

INSTITUTO FEDERAL DO SUDESTE DE MINAS GERAIS
CAMPUS BOM SUCESSO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

THIAGO EDUARDO DOS SANTOS

APLICAÇÃO DE METAHEURÍSTICAS AO PROBLEMA DA PROGRAMAÇÃO DE
HORÁRIOS ESCOLARES DO *CAMPUS* BOM SUCESSO DO IF SUDESTE MG

BOM SUCESSO
2025

THIAGO EDUARDO DOS SANTOS

APLICAÇÃO DE METAHEURÍSTICAS AO PROBLEMA DA PROGRAMAÇÃO DE
HORÁRIOS ESCOLARES DO *CAMPUS* BOM SUCESSO DO IF SUDESTE MG

Trabalho de conclusão de curso de graduação
apresentado ao IF SUDESTE MG *Campus*
Bom Sucesso como parte dos requisitos para
obtenção do grau de Tecnólogo em Análise e
Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Heber Fernandes Amaral

BOM SUCESSO

2025

Dados internacionais de catalogação na publicação (CIP)
Bibliotecária responsável Maria de Lourdes Cardoso CRB-6/3242

S237a Santos, Thiago Eduardo dos, 2004-

Aplicação de metaheurísticas ao problema da programação
de horários escolares do *campus* Bom Sucesso do IF
SUDESTE MG [recurso eletrônico] / Thiago Eduardo dos
Santos. -- 2025.

47 f. : il.

Orientador: Heber Fernandes Amaral

Trabalho de Conclusão de Curso (Tecnólogo em Análise e
Desenvolvimento de Sistemas) - Instituto Federal do Sudeste de
Minas Gerais, Campus Avançado Bom Sucesso, 2025.

1. Otimização combinatória. 2. Programação heurística. 3.
Escolas – Organização e administração. I. Amaral, Heber Fernandes.
II. Instituto Federal do Sudeste de Minas Gerais, Campus Avançado
Bom Sucesso. III. Título.

CDD: 004.21



TERMO DE APROVAÇÃO

Thiago Eduardo dos Santos

Aplicação de Metaheurísticas ao problema da programação de horários escolares do campus
Bom Sucesso do IF Sudeste mg

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais – *Campus Avançado Bom Sucesso*.

Bom Sucesso, 2 de julho de 2025 .

Assinaturas:

Membro da Banca 1

Membro da Banca 2

Membro da Banca 3

AGRADECIMENTOS

Agradeço primeiramente a Deus por me conceder forças e guiar meus caminhos ao longo dessa jornada.

Aos meus pais pelo amor incondicional, pelo apoio moral e por acreditarem em mim durante o período de formação.

Ao meu orientador Prof. Heber por todo o conhecimento científico compartilhado e suporte durante o curso.

Ao Instituto Federal do Sudeste MG - *Campus* Bom Sucesso, pela formação acadêmica de qualidade e pelo ambiente propício ao desenvolvimento científico e pessoal.

Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro por meio de bolsa de Iniciação Científica, que foi fundamental para o desenvolvimento das pesquisas que embasaram este trabalho.

*“Duas coisas enchem o meu espírito de admiração e respeito crescentes: o céu estrelado
sobre mim e a lei moral dentro de mim.”*

(Immanuel Kant)

RESUMO

O problema da programação de horários escolares é um desafio clássico de otimização combinatória, classificado como NP-Difícil e não possui um método que o resolva de forma exata em tempo polinomial. O problema é comumente abordado por métodos heurísticos e o presente trabalho propõe a sua resolução por meio da aplicação de técnicas metaheurísticas baseadas em busca local. As técnicas utilizadas foram o *Iterated Local Search* (ITS), *Variable Neighborhood Search* (VNS) e *Iterated Tabu Search* (ITS), além de uma Heurística Construtiva Aleatória (HCA) com mecanismo de Reparo para gerar soluções iniciais factíveis. A HCA demonstrou boa eficácia na geração de soluções iniciais, obtendo uma taxa de 100% de factibilidade em todas as instâncias. As metaheurísticas refinaram a solução inicial obtendo soluções com custo da função objetivo entre 1% e 10% da solução inicial. Entre as metaheurísticas avaliadas, o ITS apresentou melhores resultados em relação ao ILS, que foi superior ao VNS. A técnica *time-to-target* confirma a eficácia dos métodos utilizados, com grande parte das execuções alcançando o alvo definido.

Palavras-chave: otimização combinatória; *school timetabling problem*; *iterated local search*; *variable neighborhood search*; *iterated tabu Search*.

ABSTRACT

The school timetabling problem is a classic combinatorial optimization challenge, classified as NP-Hard and lacking an exact method that solves it in polynomial time. The problem is commonly addressed through heuristic methods, and the present work proposes its resolution through the application of metaheuristic techniques based on local search. The techniques used were Iterated Local Search (ILS), Variable Neighborhood Search (VNS), and Iterated Tabu Search (ITS), in addition to a Randomized Constructive Heuristic (RCH) with a Repair mechanism to generate feasible initial solutions. The RCH demonstrated good effectiveness in generating initial solutions, achieving a 100% feasibility rate across all instances. The metaheuristics refined the initial solution, obtaining solutions with objective function costs between 1% and 10% of the initial solution. Among the evaluated metaheuristics, ITS presented better results compared to ILS, which outperformed VNS. The time-to-target technique confirms the effectiveness of the methods used, with a large portion of the runs reaching the defined target.

Keywords: combinatorial optimizations; school timetabling problem; iterated local search; variable neighborhood search; iterated tabu search.

LISTA DE ILUSTRAÇÕES

Figura 1 – P, NP, NP- <i>hard</i> , e NP-completo	16
Figura 2 – Representação do espaço de busca	16
Figura 3 – Gráfico <i>time-to-target</i>	29
Figura 4 – Exemplo de movimento de vizinhança do tipo <i>Move</i>	33
Figura 5 – Exemplo de movimento de vizinhança do tipo <i>Swap</i>	33
Figura 6 – TTTplot 2023/1 - target = 370	37
Figura 7 – TTTplot 2023/2 - target = 270	38
Figura 8 – TTTplot 2024/1 - target = 280	38
Figura 9 – TTTplot 2024/2 - target = 45	39

LISTA DE TABELAS

Tabela 1	–	Número de trabalhos por <i>dataset</i>	22
Tabela 2	–	Valor das penalidades para cada restrição	31
Tabela 3	–	Características das Instâncias dos períodos do <i>Campus</i> Bom Sucesso	35
Tabela 4	–	Resultados obtidos	36
Tabela 5	–	Penalizações na melhor solução de cada metaheurística	36
Tabela 6	–	Alvos não alcançados	37

LISTA DE QUADROS

Quadro 1 – Características de todos os trabalhos dos anos de 2014 a 2024	20
Quadro 2 – Restrições por Trabalhos sobre HSTP	23
Quadro 3 – Restrições por Trabalhos sobre UCTP	24

LISTA DE ABREVIATURAS E SIGLAS

ILS.	<i>Iterated Local Search</i>
VNS.	<i>Variable Neighborhood Search</i>
SA.	<i>Simulated Annealing</i>
TS.	<i>Tabu Search</i>
ITS.	<i>Iterated Tabu Search</i>
STP.	<i>School Timetabling Problem</i>
HSTP.	<i>High School Timetabling Problem</i>
UCTP.	<i>University Course Timetabling Problem</i>
PE-CTT.	<i>Post Enrolment Timetabling Problem</i>
FO.	Função Objetivo

LISTA DE SÍMBOLOS

β	Letra grega Beta
Σ	Letra grega Sigma
α	Letra grega Alpha
Δ	Letra grega Delta
\emptyset	Conjunto vazio

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVO GERAL	14
1.2	OBJETIVOS ESPECÍFICOS	14
2	REFERENCIAL TEÓRICO	15
2.1	PROBLEMAS DA CLASSE NP	15
2.2	ESPAÇO DE SOLUÇÕES	15
2.3	PROBLEMA DA PROGRAMAÇÃO DE HORÁRIOS ESCOLARES .	16
2.4	REVISÃO DE LITERATURA	18
2.4.1	Crítérios de Inclusão	19
2.5	MÉTODOS HEURÍSTICOS	24
2.5.1	Busca Local	25
2.5.2	<i>Iterated Local Search</i>	25
2.5.3	<i>Simulated Annealing</i>	26
2.5.4	<i>Variable Neighborhood Search</i>	26
2.5.5	<i>Tabu Search</i>	26
2.5.6	<i>Iterated Tabu Search</i>	28
2.6	<i>TIME-TO-TARGET</i> (TTT PLOTS)	29
3	METODOLOGIA	30
3.1	DESCRIÇÃO DO PROBLEMA	30
3.1.1	Restrições	30
3.1.2	Penalidades	31
3.1.3	Função Objetivo	31
3.1.4	Estrutura de Dados e Vizinhança	33
3.2	GERAÇÃO DA SOLUÇÃO INICIAL E APLICAÇÃO DE METAHEURÍSTICAS	34
4	RESULTADOS E EXPERIMENTOS COMPUTACIONAIS .	35
4.1	ANÁLISE DOS RESULTADOS	35
5	CONCLUSÃO E TRABALHOS FUTUROS	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

O problema da programação de horários escolares, descrito como *School Timetabling Problem* (STP) é enfrentado por diferentes instituições de ensino, sendo um desafio complexo que envolve a combinação de vários elementos como disciplinas, professores e salas de aula, além da alocação de aulas em períodos de tempo. A solução manual deste problema costuma ser demorada e geralmente resulta em soluções subótimas, podendo se tornar inviável com o crescimento dos dados institucionais (Souza; Costa; Guimarães, 2002).

Soluções subótimas, ainda que não gerem conflitos de alocação, podem resultar em combinações ruins contendo alto número de janelas na grade dos professores e alunos ou concentrar um grande número de aulas no mesmo dia. Dessa forma, minimizar as janelas e balancear a carga horária, pode resultar em uma rotina mais produtiva e melhorar a qualidade do ensino da instituição. Portanto, a automação da programação de horários resulta em ganhos significativos em relação ao custo/tempo na geração de novas soluções, permitindo encontrar as melhores combinações de recursos, resultando em grades de horários bem estruturadas e com baixo custo (Brito *et al.*, 2012).

O STP é um problema clássico de otimização combinatória classificado como NP-Difícil, ou seja, não existe um algoritmo eficiente conhecido para resolver o problema de forma exata em tempo polinomial, devido ao crescimento exagerado do campo de busca a medida que uma instância cresce. Assim, o problema é frequentemente tratado através de métodos heurísticos (Souza; Costa; Guimarães, 2002).

Existem diversas variações do STP na literatura e as principais são: *High School Timetabling Problem* (HSTP), *University Course Timetabling Problem* (UCTP) e *Post-Enrolment Course Timetabling Problem* (PE-CTT). O HSTP, enfrentado principalmente por escolas com ensino médio, possui uma estrutura curricular mais rígida, em que os alunos seguem uma grade fixa e não estão diretamente envolvidos. O UCTP possui a estrutura mais flexível, com o aluno montando sua própria grade de horários baseada nas suas necessidades, sendo muito enfrentado pelas universidades. Já o PE-CTT também é enfrentado pelas universidades, porém as necessidades de alunos são consideradas para gerar as novas soluções, com a montagem da grade de horários sendo feita após as matrículas (Goh; Kendall; Sabar, 2017).

Métodos Heurísticos baseados em busca local são frequentemente utilizados para resolver o STP das instituições, gerando boas soluções em tempo hábil e com baixo custo computacional. Ao explorar vizinhanças bem definidas, a busca local consegue facilmente melhorar soluções de forma incremental, reduzindo conflitos de alocação e diminuindo o custo final da solução (Aarts; Korst; Laarhoven, 1997).

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho envolve um levantamento geral dos trabalhos que propõem soluções às diversas instituições de ensino e a aplicação de métodos metaheurísticos baseado em busca local para propor novas soluções o problema da programação de horários escolares específico enfrentado pelo *Campus* Bom Sucesso.

1.2 OBJETIVOS ESPECÍFICOS

- a) Categorizar os diversos trabalhos presentes na literatura e seus métodos de resolução;
- b) Desenvolvimento de um método construtivo para gerar soluções iniciais viáveis;
- c) Aplicação de métodos metaheurísticos presentes na literatura;
- d) Comparação destes métodos com dados reais do *Campus*.

2 REFERENCIAL TEÓRICO

Este capítulo tem por objetivo apresentar os conceitos básicos do trabalho e suas fundamentações. A seção 2.1 contextualiza a classe de problemas NP e sua complexidade. A seção 2.2 detalha o espaço de solução do problema em questão. A seção 2.3 define o problema da programação de horários e suas variações. Na seção 2.4, é feita uma revisão de literatura abrangendo os trabalhos dos últimos dez anos. A seção 2.5 aborda os métodos heurísticos mais utilizados para resolver o problema em questão. Por fim, a seção 2.6 detalha o método *time-to-target* que avaliará as metaheurísticas posteriormente.

Parte dos conteúdos desse capítulo foram previamente divulgados no artigo "Resolução do Problema de Programação de Horários Escolares Usando Busca Local - Uma Revisão de Literatura", publicado nos anais do Simpósio Brasileiro de Pesquisa Operacional (SBPO) em 2024, de autoria própria.

2.1 PROBLEMAS DA CLASSE NP

No campo da otimização combinatória, os problemas podem ser classificados com base em sua complexidade computacional, pertencendo a classes como P, NP, NP-completo e NP-difícil.

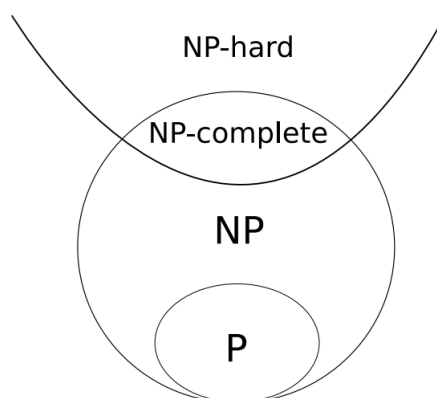
Segundo Cook (1971), o problema da satisfatibilidade booleana (SAT) foi o primeiro a ser provado como NP-completo. Isso significa que ele pertence à classe NP, possuindo a mesma complexidade de qualquer outro problema da classe. Foi demonstrado que qualquer problema em NP pode ser reduzido ao SAT, dessa forma, caso seja encontrado um algoritmo eficiente para resolver o SAT em tempo polinomial, todos os problemas de NP também poderiam ser resolvidos da mesma forma. Isso provaria que $P = NP$, uma das grandes questões em aberto da ciência da computação.

Além disso, Cook (1971) ainda define que um problema é considerado NP-difícil se todo problema da classe NP pode ser reduzido a ele em tempo polinomial. No entanto, muitos problemas de otimização, como o *Traveling Salesman Problem*, são NP-difíceis, pois encontrar a solução ótima, ou mesmo verificar, pode ser computacionalmente inviável, o que ocorre com o *School Timetabling Problem*.

2.2 ESPAÇO DE SOLUÇÕES

O espaço de busca de um problema é composto por um conjunto contendo todas as soluções possíveis para tal. As soluções existentes, sendo viáveis ou não, são chamadas de estado ou candidatos. Testar todos os estados é uma tarefa extremamente complexa e custosa em problemas da classe NP, sendo comumente abordado por algoritmo específicos

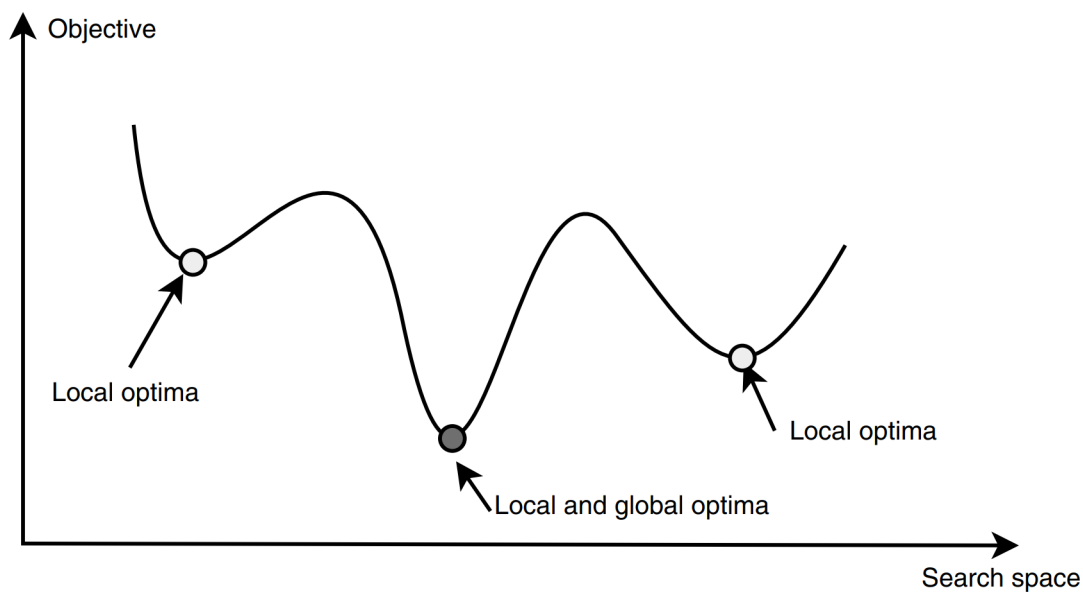
Figura 1 – P, NP, NP-hard, e NP-completo



Fonte: Aaronson (2016)

de exploração. A melhor solução possível é chamada de ótimo global, porém devido ao grande espaço de busca, os algoritmos tendem a ficar presos em ótimos locais, que são candidatos melhores que seus vizinhos, mas não necessariamente são aceitáveis (Talbi, 2009).

Figura 2 – Representação do espaço de busca



Fonte: Talbi (2009)

2.3 PROBLEMA DA PROGRAMAÇÃO DE HORÁRIOS ESCOLARES

O problema da programação de horários escolares, ou *School Timetabling Problem* (STP) é um desafio clássico presente em diversas instituições de ensino, em que as aulas devem ser alocadas à recursos como salas e professores, semelhante ao problema do escalo-

namento (*scheduling*), em que a alocação deve respeitar as restrições principais e otimizar o cenário. O principal objetivo do STP é criar um cronograma que atenda a diversas restrições, incluindo a disponibilidade de recursos, preferências de horário e capacidade das salas (Bashab *et al.*, 2022).

O STP possui diversas variações em sua formulação e restrições, de acordo com o cenário em que será aplicado, podendo mudar a forma como o problema será tratado e qual método será utilizado. As variações mais comuns na bibliografia são:

O High School Timetabling Problem (HSTP) é uma variação que trabalha com turmas e horários fixos, sem admitir janelas entre os horários. As turmas permanecem na mesma sala, com apenas os professores se deslocando (Schaerf, 1999).

O University Timetabling Problem (UCTP) é mais flexível quanto aos horários, salas e professores, permitindo que os alunos pertençam a múltiplas turmas e tolerando janelas entre os horários (Babaei; Karimpour; Hadidi, 2015).

O Post Enrolment Course Timetabling (PE-CTT) envolve a alocação de cursos em intervalos de tempo e salas, garantindo que os estudantes possam frequentar todos os cursos em que se inscreveram, sem sobreposições (Ceschia; Di Gaspero; Schaerf, 2012).

A montagem da grade de horários envolve restrições que devem ser respeitadas e garantem que a solução será factível. Sendo assim, Santos e Souza (2007) apresentam dois tipos:

- **Restrições Graves:** São restrições obrigatórias que determinam se a solução será viável. Não respeitar essas restrições, leva a solução a gerar conflitos de alocação.
- **Restrições Leves:** São restrições que não necessariamente devem ser satisfeitas, visto que elas não determinam a viabilidade de uma solução, porém são desejáveis e podem impactar significativamente no custo final.

Frangouli, Harmandas e Stamatopoulos (1995) determinam que é possível calcular o espaço de soluções aproximado do STP utilizando a seguinte equação:

Sendo E o tamanho total do espaço de busca, e_d o número de disciplinas, e_a o número de aulas, e_s o número de salas, b_{ma} o número médio de aulas e b_c o número de cursos.

$$E = (e_d, e_a, e_s)^{b_{ma} \times b_c} \quad (1)$$

Tomando como exemplo um cenário fictício que possua 20 cursos com 5 aulas semanais em média, 5 dias úteis por semana, 9 horários por dia e 12 salas de aula. Neste caso, o espaço de busca seria:

$$(5 \times 9 \times 5)^{12 \times 20} \approx 3.02 \times 10^{564} \quad (2)$$

2.4 REVISÃO DE LITERATURA

O STP, que envolve a alocação de aulas em horários tem se expandido significativamente nas últimas décadas devido à crescente demanda das instituições de ensino pelo mundo, abrangendo desde ensino básico até ensino superior, em que cada instituição possui suas particularidades e desafios. Dentre os trabalhos relevantes publicados recentemente que abordam o problema em questão, destacam-se 4 Revisões Sistemáticas de Literatura (RSL) importantes:

- Babaei, Karimpour e Hadidi (2015) cobrem metodologias aplicadas no contexto universitário, apresentando uma visão sobre o problema e discutindo potencial de futuras melhorias. As metodologias incluem algoritmos genéticos, programação linear, *tabu search*, *simulated annealing*, *GRASP*, *VND*, destacando métodos híbridos como forma de resolução;
- Tan *et al.* (2021) apresentam uma revisão abrangente dos métodos de otimização aplicados, comparando desempenho e destacando perspectivas industriais. Os métodos incluíram otimização matemática, metaheurísticas e modelos híbridos, confirmando o aumento da popularidade de modelos de otimização matemática;
- O estudo de Furtado (2022) analisa o STP no contexto brasileiro, identificando as principais metodologias aplicadas e cobrindo os estudos presentes na literatura entre 2017 e 2022. O trabalho avalia as aplicações de algoritmos genéticos, *tabu search*, heurísticas construtivas e *simulated annealing*, focando nas particularidades das instituições de ensino do Brasil.

Os trabalhos relatados exploram a aplicação de diversas técnicas para solucionar o problema, enquanto a RSL em questão é voltada às abordagens baseadas em busca local e suas variantes, bem como as diversas formulações populares e *benchmarks* utilizando no período entre 2014 e 2024, tratando as variações e formulações do problema de forma específica.

Para a revisão sistemática de literatura, as seguintes questões foram formadas:

- a) Q-1: Quais técnicas são utilizadas para resolver o problema?
- b) Q-2: Quais as características dos problemas resolvidos?
- c) Q-3: Quais as características das instâncias usadas?
- d) Q-4: Quais os *benchmarks* usados?

O critério de seleção de fontes foi realizado de maneira criteriosa, visando incluir apenas os estudos relevantes e publicações de qualidade. Foram utilizados o *Google Scholar* como

ferramenta principal na busca de trabalhos, busca manual detalhada e rigorosa e técnicas de busca automática, utilizando o *Selenium* como ferramenta auxiliar.

2.4.1 Critérios de Inclusão

Os critérios de inclusão foram:

- a) Trabalhos relacionados aos problemas de planejamento de horários em escolas e universidades;
- b) Trabalhos que relatem o uso de busca local para solução desses problemas;
- c) Trabalhos publicados em revistas científicas indexadas com avaliação *Qualis* A1, A2, A3 ou A4;
- d) Trabalhos publicados na SBPO;
- e) Trabalhos completos;
- f) Trabalhos de acesso aberto ou disponíveis através dos periódicos *Capex*;
- g) Trabalhos publicados entre 2014 e 2024.

A *string* de busca foi formulada para para abranger o máximo possível de trabalhos e formulações distintas presentes na literatura. Abaixo estão os principais componentes utilizados:

Os **Termos Principais**: "alocação de horários", "*timetabling problem*", "busca local", "*local search*", "*VNS*", "*Iterated local search*", "*ILS*", "*Tabu Search*".

Para alcançar novos trabalhos relacionados aos termos, outras variações foram buscadas, resultando na seguinte *string* de busca:

("*high school timetabling problem*" OR "*university timetabling problem*") AND
("*local search*" OR "*VNS*" OR "*Iterated local search*" OR "*ILS*" OR "*Tabu Search*").

Ao realizar a pesquisa no *Google Scholar* para o período de 2014 a 2024, com a palavra-chave "*high school timetabling problem*", foram encontrados 412 resultados. Após adicionar o termo "*local search*" à pesquisa, o número de resultados caiu para 286, constatando que quase 70% dos métodos utilizados partem da busca local. Por outro lado, a busca pelo termo "*university course timetabling problem*" retornou 840 trabalhos, sendo 460 resultados retornados ao aplicar "*local search*" na pesquisa, correspondendo a mais de 50% dos trabalhos encontrados. Apesar da abundância de literatura, ao aplicarmos os filtros de Revisão Sistemática da Literatura (RSL), como qualidade de publicação e relevância

Quadro 1 – Características de todos os trabalhos dos anos de 2014 a 2024

Trabalho	Tipo de abordagem	Formulação
(Abuhamdah <i>et al.</i> , 2014)	Population based Local Search	UCTP
(Cassemiro; Miranda; Wanner, 2014)	Tabu Search and Genetic Algorithm	HSTP
(Fonseca; Santos, 2014)	Variable Neighborhood Search	HSTP
(Saviniec <i>et al.</i> , 2015)	Iterated Local Search	HSTP
(Fonseca <i>et al.</i> , 2016)	Hybrid Local Search	HSTP
(Freire; Melo, 2016)	Local Search	UCTP
(Jardim; Semaan; Penna, 2016)	Iterated Local Search	UCTP
(Kheiri; Özcan; Parkes, 2016)	Stochastic Local Search	HSTP
(Soria-Alcaraz <i>et al.</i> , 2016)	ILS using an add and delete hyper-heuristic	UCTP
(Demirović; Musliu, 2017)	MaxSAT-based Large Neighborhood Search	HSTP
(Goh; Kendall; Sabar, 2017)	Improved local search	PE-CTT
(Saviniec; Constantino, 2017)	Local Search	HSTP
(Nagata, 2018)	Random Partial Neighborhood Search	PE-CTT
(Song <i>et al.</i> , 2018)	Iterated Local Search	UCTP
(Goh <i>et al.</i> , 2020)	Hybrid Local Search	PE-CTT
(Song <i>et al.</i> , 2021)	Competition-guided Multi-neighborhood Local Search	UCTP
(Rezaeipana; Matoori; Ahmadi, 2021)	Improved Parallel Genetic Algorithm and LS	UCTP
(Awad; Al-kubaisi; Mahmood, 2022)	Adaptive Tabu Search	PE-CTT

Fonte: Elaborado pelo Autor (2025)

ao tema, apenas 19 artigos atenderam às especificações estabelecidas. Os trabalhos que atendem aos critérios da RSL estão listados no quadro 1.

O quadro 1 detalha a formulação do STP e o método utilizado por cada trabalho publicado entre 2014 e 2024, em que cada método parte de alguma variação da busca local. O número de trabalhos selecionados foi significativamente menor que o retornado na pesquisa, do Google Scholar devido à maioria dos trabalhos não atenderem aos critérios específicos de qualidade. Além disso, muitos trabalhos apenas citavam a busca local em seu conteúdo ou como palavra-chave, sem necessariamente implementá-la ou propor novas abordagens.

Problemas como o STP geralmente são mais desafiadores do que os problemas mais clássicos de otimização combinatória, como o *Traveling Salesman Problem* (TSP), *Max-Cut*, *Max-SAT*, *Knapsack Problem* (KP). Isso se explica devido à complexidade de restrições que envolvem o STP, além do amplo espaço de busca que o problema pode gerar, mesmo com poucos recursos e eventos. Enquanto o TSP envolve apenas uma restrição grave (Passar por todos os pontos sem repeti-los e voltar ao ponto inicial), o STP pode conter entre 3 (no mínimo) e 8 restrições, podendo facilmente ultrapassar este número. Além disso, o STP ainda deve satisfazer ao máximo as restrições fracas, pois são elas que ditam a qualidade da solução e impactam diretamente no custo da função objetivo. Portanto, as técnicas de busca local simples podem não gerar soluções satisfatórias, podendo depender de técnicas mais sofisticadas, como os métodos híbridos que envolvem o uso de metaheurísticas combinadas com Programação Inteira.

Considerando a abordagem de cada trabalho e os resultados buscado inicialmente, a

busca local é a técnica mais comum e aplicável ao problema, contendo inúmeros métodos que utilizam seus mecanismos. Alguns destes métodos são clássicos de otimização combinatorial, como o *Iterated Local Search*, *Variable Neighborhood Search* e outras heurísticas híbridas.

Metaheurísticas como *Adaptive Tabu Search*, utilizada por Awad, Al-kubaisi e Mahmood (2022), utilizam mecanismo de memória com adaptação com o objetivo de escapar de ciclos e evitar movimentos repetidos. A *Variable Neighborhood Search* (VNS), como aplicada em Fonseca e Santos (2014), modifica a solução alternando entre diferentes vizinhanças com busca local aplicada e cada novo vizinho modificado. Abordagens como *Random Partial Neighborhood Search*, vista em Nagata (2018), explora sub-regiões do espaço de busca de forma mais agressiva, de forma a adiar a convergência da solução e escapar dos ótimos locais. A *MaxSAT-Based Large Neighborhood Search*, utilizada por Demirović e Musliu (2017), é baseada na formulação híbrida de *MaxSAT* combinada com a busca em grande vizinhança para melhorar a eficiência da do método. A abordagem estocástica, como em Kheiri, Özcan e Parkes (2016), parte de escolhas aleatórias na busca local para diversificar a busca e obter melhor exploração no espaço de soluções, enquanto a *Population-Based Local Search*, definida em Abuhamdah *et al.* (2014), seleciona uma população de soluções e aplica busca local em cada solução gerada. Além disso, a técnica híbrida de *Backtracking* mais busca local, Cassemiro, Miranda e Wanner (2014), alavanca as vantagens de ambas as técnicas para melhorar a busca de soluções e obter o melhor de cada metaheurística.

Contudo, dentre os trabalhos listados, grande parte que aborda o STP no contexto de escolas do ensino médio provém do (*High School Timetabling Problem* - HSTP), sendo predominante devido à demanda de escolas serem maiores. Dentre os trabalhos selecionados, Saviniec e Constantino (2017), Fonseca *et al.* (2016), Demirović e Musliu (2017), Kheiri, Özcan e Parkes (2016), Fonseca e Santos (2014), Saviniec *et al.* (2015), e Cassemiro, Miranda e Wanner (2014) destacam essa formulação. O (*University Course Timetabling Problem* - UCTP) se encaixa melhor com as regras das universidades, e ainda que seja menos popular que o HSTP, também é um problema real no contexto das universidades de todo o mundo. Esta variação é trabalhada e tratada em Song *et al.* (2018), Soria-Alcaraz *et al.* (2016), Abuhamdah *et al.* (2014), Jardim, Semaan e Penna (2016), e Freire e Melo (2016). Em alguns casos, as universidades realizam sua programação de horários após os alunos serem matriculados, configurando o *Post-Enrolment Course Timetabling Problem* (PE-CTT), estando presente em trabalhos como Goh *et al.* (2020), Awad, Al-kubaisi e Mahmood (2022), Nagata (2018) e Goh, Kendall e Sabar (2017).

A *International Timetabling Competition* (ITC) é uma competição internacional realizada por diversas universidades do mundo que tem como principal foco, a disponibilização de *datasets* públicos abertos à comunidade acadêmica para que as soluções sejam submetidas e ranqueadas. A ITC disponibiliza diversos conjunto de dados de diferentes

universidades do mundo, com diferentes cenários, eventos e recursos. A edição de 2002 considerou o PE-CTT como formulação principal para todas as suas instâncias. No ano de 2007 houveram 3 trilhas diferentes, sendo: PE-CTT, *Exam Timetabling* e Curriculum Based Course Timetabling (CB-CTT). O *Exam Timetabling* é uma formulação mais simples, que foca apenas no agendamento de exames e provas, diferente do CB-CTT que é bem próximo do UCTP, com mínimas diferenças na formulação. A edição de 2011 é totalmente focada no HSTP fornecendo 35 instâncias provenientes de escolas de 10 diferentes países.

O uso de datasets públicos como os da ITC é de grande importância para as avaliações de novos algoritmos e validações de pesquisas no campo da otimização combinatória. Além disso, a ITC disponibiliza sua plataforma de *benchmark*, em que uma solução é submetida e avaliada conforme as regras da competição.

A Tabela 1 descreve os *datasets* utilizados dentre os trabalhos selecionados. De acordo com a tabela, 2 trabalhos utilizaram o *dataset* da ITC 2002, 5 utilizaram os dados da ITC 2007, 4 aplicaram o ITC 2011, e os *datasets* próprios foram utilizados em 10 trabalhos. Alguns trabalhos aplicaram os métodos tanto em suas instâncias próprias quanto nas instâncias da ITC, servindo como meio de comparação com a literatura atual.

Tabela 1 – Número de trabalhos por *dataset*

Base de dados	Número de trabalhos
ITC 2002	2
ITC 2007	5
ITC 2011	4
Dataset próprio	10

Fonte: Elaborado pelo Autor (2025)

É possível observar grande parte dos trabalhos utilizam seus próprios conjuntos de dados, geralmente fornecidos pela própria instituição, reforçando que o problema é comum em várias instituições de ensino do mundo e geralmente era tratado de forma manual.

Abaixo as restrições fortes presentes nos trabalhos que abordam HSTP:

- R1** Nenhum professor pode estar em dois lugares ao mesmo tempo: Um professor não pode ser alocado para mais de uma turma ou sala em um mesmo horário.
- R2** Nenhuma sala de aula pode ser utilizada por mais de uma turma simultaneamente: Cada sala deve ser usada apenas por uma turma por vez.
- R3** Disponibilidade de professores: Os professores só podem ser alocados em horários nos quais estão disponíveis.
- R4** Alocação de disciplinas obrigatórias: As aulas obrigatórias devem ser alocadas dentro do horário escolar, garantindo que todas as disciplinas previstas sejam ministradas.

- R5** Limites de capacidade de salas: As turmas alocadas a uma sala devem respeitar o limite de capacidade física do espaço.
- R6** Limites de aulas diárias: Todas as aulas de uma mesma disciplina não podem ocorrer no mesmo dia

Quadro 2 – Restrições por Trabalhos sobre HSTP

Trabalhos	R1	R2	R3	R4	R5	R6
(Casseiro; Miranda; Wanner, 2014)	X	X	X	X	X	
(Fonseca; Santos, 2014)	X	X	X	X	X	
(Saviniec <i>et al.</i> , 2015)	X	X	X	X	X	X
(Fonseca <i>et al.</i> , 2016)	X	X	X	X	X	X
(Kheiri; Özcan; Parkes, 2016)	X	X	X	X	X	
(Demirović; Musliu, 2017) ¹	X	X	X	X	X	
(Saviniec; Constantino, 2017)	X	X	X	X	X	

Fonte: Elaborado pelo Autor (2025)

Nos trabalhos que abordam o UCTP, as restrições graves são muito semelhantes entre eles, com uma média de três restrições por estudo, de acordo com o quadro 3. Contudo, as restrições leves são extremamente variáveis ao comparar as abordagens selecionadas, alterando o custo da função objetivo e a qualidade da solução. O quadro ainda mostra que alguns trabalhos possuem variações na restrições graves, aumentando o grau de complexidade e reduzindo o espaço de busca, o que dificulta encontrar soluções de qualidade.

Abaixo as restrições fortes presente nos trabalhos que abordam UCTP:

- R1** Período: não podem existir duas ou mais disciplinas de um mesmo período no mesmo horário.
- R2** Espaço Físico: o total de atividades de um horário e seus respectivos espaços físicos necessários não podem ultrapassar a quantidade especificada.
- R3** Disciplina: não podem existir duas ou mais alocações de um determinado professor em um mesmo horário.
- R4** Capacidade: O número de alunos matriculados no curso deve ser menor ou igual à capacidade da sala.
- R5** Os eventos devem ser atribuídos apenas aos horários predefinidos como disponíveis.
- R6** Quando especificado, os eventos devem ser agendados para ocorrer na ordem correta.
- R7** Determinam que o número de aulas alocadas de cada turma é menor ou igual ao seu número de aulas práticas.
- R8** Para cada evento, deve-se alocar um professor dentre seu conjunto de professores compatíveis.

Quadro 3 – Restrições por Trabalhos sobre UCTP

Trabalhos	R1	R2	R3	R4	R5	R6	R7	R8
(Abuhamdah <i>et al.</i> , 2014)	X	X	X					
(Freire; Melo, 2016)	X	X	X				X	
(Jardim; Semaan; Penna, 2016)	X	X	X					
(Soria-Alcaraz <i>et al.</i> , 2016)	X	X	X		X	X		
(Song <i>et al.</i> , 2018)	X	X	X					
(Rezaeipanah; Matoori; Ahmadi, 2021)	X	X	X	X				

Fonte: Elaborado pelo Autor (2025)

As diferenças de restrições entre o problema dificultam a comparação direta com outros resultados da literatura, devido aos diversos locais de aplicação das soluções serem distintos. Mesmo com problemas em que as restrições graves são iguais, as restrições leves causam variação do resultado final do custo da solução. Portanto, embora as diferenças sejam necessárias para atender aos cenários específicos, elas apresentam desafios adicionais para a avaliação e a replicação dos estudos.

Ao comparar diretamente as formulações, as diferenças mais destacadas entre UCTP e HSTP estão na flexibilidade dos horários e no deslocamento de estudantes e professores. No HSTP, os alunos seguem um cronograma fixo respeitando totalmente as regras impostas pela instituição em relação à organização de turmas, além de não permitir janelas entre os eventos principais. No UCTP, os alunos escolhem suas disciplinas e montam sua própria grade com base em preferências pessoais, podendo haver janelas entre as aulas e deslocamento de professores, resultando em combinações variáveis e mais complexas.

Portanto, no que foi observado, as restrições leves presentes tanto no HSTP quanto no UCTP são aulas em períodos desejáveis e balanceamento do carga horária semanal dos professores. A eliminação de janelas, presente como restrição grave no HSTP, é comumente escolhida como restrição leve no UCTP, sendo aceitas mas não desejáveis. Outras restrições leves comuns do UCTP envolvem a minimização do deslocamento de professores e alunos e uso eficiente de salas. Comparada ambas as formulações, é possível observar que o UCTP carrega um conjunto de restrições leves maiores em relação ao HSTP, o que confirma a complexidade adicional do problema.

2.5 MÉTODOS HEURÍSTICOS

As heurísticas são estratégias de busca que tem como objetivo encontrar boas soluções em tempo computacional viável para problemas de otimização. Na literatura, a busca local é amplamente utilizada, propondo soluções boas para problemas combinatórios.

2.5.1 Busca Local

É um método heurístico que melhora uma solução a partir de pequenas alterações de forma iterativa (Schaerf, 1999). O algoritmo 1 recebe um solução inicial s e aplica as estruturas de vizinhança, gerando a solução s' , que é comparada com s e tomada como solução atual, caso seja melhor. Sua principal limitação é ficar presa em ótimos locais, portanto, diversas metaheurísticas sofisticadas a utilizam com mecanismos adicionais para melhorar as soluções.

Algoritmo 1: Busca Local

```

1 Entrada:  $s, Iter_{max}$ 
2 Saída: Melhor solução  $s$  encontrada
3  $Iter \leftarrow 0$ ;
4 enquanto  $Iter < Iter_{max}$  faça
5    $Iter \leftarrow Iter + 1$ ;
6    $s' \leftarrow$  Vizinho aleatório de  $s$ ;
7   se  $f(s') < f(s)$  então
8      $Iter \leftarrow 0$ ;
9      $s \leftarrow s'$ ;
10 retorna  $s$ ;
```

2.5.2 Iterated Local Search

O *Iterated Local Search* (ILS), presente no algoritmo 2, é uma metaheurística clássica que tem como objetivo escapar de ótimos locais, combinando repetidamente a busca local e um mecanismo de perturbação na solução corrente. Seu objetivo principal é escapar dos ótimos locais, explorando as estruturas de vizinhança em busca de soluções promissoras (Lourenço; Martin; Stützle, 2010).

O algoritmo é iniciado recebendo uma solução s já melhorada pela busca local. Posteriormente um *loop* é iniciado alternando entre perturbação e busca local, em que a perturbação gera uma solução s' que é melhorada pela busca local, gerando s'' . Por fim, o custo de s'' é comparado com o custo de s e caso seja melhor, a solução s assume s'' . Este processo é repetido até que o critério de parada seja atendido, geralmente com tempo máximo ou número máximo de iterações.

Algoritmo 2: *Iterated Local Search*

```

1 Entrada:  $s_0, Time_{max}$ 
2 Saída: best  $s$ 
3  $s \leftarrow BuscaLocal(s_0)$ ;
4 enquanto  $Time < Time_{max}$  faça
5    $s' \leftarrow Perturbacao(s)$ ;
6    $s'' \leftarrow BuscaLocal(s')$ ;
7   se  $f(s'') \leq f(s)$  então
8      $Iter \leftarrow 0$ ;
9      $s \leftarrow s''$ ;
10 retorna  $s$ ;

```

2.5.3 *Simulated Annealing*

A metaheurística *Simulated Annealing* (Algoritmo 3), proposta por Kirkpatrick, Gelatt e Vecchi (1983), trata de um modelo probabilístico análogo à estratégia com recozimento de metais. Esta técnica se assemelha a busca local simples, porém aceita soluções piores baseado em uma probabilidade que é variada de acordo com a temperatura.

O algoritmo inicia com a solução e temperatura iniciais s_0 e T_0 , respectivamente. O custo da geração de vizinhos é definida por $\Delta \leftarrow f(s_0) - f(s)$ e caso $\Delta > 0$, a solução é aceita, caso contrário a solução pode ser aceita com uma certa probabilidade $e^{-\Delta/T}$ em que T é igual a temperatura atual. A temperatura T sofre uma taxa de resfriamento α , tal que $T_k \leftarrow \alpha \cdot T_{k-1}$, sendo que a medida que a temperatura vai abaixando, a probabilidade de aceitar soluções piores diminui.

2.5.4 *Variable Neighborhood Search*

O VNS definido em Mladenović e Hansen (1997), opera nas estruturas de vizinhança (N_1, N_2, \dots, N_k) trabalhando em dois passos principais: o shake sendo um método de perturbação controlada que explora o espaço de busca com mudanças na solução atual, gerando uma nova solução vizinha aleatória, aplicando a k -ésima operação de N_k na solução ótima local. A busca local, por sua vez segue o mesmo procedimento definido nas seções anteriores. O algoritmo 4 descreve a implementação do VNS.

2.5.5 *Tabu Search*

Proposta inicialmente por Glover (1986), a *Tabu Search*, presente no algoritmo 5, é uma metaheurística derivada da busca local que utiliza memória de curto prazo para escapar de ótimos locais. O método trabalha com uma lista tabu que armazena os movimentos recentes de modo que não se repitam, e comumente são classificados como movimentos

Algoritmo 3: *Simulated Annealing*

```

1 Entrada:  $s_0, T_0, \alpha, timeout, IterMax$ 
2 Saída: Melhor solução  $s$  encontrada
3 enquanto  $T > T_0$  e  $elapsedTime < Time_{max}$  faça
4   enquanto  $Iter < IterMax$  faça
5      $Iter \leftarrow Iter + 1$ ;
6      $s' \leftarrow geraVizinho(s)$ ;
7      $\Delta \leftarrow f(s') - f(s)$ ;
8     se  $\Delta > 0$  então
9        $s \leftarrow s'$ ;
10      se  $f(s') < f(s^*)$  então
11         $s^* \leftarrow s'$ ;
12      senão
13        se  $e^{-\Delta/T} > [0, 1]$  então
14           $s \leftarrow s'$ ;
15     $T \leftarrow \alpha * T$ ;
16     $Iter \leftarrow 0$ ;
17  $s \leftarrow s^*$ ;
18 retorna  $s$ ;
```

Algoritmo 4: *Variable Neighborhood Search*

```

1 Entrada:  $s_0, Time_{max}$ 
2 Saída: best  $s$ 
3  $s \leftarrow BuscaLocal(s_0)$ ;
4 enquanto  $Time < Time_{max}$  faça
5    $s' \leftarrow Shake(s)$ ;
6    $s'' \leftarrow BuscaLocal(s')$ ;
7   se  $f(s'') \leq f(s)$  então
8      $Iter \leftarrow 0$ ;
9      $s \leftarrow s''$ ;
10 retorna  $s$ ;
```

tabu. Esta metaheurística tem seu funcionamento dividido em três partes, sendo a geração de vizinhos, critério de aspiração e manipulação da lista tabu. A geração de vizinhos consiste em gerar um estrutura de vizinhanca $N(s)$, em que o melhor vizinho é escolhido. O critério de aspiração avalia se o movimento já foi feito anteriormente e se pode ser aceito. Após a aplicação do movimento, a lista tabu é manipulada adicionando o novo movimento e removendo o mais antigo, caso a capacidade da lista já esteja excedida.

Algoritmo 5: *Tabu Search*

```

1 Entrada:  $s_0, Time_{max}$ 
2 Saída: best  $s$ 
3  $\mathcal{T} \leftarrow \emptyset$ ;
4  $s \leftarrow s_0$ ;
5  $Iter \leftarrow 0$ ;
6 enquanto  $Iter < Iter_{max}$  faça
7    $\mathcal{N} \leftarrow \text{GerarVizinhança}(S)$ ;
8    $m^* \leftarrow \text{SelecionaMelhorMovimento}(\mathcal{N}, \mathcal{T})$ ;
9    $s' \leftarrow \text{AplicarMovimento}(s, m^*)$ ;
10  se  $f(s') \leq f(s)$  então
11     $s \leftarrow s'$ ;
12   $\mathcal{T} \leftarrow \mathcal{T} \cup \{m^*\}$ ;
13   $\text{AtualizarListaTabu}(\mathcal{T})$ ;
14   $Iter \leftarrow Iter + 1$ ;
15 retorna  $s$ ;
```

2.5.6 *Iterated Tabu Search*

Com o objetivo de criar um método mais robusto e aproximar do ótimo global, Misevicius, Lenkevicius e Rubliauskas (2006) propuseram o *Iterated Tabu Search* (ITS), combinando os elementos de perturbação do ILS original com a *Tabu Search* clássica. A metaheurística recebe uma solução inicial s e aplica a perturbação seguida da *Tabu Search*, obtendo a solução s'' para avaliar se há melhora no valor da função objetivo. O algoritmo 6 representa sua implementação.

Algoritmo 6: *Iterated Tabu Search*

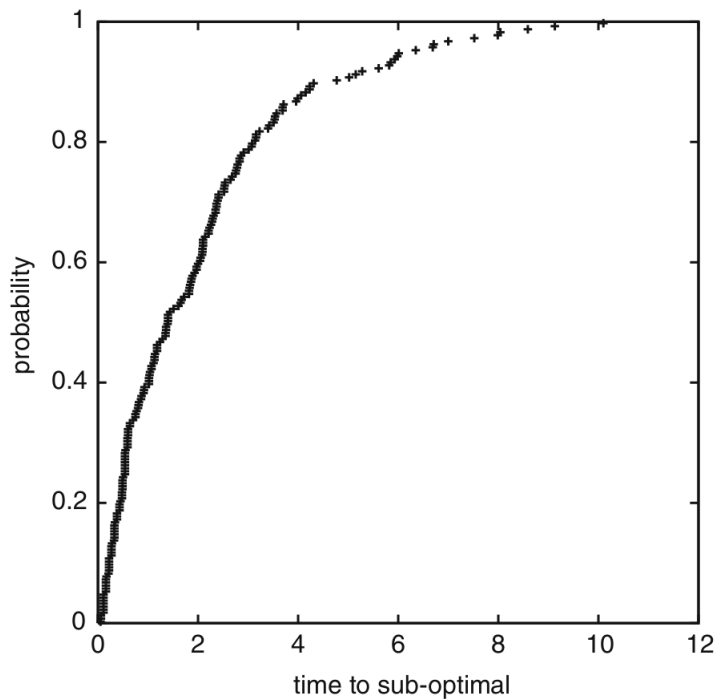
```

1 Entrada:  $s_0, Time_{max}$ 
2 Saída: best  $s$ 
3  $s \leftarrow \text{TabuSearch}(s_0)$ ;
4 enquanto  $Time < Time_{max}$  faça
5    $s' \leftarrow \text{Perturbacao}(s)$ ;
6    $s'' \leftarrow \text{TabuSearch}(s')$ ;
7   se  $f(s'') \leq f(s)$  então
8      $Iter \leftarrow 0$ ;
9      $s \leftarrow s''$ ;
10 retorna  $s$ ;
```

2.6 *TIME-TO-TARGET* (TTT PLOTS)

A técnica TTT-plots consiste em avaliar e analisar a probabilidade de um algoritmo alcançar um determinado valor alvo de função objetivo durante um intervalo de tempo definido. O algoritmo é executado n vezes, com o critério de parada sendo o valor alvo alcançado ou o tempo máximo de execução, registrando o tempo em que o valor foi alcançado. Os tempos até o alvo são registrados e ordenados em ordem crescente: $t_1 \leq t_2, \dots, \leq t_n$. Posteriormente é calculada a probabilidade associada a cada tempo, e desse modo, para todo t_i , calcula-se $p_i = (i - 1/2)/n$. Com os pontos definidos, t_i e p_i são plotados para gerar o gráfico de probabilidade acumulada. A Figura 3 apresenta um gráfico do tipo *time-to-target* para o método GRASP (Aiex; Resende; Ribeiro, 2007).

Figura 3 – Gráfico *time-to-target*



Fonte: Aiex, Resende e Ribeiro (2007)

3 METODOLOGIA

O presente capítulo aborda os passos necessários para o funcionamento do modelo do algoritmo, com a modelagem matemática necessária para descrever o problema e os dados de entrada utilizados para gerar uma saída.

3.1 DESCRIÇÃO DO PROBLEMA

Dentre as variações *Post Enrolment Course Timetabling Problem* (PE-CTT), *University Course Timetabling Problem* (UCTP) e *High School Timetabling Problem* (HSTP), o *Campus Bom Sucesso* não pode ser completamente classificado como nenhum dos problemas descritos, porém compartilha algumas características com as variações HSTP e UCTP. Quanto à estrutura de turmas, o *Campus Bom Sucesso* abstrai das restrições do HSTP devido à grade ser montada pelo próprio instituto. As estruturas de janelas são semelhantes ao UCTP, devido a não obrigatoriedade de eliminá-las. Em contrapartida, o *Campus* tem o objetivo de minimizar a janelas entre as aulas, mas como não é a preferência principal será definida como restrição leve. Desse modo, o problema em questão combina elementos de ambas as variações, porém difere em grande parte comparado ao PE-CTT devido a estrutura mais rígida de turmas do *Campus*.

3.1.1 Restrições

Cada instituição possui suas particularidades, o que pode alterar algumas restrições graves e a formulação do problema. No caso do *Campus Bom Sucesso*, houveram 7 restrições graves derivadas do HSTP e UCTP, além de 4 restrições leves que foram formuladas de acordo com a necessidade específica da instituição.

As **restrições graves** são:

- G1** Todas as aulas devem ser alocadas.
- G2** Mais de uma aula ministrada para uma turma não devem estar alocada ao mesmo tempo.
- G3** Um professor não deve lecionar mais de uma aula no mesmo intervalo de tempo.
- G4** Uma sala não pode receber mais de uma aula no mesmo intervalo de tempo.
- G5** Aulas pré-allocadas não devem ser movidas de sala ou horário.
- G6** As aulas de uma disciplina não podem ser alocadas em horários com turno diferente da oferta.
- G7** O total de aulas por semana alocadas de uma disciplina deve ser igual à carga horária semanal.

As **restrições leves** são:

- L1** O número de janelas entre as aulas de uma mesma disciplina deve ser minimizado.
- L2** O horário de preferência dos professores devem ser respeitados.
- L3** A carga horária de aulas dos professores devem ser concentradas em dias preferenciais.
- L4** Disciplinas que necessitam de recursos específicos devem preferencialmente ser alocadas em salas que atendam as necessidades.

3.1.2 Penalidades

As restrições leves recebem certas penalidades baseadas no que deve ser atendido preferencialmente, de modo que as restrições preferências tenham custo mais elevado para que mantenha a qualidade da solução. Para a modelagem do problema em questão, foi considerado uma estratégia de relaxar a função objetivo, incluindo as restrições graves à sua formulação. Esta abordagem permite o uso de metaheurísticas sem exigir viabilidade rigorosa em todo o processo de busca.

Para as restrições fortes, o custo de penalização associado foi de 10.000 para cada vez que uma restrição qualquer for violada, garantindo que o algoritmos não aceite soluções inviáveis. A tabela 2 mostra as penalizações para cada restrição leve violada.

Tabela 2 – Valor das penalidades para cada restrição

Restrição	Penalidade
G*	10.000
L1	5
L2	25
L3	20
L4	20

Fonte: Elaborado pelo Autor (2025)

3.1.3 Função Objetivo

A função objetivo descrita é responsável por calcular e minimizar o custo da solução, utilizando a soma ponderada de violação de conflitos associados a penalidades. A função é dada por:

Minimizar $f(s) =$

$$\begin{aligned}
& \sum_{k=1}^G wr_k(s) \alpha + \sum_{j=1}^D \sum_{k=1}^H \sum_{m=1}^I [w_1(D_j, H_k, I_m) \beta_1 + \\
& w_2(D_j, H_k, I_m) \beta_4] + \sum_{l=1}^P \sum_{n=1}^H \sum_{p=1}^I [w_3(P_l, H_n, I_p) \beta_2 + \\
& w_4(P_l, H_n, I_p) \beta_3]
\end{aligned} \tag{3}$$

Sendo que:

- S é o conjunto que representa as salas;
- H é o conjunto que representa os horários;
- D é o conjunto que representa as disciplinas;
- I é o conjunto que representa todas as aulas de todas as disciplinas;
- P é o conjunto que representa os professores;
- G é o conjunto que contendo todas as restrições graves;
- $wr_k(s)$ é uma função que o retorna o número de vezes que a k -ésima restrição forte foi violada;
- $w_1(D_j, H_k, I_m)$ é uma função que retorna 1 caso não haja aula da disciplina D_j no horário H_k durante o intervalo da primeira até a última aula semanal, configurando uma janela;
- $w_2(D_j, H_k, I_m)$ é uma função que retorna 1 caso uma aula de uma disciplina não esteja no conjunto de salas com recursos obrigatórios;
- $w_3(P_l, H_n, I_p)$ é uma função que retorna 1 caso uma aula de um professor esteja alocada em um horário indisponível para um professor;
- $w_4(P_l, H_n, I_p)$ é uma função que retorna 1 caso uma aula de um professor esteja fora do dia de preferência;
- α representa o peso constante para cada quebra de restrição grave;
- $\beta_1, \beta_2, \beta_3, \beta_4$ são pesos associados às restrições L1, L2, L3 e L4, respectivamente.

3.1.4 Estrutura de Dados e Vizinhaça

A estrutura principal utilizada pelo algoritmo é uma matriz 2D que representa **Sala x Intervalo de Tempo**, armazenando um índice de um vetor de eventos E , que é utilizado como uma estrutura auxiliar. Os intervalos de tempo correspondem ao número de dias da semana multiplicado pelos horários diários. Por exemplo, uma instituição que possui aulas de segunda a sexta e cada dia possui 9 horários divididos entre turnos, com 12 salas disponíveis, conta com o total de 540 *timeslots* livres para alocação. Esta estrutura já garante que a restrição grave **G4** não será violada, sendo impossível alocar mais de uma aula no mesmo intervalo de tempo na matriz 2D. Além disso, existem outras estruturas auxiliares que ajudam na consulta de disponibilidade de professores e salas, como o vetor de professores P , e o vetor de salas S .

Para a exploração do espaço de busca, são utilizadas as estruturas de vizinhaça que gera uma solução s' a partir da solução corrente s , realizando pequenas modificações que serão submetidas a uma função objetivo para serem avaliadas. Os movimentos de troca do algoritmo proposto selecionam aleatoriamente eventos e recursos de forma de haja a realocação das aulas. Desta forma, as estruturas definidas para manipulação da solução foram:

- **Move.** Uma aula aleatória a_i é movida de um *timeslot* t_1 para um *timeslot* vazio aleatório t_2 .

Figura 4 – Exemplo de movimento de vizinhaça do tipo *Move*

	Segunda	Terça	Quarta	Quinta	Sexta
13:40 - 14:30		Programação I			
14:30 - 15:20					
15:35 - 16:25			Programação Web II		
16:25 - 17:15	Banco de Dados I				Redes de Comp. Redes de Comp.
17:15 - 18:05					

Fonte: Elaborado pelo Autor (2025)

- **Swap.** Uma aula a_1 é trocada de *timeslot* com uma aula a_2 , sendo ambas escolhidas de forma aleatória.

Figura 5 – Exemplo de movimento de vizinhaça do tipo *Swap*

	Segunda	Terça	Quarta	Quinta	Sexta
13:40 - 14:30		Programação I			
14:30 - 15:20					
15:35 - 16:25			Programação Web II		
16:25 - 17:15	Banco de Dados I				Redes de Comp. Redes de Comp.
17:15 - 18:05					

Fonte: Elaborado pelo Autor (2025)

O movimento utilizado é definido de forma probabilística, em que a chance do movimento *Swap* ser escolhido é de 60%, já o *Move* é de apenas 40%. A taxa de probabilidade é alterada de acordo com a quantidade de dados fornecidos e a capacidade de alocação. Instâncias em que há muitos eventos alocados e poucos *timeslots* disponíveis, as chances do movimento *Swap* encontrarem uma melhor solução tende a ser maior, ao contrário do *Move* que encontra melhores soluções quando há muitos *timeslots* disponíveis.

3.2 GERAÇÃO DA SOLUÇÃO INICIAL E APLICAÇÃO DE METAHEURÍSTICAS

O método proposto para gerar soluções iniciais é uma Heurística Construtiva Aleatória (HCA) que possui mecanismos de reparo para garantir a factibilidade das soluções. O algoritmo recebe um conjunto de aulas a serem alocadas na matriz 2D (salas x *timeslots*), além de conjuntos auxiliares de professores e turmas. Na etapa de construção, o método seleciona uma aula a_i de forma aleatória e escolhe um *timeslot* aleatório disponível no turno em que a aula deve ocorrer. Após cada alocação, é verificado se a solução é factível, e caso não seja, o mecanismo de reparo seleciona um evento conflitante e o adiciona novamente na lista de horários não alocados. Este processo é repetido até que todas as aulas estejam alocadas. Para que o processo ocorra de forma desejada, o tamanho da lista deve ser menor que os *timeslots* disponíveis e não deve haver mais aulas que a capacidade de *timeslots* de um mesmo turno.

Algoritmo 1: Heurística Construtiva Aleatória

```

1 Entrada: Instância  $I$ 
2 Saída: Solução viável  $s$ 
3  $LC \leftarrow \text{ListaAulasNaoAlocadas}();$ 
4 enquanto  $|LC| > 0$  faça
5     escolha uma aula  $a_i$  aleatória;
6     escolha um horário  $k$  e uma sala  $j$  aleatórios;
7     se  $s$  é viável então
8         aloca aula na matriz original;
9         retira  $a_i$  de  $LC$ ;
10    senão
11        repararSolucao();
12 retorna  $s$ ;
```

A etapa de aprimoramento da solução gerada por HCA é realizada por meio das metaheurísticas *Iterated Local Search* (ILS), *Variable Neighborhood Search* (VNS) e *Iterated Tabu Search* (ITS), descritas no capítulo anterior. A escolha destes métodos se deve ao fato da eficiência para resolver problemas da classe NP em tempo hábil, que são amplamente comprovadas pela literatura.

4 RESULTADOS E EXPERIMENTOS COMPUTACIONAIS

Todos os experimentos foram realizados em um computador com o processador Ryzen 5 5600G com o clock de 3.9GHZ e 16GB de memória RAM, utilizando a linguagem C++ compilada pelo G++ 11.4.0, rodando no sistema operacional Ubuntu na versão 22.04 LTS.

Para a realização dos testes, todos os dados foram disponibilizados pelo *Campus Bom Sucesso* referente aos períodos de 2023/1 a 2024/2, contendo informação sobre professores, turmas, salas disponíveis, horários e outros dados de preferência de alocação. A tabela 3 mostra todas as características das instâncias referentes aos períodos mencionados e para a alocação, foram desconsideradas aulas EAD.

Tabela 3 – Características das Instâncias dos períodos do *Campus Bom Sucesso*

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

Fonte: Elaborado pelo Autor (2025)

4.1 ANÁLISE DOS RESULTADOS

Foram realizadas 50 execuções para cada método heurístico e cada instância. Todas as soluções obtidas se mostraram viáveis, sem violar as restrições fortes definidas no escopo do problema. Para o ILS, foi adotada uma técnica de perturbação adaptativa, com 3% de perturbação inicialmente, 5% após 50 iterações ociosas, ou seja, sem melhora, e 10% após 100 iterações ociosas. No caso do VNS, foram definidas as estruturas de vizinhanças N_1 sendo o *Move* e N_2 sendo o *Swap*, alternando sistematicamente as estruturas. Já o ITS utilizou a técnica de perturbação semelhante ao ILS, porém com taxa fixa de perturbação de 5%, e a lista tabu por sua vez foi definida com uma capacidade máxima de 100 movimentos. O critério de parada de todos os métodos foi o parâmetro $Time_{max}$ definido como 300 segundos. A tabela 4 apresenta os resultados obtidos pelos métodos, em que $f(HCA)$ representa o custo da solução inicial obtido pelo cálculo da função objetivo das instâncias geradas por Heurística Construtiva Aleatória; $f(s^*)$ representa o custo mínimo obtido por cada método e $f(\bar{s})$ mostra o custo médio que cada método retornou.

Com a Heurística Construtiva Aleatória foi possível obter uma taxa de viabilidade de 100% para todas as execuções iniciais, porém com o custo bem elevado. Aplicados os método metaheurísticos ILS, VNS e ITS, as soluções foram refinadas retornando um custo

Tabela 4 – Resultados obtidos

Período	$f(\text{HCA})$	ILS		VNS		ITS	
		$f(s^*)$	$f(\bar{s})$	$f(s^*)$	$f(\bar{s})$	$f(s^*)$	$f(\bar{s})$
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

Fonte: Elaborado pelo Autor (2025)

mínimo que correspondem entre 1% e 10% do custo da solução gerada por HCA. Em relação ao custo mínimo $f(s^*)$, todas as metaheurísticas conseguiram se igualar em algum momento, porém, ao observar o custo médio, o ITS se saiu significativamente melhor que o ILS, que por sua vez, foi superior ao VNS. A Tabela 5 apresenta uma análise quantitativa de quais foram as restrições violadas contando a partir da melhor solução obtida por cada método, provando soluções com custo igual podem ter diferentes penalidades violadas.

Tabela 5 – Penalizações na melhor solução de cada metaheurística

Período	ILS				VNS				ITS			
	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

Fonte: Elaborado pelo Autor (2025)

A técnica TTTplots de Aiex, Resende e Ribeiro (2007) foi aplicada para uma melhor visualização do quanto um método metaheurístico performou em relação ao outro, utilizando-se de um método que calcula a probabilidade de um algoritmo chegar em um determinado custo alvo de função objetivo durante um intervalo de tempo.

Para o método em questão, foram realizados 200 testes para cada instância e cada algoritmo com o $Time_{max}$ definido como 900 segundos e o alvo escolhido foi o valor mínimo do custo obtido acrescido de aproximadamente 10%, exceto a instância 2024/2 que necessitou de 13% aproximadamente, devido ao custo já estar relativamente baixo. O tempo foi aumentado para 900s com o objetivo de extrair o máximo de cada método e fazer com que todas as metaheurísticas consigam bater o valor alvo. A Tabela 6 mostra quantas execuções não bateram o alvo definido para cada método.

A figura 6 mostra que para um alvo 10% maior, o ITS foi superior comparado ao ILS, que por sua vez foi superior ao VNS. Logo nos primeiros 200s, o ITS já garantiu mais de 80% de chances e alcançou os 100% em 400s, enquanto o ILS teve sua melhor evolução até os 200s. O VNS foi inferior ao ITS e ILS, porém se igualou ao ILS após os 800s.

Tabela 6 – Alvos não alcançados

Período	<i>Target</i>	ILS	VNS	ITS
2024/2	45	26	19	0
2024/1	290	13	7	2
2023/2	270	15	11	0
2023/1	370	8	6	0

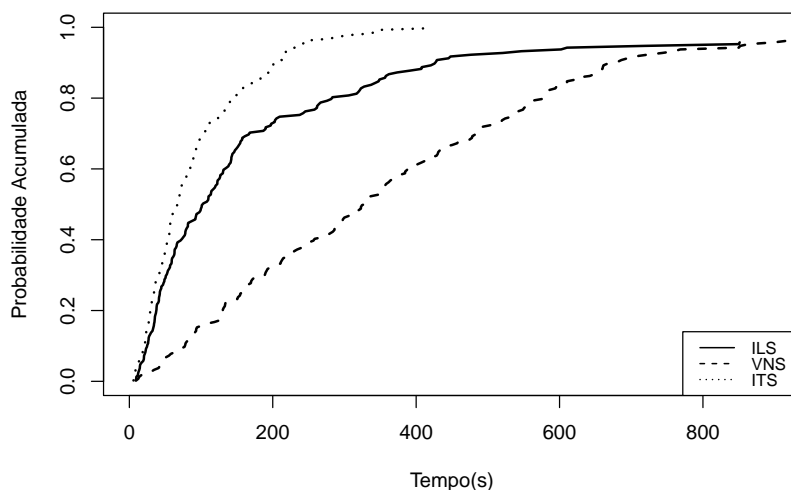
Fonte: Elaborado pelo Autor (2025)

Na figura 7, o ITS novamente foi superior ao ILS e VNS, garantindo 100% de eficácia antes do tempo estipulado. Ao comparar ILS e VNS, o ILS foi superior em todo o intervalo de tempo, exceto na marca de 400s, em que houve um empate de ambas as metaheurísticas.

O período 2024/1, presente na figura 8 foi onde houve maior equilíbrio entre ITS e ILS durante o intervalo. Nos primeiros 200s, o ILS foi levemente superior ao ITS, porém foi totalmente superado no restante do tempo. Já o VNS iniciou sua crescente a partir dos 400s, chegando a superar o ILS na marca dos 600s a 800s. Como observado, no gráfico, ainda assim o ITS foi superior novamente.

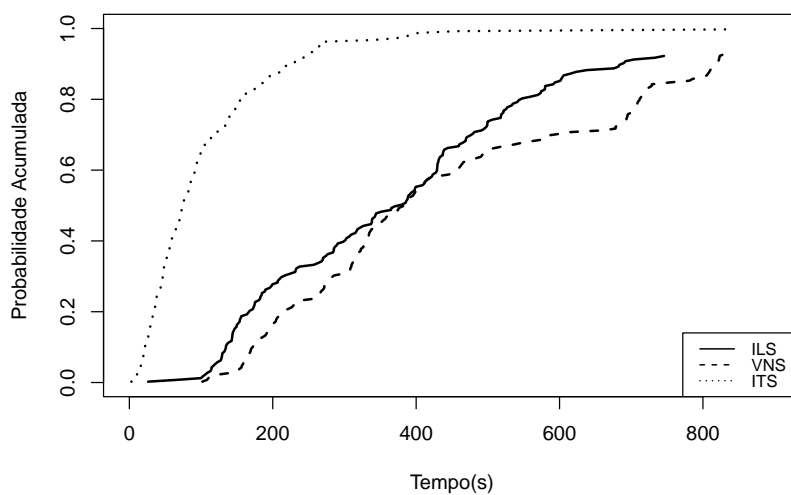
Analisando o período 2024/2, apresentado na figura 9, é possível observar que o ITS garantiu 100% de eficácia com pouco mais de 200s, sendo muito superior às outras metaheurísticas. Na disputa entre ILS e VNS, o ILS obteve vantagem em questão do tempo, porém, considerando o intervalo total, as chances do VNS alcançar o alvo foram maiores.

Figura 6 – TTTplot 2023/1 - target = 370



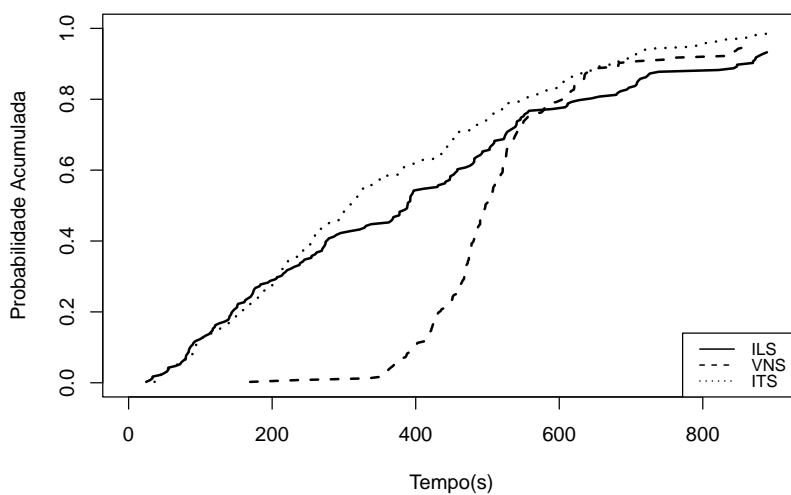
Fonte: Elaborado pelo Autor (2025)

Figura 7 – TTTplot 2023/2 - target = 270



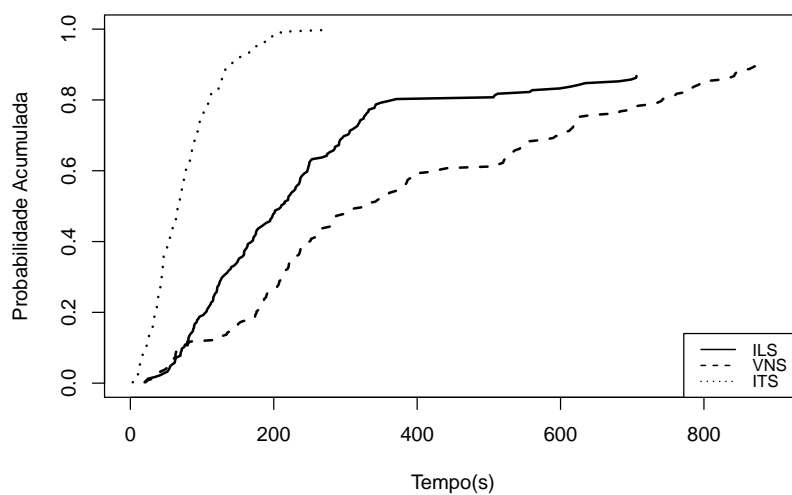
Fonte: Elaborado pelo Autor (2025)

Figura 8 – TTTplot 2024/1 - target = 280



Fonte: Elaborado pelo Autor (2025)

Figura 9 – TTTplot 2024/2 - target = 45



Fonte: Elaborado pelo Autor (2025)

5 CONCLUSÃO E TRABALHOS FUTUROS

Baseado nos experimentos, é notório que ambas as metaheurísticas obtiveram resultados extremamente satisfatórios e a implementação se mostrou promissora para resolver instâncias reais da instituição.

O ILS se mostrou superior ao VNS em velocidade e obteve melhor média de soluções em todas as instâncias no tempo de 300 segundos. Por outro lado, como observado nos gráficos TTT, o VNS consegue chegar no alvo mais vezes quando há um maior intervalo de tempo, como foi no caso dos 900 segundos.

Portanto, os resultados obtidos confirmam a eficiência das metaheurísticas *Iterated Local Search* (ILS), *Variable Neighborhood Search* (VNS) e *Iterated Tabu Search* (ITS) na resolução de problemas combinatórios como o problema abordado, respeitando as restrições grave e minimizando o custo da função objetivo. A Heurística Construtiva Aleatória foi responsável por gerar as soluções iniciais, em que se obteve 100% de taxa de factibilidade ao ser aplicada nas instâncias reais do *Campus Bom Sucesso*. As metaheurísticas por sua vez, refinaram as soluções iniciais e minimizaram o custo de forma extremamente eficiente e efetiva, com soluções que correspondem entre 1% e 10% do custo inicial. O ITS demonstrou desempenho superior em todos os cenários no cálculo médio de custo das execuções, além de superar as outras metaheurísticas de forma considerável na análise feita através do método *time-to-target*.

Dessa forma, devido às restrições do problema e as instâncias serem específicas do *Campus*, é impossível comparar o desempenho dos métodos com o que há na literatura. O caminho mais viável seria aplicar os métodos aos *datasets* da *International Competition Timetabling* (ITC) e realizar os *benchmarks* disponibilizados.

Como trabalhos futuros, propõe-se a aplicação das metaheurísticas abordadas às instâncias padronizadas da ITC, a fim de validar os métodos utilizados e obter um parâmetro de comparação com a literatura. Além disso, sugere-se a investigação de estratégias híbridas que combinem diferentes metaheurísticas, visando o aproveitamento das vantagens de cada abordagem individual.

Outra possibilidade interessante é a aplicação das técnicas às outras variações do STP como *Post Enrolment Course Timetabling Problem* (PE-CTT), que apresenta características distintas e desafios adicionais a serem adaptados.

Por fim, como objetivo futuro, espera-se integrar os métodos ao sistema de gestão de horários acadêmicos do *Campus Bom Sucesso*, para garantir a automatização de processos internos, promovendo melhorias contínuas no processo de planejamento acadêmico.

REFERÊNCIAS

- AARONSON, Scott. $P \neq NP$. *In: OPEN Problems in Mathematics*. [S. l.] Springer International Publishing, 2016. p. 1–122.
- AARTS, E; KORST, Jan HM; LAARHOVEN, Peter JM van. **Local Search in Combinatorial Optimization**. [S. l.] John Wiley e Sons, New York, NY, 1997. v. 91120.
- ABUHAMDAH, Anmar; AYOB, Masri; KENDALL, Graham; SABAR, Nasser R. Population based local search for university course timetabling problems. **Applied intelligence**, v. 40, p. 44–53, 2014. Disponível em: <https://link.springer.com/article/10.1007/s10489-013-0444-6>. Acesso em: 9 abr. 2025.
- AIEX, Renata M.; RESENDE, Mauricio G. C.; RIBEIRO, Celso C. TTT plots: a perl program to create time-to-target plots. **Optimization Letters**, 2007. Disponível em: <https://link.springer.com/article/10.1007/s11590-006-0031-4>. Acesso em: 5 jun. 2025.
- AWAD, Fouad H.; AL-KUBAISI, Ali; MAHMOOD, Maha. Large-scale timetabling problems with adaptive tabu search. **Journal of Intelligent Systems**, v. 31, n. 1, p. 168–176, 2022. Disponível em: https://www.degruyterbrill.com/document/doi/10.1515/jisys-2022-0003/html?srsltid=AfmB0ooauxJq8kR5lYH5YWvgrgYiSTx367_jTQzfJJJa-VdtEhbv7YtTX. Acesso em: 9 abr. 2025.
- BABAEI, Hamed; KARIMPOUR, Jaber; HADIDI, Amin. A survey of approaches for university course timetabling problem. **Computers Industrial Engineering**, v. 86, p. 43–59, 2015. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360835214003714>. Acesso em: 9 abr. 2025.
- BASHAB, Abeer; OSMAN, Ashraf; HASHEM, Ibrahim; AGGARWAL, Karan; MUKHLIF, Fadhil; GHALEB, Fuad; ABDELMABOUD, Abdelzahir. Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. **Computers, Materials Continua**, v. 74, p. 6461–6484, 2022. Disponível em: <https://www.techscience.com/cmc/v74n3/50939>. Acesso em: 9 abr. 2025.

BRITO, Samuel S.; FONSECA, George H.G.; TOFFOLO, Tulio A.M.; SANTOS, Haroldo G.; SOUZA, Marcone J.F. A SA-VNS approach for the High School Timetabling Problem. **Electronic Notes in Discrete Mathematics**, v. 39, p. 169–176, 2012. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1571065312000248>. Acesso em: 9 abr. 2025.

CASSEMIRO, MV; MIRANDA, Dângelo S; WANNER, Elizabeth F. Desenvolvimento de um modelo híbrido baseado em algoritmo genético e busca tabu para resolução do problema de quadro de horários escolar. *In*: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. **Anais do XLVI SBPO**. [S. l.] SOBRAPO, 2014. p. 1870–1878. Disponível em: <http://www.din.uem.br/sbpo/sbpo2014/pdf/arq0318.pdf>. Acesso em: 9 abr. 2025.

CESCHIA, Sara; DI GASPERO, Luca; SCHAERF, Andrea. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. **Computers Operations Research**, v. 39, n. 7, p. 1615–1624, 2012. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0305054811002759>. Acesso em: 9 abr. 2025.

COOK, Stephen A. The Complexity of Theorem-Proving Procedures. *In*: ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING. **Proceedings of the Third Annual ACM Symposium on Theory of Computing**. Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971. p. 151–158. Disponível em: <https://dl.acm.org/doi/10.1145/800157.805047>. Acesso em: 5 jun. 2025.

DEMIROVIĆ, Emir; MUSLIU, Nysret. MaxSAT-based large neighborhood search for high school timetabling. **Computers Operations Research**, v. 78, p. 172–180, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054816301927>. Acesso em: 9 abr. 2025.

FONSECA, George H.G.; SANTOS, Haroldo G. Variable Neighborhood Search based algorithms for high school timetabling. **Computers Operations Research**, v. 52, p. 203–208, 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054813003328>. Acesso em: 9 abr. 2025.

FONSECA, George Henrique Godim da; SANTOS, Haroldo Gambini; TOFFOLO, Túlio Ângelo Machado; BRITO, Samuel Souza; SOUZA, Marcone Jamilson Freitas. GOAL solver: a hybrid local search based solver for

high school timetabling. **Annals of Operations Research**, v. 239, p. 77–97, 2016. Disponível em: <https://link.springer.com/article/10.1007/s10479-014-1685-4>. Acesso em: 9 abr. 2025.

FRANGOULI, H.; HARMANDAS, V.; STAMATOPOULOS, P. Construction of optimum timebles for university course - a clp approach. **Proc. of the Trird International Conference and Exhibition on Practical Applications of Prolog**, p. 393–408, 1995. Disponível em: <https://cgi.di.uoa.gr/~tak/pap95.pdf>. Acesso em: 9 abr. 2025.

FREIRE, Junot; MELO, R. Formulações, heurísticas e um limite combinatório para o problema de alocação de salas de aula com demandas flexíveis. **Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional**, v. 722, p. 729, 2016. Disponível em: <http://din.uem.br/sbpo/sbpo2016/pdf/156438.pdf>. Acesso em: 9 abr. 2025.

FURTADO, Raquel Barroso Leite. **Abordagens para geração de tabelas de horários escolares: uma revisão sistemática da literatura**. Pau dos Ferros, RN: Universidade Federal Rural do Semi-Árido, 2022. Trabalho de Conclusão de Curso (Bacharelado Interdisciplinar em Ciência e Tecnologia).

GLOVER, Fred. Future paths for integer programming and links to artificial intelligence. **Computers Operations Research**, v. 13, n. 5, p. 533–549, 1986. Disponível em: <https://www.sciencedirect.com/science/article/pii/0305054886900481>. Acesso em: 9 abr. 2025.

GOH, Say Leng; KENDALL, Graham; SABAR, Nasser R. Improved local search approaches to solve the post enrolment course timetabling problem. **European Journal of Operational Research**, v. 261, n. 1, p. 17–29, 2017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221717300759>. Acesso em: 9 abr. 2025.

GOH, Say Leng; KENDALL, Graham; SABAR, Nasser R; ABDULLAH, Salwani. An effective hybrid local search approach for the post enrolment course timetabling problem. **Opsearch**, Springer, v. 57, p. 1131–1163, 2020. Disponível em: <https://link.springer.com/article/10.1007/s12597-020-00444-x>. Acesso em: 9 abr. 2025.

JARDIM, Arydiane Magalhães; SEMAAN, Gustavo Silva; PENNA, Puca Huachi Vaz. Uma heurística para o problema de programação de horários: um estudo de caso. *In*: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. **Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional**. [S. l.] SOBRAPO, 2016. p. 777–788.

Disponível em: <http://www.din.uem.br/sbpo/sbpo2016/pdf/155834.pdf>. Acesso em: 9 abr. 2025.

KHEIRI, Ahmed; ÖZCAN, Ender; PARKES, Andrew J. A stochastic local search algorithm with adaptive acceptance for high-school timetabling. **Annals of Operations Research**, v. 239, p. 135–151, 2016. Disponível em: <https://link.springer.com/article/10.1007/s10479-014-1660-0>. Acesso em: 5 jun. 2025.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. Disponível em: <https://www.science.org/doi/10.1126/science.220.4598.671>. Acesso em: 5 jun. 2025.

LOURENÇO, Helena R.; MARTIN, Olivier C.; STÜTZLE, Thomas. Iterated Local Search: Framework and Applications. *In*: HANDBOOK of Metaheuristics. Boston, MA: Springer US, 2010. P. 363–397.

MISEVICIUS, Alfonsas; LENKEVICIUS, Antanas; RUBLIAUSKAS, Dalius. Iterated tabu search: an improvement to standard tabu search. **Information Technology and Control**, v. 35, n. 3, 2006. Disponível em: <https://itc.ktu.lt/index.php/ITC/article/view/11770>. Acesso em: 9 abr. 2025.

MLADENOVIC, Nenad; HANSEN, Pierre. Variable neighborhood search. **Computers & operations research**, v. 24, n. 11, p. 1097–1100, 1997. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0305054897000312>. Acesso em: 9 abr. 2025.

NAGATA, Yuichi. Random partial neighborhood search for the post-enrollment course timetabling problem. **Computers Operations Research**, v. 90, p. 84–96, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054817302447>. Acesso em: 9 abr. 2025.

REZAEIPANAH, Amin; MATOORI, Samaneh Sechin; AHMADI, Gholamreza. A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. **Applied Intelligence**, v. 51, p. 467–492, 2021. Disponível em: <https://link.springer.com/article/10.1007/s10489-020-01833-x>. Acesso em: 9 abr. 2025.

SANTOS, H. G.; SOUZA, Marcone Jamilson Freitas. Programação de horários em instituições educacionais: formulações e algoritmos. **XXXIX SBPO-Simpósio**

Brasileiro de Pesquisa Operacional, n. 1, p. 2827–2882, 2007. Disponível em:
<http://www.din.uem.br/sbpo/sbpo2007/pdf/arq0295.pdf>. Acesso em: 9 abr. 2025.

SAVINIEC, Landir; CONSTANTINO, Ademir Aparecido. Effective local search algorithms for high school timetabling problems. **Applied Soft Computing**, v. 60, p. 363–373, 2017. Disponível em:
<https://www.sciencedirect.com/science/article/pii/S1568494617303903>. Acesso em: 9 abr. 2025.

SAVINIEC, Landir; SANTOS, Maristela Oliveira; COSTA, Alysson Machado; APARECIDO, Ademir. Multithreading iterated local search aplicado ao problema de horários escolares. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2015, Porto Seguro. **Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional**. Porto Seguro: SOBRAPO, 2015. p. 826–837. Disponível em:
<https://cdsid.org.br/sbpo2015/wp-content/uploads/2015/08/142769.pdf>. Acesso em: 9 abr. 2025.

SCHAERF, A. Local search techniques for large high school timetabling problems. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v. 29, n. 4, p. 368–377, 1999. Disponível em:
<https://ieeexplore.ieee.org/document/769755>. Acesso em: 9 abr. 2025.

SONG, Ting; CHEN, Mao; XU, Yulong; WANG, Dong; SONG, Xuekun; TANG, Xiangyang. Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem. **Applied Soft Computing**, v. 110, p. 107624, 2021. Disponível em:
<https://www.sciencedirect.com/science/article/pii/S1568494621005457>. Acesso em: 9 abr. 2025.

SONG, Ting; LIU, Sanya; TANG, Xiangyang; PENG, Xicheng; CHEN, Mao. An iterated local search algorithm for the University Course Timetabling Problem. **Applied Soft Computing**, v. 68, p. 597–608, 2018. Disponível em:
<https://www.sciencedirect.com/science/article/abs/pii/S1568494618302254>. Acesso em: 9 abr. 2025.

SORIA-ALCARAZ, Jorge A.; ÖZCAN, Ender; SWAN, Jerry; KENDALL, Graham; CARPIO, Martin. Iterated local search using an add and delete hyper-heuristic for university course timetabling. **Applied Soft Computing**, v. 40, p. 581–593, 2016. Disponível em:
<https://www.sciencedirect.com/science/article/pii/S1568494615007760>. Acesso em: 9 abr. 2025.

SOUZA, Marcone Jamilson Freitas; COSTA, FP; GUIMARÃES, IF. Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas. **Computer**, 2002. Disponível em:

<http://www.decom.ufop.br/prof/marcone/Publicacoes/ENEGEP-2002-PHE.pdf>.

Acesso em: 9 abr. 2025.

TALBI, El-Ghazali. **Metaheuristics: From Design to Implementation**. [S. l.]

Wiley Publishing, 2009. p. 91.

TAN, Joo Siang; GOH, Say Leng; KENDALL, Graham; SABAR, Nasser R. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems.

Expert Systems with Applications, v. 165, p. 113943, 2021. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0957417420307314>. Acesso em: 9 abr. 2025.