apresenta de acordo com o esperado, o aumento no número de formigas faz com que o número de operações a cada geração aumente e, conseqüentemente, há um aumento no tempo até a convergência. 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. Permitindo fixar a taxa de evaporação em 0.1.

Figura 12: Gráficos de dispersão entre Alfa e Beta pela pontuação e tempo de convergência do ACO



Podemos observar que os parâmetros alfa e beta possuem um efeito maior no resultado da performance do algoritmo. Observa-se uma correlação negativa entre o parâmetro Alfa e a qualidade da solução, indicando que uma forte influência do feromônio (histórico) pode levar a ótimos locais prematuros. Inversamente, o parâmetro Beta apresentou uma correlação positiva, indicando que o conhecimento acumulado das formigas ao longo das gerações é pouco importante para a garantia de convergência do algoritmo. Enquanto a informação da heurística (preferência dos professores), é essencial para as boas pontuações. Portanto, os valores fixados foram $\alpha = 1.5$ e $\beta = 8$.

5.2.2 Configuração dos Parâmetros

Após as análises, os valores apresentados na Tabela 5 foram fixados para os parâmetros do ACO.

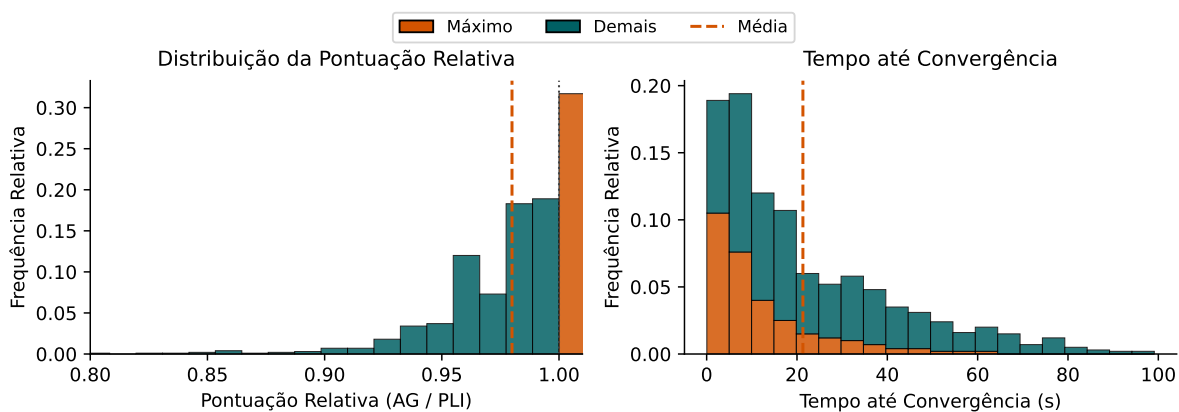
Tabela 5: Valores selecionados para os parâmetros do ACO

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

5.3 Análise e Calibração do Algoritmo Genético

5.3.1 Análise de sensibilidade dos hiperparâmetros

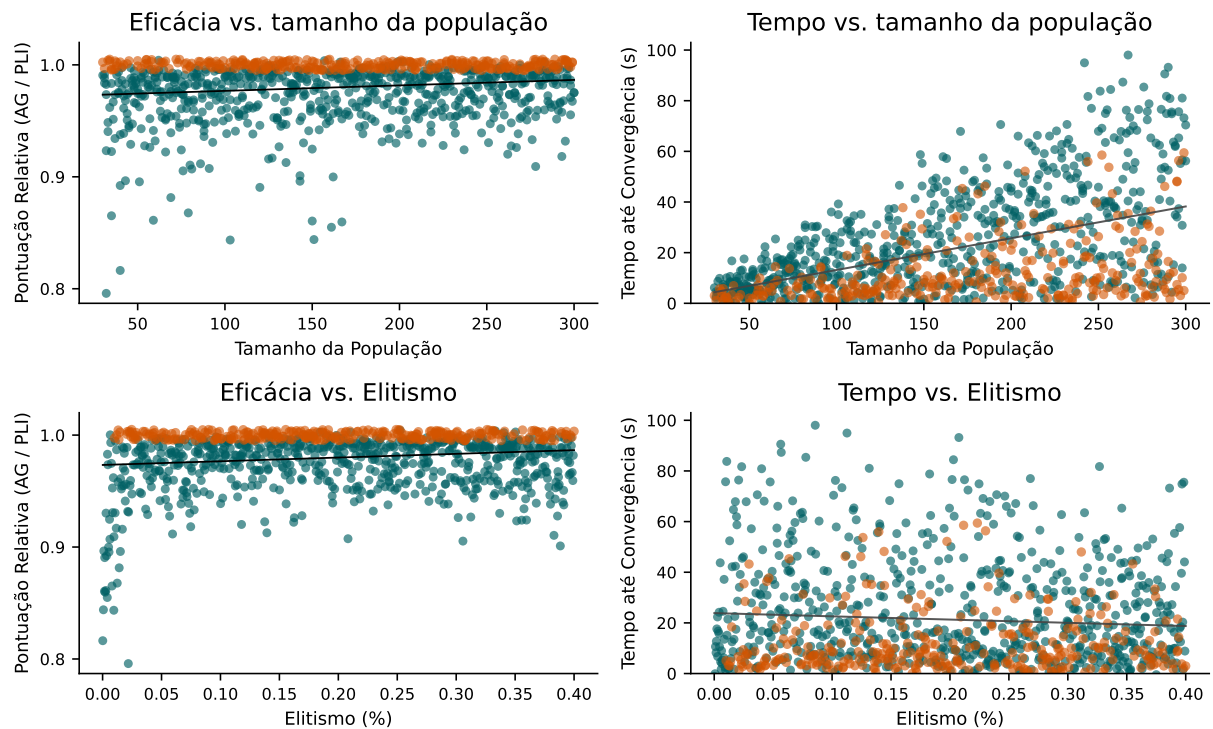
Figura 13: Gráficos de dispersão entre os hiperparâmetros e a pontuação e tempo de convergência do AG



Analisando os resultados do AG, podemos observar que a convergência do algoritmo não foi tão forte e expressiva quando 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 13 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.

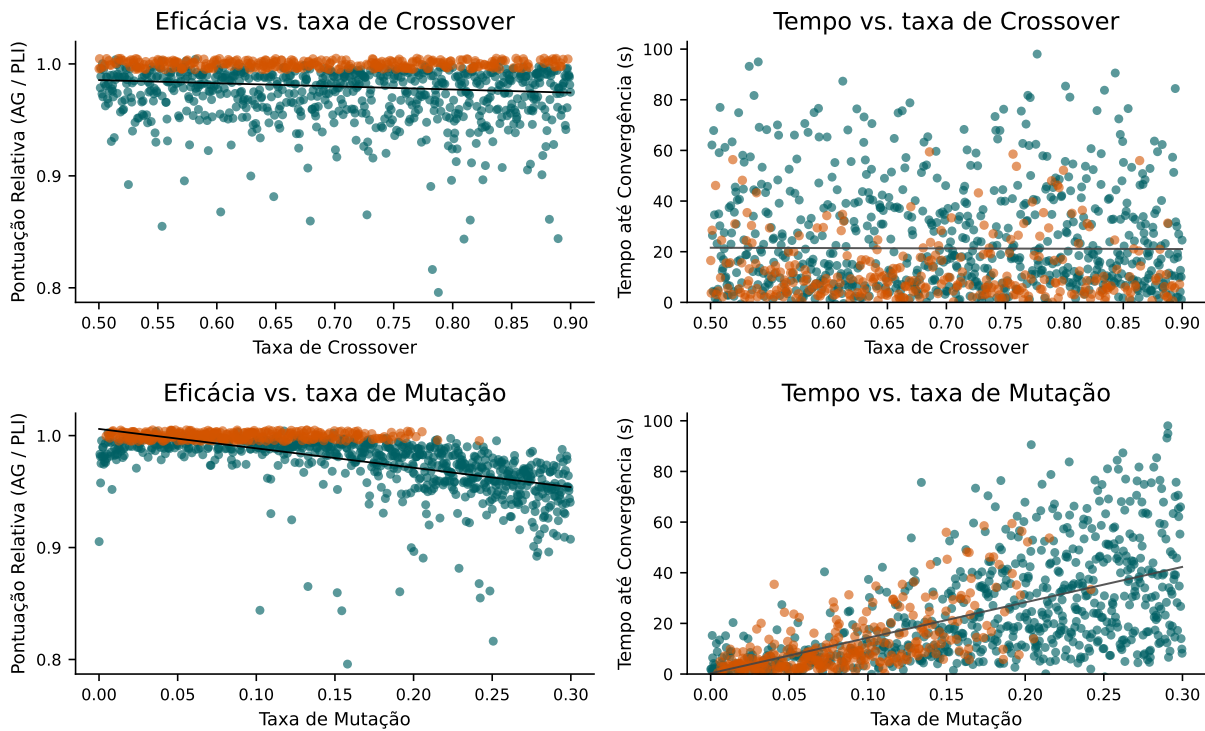
Figura 14: Gráficos de dispersão entre o tamanho da população e percentual de elitismo pela pontuação e tempo de convergência do AG



Analisando os gráficos na Figura 14 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 análise 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.

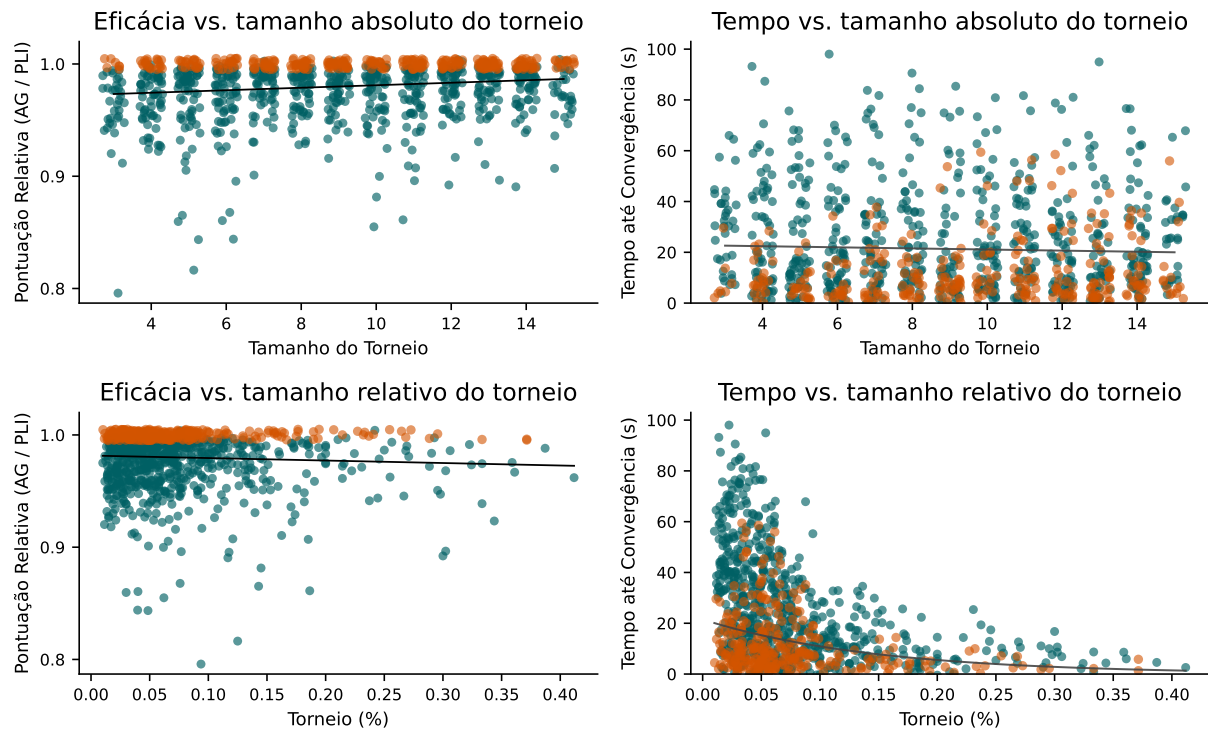
Figura 15: Gráficos de dispersão entre as taxas de *crossover* e mutação pela pontuação e tempo de convergência do AG



Observando os gráficos da Figura 15, 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 16: Gráficos de dispersão entre tamanho absoluto e relativo do torneio pela pontuação e tempo de convergência do AG



Analisando os gráficos da Figura 16, podemos observar que não há uma relação forte entre o tamanho absoluto 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 (3 a 14), e não percentual da população, os resultados com alto tempo de convergência se devem às simulações com uma grande população, que tem tamanho percentual do torneio pequeno. Confundindo as piores causadas pelas populações grandes com os resultados das simulações que tiveram um baixo percentual do grupo de torneio. Portanto, o valor utilizado para o tamanho do torneio será de 10 indivíduos.

5.3.2 Configuração dos Parâmetros

Após as análises, os valores selecionados para os parâmetros do algoritmo genético foram apresentados na Tabela 6.

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 <i>CrossOver</i>	0.7
Taxa de Mutação	0.05
Tamanho do Torneio	10

5.4 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.

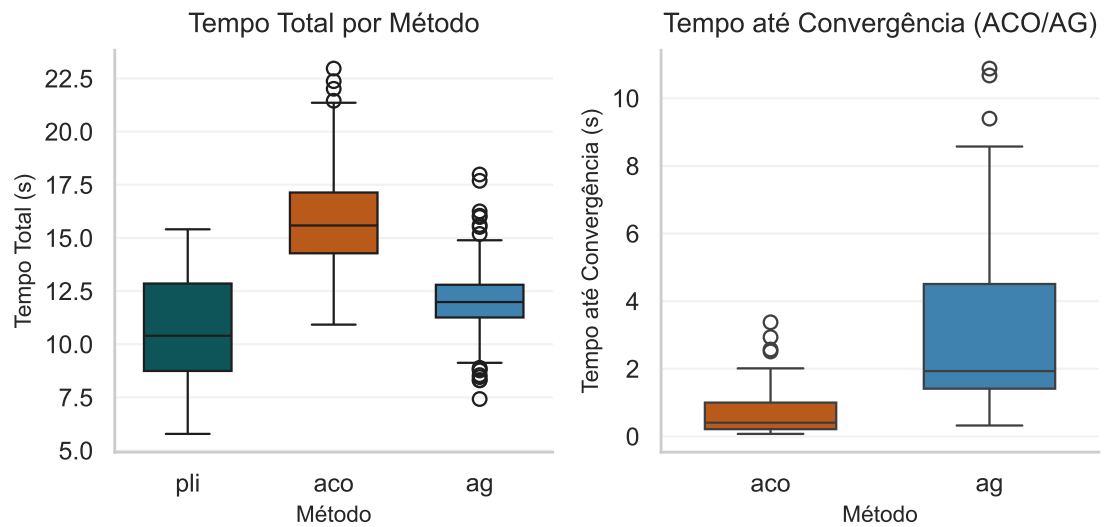
Os resultados obtidos a partir das simulações seguem apresentados na Tabela e nos gráficos a seguir.

Tabela 7: Resultado das simulações de comparação

Algoritmo	Nº de Pontuações Máximas	Média de Pontuação	Média Tempo Convergência	Média Tempo Total
PLI	-	-	10.64	10.64
AG	82	0.998	3.22	12.04
ACO	100	1	0.69	15.73

Para analisar se a boa performance é constante ao longo de diversos cenários, os gráficos a seguir foram elaborados.

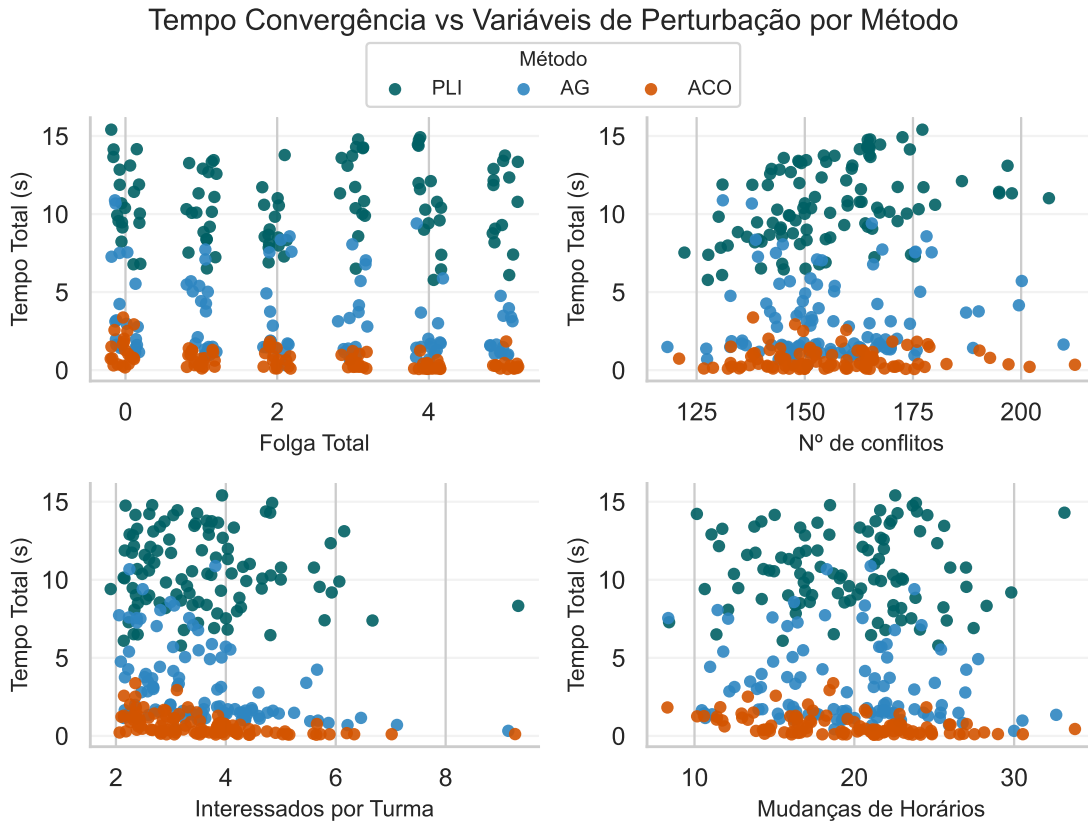
Figura 17: Boxplots do tempo de execução total e tempo até convergência dos algoritmos



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, e apresenta um tempo

Para analisar a eficiência computacional dos algoritmos nos diferentes cenários dos dados, os gráficos a seguir foram construídos.

Figura 18: Gráficos de dispersão entre o tempo até convergência e as perturbações aplicadas



Analisando os gráficos da Figura 18 podemos ver que apenas no cenário de 0 folgas o ACO apresenta uma maior variabilidade no tempo até a convergência, mas mantendo resultados extremamente bons. O AG se apresenta uma sensibilidade maior à redução nas folgas do sistema, diminuindo o tempo até a convergência conforme o número de folgas aumenta. E o PLI não aparenta sofrer nenhuma influência pela variação. O comportamento de melhora com o aumento de folgas é esperado, por permitir mais flexibilidade para que as soluções sejam construídas, enquanto com menos folgas as trocas de alocações possuem menos opções que melhorem a solução, aumentando um pouco para a convergência, apesar disso a performance dos 3 algoritmos foi consideravelmente estável.

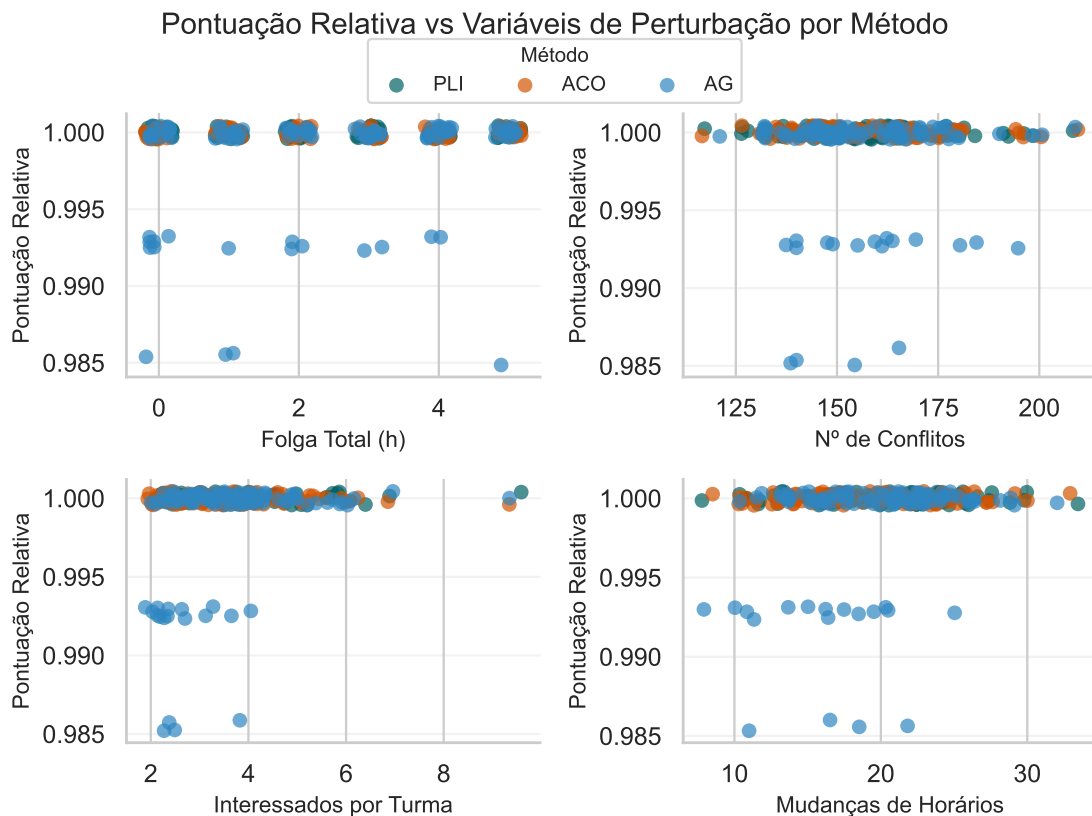
O aumento no número de conflitos nos horários das disciplinas aparenta uma influência maior no PLI por restringir mais o espaço de soluções possíveis, e para o AG há indícios de uma pequena influência, enquanto o ACO permanece inalterado. Enquanto o número de mudanças de horários não aparenta influenciar nenhum dos algoritmos, influenciando apenas quando aumenta os conflitos nas disciplinas.

As mudanças nas preferências dos professores apresenta uma relação fraca com o AG e ACO, com o aumento da média de professores interessados por turma, as opções de

alocações ficam mais abundantes e facilitam a criação de novas soluções, enquanto para a PLI não há muita diferença.

E para analisar a qualidade das soluções de acordo com a variação dos cenários, os seguintes gráficos foram construídos.

Figura 19: Gráficos de dispersão entre a pontuação relativa e as perturbações aplicadas



A PLI apresenta a solução de referência, então todas as suas pontuações foram máximas, e o ACO apresentou a solução máxima em todas as simulações, não tendo uma diminuição na qualidade da solução em nenhum cenário diferente dos dados.

O AG apresentou 18 soluções que não atingiram a pontuação máxima, e a variação nos cenários explica o comportamento das soluções com pior performance. Podemos ver que 6 falhas acontecem no cenário com folga 0, que confirmam que o algoritmo é o mais sensível por ter que fazer trocas entre pares de professores para não perder a viabilidade da solução ao testar novas preferências.

Na análise da variação do número de conflitos, podemos ver que nos cenários de alto número de conflitos há mais falhas por simulação do que as outras regiões, e as falhas na região central do número de conflitos podem estar sendo causadas pelas outras perturbações nos dados. A redução na média de professores interessados por turma tem a

maior relação com as reduções na pontuação, com a maior parte das falhas se concentrando nas regiões de menor média.

6 Conclusão

Os resultados obtidos a partir da otimização e da comparação de performance entre os 3 algoritmos, mostram que apesar de ser um problema do tipo NP-Difícil, a escala dos dados de número de professores e disciplinas do departamento fazem com que o problema seja facilmente resolvido por todos os métodos propostos.

A implementação da PLI obteve uma eficiência computacional acima das expectativas iniciais, com performances muito próximas dos algoritmos meta-heurísticos, e com a vantagem de fornecer sempre a melhor solução global. O AG e o ACO obtiveram performances surpreendentes. A adaptação dos algoritmos para favorecer as alocações com alta preferência fez com eles obtivessem performances muito boas de forma quase independente dos parâmetros ajustados, nos permitindo focar em selecionar os parâmetros que maximizassem a eficiência computacional dos métodos.

No ACO vimos que os parâmetros mais influentes na pontuação foram Alfa e Beta, que decidem a filosofia de construção das soluções do algoritmo, vimos que o aprendizado dos caminhos feito pelas formigas representado por Alfa influenciava negativamente a performance. Enquanto as soluções melhoravam com o favorecimento da heurística na construção das soluções representado por Beta.

No AG, o parâmetro que mais influenciou na qualidade das soluções foi a taxa de mutação, que para valores muito elevados apresentava uma piora drástica no comportamento do algoritmo. E os demais parâmetros foram ajustados buscando aprimorar apenas o tempo computacional até a convergência.

Analisando a implementação, as abordagens heurísticas apresentam vantagens arquiteturais. Por serem implementadas utilizando lógica procedural e estruturas de dados padrão, a verificação do código e a adaptação das regras de negócio tornam-se mais intuitivas se comparadas à rigidez da sintaxe exigida pelos *solvers* da PLI. Essa flexibilidade facilita a incorporação de novas preferências ou limitações administrativas futuras, garantindo que a ferramenta permaneça aderente à realidade dinâmica do departamento. Além de garantir mais robustez para lidar com o problema de escalabilidade, que no caso da PLI pode impactar em um aumento exponencial no custo computacional caso o número de professores ou disciplinas aumente.

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.
- MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, Taylor & Francis, v. 42, n. 1, p. 55–61, 2000.
- Mitchell, Stuart & Kean, Anita . *PuLP is an linear and mixed integer programming modeler written in Python*. 2009. Versão 3.3.0. Disponível em: <https://pypi.org/project/PuLP/>.
- PEIXOTO, K. *Sistema de Apoio à Decisão para Alocação de Docentes*. 2025. GitHub. Repositório de código-fonte. Disponível em: https://github.com/karl-peixoto/projeto_alocador.
- SANTNER, T. J. et al. *The design and analysis of computer experiments*. [S.l.]: Springer, 2003. v. 1.