



Ministério da Educação
Universidade Federal de Ouro Preto
Escola de Minas
Departamento de Engenharia de Produção



TIMETABLING SCHOOL: MODELAGEM, SOLUÇÃO E UM ESTUDO DE CASO NA ESCOLA DE MINAS DA UFOP

ULLYSSES HENRIQUE MACHADO RIBAS

Ouro Preto MG
2022

ULLYSSES HENRIQUE MACHADO RIBAS

TIMETABLING SCHOOL: MODELAGEM, SOLUÇÃO E UM ESTUDO DE CASO NA ESCOLA DE MINAS DA UFOP

Monografia apresentada ao Curso de Engenharia de Produção da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau em Bacharel em Engenharia de Produção.

Orientador: Prof. Dr. Andre Luís Silva

Ouro Preto - MG
20 de Outubro de 2022



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO,
ADMINISTRAÇÃO E ECON



FOLHA DE APROVAÇÃO

Ulysses Henrique Machado Ribas

***Timetabling School* : modelagem, solução e um estudo de casona Escola de Minas da UFOP**

Monografia apresentada ao Curso de Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção

Aprovada em 20 de outubro de 2022.

Membros da banca

Prof. Dr. André Luís Silva - Orientador (Universidade Federal de Ouro Preto)
Prof. Dr. Aloísio de Castro Gomes Júnior - Convidado (Universidade Federal de Ouro Preto)
Prof. Dr. Helton Cristiano Gomes - Convidado (Universidade Federal de Ouro Preto)

Prof. Dr. André Luís Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 25/10/2022.



Documento assinado eletronicamente por **Andre Luis Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/10/2022, às 09:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0416950** e o código CRC **8892C2AB**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.014578/2022-54

SEI nº 0416950

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: 3135591540 - www.ufop.br

Agradecimentos

Agradeço ao Prof. Dr. Marco Antônio Moreira de Carvalho do Departamento de Computação da Universidade Federal de Ouro Preto pelo tempo empenhado e suporte ao desenvolvimento do algoritmo.

Resumo

A criação de grades horárias é uma tarefa que demanda muitos recursos, pois há diferentes variáveis que possuem ligações múltiplas. É considerada, em termos de grau de complexidade computacional como NP- *hard*, exigindo elevada capacidade de processamento. Na literatura este problema é denominado de STP (*School Timetabling Problem*) que consiste em agendar encontros entre professores e estudantes. No presente trabalho desenvolveu-se um modelo de programação linear inteira baseado nas restrições presentes em um outro modelo da literatura para otimizar a criação das referidas grades de horários de forma a minimizar o número de disciplinas alocadas em um mesmo horário de aula. A solução do modelo foi desenvolvida em C++ e para a solução do algoritmo foi realizada via o *solver* Gurobi que possui um algoritmo interno para resolução de problemas lineares baseada em *Simplex, parallel barrier with crossover, concurrent and sifting*. Utilizou-se uma base de dados real da Universidade Federal de Ouro Preto do curso de Engenharia de Produção e o algoritmo desenvolvido gerou a grade horária referente aos dez períodos do curso. Os experimentos realizados levaram 0,118 segundos para obter os resultados.

Palavras-chave: School Timetabling Problem, Grades horárias, Programação linear inteira .

Abstract

The creation of timetables is a task that demands a lot of resources, as there are different variables that have multiple links. It is considered, in terms of degree of complexity computational as NP-hard, requiring high processing capacity. this problem is called STP (School Timetabling Problem) which consists of scheduling meetings between teachers and students. In the present work, an algorithm was developed based on the constraints present in a mathematical model from the literature to optimize the creation of the referred timetables in order to minimize the number of subjects allocated in a same class time. The solution of the applied model was via the Gurobi solver that has a internal algorithm for solving linear problems based on Simplex, parallel barrier with crossover, concurrent and sifting. A real database from the Federal University was used Ouro Preto of the Production Engineering course and the developed algorithm generated the timetable for the ten periods of the course. The experiments performed took 0.118 seconds to get the results.

Keywords: School Timetabling Problem, Time grids , Integer linear programming .

Lista de figuras

Figura 1 – Execução do algoritmo no console	12
Figura 2 – Solução do algoritmo exportada em texto	12

Lista de tabelas

Tabela 1 – Métodos Exatos	2
Tabela 2 – Métodos heurísticos	3
Tabela 3 – Aplicações práticas do School Timetabling	3
Tabela 4 – Dados de entrada disciplinas	4
Tabela 5 – Resultados 1º período	9
Tabela 6 – Horários 1º período UFOP 17.2	9
Tabela 7 – Resultados 2º período	9
Tabela 8 – Horários 2º período UFOP 17.2	9
Tabela 9 – Resultados 3º período	9
Tabela 10 – Horários 3º período UFOP 17.2	9
Tabela 11 – Resultados 4º período	9
Tabela 12 – Horários 4º período UFOP 17.2	10
Tabela 13 – Resultados 5º período	10
Tabela 14 – Horários 5º período UFOP 17.2	10
Tabela 15 – Resultados 6º período	10
Tabela 16 – Horários 6º período UFOP 17.2	10
Tabela 17 – Resultados 7º período	10
Tabela 18 – Horários 7º período UFOP 17.2	10
Tabela 19 – Resultados 8º período	11
Tabela 20 – Horários 8º período UFOP 17.2	11
Tabela 21 – Resultados 9º período	11
Tabela 22 – Horários 9º período UFOP 17.2	11
Tabela 23 – Resultados 10º período	11
Tabela 24 – Horários 10º período UFOP 17.2	11

Sumário

1	INTRODUÇÃO	1
2	REVISÃO BIBLIOGRÁFICA	2
3	METODOLOGIA	4
3.1	Modelo matemático	5
4	RESULTADOS	8
4.1	Implementação do modelo	8
4.2	Análise dos resultados	12
5	CONSIDERAÇÕES FINAIS	15
	Referências	17

1 Introdução

As universidades, a cada início de período letivo, lidam com uma demanda de serviço que consiste em montar as grades horárias de cursos e turmas. Este problema é conhecido na literatura como *School Timetabling Problem* (STP) tal como narrado por [Saviniec, Santos e Costa \(2018\)](#). Segundo este autor o STP consiste em agendar encontros entre professores (ou exames) e estudantes.

Vale destacar que o STP possui alguns itens que o torna desafiador, tais como: restrição de horários possíveis dos professores, listagem das disciplinas reduzido, estudantes cursando disciplinas de diferentes períodos simultaneamente, horários disponíveis pela instituição, etc.

No contexto de “tentativa e erro”, observado em algumas universidades, a solução deste problema é obtida por meio do envolvimento de muitos servidores da instituição universitária e demanda muitos dias para a sua finalização. Além de necessitar muitos funcionários e muito tempo para ser feita, a solução gerada pode não atender demandas de estudantes que precisam cursar disciplinas de diferentes períodos (estudantes que se reprovaram em uma disciplina e precisam cursar disciplinas de diferentes períodos, por exemplo).

Neste sentido há na literatura pesquisadores que trataram de diferentes aspectos do problema. Alguns autores como [Lemos et al. \(2019\)](#), [Arbaoui, Boufflet e Moukrim \(2019\)](#) e [Saviniec, Santos e Costa \(2018\)](#) utilizam de metaheurísticas para a resolução por se mostrarem mais eficientes.

Frente ao contexto citado o objetivo do trabalho aqui descrito é desenvolver um algoritmo a partir das restrições presentes no modelo desenvolvido por [Bucco, Borna-Poulsen e Bandeira \(2017\)](#) que minimize o número de disciplinas alocadas em um mesmo horário, de forma a distribuir uniformemente as disciplinas em todos os períodos do curso. Foram feitos testes com instâncias de problemas reais da Universidade Federal de Ouro Preto para aferir a questão tratada.

O modelo em questão abrange apenas a geração de grades horárias de disciplinas. Logo, não foi considerada a alocação de salas das disciplinas geradas.

O texto está estruturado em 5 partes. A primeira refere-se a introdução. A segunda parte apresenta a revisão de literatura. Na terceira parte é apresentado o modelo matemático seguido da quarta parte com os experimentos computacionais. A última parte apresenta as considerações finais.

2 Revisão Bibliográfica

O problema tratado neste trabalho (STP) consiste em agendar encontros entre professores (ou exames) e estudantes [Saviniec, Santos e Costa \(2018\)](#). Segundo [Leite, Melício e Rosa \(2019\)](#), o STP envolve o agendamento de um conjunto de entidades a um conjunto de recursos em um número limitado de intervalos de tempo, enquanto satisfazem um conjunto de restrições. [Dorneles, Araújo e Buriol \(2017\)](#) complementam que o STP é um problema de otimização clássico que tem sido extensivamente estudado devido à sua importância prática e teórica.

O STP é definido matematicamente por [Dorneles, Araújo e Buriol \(2017\)](#) como um conjunto de classes C e um conjunto de professores T . A classe $c \in C$ representa um grupo de alunos que seguem o mesmo curso e possuem disponibilidade total. O autor também define D como o conjunto dos dias, e P como um conjunto de períodos, sendo *timeslot* (intervalo de horário) um par composto por um dia e período de aula, (d, p) , com $d \in D$ e $p \in P$, em que todos os períodos têm a mesma duração.

[Saviniec, Santos e Costa \(2018\)](#) representam o problema acrescentando as definições de [Dorneles, Araújo e Buriol \(2017\)](#) com o conjunto de períodos por dia H , o conjunto de períodos H_{td} no dia $d \in D$ para o qual o professor $t \in T$ é acessível, o número de lições semanais necessárias RL_{ct} a serem ensinadas por professor $t \in T$ para classe $c \in C$. Esta variável com índices d e p referentes ao dia e período de aula pode ter valores 1 (o que significa que o curso com esses atributos ocorre neste intervalo de tempo) ou 0 (não ocorre neste intervalo de tempo). Se for tomado como exemplo uma escola com 20 aulas, 22 cursos, 33 professores, 21 salas, 2 aulas por semana para cada curso e 35 horas *slots*, existem $20 \cdot 22 \cdot 33 \cdot 21 \cdot 2 \cdot 35 = 21.344.400$ variáveis. Este é um número considerável computacionalmente falando. No entanto, segundo [Leite, Melício e Rosa \(2019\)](#) o espaço completo de pesquisa é muito maior, há 21.344.400 combinações possíveis da variável binária.

Para a resolução destes problemas pode-se utilizar de diversos métodos, como métodos exatos e métodos heurísticos. As tabelas 1 e 2 apresentam alguns autores e os métodos utilizados para resolução do problema.

Não só os métodos de solução foram descritos na literatura. Há também a descrição de

Tabela 1 – Métodos Exatos

Autores	Ano	Método
Ribic e Konjicija (2010)	2010	Programação linear inteira bifásica
NAH e Liong (2013)	2013	Programação linear inteira
Al-Yakoob e Sherali (2015)	2015	Programação linear inteira
Bucco, Bornia-Poulsen e Bandeira (2017)	2017	Programação linear inteira

Tabela 2 – Métodos heurísticos

Autores	Ano	Método
Fonseca e Santos (2014)	2014	VNS
Dorneles, Araújo e Buriol (2017)	2017	Multi commodity flow
Saviniec, Santos e Costa (2018)	2018	Parallel Local Search
Leite, Melício e Rosa (2019)	2019	Simulated annealing

casos resolvidos pelos pesquisadores. Exemplo disto é narrado por [Andrade, Scarpin e Steiner \(2012\)](#). Estes autores desenvolveram um modelo matemático de Programação Linear Binária para a geração da grade horária do curso de Engenharia de Produção da Universidade Federal do Paraná através da linguagem VBA.

Além do caso citado, [Bucco, Borna-Poulsen e Bandeira \(2017\)](#) implementaram um modelo matemático através da linguagem C++ e o software CPLEX para solução. A tabela 3 apresenta algumas aplicações práticas do STP em instituições de ensino.

Tabela 3 – Aplicações práticas do School Timetabling

Autores	Subproblema	Tipo da solução	Instituição
Ciscon et al. (2006)	Eliminação de períodos abertos e classes isoladas	Algoritmo genético em conjunto com busca local (Algoritmo memético)	Escola Estadual Padre Rogério Abdala
Bucco, Borna-Poulsen e Bandeira (2017)	Construção de Grades Horárias em Universidades	Desenvolvido um modelo de programação linear com divisão do problema e uso de programação linear inteira mista	Ocultada
Andrade, Scarpin e Steiner (2012)	Geração da grade horária do curso de engenharia de produção	Desenvolvido um modelo de programação linear binária	UFPR
Silva e Campos (2017)	Construção de Grades Horárias em Universidade privada	Desenvolvido um modelo matemático baseado em programação inteira e quatro heurísticas : AG-MaxHorario, SA-ILS, HySST e WDJU	Ocultada

3 Metodologia

O modelo desenvolvido utilizou como dados entrada uma planilha do curso de Engenharia de Produção gerada pela Universidade Federal de Ouro Preto, referente ao período letivo de 2017/2. Neste período letivo, somando todos os períodos, foram ofertadas 48 disciplinas obrigatórias (obrigatória é uma disciplina ofertada todo semestre).

A tabela 4 apresenta um exemplo destes dados de entrada disponibilizados pela instituição. Cada disciplina possui um código único e pertence à um período. O modelo leva em consideração o número de aulas teóricas e práticas representadas por T e P respectivamente, para a geração dos horários.

Tabela 4 – Dados de entrada disciplinas

ANO	SEMESTRE	CODIGO	PERIODO	COD_CURSO	CURSO	T	P
2017	2	MTM730	1	PRO	ENG PRODUCAO	4	0
2017	2	QUI701	1	PRO	ENG PRODUCAO	2	2
2017	2	MTM700	1	PRO	ENG PRODUCAO	4	0
2017	2	PRO717	1	PRO	ENG PRODUCAO	4	0
2017	2	BCC701	1	PRO	ENG PRODUCAO	2	2
2017	2	PRO106	1	PRO	ENG PRODUCAO	2	0
...
2017	2	PRO491	9	PRO	ENG PRODUCAO	4	0
2017	2	PRO714	9	PRO	ENG PRODUCAO	4	0
2017	2	PRO910	9	PRO	ENG PRODUCAO	4	0
2017	2	PRO909	10	PRO	ENG PRODUCAO	4	0
2017	2	PRO920	10	PRO	ENG PRODUCAO	4	0

Como o curso é ofertado tanto no início quanto no meio do ano letivo, as disciplinas são oferecidas no turno da tarde ou a noite, de acordo com o semestre par/ímpar.

A grade horária semanal empregada pelo modelo tem cinco dias ($D=5$), dois períodos diários de aula para o turno da tarde e dois períodos diários de aula para o noturno.

O modelo matemático foi implementado em linguagem C++ a partir do ambiente de programação CLion e resolvidos pelo pacote de otimização GUROBI versão 7.5.2. Os experimentos foram executados com um *notebook* com processador Intel Core I5 3337U 2.7Ghz, 4GB de memória RAM e sistema operacional Ubuntu 18.04.1 LTS.

O experimento tomou como base de entrada um arquivo CSV com o mesmo *layout* da tabela 4, referentes ao 2 semestre de 2017. Nesse semestre, a universidade ofereceu 48 disciplinas obrigatórias para o curso de Engenharia de Produção. As aulas foram oferecidas de segunda-feira à sexta-feira no período noturno com 4 intervalos de horário de 1 hora aula cada.

3.1 Modelo matemático

O modelo desenvolvido resolve o problema de geração de horários a partir de um conjunto de disciplinas obrigatórias ofertadas para cada curso no início do semestre. Este conjunto é disponibilizado pelo Departamento de Tecnologia da Informação da universidade e contém os respectivos professores das disciplinas previamente designados.

A função objetivo consiste em minimizar a variável inteira y que contém o número de aulas em cada *slot* de horário de aula no dia j e horário k . Logo, a função objetivo minimiza a quantidade de disciplinas alocadas em um mesmo horário.

O modelo utiliza como entrada os parâmetros listados abaixo:

- S é o conjunto de disciplinas obrigatórias do curso;
- D é o conjunto dos dias de aula semanais, $D = \{1,2,3,\dots,5\}$ referente à segunda feira, terça feira, ... , sexta feira;
- P é o conjunto de slots de horários diários permitidos pela instituição;
- TP é a soma do número de aulas teóricas e práticas semanais previstas para a disciplina;
- $DMmin_s$ e $DMmax_s$ representam o número mínimo e máximo de aulas por dia de uma disciplina respectivamente;
- pd é o conjunto referente aos períodos letivos do curso, $pd = \{1,2,3,\dots,10\}$;
- X_{ijk} é uma variável binária de três índices (i,j,k) que correspondem ao número da disciplina, dia da semana e slot de aula diário respectivamente;
- B_{lm} é uma variável auxiliar binária e seu valor representa se a disciplina l tem aula no dia m ;
- E_{ij} e F_{ij} são variáveis auxiliares que correspondentes a primeira e última aula diária de uma turma respectivamente;
- W_{jk} é uma variável inteira que corresponde ao número de disciplinas alocadas no dia j e horário k .

As restrições que limitam esta função objetivo foram baseadas nas restrições utilizadas no modelo proposto por [Bucco, Bornia-Poulsen e Bandeira \(2017\)](#).

Função objetivo:

Minimizar y

s.a

$$\sum_{j \in D} \sum_{k \in P} X_{ijk} = TP \quad \forall i \in S \quad (3.1)$$

$$B_{lm} \geq X_{ijk} \quad \forall l \in S, m \in D, i \in S, j \in D, k \in P \quad (3.2)$$

$$DMmin_s \cdot B_{lm} \leq \sum_{j \in D} X_{ijk} \leq DMmax_s \cdot B_{lm} \quad \forall l \in S, s \in S, m \in D, i \in S, k \in P \quad (3.3)$$

$$B_{lm} + B_{l,m+1} \leq 1 \quad \forall l \in S, m \in \{0, 1, \dots, |D| - 1\} \quad (3.4)$$

$$E_{ij} \leq (|P| + 1) - (|P| + 1 - p) \cdot X_{ijk} \quad \forall i \in S, j \in D, k \in P \quad (3.5)$$

$$F_{ij} \geq K \cdot X_{ijk} \quad \forall i \in S, j \in D, k \in P \quad (3.6)$$

$$F_{ij} - E_{ij} + B_{ij} - \sum_{k \in P} X_{ijk} \leq 0 \quad \forall i \in S, j \in D \quad (3.7)$$

$$\sum X_{pd,j,k} \leq 1 \quad \forall pd \in \{1, 2, \dots, 10\}, j \in D, k \in P \quad (3.8)$$

$$W_{jk} = \sum_{i \in S} X_{ijk} \quad \forall i \in S, j \in D, k \in P \quad (3.9)$$

$$y \geq \sum W_{jk} \quad \forall j \in D, k \in P \quad (3.10)$$

As restrições fortes utilizadas na modelagem do problema foram desenvolvidas de acordo com as regras da instituição, e não são consideradas algumas particularidades como o compartilhamento de salas/laboratórios para uma mesma disciplina.

Abaixo estão descritas as restrições implementadas:

1. O número de horas aula semanais de cada disciplina deve ser alocado;
2. A alocação deverá respeitar os horários disponíveis para aula de acordo com a universidade;
3. As quantidades mínimas e máximas de *slots* de aulas diárias de cada disciplina deverão ser respeitadas;

4. Aulas de uma mesma disciplina não poderão ser alocadas em dias consecutivos;
5. Aulas diárias de uma mesma disciplina deverão ser consecutivas;
6. Disciplinas do mesmo semestre de um curso não poderão ter conflitos de horários.

A equação (3.1) atende a restrição 1, ao limitar o total de disciplinas alocadas (X_{ijk}) em relação ao número de aulas teóricas e práticas (TP).

Os horários permitidos pela instituição são restringidos pela restrição 2 definida pela equação (3.2) onde B_{lm} é uma variável auxiliar binária que informa se a disciplina l tem aula no dia m .

As quantidades mínimas e máximas de aulas diárias de cada disciplina é respeitada pela restrição 3, que é definida pela equação (3.3) que restringe o total de aulas diárias de uma disciplina ao multiplicar a variável auxiliar B_{lm} ao valor dos parâmetros $DMmin_s$ e $DMmax_s$.

A equação (3.4) garante que a restrição 4 (disciplinas de um mesmo período não podem ter aulas em dias consecutivos) seja atendida ao definir que a soma das variáveis auxiliares B_{lm} (aula no dia m) e B_{lm+1} (aula no próximo dia $m + 1$) deve ser menor que 1.

O conjunto de equações (3.5) e (3.6) que definem o valor de E_{ij} e F_{ij} correspondentes a primeira e última aula diária de uma turma respectivamente. A consecutividade de horários de uma mesma disciplina (restrição 5) é definida pela equação (3.5) que calcula o valor de P (parâmetro correspondente ao total de horários permitidos na instituição diariamente).

Os conflitos de horários em disciplinas do mesmo semestre de um curso são eliminados pela equação (3.8) que limita a disponibilidade da disciplina do período pd a 1.

Finalmente, a função objetivo de minimização de y é definida pela equação (3.10), onde W é a uma variável inteira que corresponde ao número de disciplinas alocadas no dia j e horário k .

4 Resultados

A primeira fase deste trabalho foi desenvolver um modelo de programação linear inteira para o STP. O modelo desenvolvido é apresentado no Capítulo 3.

4.1 Implementação do modelo

O modelo matemático apresentado na Seção 3.1 foi implementado utilizando a linguagem C++ juntamente com a biblioteca Gurobi, que possui um algoritmo interno para resolução de problemas lineares baseada em *Simplex, parallel barrier with crossover, concurrent and sifting*.

Os parâmetros de entrada do modelo foram gerados através de uma consulta SQL disponibilizada pelo Departamento de Tecnologia da Informação da UFOP e armazenados em um arquivo CSV (*Comma-Separated values*) que é lido pelo programa que executa o modelo. Primeiramente são lidas as disciplinas, representadas por códigos alfanuméricos utilizados pela instituição, seguidas de informações como período do curso, número de aulas teóricas, práticas etc.

Ao final da execução o programa exibe o valor da função objetivo conforme a figura 1 e consulta o valor das variáveis X_{ijk} para gerar a lista de disciplinas com seus respectivos horários. Estes horários são exportados na forma de texto conforme o exemplo da Figura 2.

Através dos dados gerados no arquivo de texto referente à figura 2, foram elaboradas tabelas na forma de grade horária para apresentar a solução. Estes resultados podem ser visualizados nas tabelas 5,7,9,11,13,15,17,19,21 e 23. A primeira coluna exibe os horários padrões da instituição, e as colunas seguintes exibem os dias de aula letivos semanais.

Já as tabelas 6,8,10,12,14,16,18,20,22 e 24 apresentam os horários reais que foram utilizados no mesmo semestre dos dados utilizados pelo modelo (segundo semestre de 2017). Estes dados estão disponíveis no site da instituição em <https://zeppelin10.ufop.br/HorarioAulas/index.xhtml>.

Devido o curso de Engenharia de Produção possuir egresso semestral, as disciplinas de períodos ímpares são oferecidas em horários distintos de períodos pares (diurno / noturno).

A execução do modelo foi realizada em um computador executando o sistema operacional Ubuntu 18.04.1 LTS. e com as seguintes configurações de hardware: processador Intel(R) Core(TM) I5 3337U @ 2.7Ghz com 4GB de memória RAM e o modelo levou 0,118 segundos para processar a grade horária do curso com 45 disciplinas.

Tabela 5 – Resultados 1º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	QUI701	MTM730	QUI701	MTM730	PRO717
21:00 à 22:40	MTM700	BCC701	PRO717	BCC701	MTM700

Tabela 6 – Horários 1º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO717	QUI701	MTM730	BCC701	MTM700
21:00 à 22:40	QUI701	BCC701	MTM700	PRO717	MTM730

Tabela 7 – Resultados 2º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	FIS130	BCC702	MTM702	FIS130	ARQ700
15:20 à 17:00	MTM702	PRO718	ARQ700	BCC702	PRO718

Tabela 8 – Horários 2º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	ARQ700		BCC702	PRO718	BCC702
15:20 à 17:00	ARQ700	MTM702	FIS130	MTM702	FIS130
17:10 à 18:50		PRO718			

Tabela 9 – Resultados 3º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO227	PRO309	FIS131	PRO227	FIS131
21:00 à 22:40	PRO701	MTM703	PRO701	MTM703	PRO309

Tabela 10 – Horários 3º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO701	PRO227	MTM703	PRO227	PRO701
21:00 à 22:40	FIS131	PRO309	FIS131	PRO309	MTM703

Tabela 11 – Resultados 4º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	FIS214	PRO404	PRO226	PRO404	PRO226
15:20 à 17:00	PRO704	FIS132	FIS214	PRO704	FIS132

Tabela 12 – Horários 4º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	FIS214	PRO704	FIS214	PRO404	PRO704
15:20 à 17:00	FIS132	PRO226	FIS132	PRO226	PRO404

Tabela 13 – Resultados 5º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO706	FIS133	BCC760	FIS133	FIS215
21:00 à 22:40	PRO511	FIS215	PRO511	PRO706	BCC760

Tabela 14 – Horários 5º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	FIS133	PRO511	BCC760	FIS133	PRO706
21:00 à 22:40	BCC760	FIS215	PRO706	FIS215	PRO511

Tabela 15 – Resultados 6º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	PRO708	PRO236	PRO126	PRO236	PRO720
15:20 à 17:00	PRO126	PRO720	CAT700	PRO708	CAT700

Tabela 16 – Horários 6º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	CAT700	CAT700	PRO720	PRO236	PRO126
15:20 à 17:00	PRO720	PRO236	PRO708	PRO126	PRO708

Tabela 17 – Resultados 7º período

HORÁRIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	CAT107	PRO721	PRO723	PRO809	PRO807
21:00 à 22:40	PRO723	PRO807	PRO809	PRO721	CAT107

Tabela 18 – Horários 7º período UFOP 17.2

HORARIO	SEGUNDA	TERCA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO807	PRO721	PRO809	CAT107	PRO809
21:00 à 22:40	PRO723	CAT107	PRO723	PRO807	PRO721

Tabela 19 – Resultados 8º período

HORÁRIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	PRO710	PRO724	PRO716	PRO808	PRO725
15:20 à 17:00	PRO808	PRO725	PRO710	PRO724	PRO716

Tabela 20 – Horários 8º período UFOP 17.2

HORARIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
13:30 à 15:10	PRO724	PRO716	PRO724	PRO808	PRO716
15:20 à 17:00	PRO808	PRO710	PRO725	PRO710	PRO725

Tabela 21 – Resultados 9º período

HORÁRIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
19:00 à 20:40	PRO910	PRO714		PRO910	PRO491
21:00 à 22:40			PRO491	PRO714	

Tabela 22 – Horários 9º período UFOP 17.2

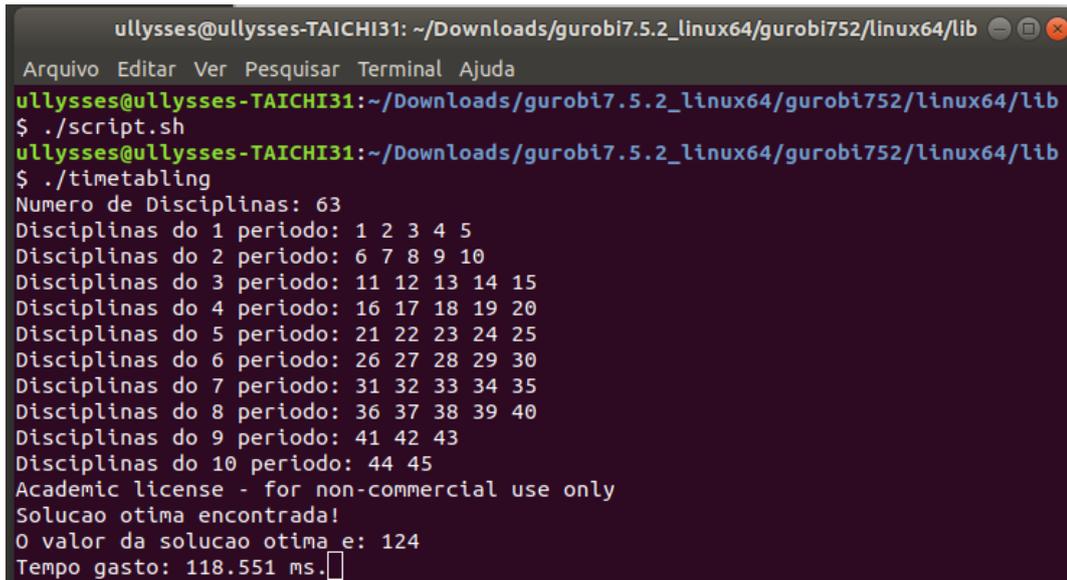
HORARIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
19:00 à 20:40		PRO714	PRO910	PRO714	
21:00 à 22:40	PRO910				

Tabela 23 – Resultados 10º período

HORÁRIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
13:30 à 15:10		PRO920			PRO909
15:20 à 17:00		PRO909			PRO920

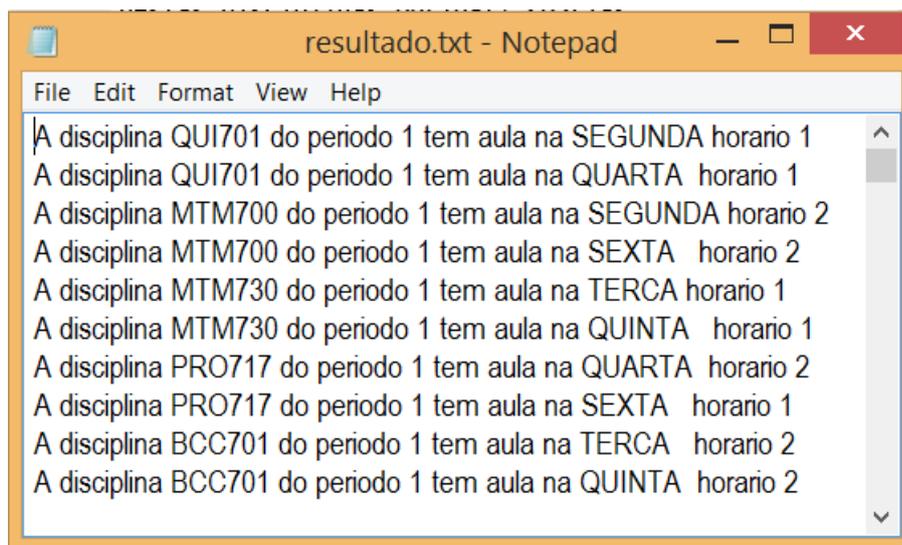
Tabela 24 – Horários 10º período UFOP 17.2

HORARIO	SEGUNDA	TERÇA	QUARTA	QUINTA	SEXTA
13:30 à 15:10					
15:20 à 17:00		PRO909		PRO909	
17:10 à 18:50		PRO920		PRO920	



```
ullys@ullys-TAICHI31: ~/Downloads/gurobi7.5.2_linux64/gurobi752/linux64/lib
Arquivo Editar Ver Pesquisar Terminal Ajuda
ullys@ullys-TAICHI31:~/Downloads/gurobi7.5.2_linux64/gurobi752/linux64/lib
$ ./script.sh
ullys@ullys-TAICHI31:~/Downloads/gurobi7.5.2_linux64/gurobi752/linux64/lib
$ ./timetabling
Numero de Disciplinas: 63
Disciplinas do 1 periodo: 1 2 3 4 5
Disciplinas do 2 periodo: 6 7 8 9 10
Disciplinas do 3 periodo: 11 12 13 14 15
Disciplinas do 4 periodo: 16 17 18 19 20
Disciplinas do 5 periodo: 21 22 23 24 25
Disciplinas do 6 periodo: 26 27 28 29 30
Disciplinas do 7 periodo: 31 32 33 34 35
Disciplinas do 8 periodo: 36 37 38 39 40
Disciplinas do 9 periodo: 41 42 43
Disciplinas do 10 periodo: 44 45
Academic license - for non-commercial use only
Solucao otima encontrada!
O valor da solucao otima e: 124
Tempo gasto: 118.551 ms.
```

Figura 1 – Execução do algoritmo no console



```
File Edit Format View Help
A disciplina QUI701 do periodo 1 tem aula na SEGUNDA horario 1
A disciplina QUI701 do periodo 1 tem aula na QUARTA horario 1
A disciplina MTM700 do periodo 1 tem aula na SEGUNDA horario 2
A disciplina MTM700 do periodo 1 tem aula na SEXTA horario 2
A disciplina MTM730 do periodo 1 tem aula na TERCA horario 1
A disciplina MTM730 do periodo 1 tem aula na QUINTA horario 1
A disciplina PRO717 do periodo 1 tem aula na QUARTA horario 2
A disciplina PRO717 do periodo 1 tem aula na SEXTA horario 1
A disciplina BCC701 do periodo 1 tem aula na TERCA horario 2
A disciplina BCC701 do periodo 1 tem aula na QUINTA horario 2
```

Figura 2 – Solução do algoritmo exportada em texto

4.2 Análise dos resultados

O modelo foi executado para uma instância com 45 disciplinas das 48 disciplinas obrigatórias da matriz curricular do curso de Engenharia de Produção, e o programa levou 0,118 segundos para realizar a alocação.

As três disciplinas restantes são referentes ao estágio supervisionado, e disciplinas de Projeto Final de Graduação que possuem horários pré-determinados, devendo ser fixados no modelo.

O modelo implementado tinha como objetivo gerar a grade horária do curso de forma a minimizar o número de disciplinas alocadas em um mesmo horário, e é possível notar através dos

resultados obtidos nas tabelas 5,7,9,11,13,15,17,19,21 que a distribuição ocorreu de forma uniforme, além de respeitar as restrições implementadas como o número de aulas semanais, horários disponíveis pela instituição, a não alocação de uma mesma disciplina em dias consecutivos, etc.

Em comparação com os horários utilizados pela instituição no mesmo semestre dos dados utilizados pelo modelo, é possível perceber que em alguns períodos os resultados deste trabalho se mostraram mais eficazes, como na tabela 7 que apresenta distribuição de disciplinas mais uniforme em relação a tabela 8 utilizada pela instituição.

Além disso, em alguns períodos a grade de horários da universidade não respeitou o intervalo de pelo menos dois dias entre as aulas de uma mesma turma de uma disciplina ao alocar as disciplinas QUI701 (Tabela 6), PRO404 (Tabela 12) e CAT700/PRO126 (Tabela 16) consecutivamente. Este intervalo é exigido pela CEPE no item 5, que diz: "Deverá ser mantido um intervalo de pelo menos dois dias entre as aulas de uma mesma turma de uma disciplina". Esta situação não acontece em nenhum dos períodos onde os horários foram gerados através do modelo desenvolvido.

Em todos os dez períodos onde foram geradas grades horárias, é possível perceber que o modelo respeitou as restrições implementadas que também são exigidas pela instituição, como o item 2 da CEPE 4945 (As aulas devem ser distribuídas em horários que preencham de maneira mais uniformemente possível todos os turnos de 2^a a 6^a feira.), o item 3 (As aulas de disciplinas de carga horária semanal par deverão ser agrupadas em duas horas aula seguidas), o item 5 (Deverá ser mantido um intervalo de pelo menos dois dias entre as aulas de uma mesma turma de uma disciplina, observados os itens 3 e 4 desta diretriz).

Em relação ao tempo computacional, o modelo desenvolvido obteve uma boa performance ao gastar apenas 0,118 segundos para processar 45 disciplinas, uma média de 2,6 mili segundos para cada disciplina. Esta robustez pode ser atribuída à otimização do pacote Gurobi, que resolveu a programação linear inteira.

Alguns autores utilizaram outros pacotes de otimização, como é o caso de [Silva e Campos \(2017\)](#) e [Bucco, Bornia-Poulsen e Bandeira \(2017\)](#) que utilizaram o software CPLEX. E [Andrade, Scarpin e Steiner \(2012\)](#) que resolveram o problema através do software LINGO. Assim como neste trabalho ambos autores utilizaram a linguagem de programação C++, com exceção de [Andrade, Scarpin e Steiner \(2012\)](#) que adotaram a linguagem *Visual Basic for Applications* (VBA).

A efeito de comparação [Silva e Campos \(2017\)](#) processaram 2248 disciplinas com 42 minutos, resultando na média de 1,12 segundos para cada disciplina. Já [Andrade, Scarpin e Steiner \(2012\)](#) realizaram a alocação de 24 disciplinas com 42 segundos, o que resulta em média 0,57 segundos por disciplina.

Através destes números podemos concluir que o modelo aqui desenvolvido consegue processar disciplinas com um tempo cerca de quatro a seis vezes menor. No entanto estes

tempos de processamento podem variar muito dependendo do número de restrições utilizada em cada modelo, bem como a utilização de outros parâmetros como alunos, professores etc. A configuração de hardware também pode influenciar, apesar que ambos trabalhos foram executados em *notebooks* com características semelhantes.

5 Considerações Finais

Este trabalho visou o desenvolvimento de um algoritmo baseado em um modelo construído utilizando-se de técnicas de programação linear inteira, tendo como objetivo automatizar a geração de grades horárias da UFOP de acordo com as normas da CEPE 4945 (Resolução interna que define diretrizes gerais para elaboração de horários de aulas de disciplinas da graduação) e minimizar a alocação de disciplinas em um mesmo horário, podendo assim, diminuir o tempo despendido pelos gestores da instituição para esta tarefa.

Neste trabalho foi desenvolvido uma solução para a primeira parte do problema STP, ou seja, a alocação de horários de aula. Foram construídas as grades horárias semanais do curso de Engenharia de Produção, através de uma distribuição uniforme de horários gerados pelo modelo.

Este objetivo foi alcançado, pois foi desenvolvido um modelo de programação linear inteira para o problema e esse modelo foi executado com instâncias geradas a partir de dados das disciplinas.

O modelo desenvolvido conseguiu atender as restrições implementadas bem como as normas das CEPE 4945 e distribuiu uniformemente a alocação das disciplinas em um tempo de processamento relativamente pequeno.

Diferente do realizado por [Bucco, Bornia-Poulsen e Bandeira \(2017\)](#) que dividiu o problema em duas partes e realizou também a alocação de salas de aulas, neste estudo esta parte foi desconsiderada devido a complexidade do problema, pois se eleva o número de variáveis e restrições. Sendo assim, este modelo pressupõe que os professores estarão disponíveis nos dias e horários gerados para as suas disciplinas e que as salas de aula serão atribuídas posteriormente de acordo com a grade horária gerada.

Um problema que pode inviabilizar o uso desse modelo de forma efetiva na universidade é que a alocação sugerida pelo modelo pode não ser viável para a alocação de professores, visto que os horários alocados para as disciplinas podem não ser compatíveis com os horários disponíveis dos professores.

Como sugestões de pesquisas futuras, observa-se a criação de mais restrições, tais como as restrições horárias de professores, de laboratórios compartilhados e restrições de exceção para tratar casos como a disciplina de Trabalho Final de Graduação que é alocada aos sábados devido a mesma não possuir aula presencial.

A utilização de métodos heurísticos para a resolução do modelo também pode ser estudada com o objetivo de se obter melhores alocações e processar maiores quantidades de informações ao mesmo tempo como por exemplo a grade horária de todos os cursos da instituição.

Sugere-se também o desenvolvimento de outro modelo para tratar da segunda etapa do

processo, que é a alocação dos espaços físicos (Salas de aula) e também a implementação de uma interface amigável para visualização dos resultados gerados, facilitando assim a utilização da solução e replicação em outros cursos da UFOP ou em outras universidades.

Referências

- AL-YAKOOB, S. M.; SHERALI, H. D. Mathematical models and algorithms for a high school timetabling problem. *Computers & Operations Research*, v. 61, p. 56 – 68, 2015. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054815000428>>. 2
- ANDRADE, P.; SCARPIN, C.; STEINER, M. Geração da grade horária do curso de engenharia de produção da ufpr através de programação linear binária. In: . [S.l.: s.n.], 2012. 3, 13
- ARBAOUI, T.; BOUFFLET, J.-P.; MOUKRIM, A. Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems. *Computers & Operations Research*, v. 106, p. 133 – 142, 2019. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054819300528>>. 1
- BUCCO, G. B.; BORNIA-POULSEN, C. J.; BANDEIRA, D. L. Desenvolvimento de um modelo de programação linear para o problema da construção de grades horárias em universidades. *Gestão & Produção*, v. 24, n. 1, p. 40 – 49, 2017. ISSN 1806-9649. Disponível em: <<https://www.scielo.br/j/gp/a/CB7rkZHkxqvd5yJMtWRWnfH/abstract/?format=html&lang=pt>>. 1, 2, 3, 5, 13, 15
- CISCON, L. A. et al. The school timetabling problem: A focus on elimination of open periods and isolated classes. In: *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*. [S.l.: s.n.], 2006. p. 70–70. 3
- DORNELES Ártion P.; ARAÚJO, O. C. de; BURIOL, L. S. A column generation approach to high school timetabling modeled as a multicommodity flow problem. *European Journal of Operational Research*, v. 256, n. 3, p. 685 – 695, 2017. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221716305392>>. 2, 3
- FONSECA, G. H.; SANTOS, H. G. Variable neighborhood search based algorithms for high school timetabling. *Computers & Operations Research*, v. 52, p. 203 – 208, 2014. ISSN 0305-0548. Recent advances in Variable neighborhood search. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054813003328>>. 3
- LEITE, N.; MELÍCIO, F.; ROSA, A. C. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, v. 122, p. 137 – 151, 2019. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417418308169>>. 2, 3
- LEMOS, A. et al. Room usage optimization in timetabling: A case study at universidade de lisboa. *Operations Research Perspectives*, v. 6, p. 100092, 2019. ISSN 2214-7160. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214716018301696>>. 1
- NAH, A.; LIONG, C.-Y. Mathematical modelling of university timetabling: A mathematical programming approach. *International Journal of Applied Mathematics and Statistics*, v. 37, p. 110–122, 01 2013. 2
- RIBIC, S.; KONJICIJA, S. A two phase integer linear programming approach to solving the school timetable problem. In: . [S.l.: s.n.], 2010. p. 651 – 656. 2

SAVINIEC, L.; SANTOS, M. O.; COSTA, A. M. Parallel local search algorithms for high school timetabling problems. *European Journal of Operational Research*, v. 265, n. 1, p. 81 – 98, 2018. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221717306562>>. 1, 2, 3

SILVA, I. L. da; CAMPOS, S. C. Programação inteira para timetabling em uma universidade privada. *Simpósio Brasileiro de Pesquisa Operacional*, v. 1, 8 2017. Disponível em: <<http://www.din.uem.br/sbpo/sbpo2017/pdf/169038.pdf>>. 3, 13