



PROGRAMAÇÃO INTEIRA PARA TIMETABLING EM UMA UNIVERSIDADE PRIVADA

Italo Luis da Silva

Faculdade Governador Ozanam Coelho
Rua Dr. Adjalme da Silva Botelho, 20 – Bairro Seminário, Ubá – MG, 36500-000
italo@maleldil.com

Saulo Cunha Campos

Faculdade Governador Ozanam Coelho
Rua Dr. Adjalme da Silva Botelho, 20 – Bairro Seminário, Ubá – MG, 36500-000
saulo@fagoc.br

RESUMO

Este artigo aborda o problema de alocação de horários, disciplinas, professores e turmas. O objetivo é encontrar uma solução que obedece a um conjunto de restrições *hard* (obrigatórias) enquanto tenta satisfazer restrições *soft* (não-obrigatórias), cada uma com um peso associado, com o menor número de violações. O estudo é baseado em uma instituição de ensino superior privada, e as restrições foram definidas a partir de trabalhos da literatura, tendo seus pesos escolhidos pelos coordenadores de curso da instituição. Como métodos de resolução foram investigados um modelo matemático baseado em programação inteira e quatro heurísticas: AG-MaxHorario, SA-ILS, HySST e WDJU. Foram efetuados experimentos computacionais para calibração e comparação dos algoritmos. Os resultados dos experimentos são avaliados por métodos estatísticos válidos. Ao final percebe-se que o SA-ILS e o WDJU obtêm resultados estatisticamente melhores que as outras heurísticas, com um *gap* médio de 14% em relação ao resultado ótimo produzido pelo modelo matemático.

PALAVRAS CHAVE. *University Course Timetabling, Programação Inteira, Meta-heurística*

Tópicos: *timetabling, meta-heurísticas*

ABSTRACT

This article approaches the problem of allocating time slots, courses, teachers and classes. The objective is to find a solution that fulfills a set of hard restrictions while tries to satisfy soft constraints, which one with an associated weight, using the least amount of violations. The study is based on a real private higher education institution, and the restrictions were defined from works in the literature, having their weights chosen by course coordinators from the institution. A mathematical model and four meta-heuristics (AG-MaxHorario, SA-ILS, HySST and WDJU) were chosen as resolution methods. Computational experiments are performed to calibrate and compare the algorithms; the results are evaluated by valid statistical methods. At the end it's possible to conclude that SA-ILS and WDJU obtain statistically better results than the other heuristics, with an average gap of 14% in relation to the optimal result obtained by the mathematical model.

KEYWORDS. *University Course Timetabling, Integer Programming, Meta-heuristic*

Paper topics: *integer programming model, meta-heuristics comparison*



1. Introdução

Segundo Schaerf [1999], o problema de *timetabling* consiste em agendar uma sequência de aulas entre professores e estudantes em um período de tempo delimitado, satisfazendo um conjunto de restrições. O problema vem sendo estudado desde 1963, por Gotlieb, e se aplica a instituições de ensino tanto superior quanto básico, privadas ou públicas.

Esse problema existe em qualquer instituição de ensino, e se torna cada vez mais complexo conforme a instituição cresce, uma vez que pertence à classe NP-Difícil, para qual não se conhecem algoritmos de tempo de polinomial para a obtenção de soluções ótimas [Levitin 2003].

O problema de *timetabling* é pesquisado há mais de 50 anos, começando por Gotlieb em 1963. Técnicas pioneiras foram desenvolvidas com base na forma como uma pessoa desenvolvia um *timetable* [Schmidt e Strohlein 1979], utilizando melhorias sucessivas. Também existem técnicas baseadas em coloração de grafos para a resolução do problema [Welsh e Powel 1967]. Even, em 1976, aplica uma técnica de encontrar uma sequência maximal de emparelhamentos no grafo bipartido composto por professores e turmas.

Esse trabalho aborda um subproblema do *Timetabling*, o *University Course Timetabling* introduzido por [De Werra 1985], que trata o problema de criar atribuições de *slots* de tempos, salas de aula, professores e disciplinas, de forma a impedir que uma sala seja ocupada por duas disciplinas simultaneamente, ou então que dois professores lecionem a mesma aula, ou que um professor lecionem duas disciplinas no mesmo *slot* de tempo [Schaerf 1999]. O indicador de qualidade de uma solução varia. [De Werra 1985] utiliza o somatório das preferências obtidas por alocações do tipo “disciplina–professor”. Autores como [Eiselt e Laporte 1987] e [Aubin e Ferland 1989] adotam restrições *soft* e as utilizam para avaliar a qualidade da solução, modelando restrições a partir de critérios de preferências dos alunos ou espaçamento entre aulas.

[Tripathy 1984] considera uma matriz de conflitos entre estudantes cursando dois cursos i e j simultaneamente. A função objetivo considera a minimização do número de casos onde i e j são agendados ao mesmo tempo. Utiliza programação inteira, aplicando a técnica de Relaxamento Lagrangiano. [Ferland e Roy 1985] formulam o problema como um problema de alocação, resolvendo-o por uma redução a um problema de alocação quadrático. [Chahal e De Werra 1989] utilizaram técnicas de resolução de fluxo em redes. Muitos autores após [Eiselt e Laporte 1987] dividiram as restrições entre *hard* e *soft*, onde as restrições *hard* definem as soluções factíveis, e as restrições *soft* são incorporadas à função objetivo, diferenciando a qualidade das soluções. [Hertz 1992] aplica Busca Tabu, enquanto [Paechter et al. 1994] utilizam algoritmos genéticos. [Carrasco e Pato 2000] utilizam um algoritmo genético multiobjetivo, combinando dois pontos de vista: o orientado aos estudantes e o orientado aos professores. [Daskalaki e Birbas 2005] formulam um outro modelo para o problema, utilizando também de restrições *soft* na função objetivo. [Silva 2014] também sugere um modelo matemático, tendo como estudo de caso a Universidade Federal Fluminense.

No ramo das meta-heurísticas, [Abuhamdah et al. 2014] estudam um *Population-Based Local Search*, adaptando o *Gravitational Emulation Local Search*. [Jardim et al. 2016] aplicam um *Iterated Local Search*, com estudo de caso na Universidade Federal Fluminense. [Silva et al. 2016] apresentam um modelo matemático para geração dos *timetables*, aplicando o AG-MaxHorário para sua resolução. Como função objetivo utiliza-se o somatório do número de horas dos agendamentos semanais dos alunos. [Kheiri et al. 2012] introduzem o HySST, uma hiper-heurística que explora o espaço de operadores de mutação. [Wilke e Killer 2010] aplicam o *Walk Down Jump Up*, uma meta-heurística que trabalha em períodos de refinamento rápido seguidos por períodos de resfriamento. [Fonseca 2013] utiliza uma combinação das meta-heurísticas SA e ILS denominada SA-ILS, em que a cada iteração a solução vigente é passada sequencialmente para o SA e depois para o ILS.

O presente trabalho trata o mesmo problema abordado por [Silva et al. 2016], aplicado a uma instituição de ensino superior privada real, porém uma formulação diferente para o problema é proposta. Ao invés de empregar uma FO obtida do somatório da carga horária semanal de cada aluno (calculada a partir dos agendamentos gerados), é adotada uma FO baseada no ponto de vista



dos coordenadores de curso, visando a construção de um horário que maximize suas preferências. As restrições *hard* se mantiveram, mas restrições *soft* foram inseridas para o cálculo da FO. Essas restrições foram obtidas através do estudo de trabalhos da literatura.

Além da elaboração de um modelo matemático, também são feitas comparações entre quatro heurísticas: o AG-MaxHorário de [Silva et al. 2016], HySST [Kheiri et al. 2012], Walk Down Jump Up [Wilke e Killer 2010] e SA-ILS [Fonseca 2013]. Sendo estas três últimas finalistas da terceira competição internacional de *timetabling* (ITC2011) [Post et al. 2016].

O artigo está organizado da seguinte forma: a seção 2 descreve o problema e a metodologia usada para definir a FO, a seção 3 define o modelo matemático, a seção 4 apresenta a metodologia dos experimentos, a seção 5 apresenta a calibração das heurísticas, a seção 6 demonstra a comparação dos algoritmos e a seção 7 expõe as considerações finais e trabalhos futuros.

2. Descrição do problema

O problema abordado é o *University Course Timetabling* (UCT), especializado para a instituição de ensino alvo do estudo de caso. Para representação da solução é elaborada uma matriz de horários tridimensional, onde a dimensão Z corresponde às turmas (períodos ou camadas), a dimensão X corresponde aos dias da semana e a dimensão Y aos horários de um dia (ou blocos). Cada célula dessa matriz é um par professor-disciplina.

Para a elaboração dessa matriz, deve-se agendar encontros entre disciplinas e turmas em horários determinados, atribuindo a cada disciplina um professor. O UCT clássico também prevê alocação de salas, mas na instituição estudada cada sala é previamente atribuída a uma turma. É necessário que as aulas sejam agrupadas de 2 em 2 horários.

O objetivo em questão é encontrar um horário (solução candidata) que satisfaça um conjunto de restrições *hard*, além de levar em conta as restrições *soft*, minimizando o número de suas violações. A função é de minimização, buscando diminuir o número de violações de cada restrição. A cada uma é atribuído um peso, de forma que restrições com maior peso penalizam mais a solução quando violadas. Para escolha das restrições foi realizado um estudo da literatura e uma entrevista com os coordenadores de curso da instituição de ensino alvo para definição dos pesos.

Foram utilizadas restrições a partir dos trabalhos *International Timetabling Competition 2007* (Di Gaspero et al. 2007), Qu (2009), Vieira (2011), Barbosa (2011), Silva (2014), Borges (2015) e Carvalho et al. (2016). A Tabela 1 mostra as restrições (e suas classificações).

Tabela 1 – Restrições usadas em FO

Restrição	Artigo							
	Barbosa - UD2 (2011)	Barbosa - UD3 (2011)	Carvalho <i>et al.</i> (2016)	Vieira (2011)	Silva (2014)	Borges (2015)	ITC 2007	Qu (2009)
Aulas	H	H	H	H	H	H	H	H
Conflitos	H	H	H	H	H	H	H	H
Disponibilidade	H	H	H	-	S	H	H	H
Mínimo de dias	S	-	-	S	-	S	S	S
Aulas isoladas	S	-	-	S	-	-	-	-
Currículo compacto	S	S	-	-	-	-	S	S
Janelas	-	S	S	-	S	-	S	S
Min e Máx de aulas por dia	-	S	-	-	S	-	-	S
Intervalo de trabalho	-	-	S	-	S	S	-	S
Aulas seguidas	-	-	S	-	-	-	-	S
Aulas seguidas de nível "difícil"	-	-	S	-	-	-	-	S
Aulas "difíceis" no último horário	-	-	S	-	-	-	-	-
Preferência do professor (disciplina)	-	-	-	S	-	S	-	-
Preferência do professor (nº de aulas)	-	-	-	-	S	S	-	-
Aulas aos sábados	-	-	-	-	-	-	-	-



Legenda: H = Hard Constraint, S = Soft Constraint

A restrição “Aulas” diz que uma disciplina não deve agendada para horários conflitantes. A restrição “Conflitos” diz que duas disciplinas do mesmo professor ou turma devem ser agendadas para horários diferentes. “Disponibilidade” afirma que uma aula não pode ser agendada num horário em que seu professor está indisponível. “Mínimo de dias” diz que as disciplinas devem estar distribuídas em um número mínimo de dias ao longo da semana. “Aulas isoladas” diz que aulas de um mesmo currículo devem ser agendadas em horários adjacentes. “Janelas” diz que não deve haver espaços entre aulas em um dia para uma turma. “Min e máx de aulas por dia” estabelece que deve haver uma quantidade mínima e máxima de aulas por dia para uma turma. “Intervalo de trabalho” é semelhante a “Janelas”, mas para os professores, e mensurada em dias, não horários. “Aulas seguidas” evita aulas seguidas do mesmo professor em uma turma. “Aulas seguidas de nível difícil” é similar, mas ao invés de ser para o mesmo professor, atua sobre disciplina que pertencem a um grupo de disciplinas “difíceis” (critério estabelecido pelo coordenador do curso). “Aulas de nível difícil no último horário” penaliza ocorrências de aulas de nível difícil no último horário de um dia. “Preferência do professor (disciplina)” busca evitar que professores lecionem disciplinas fora de sua zona de conforto. “Preferência do professor (número de aulas)” busca evitar que um professor dê mais aulas que o que ele considera desejável. “Aulas aos sábados” penaliza aulas que ocorrem durante o sábado.

As restrições “Mínimo de dias”, “Aulas isoladas” e “Min e Máx de aulas por dia” não fazem sentido na instituição alvo desse trabalho. Tendo uma agenda de 4 aulas por dia, com 5 dias na semana, o horário é pequeno demais para apresentar os problemas que essas restrições tentam solucionar. A restrição “Aulas aos sábados” não aparece em nenhum desses trabalhos, mas é necessária devido a uma particularidade da instituição. O sábado é considerado um dia letivo, e muitas vezes necessário para acomodar disciplinas. Todas as outras restrições que aparecem na tabela foram utilizadas.

Para a obtenção dos pesos, 6 coordenadores da instituição foram entrevistados. Cada restrição foi descrita e foi pedido que atribuissem um peso, de 1 a 5 (onde 1 significa “pouco importante” e 5 significa “muito importante”). O peso final foi obtido através da média aritmética. A Tabela 2 mostra o resultado. Note-se que as restrições “Aulas”, “Conflitos” e “Disponibilidade” foram utilizadas como restrições Hard no modelo.

Tabela 2 – Pesos das restrições

Restrição	Peso	Restrição	Peso	Restrição	Peso
Janelas	2	Aulas aos sábados	4,667	Aulas "difíceis" no último horário	2,333
Intervalo de trabalho	1,5	Aulas seguidas	4	Preferência do professor (disciplina)	3,167
Currículo compacto	3,5	Aulas seguidas de nível "difícil"	2,5	Preferência do professor (número de aulas)	1,667

4. Modelo Matemático

Nesta seção é apresentado o modelo de Programação Inteira desenvolvido para a resolução do problema. A modelagem das restrições *hard* é derivada diretamente a partir de [Da Silva et al. 2016], e as restrições *soft* são fortemente inspiradas em [Silva 2014], com alterações para o contexto do problema, uma vez este modelo também fará a alocação dos professores, o que não acontecia em [Silva 2014].

No modelo a solução é representada como uma matriz de quatro dimensões: Professor, Disciplina, Horário e dia. Uma célula $x_{p,d,i,j}$ significa que o professor p leciona a disciplina d no horário i no dia j . Os dados de entrada do modelo são: C , o conjunto de turmas; D , o conjunto de disciplinas; P , o conjunto de professores; $NumDias$, o número de dias letivos por semana; $NumHorarios$, o número de horários por dia; I , o conjunto de horários válidos; J , o conjunto de dias letivos; $HorariosÍmpares$, conjunto de horários de índice ímpar; π_i , o peso da i -ésima



restrição; $H_{d,c}$ identifica se o professor p pode lecionar a disciplina d ; $A_{p,i,j}$ indica se o professor p está disponível no horário i do dia d ; G_d identifica se d é uma disciplina difícil; $F_{p,d}$ identifica se o professor p possui preferência pela disciplina d ; Q_p é o número de horas desejadas pelo professor p ; N_p é número de horas de aula no contrato do professor p ; K_d é carga horária da disciplina d .

As variáveis de decisão modelo são: $x_{p,d,i,j}$, variável da solução, descrita acima; $r_{c,i,j}$ indica se a turma c tem aula no horário i do dia j ; $\alpha'_{k,c,i,j}$ indica se existe uma janela de tamanho k começando em i para a turma c no dia j ; α_c indica o número total de janelas para a turma c ; $w_{p,j}$ indica se o professor p dará aula no dia j ; $\beta'_{k,p,j}$ indica se existe um intervalo de trabalho de tamanho k começando no dia j para o professor p ; β_p indica o número de intervalos de trabalho para o professor p ; $g_{c,j}$ indica se a turma c possui aula no dia j ; γ_c , número de dias com aula da turma c ; δ_c , número de aulas que a turma c tem aos sábados; $\epsilon_{d,j}$, número de aulas seguidas da disciplina d no dia j ; $\theta_{j,c}$, número de aulas difíceis na turma c no dia j ; $\kappa_{c,j}$ indica se existe uma aula difícil no último horário da turma c no dia j ; λ_p , número de disciplina atribuídas a p que não são de sua preferência; μ_p , número de aulas que excedem a preferência de p ; $Lec_{p,d}$ indica se o professor p foi alocado para lecionar a disciplina d ; $Gem_{p,d,i,j}$ indica se uma aula geminada da disciplina d lecionada pelo professor p começa no horário i do dia j ; Pen_i indica o número de violações da restrição i .

Função objetivo: $\min z = \sum_i \pi_i Pen_i$

Sujeito a:

$$\sum_{d \in D} x_{p,d,i,j} \leq A_{p,i,j} \quad \forall i \in I, j \in J, p \in P \quad (1)$$

$$\sum_{p \in P} \sum_{d \in D} x_{p,d,i,j} H_{d,c} \leq 1 \quad \forall i \in I, j \in J, c \in C \quad (2)$$

$$x_{p,d,i,j} \leq h_{p,d} \quad \forall p \in P, d \in D, i \in I, j \in J \quad (3)$$

$$Lec_{p,d} \geq x_{p,d,i,j} \quad \forall p \in P, d \in D, i \in I, j \in J \quad (4.1)$$

$$Lec_{p,d} \leq \sum_{i \in I} \sum_{j \in J} x_{p,d,i,j} \quad \forall p \in P, d \in D \quad (4.2)$$

$$\sum_p Lec_{p,d} = O_d \quad \forall d \in D \quad (5)$$

$$\sum_{d \in D} \sum_{i \in I} \sum_{j \in J} x_{p,d,i,j} h_{p,d} \leq N_p \quad \forall p \in P \quad (6)$$

$$\sum_{p \in P} \sum_{i \in I} \sum_{j \in J} x_{p,d,i,j} = K_d O_d \quad \forall d \in D \quad (7)$$

$$gem_{p,d,i,j} \leq x_{p,d,i,j} \quad \forall p \in P, d \in D, i \in \text{HorariosPares}, j \in J \quad (8.1)$$

$$gem_{p,d,i,j} \leq x_{p,d,i+1,j} \quad \forall p \in P, d \in D, i \in \text{HorariosPares}, j \in J \quad (8.2)$$

$$gem_{p,d,i,j} \geq x_{p,d,i,j} + x_{p,d,i+1,j} - 1 \quad \forall p \in P, d \in D, i \in \text{HorariosPares}, j \in J \quad (8.3)$$

$$gem_{p,d,i+1,j} = 0 \quad \forall p \in P, d \in D, i \in \text{HorariosPares}, j \in J \quad (9.1)$$

$$\sum_{p \in P} \sum_{i \in I} \sum_{j \in J} gem_{p,d,i,j} = B_d \quad \forall d \in D \quad (9.2)$$

$$r_{c,i,j} = \sum_{p \in P} \sum_{d \in D} x_{p,d,i,j} H_{d,c} \quad \forall c \in C, i \in I, j \in J \quad (10.1)$$

$$r_{c,i,j} - \left(\sum_{l=1}^k r_{c,i+l,j} \right) + r_{c,i+k,j} - \alpha'_{k,c,i,j} \leq 1 \quad (10.2)$$

$$\forall c \in C, i \in I, j \in J, k \in [1, \text{NumHorarios} - 1 - i] \quad (10.3)$$



$$\alpha_c = \sum_{k=1}^{NumHorarios-2} \sum_{i \in I} \sum_{j \in J} \alpha'_{k,c,i,j} \quad \forall c \in C \quad (10.4)$$

$$Pen_1 = \sum_{c \in C} \alpha_c \quad (10.5)$$

$$w_{p,j} \geq x_{p,d,i,j} \quad \forall p \in P, d \in D, i \in I, j \in J \quad (11.1)$$

$$w_{p,j} \leq \sum_{i \in I} \sum_{d \in D} x_{p,d,i,j} \quad \forall p \in P, j \in J \quad (11.2)$$

$$w_{p,j} - \left(\sum_{i=1}^k w_{p,j+i} \right) + w_{p,j+k} - \beta'_{k,p,j} \leq 1 \quad \forall p \in P, j \in J, k \in [1, NumDias - 1 - j] \quad (11.3)$$

$$\beta_p = \sum_{k=1}^{NumDias-2} \sum_{j \in J} \beta'_{k,p,j} \quad \forall p \in P \quad (11.4)$$

$$Pen_2 = \sum_{p \in P} \beta_p \quad (11.5)$$

$$g_{c,j} \geq r_{c,i,j} \quad \forall c \in C, i \in I, j \in J \quad (12.1)$$

$$g_{c,j} \leq \sum_{i \in I} r_{c,i,j} \quad \forall c \in C, j \in J \quad (12.2)$$

$$\gamma_c = \sum_{j \in J} g_{c,j} \quad \forall c \in C \quad (12.3)$$

$$Pen_3 = \sum_{c \in C} \gamma_c \quad (12.4)$$

$$\delta_c = \sum_{p \in P} \sum_{d \in D} \sum_{i \in I} x_{p,d,i, NumDias} H_{d,c} \quad \forall c \in C \quad (13.1)$$

$$Pen_4 = \sum_{c \in C} \delta_c \quad (13.2)$$

$$\left(\sum_{p \in P} \sum_{i \in I} x_{p,d,i,j} \right) - \epsilon_{d,j} \leq 2 \quad \forall d \in D, j \in J \quad (14.1)$$

$$Pen_5 = \sum_{d \in D} \sum_{j \in J} \epsilon_{d,j} \quad (14.2)$$

$$\left(\sum_{i \in I} \sum_{d \in D} \sum_{p \in P} x_{p,d,i,j} G_d H_{d,c} \right) - \theta_{j,c} \leq 2 \quad \forall c \in C, j \in J \quad (15.1)$$

$$Pen_6 = \sum_{j \in J} \sum_{c \in C} \theta_{j,c} \quad (15.2)$$

$$\kappa_{c,j} = \sum_{p \in P} \sum_{d \in D} \sum_{i=NumHorarios-1}^{j=NumHorarios} x_{p,d,i,j} G_d H_{d,c} \quad \forall c \in C, j \in J \quad (16.1)$$

$$Pen_7 = \sum_{j \in J} \sum_{c \in C} \kappa_{c,j} \quad (16.2)$$

$$\lambda_p = \sum_{d \in D} \sum_{i \in I} \sum_{j \in J} x_{p,d,i,j} (1 - F_{p,d}) \quad \forall p \in P \quad (17.1)$$

$$Pen_8 = \sum_{p \in P} \lambda_p \quad (17.2)$$

$$\mu_p \geq \left(\sum_{d \in D} \sum_{i \in I} \sum_{j \in J} x_{p,d,i,j} \right) - Q_p \quad \forall p \in P \quad (18.1)$$

$$Pen_9 = \sum_p \mu_p \quad (18.2)$$

As restrições definem: (1) que um professor só pode ser alocado para um dia e horário em que estiver disponível; (2) que duas disciplinas da mesma turma não podem estar alocadas para o mesmo dia e horário; (3) que professores só podem ensinar disciplinas para as quais estão habilitados; (4) o impedimento para que a disciplina tenha mais de um professor associado; (5) que a disciplina só pode ser lecionada se for oferecida no semestre; (6) que os professores não podem lecionar horas além do seu contrato; (7) que disciplinas devem ter alocadas toda a sua carga horária; (8) que as aulas devem ser agrupadas de 2 em 2 (geminadas) até satisfazer o número necessário, por exemplo, aulas com 5 créditos precisam de 2 aulas geminadas e uma solitária; (9) o impedimento para que aula geminada comece num horário ímpar; (10) o número de janelas de disciplinas; (11) o número de intervalo de trabalho dos professores; (12) o número de aulas de cada turma; (13) o número de aulas aos sábados; (14) o número de aulas seguidas da mesma disciplina; (15) o número de aulas seguidas de nível difícil; (16) o número de aulas de nível difícil no último



horário; (17) o número de violações das preferências do professor em relação a disciplinas; (18) o número de violações das preferências do professor em número de aulas.

5. Experimentos Computacionais

Esta seção apresenta os experimentos computacionais realizados para calibração e comparação dos algoritmos. Foi reproduzido o mesmo ambiente utilizado em Da Silva *et al.* (2016), assim como os mesmos dados, que consistem em: 1392 alunos, 11 cursos, 2248 disciplinas, 68 professores e 24 horários e 33 períodos.

A execução do modelo matemático foi possível através da utilização do *solver* IBM CPLEX 12.5. O valor para a função objetivo obtido foi de 530.835, valor que será referido como FO ótima. O tempo total de execução foi de 42 minutos em um computador com processador *dual-core* Intel Core i5-5200U, com *clock* de 2.20 Ghz, sistema operacional Windows 10 64-bits e 8Gb de memória.

Todos os algoritmos foram reimplementados seguindo os fluxogramas e pseudocódigos disponíveis nos artigos originais. As estruturas de vizinhanças, apesar de tradicionais, foram implementadas de forma a melhor se adequar à representação do problema proposto por Da Silva *et al.* (2016). Por exemplo, o movimento de *swap* escolhe duas células de uma camada e as troca de lugar, um cuidado especial foi tomado para evitar que aulas geminadas fossem quebradas, de forma que os operadores sempre atuam em conjuntos de cardinalidade par.

A implementação foi feita em C++14 e compilados com MSVC14. As execuções foram distribuídas entre 40 computadores iguais, possuindo o processador AMD A8-5600K com *clock* de 3.6Ghz, 8.0 Gb de memória RAM e sistema operacional Windows 7 64-bits. Os dados foram os mesmos utilizados por Silva *et al.* (2016)

Para avaliação dos dados é considerado o RPD (*Relative Percentage Deviation*) médio. O RPD é calculado de acordo com a seguinte fórmula:

$$RPD = \frac{f_{method} - f_{best}}{f_{best}} \times 100$$

Onde f_{method} representa a função objetivo daquela solução e f_{best} representa a melhor função objetivo conhecida.

6. Calibração Dos Algoritmos

Essa seção apresenta os experimentos de calibração dos algoritmos, baseados em Montgomery (2006). Foi elaborado um conjunto de valores, definidos empiricamente, para cada parâmetro. Em seguida gerado um desenho fatorial completo a partir da combinação dos parâmetros e seus valores. Cada combinação foi testada em 10 rodadas, com um tempo limite de 180 segundos. A melhor configuração possui menor RPD médio.

6.1 Calibração do algoritmo AG-MaxHorário

Os parâmetros do AG-MaxHorário são taxa de mutação (TM), com valores 15 e 30; o número de indivíduos (NI), com valores 20 e 40; porcentagem de cruzamento (PC), com valores (20 e 30); operador de cruzamento (Xo), com valores CX, OX e PMX; número de mutações (NM), com valores 2 e 4; e número de indivíduos no torneio, com valores 2 e 4. Tem-se um total de 96 combinações possíveis. Como são muitas, a Tabela 3 mostra apenas as 5 melhores, ordenadas de forma decrescente a partir do RPD médio¹. A partir desses resultados é possível afirmar que a melhor configuração é a 10.

Tabela 3: Configurações do AG-MaxHorário

Nº	Parâmetros	RPD médio (%)	Coefficiente de variação (%)	Nº	Parâmetros	RPD médio (%)	Coefficiente de variação (%)
10	TM15.NI20.PC20	20.97418	1.799221	51	TM30.NI20.PC20	22.06367	4.787115

¹ Os resultados completos dos experimentos deste artigo estão disponíveis em: <https://goo.gl/2kpTod>



83	.XoPMX.NM2.NT4 TM30.NI40.PC20	21.79679	2.952942	69	.XoCX.NM4.NT2 TM30.NI20.PC30	22.09506	3.084353
71	.XoPMX.NM4.NT2 TM30.NI20.PC30 .XoPMX.NM4.NT2	21.83132	3.063377		.XoPMX.NM2.NT2		

6.2 Calibração do algoritmo WDJU

Os parâmetros do WDJU são o limite de estagnação (SL), com valores 10, 20 ou 30 iterações; e o fator de salto (JF), com valores de 0.1%, 0.2% e 0.5%. Tem-se um total de 9 combinações possíveis. A Tabela 4 mostra o valor do RPD médio para cada configuração. A partir desses resultados é possível afirmar que a melhor configuração é a 1.

Tabela 4: Configurações do WDJU

N°	Parâmetros	RPD médio (%)	Coefficiente de variação (%)	N°	Parâmetros	RPD médio (%)	Coefficiente de variação (%)
1	SL10.JF0.002	13.75563	2.275914	6	SL30.JF0.005	17.96615	4.032503
2	SL10.JF0.001	13.86864	2.359249	7	SL30.JF0.001	18.58470	3.782954
3	SL20.JF0.001	14.05704	2.099953	8	SL20.JF0.005	19.09023	3.142518
4	SL10.JF0.005	15.35696	2.902901	9	SL30.JF0.002	19.20640	4.943025
5	SL20.JF0.002	16.47162	2.805117				

6.3 Calibração do algoritmo HySST

Os parâmetros do HySST são o nível máximo (ML) com valores 5, 15 e 30 iterações, o começo da lista de barreiras (T0) com valores 1, 5 e 10, o incremento de cada nível (TS) com 5, 10 e 15, o número de iterações em cada aplicação das heurísticas *hill climbing* (IH) (valores 5, 10 e 15) e das mutacionais (IM) (3, 4 e 5). Tem-se um total de 243 combinações possíveis. Como são muitas, a Tabela 5 mostra apenas as 5 melhores, ordenadas de forma descendente a partir do RPD médio. A partir desses resultados é possível afirmar que a melhor configuração é a XX.

Tabela 5: Configurações do HySST

N°	Parâmetros	RPD médio (%)	Coefficiente de variação (%)	N°	Parâmetros	RPD médio (%)	Coefficiente de variação (%)
163	ML5.T01.TS10 .IH10.IM3	48.09297	2.847001	185	ML5.T01.TS5 .IH15.IM4	50.88108	3.307115
166	ML5.T01.TS10 .IH15.IM3	48.79317	5.236274	169	ML5.T01.TS10 .IH5.IM3	50.97533	3.536643
167	ML5.T01.TS10 .IH15.IM4	50.79955	3.120940				

6.4 Calibração do algoritmo SA-ILS

Os parâmetros do SA-ILS são o *timeout* para cada algoritmo em relação à fração do tempo total (FT) com valores de 100 e 200 e os parâmetros que essas meta-heurísticas normalmente adotam. Para o *Simulated Annealing* tem-se o alfa (*af*) com valores 0.97 e 0.935, a temperatura inicial (*t0*) com 0.5, 1 e 2, número de iterações (*SI*) com 100 e 150, o número de reaquecimentos (*SR*) com 250 e 500, e as chances de escolha das vizinhanças (*SC*), na ordem ES, EM, RS, RM, KM, com valores percentuais {25, 43, 20, 20, 10, 2} (conjunto 0) e {35, 43, 10, 5, 7} (conjunto 1).

Para o *Iterated Local Search* tem-se o número de iterações por ciclo (*II*) com valores 10 e 30, o número máximo de perturbações (*PM*) com 10, 15 e 30, e o número inicial de perturbações (*PO*) com 1 e 3. Tem-se um total de 1152 configurações possíveis. Como são muitas, a Tabela 6 mostra apenas as 5 melhores, ordenadas de forma descendente a partir do RPD médio. A partir desses resultados é possível afirmar que a melhor configuração é a 752.



Tabela 6: Configurações do SA-ILS

Nº	Parâmetros	RPD médio (%)	Coefficiente de variação (%)
752	FT200.af0.935.t01.SI150.SR500.SCO.II30.PM10.P03	10.70685	1.9412073
1059	FT200.af0.97.t02.SI100.SR250.SCO.II10.PM15.P01	10.85441	2.9358271
941	FT200.af0.97.t00.5.SI150.SR500.SCO.II10.PM30.P01	10.95489	2.0380453
914	FT200.af0.97.t00.5.SI150.SR250.SCO.II10.PM10.P03	10.97373	2.4989780
506	FT100.af0.97.t02.SI100.SR500.SCO.II10.PM10.P03	10.99884	1.5944709

7. Experimentos de Comparação dos Algoritmos

Tendo sido constatado quais os parâmetros para melhor performance dos algoritmos, foi realizada uma nova rodada de execuções. Desta vez, cada algoritmo foi executado 30 vezes, sendo cada execução com um tempo limite de 180 segundos. A Tabela 7 mostra os resultados.

Tabela : Comparação dos Algoritmos

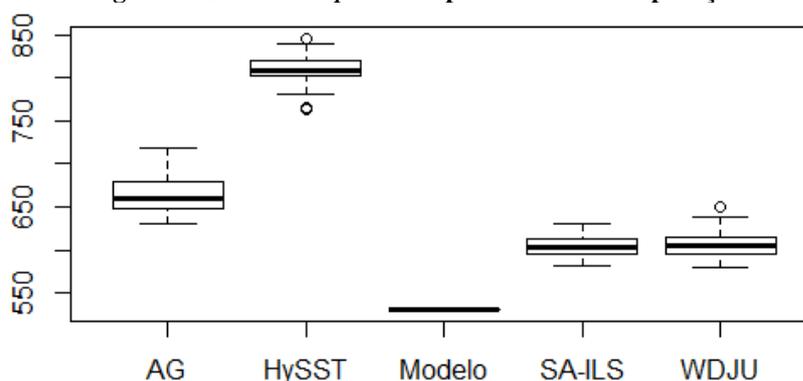
Nº	AG-MaxHorário		HySST		SA-ILS		WDJU	
	FO	RPD (%)	FO	RPD (%)	FO	RPD (%)	FO	RPD (%)
1	662.174	24.741963	801.846	51.053717	604.005	13.783944	615.172	15.887611
2	685.675	29.169139	808.513	52.309663	600.838	13.187337	604.338	13.846676
3	687.175	29.451713	765.012	44.114838	598.171	12.684921	597.671	12.590730
4	662.674	24.836154	815.847	53.691260	622.339	17.237748	619.172	16.641141
5	642.673	21.068317	817.680	54.036565	603.505	13.689753	605.838	14.129249
6	629.839	18.650616	809.847	52.560965	617.172	16.264376	594.004	11.899931
7	652.007	22.826679	839.349	58.118624	599.671	12.967495	593.838	11.868660
8	632.839	19.215764	763.677	43.863347	594.171	11.931391	582.837	9.796264
9	647.840	22.041689	819.180	54.319139	584.504	10.110298	599.671	12.967495
10	667.341	25.715335	823.180	55.072669	613.005	15.479386	633.506	19.341415
11	647.840	22.041689	805.514	51.744704	617.172	16.264376	611.005	15.102621
12	655.507	23.486017	803.013	51.273560	595.671	12.213965	606.838	14.317632
13	680.008	28.101576	818.513	54.193488	581.337	9.513691	595.671	12.213965
14	660.007	24.333738	830.182	56.391722	603.171	13.626833	615.005	15.856151
15	638.173	20.220596	809.513	52.498046	602.838	13.564102	650.673	22.575377
16	641.673	20.879934	821.847	54.821555	602.338	13.469911	604.338	13.846676
17	684.009	28.855294	804.347	51.524862	588.004	10.769637	613.672	15.605037
18	718.343	35.323217	811.513	52.874810	602.338	13.469911	605.838	14.129249
19	669.508	26.123560	805.847	51.807435	610.005	14.914239	605.838	14.129249
20	658.340	24.019705	764.678	44.051918	604.338	13.846676	579.837	9.231117
21	704.010	32.623131	805.514	51.744704	611.672	15.228272	630.006	18.682076
22	672.508	26.688707	790.679	48.950050	614.505	15.761960	608.338	14.600205
23	689.175	29.828478	815.347	53.597069	612.505	15.385195	604.838	13.940867
24	633.006	19.247224	780.845	47.097497	596.171	12.308156	628.839	18.462234
25	672.675	26.720167	799.513	50.614221	599.171	12.873303	610.838	15.071161
26	647.840	22.041689	792.179	49.232624	595.671	12.213965	596.171	12.308156
27	704.343	32.685863	846.015	59.374382	630.006	18.682076	586.337	10.455603
28	651.507	22.732487	819.681	54.413518	590.004	11.146401	604.338	13.846676
29	656.340	23.642940	829.680	56.297154	599.338	12.904763	584.004	10.016107
30	647.673	22.010229	807.514	52.121469	614.005	15.667769	637.839	20.157676

Inicialmente foi proposta a utilização do teste ANOVA (Fisher, 1921), com o objetivo de constatar que os diferentes algoritmos produzem resultados com diferença estatística significativa. Esse teste possui as seguintes exigências: as amostras são aleatórias e independentes, as populações têm distribuição normal e as variâncias populacionais são equivalentes. A metodologia aplicada satisfaz a primeira exigência. Porém, a segunda exigência precisa ser verificada, e o teste de Shapiro-Wilk (Shapiro e Wilk, 1965) foi escolhido para isso. Esse teste tem como hipótese nula a de que a população tem distribuição normal. O resultado desse teste foi $W = 0.82052$ e $p\text{-value} = 7.69e-11$, que comprova que a população não tem distribuição normal e, portanto, não se pode aplicar um teste paramétrico a ela. Como alternativa, é aplicado o teste não-paramétrico de Kruskal-Wallis (Kruskal e Wallis, 1952). Esse teste não tem as exigências do ANOVA, e é comumente aplicado quando elas são violadas. Sua hipótese nula é de que as medianas populacionais são iguais. Um resultado significativo indica que pelo menos uma amostra é significativa sobre uma outra



amostra. O resultado foi: $\chi^2 = 100.74$, $df = 4$, $p\text{-value} = 2.2e-16$. A Figura 1 apresenta um gráfico de *boxplot* relativo aos dados do experimento. Cada *box* representa os valores mínimo, máximo e médio das FOs de um algoritmo. É possível constatar que existe significância estatística dentre os dados, o que permite dizer com confiança que SA-ILS e WDJU são os melhores algoritmos (a Figura 1 também evidencia que não existe significância estatística entre esses dois). A melhor FO obtida pelo SA-ILS é 581.337, e a para o WDJU é 579.837.

Figura 1: Gráfico *boxplot* do experimento de comparação



Em nenhuma rodada o resultado ótimo foi encontrado por alguma das heurísticas. Os resultados mostram os seguintes *gap* médios em relação a solução ótima: SA-ILS (13.70), WDJU (14.45), AG-MaxHorario (24.98), HySST (52.12). Em função disso, um novo experimento foi realizado, onde cada algoritmo foi executado 10 vezes com 1 hora de tempo limite. No entanto não foi constatada uma melhoria significativa. Embora o WDJU tenha alcançado uma FO melhor (568.170), a maior FO obtida de todas as execuções nesse experimento, ainda não alcança o resultado ótimo.

8. Conclusão

Esse trabalho abordou o problema de *University Course Timetabling*, apresentando uma FO baseada na violação de restrições *soft*. Foi construído um modelo matemático para sua resolução e realizado uma comparação com quatro algoritmos da literatura voltados para resolução deste problema. Para tal foram executados experimentos computacionais e a avaliação de seus resultados através de testes estatísticos. Foi constatado que a função objetivo obtida pelo modelo matemático foi de 530.835, enquanto a melhor FO obtida por uma heurística foi a do SA-ILS, com 587.670.

Como o *gap* médio do SA-ILS é pequeno (13.7%) e a diferença entre o tempo de execução do modelo e do algoritmo é grande (840%), o SA-ILS é mais viável em um cenário em que vários *timetables* tenham que ser gerados. Se for necessária a geração de apenas um *timetable*, o tempo de execução do modelo se torna insignificante, uma vez que esse processo normalmente é executado apenas uma vez por semestre.

REFERÊNCIAS

- Abuhamdah, A., Ayob, M., Kendall, G., e Sabar, N. R. (2014). Population based Local Search for university course timetabling problems. *Applied intelligence*, vol. 40(1).
- Aubin, J., e Ferland, J. A. (1989). A large scale timetabling problem. *Computers & Operations Research*, vol. 16(1), pp. 67-77.
- Barbosa, S. H.; Souza, S. R. (2011). Resolução do problema de programação de cursos universitários baseada em currículos via uma meta-heurística híbrida GRASP-ILS-relaxado.



Proceedings of XLIII SBPO XLIII Simpósio Brasileiro de Pesquisa Operacional. SOBRAPO, Ubatuba, p. 1.

Borges, A. et al. (2015). Estudo de caso de programação inteira para automação de grade de horários do departamento de estatística da Universidade Federal de Pernambuco. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 3, n. 1.

Carrasco, M. P. e Pato, M. V. (2000). A multiobjective genetic algorithm for the class/teacher timetabling problem. In *Practice and Theory of Automated Timetabling III*, pp. 3-17. Editora Springer Berlin Heidelberg.

Carvalho, A. S. et al. (2016). Simulated Annealing aplicado ao problema de programação de horários do CCA-UFES. *Blucher Marine Engineering Proceedings*, v. 2, n. 1, p. 341-352.

Chahal, N. e De Werra, D. (1989). An interactive system for constructing timetables on a PC. *European Journal of Operational Research*, v. 40, n. 1, p. 32-37.

Daskalaki, S. e Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, v. 160, n. 1, p. 106-120;

Da Silva, I. L., Machado, P. H. M., e Campos, S. C (2016). Algoritmo genético para resolução do problema de university course timetabling para instituições de ensino superior privadas: Um estudo de caso na FAGOC. *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional*, pp. 698-709.

De Werra, D. (1985). An Introduction to Timetabling. *European Journal of Operational Research*, vol. 19, pp. 151-162.

Di Gaspero, L; McCollum, B; Schaerf, A. (2007). The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). *Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1. 0*, Queen's University, Belfast, United Kingdom.

Eiselt, H. A. e Laporte, G. (1987). Combinatorial optimization problems with soft and hard requirements. *Journal of the Operational Research Society*, pp. 785-795.

Even, S., Itai, A., e Shamir, A. (1976). On the complexity of timetabling and multicommodity flow problems. *SIAM Journal of Computation*, vol. 5(4), pp. 691-703.

Ferland, J. A. e Roy, S. (1985). Timetabling problem for university as assignment of activities to resources. *Computers & operations research*, vol. 12(2), pp. 207-218.

Fisher, R. A. (1921). On the probable error of a coefficient of correlation deduced from a small sample. *Metron*, vol. 1, pp. 3-32.

Fonseca, G. (2013). Métodos de Busca Heurística para Problemas de Programação de Horários Modelados em XHSTT. *Tese de Doutorado*. Universidade Federal de Ouro Preto.

Gotlieb, C. C. (1963). The construction of class-teacher timetables. In *C. M. Popplewell, IFIP Congress 62*, pp. 73-77. North-Holland.

Hertz, A. (1992). Finding a feasible course schedule using tabu search. *Discrete Applied Mathematics*, vol. 35(3), pp. 255-270.



Jardim, A; Semaan, G; Penna, P. (2016). Uma heurística para o problema de programação de horários: um estudo de caso. *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional*, pp. 777-788.

Kheiri, A; Ozcan, E; Parkes, A J. (2012). Hyper-heuristic search strategies and timetabling. *In: Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012)*. p. 497-499.

Kruskal, W. H., e Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, vol. 47(260), pp. 583-621.

Levitin, A e Mukherjee, S. (2003). *Introduction to the design & analysis of algorithms*. Reading: Addison-Wesley.

Qu, R et al. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of scheduling*, v. 12, n. 1, p. 55-89.

Montgomery, D. C. (2006). *Design and analysis of experiments*. 7ed. New York: Wiley.

Paechter, B., Cumming, A., Luchian, H. e Petriuc, M. (1994). Two solutions to the general timetable problem using evolutionary methods. *In Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference*, pp. 300-305. IEEE.

Post, G. et al. The third international timetabling competition. *Annals of Operations Research*, v. 239, n. 1, p. 69-75, 2016.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, vol. 13(2), pp. 87-127.

Schmidt, G. e Strohleim, T. (1979). Timetable construction – an annotated bibliography. *The Computer Journal*, vol. 23(43), pp. 07-316.

Shapiro, S. S., e Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, vol. 52(3-4), pp. 591-611.

Silva, A. R. V. (2014). Uma formulação matemática para o problema da alocação de horários em um curso universitário: um estudo de caso. *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional (SBPO 2014)*, pp. 2704-2715.

Tripathy, A. (1984). School timetabling—a case in large binary integer linear programming. *Management science*, vol. 30(12), pp. 1473-1489.

Vieira, F e Macedo, H. (2011). Sistema de alocação de horários de cursos universitários: um estudo de caso no departamento de computação da Universidade Federal de Sergipe. *Scientia Plena*, v. 7, n. 3.

Welsh, J. A. e Powell, M. B. (1967). An upper bound to the chromatic number of a graph and its applications to timetabling problems. *The Computer Journal*, vol.10, pp.85-86.

Wilke, P e Killer, H. Walk down jump up-a new hybrid algorithm for time tabling problems. *PATAT 2010*, p. 440, 2010.