

# Programação de Horários em Instituições Educativas: Formulações e Algoritmos

Haroldo Gambini Santos<sup>1</sup>  
Marcos Jamilson Freitas Souza<sup>2</sup>

29 de agosto de 2007

<sup>1</sup>UFRJ-Universidade Federal do Rio de Janeiro

<sup>2</sup>UFOP: Universidade Federal de Ouro Preto

# Lista de Siglas

BL	Busca Local
ITP	International Timetabling Competition
PAAS	Problema de Alocação de Aulas a Salas
PLIM	Programação Linear Inteira Mista
PLIM	Programação Linear Inteira Mista
PMO	Problema com Múltiplos Objetivos
PPCPM	Problema de Programação de Cursos Pós-Matrícula
PPE	Problema de Programação de Exames
PPH	Problemas de Programação de Horários
PPHE	Problema de Programação de Horários em Escolas
PPHEU	Problemas de Programação de Horários de Exames em Universidades
PPHU	Problema de Programação de Horários em Universidades
PPT	Problema de Programação de Horários Professores $\times$ Turmas
PPTC	Problema de Programação de Horários Professores $\times$ Turmas com restrições de Capacidade

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Programação de Horários</b>	<b>4</b>
2.1	Otimização e Factibilidade . . . . .	5
2.2	Competição Internacional de Programação de Horários . . . . .	7
2.3	Programação de Horários em Escolas . . . . .	7
2.3.1	O Problema de Programação de Horários Professores × Turmas com restrições de Capacidade . . . . .	10
2.4	Programação de Horários de Cursos em Universidades . . . . .	13
2.5	O Problema de Alocação de Salas . . . . .	14
2.6	Programação de Horários de Exames em Universidades . . . . .	15
<b>3</b>	<b>Heurísticas para Prog. de Horários</b>	<b>18</b>
3.1	Heurísticas Construtivas . . . . .	18
3.2	Busca Local . . . . .	20
3.3	Procedimento Intraturmas-Interturmas . . . . .	21
3.3.1	Procedimento Intraturmas . . . . .	23
3.3.2	Procedimento Interturmas . . . . .	27
3.3.3	Procedimento II . . . . .	28
3.4	Metaheurísticas . . . . .	31
3.4.1	Busca Tabu . . . . .	31
3.4.1.1	Um Algoritmo de Busca Tabu para o Problema de Programação de Horários em Escolas . . . . .	32
3.4.2	<i>Simulated Annealing</i> . . . . .	37
3.4.3	Algoritmos Evolutivos . . . . .	37
<b>4</b>	<b>Produção de Quadros de Horários Ótimos</b>	<b>39</b>
4.1	Cortes . . . . .	40
4.2	Formulação Estendida com Geração de Colunas e Cortes . . . . .	43
4.2.1	Geração de Colunas . . . . .	44
4.2.2	Geração de Cortes . . . . .	45
4.2.3	O Algoritmo de Geração de Colunas e Cortes . . . . .	48

# Capítulo 1

## Introdução

Neste trabalho, serão considerados alguns Problemas de Programação de Horários (*Timetabling*). Wren[78] define esses problemas como:

“A alocação, sujeita a restrições, de recursos a objetos colocados no espaço e no tempo, de modo a satisfazer, tanto quanto possível, um conjunto de objetivos desejáveis”

Estes problemas têm motivado diversas pesquisas no campo da Otimização Combinatória e Inteligência Artificial. Neste trabalho, serão apresentados os principais tipos de Problemas de Programação de Horários (PPHs) encontrados em instituições de ensino. Apesar de que cada um desses problemas apresenta características únicas, os mesmos compartilham uma estrutura bastante similar. Desse modo, a experiência adquirida na resolução de cada um desses é, geralmente, facilmente generalizável para a resolução de outros. De modo mais abrangente, a adaptação eficiente de técnicas genéricas como Programação Linear Inteira Mista (PLIM) e Metaheurísticas para PPHs também produz resultados e conclusões com reconhecida utilidade na resolução de problemas fora escopo dos considerados neste trabalho.

Neste trabalho serão revisadas algumas das principais técnicas de resolução de PPHs em instituições de ensino. Preferencialmente, foram selecionadas técnicas genéricas. A apresentação de algumas das técnicas inclui exemplos de suas aplicações em PPHs, nesse caso, com ênfase no Problema de Programação de Horários em Escolas (PPHE) .

## Capítulo 2

# Problemas de Programação de Horários em Instituições de Ensino

Problemas de Produção de Quadros Horários em Instituições de Ensino têm sido bastante estudados nas últimas décadas. Conferências regulares discutem o tema no meio científico [17, 15, 16]. Algumas das razões para esse interesse são:

1. **dificuldade de resolução:** encontrar um quadro de horários que satisfaça todos os interesses envolvidos é uma tarefa difícil, ademais, frequentemente, a simples construção de um quadro de horários válido já é uma tarefa complicada;
2. **importância prática:** como será detalhado neste trabalho, a confecção de um bom quadro de horários pode melhorar a satisfação do corpo docente e permitir que a instituição de ensino seja mais eficiente na gestão de seus recursos, além do mais, a programação adequada das atividades letivas permite um melhor desempenho dos alunos;
3. **importância teórica:** problemas deste tipo representam exemplos clássicos de problemas “fáceis de serem explicados e difíceis de serem resolvidos”, de fato, os problemas apresentados neste trabalho constituem parte dos problemas ditos NP-Completo ou NP-Difíceis[38], progressos na solução de problemas desse tipo são um dos grandes objetivos das pesquisas correntes em computação, matemática e pesquisa operacional.

Uma maneira de lidar com (1.) é o aproveitamento de quadros de horários antigos: a cada ano, reutiliza-se o quadro de horários anterior, realizando-se apenas mudanças mínimas para acomodar as alterações requisitadas. Essa abordagem, entretanto, é facilmente comprometida com alterações no corpo docente ou no currículo, por exemplo.

Neste trabalho os PPHs considerados são divididos em três tipos: Problemas Programação de Horários em Escolas (PPHE), apresentados na seção 2.3, Problemas Programação de Horários de Cursos em Universidades (PPHU), na

seção 2.4 e Problemas de Programação de Horários de Exames em Universidades (PPHEU), na seção 2.6. Essa distinção não é estrita, conforme discutido na literatura[62], de modo que certos autores alertam para uma certa confusão nos nomes empregados na literatura. Em especial, são encontrados vários PPHUs e PPHEs bastante similares. Assim, os problemas considerados neste trabalho correspondem a casos típicos selecionados pelos autores.

## 2.1 Otimização e Factibilidade

A dificuldade de produção de bons quadros de horários advém, basicamente, de dois pontos, como especificado em [35]. Primeiramente, tem-se a dificuldade do tomador de decisões em definir claramente quais as características desejadas no quadro de horários. Por exemplo, como as aulas de um professor devem se alocadas para que o mesmo fique satisfeito? Como as aulas de matemática devem ser distribuídas no decorrer de uma semana para que o aproveitamento dos alunos seja o máximo possível? O que constitui um quadro de horários eficiente para uma dada instituição de ensino? Estes problemas são geralmente problemas com múltiplos objetivos. Não raramente esses objetivos são conflitantes, sendo que a definição de cada um desses, bem como a sua importância relativa, é uma tarefa do tomador de decisões e do analista.

A outra dificuldade com os PPHs é puramente matemática: problemas desse tipo são problemas de associação, de natureza combinatória. Os métodos computacionais exatos de resolução existentes para PPHs geralmente trabalham com enumeração do espaço de soluções, mesmo que implicitamente, limitando severamente o tamanho das instâncias que podem ser resolvidas em tempos computacionais razoáveis.

Considerando a modelagem de um PPH, inicialmente devem ser definidos os *requisitos*. Esses requisitos devem ser expressos em termos de restrições do problema, que usualmente são de dois tipos:

- **restrições fortes:** restrições desse tipo devem ser satisfeitas a qualquer custo, visto que não é possível a implementação de um quadro de horários que não as satisfaça. A restrição mais comum desse tipo é a não ocorrência de conflitos, como exemplo, considere a situação impraticável de uma turma assistir a duas aulas ao mesmo tempo; restrições físicas também são desse tipo - a decisão de em qual sala de aula será dada um curso deve considerar que a sala escolhida terá capacidade suficiente para comportar a turma; desse modo, as restrições fortes determinam o espaço de busca que será considerado: somente soluções que respeitam todas as restrições desse tipo são consideradas *factíveis*; se o problema considerado consiste em obter uma solução factível qualquer, então esse denomina-se *problema de busca*;
- **restrições fracas:** restrições desse tipo são aquelas cuja satisfação é desejável, mas caso não seja possível respeitá-las, pode-se, ainda assim, implementar o quadro de horários; restrições fracas comuns são, por exemplo, preferências de professores - um professor pode preferir dar aulas nos períodos da manhã, mas eventualmente será alocado no período da tarde caso isso não seja possível ou caso interesses mais importantes estejam envolvidos; no caso de programação de exames (seção 2.6) uma restrição

fracas comum é o espaçamento dos exames para os alunos - com o objetivo de minimizar o estresse, idealmente, cada exame de cada aluno deve ser precedido por um certo número de períodos de folga; o atendimento das restrições fracas é a medida de qualidade de um quadro de horários; se o problema considerado consiste em encontrar uma solução que minimize a violação das restrições fracas, tem-se então um *problema de otimização*.

O conjunto de requerimentos, bem como o seu particionamento em restrições fortes e fracas é bastante dependente da instituição de ensino e do sistema educacional de cada localidade. Em geral os requerimentos podem ser divididos em três classes:

- **organizacionais**: relativos à instituição de ensino, nesse caso são incluídos os requerimentos que tratam da gestão de recursos, bem como do atendimento da legislação vigente, como:
  - alocação de salas: a realização de determinadas atividades letivas exige salas com uma capacidade especificada, adicionalmente, alguns cursos requerem laboratórios ou ginásios.
- **pedagógicos**: pedidos importantes para o bom aproveitamento das aulas, como:
  - duração das aulas: para a maioria das disciplinas é preferível uma distribuição semanal uniforme das atividades ao invés da concentração de todas as aulas em um único dia, o que tornaria a atividade cansativa;
  - intervalos: determinados cursos envolvem tarefas de casa, de modo que alguns dias de intervalo entre as aulas facilitam o acompanhamento dos alunos, no caso da Programação de Exames (seção 2.6), o oferecimento de intervalo de descanso para os alunos é um dos principais requerimentos.
- **pessoais**: requisitados de acordo com as preferências e necessidades pessoais dos membros do corpo docente, alguns requerimentos desse tipo são:
  - preferências de horários: alguns professores preferem os primeiros ou últimos períodos;
  - deslocamento: caso um professor leccione em vários campi geograficamente distantes, alocações consecutivas em diferentes campi devem ser realizadas prevendo um tempo de deslocamento entre esses;

Desse modo, os PPHs em instituições de ensino aparecem naturalmente como Problemas com Múltiplos Objetivos (PMOs). Uma possível abordagem para esses problemas é a utilização de métodos de otimização específicos para PMOs, como o empregado em [20]. Nesses métodos, constrói-se um conjunto de soluções não dominadas[7]. Com esse conjunto disponível, o decisor pode selecionar a solução que apresenta a melhor relação entre os objetivos, de acordo com seus critérios. Essa estratégia, no entanto, não é prática quando o número de objetivos é muito grande, visto que o conjunto de soluções não dominadas pode crescer imensamente. De fato, na abordagem para programação de horários de

Carrasco & Pato[20], por exemplo, os diferentes objetivos são agrupados em apenas dois.

Caso o decisor conheça claramente a relação de importância entre os diferentes objetivos, esses podem ser combinados em uma única função objetivo que consiste em uma combinação linear com pesos. Considere um problema com  $p$  funções objetivo:  $f_1(x), f_2(x), \dots, f_p(x)$ , sendo que  $w_k \in \mathbb{R}^+$  indica a importância da  $k$ -ésima função objetivo. Assim, pode-se definir uma única função objetivo  $f(x)$  como:

$$f(x) = \sum_{k=1}^p w_k \cdot f_k(x)$$

Nos problemas considerados no restante deste texto será suposto que os pesos  $w_k$  são conhecidos, sendo informados como dados de entrada do problema.

## 2.2 Competição Internacional de Programação de Horários

Em 2002, a Metaheuristics Network[2] organizou uma competição, a *International Timetabling Competition (ITP)*[55] de métodos de produção automatizada de quadros de horários. Nesta competição os proponentes deveriam desenvolver programas para resolver uma versão simplificada do problema de programação de cursos em universidades. O programa que produzisse a melhor solução em um tempo estipulado seria o vencedor. Essa atividade foi um marco bastante importante para a pesquisa na área, visto que, tradicionalmente, poucos autores realizavam experimentos computacionais comparando diferentes métodos de solução. Até então, era bastante comum a ocorrência de publicações que comparavam os resultados computacionais com a solução produzida manualmente para o problema, um critério bastante impreciso. Essa situação ocorria principalmente devido à já mencionada diversidade de necessidades das diferentes instituições de ensino. No caso do modelo considerado na ITP, sua qualidade não é ser genérico suficiente para modelar todos os casos existentes, mas sim incorporar algumas características que são comuns à grande maioria dos problemas de programação de cursos encontrados em universidades.

Recentemente, foi lançada uma nova edição da competição [1], nessa edição são consideradas três categorias: Programação de Exames, Programação de Cursos Pós-Matrículas e Programação de Cursos baseada em Currículos.

## 2.3 Programação de Horários em Escolas

O Problema de Programação de Horários em Escolas (PPHE), também conhecido na literatura como Problema de Programação de Horários Professor  $\times$  Turma (PPT) é um problema clássico[41, 28, 38] de otimização combinatória sendo provavelmente o problema mais conhecido dos problemas de programação de horários. No Brasil, o problema corresponde ao problema de programação de horários encontrado em escolas (5<sup>a</sup> a 8<sup>a</sup> séries) de ensino fundamental e ensino médio. Nesse problema, cada professor deve lecionar um determinado número de aulas para cada turma em um conjunto de períodos. As alocações devem

ser feitas considerando-se a não existência de conflitos, ou seja, cada professor e turma deve se envolver com, no máximo, uma atividade letiva por período. Esse problema, inicialmente abordado nos trabalhos pioneiros de Gotlieb [41, 28], tem sido bastante estudado nas últimas décadas. Revisões de literatura podem ser encontradas em [62, 74].

No PPT consideram-se os seguintes dados de entrada:

- $P$ : conjunto de professores, com elementos  $p \in P$  numerados como  $1, \dots, |P|$ ;
- $T$ : conjunto de turmas, com elementos  $t \in T$  numerados como  $1, \dots, |T|$ ;
- $D$ : conjunto de dias letivos, com elementos  $d \in D$  numerados como  $1, \dots, |D|$ ;
- $H$ : conjunto de períodos letivos por dia, com elementos  $h \in H$  numerados como  $1, \dots, |H|$ ;
- $\tilde{R}_{|P| \times |T|}$ : matriz de requerimentos, onde  $\tilde{r}_{pt}$  indica quantas aulas o professor  $p$  deve lecionar para a turma  $t$ ;
- $\tilde{P}_{|P| \times |D| \times |H|}$ : matriz de disponibilidade de professores, onde  $\tilde{p}_{pdh} = 1$  indica que o professor  $p$  está disponível no dia  $d$  no período  $h$ ,  $\tilde{p}_{pdh} = 0$  caso contrário.

Adicionalmente, define-se a seguinte variável de decisão:

$$x_{ptdh} = \begin{cases} 1 & \text{professor } p \text{ lecionando para a turma } t \text{ em um dia } d \text{ e horário } h \\ 0 & \text{caso contrário} \end{cases}$$

Com esses dados pode-se definir formalmente o problema de busca PPT como:

$$\text{encontre } x_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (2.1)$$

sujeito a

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \quad (2.2)$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H \quad (2.3)$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{p}_{pdh} \quad \forall p \in P, d \in D, h \in H \quad (2.4)$$

$$x_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (2.5)$$

As restrições 2.2 asseguram o cumprimento da carga horária, 2.3 asseguram a não ocorrência de conflitos em horários para turmas e 2.4 asseguram o respeito a disponibilidade dos professores e a não existência de conflitos em horários para professores.

Enquanto casos especiais do PPT podem ser resolvidos em tempo polinomial [75], a simples consideração de restrições de disponibilidade, ou seja, o caso de professores com  $\sum_{d \in D} \sum_{h \in H} \tilde{p}_{pdh} < |D| \cdot |H|$ , coloca o PPT na classe de

problemas combinatórios NP-Completo [36]. Em [26] é discutida a intratabilidade computacional do PPT devido a vários fatores cuja ocorrência é bastante comum em problemas reais, como pedidos relativos à duração das aulas [70].

Como mencionado anteriormente, um grande número de requerimentos é considerado em PPHs encontrados na prática. Considerando o PPHE, apesar dos diferentes modelos encontrados na literatura, alguns requerimentos são bastante freqüentes e serão apresentados nos parágrafos seguintes.

Considerando-se requerimentos relativos ao **corpo docente**, geralmente devem ser evitados “buracos” em dias letivos dos professores, ou seja, períodos de inatividade entre períodos com atividade. Dias com poucas alocações isoladas também devem ser evitados, favorecendo a construção de quadros de horários compactos para professores, de modo que suas aulas concentrem-se em poucos dias da semana. Essas questões foram consideradas em [27, 61, 6, 25, 19, 32, 44, 8, 31, 56, 33]. Deve-se observar que restrições desse tipo, desejáveis na maioria dos casos, são essenciais quando são considerados professores que trabalham em mais de uma instituição, situação bastante comum na realidade brasileira. A compactidade de horários para turmas geralmente não é considerada como um critério para avaliação da qualidade de um quadro de horários visto que, na maioria das vezes, as turmas devem ter todo o seu horário preenchido com aulas.

Do ponto de vista **pedagógico**, restrições importantes tratam da distribuição das aulas de um professor  $\times$  turma no decorrer da semana: dependendo do assunto lecionado, aulas com maior ou menor duração em cada dia podem ser necessárias. Em [33, 6], por exemplo, a distribuição das aulas de um professor  $\times$  turma no decorrer da semana é especificada por padrões de blocos com tamanhos variados, já em [27], de um modo mais específico, determina-se quais aulas devem durar um ou dois períodos (aulas geminadas). Em [61, 67, 66, 65], especifica-se um número mínimo de aulas geminadas para que as certas atividades letivas sejam realizadas. De qualquer modo a existência de dias letivos com muitas aulas de um mesmo assunto para uma turma deve ser evitada, seja através da penalização dos chamados “agrupamentos”, como em [25], seja através da remoção de horários com essas características do espaço de busca, como em [11]. Alguns trabalhos também consideram a alocação uniforme de diferentes conteúdos em um mesmo dia, evitando na turma a sobrecarga de disciplinas consideradas “difíceis”, ou de assuntos relacionados [27], como matemática e física. De modo semelhante, [33] divide as disciplinas em “principais”, no caso de ensino da língua nativa e matemática, e “subsidiárias”, no caso biologia.

Várias questões logísticas também podem ser consideradas pela **instituição de ensino**. A ocorrência de turmas com tamanhos variados pode requerer que o programador de horários decida também sobre a alocação de salas para as atividades letivas, como em [33]. Em instituições de grande porte, onde os professores lecionam aulas em várias localidades, possivelmente distantes, determinadas seqüências de aulas em um determinado dia podem ser infactíveis, dada a impossibilidade de deslocamento do professor entre as salas de aula em um tempo viável.

### 2.3.1 O Problema de Programação de Horários Professores $\times$ Turmas com restrições de Compacidade

O Problema de Programação de Horários Professores  $\times$  Turmas com restrições de Compacidade é uma generalização do PPT que considera aspectos práticos importantes discutidos na seção anterior. Esse problema, apresentado inicialmente em [65] e posteriormente considerado em [67, 60], é aqui denotado por PPTC. Os requisitos adicionais tratam da satisfação dos professores com o quadro de horários, especificamente a compacidade de horários, e do atendimento de requisitos pedagógicos, especificamente a duração das aulas. Com relação às restrições fortes, além das consideradas no PPT, uma nova restrição é incluída: especifica-se um limite máximo diário para as aulas que um professor pode lecionar para cada turma.

São consideradas as seguintes restrições fracas, cujo não atendimento será medido na função objetivo:

1. compacidade de horários: o quadro de horários de um professor deve se estender por um número mínimo possível de dias;
2. aulas geminadas: alguns professores requerem um número mínimo de aulas geminadas, em determinadas turmas, para a execução dos conteúdos;
3. “buracos” em dias letivos dos professores devem ser evitados;

A seguir será apresentado o modelo formal do PPTC como um problema de programação linear inteira mista, utilizando uma versão ligeiramente modificada da formulação apresentada em [65], que considera os seguintes dados de entrada:

- $P$ : conjunto de professores, com elementos  $p \in P$  numerados como  $1, \dots, |P|$ ;
- $T$ : conjunto de turmas, com elementos  $t \in T$  numerados como  $1, \dots, |T|$ ;
- $D$ : conjunto de dias letivos, com elementos  $d \in D$  numerados como  $1, \dots, |D|$ ;
- $H$ : conjunto de períodos letivos por dia, com elementos  $h \in H$  numerados como  $1, \dots, |H|$ ;
- $\tilde{R}_{|P| \times |T|}$ : matriz de requerimentos, onde  $\tilde{r}_{pt}$  indica quantas aulas o professor  $p$  deve lecionar para a turma  $t$ ;
- $\tilde{P}_{|P| \times |D| \times |H|}$ : matriz de disponibilidade de professores, onde  $\tilde{p}_{pdh} = 1$  indica se o professor  $p$  está disponível no dia  $d$  no período  $h$ ,  $\tilde{p}_{pdh} = 0$  caso contrário;
- $\tilde{M}_{|P| \times |T|}$ : matriz de limites diários de aulas, onde  $\tilde{m}_{pt}$  indica o máximo de aulas que o professor  $p$  pode lecionar na turma  $t$  em qualquer dia da semana, deve-se observar que como somente são consideradas aulas simples ou geminadas tem-se  $\tilde{m}_{pt} \in \{1, 2\} \quad \forall p \in P, t \in T$ ;
- $\tilde{G}_{|P| \times |T|}$ : matriz de requisições de aulas geminadas, onde  $\tilde{g}_{pt}$  indica quantas aulas geminadas, no mínimo, são requisitadas pelo professor  $p$ , para aulas com a turma  $t$ ;

- $W'_{|P|}$ : vetor de pesos onde  $w'_p$  indica a importância da não ocorrência de buracos na agenda do professor  $p$ ;
- $W''_{|P|}$ : vetor de pesos onde  $w''_p$  indica a importância de se oferecer uma agenda compacta para o professor  $p$ ;
- $W'''_{|P|}$ : vetor de pesos onde  $w'''_p$  indica a importância da satisfação dos pedidos de aulas geminadas do professor  $p$ ;

Dependendo da disponibilidade do professor, um subconjunto de períodos do dia permitirão a existência de aulas geminadas. Será denotado por  $\widehat{G}_{pd} \subset H$  o subconjunto de períodos na agenda do professor  $p$  no dia  $d$  com a propriedade que  $\tilde{p}_{ptdh} \cdot \tilde{p}_{ptdh+1} = 1 \forall h \in \widehat{G}_{pd}$ , ou seja, os períodos que podem ser o início de uma aula geminada.

A seguir apresenta-se uma formulação para o PPTC, aqui denotada por  $\mathcal{F}_1$ , onde a variável de decisão  $x_{ptdh}$  indica se o professor  $p$  está dando aula para a turma  $t$  no dia  $d$  no período  $h$  ( $x_{ptdh} = 1$ ) ou não ( $x_{ptdh} = 0$ ). As variáveis auxiliares  $v_{pd}$ ,  $b_{pd}$  e  $g_{pt}$  medem a violação das restrições fracas relacionadas à capacidade de horários em número de dias, não ocorrência de buracos e satisfação de pedidos de aulas geminadas, respectivamente, como segue:

$$\text{minimizar} \quad \sum_{p \in P} \sum_{d \in D} w'_p \cdot b_{pd} + \sum_{p \in P} \sum_{d \in D} w''_p \cdot v_{pd} + \sum_{p \in P} \sum_{t \in T} w'''_p \cdot g_{pt} \quad (2.6)$$

sujeito a

$$\sum_{d \in D} \sum_{h \in H} x_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \quad (2.7)$$

$$\sum_{p \in P} x_{ptdh} \leq 1 \quad \forall t \in T, d \in D, h \in H \quad (2.8)$$

$$\sum_{t \in T} x_{ptdh} \leq \tilde{p}_{pdh} \quad \forall p \in P, d \in D, h \in H \quad (2.9)$$

$$\sum_{h \in H} x_{ptdh} \leq \tilde{m}_{pt} \quad \forall p \in P, t \in T, d \in D \quad (2.10)$$

$$v_{pd} \geq \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H \quad (2.11)$$

$$\bar{a}_{pd} \leq (|H| + 1) - (|H| + 1 - h) \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H \quad (2.12)$$

$$\underline{a}_{pd} \geq h \cdot \sum_{t \in T} x_{ptdh} \quad \forall p \in P, d \in D, h \in H \quad (2.13)$$

$$b_{pd} \geq \underline{a}_{pd} - \bar{a}_{pd} + v_{pd} - \sum_{t \in T} \sum_{h \in H} x_{ptdh} \quad \forall p \in P, d \in D \quad (2.14)$$

$$y_{ptdh} \leq x_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in \hat{G}_{pd} \quad (2.15)$$

$$y_{ptdh} \leq x_{ptdh+1} \quad \forall p \in P, t \in T, d \in D, h \in \hat{G}_{pd} \quad (2.16)$$

$$\sum_{h'=h}^{h+2} y_{ptdh'} \leq 1 \quad \forall p \in P, t \in T, d \in D, h \in \hat{G}_{pd} |h < |H| - 1 \quad (2.17)$$

$$g_{pt} \geq \tilde{g}_{pt} - \sum_{d \in D} \sum_{h \in \hat{G}_{pt}} y_{ptdh} \quad \forall p \in P, t \in T \quad (2.18)$$

$$x_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (2.19)$$

$$y_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in \hat{G}_{pd} \quad (2.20)$$

$$v_{pd} \in \{0, 1\} \quad \forall p \in P, d \in D \quad (2.21)$$

$$\bar{a}_{pd} \in \{0, \dots, |H|\} \quad \forall p \in P, d \in D \quad (2.22)$$

$$\underline{a}_{pd} \in \{0, \dots, |H|\} \quad \forall p \in P, d \in D \quad (2.23)$$

$$b_{pd} \in \{0, \dots, |H| - 2\} \quad \forall p \in P, d \in D \quad (2.24)$$

As restrições 2.7, 2.8 e 2.9 já foram apresentadas com o PPT. As restrições 2.10 tratam do limite diário de alocações professor  $\times$  turma. As variáveis  $v_{pd}$ , cuja ligação com as variáveis de decisão é feita pelas restrições 2.11, indicam se há atividade letiva para o professor  $p$  no dia  $d$ . A medição do número de buracos na agenda do professor (restrições 2.14) é feita nas variáveis  $b_{pd}$ , considerando as variáveis  $\bar{a}_{pd}$  e  $\underline{a}_{pd}$  (cuja amarração é feita nas restrições 2.12 e 2.13) que indicam, respectivamente, o primeiro e último períodos do dia letivo com alguma alocação. A existência de uma aula geminada para o professor  $p$  para a turma  $t$  no dia  $d$  iniciando no período  $h$  é avaliada na variável  $y_{ptdh}$ , através das restrições 2.15 e 2.16 <sup>1</sup>, as restrições 2.17 garantem que nenhum

<sup>1</sup> A amarração das variáveis  $y_{ptdh}$  com as variáveis  $x_{ptdh}$  utilizando menos restrições do que em [65] é possível devido ao fato de que a função objetivo assegura que sempre que necessário e possível o valor das variáveis  $y_{ptdh}$  estará no limite superior para garantir os valores corretos nas variáveis  $g_{pt}$ .

período será compartilhado por duas aulas geminadas. Finalmente, mede-se o não atendimento do número mínimo de aulas geminadas para um professor  $p$  e turma  $t$  em 2.18. As restrições restantes determinam o domínio das variáveis.

## 2.4 Programação de Horários de Cursos em Universidades

O Problema de Programação de Cursos em Universidades (PPHU) consiste, tipicamente na determinação do horário e local em que serão lecionados **cursos** ou realizados **exames** em universidades. O segundo caso é bastante peculiar e será discutido na seção seguinte. No agendamento dessas atividades deve-se considerar a utilização de recursos restritos, como salas de aula. Adicionalmente, determinados horários ou salas podem ser preferenciais para determinados professores ou cursos. Essas preferências são tratadas como restrições fracas.

No PPHU, o conceito de turmas é menos relevante do que no PPHE, visto que a possibilidade dos alunos personalizarem o currículo é maior. Não é incomum, inclusive, alunos inscreverem-se em disciplinas em mais de um departamento. Considerando a alocação de cursos, geralmente, o horário é construído antes da matrícula dos alunos, já no caso da programação de exames, tipicamente, a agenda é construída após a inscrição desses.

Na *II International Timetabling Competition*, o problema de programação de cursos em universidades é dividida em duas categorias:

1. **programação de cursos pós-matrícula**: nesse caso, os alunos matriculam-se nos cursos que gostariam de assistir; a programação de horários deve então ser construída de modo que os estudantes consigam assistir aos cursos que escolheram;
2. **programação de cursos baseada em currículo**: nesse problema deve-se realizar o agendamento em um conjunto de períodos semanal de vários cursos considerando também a alocação de um número limitado de salas, os conflitos entre cursos são determinados de acordo com o currículo dos cursos e não de acordo com as matrículas efetuadas.

O problema da categoria 2 apresenta várias semelhanças com o PPHE, de modo que os comentários dessa seção serão mais relacionados ao problema 1. O Problema de Programação de Cursos Pós-Matrícula (PPCPM) considerado na II ITP [52] considera os seguintes dados de entrada:

- $n$  **eventos** que devem ser agendados um conjunto de períodos divididos em dias;
- $r$  **salas** cada uma com sua capacidade específica;
- $f$  conjunto de **recursos** que podem ser disponibilizados em uma sala;
- $s$  **estudantes** que desejam assistir um ou mais eventos;
- a **disponibilidade** de cada evento - nem todos os eventos podem ser agendados em todos os períodos;

- um conjunto de requerimentos de **precedência** que informa que certos eventos devem ser agendados antes de outros.

O objetivo é alocar um período e uma sala para cada um dos  $n$  eventos considerados, respeitando as seguintes restrições fortes:

1. cada estudante deve assistir, no máximo, um evento por período;
2. a alocação da sala deve respeitar sua capacidade e garantir que a mesma provê os recursos suficientes para o evento;
3. cada sala deve receber no máximo um evento por período;
4. a disponibilidade dos eventos deve ser respeitada;
5. a precedência requerida entre os eventos deve ser respeitada.

Como restrições fracas, consideradas na avaliação de um quadro de horários, considera-se:

1. estudantes não devem ser alocados para eventos nos últimos períodos do dia;
2. estudantes não devem assistir mais que três eventos por dia;
3. as alocações devem evitar que algum estudante tenha apenas um evento agendado em um determinado dia.

## 2.5 O Problema de Alocação de Salas

Um problema bastante comum em instituições de ensino superior, que aparece como decomposição do problema anteriormente apresentado é a alocação de aulas previamente programadas a salas, ou seja, resolve-se o problema tratado em 2.4 desconsiderando-se a alocação de salas, o que será posteriormente decidido. Tal problema, conhecido na literatura inglesa como *Classroom Assignment Problem*, e aqui denotado por Problema de Alocação de Aulas a Salas (PAAS), consiste em alocar aulas, com horários de início e término previamente conhecidos, a um número fixo de salas.

No processo de alocação de aulas a salas, são observados vários requisitos, sendo os mais comuns listados a seguir. Como restrições fortes considera-se:

- em uma mesma sala e horário não pode haver mais de uma aula;
- uma sala não pode receber uma turma cuja quantidade de alunos seja superior à sua capacidade;
- algumas salas têm alguns horários previamente reservados para a realização de outras atividades e nesses horários elas ficam indisponíveis.

Como restrições fracas mais frequentes, têm-se:

- certas aulas requerem recursos especiais, por exemplo, projetores multimídia, requerendo assim salas que tenham esses recursos;

- certas salas têm restrições de uso e a utilização delas deve ser evitada tanto quanto possível. Em geral são auditórios, salas de defesa de tese e ambientes mais sofisticados, os quais devem ser evitados;
- sempre que possível, alocar a uma mesma sala alunos de um mesmo curso e período; o atendimento a esse requisito visa, também, a diminuição do fluxo de alunos mudando de uma sala para outra, evento que aumenta o nível de ruído no ambiente e pode comprometer a realização de outras aulas que ocorrem no período de transição;
- sempre que possível, alocar todas as aulas semanais de uma turma de uma disciplina a uma mesma sala; em geral, cada turma de disciplina tem dois encontros semanais com seu professor. Como os alunos fazem várias disciplinas por semestre, normalmente seis ou sete, seriam muitas as salas para os alunos memorizarem;
- evitar alocar aulas de turmas pequenas a salas de maior capacidade. Esse requisito é normalmente avaliado pela quantidade de carteiras vazias;
- sempre que possível, deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza.

De acordo com Carter e Tovey [24], o PAAS é, em seu caso geral, da classe NP-difícil. Para sua resolução e garantia de existência de solução, são geralmente criadas salas fictícias. A existência na solução final de uma aula alocada a uma sala desta natureza indica que o responsável pela alocação das aulas às salas deve negociar a realização da aula em prédio de outra unidade acadêmica. Alguns trabalhos sobre o PAAS são [24, 68, 63].

## 2.6 Programação de Horários de Exames em Universidades

No Problema de Programação de Exames (PPE) tem-se um conjunto de alunos que devem realizar um ou mais exames, os quais devem ser agendados em um conjunto de períodos, que possivelmente estende-se por mais de uma semana. O PPE é um problema combinatório difícil, tipicamente encontrado em universidades, ocorrendo no contexto de exames de final de período, onde cada exame corresponde a uma disciplina. Esse problema é enfrentado por instituições de ensino até 4 vezes por ano. Como reportado por Carter [22], a existência bastante comum de disciplinas eletivas e a conseqüente possibilidade de personalização da grade de horários de alunos de acordo com preferências individuais torna o problema bastante difícil de se resolver na prática.

As alocações devem ser feitas de modo a evitar conflitos para alunos: nenhum aluno pode ter mais que um exame agendado para o mesmo período. Eventualmente essa restrição é relaxada e tem-se como objetivo a minimização dos conflitos para estudantes. Adicionalmente, dentro do possível, a agenda de exames de cada aluno deve apresentar um certo espalhamento das atividades, evitando-se a alocação de mais de um exame por dia e, especialmente, exames em períodos consecutivos do mesmo dia. De modo genérico, [50] define uma função que mede o não atendimento do espalhamento de exames para alunos. Seja  $x_{it}$  a variável binária que indica se o exame  $i$  será realizado no período  $t$ ,

Do ponto de vista teórico, o problema de se definir uma programação de exames sem conflitos pode ser visto como um problema de coloração de vértices em grafos [47], sendo considerado intratável computacionalmente (NP-Completo) no caso geral.

Os recursos limitados que tipicamente restringem a alocação simultânea de alguns exames são a disponibilidade de salas com capacidade adequada e a disponibilidade de supervisores de exames [22, 23]. Essa restrição, entretanto, pode ser relaxada em determinadas situações, visto que salas com grande capacidade ou exames com poucos alunos permitem a utilização compartilhada de salas ou supervisores.

No caso de as salas estarem distribuídas geograficamente em grandes áreas, deve-se considerar que na agenda de supervisores de exames, a alocação de atividades em período próximos deve respeitar o intervalo de tempo necessário para o deslocamento de supervisores entre as salas dos exames em questão [23].

Em [3] considera-se um PPE não capacitado, ou seja, que não leva em conta a capacidade das salas. Nesse problema, o objetivo é espalhar os exames dos alunos pelo conjunto de períodos. Nesse problema consideram-se os seguintes dados de entrada:

- $N$ : número de exames;
- $E_i$ : exame, com  $i \in 1, \dots, N$
- $B$ : conjunto de todos os  $N$  exames, com  $B = E_1, \dots, E_N$
- $D$ : conjunto de dias;
- $T$ : número de períodos disponíveis (em todos os dias);
- $M$ : número de estudantes;
- $C_{N \times N}$ : matriz de conflitos onde  $c_{ij}$  indica quantos alunos devem realizar tanto o exame  $i$  quanto o exame  $j$ ;

Ainda, considere a variável  $t_k$ , que indica o período selecionado para o exame  $k$ . Dessa forma, o PPE considerado em [3] é:

$$\text{minimizar} \quad \frac{\sum_{i=1}^{N-1} F_1(i)}{M}$$

sendo que

$$F_1(i) = \sum_{j=i+1}^N c_{ij} \cdot \text{proximidade}(t_i, t_j) \quad (2.25)$$

$$\text{proximidade}(t_i, t_j) = \begin{cases} 2^5/2^{|t_i-t_j|} & \text{se } |t_i - t_j| \leq 5 \\ 0 & \text{caso contrário} \end{cases} \quad (2.26)$$

sujeito a

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} \cdot \lambda(t_i, t_j) = 0 \quad (2.27)$$

sendo que

$$\lambda(t_i, t_j) = \begin{cases} 1 & \text{se } t_i = t_j \\ 0 & \text{caso contrário} \end{cases} \quad (2.28)$$

A equação 2.26 computa a penalidade referente à alocação de exames muito próximos (com menos de 5 períodos de distância), de modo que penalidade aumenta significativamente à medida que os exames tem menor distância, sendo que em 2.26 essa penalidade é multiplicada pelo número de alunos envolvidos nesses exames. Conflitos são estritamente proibidos, como mostra a equação 2.27.

## Capítulo 3

# Produção Automatizada de Quadros de Horários usando Heurísticas

O desenvolvimento de algoritmos para programação de horários que serão utilizados no dia-a-dia das instituições deve considerar uma importante restrição: o tempo de execução. Apesar de que em muitas instituições um tempo considerável é disponibilizado para essa atividade de planejamento, alguns meses até, um método que produza uma solução de boa qualidade rapidamente será sempre preferível a um que necessite um tempo computacional maior para produzir uma solução de qualidade equivalente. Ademais, uma característica bastante desejada em programas de programação de horários é a interatividade, a visualização rápida de uma solução por parte do usuário permite que o mesmo identifique possíveis problemas na solução apresentada. Em muitos casos, a identificação desses problemas exige a modificação dos dados de entrada e re-execução do algoritmo de solução. Esse processo pode se repetir várias vezes, tornando muito importante a disponibilidade de um algoritmo rápido.

### 3.1 Heurísticas Construtivas

Nas heurísticas construtivas busca-se a obtenção de uma solução inicial, preferencialmente factível. No caso dos problemas de programação de horários, uma qualidade certamente desejada nas soluções iniciais é a não existência de conflitos. A construção de um quadro de horários sem conflitos pode ser vista como um Problema de Coloração de Grafos (PCG) [47], por essa razão, grande parte das heurísticas construtivas para programação de horários, consiste, ou baseia-se fortemente, em heurísticas de coloração de grafos.

Para visualizar o problema de programação de horários como um PCG, considere que cada nó representa uma atividade, como uma aula, por exemplo, que deve ser alocada a um período. O conjunto de períodos é o conjunto de “cores” do grafo e dois nós são adjacentes caso correspondam a atividades que não podem ocorrer simultaneamente, como é o caso de duas aulas que devem ser ministradas pelo mesmo professor. De modo geral, dois nós são considera-

dos adjacentes caso disputem algum recurso não compartilhável, casos típicos envolvem também salas especiais, como um laboratório de informática ou um ginásio de esportes.

Encontrar uma programação de horários sem conflitos, ou seja, uma coloração factível de um grafo utilizando um dado número de cores (períodos), no caso geral, é uma tarefa de grande dificuldade computacional. De fato, o problema de coloração de grafos é um clássico problema combinatório NP-Completo [38]. Desse modo, se o objetivo é a construção rápida de uma solução sem conflitos, uma boa opção é utilização de heurísticas, que em muitos casos práticos podem ser suficientes para a construção de soluções factíveis. Boa parte dos trabalhos iniciais em programação de horários utilizava diretamente heurísticas de coloração de grafos, como apontado em [22, 23], sendo que as heurísticas mais comuns desse tipo são heurísticas de seqüenciamento. Essas heurísticas selecionam, passo-a-passo, um nó não colorido do grafo e atribuem uma cor. Os métodos apontados em [22] são:

1. **maior grau primeiro**[73]: nós com maior grau (maior número de vizinhos) são coloridos primeiro, selecionando-se sempre a cor de menor número que não cause conflitos;
2. **maior grau primeiro: preenchimento do topo**[57]: do mesmo modo, os nós são ordenados por grau, percorre-se a lista alocando tantos nós quanto possível na primeira cor disponível, volta-se para o início da lista e inicia-se o preenchimento do próximo período, até que não restem mais nós sem cor;
3. **maior grau primeiro recursivo: preenchimento do topo**[21]: similar ao anterior, exceto que na remoção de um nó da lista o os graus dos vértices ainda não coloridos são computados considerando-se o subgrafo remanescente;
4. **maior grau modificado primeiro**[76]: nesse caso, considera-se que a definição da cor de um nó é crítica se o mesmo for válido para os seus vizinhos, nesse sentido, utiliza-se a seguinte fórmula para o cálculo da prioridade  $p_1(i)$  de alocação de um vértice  $i$ :  $p_1(i) = \sum_{j \in N(i)} d(j)$ , onde  $N(i)$  é o conjunto de nós adjacentes ao nó  $i$  e  $d(j)$  é o grau de um vértice  $j$ , posteriormente, os valores de  $p_1(i)$  são normalizados e computa-se  $p_2(i) = \sum_{j \in N(i)} p_1(j)$  e assim recursivamente  $p_{k+1}(i) = \sum_{j \in N(i)} p_k(j)$ , para todos os vértices  $i$ , até o ponto de estabilização; o autor demonstra que essa estratégia apresenta melhores resultados que a estratégia “maior grau primeiro”, considerando o número de cores utilizadas;
5. **maior 1-grau modificado primeiro**[34]: nesse caso utiliza-se apenas a prioridade  $p_1$  apresentada anteriormente, experimentos de [34] mostraram que essa estratégia é melhor do que a estratégia “maior grau primeiro”, sendo que a recursão adicional em para  $p_k(i)$ , com  $k > 1$  não produz melhora significativa;
6. **menor grau por último recursivo**[53]: utilizando o princípio considerado em (1), de que os nós com menor grau são mais facilmente coloridos, em cada passo um nó de menor grau é removido e colocado no final de uma lista, recomputa-se o grau do subgrafo remanescente e continua-se o

processo até que a lista esteja completa, quando então inicia-se a coloração considerando o nó do topo da lista;

7. **menor grau por último recursivo com troca** [53]: considera-se uma regra de troca, que aplica-se a qualquer uma das heurísticas “maior grau primeiro”, procedendo do seguinte modo:
  - (a) o vértice  $i$ , no topo da lista, é associado com a cor de menor número na qual não ocorrerão conflitos, considerando as cores já utilizadas;
  - (b) se um vértice  $i$  conflitar em todas as cores correntes, procura-se uma cor  $c_j$  na qual exista apenas um nó  $j$  em conflito; se possível, mude a cor do vértice  $j$ ; caso contrário tente uma troca bicromática do seguinte modo: para toda cor  $r$  considere o conjunto  $C_r$  de vértices com essa cor, se nenhum vértice em  $C_r$  conflitar com  $i$  então mude a cor do vértice troque a cor de  $j$  para  $r$  e a cor dos vértices em  $C_r$  para  $c_j$ , o que permite a atribuição da cor  $c_j$  para o vértice  $i$ ; se essa troca não for possível, cria-se uma nova cor para o curso corrente e continua-se a alocação.
8. **maior grau de saturação primeiro recursivo** [14]: os vértices não coloridos são ordenados considerando-se primeiramente o número de cores em vértices adjacentes já coloridos (o grau de saturação), como critério de desempate utiliza-se o grau do vértice no subgrafo não colorido.

Métodos desse tipo têm a vantagem de serem geralmente muito rápidos e consumirem pouca memória. Em [37], por exemplo, é detalhada a experiência de utilização de um algoritmo desse tipo para a construção da programação de exames durante o final da década de 60, na universidade de Nottingham, quando a aquisição de um computador mais poderoso permitiu o agendamento de 651 exames, utilizando 12 Kilobytes de memória RAM.

## 3.2 Busca Local

O propósito dos Métodos de Busca Local (BL) é a melhora de soluções. Iniciando com uma solução provida por algum método construtivo, como os apresentados na seção 3.1, esses métodos exploram um conjunto de soluções, denominadas soluções vizinhas, procurando soluções melhores.

Idealmente, a solução inicial recebida é uma solução factível. No entanto, como visto anteriormente, não se conhece um algoritmo eficiente, com complexidade de tempo polinomial, para a construção de uma solução sem conflitos para o caso geral do problema de programação de horários. Desse modo, a abordagem mais usual é a relaxação das restrições de difícil atendimento, como é o caso dos conflitos. Essas restrições são dualizadas, isto é, embutidas na função objetivo. Assim, mede-se o não atendimento de algumas restrições fortes, com penalidades suficientemente grandes associadas.

Um conceito chave em BL é o de movimento. Um movimento é uma alteração possível em um solução. Em um quadro de horários, por exemplo, um movimento pode ser a troca de período de uma aula de um professor. A definição de vizinhança decorre da definição anterior: a vizinhança  $\mathcal{N}(s)$  de uma solução  $s$  é o conjunto de todas as soluções que podem ser alcançadas, partindo-se de  $s$ ,

com a aplicação de um único movimento. Em PPHs típicos de instituições de ensino, as soluções factíveis encontram-se bastante distantes umas das outras, ou seja, movimentos que realizam pequenas alterações no quadro de horários têm pouca chance de encontrar soluções melhores. O procedimento da seção 3.3 contorna esse problema através da descoberta de cadeias de movimentos que se realizados juntamente produzem uma solução melhor.

### 3.3 Procedimento Intraturmas-Interturmas

Intraturmas-Interturmas (II) é um procedimento de busca local baseado em caminhos mínimos, proposto em [64].

Seu funcionamento será mostrado tendo por base o PPTC, conforme definido na seção 2.3.1. Considere os requerimentos do PPTC:

- (a) um professor não pode ministrar aula para mais de uma turma ao mesmo tempo;
- (b) uma turma não pode ter aula com mais de um professor em um mesmo horário;
- (c) cada professor  $i$  tem que cumprir sua carga horária semanal  $r_{ij}$  para cada turma  $j$ ;
- (d) um professor não pode ser alocado a um horário no qual não esteja disponível;
- (e) uma turma não pode ter mais do que  $u_{ij}$  horários diários de aula de uma mesma matéria;
- (f) atendimento ao maior número possível de aulas geminadas para os professores que as requererem;
- (g) a agenda dos professores deve ser tão compacta quanto possível.

Considere, também, as seguintes definições:

**Inviabilidade:** um quadro de horário de professores que não satisfaz a pelo menos uma das restrições fortes (a),  $\dots$ , (e) é considerado **inviável**.

**Inviabilidade do tipo 1:** um quadro  $Q$  de horário de professores se diz **inviável do tipo 1** se, em pelo um horário, uma das condições a seguir for satisfeita:

- existe turma tendo aula com mais de um professor (restrição (b) não é verificada);
- existe uma turma sem aula.

**Inviabilidade do tipo 2:** Um quadro  $Q$  de horário de professores se diz **inviável do tipo 2** se a restrição (e) não for satisfeita para algum professor.

Um quadro de horário de professores é representado como uma matriz  $Q_{m \times p}$ , onde  $m$  é o número de professores e  $p$  é o número total de períodos ( $p = |D| \cdot |P|$ , considerando a notação de 2.3.1) de valores inteiros, onde em cada linha  $i$  de  $Q$  representa-se a alocação semanal do professor  $i$ . Cada elemento  $q_{ik} \in \{-1, 0, 1, 2, \dots, n\}$  indica a atividade do professor  $i$  no horário  $k$ . Valores negativos indicam que o professor está indisponível, enquanto valores nulos indicam inatividade no horário.

Um quadro de horário é avaliado pela seguinte função, a qual deve ser minimizada:

$$f(Q) = \omega \times f_1(Q) + \delta \times f_2(Q) + \rho \times f_3(Q) \quad (3.1)$$

em que as duas primeiras componentes mensuram o nível de inviabilidade dos tipos 1 e 2, respectivamente, e a terceira, o nível de satisfação dos professores com relação ao atendimento de seus requisitos pessoais.

$\omega$ ,  $\delta$  e  $\rho$  são pesos que refletem a importância relativa de cada uma das componentes de  $f$ . De forma a gerar uma estrutura hierárquica, tais pesos são escolhidos satisfazendo a condição:  $\omega > \delta \gg \rho$ .

O nível de inviabilidade do tipo 1 de  $Q$ ,  $f_1(Q)$ , é mensurado somando-se, para cada horário  $k$ : (a) o número de vezes  $l_k$  que uma turma está sem atividade no horário  $k$ ; (b) o número de vezes  $s_k$  que mais de um professor dá aula para uma mesma turma no horário  $k$ .

$$f_1(Q) = \sum_{k=1}^p (l_k + s_k) \quad (3.2)$$

Com relação ao nível de inviabilidade do tipo 2, o quadro  $Q$  é avaliado somando-se o número de vezes  $e_i$  que a restrição (e) não é atendida para cada professor  $i$ , isto é:

$$f_2(Q) = \sum_{i=1}^m e_i \quad (3.3)$$

A satisfação ao atendimento dos requisitos pessoais dos professores é medida com relação à compacidade do quadro de horários (restrição (g)), bem como ao atendimento do número de aulas geminadas requeridas (restrição (f)). Mais precisamente,

$$f_3(Q) = \sum_{i=1}^m (\omega'_i \times b_i + \omega''_i \times v_i + \omega'''_i \times g_i) \quad (3.4)$$

sendo  $\omega'_i$ ,  $\omega''_i$  e  $\omega'''_i$  pesos que refletem, respectivamente, a importância relativa do número de buracos  $b_i$  (horários ociosos entre dois horários de aula de um mesmo turno), do número de dias  $v_i$  na semana que cada professor está envolvido em alguma atividade de ensino em um mesmo turno e da diferença não-negativa  $g_i$  entre o número mínimo requerido de aulas geminadas e o efetivamente existente na agenda corrente de cada professor  $i$ , como segue:

$$g_i = \max \left\{ 0, \text{geminadas}(Q_i^{\text{requerido}}) - \text{geminadas}(Q_i^{\text{corrente}}) \right\}$$

O procedimento Intraturmas-Interturmas (II) é acionado quando uma solução sem inviabilidade do tipo 1 está disponível. Primeiramente, ele procura

restaurar a viabilidade do quadro de horário, isto é, zerar a parcela  $f_2$  da função objetivo (3.1). Se bem sucedido, ele tenta, numa segunda etapa, melhorar os requisitos de qualidade exigidos para o quadro de horário, navegando no espaço das soluções viáveis, isto é, ele tenta diminuir o valor da componente  $f_3$  respeitando as restrições fortes (a), ..., (e).

Como a forma de atuação de II tanto para restaurar a viabilidade quanto para melhorar um quadro de horário viável é semelhante, é mostrado seu princípio de funcionamento apenas para o segundo caso.

### 3.3.1 Procedimento Intraturmas

Suponha disponível, então, uma solução sem inviabilidade.

Assim, dado um quadro  $Q$  de horário de professores nessas condições ( $f_1(Q) = f_2(Q) = 0$ ), define-se o grafo da turma  $j$  por  $G_j = (V_j, A_j)$ , onde  $V_j$  é o conjunto dos horários reservados para a turma  $j$  e  $A_j$  é o conjunto dos arcos orientados definido conforme a seguir:

$A_j = \{(k, \bar{k}) : \text{professor que leciona para a turma } j \text{ no horário } k \text{ está disponível no horário } \bar{k} \text{ e a restrição forte (e) é respeitada no horário } \bar{k}\}$

A cada arco  $(k, \bar{k}) \in G_j$  associa-se um custo  $\Delta f_i(k, \bar{k})$ , o qual representa a variação do custo de se transferir o professor  $i$  do horário  $k$  para o horário  $\bar{k}$ , tendo em vista a componente  $f_3$  da função objetivo (3.1). Desta forma, o custo é obtido calculando-se a diferença entre os valores da função objetivo, relativa ao professor, nas configurações nova e antiga, isto é:

$$\Delta f_i(k, \bar{k}) = f_i(\bar{k}) - f_i(k) \quad (3.5)$$

sendo  $f(\cdot) = (\rho \times f_3)(\cdot)$

A Tabela 3.1 mostra um fragmento de um quadro  $Q_1$  de horário de professores. Cada linha  $i$  representa um professor ( $i = P1, P2, P3, P4$ ) e cada coluna  $k$  um horário ( $k = H1, H2, H3, H4, H5$ ) de um mesmo dia. Cada elemento  $q_{ik}$  desta tabela representa a atividade do professor  $i$  no horário  $k$ . A, B, C e D são turmas. Um traço (-) significa que o professor está indisponível, enquanto uma célula vazia indica que não há atividade no horário. A coluna  $f_i$  indica o valor da função objetivo de cada professor, calculada conforme (3.1) e (3.4), com  $\rho = 1$ ,  $\omega_i' = 1$  e  $\omega_i'' = \omega_i''' = 0 \forall i$ . Para esse quadro de horário, tem-se  $f(Q_1) = f_{P1} + f_{P2} + f_{P3} + f_{P4} = 1 + 1 + 0 + 0 = 2$ .

Tabela 3.1: Quadro  $Q_1$

	H1	H2	H3	H4	H5	$f_i$
P1	A		B	B		1
P2	B	C		A	A	1
P3		B	A	C	B	0
P4	C	A	C	D	-	0

A Figura 3.1 ilustra  $G_A$ , o grafo de horários da turma A. Cada horário é representado por um vértice, ao qual está associado um professor. O arco  $(H1, H5)$  de custo  $-1$  indica, por exemplo, que se o professor  $P1$  mudar sua aula do horário  $H1$  para o horário  $H5$ , haverá uma diminuição no valor da sua função objetivo de 1 unidade ( $\Delta f_{P1}(H1, H5) = f_{P1}(H5) - f_{P1}(H1) = 0 - 1 = -1$ ).

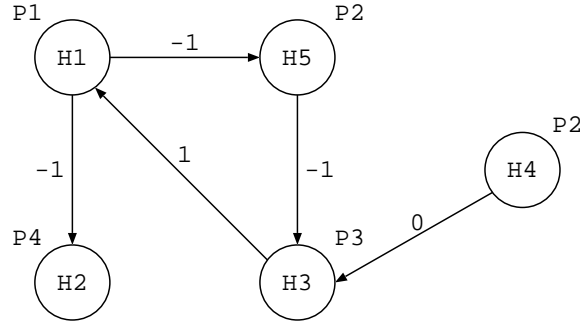


Figura 3.1:  $G_A$ , Grafo da Turma A

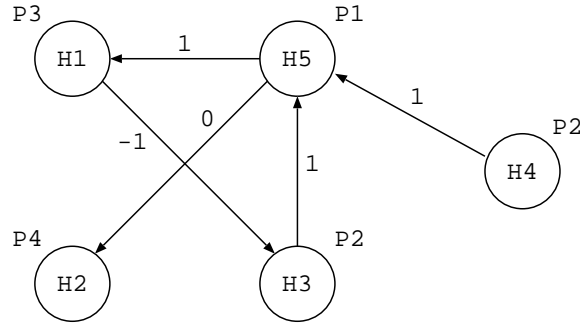


Figura 3.2:  $G'_A$ , Grafo da Turma A após os movimentos

Para encontrar um quadro de horário com um valor menor para a função objetivo 3.1, é suficiente procurar um ciclo de custo negativo em  $G_j$ . Observa-se que a existência de um ciclo garante a re-alocação de todas as aulas nele envolvidas, preservando o atendimento às restrições fortes (a), ..., (d). O atendimento à restrição forte (e) será discutido mais adiante.

No exemplo em questão, a sequência de arcos  $\{(H1, H5), (H5, H3), (H3, H1)\}$  forma um ciclo de custo total  $-1$  ( $= -1 + (-1) + 1$ ). Tal sequência, por envolver aulas de uma mesma turma, define um conjunto de movimentos ditos *intraturmas*.

A Tabela 3.2 mostra  $Q'_1$ , o quadro de horário dos professores após os movimentos e a Figura 3.2 ilustra  $G'_A$ , o novo grafo de horários da turma A. Em função dos movimentos promovidos em  $G_A$ , o novo quadro  $Q'_1$  tem função objetivo dada por  $f(Q'_1) = f_{P1} + f_{P2} + f_{P3} + f_{P4} = 0 + 0 + 1 + 0 = 1$ .

Tabela 3.2: Quadro  $Q'_1$

	H1	H2	H3	H4	H5	$f_i$
P1			B	B	A	0
P2	B	C	A	A		0
P3	A	B		C	B	1
P4	C	A	C	D	-	0



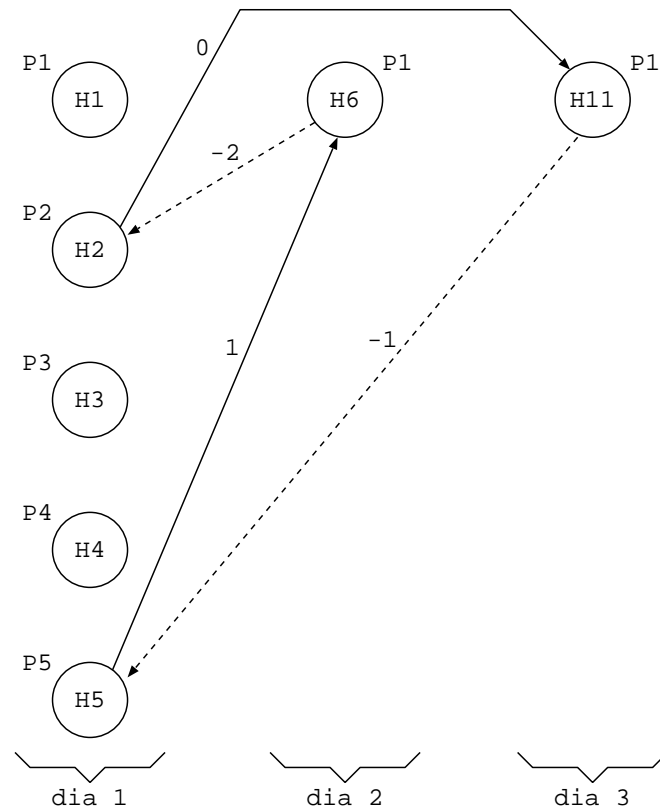


Figura 3.4: Exemplo de ciclo de custo negativo que produz inviabilidade do tipo 2

que cada professor possa ministrar no máximo dois horários de aula por dia para cada turma, são possíveis os movimentos das aulas dos horários  $H6$  e  $H11$  para o dia 1, tomados individualmente. No entanto, se os arcos correspondentes a esses movimentos fizerem parte de um mesmo ciclo de custo negativo, tal como o mostrado na Figura 3.4, a viabilidade não estará garantida.

Essas situações, entretanto, só ocorrem quando um mesmo professor participa em mais de um arco no ciclo. Assim, faz-se necessário checar a viabilidade e o valor da função objetivo após os movimentos candidatos. De forma a encontrar outros ciclos de custo negativo em um grafo nessas condições, procede-se como segue: Escolhe-se um arco qualquer do ciclo,  $(k, \bar{k}) \in G_j$ , inserindo-o em uma lista  $L$  de movimentos proibidos. A seguir, atualiza-se o grafo da turma sob avaliação, excluindo de  $G_j$  os arcos pertencentes à  $L$ , e procura-se outro ciclo de custo negativo. Quando não mais for possível encontrar ciclos de custo negativo, passa-se para uma outra turma e zera-se a lista  $L$ .

O procedimento iterativo Intraturmas analisa, a cada vez, uma turma. Para cada turma  $j$  constrói-se seu grafo  $G_j$  e aplica-se um algoritmo para detectar ciclos de custo negativo. Enquanto houver ciclo de custo negativo que melhore o valor da função objetivo e não produza inviabilidade, são realizados os movimentos e atualizados o quadro de horário e o grafo  $G_j$ . Inexistindo mais tais ciclos no grafo da turma  $j$  parte-se para uma nova turma. Esse procedimento

é encerrado quando nenhum movimento de melhora for possível para todas as turmas. O procedimento Intraturmas, descrito na Figura 3.5, formaliza esse mecanismo.

```

procedimento Intraturmas( $Q, f(\cdot)$ )
1   $j \leftarrow 1$ ;   {Número da turma}
2  novarodada  $\leftarrow$  false;
3  enquanto ( $j \leq n$ ) faça
4      enquanto ( houver ciclo de custo negativo em  $G_j$ 
5                  que melhore a função objetivo
6                  e não viole a viabilidade ) faça
7          Atualize  $G_j$  e o Quadro de Horário da turma  $j$ ;
8          novarodada  $\leftarrow$  true;
9      fim-enquanto;
10     se (novarodada = true e  $j = n$ )
11         então
12              $j \leftarrow 1$ ;
13             novarodada  $\leftarrow$  false;
14         senão
15              $j \leftarrow j + 1$ ;
16     fim-se;
17 fim-enquanto;
fim Intraturmas;

```

Figura 3.5: Procedimento Intraturmas

### 3.3.2 Procedimento Interturmas

Ao final do procedimento Intraturmas podem restar, ainda, arcos de custo negativo nos grafos das turmas. A Figura 3.6, que considera os grafos de duas turmas,  $j$  e  $\bar{j}$ , ilustra tal situação.

Nesta figura, há um arco de custo negativo na turma  $j$ , de  $k$  a  $\bar{k}$ , a saber  $c_j(k, \bar{k})$ , sinalizando que poderá haver uma melhora no valor da função objetivo se o professor  $i$  tiver sua aula do horário  $k$  transferida para o horário  $\bar{k}$ .

Esta transferência, entretanto, não pode ocorrer porque o professor  $\bar{i}$ , do horário  $\bar{k}$ , não está disponível no horário  $k$  (Neste horário, ele está lecionando para a turma  $\bar{j}$ ).

A idéia, então, é trocar o horário dessas aulas do professor  $\bar{i}$ , de forma a permitir a procura de um ciclo de custo negativo envolvendo os grafos das duas turmas, conectados nos horários  $k$  e  $\bar{k}$ .

Seja  $c_{\bar{j}}(\bar{k}, k)$  o custo do caminho mínimo de  $\bar{k}$  a  $k$  em  $G_{\bar{j}}$ . A existência de um ciclo de custo negativo envolvendo as duas turmas pode ser observada verificando se a condição  $c_j(k, \bar{k}) + c_{\bar{j}}(\bar{k}, k) < 0$  é satisfeita ao transferir o professor  $\bar{i}$  da turma  $j$  para a turma  $\bar{j}$  no horário  $\bar{k}$  e da turma  $\bar{j}$  para a turma  $j$  no horário  $k$ . Observa-se que não há custos envolvidos na transferência de um professor de uma turma para outra em um mesmo horário (arcos de custo nulo da Figura 3.6).

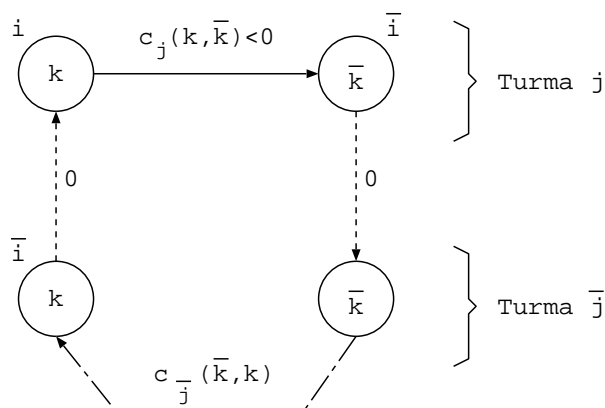


Figura 3.6: Grafos das turmas  $j$  e  $\bar{j}$  resultantes do procedimento Intraturmas

Desta forma, para cada arco  $(k, \bar{k}) \in G_j$ , de custo negativo, pesquisa-se a existência de um ciclo de custo negativo envolvendo esse arco de  $G_j$ , o grafo  $G_{\bar{j}}$  e os arcos de custo nulo que os conectam. Esta sequência de arcos, por envolverem turmas diferentes, define os chamados movimentos *interturmas*. Assim como em Intraturmas, faz-se necessário checar a viabilidade e o valor da função objetivo antes e depois dos movimentos candidatos.

O procedimento Interturmas, descrito na Figura 3.7, formaliza essa idéia.

Observa-se que toda vez que o Interturmas produz uma solução melhor, o procedimento Intraturmas é novamente acionado (linha 12 da Figura 3.7). Isso se deve ao fato de que o quadro de horário dos professores poderá ser melhorado com movimentos intraturmas, em função das alterações promovidas nos quadros de horário das turmas  $j$  e  $\bar{j}$ .

### 3.3.3 Procedimento II

O procedimento II consta de dois procedimentos, os quais são usados em conjunto. O primeiro, Intraturmas, realiza somente movimentos intraturmas e sua aplicação resulta em  $n$  grafos  $G_j$ ,  $\forall j = 1, \dots, n$ , com, possivelmente, arcos de custo negativo. O segundo, Interturmas, realiza movimentos interturmas envolvendo esses arcos de custo negativo e aciona Intraturmas sempre que há uma melhora no valor da função objetivo.

Conforme mencionado anteriormente, o procedimento II é acionado tão logo haja uma solução sem inviabilidade do tipo 1. II atua em duas etapas: primeiramente, ele tenta recuperar a viabilidade. Sendo bem sucedido, ele tenta, em uma segunda etapa, melhorar os requisitos de qualidade do quadro de horário. Na primeira etapa II usa somente a componente  $f_2$  da função objetivo como função avaliadora dos custos dos movimentos, isto é, a expressão 3.5 é avaliada tomando-se unicamente  $f(\cdot) = f_2(\cdot)$ . Na etapa seguinte a expressão 3.5 volta a ser avaliada usando-se a função 3.1, no caso, tomando-se  $f(\cdot) = (\rho * f_3)(\cdot)$ , uma vez que  $f_1(Q) = f_2(Q) = 0$ .

O algoritmo II encontra-se descrito pela Figura 3.8.

**procedimento** *Interturmas*( $Q, f(\cdot)$ )

- 1  $j \leftarrow 1$ ; {Número da turma}
- 2 *novarodada*  $\leftarrow$  *false* ;
- 3 enquanto ( $j \leq n$ ) faça
- 4     para ( todo arco  $(k, \bar{k}) \in G_j$  de custo negativo ) faça
- 5         Seja  $\bar{i}$  o professor do horário  $\bar{k}$  ;
- 6         Seja  $\bar{j}$ ,  $\bar{j} \neq j$ , a turma na qual  $\bar{i}$  está em atividade no horário  $k$  ;
- 7         Seja  $c_j(k, \bar{k})$  o custo de se transferir, na turma  $j$ ,  
a aula do horário  $k$  para o horário  $\bar{k}$  ;
- 8         Seja  $c_{\bar{j}}(\bar{k}, k)$  o custo, na turma  $\bar{j}$ , do caminho mí nimo  
entre  $\bar{k}$  e  $k$  ;
- 9         se ( $c_j(k, \bar{k}) + c_{\bar{j}}(\bar{k}, k) < 0$   
e função objetivo melhorar  
e viabilidade não for violada ) então
- 10             Atualize  $G_j$  e o Quadro de Horário da turma  $j$  ;
- 11             Atualize  $G_{\bar{j}}$  e o Quadro de Horário da turma  $\bar{j}$  ;
- 12             Aplique o Algoritmo *Intraturmas* ;
- 13             *novarodada*  $\leftarrow$  *true* ;
- 14         fim-se;
- 15     fim-para;
- 16     se (*novarodada* = *true* e  $j = n$  )
- 17         então
- 18              $j \leftarrow 1$ ;
- 19             *novarodada*  $\leftarrow$  *false* ;
- 20         senão
- 21              $j \leftarrow j + 1$ ;
- 22         fim-se;
- 23 fim-enquanto;

**fim** *Interturmas*;

Figura 3.7: Procedimento Interturmas

Tabela 3.3: Quadro  $Q_2$ 

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10		$f_i$
P1	A		A			A						5
P2		A						B	B	A		4
P3				A	A		A					4
P4								A	A			2

Tabela 3.4: Quadro  $Q'_2$ 

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10		$f_i$
P1	A						A			A		6
P2		A	A					B	B			4
P3				A	A	A						4
P4								A	A			2

**procedimento**  $II(Q, f(\cdot))$

- 1 Seja  $Q$  solução inicial satisfazendo  $f_1(Q) = 0$ ;
- 2 Seja  $Q^*$  a melhor solução resultante de II;
- 3 se  $(f_2(Q) \neq 0)$  então
- 4      $Q \leftarrow \text{Intraturmas}(Q, f_2)$ ;
- 5     se  $(f_2(Q) \neq 0)$  então  $Q \leftarrow \text{Interturmas}(Q, f_2)$ ;
- 6 fim-se;
- 7 se  $(f_2(Q) = 0)$  então
- 8      $Q \leftarrow \text{Intraturmas}(Q, f_3)$ ;
- 9      $Q \leftarrow \text{Interturmas}(Q, f_3)$ ;
- 10 fim-se;
- 11  $Q^* \leftarrow Q$ ;
- 12 Retorne  $Q^*$ ;

**fim**  $II$ ;

Figura 3.8: Procedimento II

### 3.4 Metaheurísticas

Nesta seção serão comentadas algumas técnicas genéricas que têm sido aplicadas com sucesso [18] na solução de diversos problemas de programação de horários. Voss [72] define esses métodos como:

Uma metaheurística é um processo mestre iterativo que guia e modifica operações de heurísticas subordinadas para a produção de solução de alta qualidade. Este processo pode manipular uma (ou múltiplas soluções completas (ou incompletas) a cada iteração. As heurísticas subordinadas podem ser procedimentos de alto (ou baixo) nível, ou uma busca local simples ou simplesmente um método de construção.

#### 3.4.1 Busca Tabu

O método de Busca Tabu, proposto independentemente por Glover [39] e Hansen [43] faz uso explícito de estruturas de memória para guiar métodos de descida de modo que esses continuem a exploração mesmo na ausência de movimentos de melhora. O Algoritmo 1 apresenta uma versão simplificada de uma implementação comum do método de Busca Tabu.

<p><b>Entrada:</b> <math>s^0</math> // solução inicial  <b>Saída:</b> <math>s^*</math> // melhor solução encontrada</p> <pre> 1 <math>s \leftarrow s^0</math>; 2 <math>s^* \leftarrow s^0</math>; 3 <b>enquanto</b> (<i>Critério de parada não satisfeito</i>) <b>faça</b> 4   Dada uma função de vizinhança <math>\mathcal{N}</math>, lista tabu <math>\mathcal{T}</math> e um critério de    aspiração encontre a melhor solução admissível <math>s' \in \mathcal{N}(s)</math>; 5   <math>s \leftarrow s'</math>; 6   insira a solução <math>s</math> (ou um atributo, ex.: movimento realizado) na lista    tabu <math>\mathcal{T}</math>; 7   <b>se</b> <math>f(s) &lt; f(s^*)</math> <b>então</b> 8     <math>s^* \leftarrow s</math>; 9   <b>fim</b> 10  atualize a lista tabu <math>\mathcal{T}</math>; 11 <b>fim</b></pre>
--

**Algoritmo 1:** Algoritmo de Busca Tabu

Hertz [44] propôs métodos de busca tabu para a geração de horários em universidades baseados em uma implementação prévia [45] do método para coloração de grafos. Uma peculiaridade dessa proposta é a decisão de colocar a violação de algumas restrições fracas fora do espaço de busca (qualquer solução sempre satisfará essas) e tentar então diminuir a ocorrência de conflitos (restrição forte). O teste do métodos em duas instituições mostrou que soluções consideravelmente melhores do que as soluções manuais podem ser obtidas.

Mais especificamente tratando de problemas de programação de horários em escolas, algoritmos baseados em busca tabu foram propostos em [27, 61, 25]. Nos três trabalhos, instâncias de problemas reais foram consideradas.

Na busca tabu de Costa [27] a vizinhança é definida pela mudança no período de alocação de uma determinada aula. A memória de curto prazo é implementada através de duas listas tabu:  $T_1$  e  $T_2$  que consideram, respectivamente, a movimentação de uma aula e a movimentação de uma aula para um determinado período, sendo  $|T_1| < |T_2|$ . O critério de aspiração considera alterações não somente no valor da função objetivo, mas em componentes desta. A diversificação é implementada através da oscilação dos pesos na função objetivo relacionados à ocorrência de conflitos: faz-se uma redução periódica desses pesos permitindo que o processo de busca visite outras regiões do espaço de busca.

No método de Schaerf [61] os movimentos podem envolver a mudança de período de uma aula na agenda de um professor mas também a troca de períodos entre duas aulas. A busca tabu de Schaerf é temperada com componentes aleatórios: após um certo número de iterações sem melhora é realizada a fase de *RNA* (*randomized non-ascendent method*) onde movimentos são sorteados e, caso não gerem conflitos, são realizados. Caso o movimento sorteado gere um conflito, considera-se a reparação desse por meio da exploração (nesse caso, sistemática) de um segundo movimento, gerando os chamados movimentos duplos. O método também considera a utilização de relaxação adaptativa como em [?].

O método híbrido baseado em Busca Tabu e GRASP [59] GBT-II, proposto por Souza [65, 67], destacou-se entre os algoritmos híbridos estudados em [65], onde são apresentadas e avaliadas computacionalmente várias propostas de heurísticas para o PPTC, incluindo *Simulated Annealing*, Busca Tabu e Otimização Microcanônica [71]. Os algoritmos propostos agregavam um método de melhoramento baseado em caminhos mínimos [66] para a pesquisa de movimentos compostos, o qual é utilizado tanto para acelerar a obtenção de soluções factíveis quanto para o melhoramento dessas. A diversificação é implementada através da reinicialização com a fase de construção da metaheurística GRASP. Nesse trabalho, o método híbrido baseado em Busca Tabu superou consideravelmente os métodos baseados em *Simulated Annealing* e Otimização Microcanônica.

Em [60] é apresentado um método de Busca Tabu que utiliza uma diversificação informada que considera dois tipos de memória de longo prazo. O método proposto melhorou os resultados de [65, 67] em todas as instâncias.

### 3.4.1.1 Um Algoritmo de Busca Tabu para o Problema de Programação de Horários em Escolas

Nesta seção será apresentado o algoritmo de melhoramento de quadros de horários BT-PPTC[60], implementado para o PPTC. O método implementa a idéia da metaheurística Busca Tabu de utilização de memórias de curto e longo prazo para o direcionamento de métodos de busca local. Nesse sentido, a composição de diferentes tipos de memórias relacionadas ao PPTC é estudada.

#### Representação e avaliação de soluções:

A solução é armazenada em uma matriz  $Q_{|P| \times |D| \times |H|}$ , onde  $q_{pdh}$  indica se o professor  $p$  no dia  $d$  no período  $h$  está com alguma atividade letiva em alguma turma ( $q_{pdh} \in \{1, \dots, |T|\}$ ) ou está ocioso ( $q_{pdh} = 0$ ). Como exemplo, uma solução parcial é mostrada na Figura 3.9, onde células com “×” indicam indisponibilidade do professor. Como observado em [61], apesar dessa representação ser menos compacta do que uma matriz que armazena as alocações em função

Professor \ Período	1	2	3	4	5	... $ D  \times  H $
1	1	0	0	2	2	...
2	0	×	×	0	1	...
3	×	×	1	0	3	...
4	0	1	0	1	0	...
5	0	0	2	3	×	...

Figura 3.9: Fragmento de um quadro de horários gerado

das turmas, ela facilita a computação da função objetivo - mais ligada às preferências dos professores. Conflitos na agenda de diferentes professores, ou seja, a ocorrência de mais de um professor lecionando para a mesma turma no mesmo período, são permitidos e penalizados na função objetivo, com pesos significativamente maiores do que os pesos atribuídos às restrições fracas. Desse modo, a função objetivo apresentada na seção 2.3.1 é complementada por uma medida de distância da factibilidade. Optou-se por considerar soluções infactíveis dada a dificuldade de geração rápida de soluções que satisfaçam todas as restrições fortes e também porque nos métodos de busca local apresentados na sequência a consideração de apenas soluções factíveis restringiria muito a existência de movimentos válidos. Penaliza-se com peso  $\omega$  o número de professores em conflito em cada período, ou seja, dado que em um dia  $d$  e período  $h$  existam  $c$  professores lecionando para uma dada turma  $t$ , a função objetivo será acrescida do valor  $\max\{c - 1, 0\} \times \omega$ . O mesmo peso  $\omega$  é aplicado para penalizar cada ocorrência de períodos de inatividade na agenda das turmas. Finalmente, cada aula que extrapola o limite diário professor  $\times$  turma é penalizada com peso  $\delta$ .

### O algoritmo BT-PPTC:

O algoritmo BT-PPTC (Algoritmo 2) parte de um quadro de horários  $Q$  e a cada iteração explora toda a vizinhança  $\mathcal{N}(Q)$  para selecionar o próximo movimento  $m$ . A definição de movimento aqui utilizada é a mesma de [61] e envolve a troca de dois valores no quadro de horários de um professor  $p \in P$  nas posições  $(d_1, h_1)$  e  $(d_2, h_2)$  sendo que  $q_{pd_1h_1} \neq q_{pd_2h_2}$  e  $d_1 \times |H| + h_1 < d_2 \times |H| + h_2$ . A complexidade de tempo de exploração de  $\mathcal{N}(Q)$  é  $O(|P| \cdot (|D| \cdot |H|)^2 \cdot EV)$  onde  $EV$  é a complexidade de tempo de avaliação de cada vizinho, o que pode ser feito de forma eficiente considerando somente as mudanças na agenda do professor  $p$  e a possível mudança no número de conflitos em  $(d_1, h_1)$  e  $(d_2, h_2)$ . Em (Algoritmo 2), denota-se por  $\oplus$  a aplicação do movimento.

Sempre que um movimento  $m$  for selecionado, esse será mantido na lista tabu por  $validadeTabu(m)$  iterações. Para dificultar a ocorrência de ciclagem, a implementação proposta não utiliza um valor fixo para  $validadeTabu(m)$ , mas seleciona de modo aleatório entre uma faixa de valores próximos a um valor central ( $validTabu$ ), com desvio permitido controlado pelo parâmetro  $\varphi \in [0, 1]$  (linha 21). Como critério de aspiração considera-se a melhora com relação à melhor solução conhecida. A atribuição de um movimento válido para a variável  $melhorMov$  (linha 6) é necessária visto que não há garantias que a condição da linha 13 será verdadeira em pelo menos uma das iterações do laço das linhas 7-18. Isso pode ser verificado considerando a execução de muitas iterações da busca tabu com  $validTabu = \infty$  e um quadro de horários inicial ótimo.

Como a memória de curto prazo não é suficiente para prevenir que o processo de busca fique preso em determinadas regiões do espaço de busca [40], alguma estratégia de diversificação deve ser empregada. No método proposto a memória de longo prazo é utilizada para guiar o procedimento de diversificação. Com essa estratégia pretende-se evitar a perda de informação decorrente do reinício aleatório, bem como diminuir o risco de que áreas já visitadas do espaço de busca sejam novamente exploradas.

Dois tipos de memórias de longo prazo são propostas, o primeiro tipo armazena medidas de transição, contando a frequência de movimentos envolvendo cada professor  $\times$  turma, já o segundo tipo armazena medidas de residência, contando o número de iterações que cada aula permaneceu em determinado dia e período. A cada movimento feito a memória de longo prazo é atualizada (linha 19), sendo reinicializada sempre que a solução incumbente for renovada (linha 25).

Enquanto o processo de diversificação estiver ativo a informação das memórias de longo prazo será utilizada para guiar a seleção de movimentos, de modo que movimentos em quadros de horários de professores onde poucas mudanças foram feitas ou movimentos fazendo alocações ainda não experimentadas serão encorajados, por exemplo. Isso é feito através da incorporação de penalidades na avaliação de movimentos. Nos parágrafos seguintes a computação dessas penalidades será detalhada.

**Memória de Longo Prazo baseada em Transição:** nesse tipo de memória as medidas de transição são armazenadas em uma matriz  $\hat{M}_{|P|\times|T|}$  onde  $\hat{m}_{pt}$  indica quantos movimentos foram realizados envolvendo o professor  $p$  e a turma  $t$ . Usando esses valores, as taxas de transição  $\hat{t}_{pt}$  são calculadas:

$$\hat{t}_{pt} = \frac{\hat{m}_{pt}}{\max\{\hat{m}_{pt}|p \in P, t \in T\}}$$

Uma vez que um movimento pode envolver duas aulas ou uma aula e um período livre, a penalidade  $\hat{p}_{pa_1a_2}$  relacionada com um movimento na agenda de um professor  $p$ , nas posições  $(d_1, h_1)$  e  $(d_2, h_2)$  com conteúdos  $a_1 = q_{pd_1h_1}$  e  $a_2 = q_{pd_2h_2}$ , considerando o custo  $f(Q)$  da solução corrente é:

$$\hat{p}_{pa_1a_2} = \begin{cases} \hat{t}_{pa_1} \times f(Q) & \text{se } a_1 \neq 0 \text{ e } a_2 = 0 \\ \hat{t}_{pa_2} \times f(Q) & \text{se } a_1 = 0 \text{ e } a_2 \neq 0 \\ (\hat{t}_{pa_1} + \hat{t}_{pa_2})/2 \times f(Q) & \text{se } a_1 \neq 0 \text{ e } a_2 \neq 0 \end{cases}$$

**Memória de Longo Prazo baseada em Residência:** nesse tipo de memória medidas de residência são armazenadas em uma matriz  $\hat{M}_{|P|\times|T|\times|D|\times|H|}$ <sup>1</sup> sendo que  $\hat{m}_{ptdh}$  indica quantas iterações no decorrer da busca contaram com alguma aula do professor  $p$  para a turma  $t$  no dia  $d$  no período  $h$ . Apesar de ser uma matriz com 4 dimensões, deve-se observar que a mesma é, em geral, bastante esparsa, de modo que implementações eficientes a tornam bastante prática para os problemas considerados nesse trabalho. Computa-se a taxa de residência  $\hat{r}_{ptdh}$  como:

$$\hat{r}_{ptdh} = \frac{\hat{m}_{ptdh}}{\max\{\hat{m}_{ptdh}|p \in P, t \in T, d \in D, h \in H\}}$$

<sup>1</sup>Essa estrutura é ligeiramente diferente de [60]. Como se observou em experimentos preliminares, resultados semelhantes foram obtidos comparando-se com a implementação prévia, sendo que a estrutura aqui utilizada é mais compacta.

Desse modo, computa-se a penalidade  $\dot{p}_{ptdh}$  de alocação de alguma aula do professor  $p$  para a turma  $t$  no dia  $d$  e período  $h$  como:

$$\dot{p}_{ptdh} = \dot{r}_{ptdh} \times f(Q)$$

Para movimentos envolvendo duas aulas no quadro de horários de um dado professor a penalidade considerada será a média das penalidades das aulas envolvidas.

A estratégia de diversificação é aplicada sempre que sinais de entrincheiramento regional são detectados. Nesse caso, o número de iterações sem melhora é avaliado antes do início da estratégia de diversificação. O número de iterações sem melhora necessários para ativar a diversificação (*ativDiv*) e o número de iterações que a diversificação ficará ativada (*itDiv*) são parâmetros do algoritmo. O processo é cíclico e reinicia sempre que um múltiplo *ativDiv* iterações sem melhora são processadas. Movimentos realizados nessa fase podem ser vistos como *movimentos de influência* [40], uma vez que o seu objetivo é modificar a solução de um modo “influyente” (não aleatório). O cálculo da penalidade utilizada na diversificação (linha 10) pode considerar uma das (ou ambas) memórias de longo prazo.

```

Entrada:  $Q, \text{ativDiv}, \text{itDiv}, \text{validTabu}, \varphi$ 
Saída:  $Q^*$ 
1  $Q^* \leftarrow Q; \text{ListaTabu} \leftarrow \emptyset;$ 
2  $\text{semMelhora} \leftarrow 0; \text{it} \leftarrow 0;$ 
3 inicialize memória de longo prazo;
4 repita
5    $\Delta \leftarrow \infty; \text{it} ++;$ 
6    $\text{melhorMov} \leftarrow$  movimento aleatório;
7   para cada movimento  $m$  tal que  $Q \oplus m \in \mathcal{N}(Q)$  faça
8      $\text{penalidade} \leftarrow 0;$ 
9     se  $\text{semMelhora} \bmod \text{ativDiv} < \text{itDiv}$  e  $\text{it} \geq \text{ativDiv}$  então
10      calcule  $\text{penalidade}$  para o movimento  $m$ ;
11    fim
12     $\Delta' \leftarrow f(Q \oplus m) - f(Q);$ 
13    se  $(\Delta' + \text{penalidade} < \Delta)$  e  $(m \notin \text{ListaTabu})$  ou  $(f(Q \oplus m) <$ 
14       $f(Q^*)$  e  $\Delta' < \Delta)$  então
15       $\text{melhorMov} \leftarrow m;$ 
16       $\Delta \leftarrow \Delta';$ 
17      se  $f(Q \oplus m) \geq f(Q^*)$  então  $\Delta \leftarrow \Delta + \text{penalidade}$ 
18    fim
19  atualize memória de longo prazo ;
20   $Q \leftarrow Q \oplus \text{melhorMov};$ 
21   $\text{validadeTabu}(\text{melhorMov}) \leftarrow$ 
22     $\text{random}([\text{validTabu} - \varphi \times \text{validTabu}], [\text{validTabu} + \varphi \times \text{validTabu}]) + \text{it};$ 
23  atualize  $\text{ListaTabu}$ ;
24  se  $f(Q) < f(Q^*)$  então
25     $Q^* \leftarrow Q; \text{semMelhora} \leftarrow 0;$ 
26    reinicialize memória de longo prazo;
27  senão
28     $\text{semMelhora} ++;$ 
29  fim
até critério de parada atingido ;

```

Algoritmo 2: Algoritmo BT-PPTC

### 3.4.2 *Simulated Annealing*

O método *Simulated Annealing* foi apresentado no contexto de otimização em [49]. Baseado em busca local, assim como busca tabu, também considera a realização de movimentos que pioram o valor da função objetivo. Nesse caso, no entanto, um componente aleatório ao invés da utilização de memória é considerado para decidir a aceitação do movimento. A probabilidade de aceitação de movimentos desse tipo diminui gradualmente no decorrer da busca. O Algoritmo 3 apresenta uma versão simplificada de uma implementação comum do método de *Simulated Annealing*.

<p><b>Entrada:</b> <math>s^0</math> // solução inicial  <b>Saída:</b> <math>s^*</math> // melhor solução encontrada</p> <pre> 1 determine a temperatura inicial <math>t^0</math>; 2 <math>t \leftarrow t^0</math> <math>s \leftarrow s^0</math>; 3 <math>s^* \leftarrow s^0</math>; 4 <b>enquanto</b> (<i>Critério de parada não satisfeito</i>) <b>faça</b> 5   seleccione uma solução <math>s' \in \mathcal{N}(s)</math>; 6   calcule <math>\Delta f = f(s') - f(s)</math>; 7   calcule um número randômico uniforme <math>r</math> no intervalo <math>[0, 1]</math>; 8   <b>se</b> (<math>\Delta f &lt; 0</math>) <b>ou</b> (<math>r &lt; e^{-\Delta f/t}</math>) <b>então</b> 9     <math>s \leftarrow s'</math>; 10  <b>fim</b> 11  <b>se</b> <math>f(s) &lt; f(s^*)</math> <b>então</b> 12    <math>s^* \leftarrow s</math>; 13  <b>fim</b> 14  atualize a temperatura <math>t</math>; 15 <b>fim</b> </pre>
---

**Algoritmo 3:** Algoritmo de *Simulated Annealing*

Abramson [4] realiza experimentos com versões seqüenciais e paralelas do *Simulated Annealing*. O conjunto de problemas teste consistia em problemas artificiais, gerados de modo que a solução ótima fosse conhecida (sem conflitos, visto que havia sido a única restrição considerada) e um problema real, com algumas restrições adicionais. A versão paralela do método de *Simulated Annealing*, que consistia em uma implementação com granularidade fina [29], implementada em um sistema paralelo com memória compartilhada, apresentou uma aceleração considerável.

### 3.4.3 Algoritmos Evolutivos

Algoritmos Evolutivos, juntamente com Algoritmos Genéticos[46], o representante mais conhecido desses métodos, inspiram-se na teoria da evolução e genética. Esses algoritmos tentam simular alguns aspectos da seleção natural de Darwin. O Algoritmo 4 apresenta uma versão simplificada de uma implementação comum do método de *Simulated Annealing*.

Vários algoritmos genéticos foram propostos para a resolução de problemas de programação de horários. Abramson e Abela [5] propuseram um algoritmo genético paralelo para a resolução de problemas de programação de horários em

<pre> <b>Entrada:</b> <math>P</math> // população inicial <b>Saída:</b> <math>s^*</math> // melhor solução encontrada <math>s^* \leftarrow \arg \min_{s \in P} f(s);</math> 1 2 <b>enquanto</b> (<i>Critério de parada não satisfeito</i>) <b>faça</b> 3   <b>para</b> <math>i \leftarrow 1</math> <b>to</b> <math>nrFilhos</math> <b>faça</b> 4     selecione os pais <math>s, s' \in P;</math> 5     <math>s''' \leftarrow \text{cruzamento}(s', s'');</math> 6     <math>P \leftarrow P \cup \{s'''\};</math> 7     <b>se</b> <math>f(s''') &lt; f(s^*)</math> <b>então</b> 8       <math>s^* \leftarrow s''';</math> 9     <b>fim</b> 10    aplique mutação em alguns membros de <math>P;</math> 11    <math>s^* \leftarrow \arg \min_{s \in P} f(s);</math> 12  <b>fim</b> 13  atualize a população <math>P;</math> 14 <b>fim</b> </pre>
---

**Algoritmo 4:** Algoritmo Genético

escolas. No problema considerado, somente leva-se em conta a não existência de conflitos. Experimentos com instâncias artificiais são apresentados. O algoritmo genético híbrido de Colorni, Dorigo e Maniezzo [25] considera um problema com mais requisições reais. Nesse algoritmo a combinação de operadores genéticos que consideram características do problema e uma busca local em múltiplas vizinhanças é utilizada. Comparações com implementações bastante simples de *Simulated Annealing* e Busca Tabu são apresentadas, sendo que os melhores resultados foram apresentados pelo algoritmo genético híbrido e pela busca tabu. Um algoritmo genético híbrido também é apresentado em [32].

## Capítulo 4

# Produção de Quadros de Horários Provavelmente Ótimos

Nesta capítulo serão estudadas técnicas de Programação Linear Inteira Mista (PLIM), as quais permitem a obtenção de quadros de horários provavelmente ótimos. Para o leitor não familiarizado, os seguintes livros apresentam uma boa introdução ao tema:

Título: Otimização Linear  
Autor: Nelson Maculan e Márcia H. Costa Fampa  
Editora: UnB  
Ano: 2006

Título: Integer Programming  
Autor: Laurence A. Wolsey  
Editora: Wiley-Interscience  
Ano: 1998

A utilização de Programação Linear Inteira Mista permite que inúmeros problemas de programação de horários sejam fielmente modelados, incluindo o PPTC apresentado na seção 2.3.1, formulado como PLIM na formulação  $\mathcal{F}_1$  e vários outros problemas de universidades de escolas [8, 11, 31, 54]. Uma vez modelados, em princípio, esses problemas podem ser resolvidos de maneira exata através da utilização de pacotes de *software* resolvidores de PLIM, muitos dos quais estão livremente disponíveis[69] na Internet.

Considerando que a prova da otimalidade pode implicar a resolução de um número computacionalmente intratável de problemas lineares, trabalhos iniciais como [51] tratavam apenas parte do problema hoje conhecido como o problema de programação de horários. Em [51], além do problema considerar apenas a não ocorrência de conflitos, cabia ao usuário do software informar previamente quais cursos podiam ser oferecidos concomitantemente, através da criação de “arranjos”. Em [33] a formulação compacta apresentada, com variáveis binárias, permitiu somente a resolução de instâncias artificiais de menor dimensão, considerando o tamanho de problemas reais típicos.

Mais recentemente, progressos[48] na capacidade de resolução de problemas de programação de Programação Linear Inteira Mista (PLIM) trouxeram um renovado interesse na aplicação de PLIM para a resolução de problemas de programação de horários e permitiram que problemas reais com importantes considerações práticas fossem tratados em [8, 31, 11]. Uma particularidade dos modelos utilizados em [8, 11], bastante relacionados com o PPTC, é o fato de que a questão da compacidade de horários para professores não é avaliada na função objetivo. Em [11], por exemplo, especifica-se um conjunto de dias de folga para cada professor, já em [8] especifica-se um número máximo de dias em que o professor deve comparecer à instituição de ensino. Essa abordagem, apesar de possibilitar uma considerável diminuição na dimensão do espaço de busca, traz a desvantagem de que na especificação do número de dias de folga/trabalho de cada professor devem ser feitas escolhas sábias: a requisição de horários tão compactos quanto possível considerando cada professor isoladamente pode resultar em um problema sem solução viável, que somente poderá tornar-se viável através da relaxação desse requisito para alguns professores. A alteração manual do modelo para permitir a factibilidade e oferecer quadros de horários compactos para professores pode excluir soluções ótimas quanto a esse requisito.

Em [56] uma aplicação de Geração de Colunas [30] para um modelo com um número exponencial de variáveis para uma variante do PPT encontrada na Grécia é apresentada. Nesse modelo, cada variável binária representa um possível quadro de horários semanal de um professor, sendo que a maioria das restrições é considerada na geração das colunas, ou seja, tem-se quase que exclusivamente restrições fortes, de modo que a função objetivo tem poucos (2) componentes. A determinação de colunas atrativas é realizada através da busca no espaço de soluções factíveis com a estratégia de mais-profundo-primeiro. O procedimento procede iterativamente com a adição de novas colunas enquanto alguma melhora significativa na função objetivo é observada. Uma vez que a solução resultante do procedimento apresentado geralmente não é inteira, os autores desenvolveram um procedimento específico que tenta factibilizar a integralidade através da fixação de alguns quadros de horários de professores. Deve-se observar que o procedimento proposto por esses autores é heurístico, sem garantia de produção da solução ótima.

## 4.1 Cortes

O desempenho dos resolvidores de PLIM depende fortemente da formulação utilizada. Em particular, observa-se que algumas formulações apresentam limites duais bastante apertados, essas formulações são ditas “fortes”. A disponibilidade de limites apertados possibilita uma poda mais efetiva da árvore de busca, facilitando, geralmente, o trabalho dos resolvidores PLIM.

Uma maneira de melhorar formulações PLIM é através da adição de cortes. Cortes são restrições válidas para a formulação original do PLIM, mas que removem (cortam) alguma solução fracionária válida para a formulação sem as restrições de integralidade.

A seguir, são apresentados alguns cortes para reforçar a formulação  $\mathcal{F}_1$  (seção 2.3.1):

Primeiramente, cortes que tratam da alocação de **aulas geminadas** são apresentados. Considere qualquer seqüência de períodos contíguos  $h', \dots, h''$

na agenda de um professor  $p$  em um dia  $d$  tal que  $h'' - h' + 1$  seja ímpar,  $\sum_{h=h'}^{h''} \tilde{p}_{pdh} = h'' - h' + 1$  e  $\{h', \dots, h'' - 1\} \subseteq \widehat{G}_{pd}$ . Nesse caso, o seguinte corte pode ser utilizado:

$$\sum_{t \in T} \sum_{h=h'}^{h''-1} y_{ptdh} \leq v_{pd} \left\lfloor \frac{|\{h', \dots, h''\}|}{2} \right\rfloor \quad (4.1)$$

De modo análogo, considerando a grade de horários de uma turma  $t$  em um dia  $d$  e uma seqüência de períodos  $h', \dots, h''$  a seguinte desigualdade é válida para o PPTC:

$$\sum_{p \in P} \sum_{h \in \widehat{G}_{pd} \cap \{h', \dots, h'' - 1\}} y_{ptdh} \leq \left\lfloor \frac{|\{h', \dots, h''\}|}{2} \right\rfloor \quad (4.2)$$

Ainda, considerando que somente aulas geminadas ou simples são consideradas no PPTC, observa-se que a alocação de alguma aula em determinados períodos impede a alocação de aulas geminadas em outros. Desse modo, a consideração de uma aula de um professor  $p$  em um dia  $d$  para uma turma  $t$  em um horário  $h$  gera o seguinte corte:

$$\sum_{h_1 \in \widehat{G}_{pd} \setminus \{h, h-1\}} y_{ptdh_1} + x_{ptdh} \leq v_{pd} \quad (4.3)$$

Agora, serão apresentados cortes relacionados com as variáveis que indicam a atividade diária do professor, bem como sua soma indicando o número de dias com alguma alocação na agenda do professor. Primeiramente será apresentado o corte proposto em [65], que considera o número mínimo de dias que cada professor deve comparecer à escola:

$$\sum_{d \in D} v_{pd} \geq \max \left\{ \left\lceil \frac{\sum_{t \in T} \tilde{r}_{pt}}{|H|} \right\rceil, \max_{t \in T} \left\lceil \frac{\tilde{r}_{pt}}{\tilde{m}_{pt}} \right\rceil \right\} \quad \forall p \in P \quad (4.4)$$

Observa-se que o limite inferior  $\left\lceil \frac{\sum_{t \in T} \tilde{r}_{pt}}{|H|} \right\rceil$  para o número de dias letivos para o professor  $d$  pode ser melhorado na situação específica da ocorrência de dias na agenda do professor com disponibilidade parcial. Dado um professor  $p$  e um dia  $d$  em sua agenda tal que  $\sum_{h \in H} \tilde{p}_{pdh} < |H|$ , seja  $\dot{p}_{pk}$  o número de períodos disponíveis no  $k$ -ésimo dia com maior disponibilidade do professor  $p$ , e  $\bar{k}$  como o menor inteiro tal que  $\sum_{k=1}^{\bar{k}} \dot{p}_{pk} \geq \sum_{t \in T} \tilde{r}_{pt}$ , as seguintes desigualdades são válidas para o PPTC:

$$\begin{aligned} \sum_{d \in D} v_{pd} &\geq \bar{k} \quad \forall p \in P, \bar{k} \in D \\ \left| \sum_{k=1}^{\bar{k}} \dot{p}_{pk} \geq \sum_{t \in T} \tilde{r}_{pt} \wedge (\bar{k} - 1 = 0 \vee \sum_{k=1}^{\bar{k}-1} \dot{p}_{pk} < \sum_{t \in T} \tilde{r}_{pt}) \right. & \quad (4.5) \end{aligned}$$

Claramente, a utilização do lado direito de 4.5 na primeira parcela da função max em 4.4 produz cortes que dominam os anteriormente propostos.

O número de dias ocupados na agenda do professor previsto no corte anterior corresponde à situação de ocupação máxima dos dias na agenda com maior disponibilidade, que é um limite inferior para o número de dias necessário para o cumprimento das atividades letivas do professor. Irá se considerar agora a inclusão de um número qualquer de aulas em dias com menor disponibilidade. Considere novamente um professor  $p$  e o menor inteiro  $\bar{k}$  tal que  $\sum_{k=1}^{\bar{k}} \dot{p}_{pk} \geq \sum_{t \in T} \tilde{r}_{pt}$ . Considere ainda um dia  $d'$  tal que  $\sum_{k=1}^{\bar{k}-1} \dot{p}_{pk} + \sum_{h \in H} \tilde{p}_{pd'h} < \sum_{t \in T} \tilde{r}_{pt}$ , propõe-se assim a seguinte desigualdade válida para o PPTC:

$$\begin{aligned} \sum_{d \in D} v_{pd} &\geq \sum_{t \in T} (\bar{k} + 1) x_{ptd'h} \\ &\quad \forall p \in P, d' \in D \\ \left| \sum_{d''=1}^{\bar{k}} \dot{p}_{pd''} \geq \sum_{t \in T} \tilde{r}_{pt} \wedge (\bar{k} - 1 = 0 \vee \sum_{d''=1}^{\bar{k}-1} \dot{p}_{pd''} < \sum_{t \in T} \tilde{r}_{pt}) \right. & \quad (4.6) \\ \left. \sum_{k''=1}^{\bar{k}-1} \dot{p}_{k''} + \sum_{h \in H} \tilde{p}_{pd'h} < \sum_{t \in T} \tilde{r}_{pt} \right) & \end{aligned}$$

Como exemplo ilustrativo de um corte que pode ser gerado a partir de 4.6, considere um professor  $t$  que deve lecionar 13 horas aula na semana. Considere ainda que a disponibilidade diária desse professor em uma grade horários com 5 dias é apresentada no conjunto ordenado  $\{\dot{p}_{t1}, \dots, \dot{p}_{t5}\}$ :  $\{5, 5, 4, 2, 2\}$ , ou seja, o professor  $t$  tem 2 dias com disponibilidade de 5 períodos, 2 dias com disponibilidade de 2 períodos e um dia com disponibilidade de 4 períodos. Um quadro de horários compacto, que ocupe um número mínimo de dias para esse professor (3 dias), pode ser gerado se as alocações de suas aulas forem feitas nos dias com maior disponibilidade, no caso, os dias em que ele tem disponibilidade maior ou igual a 4 períodos. Considere agora a ocorrência de uma alocação em um dos dias com disponibilidade mais baixa, 2 no caso. Essa situação implicará em um quadro de horários com no mínimo 4 dias letivos com alguma alocação para esse professor.

O limite mínimo para o número de dias de trabalho de um professor também pode ser ligado à ativação das variáveis  $x_{ptdh}$  do seguinte modo:

$$\tilde{m}_{pt} v_{pd} \geq \sum_{h \in H} x_{ptdh} \quad \forall p \in P, t \in T, d \in D \quad (4.7)$$

Outras variáveis e restrições podem ser incluídas para fortalecer a formulação. A medição dos buracos na agenda do professor, além de incluir as restrições 2.13, 2.12 e 2.14 (seção 2.3.1), pode incluir as seguintes restrições, que medem a ocorrência de buracos de tamanho 1, envolvendo três períodos contíguos, nas variáveis  $\check{b}_{pdh}$ , onde  $\check{b}_{pdh}$  indica a existência de um buraco de tamanho 1, gerado pela alocação de aulas nos períodos  $h$  e  $h + 2$ , juntamente com a falta de alguma alocação no período  $h + 1$ , para um professor  $p$  em um dia  $d$ . Considere o conjunto de períodos  $\widehat{B}_{pd}$  que denota os períodos da agenda de um professor

$p$  no dia  $d$  que podem sediar a aula inicial que ocasionará um buraco com as características anteriores, ou seja, para todo  $h \in \widehat{B}_{pd}$  se tem  $\tilde{p}_{pdh} = \tilde{p}_{pdh+2} = 1$ , as seguintes desigualdades são válidas para o PPTC :

$$\check{b}_{pdh} \geq \sum_{t \in T} (x_{ptdh} + x_{ptdh+2} - x_{ptdh+1}) - 1 \quad \forall p \in P, d \in D, h \in \widehat{B}_{pd} \quad (4.8)$$

$$b_{pd} \geq \sum_{h \in \widehat{B}_{pd}} \check{b}_{pdh} \quad \forall p \in P, d \in D \quad (4.9)$$

## 4.2 Formulação Estendida com Geração de Colunas e Cortes

Nesta seção será apresentada uma formulação alternativa para o PPTC que contém um número exponencial de variáveis. A utilização de uma formulação desse tipo se faz importante devido ao fato de que, em muitos casos, essas formulações oferecem relaxações lineares mais próximas da envoltória convexa das soluções factíveis, bem como são menos propensas a problemas de simetria [10]. Tal formulação é posteriormente complementada com a adição de cortes, resultando em uma formulação com um número exponencial de colunas e linhas. Como observado em [58], a união das técnicas de geração de colunas e cortes de modo eficiente não é uma tarefa trivial, dada a forte relação entre esses dois problemas. Finalmente, o tratamento computacional de tal formulação é apresentado nesta seção, com a descrição das técnicas que tornam possível sua utilização em instâncias reais do PPTC.

Na formulação proposta, aqui denotada por  $\mathcal{F}_2$ , são considerados os possíveis padrões de alocação que podem ocorrer em dias letivos de cada professor. Considere, além dos dados disponíveis na seção 2.3.1, que  $\check{P}_p$  seja o conjunto de padrões possíveis de alocação para o professor  $p$  em um dia supondo disponibilidade total e que  $\check{x}_{pjth} = 1$  indique que professor  $p$  no padrão de alocação  $j$ , no período  $h$  tem alocação para a turma  $t$  ou  $\check{x}_{pjth} = 0$  caso contrário. O custo do padrão de alocação  $j$  para o professor  $p$  é denotado por  $c_{pj}$  e engloba, nesse caso, a computação da existência de buracos na agenda do professor e, caso o padrão de alocação não seja vazio, o custo de um dia adicional de trabalho. Considere também o número de aulas geminadas  $\check{g}_{pjt}$  existentes para o professor  $p$  na turma  $t$  no padrão de alocação  $j$ . A variável de decisão  $\lambda_{pjd}$  indica se o padrão de alocação  $j$  será selecionado para o dia  $d$  na agenda do professor  $p$  ( $\lambda_{pjd} = 1$ ) ou não ( $\lambda_{pjd} = 0$ ). Deve-se observar que a existência de um dia  $d'$  onde um professor  $p$  tem apenas um subconjunto de períodos disponíveis irá implicar na existência de um conjunto menor de combinações ( $p \in P, j \in \check{P}_p$ ) válidas para esse dia, já que nem todas as alocações contidas nos padrões considerados podem ser compatíveis com a disponibilidade em questão, denota-se por  $\check{P}_{pd}$  esse subconjunto de  $\check{P}_p$ . Desse modo, apresenta-se a formulação  $\mathcal{F}_2$  para o PPTC:

$$\text{minimizar} \quad \sum_{p \in P} \sum_{d \in D} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot c_{pj} + \sum_{p \in P} \sum_{t \in T} w_p''' \cdot g_p \quad (4.10)$$

sujeito a

$$\sum_{j \in \check{P}_{pd}} \lambda_{pjd} = 1 \quad \forall p \in P, d \in D \quad (4.11)$$

$$\sum_{p \in P} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot \check{x}_{pjth} \leq 1 \quad \forall t \in T, d \in D, h \in H \quad (4.12)$$

$$\sum_{j \in \check{P}_{pd}} \sum_{d \in D} \sum_{h \in H} \lambda_{pjd} \cdot \check{x}_{pjth} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \quad (4.13)$$

$$\sum_{d \in D} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot \check{g}_{pjt} + g_{pt} \geq \tilde{g}_{pt} \quad \forall p \in P, t \in T \quad (4.14)$$

$$\lambda_{pjd} \in \{0, 1\} \quad \forall p \in P, d \in D, j \in \check{P}_{pd} \quad (4.15)$$

$$g_{pt} \in \mathbb{N} \quad \forall p \in P, t \in T \quad (4.16)$$

Onde 4.11, a restrição de convexificação, obriga a seleção de um padrão de para cada professor e dia. Apesar de que o número de variáveis envolvidas em  $\mathcal{F}_2$  pode ser proibitivamente grande <sup>1</sup>, a resolução da relaxação linear de  $\mathcal{F}_2$ , denotada aqui por  $\mathcal{F}_{2r}$ , pode ser realizada sem a utilização explícita de todas as variáveis envolvidas através da técnica de Geração de Colunas [30]. Nessa técnica, inicia-se com um subconjunto pequeno de variáveis e a cada iteração, utilizando-se informação da solução dual do problema, são identificadas variáveis promissoras, através do custo reduzido. Essas variáveis são inseridas progressivamente até que não existam mais variáveis que permitam a melhora da solução.

### 4.2.1 Geração de Colunas

Inicialmente, para a resolução de  $\mathcal{F}_{2r}$  são inseridas somente colunas (variáveis  $\lambda$ ) que correspondem a dias de inatividade e variáveis artificiais, com custo suficientemente alto, que permitem a factibilização da solução inicial considerando as restrições existentes, como o atendimento da carga horária.

O problema de *pricing* consiste em encontrar, para cada professor, considerando cada respectivo dia da semana com com alguma disponibilidade, o padrão de alocação com menor custo reduzido. Essa computação utiliza variáveis duais associadas às restrições envolvidas. Denotam-se aqui por  $\mu$ ,  $\nu$ ,  $\pi$  e  $\kappa$  as variáveis duais associadas com as restrições 4.11, 4.12, 4.13 e 4.14, respectivamente. Considere a descoberta de uma coluna com custo reduzido negativo para um professor  $p$  em um dia  $d$ , cujo conteúdo é expresso em variáveis  $\check{x}_{tth}$  ( $t \in T, h \in H$ ), sendo que  $\check{x}_{tth} = 1$  caso o professor  $p$  esteja alocado para uma aula com a turma  $t$  no dia  $d$  e período  $h$ ,  $\check{x}_{tth} = 0$  caso contrário. Desse modo, o problema de *pricing* relacionado ao professor  $p$  e dia  $d$ , aqui denotado por  $\mathcal{P}_{pd}$  pode ser formulado como um PPLIM:

<sup>1</sup>Em nossos experimentos a utilização explícita da formulação  $\mathcal{F}_2$  só foi possível para as instâncias 1 e 2, mesmo considerando somente a resolução do problema linear relaxado, visto que o resolvidor CPLEX 10 abortou sua execução prematuramente para as demais instâncias em uma máquina com 1GB de RAM.

$$\mathcal{P}_{pd} = \left\{ \begin{array}{l}
 \text{minimizar} \quad \bar{c}_{pd} = \check{c}_{pd} - \check{d}_{pd} \\
 \text{sujeito a} \\
 \sum_{h \in H} \check{x}_{th} \leq \min\{\check{m}_{pt}, \check{r}_{pt}\} \quad \forall t \in T \\
 \check{c}_{pd} = \check{b}.w'_p + \check{v}.w''_p \\
 \check{d}_{pd} = \mu_{pd} + \sum_{t \in T} \sum_{h \in H} \nu_{tdh} \cdot \check{x}_{th} + \sum_{t \in T} \sum_{h \in H} \cdot \pi_{pt} \cdot \check{x}_{th} + \sum_{t \in T} \sum_{h \in \hat{G}_{pd}} \check{y}_{th} \cdot \kappa_{pt} \\
 \bar{a} \leq \sum_{t \in T} (+h - |H|) \cdot \check{x}_{th} + |H| \quad \forall h \in H \\
 \underline{a} \geq \sum_{t \in T} h \cdot \check{x}_{th} \quad \forall h \in H \\
 \check{b} = 1 + \underline{a} - \bar{a} - \sum_{t \in T} \sum_{h \in H} \check{x}_{th} \\
 \sum_{t \in T} \check{x}_{th} \leq \check{p}_{pdh} \cdot \check{v} \quad h \in H \\
 \check{y}_{th} \leq \check{x}_{th} \quad \forall t \in T, h \in \hat{G}_{pd} \\
 \check{y}_{th} \leq \check{x}_{th+1} \quad \forall t \in T, h \in \hat{G}_{pd} \\
 \check{y}_{th} \geq \check{x}_{th} + \check{x}_{th+1} - 1 \quad \forall t \in T, h \in \hat{G}_{pd} \\
 \check{x}_{th} \in \{0, 1\} \quad \forall t \in T, h \in H \\
 \check{y}_{th} \in \{0, 1\} \quad \forall t \in T, h \in \hat{G}_{pd} \\
 \check{v} \in \{0, 1\} \\
 \bar{a} \in \mathbb{N} \\
 \underline{a} \in \mathbb{N} \\
 \check{b} \in \mathbb{N}
 \end{array} \right.$$

Problemas  $\mathcal{P}_{pd} \forall p \in P, d \in D$  são resolvidos a cada iteração, aumentando o problema em no máximo  $|P| \times |D|$  colunas por iteração. A solução ótima para  $\mathcal{F}_{2r}$  é obtida quando  $\nexists \bar{c}_{pd} < 0 \forall p \in P, d \in D$ . Apesar de que, de nosso conhecimento, não estejam disponíveis algoritmos que resolvam  $\mathcal{P}_{pd}$  de maneira eficiente (com complexidade de tempo polinomial), uma observação deve ser feita sobre a praticidade da utilização do problema de *pricing* acima apresentado: as dimensões do espaço de busca são relacionadas diretamente ao número de períodos ( $|H|$ ). Em instâncias reais do PPTC o valor de  $|H|$  é bastante limitado, assim como os valores de  $\check{m}_{pt}$ , por questões pedagógicas, visto que as aulas são distribuídas no decorrer da semana.

### 4.2.2 Geração de Cortes

A formulação  $\mathcal{F}_2$  permite a definição de novos cortes que não seriam possíveis com a formulação  $\mathcal{F}_1$ . Os cortes que serão apresentados se referem ao fechamento da carga horária semanal, considerando escolhas diárias para a carga horária de professores, bem como escolhas diárias para a carga horária de encontros entre professores e turmas. Primeiramente vamos estender  $\mathcal{F}_2$  com novas variáveis e restrições que facilitarão a definição dos cortes. Essa formulação estendida, denotada aqui por  $\mathcal{F}'_2$  incorpora variáveis  $\check{a}_{pdh}$  que indicam se o professor  $p$  irá lecionar exatamente  $h$  aulas em um dia  $d$  ( $\check{a}_{pdh} = 1, \check{a}_{pdh} = 0$  c.c.), bem como variáveis  $\check{a}_{ptdh}$  que indicam se o professor  $p$  irá lecionar exatamente  $h$  aulas para a turma  $t$  em um dia  $d$  ( $\check{a}_{ptdh} = 1, \check{a}_{ptdh} = 0$  c.c.). Seja  $\check{P}_{pdh}$  o subconjunto de colunas de  $\check{P}_{pd}$  tal que o número total de aulas em cada coluna de  $\check{P}_{pdh}$  seja exatamente  $h$  e seja  $\check{P}_{ptdh}$  o subconjunto de  $\check{P}_{pd}$  contendo todas as colunas com

exatamente  $h$  aulas do professor  $p$  para a turma  $t$ , então as restrições seguintes, 4.17, 4.18 e 4.19, controlam a seleção única de uma carga de trabalho diária para os professores e a amarração de tal escolha com as colunas  $\lambda$ , já as restrições 4.20, 4.21 e 4.22 garantem a seleção da carga horária para encontros diários entre professores e turmas e a relação dessas escolhas com as variáveis  $\lambda$ , como segue:

$$\sum_{h \in \{0, \dots, |H|\}} \check{\alpha}_{pdh} = 1 \quad \forall p \in P, d \in D \quad (4.17)$$

$$\sum_{j \in \check{P}_{pdh}} \lambda_{pjd} = \check{\alpha}_{pdh} \quad \forall p \in P, d \in D, h \in H \quad (4.18)$$

$$\check{\alpha}_{pdh} \in \{0, 1\} \quad \forall p \in P, d \in D, h \in H \quad (4.19)$$

$$\sum_{h \in \{0, \dots, |H|\}} \check{\alpha}_{ptdh} = 1 \quad \forall p \in P, t \in T, d \in D \quad (4.20)$$

$$\sum_{j \in \check{P}_{ptdh}} \lambda_{pjd} = \check{\alpha}_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (4.21)$$

$$\check{\alpha}_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (4.22)$$

As restrições 4.17 e 4.20 não são necessárias, considerando a seleção única de um padrão de trabalho diário para cada professor, mas foram colocadas por motivos de clareza.

O problema de *pricing* relativo à formulação  $\mathcal{F}'_2$ , aqui denotado por  $\mathcal{P}'_{pd}$ , considerando um professor  $p$  e dia  $d$ , deve ser atualizado com relação a  $\mathcal{P}_{pd}$  visto que as colunas  $\lambda$  aparecem agora em duas restrições adicionais.  $\mathcal{P}'_{pd}$  consiste em  $\mathcal{P}_{pd}$  aumentado das variáveis  $\check{\alpha}_h \in \{0, 1\} \forall h \in \{0, \dots, |H|\}$ , que indicam a seleção de uma carga horária total  $h$  para um professor  $p$  em um dia  $d$  ( $\check{\alpha}_h = 1, \check{\alpha}_h = 0$  c.c.), bem como as variáveis  $\check{\alpha}_{th} \forall t \in T, h \in \{0, \dots, |H|\}$ , que indicam a seleção de uma carga horária diária com  $h$  aulas para o professor  $p$  em uma turma  $t$  em um dia  $d$  ( $\check{\alpha}_{th} = 1, \check{\alpha}_{th} = 0$  c.c.). As novas variáveis são amarradas com a escolha do padrão de trabalho com as seguintes restrições:

$$\sum_{h \in H} h \cdot \check{\alpha}_h = \sum_{h \in H} \sum_{t \in T} \check{x}_{th} \quad (4.23)$$

$$\sum_{h \in \{0, \dots, H\}} \check{\alpha}_h = 1 \quad (4.24)$$

$$\check{\alpha}_h \in \{0, 1\} \quad \forall h \in \{0, \dots, |H|\} \quad (4.25)$$

$$\sum_{h \in H} h \cdot \check{\alpha}_{th} = \sum_{h \in H} \check{x}_{th} \quad \forall t \in T \quad (4.26)$$

$$\sum_{h \in \{0, \dots, H\}} \check{\alpha}_{th} = 1 \quad \forall t \in T \quad (4.27)$$

$$\check{\alpha}_{th} \in \{0, 1\} \quad \forall t \in T, h \in \{0, \dots, |H|\} \quad (4.28)$$

Finalmente, em  $\mathcal{P}'_{pd}$  a computação da porção  $\check{d}_{pd}$  do custo reduzido, referente às variáveis duais, deve incorporar as equações 4.18 e 4.21. Desse modo, definiremos  $\check{d}_{pd}$  para  $\mathcal{P}'_{pd}$ , agora considerando as variáveis duais  $\iota$  e  $\vartheta$ , relacionadas às restrições 4.18 e 4.21, respectivamente:

$$\begin{aligned}
 \check{d}_{pd} = & \mu_{pd} + \sum_{t \in T} \sum_{h \in H} \nu_{tdh} \cdot \check{x}_{th} + \sum_{t \in T} \sum_{h \in H} \pi_{pt} \cdot \check{x}_{th} + \\
 & \sum_{t \in T} \sum_{h \in \tilde{G}_{pd}} \check{y}_{th} \cdot \check{k}_{pt} + \sum_{h \in H} \check{a}_h \cdot \check{l}_{pdh} + \sum_{t \in T} \sum_{h \in H} \check{a}_{th} \cdot \check{\vartheta}_{ptdh}
 \end{aligned} \tag{4.29}$$

Um dos cortes possíveis válidos para a formulação  $\mathcal{F}'_2$  são cortes 0-1 *knapsack cover* [9, 42, 77]. Considere as restrições implícitas existentes em  $\mathcal{F}'_2$ :

$$\sum_{h \in H} \sum_{d \in D} h \cdot \check{a}_{pdh} = \sum_{t \in T} \tilde{r}_{pt} \quad \forall p \in P \tag{4.30}$$

$$\sum_{h \in H} \sum_{d \in D} h \cdot \check{a}_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \tag{4.31}$$

Como exemplo de um corte de *cover* levando em conta as restrições 4.30 considere um professor  $p$  e um conjunto de pares ordenados  $\check{C}$  com elementos  $(d_1, h_1), \dots, (d_{\check{c}}, h_{\check{c}}), d_i \in D \forall i \in \{1, \dots, \check{c}\}, h_i \in H \forall i \in \{1, \dots, \check{c}\}$  tal que:

$$\sum_{i \in \{1, \dots, \check{c}\}} h_i \cdot \check{a}_{pd_i h_i} > \sum_{t \in T} \tilde{r}_{pt}$$

Desse modo, a seguinte desigualdade, denominada desigualdade *cover* é válida para  $\mathcal{F}'_2$ :

$$\sum_{i \in \{1, \dots, \check{c}\}} \check{a}_{pd_i h_i} \leq \check{c} - 1 \tag{4.32}$$

Apesar da possibilidade de utilização de uma implementação específica para a separação dos cortes de *cover*, como a descrita em [79], foi utilizada a abordagem mais genérica dos planos de cortes por enumeração, os *Fenchel Cuts*, descritos por Boyd et al. em [12, 13], que requerem a solução de um problema linear por iteração. Tal opção se mostrou bastante eficiente, novamente devido ao fato do número de períodos  $|H|$  ser bastante limitado. Nesse caso, dois problemas de separação foram considerados: no primeiro, foram considerados os pontos extremos do poliedro definido pelas equações 4.17, 4.30 e 4.19, já no segundo foram considerados os pontos extremos do poliedro definido pelas equações. 4.20, 4.31 e 4.22. Uma vantagem dessa abordagem é o fato de que o uso desses planos de corte permite que outros cortes, além dos cortes de *cover*, sejam inseridos automaticamente, caso a solução fracionária<sup>2</sup> gerada esteja fora da envoltória convexa dos pontos extremos enumerados.

O procedimento de geração de cortes acima apresentado apenas considera inclusão de variáveis ativas (que aparecem na solução fracionária com valores positivos) nos cortes gerados, visto que o problema de separação trata apenas da produção de um corte com violação máxima na solução corrente. De modo a produzir desigualdades mais fortes, a realização de um *lifting* pode ser

<sup>2</sup>Solução fracionária considerando as variáveis  $\check{a}$

considerada. Uma maneira bastante simples de obtenção de tais desigualdades é a inclusão de coeficientes nas variáveis inativas na construção do problema de separação. Tais coeficientes devem ser suficientemente pequenos para não atrapalharem a seleção do corte mais violado.

### 4.2.3 O Algoritmo de Geração de Colunas e Cortes

O algoritmo de Geração de Colunas e Cortes para o PPTC (Algoritmo 5), aqui denominado GCCPPTC, resolve na otimalidade a relaxação linear da formulação  $\mathcal{F}'_2$  estendida com a adição de cortes, sendo útil para a determinação de limites inferiores, bem como para a busca da solução inteira ótima, dentro do contexto dos métodos de *branch-cut-and-price*. O algoritmo realiza, a cada iteração (linhas 4-30), a adição de colunas ou cortes. A adição de cortes é realizada quando não existem mais colunas com custo reduzido negativo fora do programa linear  $\mathcal{PL}$ , indicando que a solução ótima para  $\mathcal{PL}$  foi atingida. A inserção de novos cortes requer a resolução de novos problemas de *pricing* e assim o processo continua até a iteração em que nem novos cortes nem novas colunas são inseridas.

Em se tratando da geração de cortes, como mencionado na seção anterior, são resolvidos problemas de separação relacionados ao fechamento da carga horária semanal de professores e ao fechamento da carga horária semanal de encontros entre professores e turmas. Denota-se por  $\mathcal{S}_i$  o problema de separação de cortes considerando soluções fracionárias para as variáveis  $\check{a}_{pdh} \forall p \in P, d \in D, h \in \{0, \dots, h\}$  e por  $\mathcal{S}_{pt}$  o problema de separação de cortes considerando as variáveis fracionárias  $\check{a}_{ptdh} \forall p \in P, t \in T, d \in D, h \in \{0, \dots, h\}$ .

Uma característica importante dos cortes adicionados em  $\mathcal{F}'_2$  é que eles não alteram a estrutura do problema de *pricing*. Essa é uma característica fundamental para a construção dos métodos de *Branch, Cut and Price* robustos (RBCP), como apresentado em [58]. Primeiramente, deve-se observar que o problema de geração de cortes pode eventualmente ser resolvido de modo heurístico sem prejudicar a determinação de um limite dual válido. Esse não é o caso da resolução do problema de *pricing*: a parada prematura da adição de colunas antes da iteração final (no caso, quando se obtém que  $\bar{c}_{pd} \geq 0 \forall p \in P, d \in D$  na resolução de  $\mathcal{P}_{pd}$ ) prejudica a determinação direta de um limite dual válido. Tem-se então uma relação assimétrica na qual a adição de colunas pode dificultar a geração de cortes, mas a adição de cortes não deveria dificultar a geração de colunas. Desse modo, considerando a incorporação de GCCPPTC em um método de *branch-cut-and-price*, é possível o desenvolvimento de um **branch-cut-and-price robusto**, como discutido em [58], o que exigiria apenas a inclusão das variáveis originais em  $\mathcal{F}'_2$  e sua consideração em  $\mathcal{P}_{pd}$ , para a realização do *branch* em variáveis  $x$ .

```

Entrada: Instância do PPTC
Saída: Vetor de solução  $\bar{\lambda}$  e limite dual  $f(\bar{\lambda})$ 
1 Inicialize o programa linear  $\mathcal{PL}$  com os dados do problema, utilizando a
  formulação  $\mathcal{F}'_2$ , considerando apenas variáveis artificiais e colunas  $\lambda$ 
  correspondentes a dias de folga;
2 Adicione em  $\mathcal{PL}$  o corte relativo ao número mínimo de dias de trabalho;
3  $novasColunas \leftarrow 1$ ;
4  $novosCortes \leftarrow 0$ ;
5 enquanto ( $novosCortes + novasColunas > 0$ ) faça
6   Resolva  $\mathcal{PL}$ , atualize a solução (variáveis  $\bar{\lambda}$  e  $\check{a}$ ) e a informação dual;
7    $novasColunas \leftarrow 0$ ;
8   para cada  $p \in P$  faça
9     para cada  $d \in D \mid \sum_{h \in H} \tilde{p}_{pdh} > 0$  faça
10      Resolva  $\mathcal{P}'_{pd}$ ;
11      se ( $\check{c}_{pd} < 0$ ) então
12        Adicione nova coluna em  $\mathcal{PL}$ ;
13         $novasColunas \leftarrow novasColunas + 1$ ;
14      fim
15    fim
16  fim
17   $novosCortes \leftarrow 0$ ;
18  se ( $novasColunas = 0$ ) e (soluçãoFracionária( $\bar{\lambda}$ )) então
19    para cada  $p \in P$  faça
20      Resolva  $\mathcal{S}_p$ ;
21      se Corte violado encontrado então
22        Adicione corte em  $\mathcal{PL}$ ;
23         $novosCortes \leftarrow novosCortes + 1$ ;
24      fim
25    fim
26  fim
27 fim

```

**Algoritmo 5:** Algoritmo de Geração de Colunas e Cortes - GCCPPTC

# Referências Bibliográficas

- [1] Second international timetabling competition. [Disponível on-line em: <http://www.cs.qub.ac.uk/itc2007/>; acessado 16-Agosto-2007].
- [2] Metaheuristics network, 2007. [Disponível on-line em: <http://www.metaheuristics.net/>; acessado 16-Agosto-2007].
- [3] S. Abdullah, S. Ahmadi, E.K. Burke, and M. Dror. Investigating ahujarlin's large neighbourhood search approach for examination timetabling. *OR Spectrum*, 29:351–372, 2007.
- [4] D. Abramson. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, 37:98–113, 1991.
- [5] D. Abramson and J. Abela. A parallel genetic algorithm for solving the school timetabling problem. In *Proceedings of the 15th Australian Computer Science Conference*, 1992.
- [6] R. Alvarez-Valdés, G. Martin, and M. Tamarit. Constructing good solutions for the spanish school timetabling problem. *Journal of the Operational Research Society*, 47:1203–1215, 1996.
- [7] J.E.C. Arroyo. *Heurísticas e Metaheurísticas para Otimização Combinatória Multiobjetivo*. PhD thesis, Universidade Estadual de Campinas, 2002.
- [8] P. Avella and I. Vasil'ev. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8:497–514, 2005.
- [9] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8(1):146–164, 1975.
- [10] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.
- [11] T. Birbas, S. Daskalaki, and E. Housos. Timetabling for greek high schools. *Journal of the Operational Research Society*, 48:1191–1200, 1997.
- [12] E.A. Boyd. Fenchel cutting planes for integer programming. *Operations Research*, 42:53–64, 1992.
- [13] E.A. Boyd. Solving 0/1 integer programs with enumeration cutting planes. *Annals of Operations Research*, 50:61–72, 1994.

- [14] D. Brelaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, April 1979.
- [15] E. Burke and M. Carter, editors. *Practice and Theory of Automated Timetabling II*, Lecture Notes in Computer Science. Springer, 1997.
- [16] E. Burke and W. Erben, editors. *Practice and Theory of Automated Timetabling III*, Lecture Notes in Computer Science. Springer, 2000.
- [17] E. Burke and P. Ross, editors. *Practice and Theory of Automated Timetabling*, Lecture Notes in Computer Science. Springer, 1996.
- [18] E.K. Burke, J. H. Kingston, and R. Weare. Automated university timetabling: The state of the art. *The Computer Journal*, 40:565–571, 1997.
- [19] J. Caldeira and C. Agostinho. School timetabling using genetic search. In *Practice and Theory of Automated Timetabling (1997)*, pages 115–122, Toronto, 1997.
- [20] M. P. Carrasco and M.V. Pato. A multiobjective genetic algorithm for the class/teacher timetabling problem. In *Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes In Computer Science*, pages 3–17. Springer-Verlag, 2000.
- [21] M.W. Carter. Examination scheduling: Documentation. Technical report, Data Processing Department, University of Waterloo, 1978.
- [22] M.W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2):193–202, 1986.
- [23] M.W. Carter and G. Laporte. Recent developments in practical examination timetabling. In E. Burke and P. Ross, editors, *Practice and Theory of Automated Timetabling: First International Conference*, volume 1153 of *Lecture Notes in Computer Science*, pages 3–21, 1996.
- [24] M.W. Carter and C.A. Tovey. When is the classroom assignment problem hard? *Operations Research Supplement 1*, 40:28–39, 1992.
- [25] A. Colomi, M. Dorigo, and V. Maniezzo. Metaheuristics for high-school timetabling. *Computational Optimization and Applications*, 9(3):277–298, 1998.
- [26] T. Cooper and J. H. Kingston. *The complexity of timetable construction problems*, volume 1153 of *Springer Lecture Notes in Computer Science*, pages 283–295. Springer-Verlag, London, UK, 1996.
- [27] D. Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76:98–110, 1994.
- [28] J. Csima and C.C. Gotlieb. Tests on a computer method for constructing school timetables. *Communications of the ACM*, 7(3):160–163, 1964.
- [29] V.D. Cung, S.L. Martins, C.C. Ribeiro, and C. Roucairol. *Strategies for the parallel implementation of metaheuristics*, pages 263–308. Kluwer Academic Publishers, 2002.

- [30] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.
- [31] S. Daskalaki. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153:117–135, 2004.
- [32] C. Di Stefano and A. Tettamanzi. An evolutionary algorithm for solving the school time-tabling problem. In E. Boers et al., editor, *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*, volume 2037, pages 452–462, London, UK, 2001. Springer-Verlag.
- [33] A. Drexler and F. Salewski. Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, 102(1):193–214, October 1997.
- [34] F.D.J Dunstan. Sequential coloring of graphs. In *Proceedings of the 5th British Combined Conference*, Utilitas Math., pages 151–158, 1975.
- [35] H. A. Eiselt and G. Laporte. Combinatorial optimization problems with soft and hard requirements. *Journal of Operations Research*, 38(9):785–795, 1987.
- [36] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal of Computing*, 5(4):691–703, December 1976.
- [37] E. Foxley and K. Lockyer. The construction of examination timetables by computer. *Computer Journal*, 11:264–268, 1968.
- [38] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, USA, 1979.
- [39] F. Glover. Future paths for integer programming and artificial intelligence. *Computers & Operations Research*, 13:533–549, 1986.
- [40] F. Glover and M. Laguna. *Tabu Search*. Kluwer, Boston, 1997.
- [41] C.C. Gotlieb. The construction of class-teacher timetables. In C. M. Poplewell, editor, *Proceeding IFIP congress 1962*, pages 73–77, Amsterdam, 1963.
- [42] P.L. Hammer, E.L. Johnson, and U.N. Peled. Facets of regular 0-1 polytopes. *Mathematical Programming*, 8(1):179–206, 1975.
- [43] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy, 1986.
- [44] A. Hertz. Tabu search for large scale timetabling problems. *European Journal of Operational Research*, 54:39–47, 1991.
- [45] A. Hertz and D. Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, December 1987.

- [46] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [47] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley-Interscience, 1994.
- [48] E.L. Johnson, G.L. Nemhauser, and M.W.P. Savelsbergh. Progress in linear programming-based algorithms for integer programming: An exposition. *INFORMS Journal on Computing*, 12(1), 2000.
- [49] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [50] G. Laporte and Desroches S. Examination timetabling by computer. *Computers and Operations Research*, 11(4):351–360, August 1984.
- [51] N.L. Lawrie. An integer linear programming model of a school timetabling problem. *Computer Journal*, 12:307–316, 1969.
- [52] R. Lewis, B. Paechter, and B. McCollum. Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition, 2007. [Disponível on-line em: <http://www.cs.qub.ac.uk/itc2007/postenrolcourse/report/Postsed>
- [53] D.W. Matula, G. Marble, and I.D. Isaacson. Graph coloring algorithms. *Graph Theory and Computing*, 1972.
- [54] S.A. MirHassani. A computational approach to enhancing course timetabling with integer programming. *Applied Mathematics and Computation*, 175:814–822, 2006.
- [55] Metaheuristics Network. International timetabling competition, 2002. [Disponível on-line em: <http://www.idsia.ch/Files/ttcomp2002/>; acessado 16-Agosto-2007].
- [56] K. Papoutsis, C. Valouxis, and E. Housos. A column generation approach for the timetabling problem of greek high schools. *Journal of the Operational Research Society*, 54:230–238, 2003.
- [57] J.E.L. Peck and M.R. Williams. Algorithm 286: Examination scheduling. *Communications of the ACM*, 9(6):433–434, June 1966.
- [58] M.V. Poggi de Aragão and E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price. In *Annals of Mathematical Programming in Rio*, pages 56–61, Buzios, Brazil, 2003.
- [59] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In *Handbook of Metaheuristics*, pages 219–249. Kluwer, 2003.
- [60] H. G. Santos, L.S. Ochi, and M.J.F. Souza. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *Journal of Experimental Algorithmics (JEA)*, 10, 2005.
- [61] A. Schaerf. Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics Part A:systems and Humans*, 29(4):368–377, July 1999.

- [62] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.
- [63] M. J. F. Souza, A. X. Martins, and C. R. Araújo. Experiências com a utilização de simulated annealing e busca tabu na resolução do problema de alocação de salas. In *Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional*, pages 1100–1110, 2002.
- [64] M. J. F. Souza, L. S. Ochi, and N. Maculan. A GRASP-Tabu Search Algorithm for solving School Timetabling Problems. In M. G. C. Resende and J. P. Souza, editors, *Metaheuristics: Computer Decision-Making*, volume 1153, pages 659–672. Kluwer Academic Publishers, 2003.
- [65] M.J.F. Souza. *Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas*. PhD thesis, Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [66] M.J.F. Souza, N. Maculan, and L.S. Ochi. Melhorando quadros de horários de escolas através de caminhos mínimos. *Tendências em Matemática Aplicada e Computacional*, 1(1):515–524, 2000.
- [67] M.J.F. Souza, L.S. Ochi, and N. Maculan. *A GRASP-Tabu Search Algorithm for solving School Timetabling Problems*, pages 659–672. Kluwer Academic Publishers, 2003.
- [68] A. Subramanian, J. M. F. Medeiros, L. A. F. Cabral, and M. J. F. Souza. Aplicação da metaheurística busca tabu na resolução do problema de alocação de salas do centro de tecnologia da UFPB. In *Anais do XXVI Encontro Nacional de Engenharia de Produção*, pages 1–9, 2006.
- [69] Linderoth J. T. and T. K. Ralphs. Noncommercial software for mixed-integer linear programming. In J. Karlof, editor, *Integer Programming: Theory and Practice*, volume 3 of *Operations Research Series*. 2005.
- [70] H.M.M. ten Eikelder and R.J. Willemsen. Some complexity aspects of secondary school timetabling problems. In E.K. Burke and W. Erben, editors, *Practice and Theory of Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 3–17. Springer-Verlag, 2001.
- [71] J. R. A. Torreão and E. Roe. Microcanonical optimization applied to visual processing. *Physics Letters*, 205:377–382, February 1995.
- [72] S. Voss, S. Martello, I.H. Osman, and C. Roucairo. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Boston, 1999.
- [73] D.J.A. Welsh and M.B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *Computer Journal*, 10:85–86, 1967.
- [74] D. Werra. An introduction to timetabling. *European Journal of Operational Research*, 19:151–162, 1985.

- [75] D. Werra. The combinatorics of timetabling. *European Journal of Operational Research*, 96(3):504–513, 1997.
- [76] M.R. Williams. Heuristic procedures (if they work - leave them alone). *Software: Practice and Experience*, 4(3):237–240, 1974.
- [77] L.A. Wolsey. Faces for a linear inequality in 0-1 variables. *Mathematical Programming*, 8:165–178, 1975.
- [78] A. Wren. Scheduling, timetabling and rostering - a special relationship? In E.K. Burke and M.W. Carter, editors, *The Practice and Theory of Automated Timetabling I*, volume 1153 of *Lecture Notes in Computer Science*, pages 46–75. Springer-Verlag, 1996.
- [79] G. Zonghao, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437, 1998.