57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

APLICAÇÃO DE METAHEURÍSTICAS AO PROBLEMA DA PROGRAMAÇÃO DE HORÁRIOS ESCOLARES

Thiago E. dos Santos[1]
thiaagosantos33@gmail.com

Heber F. Amaral[1]
heber.amaral@ifsudestemg.edu.br

Antônio R. Santana[1]
antonio.santana@ifsudestemg.edu.br
Instituto Federal do Sudeste De Minas Gerais, Campus Bom Sucesso
R. da Independência, 30 - Aparecida, Bom Sucesso - MG, 37220-000

RESUMO

O presente trabalho investiga a aplicação de técnicas metaheurísticas na resolução do Problema da Programação de Horários Escolares , um desafio clássico de otimização combinatória, classificado como NP-difícil. Considerando as múltiplas restrições inerentes ao problema, propõe-se o uso das abordagens Iterated Local Search (ILS), Variable Neighborhood Search (VNS) e Iterated Tabu Search (ITS), a partir de soluções iniciais geradas por uma Heurística Construtiva Aleatória com mecanismo de reparo. A formulação contempla restrições fortes e fracas, incorporadas a uma função objetivo com penalidades ponderadas. Os experimentos foram realizados com dados reais de quatro períodos letivos de uma instituição de ensino federal, e indicam que todas as soluções obtidas foram viáveis. Entre os métodos avaliados, o ITS apresentou os melhores resultados médios em termos de qualidade da solução. Os achados confirmam a efetividade das metaheurísticas propostas na geração de soluções consistentes e de baixo custo computacional.

PALAVRAS CHAVE. Problema da Programação de Horários Escolares, Iterated Local Search, Variable Neighborhood Search.

Tópicos: Metaheurísticas, Otimização Combinatória

ABSTRACT

Present study investigates the application of metaheuristic techniques to the School Timetabling Problem (STP), a classical combinatorial optimization challenge classified as NP-hard. Considering the multiple constraints inherent to the problem, this work proposes the use of Iterated Local Search (ILS), Variable Neighborhood Search (VNS), and Iterated Tabu Search (ITS), starting from initial solutions generated by a Randomized Constructive Heuristic with a repair mechanism. The formulation incorporates both hard and soft constraints into a weighted objective function with associated penalties. Experiments were conducted using real-world data from four academic terms at a federal educational institution, and all resulting solutions were feasible. Among the evaluated methods, ITS yielded the best average performance in terms of solution quality. The findings confirm the effectiveness of the proposed metaheuristics in producing consistent solutions with low computational cost.

KEYWORDS. School Timetabling Problem, Iterated Local Search, Variable Neighborhood Search.

Paper topics: Metaheuristics, Combinatorial Optimization

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

1. Introdução

O Problema da Programação de Horários Escolares (PPHE), inicialmente proposto por [Gotlieb, 1962], consiste na alocação de aulas em salas durante um determinado horizonte temporal, de forma que nenhuma sala receba mais de uma atividade simultaneamente e nenhum docente seja designado para ministrar duas aulas no mesmo período. A resolução manual desse problema revela-se uma tarefa complexa e demorada, podendo demandar vários dias em virtude do elevado número de restrições envolvidas e da combinatória associada, frequentemente resultando em soluções subótimas.

A automatização da elaboração de horários escolares proporciona ganhos expressivos em termos de eficiência e qualidade. Por meio da identificação de arranjos mais adequados entre os recursos disponíveis, é possível reduzir períodos ociosos e estabelecer uma rotina mais estruturada, beneficiando tanto docentes quanto discentes [Brito et al., 2012]. A aplicação de técnicas de otimização combinatória viabiliza a construção de horários que consideram restrições de diferentes naturezas, como preferências de professores, disponibilidade de ambientes e limites temporais. Intervalos excessivos entre aulas, por exemplo, tendem a comprometer a eficiência e a conveniência da rotina acadêmica. Assim, sua redução ou eliminação contribui para a continuidade das atividades, favorecendo a qualidade do processo de ensino-aprendizagem e a satisfação dos envolvidos, além de otimizar o uso dos recursos institucionais.

De acordo com [Even et al., 1975], o PPHE é classificado como um problema NP-difícil, o que implica a inexistência de algoritmos conhecidos que o resolvam em tempo polinomial. Diante dessa complexidade, métodos heurísticos e metaheurísticos têm sido amplamente empregados, fornecendo soluções de boa qualidade em tempo computacional viável.

As estratégias de resolução para problemas de otimização combinatória podem ser divididas em duas categorias principais: métodos exatos, que garantem a obtenção da solução ótima, e métodos heurísticos, que buscam soluções satisfatórias em prazos computacionais aceitáveis. Entre estes últimos, destaca-se a classe de métodos baseada em *busca local*, foco deste estudo. A *busca local* fundamenta-se no conceito de vizinhança: a partir de uma solução inicial viável, são realizadas modificações sucessivas que geram soluções vizinhas. Entre essas soluções, selecionam-se aquelas que promovem melhoria segundo um critério definido. Esse processo é repetido iterativamente até a convergência, momento em que nenhuma solução vizinha apresenta ganho adicional, configurando-se, assim, um ótimo local [Aarts et al., 1997].

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, destacando suas abordagens e resultados. A Seção 3 descreve três variações clássicas do problema e detalha as características específicas do problema do *campus* considerado. As Seções 4 e 5 discutem os métodos heurísticos adotados e seus resultados. Por fim, a Seção 6 apresenta as conclusões.

2. Trabalhos Relacionados

Diversos estudos têm investigado a aplicação de heurísticas e metaheurísticas na resolução do Problema da Programação de Horários Escolares (PPHE), contemplando distintos enfoques metodológicos e níveis de complexidade. Uma revisão sistemática prévia, apresentada em SANTOS et al. [2024], fornece um panorama abrangente sobre os avanços recentes na área. Com base nesse referencial, destacam-se a seguir alguns trabalhos relevantes que servem de fundamento para a presente pesquisa.

Em [Fonseca et al., 2012], as técnicas *Simulated Annealing* (SA) e *Iterated Local Search* (ILS) foram aplicadas ao *High School Timetabling Problem* (HSTP), considerando as instâncias da *International Timetabling Competition* (ITC). A solução inicial foi gerada por meio da plataforma KHE [Kingston, 2006], que auxilia na manipulação das instâncias e fornece soluções iniciais de boa

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

qualidade. Os resultados obtidos mostraram que, entre as instâncias analisadas, dez de dezessete soluções factíveis foram encontradas e, em sete de vinte e uma instâncias, o algoritmo superou as melhores soluções conhecidas até então.

O trabalho de [Moreira et al., 2016] aborda o problema de elaboração de horários no Departamento de Computação da CCA-UFES, utilizando a metaheurística *Greedy randomized adaptive search procedure (GRASP)* para a construção de soluções iniciais, seguida pela aplicação do SA para seu refinamento. O método proposto gerou soluções factíveis que superaram a alocação manual, com tempo de execução adequado às necessidades institucionais.

Em [Costa et al., 2017], o problema foi decomposto em duas etapas: alocação de salas e programação de horários para o IME/UERJ. Os autores propuseram um algoritmo construtivo para gerar a solução inicial, alocando cada evento em um dos horários possíveis e em salas compatíveis com os requisitos. Na etapa de melhoria, foram aplicadas as metaheurísticas SA clássico e *Late Acceptance Hill-Climbing* (LAHC). Este último avalia novas soluções vizinhas comparando-as não apenas com a solução corrente, mas também com soluções armazenadas em uma lista de histórico, permitindo escapar de ótimos locais.

Já [Song et al., 2018] investigam a aplicação do ILS com uma estratégia de perturbação denominada *maxmatch*, partindo de uma solução inicial gerada por um procedimento guloso que aloca eventos de forma sequencial, período a período. O trabalho utilizou um conjunto de 60 instâncias descritas em [Lewis et al., 2007], obtendo soluções viáveis em 58 delas. Os autores destacam que o ILS apresentou desempenho superior em instâncias de maior complexidade, superando algoritmos comparados no estudo.

O presente trabalho propõe uma abordagem voltada ao problema específico de nossa instituição, cujas restrições apresentam particularidades em relação às definidas na ITC e nos trabalhos mencionados. A resolução parte de uma solução inicial construída por meio de uma Heurística Construtiva Aleatória, posteriormente aprimorada por técnicas de busca local baseadas em ILS, VNS e ITS. Diferentemente dos estudos anteriores, a proposta enfatiza a adaptação às restrições institucionais específicas, o que confere maior aplicabilidade prática no contexto do campus considerado.

3. Descrição do Problema

O Problema da Programação de Horários Escolares apresenta diversas formulações, que variam de acordo com o nível de ensino e as restrições institucionais envolvidas. Nesta seção, são descritas as principais variações clássicas do problema reportadas na literatura e discutida a formulação específica enfrentada por nossa instituição. Além disso, são apresentadas as restrições fortes e fracas consideradas, a função objetivo utilizada, bem como a estrutura de dados e os movimentos de vizinhança empregados na exploração do espaço de busca.

O *High School Timetabling Problem* (HSTP) caracteriza-se por estruturas fixas de horário, nas quais os alunos permanecem em uma mesma sala, sendo o deslocamento realizado apenas pelos professores. Essa configuração reflete a realidade típica do ensino médio, no qual as cargas horárias das disciplinas são padronizadas entre diferentes turmas de uma mesma série [Schaerf, 1999].

O *University Course Timetabling Problem* (UCTP), por sua vez, apresenta maior flexibilidade em relação a alunos, horários e salas. Nesse caso, um aluno pode estar matriculado em diferentes turmas, uma vez que cada matrícula em disciplina de um período gera uma nova turma. Tanto alunos quanto professores se deslocam entre salas, e um dos objetivos dessa variação consiste em reduzir o deslocamento entre aulas consecutivas [Burke et al., 1997].

Já o *Post-Enrolment Course Timetabling Problem* (PE-CTT) é formulado após o processo de matrícula dos alunos. Seu foco está na minimização de conflitos individuais, bem como na redução do número de janelas e na limitação de aulas consecutivas. Apesar de ser considerado uma

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

variação distinta, o PE-CTT mantém restrições de natureza semelhante às encontradas nas demais formulações [Goh et al., 2017].

A formulação enfrentada em nossa instituição não se enquadra perfeitamente em nenhuma das variações mais consolidadas, mas compartilha elementos de diferentes abordagens. A estrutura de turmas segue o modelo do HSTP, uma vez que a grade é definida pelo próprio instituto. Contudo, as restrições relativas ao uso das salas aproximam o problema ao UCTP, pois uma mesma turma pode ter suas aulas distribuídas em diferentes ambientes, implicando o deslocamento dos alunos. Por outro lado, como a formulação do campus se aproxima mais do HSTP, os elementos do PE-CTT não foram considerados. A preocupação com conflitos envolvendo alunos surge apenas em casos específicos, como na oferta de disciplinas optativas ou interperíodos, quando é necessário criar turmas especiais com horários que não coincidam com as disciplinas regulares.

3.1. Restrições

Em problemas de otimização combinatória, a definição de restrições desempenha papel fundamental na avaliação da qualidade de uma solução. Essas restrições podem ser classificadas em duas categorias: **fortes**, cuja violação compromete a viabilidade da solução, e **fracas**, que, embora possam ser violadas, implicam penalizações associadas à qualidade final da solução. Assim, qualquer solução que desrespeite restrições fortes é considerada inviável, enquanto a violação de restrições fracas resulta em custos adicionais no valor da função objetivo.

As **restrições fortes** consideradas neste trabalho são:

HC1 Todas as aulas devem ser alocadas.

HC2 Duas disciplinas diferentes de uma mesma turma não podem ocorrer simultaneamente.

HC3 Um professor não pode lecionar mais de uma aula no mesmo intervalo de tempo.

HC4 Uma sala não pode receber mais de uma aula no mesmo intervalo de tempo.

HC5 Aulas pré-alocadas não podem ser movidas de sala ou horário.

HC6 As aulas de uma disciplina devem ser alocadas apenas em turnos compatíveis com sua oferta.

HC7 O total de aulas semanais alocadas de cada disciplina deve corresponder à sua carga horária estabelecida.

As **restrições fracas** são:

SC1 Minimizar o número de janelas entre aulas de uma mesma disciplina.

SC2 Respeitar os horários de preferência dos professores.

SC3 Concentrar a carga horária dos docentes em dias preferenciais.

SC4 Alocar disciplinas que exigem recursos específicos em salas compatíveis com tais necessidades.

3.2. Penalidades

As restrições fracas são associadas a custos de penalização, proporcionalmente à sua relevância. Entre elas, SC1 (minimização de janelas) recebe maior penalidade, dado seu impacto sobre a qualidade do horário. Já as restrições fortes possuem custo fixo de penalização igual a 10.000 por violação, garantindo que soluções inviáveis não sejam aceitas. A Tabela 1 apresenta as penalizações associadas a cada restrição fraca.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

Restrição	Penalidade
HC*	10.000
SC1	5
SC2	25
SC3	20
SC4	20

Tabela 1: Valor das penalidades para cada restrição

3.3. Função Objetivo

O objetivo do modelo consiste em minimizar a soma ponderada das violações de restrições fortes e fracas. A função objetivo é definida como:

Minimizar $f(s)$ =

$$\sum_{k=1}^{HC} HC_k(s)\alpha + \sum_{j=1}^{D}\sum_{k=1}^{H}\sum_{m=1}^{I}[wj(D_j, H_k, I_m)SC_1 + wd(D_j, H_k, I_m)SC_4]$$

$$+\sum_{l=1}^{P}\sum_{n=1}^{H}\sum_{p=1}^{I}[wp(P_l, H_n, I_p)SC_2 + wr(P_l, H_n, I_p)SC_3]$$

onde:

- S é o conjunto de salas;
- H é o conjunto de horários;
- D é o conjunto de disciplinas;
- I é o conjunto de aulas de todas as disciplinas;
- P é o conjunto de professores;
- HC é o conjunto das restrições fortes;
- $HC_k(s)$ indica o número de violações da k-ésima restrição forte;
- $wj(D_j, H_k, I_m)$ retorna 1 se houver uma janela entre aulas da disciplina D_j;
- $wd(D_j, H_k, I_m)$ retorna 1 se a aula de uma disciplina não for alocada em sala com recursos obrigatórios;
- $wp(P_l, H_n, I_p)$ retorna 1 se a aula de um professor for alocada em horário indisponível;
- $wr(P_l, H_n, I_p)$ retorna 1 se a aula de um professor ocorrer fora de seus dias preferenciais;
- α representa o peso fixo de penalização por violação de restrição forte.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

3.4. Estrutura de Dados e Movimentos de Vizinhança

A principal estrutura de dados utilizada pelo algoritmo é uma matriz bidimensional **Sala × Intervalo de Tempo**. Cada célula, correspondente a um par (s, h), pode armazenar um evento e_i, representando uma aula semanal de uma disciplina. Essa representação garante, de forma natural, o atendimento da restrição HC4, uma vez que impede a alocação de mais de um evento no mesmo *timeslot* de uma sala.

A exploração do espaço de busca é realizada por movimentos de vizinhança que modificam a solução corrente. Dois operadores principais foram empregados:

- ***Move***: uma aula a_1 é movida de um *timeslot* t_1 para um *timeslot* vazio t_2, escolhido aleatoriamente (Figura 1).
- ***Swap***: duas aulas, a_1 e a_2, são trocadas entre seus respectivos *timeslots*, ambos selecionados aleatoriamente (Figura 2).

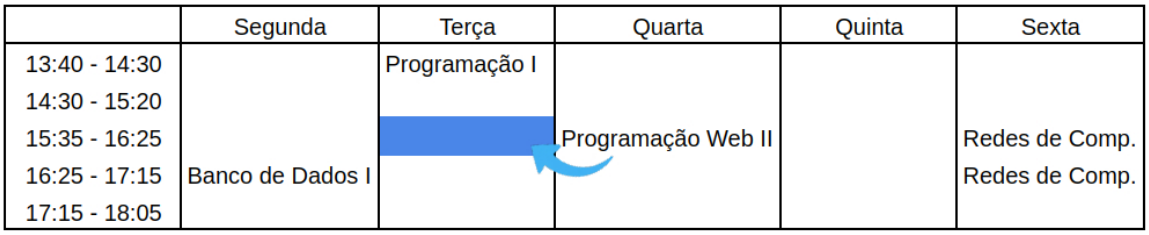


Figura 1: Exemplo de movimento de vizinhança do tipo *Move*

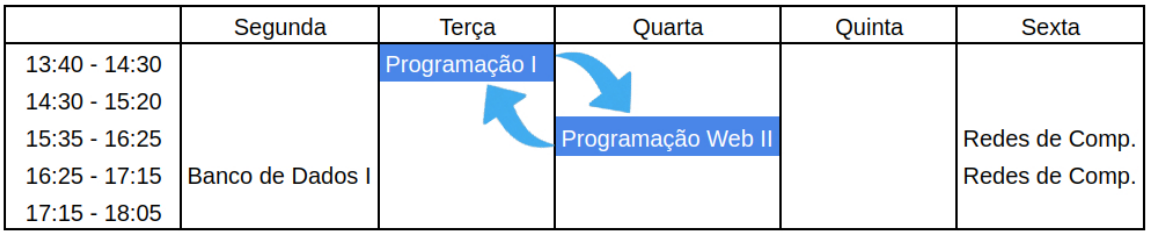


Figura 2: Exemplo de movimento de vizinhança do tipo *Swap*

A escolha do operador ocorre de forma probabilística: em 60% das iterações aplica-se o movimento *Move*, enquanto em 40% aplica-se o *Swap*. Essa proporção pode ser ajustada conforme as características da instância. Quando há muitos horários livres, o movimento *Move* tende a ser mais efetivo; em instâncias densas, com poucas vagas disponíveis, o *Swap* apresenta melhor desempenho.

4. Métodos Heurísticos

Os métodos heurísticos baseados em busca local têm como objetivo melhorar uma solução corrente por meio da exploração de soluções vizinhas. Contudo, a aplicação de metaheurísticas requer a disponibilidade de uma solução inicial viável. Para isso, propõe-se neste trabalho um algoritmo de geração de soluções iniciais adaptado às particularidades do problema em questão. A seguir, são descritos os métodos utilizados.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

4.1. Heurística Construtiva Aleatória com Reparo

Inspirada em [Lü e Hao, 2010], a Heurística Construtiva Aleatória (HCA), demonstrada no Algoritmo 1, percorre iterativamente um conjunto de eventos não alocados, atribuindo cada evento a uma sala j e a um horário k selecionados aleatoriamente, desde que a alocação preserve a viabilidade da solução parcial. Quando não existem *timeslots* disponíveis sem conflito, é acionada a etapa de reparo. Nesta etapa, um elemento conflitante — professor, sala ou horário — é selecionado e movido para outro *timeslot*. Caso o conflito persista, um evento aleatório é removido da grade e retorna à lista de aulas não alocadas, liberando espaço para a inserção do evento original.

Algoritmo 1: Heurística Construtiva Aleatória com Reparo

```
1  Entrada: Instância I
2  Saída: Solução viável s
3  LC ← ListaAulasNaoAlocadas()
4  while |LC| > 0 do
5      selecione aleatoriamente uma aula a_i;
6      selecione aleatoriamente um horário k e uma sala j;
7      if s é viável then
8          aloque a_i na matriz de horários;
9          remova a_i de LC;
10     else
11         reparar Solucao();
12 return s
```

4.2. *Iterated Local Search* (ILS)

O ILS estende a busca local simples ao introduzir perturbações periódicas que permitem escapar de ótimos locais. A partir de uma solução s melhorada por busca local, após um número definido de iterações sem melhora, aplica-se uma perturbação que gera uma nova solução s'. Em seguida, executa-se novamente a busca local, obtendo uma solução s''. Caso s'' apresente melhor valor da função objetivo em relação a s, a solução corrente é atualizada. Como pode ser observado no Algoritmo 2, esse mecanismo permite explorar regiões mais amplas do espaço de busca, ainda que temporariamente o custo da solução seja elevado [Lourenço et al., 2010].

Algoritmo 2: *Iterated Local Search*

```
1  Entrada: s_0, Time_max
2  Saída: best s
3  s ← BuscaLocal(s_0);
4  while Time < Time_max do
5      s' ← Perturbação(s);
6      s'' ← BuscaLocal(s');
7      if f(s'') ≤ f(s) then
8          Iter ← 0;
9          s ← s'';
10 return s;
```

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

4.3. *Variable Neighborhood Search* (VNS)

O VNS, proposto em [Mladenović e Hansen, 1997], como pode ser observado no Algoritmo 3, baseia-se na exploração sistemática de diferentes estruturas de vizinhança. O método alterna entre dois procedimentos principais: (*i*) o *shake*, que introduz perturbações controladas movendo a solução corrente para uma vizinhança distinta, e (*ii*) a busca local, aplicada sobre a solução perturbada. Esse processo é repetido até que não se obtenham mais melhorias ou seja atingido o número máximo de iterações, favorecendo a diversificação e evitando estagnação em ótimos locais.

Algoritmo 3: *Variable Neighborhood Search*

Entrada: $s_0, Time_{max}$
Saída: best s
$s \leftarrow BuscaLocal(s_0)$;
while $Time < Time_{max}$ **do**
    $s' \leftarrow Shake(s)$;
    $s'' \leftarrow BuscaLocal(s')$;
    if $f(s'') \leq f(s)$ **then**
        $Iter \leftarrow 0$;
        $s \leftarrow s''$;
return s;

4.4. *Iterated Tabu Search* (ITS)

A *Tabu Search* (TS), proposta por [Glover, 1986], é uma metaheurística de busca local guiada por memória. O algoritmo utiliza uma lista tabu para armazenar movimentos recentemente aplicados, de forma a impedir a repetição imediata desses movimentos e estimular a exploração de novas soluções. A lista tabu tem tamanho limitado, sendo atualizada de acordo com a estratégia FIFO (First-In First-Out).

Com o intuito de ampliar sua capacidade de exploração, [Misevicius et al., 2006] propuseram o *Iterated Tabu Search* (ITS), demonstrado no Algoritmo 4, que combina a etapa de perturbação do ILS com os mecanismos de memória da TS. Nesse esquema, a cada ciclo, uma solução s é perturbada, seguida da aplicação da TS, resultando em uma nova solução s''. Se s'' apresenta qualidade superior, ela substitui a solução corrente, analogamente ao processo do ILS. Essa integração permite maior diversificação do espaço de busca e aumenta as chances de aproximação do ótimo global.

Algoritmo 4: *Iterated Tabu Search*

Entrada: $s_0, Time_{max}$
Saída: best s
$s \leftarrow TabuSearch(s_0)$;
while $Time < Time_{max}$ **do**
    $s' \leftarrow Perturbação(s)$;
    $s'' \leftarrow TabuSearch(s')$;
    if $f(s'') \leq f(s)$ **then**
        $Iter \leftarrow 0$;
        $s \leftarrow s''$;
return s;

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

5. Experimentos Computacionais e Resultados

Os experimentos computacionais foram conduzidos em um computador equipado com processador AMD Ryzen 5 5600G, frequência de 3.9 GHz, 16 GB de memória RAM, utilizando a linguagem C++ compilada pelo G++ 11.4.0, sob o sistema operacional Ubuntu 22.04 LTS.

As instâncias utilizadas foram fornecidas pelo Campus Bom Sucesso, abrangendo os períodos de 2023/1 a 2024/2. Os dados contemplam informações referentes a professores, turmas, salas disponíveis, horários e preferências de alocação. As aulas em modalidade a distância (EAD) foram desconsideradas para a geração dos horários. A Tabela 2 apresenta as principais características das instâncias consideradas.

Período	Professores	Salas	Turmas	Disciplinas	Aulas
2024/2	16	9	9	53	131
2024/1	17	9	10	60	157
2023/2	15	9	11	61	154
2023/1	16	9	10	81	168

Tabela 2: Características das Instâncias dos períodos do *Campus* Bom Sucesso

5.1. Análise dos Resultados

Cada algoritmo foi executado 50 vezes por instância, garantindo robustez estatística dos resultados. Todas as soluções obtidas foram viáveis, isto é, não apresentaram violações de restrições fortes. No caso do ILS, foi empregada uma estratégia de perturbação adaptativa: inicialmente, 3% da solução corrente era perturbada; após 50 iterações sem melhora, a taxa aumentava para 5%; e após 100 iterações, para 10%. No ITS, a perturbação foi mantida fixa em 5%, enquanto o tamanho da lista tabu foi definido em 100 movimentos. O tempo limite estabelecido para cada execução foi de 300 segundos. Os resultados médios estão sumarizados na Tabela 3.

		ILS		VNS		ITS	
Período	f(HCA)	$f(s^*)$	$f(\bar{s})$	$f(s^*)$	$f(\bar{s})$	$f(s^*)$	$f(\bar{s})$
2024/2	4110	40	72,5	40	81,9	40	**42,7**
2024/1	4305	255	370,3	255	391,2	255	**346,5**
2023/2	4120	245	391,8	245	406,5	245	**252,1**
2023/1	3685	340	458,7	340	470,9	340	**351,1**

Tabela 3: Resultados obtidos

A HCA produziu soluções iniciais viáveis para todas as instâncias, mas com custos elevados. A aplicação subsequente das metaheurísticas ILS, VNS e ITS reduziu esses custos para valores entre 1% e 10% do custo inicial gerado pela HCA. As três abordagens apresentaram desempenho semelhante em termos de melhor solução encontrada; contudo, o ITS obteve custos médios consistentemente inferiores ao ILS, que por sua vez superou o VNS em todas as instâncias. A Tabela 4 detalha as violações de restrições fracas por instância, considerando a melhor solução encontrada em cada método.

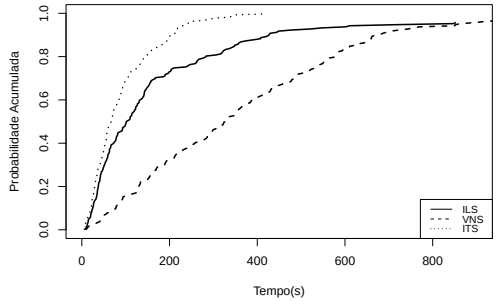
Com o intuito de avaliar a eficiência temporal dos algoritmos, aplicou-se a técnica TTT-plots [Aiex et al., 2007], que permite analisar a probabilidade de um método alcançar um determinado alvo dentro de um tempo limite. Foram realizadas 200 execuções para cada instância e algoritmo, com tempo limite de 900 segundos. O alvo foi definido como 10% acima do custo da melhor solução conhecida, exceto na instância 2024/2, na qual o alvo foi ajustado para 13% devido ao baixo custo absoluto da solução, o que aumentava a dificuldade da aproximação.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

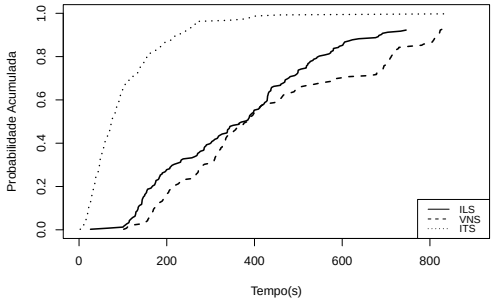
	ILS				VNS				ITS			
Período	SC1	SC2	SC3	SC4	SC1	SC2	SC3	SC4	SC1	SC2	SC3	SC4
2024/2	4	0	1	0	0	0	1	1	0	0	2	0
2024/1	3	0	10	2	3	0	11	1	6	1	9	1
2023/2	6	3	6	1	5	4	1	5	11	0	8	2
2023/1	6	2	8	5	2	2	4	10	14	2	8	3

Tabela 4: Penalizações na melhor solução de cada metaheurística

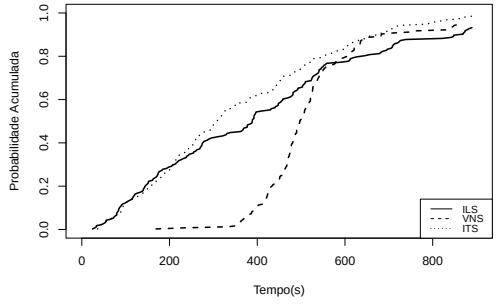
Os resultados mostram que, no período 2023/1, todas as execuções do ITS atingiram o alvo, enquanto oito execuções do ILS e seis do VNS não o alcançaram. No período 2023/2, o ITS também obteve 100% de sucesso, enquanto o ILS falhou em 15 execuções e o VNS em 11. Para o período 2024/1, o ILS não atingiu o alvo em 13 execuções, o VNS em 7, e o ITS em apenas 2. Por fim, no período 2024/2, o mais desafiador, o ILS falhou em 26 execuções, o VNS em 19, enquanto o ITS novamente apresentou sucesso em todas as execuções. Esses resultados confirmam a superioridade do ITS em termos de robustez e consistência na obtenção de soluções de alta qualidade.



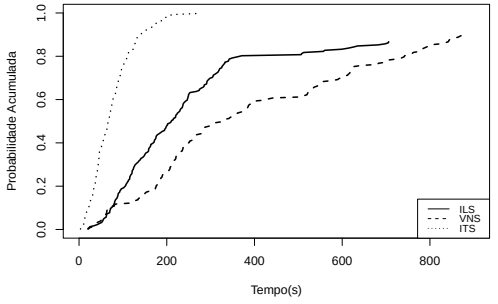
TTTplot 2023/1
target = 370



TTTplot 2023/2
target = 270



TTTplot 2024/1
target = 280



TTTplot 2024/2
target = 45

6. Conclusões

Os experimentos realizados demonstraram que as metaheurísticas avaliadas foram capazes de produzir resultados satisfatórios e consistentes, confirmando a aplicabilidade das abordagens

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

propostas para instâncias reais do *campus*. Em particular, o ILS apresentou desempenho superior ao VNS em termos de eficiência computacional, obtendo melhores soluções médias no limite de 300 segundos. Por outro lado, a análise dos *TTT-plots* revelou que o VNS apresentou maior capacidade de atingir soluções-alvo quando o tempo de execução foi ampliado para 900 segundos. O ITS, por sua vez, destacou-se como a técnica mais robusta, alcançando soluções de alta qualidade em todas as execuções e mantendo desempenho consistente ao longo das diferentes instâncias testadas.

De modo geral, os resultados confirmam a efetividade das metaheurísticas *Iterated Local Search* (ILS), *Variable Neighborhood Search* (VNS) e *Iterated Tabu Search* (ITS) na resolução do Problema da Programação de Horários Escolares (PHSE), considerando as restrições operacionais e estruturais de uma instituição de ensino real. Todas as abordagens foram capazes de gerar soluções totalmente viáveis, respeitando integralmente as restrições fortes do problema. O ITS apresentou o melhor desempenho médio, enquanto o ILS se destacou pela rapidez em tempos curtos de execução, reforçando sua utilidade em aplicações que demandam respostas rápidas.

Como perspectivas futuras, destaca-se a possibilidade de investigar estratégias híbridas que combinem diferentes metaheurísticas, explorando suas características complementares. Além disso, pretende-se aplicar os métodos a outras variantes do PHSE, como o *Post Enrolment Course Timetabling Problem* (PE-CTT), bem como adaptar os algoritmos para cenários de otimização multiobjetivo, contemplando aspectos adicionais como equidade na distribuição de horários e balanceamento de carga docente. Por fim, considera-se relevante o desenvolvimento de uma interface computacional acessível, que possibilite a utilização prática das soluções por gestores acadêmicos, ampliando o impacto e a aplicabilidade dos resultados obtidos.

Referências

Aarts, E., Korst, J. H., e van Laarhoven, P. J. (1997). *Simulated annealing. E. Aarts, JK Lenstra, eds., Local Search in Combinatorial Optimization*, volume 91120. John Wiley and Sons, New York, NY.

Aiex, R. M., Resende, M. G. C., e Ribeiro, C. C. (2007). Ttt plots: a perl program to create time-to-target plots. *Optimization Letters*.

Brito, S. S., Fonseca, G. H., Toffolo, T. A., Santos, H. G., e Souza, M. J. (2012). A sa-vns approach for the high school timetabling problem. *Electronic Notes in Discrete Mathematics*, 39:169–176.

Burke, E., Jackson, K., Kingston, J. H., e Weare, R. (1997). Automated university timetabling: The state of the art. *The computer journal*, 40(9):565–571.

Costa, D. H. d. S., Coelho, I. M., e Pinto, P. E. D. (2017). Programação de horários e alocação de salas de aula no ime/uerj com simulated annealing e lahc. p. 2077–2088.

Even, S., Itai, A., e Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *16th annual symposium on foundations of computer science (sfcs 1975)*, p. 184–193. IEEE.

Fonseca, G. H., Toffolo, T. A., Brito, S. S., e Santos, H. G. (2012). Técnicas de busca local para o problema da programaçao de horários escolares. *Anais do SBPO*, p. 1092–1103.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers Operations Research*, 13(5):533–549. Applications of Integer Programming.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085

Goh, S. L., Kendall, G., e Sabar, N. R. (2017). Improved local search approaches to solve the post enrolment course timetabling problem. *European Journal of Operational Research*.

Gotlieb, C. C. (1962). The construction of class-teacher time-tables. In *IFIP Congress*, p. 73–77. North-Holland.

Kingston, J. H. (2006). Hierarchical timetable construction. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling VI*, Berlin, Heidelberg. Springer-Verlag.

Lewis, R., Paechter, B., e Mccollum, B. (2007). Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. *Cardiff University, Cardiff Business School, Accounting and Finance Section, Cardiff Accounting and Finance Working Papers*.

Lourenço, H. R., Martin, O. C., e Stützle, T. (2010). *Iterated Local Search: Framework and Applications*, p. 363–397. Springer US, Boston, MA. ISBN 978-1-4419-1665-5.

Lü, Z. e Hao, J.-K. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200:235–244.

Misevicius, A., Lenkevicius, A., e Rubliauskas, D. (2006). Iterated tabu search: an improvement to standard tabu search. *Information Technology and Control*, 35(3).

Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers Operations Research*.

Moreira, L. V., Monteiro, R. C., Kampke, E. H., e Mauri, G. R. (2016). Meta-heurística grasp para o problema de tabela-horário de disciplinas do departamento de computação do cca-ufes. *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional*, p. 2171–2182.

SANTOS, T. E. D., VILELA, S. G., SANTANA, A. R., e AMARAL, H. F. (2024). Resolução do problema de programação de horários escolares usando busca local - uma revisão de literatura. In *SBPO 2024-Simpósio Brasileiro de Pesquisa Operacional*.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, 13:87–127.

Song, T., Liu, S., Tang, X., Peng, X., e Chen, M. (2018). An iterated local search algorithm for the university course timetabling problem. *Applied Soft Computing*.

DOI: 10.59254/sbpo-2025-212085
http://live-proceedings.pantheonsite.io/p/212085