Pesquisa Operacional

Prof. Rian Gabriel S. Pinheiro, 2024.2. *https://ic.ufal.br/professor/rian/*

Alocação de Horários com BRKGA: Uma Abordagem Baseada em Códigos Aleatórios Tendenciosos

José Arthur Lopes Sabino
jals@ic.ufal.br

Karla Sophia Santana da Cruz
kssc@ic.ufal.br

Resumo

Este trabalho apresenta uma abordagem baseada no algoritmo BRKGA (*Biased Random-Key Genetic Algorithm*) para o problema de alocação de horários universitários, considerando restrições rígidas (como compatibilidade de salas e disponibilidade docente) e critérios qualitativos (como continuidade pedagógica). A modelagem formal, a função de avaliação baseada em entropia e o processo evolutivo do algoritmo são detalhados. Experimentos com três níveis de complexidade demonstram a eficácia da abordagem em termos de taxa de alocação, fragmentação e tempo computacional.

Palavras-chave: Alocação de Horários, BRKGA, Metaheurísticas, Entropia, Timetabling, Ensino Superior.

1 Introdução

A alocação de horários acadêmicos, conhecida como *University Timetabling Problem* (UTP), é um desafio enfrentado por instituições de ensino superior, caracterizado por múltiplas restrições entrelaçadas envolvendo professores, turmas, salas e horários disponíveis. A elevada complexidade combinatória do problema, agravada por currículos rígidos e recursos limitados, torna o uso de métodos exatos inviável em larga escala.

Diante disso, técnicas baseadas em metaheurísticas têm sido amplamente adotadas. Algoritmos Genéticos (AGs), em particular, destacam-se por sua flexibilidade frente a múltiplos critérios e restrições. A literatura recente aponta avanços com variantes híbridas (Arias-Osorio & Mora-Esquivel, 2020), coevolutivas (Zhang, 2022) e multiobjetivo (Zhang et al., 2021). Uma dessas variantes promissoras é o BRKGA (*Biased Random-Key Genetic Algorithm*), que utiliza vetores de números reais como codificação, permitindo priorizações flexíveis na construção das soluções. Além dos Algoritmos Genéticos tradicionais, diversas abordagens como heurísticas construtivas, busca tabu, programação por restrições e GRASP têm sido aplicadas ao problema de alocação de horários, com diferentes graus de sucesso. No entanto, muitos desses métodos enfrentam dificuldades em adaptar-se a múltiplas restrições institucionais ou critérios pedagógicos mais sutis, como a continuidade das aulas. Nesse contexto, algoritmos baseados em chaves aleatórias, como o BRKGA, oferecem uma alternativa promissora por sua flexibilidade e facilidade de adaptação

a múltiplos objetivos.

Este trabalho propõe uma abordagem baseada no BRKGA para resolver o problema de alocação de horários, avaliando sua robustez frente a diferentes níveis de complexidade de instâncias simuladas. A proposta com BRKGA busca não apenas maximizar a taxa de alocação, mas também garantir continuidade pedagógica nas grades horárias, por meio de critérios como entropia e fragmentação, discutidos nas seções seguintes.

2 O Problema

A alocação de horários acadêmicos — também conhecida como University Timetabling Problem (UTP) — é um problema clássico de otimização combinatória, amplamente estudado devido à sua alta complexidade e relevância prática em instituições de ensino. Consiste em organizar os horários e locais em que cada disciplina será ministrada, garantindo que não haja conflitos de horário para turmas, professores e salas, além de atender às exigências pedagógicas e operacionais da instituição.

Cada disciplina requer uma certa quantidade de aulas por semana, deve ser ministrada por um professor específico, em uma sala com características compatíveis (por exemplo, laboratório, auditório, sala comum), e precisa ser encaixada em uma grade horária predefinida. O objetivo é construir uma programação semanal completa e coerente, respeitando simultaneamente múltiplas restrições:

Restrições operacionais:

- Evitar sobreposição de aulas para professores, salas e turmas;
- Garantir que cada aula ocorra em um local compatível com o tipo de sala exigido.

Restrições pedagógicas:

- Favorecer uma distribuição lógica das aulas ao longo da semana;
- Evitar fragmentação excessiva, como uma mesma disciplina distribuída em muitos dias ou horários distantes.

Devido ao grande número de combinações possíveis e às restrições interdependentes, o UTP é classificado como um problema $\mathcal{NP}$-completo— ou seja, não há um algoritmo conhecido capaz de encontrar soluções ótimas em tempo polinomial. Essa complexidade pode ser associada ao problema de coloração de grafos, no qual cada vértice representa um evento (como uma aula), e arestas indicam conflitos que impedem a sobreposição. Atribuir cores distintas a vértices adjacentes equivale à definição de horários distintos, o que ilustra bem o grau de dificuldade do problema e a motivação para o uso de técnicas heurísticas (Jain & Pydimarri, 2018).

Neste trabalho, adotamos uma abordagem baseada no BRKGA (Biased Random-Key Genetic Algorithm), visando não apenas atender às restrições operacionais — como disponibilidade de professores, salas e turmas —, mas também incorporar critérios pedagógicos, como a continuidade das aulas. Em particular, introduzimos o conceito de entropia temporal, que quantifica o grau de dispersão das aulas de uma disciplina ao longo da semana. Valores elevados de entropia indicam fragmentação excessiva (como aulas muito espaçadas), o que pode prejudicar a aprendizagem e dificultar o planejamento docente. Assim, a proposta busca minimizar essa entropia, promovendo grades mais concentradas e eficazes.

2.1 Modelagem

Conjuntos:

- C: conjunto de turmas, indexado por c;
- D: conjunto de disciplinas, indexado por d;
- P: conjunto de professores, indexado por p;
- S: conjunto de salas, indexado por s;
- T: conjunto de horários disponíveis, representado como pares (g, h) de dia g e hora h;
- R: conjunto de tipos de sala, indexado por r.

Parâmetros:

- u_d: número de unidades letivas necessárias para a disciplina d;
- p_d: professor responsável pela disciplina d;
- r_d: tipo de sala exigido pela disciplina d;
- s_r: subconjunto de salas do tipo r;
- A: conjunto total de alocações possíveis (c, d, t, s), onde $t \in T$ e s é compatível com r_d.

Variáveis de decisão:

- $x_{c,d,t,s} \in \{0, 1\}$: vale 1 se a disciplina d for alocada para a turma c no horário t e na sala s; 0 caso contrário.

Função Objetivo: Maximizar a alocação ponderada pela entropia e penalização de não alocação:

$$\max \sum_{(c,d,t,s)\in A} \alpha \cdot x_{c,d,t,s} - \beta \cdot E_{c,d} - \gamma \cdot N_{c,d}$$

Onde:

- $E_{c,d}$: entropia da distribuição temporal das aulas da disciplina d na turma c;
- $N_{c,d}$: número de unidades letivas não alocadas para (c, d).

Restrições:

1. **Carga horária por disciplina:**

$$\sum_{t\in T} \sum_{s\in s_{r_d}} x_{c,d,t,s} = u_d, \quad \forall c \in C, d \in D$$

2. **Sem sobreposição para turmas:**

$$\sum_{d\in D} \sum_{s\in S} x_{c,d,t,s} \leq 1, \quad \forall c \in C, t \in T$$

3. **Sem sobreposição para professores:**

$$\sum_{c \in C} \sum_{\substack{d \in D \\ p_d = p}} \sum_{s \in S} x_{c,d,t,s} \leq 1, \quad \forall p \in P, t \in T$$

4. **Sem sobreposição para salas:**

$$\sum_{c \in C} \sum_{d \in D} x_{c,d,t,s} \leq 1, \quad \forall s \in S, t \in T$$

5. **Compatibilidade de sala:**

$$x_{c,d,t,s} = 0, \quad \text{se } s \notin s_{r_d}$$

Essa modelagem fundamenta a verificação computacional de viabilidade implementada no código, que analisa previamente a suficiência de recursos por tipo de sala e por professor antes da execução do algoritmo.

Além da viabilidade técnica da alocação, o BRKGA aqui proposto considera também critérios qualitativos, como entropia e fragmentação temporal. A função objetivo pondera essas penalizações para equilibrar a completude da alocação com a continuidade pedagógica. Os experimentos avaliam o comportamento do algoritmo sob três níveis distintos de carga e competição por recursos: instâncias fáceis, médias e difíceis.

3 A Solução

3.1 Inicialização e Representação

A etapa de inicialização no BRKGA é fundamental para garantir diversidade e qualidade na população inicial de soluções. Diferentemente dos algoritmos genéticos tradicionais, que utilizam permutações discretas, o BRKGA representa cada solução como um vetor de chaves contínuas, com valores no intervalo $[0, 1]$. Cada chave representa a prioridade relativa de uma disciplina no processo de alocação.

A representação contínua evita duplicações, facilita cruzamentos e permite ordenações parciais mais flexíveis.

A geração dos vetores de prioridades pode ser realizada segundo diferentes estratégias, definidas pela modalidade de inicialização:

- **Aleatória:** os valores são sorteados uniformemente no intervalo $[0, 1]$;
- **Sala escassa:** disciplinas que exigem salas com menor disponibilidade são priorizadas;
- **Combinada:** disciplinas de professores com alta carga horária ou múltiplas turmas recebem prioridade;
- **Carga horária:** disciplinas com maior carga são priorizadas diretamente.

Independentemente da modalidade, um ruído aleatório é adicionado ao vetor para aumentar a diversidade populacional. Os valores são então normalizados para o intervalo $[0, 1]$, e o vetor resultante é passado para a função de decodificação, que gera a estrutura completa do indivíduo.

A seguir, apresenta-se o pseudocódigo que descreve esse processo de geração inicial:

Algoritmo 1 Gerar indivíduo inicial

Require: Modalidade de geração (`modalidade`), dados do problema (`dados`)
Ensure: Indivíduo com vetor de prioridades e alocação decodificada
if modalidade = "aleatório" **then**
    Gerar vetor de prioridades randômico
else if modalidade = "sala_escassa" **then**
    Ordenar disciplinas pela razão carga horária / número de salas compatíveis
else if modalidade = "combinada" **then**
    Priorizar disciplinas de professores com muitas aulas e alta carga
else
    Ordenar disciplinas por carga horária decrescente
end if
Adicionar ruído aleatório ao vetor de prioridades
Normalizar os valores para o intervalo $[0, 1]$
Decodificar vetor usando `decode_individual(keys, dados)`
return Indivíduo decodificado

Esse vetor será usado na fase de decodificação, que transforma a ordenação implícita das disciplinas em alocações reais na grade semanal.

3.2 Decodificação das Soluções

O processo de decodificação segue uma lógica simples: para cada disciplina e turma, os horários são percorridos em ordem crescente de dias e horas, buscando sempre o primeiro bloco de tempo viável que permita a alocação completa das unidades letivas. Não há reordenamento adaptativo durante essa fase — uma vez que uma posição viável é encontrada, ela é imediatamente utilizada. Isso torna o processo eficiente e previsível, embora possa deixar de explorar posições alternativas potencialmente melhores.

O algoritmo percorre todas as disciplinas em ordem de prioridade, tentando alocar suas unidades letivas para cada turma. A alocação ocorre apenas se três condições forem simultaneamente satisfeitas: (I) o professor responsável estiver disponível, (II) houver uma sala do tipo apropriado livre naquele horário, e (III) a turma não estiver alocada a outra disciplina no mesmo horário.

Para garantir alocações viáveis e evitar conflitos, o algoritmo realiza bloqueios preventivos de horários. Sempre que uma aula é alocada, o horário correspondente é marcado como indisponível para todas as outras turmas e disciplinas potencialmente conflitantes. Além disso, são priorizados blocos contíguos de aulas (por exemplo, duas unidades de uma disciplina consecutivas), o que favorece a continuidade pedagógica.

Caso não seja possível alocar todas as unidades letivas de uma disciplina nos blocos preferenciais, o algoritmo executa um mecanismo de fallback, alocando as unidades restantes de forma pontual em horários livres, respeitando sempre as restrições de disponibilidade e compatibilidade.

Algoritmo 2 Decodificar uma solução

```
Ordenar disciplinas pela prioridade (vetor de chaves)
for cada disciplina d e turma t do
    while unidades restantes de d não alocadas do
        for cada horário e sala disponível do
            if professor, sala e turma estão livres then
                Alocar disciplina em (t, s, h)
                Atualizar matrizes de disponibilidade
            end if
        end for
    end while
end for
```

3.3 Função de Avaliação

A função de avaliação combina critérios quantitativos e qualitativos. A taxa de alocação, $A(x)$, mede diretamente a proporção de aulas que foram efetivamente alocadas. A entropia média $E(x)$ atua como penalidade para distribuições muito fragmentadas. Já o termo $N(x)$ contabiliza o número de unidades letivas que não foram alocadas, penalizando fortemente soluções incompletas.

A estrutura ponderada da função objetivo, parametrizada por α, β e γ, permite adaptar a estratégia de busca conforme o foco desejado — maximização da alocação, continuidade horária ou penalização de falhas.

Logo, a qualidade de cada solução é medida por meio de uma função fitness que combina três elementos principais: taxa de alocação bem-sucedida, penalidade por entropia e penalidade por não alocação. A fórmula é expressa por:

$$F(x) = \alpha \cdot A(x) - \beta \cdot E(x) - \gamma \cdot N(x) \tag{1}$$

A seguir, o pseudo-código da avaliação:

Algoritmo 3 Avaliar uma solução

```
Calcular A(x) ← taxa de alocação
Calcular E(x) ← entropia média da grade
Calcular N(x) ← número de aulas não alocadas
F(x) ← α · A(x) − β · E(x) − γ · N(x)
return F(x)
```

A entropia é calculada com base na distribuição das aulas ao longo dos dias da semana, penalizando soluções com grande fragmentação.

Os pesos são ajustados dinamicamente via funções sigmoides parametrizadas pela taxa de alocação, o que permite ao algoritmo focar na viabilidade em estágios iniciais e, gradualmente, refinar a qualidade da grade horária. Além da entropia, também é considerada a fragmentação, calculada como o número médio de dias distintos em que uma mesma disciplina é ministrada.

3.4 Processo Evolutivo

A lógica do BRKGA inclui a divisão da população em três grupos: elite, mutantes e descendentes. O cruzamento é realizado de forma enviesada — para cada posição no vetor de prioridades, há uma probabilidade ρ_e de herdar o valor do indivíduo elite, e $1 - \rho_e$ de herdar do outro genitor. Essa abordagem equilibra a preservação de boas soluções com a geração de novas combinações promissoras.

O algoritmo também adota uma estratégia de reinicialização parcial para mitigar a estagnação evolutiva. Sempre que o progresso evolutivo se estabiliza por muitas gerações consecutivas, uma parte da população é substituída por novos indivíduos aleatórios, promovendo a diversidade e ajudando a escapar de mínimos locais.

O processo evolutivo foi testado com diferentes tamanhos de população, modos de cruzamento (enviesado, interpolado e híbrido) e com ou sem o uso de busca local. Essas variações foram avaliadas em conjunto com três instâncias de entrada — fácil, média e difícil — permitindo observar o comportamento do algoritmo sob diferentes níveis de complexidade e restrição. Os resultados obtidos demonstram a capacidade do BRKGA de adaptar-se a diferentes cenários e alcançar soluções viáveis e de qualidade em um tempo computacional competitivo.

Algoritmo 4 Executar BRKGA

```
Inicializar população com estratégias diversas
while geração atual ≤ máximo ou sem convergência do
    Avaliar população
    Selecionar elite, mutantes e descendentes
    Realizar cruzamentos enviesados
    if fitness estagnado then
        Reinicializar parte da população
    end if
    Atualizar melhor indivíduo
end while
return melhor solução encontrada
```

4 Resultados e Discussões

Foram realizados testes com três instâncias de entrada simuladas, categorizadas como fácil, média e difícil, de acordo com o número de turmas, disciplinas, professores e tipos de sala. Para cada instância, o algoritmo foi executado sob três configurações distintas de parâmetros, variando o tamanho da população, o número máximo de gerações, o tipo de cruzamento e o uso da busca local.

As métricas avaliadas foram: fitness final, taxa de alocação e tempo de execução. Para cada combinação de instância e configuração, foram realizadas múltiplas execuções, gerando análises estatísticas com boxplots e seleção do melhor indivíduo. As grades horárias geradas para as turmas foram visualizadas para inspeção qualitativa.

Os resultados indicam que o algoritmo é capaz de se adaptar a diferentes graus de complexidade, mantendo boa eficiência computacional e qualidade nas alocações. A configuração que utiliza busca local e cruzamento híbrido (Configuração B) obteve, em geral, os melhores resultados de fitness, especialmente em instâncias mais difíceis, embora com maior tempo de execução.

4.1 Configurações dos Experimentos

Três configurações distintas de parâmetros foram definidas para o BRKGA, visando avaliar o impacto de diferentes estratégias evolutivas e do uso de busca local. A Tabela 1 resume os principais parâmetros utilizados em cada configuração.

Tabela 1: Configurações utilizadas nos experimentos

Configuração	Tamanho da População	Gerações	Cruzamento	Busca Local
A	100	10	Biased	Não
B	150	20	Híbrido (mix)	Sim
C	200	15	Interpolado	Não

4.2 Instâncias de Entrada

Foram utilizadas três instâncias simuladas de alocação de horários, classificadas conforme o nível de dificuldade: fácil, média e difícil. A Tabela 2 apresenta um resumo das características principais de cada uma.

Tabela 2: Resumo das instâncias de entrada utilizadas

Instância	Turmas	Disciplinas	Professores	Tipos de Sala	Slots Semanais
Fácil	2	10	10	Sala (6), Lab (5)	35
Média	4	20	20	Sala (4), Lab (2)	50
Difícil	6	20	20	Sala (5), Lab (3)	55

4.3 Análise Quantitativa por Instância (Configuração B)

Para avaliar o comportamento do BRKGA sob diferentes condições, foram realizadas múltiplas execuções para cada instância de entrada utilizando a Configuração B. Os boxplots a seguir mostram a variação dos valores de *fitness* e de taxa de alocação em cenários de dificuldade crescente.

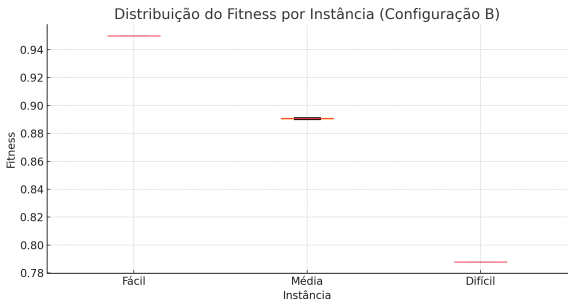

Figura 1: Distribuição do *fitness* por instância (Configuração B).

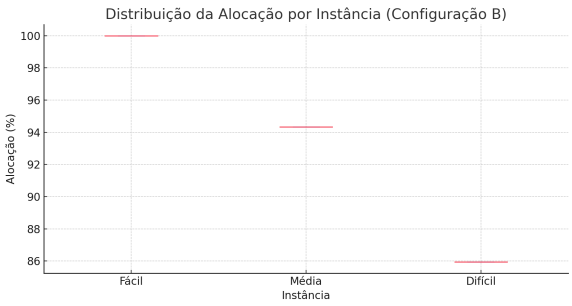

Figura 2: Distribuição da taxa de alocação por instância (Configuração B).

Os resultados demonstram que o BRKGA mantém alta taxa de alocação e estabilidade mesmo em cenários com crescente complexidade. À medida que a dificuldade das instâncias aumenta — seja pelo maior número de turmas, professores ou escassez de recursos —, observa-se um crescimento na variabilidade dos resultados, refletindo o aumento na competição por slots viáveis. Ainda assim, a qualidade das soluções se mantém elevada.

Tabela 3: Comparação dos Resultados por Configuração e Instância

Cfg	Instância	Fitness médio	Desv. Padrão	Mín.	Máx.	Alocação Média (%)	DP	Mín.	Máx.	Tempo (s)
A	Fácil	0,9500	0,0000	0,9500	0,9500	100,00	0,0000	100,00	100,00	**91,84**
B	Fácil	0,9500	0,0000	0,9500	0,9500	100,00	0,0000	100,00	100,00	214,58
C	Fácil	0,9500	0,0000	0,9500	0,9500	100,00	0,0000	100,00	100,00	218,92
A	Média	**0,8905**	0,0009	0,8895	**0,8913**	94,34	0,0000	94,34	94,34	**80,53**
B	Média	**0,8905**	0,0009	0,8895	**0,8913**	94,34	0,0000	94,34	94,34	306,34
C	Média	0,8903	0,0006	0,8895	0,8907	94,34	0,0000	94,34	94,34	383,82
A	Difícil	0,9467	0,0000	0,9467	0,9467	100,00	0,0000	100,00	100,00	**115,78**
B	Difícil	**0,9484**	0,0000	**0,9484**	**0,9484**	100,00	0,0000	100,00	100,00	366,12
C	Difícil	0,9467	0,0000	0,9467	0,9467	100,00	0,0000	100,00	100,00	402,56

Entre as configurações testadas, a Configuração B se destaca por alcançar, de forma consistente, os melhores valores de fitness, especialmente nas instâncias de nível médio e difícil. Esse desempenho superior é atribuído à combinação do cruzamento híbrido com busca local, que favorece uma exploração mais refinada do espaço de busca, permitindo escapar de mínimos locais e otimizar a continuidade pedagógica das grades.

Em contrapartida, o custo computacional da Configuração B é significativamente maior, o que pode limitar sua aplicabilidade em contextos com restrições de tempo ou capacidade de processamento. Nesses casos, a Configuração A oferece um bom compromisso, apresentando tempos de execução mais curtos e ainda mantendo boa qualidade nas alocações, sobretudo nas instâncias mais simples.

Essa análise mostra que os parâmetros do BRKGA podem ser ajustados conforme a realidade institucional: cenários que demandam rapidez na geração de horários podem optar por configurações mais enxutas, enquanto situações que exigem maior qualidade e coerência pedagógica podem justificar o uso de estratégias mais intensivas, como a Configuração B.

A Tabela 3 consolida os resultados médios de fitness, alocação e tempo de execução para cada combinação de configuração e instância. Os melhores resultados de cada grupo estão destacados em negrito.

4.4 Visualização das Grades Geradas

Além das análises quantitativas, foram geradas visualizações das grades horárias resultantes para duas turmas de cada instância de entrada (fácil, média e difícil), utilizando a Configuração B. As visualizações a seguir ilustram como o BRKGA distribuiu as disciplinas nas turmas das três instâncias, respeitando restrições de disponibilidade e promovendo continuidade pedagógica.

Observa-se, em geral, que o BRKGA foi capaz de construir grades viáveis sem sobreposição de recursos, promovendo continuidade em disciplinas práticas, mantendo blocos contínuos para disciplinas práticas e uma ocupação equilibrada dos recursos.

Comparado a outras estratégias conhecidas — como GRASP, Tabu Search e Programação por Restrições — o BRKGA apresentou maior flexibilidade na incorporação de critérios qualitativos como entropia e fragmentação, o que o torna especialmente adequado a contextos pedagógicos reais.

	Segunda	Terca	Quarta	Quinta	Sexta
08:00	Ética Profissional Ethics - Sala1	Banco de Dados Coutinho - Lab1	Matemática Discreta Pedro - Sala2		
09:00	Inglês Técnico English - Sala2	Banco de Dados Coutinho - Lab1	Matemática Discreta Pedro - Sala2		
10:00	Metodologia Científica Academic - Sala1	Algoritmos Willy - Lab2	Lógica Laura - Sala1		
11:00	Metodologia Científica Academic - Sala1	Algoritmos Willy - Lab2	Lógica Laura - Sala1		
14:00	Engenharia de Software Arturo - Sala2	Álgebra Linear João - Sala1	Programação I Paes - Lab1		
15:00	Engenharia de Software Arturo - Sala2	Álgebra Linear João - Sala1	Programação I Paes - Lab1		
16:00			Programação I Paes - Lab1		

(a) Fácil - Turma A

	Segunda	Terca	Quarta	Quinta	Sexta
08:00	Inglês Técnico English - Sala2	Algoritmos Willy - Lab2	Álgebra Linear João - Sala1	Programação I Paes - Lab1	
09:00	Ética Profissional Ethics - Sala1	Algoritmos Willy - Lab2	Álgebra Linear João - Sala1	Programação I Paes - Lab1	
10:00	Engenharia de Software Arturo - Sala2	Banco de Dados Coutinho - Lab1		Programação I Paes - Lab1	
11:00	Engenharia de Software Arturo - Sala2	Banco de Dados Coutinho - Lab1			
14:00	Metodologia Científica Academic - Sala1	Matemática Discreta Pedro - Sala2	Lógica Laura - Sala1		
15:00	Metodologia Científica Academic - Sala1	Matemática Discreta Pedro - Sala2	Lógica Laura - Sala1		
16:00					

(b) Fácil - Turma B

Figura 3: Grades horárias geradas para a instância fácil (Configuração B).

	Segunda	Terca	Quarta	Quinta	Sexta
07:00	Calculo II Xu - Sala1	Algoritmos e Estruturas de Dado Willy - Lab1	Matematica Discreta Pedro - Sala1	Programacao 1 Paes - Lab1	Redes de Computadores Almir - Lab1
08:00	Calculo II Xu - Sala1	Algoritmos e Estruturas de Dado Willy - Lab1	Matematica Discreta Pedro - Sala1	Programacao 1 Paes - Lab1	Redes de Computadores Almir - Lab1
09:00	Calculo II Xu - Sala1	Algoritmos e Estruturas de Dado Willy - Lab1	Matematica Discreta Pedro - Sala1	Programacao 1 Paes - Lab1	Empreendedorismo Business - Sala2
10:00	Engenharia de Software Arturo - Sala2	Algoritmos e Estruturas de Dado Willy - Lab1	Estatistica Laura - Sala2	Programacao 1 Paes - Lab1	Empreendedorismo Business - Sala2
11:00	Engenharia de Software Arturo - Sala2	Algoritmos e Estruturas de Dado Willy - Lab1	Estatistica Laura - Sala2	Programacao 2 Márcio - Lab2	Banco de Dados Coutinho - Lab2
13:00	Engenharia de Software Arturo - Sala2	Sistemas Operacionais Aquino - Lab2	Algebra Linear Joao - Sala2	Programacao 2 Márcio - Lab2	Ingles Tecnico English - Sala3
14:00	Metodologia Científica Academic - Sala1	Sistemas Operacionais Aquino - Lab2	Algebra Linear Joao - Sala1	Programacao 2 Márcio - Lab2	Calculo I Xu - Sala2
15:00	Metodologia Científica Academic - Sala1	Sistemas Operacionais Aquino - Lab2	Algebra Linear Joao - Sala1	Programacao 2 Márcio - Lab2	Calculo I Xu - Sala1
16:00	Gestao de Projetos Manager - Sala2	Geometria Analitica Tiago - Sala3	Inteligencia Artificial Aydano - Lab2	Comunicacao e Expressao Language - Sala2	Calculo I Xu - Sala1
17:00	Gestao de Projetos Manager - Sala2	Geometria Analitica Tiago - Sala3	Inteligencia Artificial Aydano - Lab2	Comunicacao e Expressao Language - Sala1	Ingles Tecnico English - Sala2

(a) Média - Turma A

	Segunda	Terca	Quarta	Quinta	Sexta
07:00	Engenharia de Software Arturo - Sala2	Sistemas Operacionais Aquino - Lab2	Geometria Analitica Tiago - Sala2	Programacao 2 Márcio - Lab2	Comunicacao e Expressao Language - Sala2
08:00	Engenharia de Software Arturo - Sala2	Sistemas Operacionais Aquino - Lab2	Geometria Analitica Tiago - Sala2	Programacao 2 Márcio - Lab2	Comunicacao e Expressao Language - Sala2
09:00	Engenharia de Software Arturo - Sala2	Sistemas Operacionais Aquino - Lab2	Empreendedorismo Business - Sala2	Programacao 2 Márcio - Lab2	Redes de Computadores Almir - Lab1
10:00	Calculo II Xu - Sala1	Metodologia Científica Academic - Sala3	Matematica Discreta Pedro - Sala1	Programacao 2 Márcio - Lab2	Redes de Computadores Almir - Lab1
11:00	Calculo II Xu - Sala1	Metodologia Científica Academic - Sala3	Matematica Discreta Pedro - Sala1	Programacao 1 Paes - Lab1	Algebra Linear Joao - Sala1
13:00	Calculo II Xu - Sala1	Algoritmos e Estruturas de Dado Willy - Lab1	Matematica Discreta Pedro - Sala1	Programacao 1 Paes - Lab1	Algebra Linear Joao - Sala1
14:00	Gestao de Projetos Manager - Sala2	Algoritmos e Estruturas de Dado Willy - Lab1	Calculo I Xu - Sala2	Programacao 1 Paes - Lab1	Algebra Linear Joao - Sala1
15:00	Gestao de Projetos Manager - Sala2	Algoritmos e Estruturas de Dado Willy - Lab1	Calculo I Xu - Sala2	Programacao 1 Paes - Lab1	Ingles Tecnico English - Sala3
16:00	Estatistica Laura - Sala3	Algoritmos e Estruturas de Dado Willy - Lab1	Calculo I Xu - Sala2	Inteligencia Artificial Aydano - Lab2	Ingles Tecnico English - Sala3
17:00	Estatistica Laura - Sala3	Algoritmos e Estruturas de Dado Willy - Lab1	Empreendedorismo Business - Sala2	Inteligencia Artificial Aydano - Lab2	Etica Profissional Ethics - Sala3

(b) Média - Turma B

	Segunda	Terca	Quarta	Quinta	Sexta
07:00	Matemática Discreta Xu - Sala4	Ética Willy - Sala2	Empreendedorismo Willy - Sala1	Redes Carlos - Lab3	Álgebra Linear Paulo - Sala3
08:00	Matemática Discreta Xu - Sala4	Ética Willy - Sala2	Empreendedorismo Willy - Sala1	Redes Carlos - Lab3	Álgebra Linear Paulo - Sala3
09:00	Matemática Discreta Xu - Sala4	Química Gustavo - Sala2	Gestão de Projetos Felipe - Sala2	Redes Carlos - Lab3	
10:00	Probabilidade Pedro - Sala5	Química Gustavo - Sala2	Gestão de Projetos Felipe - Sala2	Estatística Pedro - Sala1	IA Rafael - Lab3
11:00	Probabilidade Pedro - Sala5	Engenharia de Software Ana - Lab1	Cálculo I Laura - Sala2	Estatística Pedro - Sala1	IA Rafael - Lab2
13:00	Probabilidade Pedro - Sala5	Engenharia de Software Ana - Lab1	Cálculo I Laura - Sala2	Estatística Pedro - Sala1	IA Rafael - Lab2
14:00	Sistemas Operacionais Carlos - Lab1	Engenharia de Software Ana - Lab1	Cálculo I Laura - Sala2	Programação II Paes - Lab2	Programação I Paes - Lab3
15:00	Sistemas Operacionais Carlos - Lab1	Engenharia de Software Ana - Lab1	Cálculo I Laura - Sala2	Programação II Paes - Lab2	Programação I Paes - Lab3
16:00	Direito Felipe - Sala1	Física II Rafaela - Sala5	Cálculo II Laura - Sala2	Programação II Paes - Lab1	Programação I Paes - Lab3
17:00	Direito Felipe - Sala1	Física II Rafaela - Sala3	Cálculo II Laura - Sala2	Programação II Paes - Lab1	Programação I Paes - Lab3
18:00	Física II Rafaela - Sala2	Física II Rafaela - Sala1	Álgebra Linear Paulo - Sala2	Álgebra Linear Paulo - Sala2	Programação I Paes - Lab3

(c) Difícil - Turma A

	Segunda	Terca	Quarta	Quinta	Sexta
07:00	Sistemas Operacionais Carlos - Lab1	Probabilidade Pedro - Sala4	Cálculo I Laura - Sala4	Estatística Pedro - Sala1	Programação II Paes - Lab2
08:00	Sistemas Operacionais Carlos - Lab1	Probabilidade Pedro - Sala4	Cálculo I Laura - Sala4	Estatística Pedro - Sala1	Programação II Paes - Lab2
09:00	Química Gustavo - Sala3	Probabilidade Pedro - Sala5	Cálculo I Laura - Sala4	Estatística Pedro - Sala1	Programação II Paes - Lab2
10:00	Química Gustavo - Sala3	Matemática Discreta Xu - Sala4	Cálculo I Laura - Sala3	Álgebra Linear Paulo - Sala4	Programação II Paes - Lab2
11:00	Ética Willy - Sala2	Matemática Discreta Xu - Sala4	Física II Rafaela - Sala3	Engenharia de Software Ana - Lab1	Redes Carlos - Lab1
13:00	Ética Willy - Sala2	Matemática Discreta Xu - Sala4	Física II Rafaela - Sala3	Engenharia de Software Ana - Lab1	Redes Carlos - Lab1
14:00	Direito Felipe - Sala1	Gestão de Projetos Felipe - Sala2	Física II Rafaela - Sala3	Engenharia de Software Ana - Lab1	Redes Carlos - Lab1
15:00	Direito Felipe - Sala1	Gestão de Projetos Felipe - Sala2	Física II Rafaela - Sala3	Engenharia de Software Ana - Lab1	Programação I Paes - Lab2
16:00	Banco de Dados Ana - Lab2	Empreendedorismo Willy - Sala1	IA Rafael - Lab3	Álgebra Linear Paulo - Sala3	Física I Rafaela - Sala3
17:00	Banco de Dados Ana - Lab3	Empreendedorismo Willy - Sala1	IA Rafael - Lab3	Álgebra Linear Paulo - Sala2	Física I Rafaela - Sala3
18:00	Banco de Dados Ana - Lab3	Álgebra Linear Paulo - Sala3	IA Rafael - Lab3	Cálculo II Laura - Sala1	Física I Rafaela - Sala3

(d) Difícil - Turma B

Figura 4: Grades horárias geradas para as instâncias média e difícil (Configuração B).

5 Conclusão

Este trabalho apresentou uma abordagem baseada no BRKGA para alocação de horários universitários, incorporando critérios pedagógicos como a continuidade das aulas por meio da

entropia temporal. A proposta mostrou-se eficaz tanto em termos de viabilidade quanto de qualidade didática das soluções.

A avaliação com instâncias de diferentes níveis de dificuldade evidenciou a robustez do algoritmo, e as visualizações reforçaram sua aplicabilidade prática. A Configuração B obteve os melhores resultados, ainda que com maior tempo de execução, sugerindo que o BRKGA pode ser ajustado conforme o contexto institucional.

Como trabalho futuro, destaca-se sua extensão para cenários mais complexos e integração com sistemas acadêmicos. Em suma, trata-se de uma solução flexível e eficiente para a automação da alocação de horários.

Referências

Arias-Osorio, J. & Mora-Esquivel, A. (2020), 'A solution to the university course timetabling problem using a hybrid method based on genetic algorithms', *Dyna* **87**, 47–56.

Jain, R. & Pydimarri, A. (2018), 'University time table scheduling using graph coloring technique', *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **3**(1), 245–249.

Zhang, Q. (2022), 'An optimized solution to the course scheduling problem in universities under an improved genetic algorithm', *Journal of Intelligent Systems* **31**(1), 1065–1073.

Zhang, Z., Ma, J. & Xiao, C. (2021), 'Genetic algorithm with three-dimensional population dominance strategy for university course timetabling problem', *International Journal of Grid and High Performance Computing* **13**, 56–69.

Histórico do Artigo:
Versão 1
Entregue: 29 de maio de 2025