

UNIVERSIDADE FEDERAL FLUMINENSE

Haroldo Gambini Santos

Formulações e Algoritmos para o Problema de
Programação de Horários em Escolas

NITERÓI

2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL FLUMINENSE

Haroldo Gambini Santos

Formulações e Algoritmos para o Problema de Programação de Horários em Escolas

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor. Área de concentração: Otimização Combinatória e Inteligência Artificial.

Orientador:
Luiz Satoru Ochi

Co-orientador:
Eduardo Uchoa

NITERÓI

2007

Formulações e Algoritmos para o Problema de Programação de Horários em Escolas

Haroldo Gambini Santos

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor.

Aprovada por:

Prof. D.Sc. Luiz Satoru Ochi / UFF (Presidente)

Prof. Ph.D. Abilio Pereira de Lucena Filho / UFRJ

Prof. D. Hab. Celso da Cruz Carneiro Ribeiro / UFF

Prof. D.Sc. Eduardo Uchoa / UFF

Prof. D.Sc. Marcone Jamilson Freitas Souza / UFOP

Prof. D. Hab. Nelson Maculan Filho / UFRJ

Prof. D.Sc. Simone de Lima Martins / UFF

Niterói, 30 de Março de 2007.

Para Hique, responsável pelos “ótimos globais” de felicidade da minha vida.

Haroldo Gambini Santos

Agradecimentos

Agradeço à minha família: meus pais Neusa e Flávio e meu irmão Henrique, fontes de carinho e motivação nessa jornada. Ao Henrique, por ter sido o melhor irmão que se pode ter durante a abreviada e intensa passagem dele por esse mundo.

Ao professor Satoru, pela acolhida na UFF e pela orientação desde o início desse trabalho, obrigado por acreditar em mim. Ao professor Uchoa, por aceitar a orientação desse trabalho e por participar ativamente desta pesquisa. A ambos, mais que orientadores, amigos, o meu eterno agradecimento por me ensinarem a trabalhar com gosto e alegria. Meu agradecimento também ao professor Marcone, essa pessoa fantástica e pesquisador exemplar que colaborou desde o início desta pesquisa.

Obrigado ao membros da banca examinadora desta tese, em especial aos professores Maculan e Celso, que contribuíram para este trabalho desde a defesa de proposta de tese.

Aos amigos, em especial aos que acompanharam de perto as dificuldades e realizações nesse tempo: Luiz Henrique, Euler e Olinto. Muito obrigado por existirem. Obrigado também à Tatiana.

Finalmente, meu sincero agradecimento aos professores e colegas da Universidade Federal Fluminense, componentes mais importantes desse agradável e produtivo ambiente à beira da Baía da Guanabara. Obrigado às professoras Lúcia, Simone e Helena pelas disciplinas ótimamente ministradas.

Resumo

Esta tese trata de uma variante de um clássico problema combinatório NP-Completo: o Problema de Programação de Horários Professor-Turma. Nessa variante, importantes considerações práticas são incorporadas, especificamente tratando de preferências de professores e distribuição das aulas. As propostas apresentadas nesta tese dividem-se em 3 partes. Na primeira parte é apresentada uma nova heurística híbrida, baseada em Busca Tabu. Experimentos computacionais demonstraram que a heurística proposta melhora os resultados da literatura, apresentando um desempenho consistentemente superior em todos os problemas teste. Na segunda parte propostas foram apresentadas para a obtenção de quadros de horários provadamente ótimos, considerando técnicas de Programação Linear Inteira Mista. Nesse sentido é desenvolvida uma formulação com um número exponencial de linhas e colunas, bem como um algoritmo que utiliza as técnicas de geração de colunas e cortes para o tratamento dessa formulação. Experimentos computacionais demonstraram que a formulação apresentada permite a obtenção de limites inferiores bastante fortes, os quais permitiram a prova da otimalidade para 3 instâncias em aberto da literatura. Finalmente, na terceira parte, é explorada a sinergia entre heurísticas e métodos exatos, através da resolução ótima de sub-problemas em quadros de horários construídos heurísticamente, esses algoritmos híbridos ofereceram os melhores limites superiores disponíveis.

Palavras-chave: Programação de Horários em Escolas; Programação Linear Inteira; Metaheurísticas.

Abstract

This thesis considers an variation of a classical combinatorial NP-Complete problem: The Class-Teacher Timetabling Problem. In this variant, important practical considerations are incorporated, specifically, teachers' preferences and distribution of lessons. The proposals of this work are divided in three parts. In the first part a new hybrid heuristic is presented, based on Tabu Search. Computational experiments demonstrated that the proposed heuristic improves upon existing methods presenting a consistently better performance in all test problems. In the second part, proposals for producing optimal timetables are presented, considering Mixed Integer Linear Programming Techniques. In this sense, a formulation with an exponential number of rows and columns is developed, as well an algorithm to manage this formulation by cut and column generation. Computational experiments demonstrated that the proposed formulation provides stronger bounds, which allowed the proof of optimality for 3 open instances from literature. Finally, in the third part, the synergy of heuristics and exact methods is explored, by the optimal resolution of sub-problems from timetables heuristically generated. These hybrid algorithms offered the best upper bounds available.

Keywords: School Timetabling; Mixed Integer Programming; Metaheuristics.

Palavras-chave

1. Otimização Combinatória
2. Programação de Horários
3. Metaheurísticas
3. Programação Linear Inteira Mista

Lista de Siglas e Acrônimos

BT-PPTC	: Algoritmo de Busca Tabu para o Problema de Programação de Horários com Restrições de Compacidade;
BT-PPTC-T	: Algoritmo de Busca Tabu com diversificação baseada e memória de Transição para o Problema de Programação de Horários com Restrições de Compacidade ;
BT-PPTC-R	: Algoritmo de Busca Tabu com diversificação baseada e memória de Residência para o Problema de Programação de Horários com Restrições de Compacidade ;
BT-PPTC-TR	: Algoritmo de Busca Tabu com diversificação baseada e memória de Transição e Residência para o Problema de Programação de Horários com Restrições de Compacidade;
CGA-PPTC	: Algoritmo Construtivo, Guloso e Aleatorizado para o Problema de Programação de Horários com Restrições de Compacidade;
IC	: Instituto de Computação;
LB	: <i>Local Branching</i> ;
LRC	: Lista Restrita de Candidatos;
PPT	: Problema de Programação de Horários Professor×Turma
PPTC	: Problema de Programação de Horários Professor×Turma com Restrições de Compacidade;
PLIM	: Programação Linear Inteira Mista;
PPLIM	: Problema de Programação Linear Inteira Mista;
RC	: Reconexão de Caminhos;
RBCP	: <i>Robust Branch, Cut and Price</i> ;
UFF	: Universidade Federal Fluminense;
VNS	: Busca em Vizinhança Variável.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Algoritmos	12
1 Introdução	1
2 O Problema de Programação de Horários Professor \times Turma com Compacidade - PPTC	3
2.1 Tratando problemas reais de programação de horários em escolas	5
2.2 O Problema de Programação de Horários Professor \times Turma com Compacidade	7
2.3 O conjunto de instâncias-teste	10
3 Estado da Arte	12
3.1 Busca Local	12
3.2 Busca Tabu	13
3.3 <i>Simulated Annealing</i>	14
3.4 Algoritmos Evolutivos	15
3.5 Métodos Baseados em Programação Linear	15
3.6 Comparações entre os métodos	16
4 Heurísticas para o PPTC	18
4.1 Representação e avaliação de soluções:	18

4.2	Algoritmo construtivo - CGA-PPTC	19
4.3	A heurística de melhoramento BT-PPTC	20
4.4	Experimentos computacionais	25
5	Programação Linear Inteira Mista para o PPTC	35
5.1	Cortes	36
5.2	Formulação Estendida com Geração de Colunas e Cortes	39
5.2.1	Geração de Colunas	41
5.2.2	Geração de Cortes	42
5.2.3	O Algoritmo de Geração de Colunas e Cortes	46
5.2.4	Experimentos Computacionais	48
6	Métodos Híbridos para o PPTC	52
6.1	Experimentos computacionais	56
7	Conclusões e Trabalhos Futuros	60
	Apêndice A - Publicações	62
	Apêndice B - Quadros de Horários Ótimos e Sub-ótimos	65
	Referências	70

Lista de Figuras

4.1	Fragmento de um quadro de horários gerado	19
4.2	Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 1 e 2	31
4.3	Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 3 e 4	32
4.4	Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 5 e 6	33
4.5	Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para a instância 7	34
6.1	Exemplo de execução do método de reconexão de caminhos, um caminho entre uma solução base e uma solução guia é explorado.	54
6.2	Exemplo de execução da busca na vizinhança elipsoidal com $k = 0$, todas as soluções com qualquer combinação de atributos das soluções base e guia são consideradas.	55
6.3	Número de melhoras obtidas com a busca na vizinhança <i>Local Branching</i> , formulações fraca e forte, execuções com limite de tempo.	58
B.1	Solução ótima - instância 1 - avaliação: 202	66
B.2	Solução ótima - instância 2 - avaliação: 303	67
B.3	Solução ótima - instância 3 - avaliação: 423	68
B.4	Solução sub-ótima - instância 4 - avaliação: 653 - limite inferior: 652	69

Lista de Tabelas

2.1	Características das instâncias	11
4.1	limite de tempo de cada instância	25
4.2	Distância média das melhores soluções conhecidas - BT-PPTC-S	26
4.3	Distância média das melhores soluções conhecidas - BT-PPTC-T	27
4.4	Distância média das melhores soluções conhecidas - BT-PPTC-R	27
4.5	Distância média das melhores soluções conhecidas - BT-PPTC-TR	27
4.6	Custos médios produzidos - execuções com limites de tempo	28
4.7	Valores médios dos componentes da função de custo em soluções geradas pelo algoritmo construtivo e em soluções do alg. BT-PPTC-TR	29
4.8	Tempo (em segundos) para que 25%, 50% e 75% das execuções atinjam o valor alvo	30
5.1	Diferentes variações da formulação \mathcal{F}_1 avaliadas.	48
5.2	Resultados da formulação \mathcal{F}_1	49
5.3	Resultados da formulação estendida com geração de colunas e cortes	51
6.1	Custos das soluções de partida para cada instância	57
6.2	Comparação da otimização na vizinhança definida pelos cortes de <i>local branching</i> e pelo corte elipsoidal	57

Lista de Algoritmos

1	Construtivo guloso aleatorizado	20
2	Algoritmo BT-PPTC	24
3	Algoritmo de Geração de Colunas e Cortes GCCPPTC	47

Capítulo 1

Introdução

Este trabalho propõe métodos computacionais de solução para uma importante variante de um problema clássico de otimização combinatória, o Problema de Programação de Horários Professor \times Turma (PPT) [1, 2]. O problema abordado considera requisitos necessários para o tratamento de muitas instâncias reais do PPT: a compacidade de horários e a duração das aulas. Problemas de Programação de Horários, há várias décadas, têm recebido grande interesse por parte dos pesquisadores em computação. Atualmente, conferências regulares discutem o tema no meio científico [3, 4, 5]. A motivação ocorre devido à dificuldade de obtenção de soluções ótimas em tempos computacionais aceitáveis e também devido à importância prática do problema, visto que instituições educacionais de todo o mundo, rotineiramente, enfrentam a dificuldade de obtenção de bons quadros de horários considerando os inúmeros requisitos pedagógicos, pessoais e institucionais envolvidos.

O presente trabalho considera métodos de produção automatizada de horários com dois propósitos, primeiramente, buscam-se métodos heurísticos que em tempos computacionais aceitáveis produzam bons quadros de horários (Capítulo 4). Em um segundo momento, pesquisa-se a produção de horários provavelmente ótimos, utilizando métodos baseados em Programação Linear Inteira Mista (PLIM) (Capítulo 5).

Apesar do número considerável de heurísticas existentes na literatura (Capítulo 3), a análise comparativa destas é muitas vezes prejudicada pela metodologia empregada nos trabalhos em questão. Comparações entre métodos são raras, sendo que a avaliação das soluções obtidas se dá na maioria dos casos por meios subjetivos, através da descrição do sucesso obtido pelo ponto de vista de usuários das aplicações. Neste trabalho, uma nova heurística baseada em Busca Tabu é proposta e avaliada computacionalmente (Capítulo 4). O método incorpora mecanismos de diversificação baseados em memória de longo

prazo. Os resultados que serão posteriormente apresentados neste trabalho permitem a avaliação da qualidade das soluções obtidas com relação a ótimos globais, em alguns casos, bem como com relação a limites inferiores. O desempenho heurístico também é comparado com outros trabalhos da literatura [6, 7, 8].

Métodos baseados em PLIM também foram analisados, sendo que diferentes formulações foram propostas e avaliadas utilizando a biblioteca de funções do programa resolvidor CPLEX 10, que representa o estado da arte em programas desse tipo. A obtenção de formulações fortes também é considerada através do estudo de cortes para as formulações apresentadas, bem como através da técnica de geração de colunas. Este estudo permitiu, pela primeira vez, a determinação da solução ótima para três instâncias disponíveis na literatura, utilizadas em [6, 7, 8, 9], como será visto no Capítulo 5.

A eficiência dos métodos de PLIM para a resolução ótima de problemas com dimensões limitadas motivou a proposta de algoritmos que utilizam PLIM para a pesquisa em vizinhanças de soluções obtidas através das heurísticas propostas, como métodos de melhoramento, no espírito do método de *Local Branching* recentemente proposto [10]. Vizinhanças alternativas, que utilizam os denominados cortes elipsoidais [11], também são consideradas para programação de horários. Estes métodos, aqui denominados Métodos Híbridos, são discutidos no Capítulo 6.

Capítulo 2

O Problema de Programação de Horários Professor \times Turma com Compacidade - PPTC

Neste capítulo será apresentado o problema considerado nesse trabalho. Inicialmente será descrito um problema clássico de programação de horários: o Problema de Programação de Horários Professor \times Turma. 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 [1, 2], tem sido bastante estudado nas últimas décadas. Revisões de literatura podem ser encontradas em [12, 13].

Para facilitar a legibilidade das seções posteriores, os períodos do PPT aqui apresentado serão considerados distribuídos igualmente por um conjunto D de dias da semana. Assim, os dados que definem um PPT são:

- 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

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

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

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

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

Onde 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 [14], 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 [15]. Em [16] 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.

2.1 Tratando problemas reais de programação de horários em escolas

O problema apresentado na seção anterior, apesar da importância teórica, não considera muitos requisitos importantes na construção de quadros de horários em instituições educacionais. Em geral, além de se procurar um quadro viável com relação a conflitos em horários, vários requisitos adicionais são considerados. Esses requisitos podem ser basicamente de três tipos:

- **organizacionais:** relativos à instituição de ensino;
- **pedagógicos:** pedidos importantes para o bom aproveitamento das aulas;
- **peçoais:** requisitados de acordo com as preferências pessoais dos membros do corpo docente.

A importância relativa dos pedidos de cada tipo é bastante dependente do problema considerado, como indicado em [12]. No entanto, algumas tendências gerais podem ser observadas, como será apresentado nos parágrafos seguintes.

Considerando-se pedidos 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 blocos contíguos, em poucos dias da semana. Essas questões foram consideradas em [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. 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 compatidade 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 [27, 19], 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 [17], de um modo mais específico, determina-se quais aulas devem durar um ou dois períodos (aulas geminadas).

Em [18, 8, 7, 6], 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 [20], seja através da remoção de horários com essas características do espaço de busca, como em [28]. 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 [17], como matemática e física. De modo semelhante, [27] divide as disciplinas em “principais”, no caso o 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 [27]. 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 ineficazes, dada a impossibilidade de deslocamento do professor entre as salas de aula em um tempo viável.

A satisfação de todos os requisitos desejáveis na construção de um quadro de horários pode ser, e geralmente é, impossível de se realizar na prática, inclusive porque os pedidos podem ser conflitantes: o atendimento de um determinado padrão de distribuição de aulas, por exemplo, pode resultar em um quadro de horários não tão compacto para um determinado professor. Dessa forma, as diferentes requisições devem ser organizadas em diferentes níveis de importância [29]. A primeira divisão trata da especificação de quais pedidos, se não atendidos, definem um quadro de horários como ineficaz. Essas restrições, chamadas de **restrições fortes**, definem o espaço de busca do problema de programação de horários. Como restrições fortes típicas tem-se a não ocorrência de conflitos, pré-alocações, visto que algumas aulas podem já estar com o horário fixado e o respeito a disponibilidade de recursos humanos ou físicos, como salas de aulas.

Um outro conjunto de restrições, chamadas **restrições fracas**, serve para a avaliação da qualidade de um quadro de horários. Nesse caso procura-se penalizar na função objetivo o não atendimento dessas restrições. Típicas restrições fracas são a compactidade de horários e a distribuição das aulas.

Dependendo da instituição envolvida, ou algumas vezes de modo mais específico, do professor considerado, o atendimento de algumas restrições fracas é mais importante do que outras. Desse modo, o tratamento dos problemas de programação de horários como

problemas de otimização geralmente inclui a definição de uma função mono-objetivo composta pela combinação linear com penalidades associadas ao não atendimento das diferentes restrições [18, 17, 20].

2.2 O Problema de Programação de Horários Professor \times Turma com Compacidade

Neste trabalho será abordado um problema com considerações adicionais ao PPT, que trata importantes aspectos práticos mencionados na seção anterior. Esse problema, apresentado inicialmente em [6] e posteriormente considerado em [8, 30], 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:

- (a) compacidade de horários: o quadro de horários de um professor deve se estender por um número mínimo possível de dias;
- (b) aulas geminadas: alguns professores requerem um número mínimo de aulas geminadas, em determinadas turmas, para a execução dos conteúdos;
- (c) “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 [6], 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 $\hat{G}_{pd} \subset H$ o subconjunto de períodos na agenda do professor p no dia d com a propriedade que $\tilde{p}_{ptdh} \cdot \tilde{p}_{ptdh+1} = 1 \forall h \in \hat{G}_{pd}$, 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 compacidade 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

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

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

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

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

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

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

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

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

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

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

$$g_{pt} \geq \tilde{g}_{pt} - \sum_{d \in D} \sum_{h \in \hat{G}_{pt}} y_{ptdh} \quad \forall p \in P, t \in T \quad (2.17)$$

$$x_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (2.18)$$

$$y_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in \hat{G}_{pd} \quad (2.19)$$

$$v_{pd} \in \{0, 1\} \quad \forall p \in P, d \in D \quad (2.20)$$

$$\bar{a}_{pd} \in \{0, \dots, |H|\} \quad \forall p \in P, d \in D \quad (2.21)$$

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

$$b_{pd} \in \{0, \dots, |H| - 2\} \quad \forall p \in P, d \in D \quad (2.23)$$

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 ¹. Finalmente, mede-se o não atendimento do número mínimo de aulas geminadas para um professor p e turma t em 2.17. As restrições restantes determinam o domínio das variáveis.

2.3 O conjunto de instâncias-teste

Nos experimentos computacionais realizados no decorrer desse trabalho foram utilizadas instâncias (Tabela 2.1) originárias dos trabalhos de Mata [9] e Souza [6]. As instâncias 1 e 3 são provenientes de [9], sendo que o primeiro problema foi criado artificialmente e problema 3 trata da programação de horários de uma escola de ensino médio de Brasília (DF), no ano de 1989. Os problemas 2, 4 e 6 referem-se ao planejamento das aulas dos turnos da manhã, tarde e noite, respectivamente, da escola Dom Silvério, de Mariana, Minas Gerais, no ano de 2000. O problema 5, dessa mesma escola, refere-se aos dados de 1998, no turno da manhã. O problema 7 refere-se a uma simulação da situação de que essa escola teria que oferecer aulas para mais turmas concentradas em um único turno. Em todos os problemas a dimensão da grade de horários é $|D| = |H| = 5$. Entre as características dos problemas na Tabela 2.1 está o número total de pedidos de aulas geminadas, a taxa de esparsidade o número de variáveis binárias e contínuas² produzidas com a formulação \mathcal{F}_1 para cada um dos problemas. A taxa de esparsidade é a indicação da liberdade de escolha de quais períodos do professor terão alguma alocação. Valores baixos indicam que a maioria dos períodos disponíveis será preenchido com aulas. Essa medida foi computada considerando-se o número total de aulas na instância e o número total de indisponibilidades, como segue:

$$e = \frac{|P| \times |D| \times |H| - (\sum_{p \in P} \sum_{t \in T} \tilde{r}_{pt} + \sum_{t \in T} \sum_{d \in D} \sum_{h \in H} (1 - \tilde{p}_{pdh}))}{|P| \times |D| \times |H|} \quad (2.24)$$

Na instância 4, por exemplo, o professor 13 (ver Anexo B.4), leciona 20 horas aula na semana e tem apenas 20 períodos de disponibilidade, o que contribuiu para a baixa

¹A amarração das variáveis y_{ptdh} com as variáveis x_{ptdh} utilizando menos restrições do que em [6] 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} .

²As únicas variáveis que necessitam ser declaradas explicitamente como inteiras são as variáveis x_{ptdh} , visto que a integralidade das demais decorre naturalmente da formulação. Apesar disso, em nossos experimentos, observamos que o resolvidor de PLIM testado apresenta uma melhor desempenho quando um número maior de variáveis inteiras é explicitamente declarado, visto que mais cortes são rapidamente gerados.

Inst.	P	T	Total de Aulas	Aulas Geminadas	Taxa de Esparsidade (e)	Variáveis	
						Binárias	Contínuas
1	8	3	75	21	0,43	420	383
2	14	6	150	29	0,50	1460	1543
3	16	8	200	4	0,30	1380	641
4	23	12	300	66	0,18	2471	1053
5	31	13	325	71	0,58	2975	3861
6	30	14	350	63	0,52	3455	3417
7	33	20	500	84	0,39	5125	3514

Tabela 2.1: Características das instâncias

esparsidade da instância 4 (Tabela 2.1).

Em todos os experimentos realizados nesta tese a importância atribuída aos diferentes componentes da função objetivo foi determinada do mesmo modo que em [8]: $w'_p = 3, w''_p = 9, w'''_p = 1, \forall p \in P$.

Capítulo 3

Estado da Arte

Neste capítulo será feita uma discussão sobre os trabalhos existentes na literatura de métodos computacionais para produção de quadros de horários, com ênfase em trabalhos que consideram problemas iguais ou relacionados ao PPTC.

Vários métodos têm sido utilizados para a produção automatizada de quadros de horários. Entre os métodos heurísticos, em geral, versões sofisticadas são necessárias: mesmo a geração de soluções factíveis para problemas reais de programação de horários é uma tarefa difícil e uma vez que se obtenha uma solução factível o método de melhora preferencialmente deve ser capaz de escapar dos ótimos locais, na maioria das vezes distantes do ótimo global para esse tipo de problema. Metaheurísticas baseadas em busca local como Busca Tabu e *Simulated Annealing* têm se mostrado técnicas genéricas eficientes para esse tipo de problema, assim como algoritmos evolutivos híbridos. Entre as técnicas exatas destacam-se métodos de Programação Linear Inteira Mista. A seguir serão apresentados alguns trabalhos que propuseram métodos de produção automatizada de horários para instituições educacionais.

3.1 Busca Local

O trabalho de Ferland e Lavoie [31], apresenta uma modelagem de programação inteira 0-1 para problemas de programação de horários, de modo que o tratamento de problemas como o PPT é possível através da utilização das chamadas *side constraints*. Essa modelagem relativamente genérica permite a modelagem de várias restrições fortes, mas não permite a modelagem de problemas com restrições fracas, como as existentes no PPTC. Um procedimento heurístico baseado em busca local é apresentado. Nesse procedimento, a busca local primeiramente é aplicada para a diminuição progressiva das inactibilida-

des. Logo que uma solução factível é obtida, inicia-se a busca por melhores soluções. Três métodos são apresentados: no primeiro utiliza-se trocas simples como busca local, o segundo procedimento realiza uma busca recursiva que apresenta similaridades em relação ao método de busca tabu e finalmente um procedimento de relaxação lagrangeana é apresentado. Nos experimentos computacionais realizados verificou-se que o procedimento recursivo teve mais sucesso em produzir soluções factíveis, enquanto a relaxação lagrangeana foi útil para a determinação de limites inferiores e também para a melhora da função objetivo.

3.2 Busca Tabu

O método de Busca Tabu, proposto independentemente por Glover [32] e Hansen [33] 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.

Hertz [23] propôs métodos de busca tabu para a geração de horários em universidades baseados em uma implementação prévia [34] 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 [17, 18, 20]. Nos três trabalhos, instâncias de problemas reais foram consideradas.

Na busca tabu de Costa [17] 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 [18] 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 [35].

O método híbrido baseado em Busca Tabu e GRASP [36] GBT-II, proposto por Souza [6, 8], destacou-se entre os algoritmos híbridos estudados em [6], onde são apresentadas e avaliadas computacionalmente várias propostas de heurísticas para o PPTC, incluindo *Simulated Annealing*, Busca Tabu e Otimização Microcanônica [37]. Os algoritmos propostos agregavam um método de melhoramento baseado em caminhos mínimos [7] 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.

3.3 *Simulated Annealing*

O método *Simulated Annealing* foi apresentado no contexto de otimização em [38]. 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. Abramson [39] realiza experimentos com versões sequenciais 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 [40], implementada em um sistema paralelo com memória compartilhada, apresentou uma aceleração considerável.

3.4 Algoritmos Evolutivos

Vários algoritmos genéticos foram propostos para a resolução de problemas de programação de horários. Abramson e Abela [41] propuseram um algoritmo genético paralelo para a resolução de problemas de programação de horários em 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 [20] 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 [22].

3.5 Métodos Baseados em Programação Linear

Técnicas de Programação Linear, incluindo Programação Linear Inteira, também têm sido utilizadas para a geração de quadros de horários. Na maioria dos casos, busca-se a produção de quadros de horários provadamente ótimos. Considerando que a prova da otimalidade pode implicar a resolução de um número computacionalmente intratável de problemas lineares, trabalhos iniciais como [42] tratavam apenas parte do problema hoje conhecido como o problema de programação de horários. Em [42], 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 [27] 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 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 [24, 25, 28]. Uma particularidade dos modelos utilizados em [24, 28], bastante relacionados com o PPTC, é o fato de que a questão da compatibilidade de horários para professores não é avaliada na função objetivo. Em [28], por exemplo, especifica-se um conjunto de dias de folga para cada professor, já em [24] 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 [26] uma aplicação de Geração de Colunas [43] 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 por esses autores é heurístico, sem garantia de produção da solução ótima.

3.6 Comparações entre os métodos

Apesar da existência de várias publicações que tratam de problemas iguais ou semelhantes, poucos progressos foram realizados no campo de análises comparativas entre as implementações dos diferentes métodos, com eventuais exceções para casos em que os autores implementam as alternativas concorrentes do início [6, 20].

No trabalho de Abramson [39], a comparação com os resultados de Gans [44] é realizada utilizando-se instâncias diferentes das do trabalho anterior, visto que o autor não teve acesso aos dados completos que definiam as instâncias. Instâncias de mesma dimensão e conteúdos aleatórios foram utilizadas para os experimentos comparativos. O autor afirma [39] que os resultados desse experimento não permitem nenhuma comparação absoluta entre os trabalhos.

Schaerf [18], argumenta que suas escolhas com relação à representação de solução e definição da vizinhança são mais apropriadas do que as utilizadas em [20] e [17], respectivamente, sem apresentar resultados experimentais comparativos que justifiquem essa afirmação. De fato, nenhum dos trabalhos considerou o mesmo conjunto de instâncias, o que torna difícil a elaboração de conclusões sobre quais escolhas foram as mais apropriadas. Nos três trabalhos a avaliação dos resultados é realizada considerando comparações com soluções manuais, um critério importante mas nem sempre preciso.

Capítulo 4

Heurísticas para o PPTC

Neste capítulo são apresentados os métodos heurísticos desenvolvidos nesta tese para a produção automatizada de quadros de horários para o PPTC. Inicialmente é apresentada a representação de solução utilizada, seguida pela descrição do método que gera soluções iniciais que posteriormente serão melhoradas pelo método de busca tabu proposto. O desempenho do método proposto é avaliado experimentalmente considerando comparações com os melhores resultados disponíveis para o PPTC [6].

4.1 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 4.1, onde células com “×” indicam indisponibilidade do professor. Como observado em [18], apesar dessa representação ser menos compacta do que uma matriz que armazena as alocações em função 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 2.2 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 ω o número de professores

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 4.1: Fragmento de um quadro de horários gerado

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 ω 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 δ .

4.2 Algoritmo construtivo - CGA-PPTC

O algoritmo construtivo utilizado (Algoritmo 1), consiste basicamente de um procedimento construtivo guloso aleatorizado [36], com poucas diferenças em relação ao construtivo utilizado em [8]. Na construção da solução, a cada iteração, o princípio de alocação das aulas *mais urgentes* nos períodos *mais apropriados* é utilizado. Utilizando os dados de entrada descritos na seção 2.2 e mantendo matrizes de requerimentos e disponibilidades residuais (atualizados de acordo com as alocações já feitas) \ddot{R} e \ddot{P} , respectivamente, o algoritmo computa, a cada iteração, o grau de urgência θ_{pt} , de se alocar uma aula do professor p para a turma t . Após a verificação da maior e menor urgência, uma lista restrita de candidatos LRC é construída (linha 9) incluindo as aulas mais urgentes. A restritividade da lista é controlada pelo parâmetro $\alpha \in [0, 1]$, de maneira que no modo mais restrito ($\alpha = 0$), somente as aulas com maior grau de urgência serão consideradas para alocação. Um outro nível de diversificação é introduzido considerando a seleção do período (linha 16), de modo que períodos ainda não ocupados com baixa disponibilidade de professores

terão uma probabilidade exponencialmente maior de serem escolhidos primeiro [45].

<p>Entrada: $P, T, D, H, \tilde{R}, \tilde{P}, \alpha$</p> <p>Saída: Q</p> <ol style="list-style-type: none"> 1 $\ddot{R} \leftarrow \tilde{R};$ 2 $\ddot{P} \leftarrow \tilde{P};$ 3 $\ddot{t}_{tdh} \leftarrow 1 \quad \forall t \in T, \forall d \in D, \forall h \in H;$ 4 $q_{pdh} \leftarrow 0 \quad \forall p \in P, \forall d \in D, \forall h \in H;$ 5 enquanto $(\sum_{p \in P} \sum_{t \in T} \ddot{r}_{pt} > 0)$ faça <li style="padding-left: 20px;">6 $\theta_{pt} \leftarrow \ddot{r}_{pt} / [1 + \sum_{d \in D} \sum_{h \in H} (\ddot{t}_{tdh} \cdot \ddot{p}_{pdh})] \quad \forall p \in P, t \in T;$ <li style="padding-left: 20px;">7 $\bar{\theta} \leftarrow \max\{\theta_{pt} p \in P, t \in T\};$ <li style="padding-left: 20px;">8 $\underline{\theta} \leftarrow \min\{\theta_{pt} p \in P, t \in T\};$ <li style="padding-left: 20px;">9 $LRC \leftarrow \{(p, t) \theta_{pt} \geq \bar{\theta} - (\bar{\theta} - \underline{\theta})\alpha\};$ 10 selecione aleatoriamente $(p', t') \in LRC;$ 11 se $(\sum_{d \in D} \sum_{h \in H} \ddot{t}_{tdh} \cdot \ddot{p}_{pdh} > 0)$ então <li style="padding-left: 20px;">12 $L \leftarrow \{(d, h) (\ddot{t}_{t'dh} \cdot \ddot{p}_{p'dh} = 1)\};$ 13 senão <li style="padding-left: 20px;">14 $L \leftarrow \{(d, h) (\ddot{p}_{p'dh} = 1)\};$ 15 fim 16 associe probabilidades de seleção para cada $(d, h) \in L$ e sorteie $(d', h') \in L;$ 17 $q_{p'd'h'} \leftarrow t';$ 18 $\ddot{r}_{p't'} \leftarrow \ddot{r}_{p't'} - 1;$ 19 $\ddot{p}_{p'd'h'} \leftarrow \ddot{p}_{p'd'h'} - 1;$ 20 $\ddot{t}_{t'd'h'} \leftarrow \ddot{t}_{t'd'h'} - 1;$ 21 fim
--

Algoritmo 1: Construtivo guloso aleatorizado

4.3 A heurística de melhoramento BT-PPTC

Nesta seção será apresentado o algoritmo de melhoramento de quadros de horários BT-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.

O algoritmo BT-PPTC (Algoritmo 2) parte de um quadro de horários Q gerado no procedimento da seção 4.2 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 [18] 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 [46], 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 $\hat{M}_{|P| \times |T| \times |D| \times |H|}$ ¹ 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:

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

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

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

¹Essa estrutura é ligeiramente diferente de [30]. 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.

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* [46], 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. Nas seções seguintes a implementação que considera somente a memória de transitividade será denominada BT-PPTC-T, enquanto que a implementação que considera somente a memória de residência será denominada BT-PPTC-R. Outra implementação, que considera ambas as memórias de longo prazo, será denominada BT-PPTC-TR. Em BT-PPTC-TR ambas as penalidades são somadas e utilizadas na avaliação de movimentos, durante o período de diversificação. Para propósitos de comparação uma implementação

sem qualquer mecanismo de diversificação também será considerada (BT-PPTC-S).

```

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  $(f(Q \oplus m) < f(Q^*))$  e  $\Delta' < \Delta$ 
    então
14       $\text{melhorMov} \leftarrow m;$ 
15       $\Delta \leftarrow \Delta';$ 
16      se  $f(Q \oplus m) \geq f(Q^*)$  então  $\Delta \leftarrow \Delta + \text{penalidade}$ 
17    fim
18  fim
19  atualize memória de longo prazo ;
20   $Q \leftarrow Q \oplus \text{melhorMov};$ 
21   $\text{validadeTabu}(\text{melhorMov}) \leftarrow$ 
     $\text{random}(\lfloor \text{validTabu} - \varphi \times \text{validTabu} \rfloor, \lceil \text{validTabu} + \varphi \times \text{validTabu} \rceil) + \text{it};$ 
22  atualize  $\text{ListaTabu}$ ;
23  se  $f(Q) < f(Q^*)$  então
24     $Q^* \leftarrow Q; \text{semMelhora} \leftarrow 0;$ 
25    reinicialize memória de longo prazo;
26  senão
27     $\text{semMelhora} ++;$ 
28  fim
29 até critério de parada atingido ;

```

Algoritmo 2: Algoritmo BT-PPTC

instância	limite de tempo
1	90
2	280
3	380
4	870
5	1930
6	1650
7	2650

Tabela 4.1: limite de tempo de cada instância

4.4 Experimentos computacionais

Experimentos computacionais foram feitos para validação dos algoritmos propostos. Os algoritmos foram implementados em C++, sendo que experimentos comparativos também incluíram a avaliação do método GBT-II [8, 6], cujo autor gentilmente cedeu sua implementação. Todos os códigos foram compilados com o compilador GCC [47] 3.2.3, com a opção `-O3`. Os experimentos foram executados em um micro-computador com processador AMD Athlon XP 1800+, 512 Megabytes de RAM, sob o sistema operacional Linux.

A adição dos componentes que medem a infactibilidade das soluções na função objetivo (seção 4.1) é feita com pesos $\omega = 100$ e $\delta = 30$. Os parâmetros do algoritmo GBT-II foram os mesmos de [6].

Como critério de parada padrão para as execuções dos algoritmos estipulou-se um limite de tempo, apresentado na Tabela 4.1, originário de [6], o qual corresponde ao tempo necessário para que ocorresse a estabilização do algoritmo de *Simulated Annealing* implementado por Souza nos experimentos do referido trabalho.

Como os algoritmos contém componentes aleatórios, sua avaliação é realizada considerando o custo médio de solução produzida em dez execuções independentes, com diferentes sementes aleatórias.

Inicialmente experimentos para calibração dos parâmetros dos algoritmos são apresentados. Foram realizados experimentos com a variação do parâmetro *validTabu*. Os valores $validTabu \in \{15, 20, 25, 30\}$ foram utilizados, enquanto os outros parâmetros permaneceram fixos: $ativDiv = 500$, $itDiv = 10$, $\varphi = 0.1$, $\alpha = 0.1$. Nas Tabelas 4.2, 4.3, 4.4 e 4.5 é apresentada a distância média entre as soluções produzidas por cada algoritmo e a melhor solução conhecida para cada instância, considerando diferentes parâmetros. Calcula-se a distância média percentual (dm), considerando o custo médio das soluções

Instância	Valor de <i>validTabu</i>				Média
	15	20	25	30	
1	5,13	2,32	0,78	1,28	2,38
2	3,47	2,21	1,15	1,72	2,14
3	7,94	6,68	2,48	0,92	4,50
4	0,97	0,96	0,82	0,24	0,75
5	4,83	6,32	1,96	0,45	3,39
6	3,45	2,14	0,94	0,30	1,71
7	1,75	1,65	0,54	0,82	1,19
Média	3,93	3,18	1,24	0,82	2,29

Tabela 4.2: Distância média das melhores soluções conhecidas - BT-PPTC-S

produzidas cm e o valor da melhor solução conhecida bk , de modo que $dm = \frac{cm-bk}{bk} \times 100$. Como mostram os resultados da Tabela 4.2 para o algoritmo BT-PPTC-S, que não utiliza memória de longo prazo, os melhores resultados ocorreram com valores altos para o parâmetro *validTabu*. De qualquer maneira, implementações que utilizam a estratégia de diversificação com memória de longo prazo obtiveram melhores resultados, para qualquer valor de *validTabu*, do que o algoritmo BT-PPTC-S. Enquanto em média BT-PPTC-T produziu melhores resultados que BT-PPTC-R, os melhores resultados em geral foram obtidos com a implementação com diversificação que considera ambos os tipos de memória de longo prazo propostas: BT-PPTC-TR, usando valores baixos para o parâmetro *validTabu*. O algoritmo BT-PPTC-TR com *validTabu* = 15 produziu, em média, soluções somente 0,2% distantes das melhores soluções conhecidas². Os resultados dos experimentos seguintes com os diferentes algoritmos consideram apenas os parâmetros que produziram os melhores resultados: *validTabu* = 30 para BT-PPTC-S e *validTabu* = 15 para os demais algoritmos, por exemplo.

Uma visão diferente dos resultados previamente apresentados é mostrada na Tabela 4.6, onde os custos médios de solução gerados pelos algoritmos propostos são comparados com os custos médios do algoritmo GTB-II, produzidos nos mesmos limites de tempo. Os melhores resultados são destacados em negrito. Esses resultados mostram que embora pouca diferença seja observável entre as versões com diferentes composições de memórias de longo prazo, fica clara a superioridade das versões que utilizam estratégias de diversificação informadas sobre versões que não incorporam esse recurso, especificamente GBT-III e BT-PPTC-S.

²Esses experimentos foram realizados antes dos experimentos com técnicas de programação inteira mista, de modo que as soluções conhecidas até o momento não incluíam os resultados dos experimentos com essa técnica.

Instância	Valor de <i>validTabu</i>				Média
	15	20	25	30	
1	0,35	0,30	0,15	0,10	0,22
2	0,00	0,12	0,52	0,38	0,25
3	0,46	1,03	0,27	0,00	0,44
4	0,36	0,27	0,27	0,31	0,30
5	0,22	0,21	0,24	0,37	0,26
6	0,19	0,27	0,45	0,45	0,34
7	0,00	0,19	0,43	0,55	0,29
Média	0,22	0,34	0,33	0,31	0,30

Tabela 4.3: Distância média das melhores soluções conhecidas - BT-PPTC-T

Instância	Valor de <i>validTabu</i>				Média
	15	20	25	30	
1	0,59	0,15	0,00	0,20	0,23
2	0,61	0,32	0,20	0,41	0,39
3	0,14	0,60	0,76	1,17	0,66
4	0,24	0,21	0,34	0,37	0,29
5	0,30	0,00	0,35	0,31	0,24
6	0,00	0,39	0,39	0,41	0,30
7	0,45	0,63	0,38	0,72	0,55
Média	0,33	0,33	0,34	0,51	0,38

Tabela 4.4: Distância média das melhores soluções conhecidas - BT-PPTC-R

Instância	Valor de <i>validTabu</i>				Média
	15	20	25	30	
1	0,25	0,10	0,15	0,25	0,19
2	0,26	0,44	0,61	0,61	0,48
3	0,73	0,30	0,53	0,53	0,52
4	0,00	0,15	0,19	0,43	0,19
5	0,05	0,09	0,56	0,55	0,31
6	0,08	0,19	0,53	0,59	0,35
7	0,04	0,20	0,36	0,44	0,26
Média	0,20	0,21	0,42	0,49	0,33

Tabela 4.5: Distância média das melhores soluções conhecidas - BT-PPTC-TR

Instância	GBT-II	BT-PPTC-S	BT-PPTC-T	BT-PPTC-R	BT-PPTC-TR
1	204,80	205,30	203,40	203,00	203,20
2	350,10	349,20	343,30	344,40	344,20
3	455,70	440,90	438,90	439,50	440,10
4	686,30	670,50	671,30	670,30	668,90
5	796,30	782,70	780,90	779,20	779,60
6	799,10	781,50	780,70	782,20	779,80
7	1.076,20	1.063,80	1.055,20	1.061,90	1.055,60

Tabela 4.6: Custos médios produzidos - execuções com limites de tempo

Para que se avalie a qualidade das soluções obtidas com os métodos propostos levando em conta sua aplicação e para verificação do quão significativa foi a melhoria do método BT-PPTC-TR produzida nas soluções geradas pela heurística construtiva, é apresentado na Tabela 4.7 um detalhamento do número médio de violações de cada um dos requisitos que compõe a função objetivo nas soluções produzidas pelo algoritmo construtivo e nas soluções melhoradas por BT-PPTC-TR. Considera-se a violação das restrições fortes relaxadas (ver seção 4.1) e também a violação das restrições fracas. A coluna “falta de aulas duplas” indica o não atendimento desses pedidos. Considere os dados do PPTC e as variáveis da formulação \mathcal{F}_1 , desse modo, a violação em unidades dos pedidos de aulas duplas (coluna #) é: $\sum_{p \in P} \sum_{t \in T} g_{pt}$ e o valor percentual (coluna %): $(\sum_{p \in P} \sum_{t \in T} g_{pt}) / (\sum_{p \in P} \sum_{t \in T} \tilde{g}_{pt}) \times 100$. De modo semelhante, a violação em unidades do número de buracos é: $\sum_{p \in P} \sum_{d \in D} b_{pd}$ e o valor percentual é: $(\sum_{p \in P} \sum_{d \in D} b_{pd}) / (\sum_{p \in P} \sum_{t \in T} \tilde{r}_{pt}) \times 100$. Na última coluna temos uma medida do não atendimento do requisito compacidade de horários para professores, aqui denotado por grau de esparsidade³. Nessa medida, um quadro de horários está tão compacto quanto possível (aulas agrupadas em poucos dias letivos para cada professor), quando a esparsidade é igual a 1.0. A computação da esparsidade de cada solução considera o número total de dias letivos de cada professor na solução ($\sum_{p \in P} \sum_{d \in D} v_{pd}$) e um limite inferior⁴ para esse valor ($\sum_{p \in P} \lceil (\sum_{t \in T} r_{pt} / |H|) \rceil$). Desse modo, a esparsidade de um quadro de horários Q , definido em termos das variáveis da formulação \mathcal{F}_1 é:

$$esparsidade(Q) = \frac{\sum_{p \in P} \sum_{d \in D} v_{pd}}{\sum_{p \in P} \lceil (\sum_{t \in T} r_{pt} / |H|) \rceil} \quad (4.1)$$

Como se observa na Tabela 4.7, as soluções oferecidas pelo método BT-PPTC-TR, além

³Não confundir com a esparsidade da instância, apresentada no Capítulo 2.

⁴Quadros de horários com esparsidade > 1.0 podem estar com compacidade máxima, visto que o limite inferior utilizado no denominador de 4.1 pode ser melhorado, como será discutido no Capítulo 5.

inst.	CGA-PPTC						
	violações						
	restrições fortes		restrições fracas				esparsidade
	conflitos	excesso aulas	falta de aulas duplas		buracos		
#			%	#	%		
1	0,0	0,5	15,1	71,5	17,2	22,9	1,6
2	0,0	0,0	24,3	83,8	24,8	16,5	1,3
3	0,3	2,5	2,0	50,0	31,2	15,6	1,4
4	4,3	0,9	35,5	53,8	21,0	7,0	1,2
5	0,0	0,2	54,1	76,1	46,4	14,3	1,5
6	0,2	0,0	53,7	85,2	53,4	15,3	1,4
7	0,5	0,2	69,6	82,9	74,1	14,8	1,3
BT-PPTC-TR							
1	0,0	0,0	1,9	9,0	4,1	5,5	1,2
2	0,0	0,0	7,3	25,2	1,3	0,9	1,0
3	0,0	0,0	0,4	10,0	5,4	2,7	1,1
4	0,0	0,0	19,4	29,4	3,8	1,3	1,0
5	0,0	0,0	13,7	19,3	3,5	1,1	1,1
6	0,0	0,0	15,4	24,4	8,8	2,5	1,0
7	0,0	0,0	23,0	27,4	10,6	2,1	1,0

Tabela 4.7: Valores médios dos componentes da função de custo em soluções geradas pelo algoritmo construtivo e em soluções do alg. BT-PPTC-TR

de sempre serem factíveis, em geral são ótimas, ou bastante próximas da solução ótima, com relação a restrição fraca com maior peso na função objetivo, a compacidade do horário dos professores. Observa-se também uma melhora considerável no atendimento das demais restrições fracas em relação à solução oferecida pelo método construtivo, apesar da distância da otimalidade nesses componentes não ser tão evidente, considerando somente os dados da Tabela 4.7.

Em um outro experimento procurou-se verificar o quão rápido cada método produzia soluções com uma medida mínima de qualidade. O experimento, inspirado no trabalho de [48], considera a distribuição empírica de probabilidade de se atingir um determinado valor alvo na função de custo da solução incumbente, ou seja, considera-se o tempo necessário para produção de soluções com custo menor ou igual o valor alvo. Os valores alvo considerados nos experimentos deste trabalho foram definidos de modo a serem custos sub-ótimos que permitam que mesmo o mais lento dos algoritmos termine em um tempo computacional razoável. Nesse experimento foram avaliados os métodos GTB-II e BT-PPTC-TR, sendo que os tempos de execução para 150 execuções em cada método foram computados. Os resultados de cada algoritmo foram plotados associando-se o i -ésimo

Instância	GTB-II			BT-PPTC-TR		
	25%	50%	75%	25%	50%	75%
1	7,64	9,57	12,15	2,13	3,36	6,39
2	21,39	26,57	34,68	9,03	13,48	19,71
3	28,57	46,84	85,41	16,29	27,66	46,47
4	49,22	92,57	146,50	2,65	3,40	5,45
5	47,79	62,85	102,20	27,63	37,85	54,51
6	35,81	48,00	72,12	25,20	33,97	44,38
7	92,41	150,72	287,48	89,57	118,82	155,72

Tabela 4.8: Tempo (em segundos) para que 25%, 50% e 75% das execuções atinjam o valor alvo

menor tempo de execução t_i com a probabilidade $p_i = (i - \frac{1}{2})/150$, gerando os pontos $z_i = (t_i, p_i)$, for $i = 1, \dots, 150$. Os resultados mostram que o método BT-PPTC-TR atinge valores altos de probabilidade ($\geq 50\%$) em tempos significativamente menores, em todos os problemas. Resultados desse experimento são apresentados nas Figuras 4.2, 4.3, 4.4 e 4.5.

A diferença é acentuada principalmente no problema 4, o qual apresenta uma taxa esparsidade bastante baixa (ver seção 2.3). Esse resultado pode estar relacionado com o fato de que o procedimento “Interclasses-Intraclasses” do algoritmo GTB-II trabalha com movimentos em períodos de disponibilidade do professor, mais escassos nesse problema. Outra análise, levando em conta todos os problemas, mostra que no tempo em que 95% das execuções de BT-PPTC-TR atingiram o valor alvo, em média, somente 64% das execuções de GTB-II atingiram o valor alvo. A Tabela 4.8 apresenta os tempos de execução médios necessários para que GTB-II e BT-PPTC-TR alcancem diferentes probabilidades de atingirem os valores alvo.

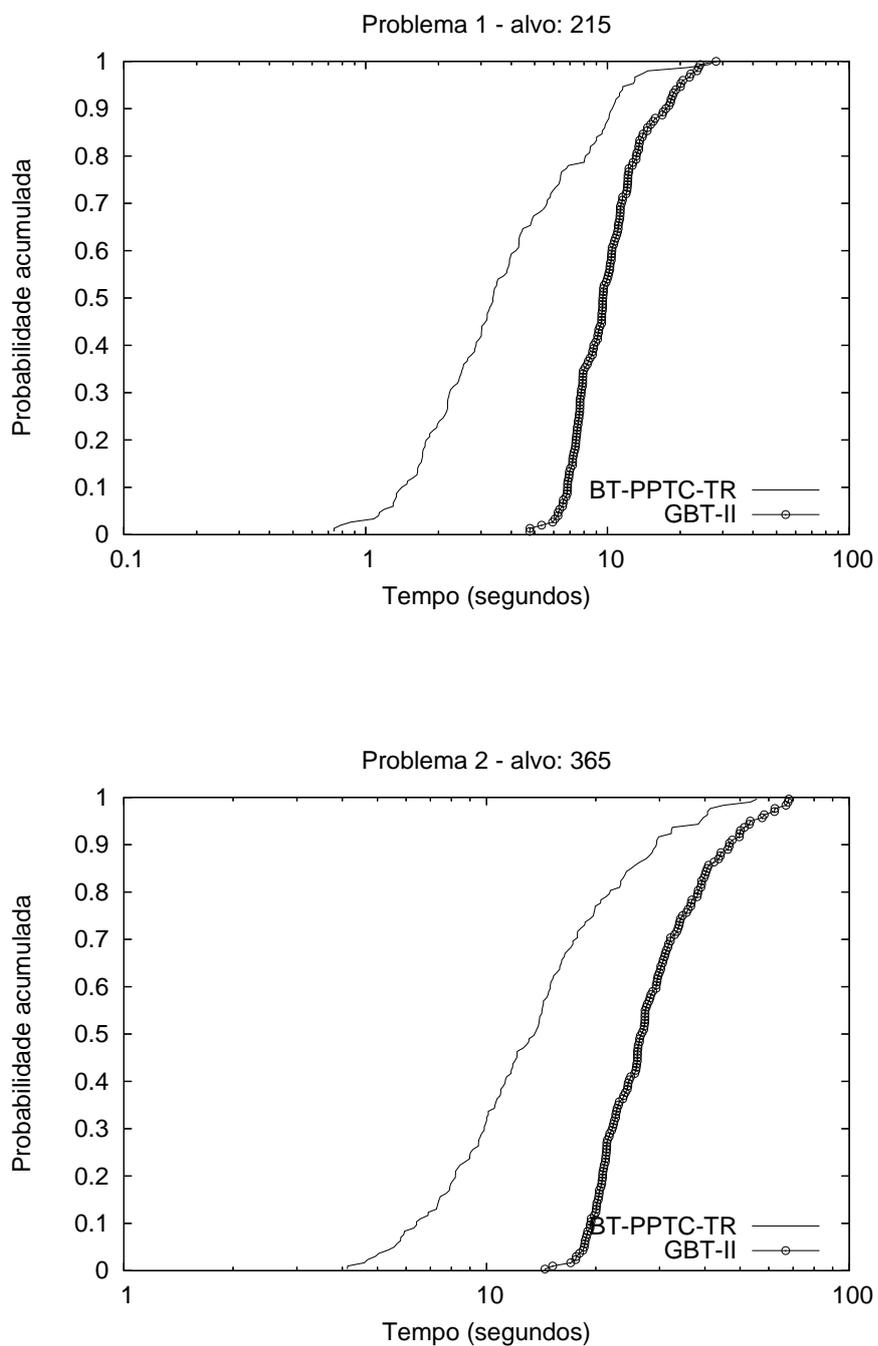


Figura 4.2: Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 1 e 2

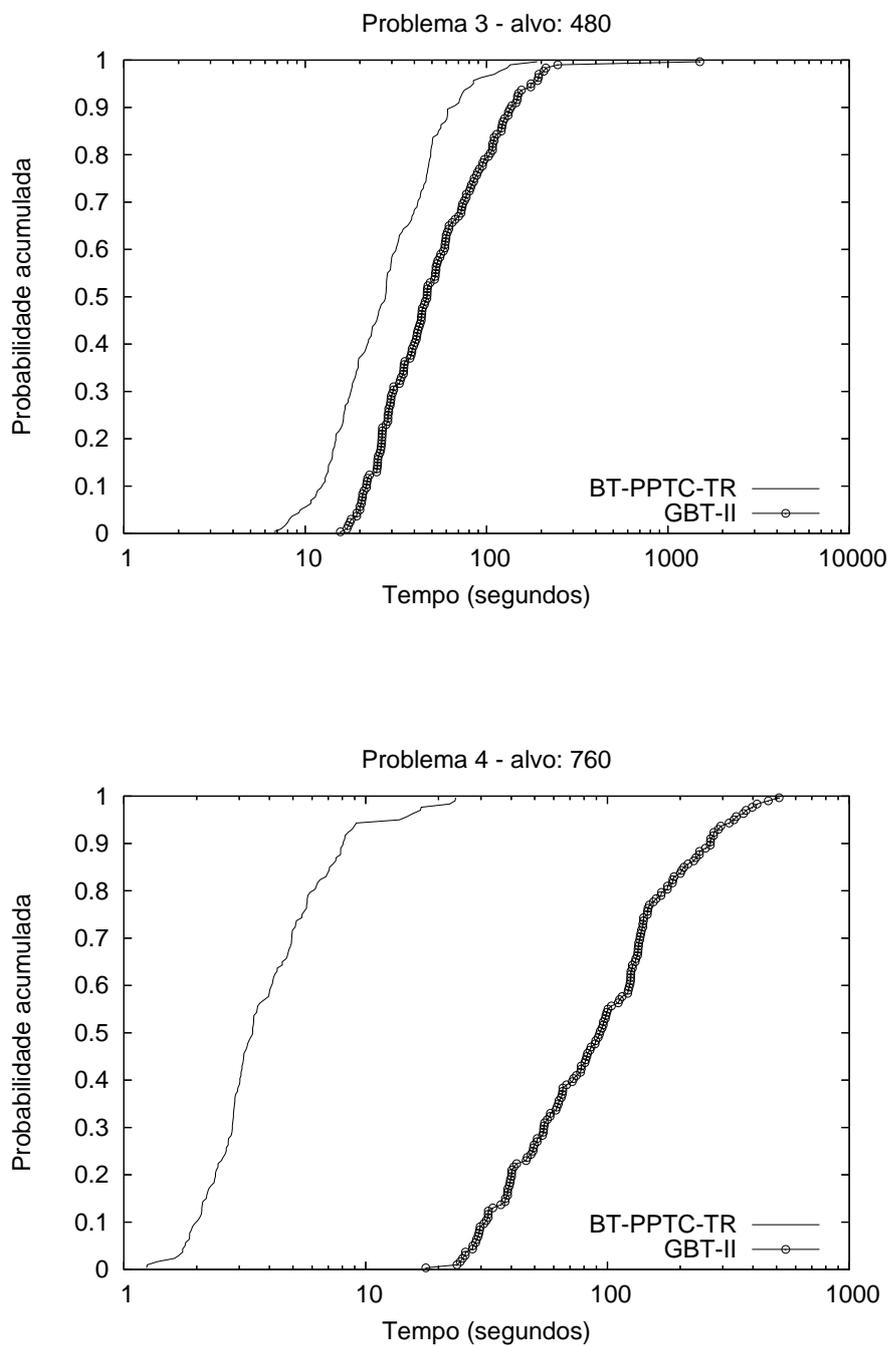


Figura 4.3: Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 3 e 4

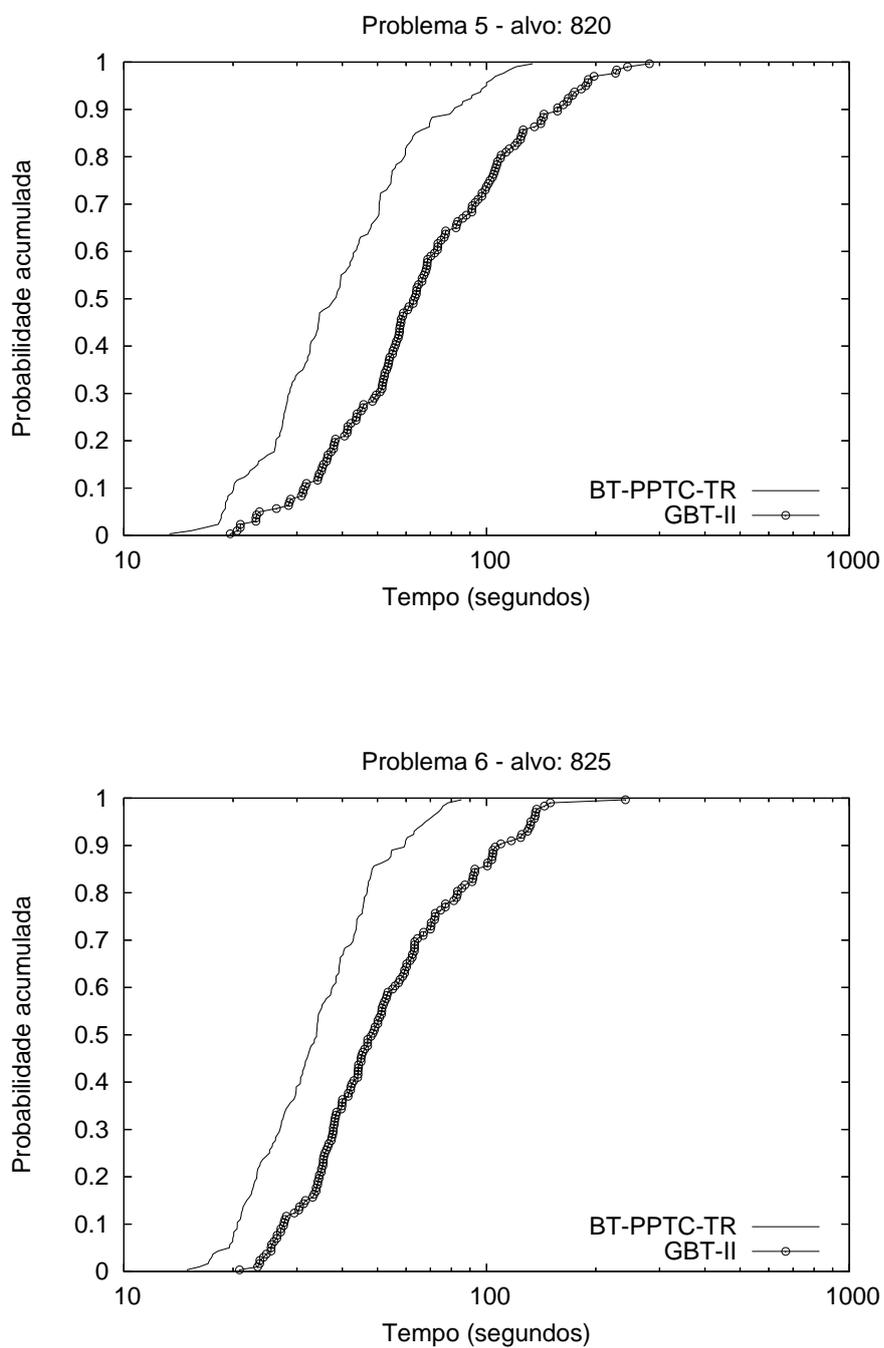


Figura 4.4: Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para as instâncias 5 e 6

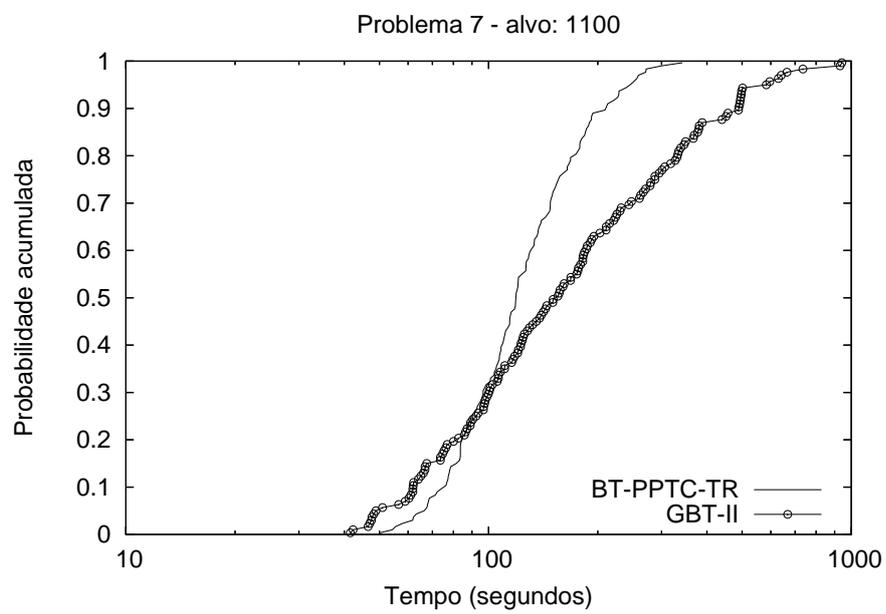


Figura 4.5: Distribuição empírica de probabilidade de se atingir um valor alvo em função do tempo para a instância 7

Capítulo 5

Programação Linear Inteira Mista para o PPTC

A utilização de Programação Linear Inteira Mista permite que inúmeros problemas de programação de horários sejam fielmente modelados, como as referências da seção 3.5 indicam. 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[49] na Internet. Dada a característica enumerativa dos métodos de resolução de PLIM, que podem requerer a resolução de um número potencialmente exponencial de subproblemas, tem-se uma limitação no conjunto de instâncias que podem ser resolvidas considerando as limitações de tempo e recursos computacionais existentes. Em particular, deve-se observar que o desempenho dos resolvidores de PLIM depende fortemente da formulação utilizada. A formulação utilizada por Drexler [27], por exemplo, não permitiu a resolução de nenhuma instância de tamanho razoável. Neste capítulo são discutidas alternativas para que a formulação \mathcal{F}_1 seja mais tratável computacionalmente, bem como é desenvolvida uma nova formulação, com um número exponencial de variáveis e restrições.

Seja X o conjunto de soluções factíveis para o PPTC (seção 2.2, formulação \mathcal{F}_1), a determinação da envoltória convexa $\text{conv}(X)$ permitiria a reformulação do PPTC com a remoção das restrições de integralidade (2.18), de modo que o mesmo poderia ser resolvido como um problema de programação linear. Dada a dificuldade de determinação de $\text{conv}(X)$ para problemas combinatórios difíceis como o PPTC, o que se busca são formulações que produzam boas aproximações de $\text{conv}(X)$. Essas formulações, denominadas formulações fortes, apresentam melhores limites duais, de modo que, geralmente, são mais facilmente resolvidas através de programas resolvidores de PLIM que utilizam métodos do tipo *branch-and-bound* [50, 51].

5.1 Cortes

Seja LP o poliedro definido pelas restrições do PPTC menos a restrição de integralidade, procura-se encontrar desigualdades que não sejam válidas para LP e sejam válidas para o PPTC. Nesta seção novos cortes são apresentados para fortalecer a formulação \mathcal{F}_1 do PPTC.

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:

$$\sum_{t \in T} \sum_{h=h'}^{h''-1} y_{ptdh} \leq v_{pd} \left\lfloor \frac{|\{h', \dots, h''\}|}{2} \right\rfloor \quad (5.1)$$

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:

$$\sum_{p \in P} \sum_{h \in \widehat{G}_{pd} \cap \{h', \dots, h'' - 1\}} y_{ptdh} \leq \left\lfloor \frac{|\{h', \dots, h''\}|}{2} \right\rfloor \quad (5.2)$$

As restrições 5.1 e 5.2, além de melhorar a formulação \mathcal{F}_1 , tem uma outra função, que é a de permitir uma interessante extensão do modelo apresentado para o PPTC. Na seção 2.2, indica-se que o dado de entrada \tilde{m}_{pt} , determinante do máximo de aulas que um professor p pode lecionar para uma turma t , restringe-se ao domínio $\{1, 2\}$ (página 8). Essa limitação decorre do fato que as restrições 2.15 e 2.16 não são suficientes para que valores válidos para g_{pt} sejam obtidos quando a formulação \mathcal{F}_1 é utilizada com valores de $\tilde{m}_{pt} > 2$. Desse modo, a incorporação de 5.1 e 5.2 permite que \mathcal{F}_1 modele uma generalização do PPT, visto que assim pode-se substituir $\tilde{m}_{pt} \in \{1, 2\}$ por $\tilde{m}_{pt} \in \{|H|\}$. Essa extensão pode ser útil em casos específicos de problemas de programação de horários. No desenvolvimento deste trabalho, através de entrevistas com programadores de horários em escolas, verificou-se a necessidade de construção de horários com blocos maiores de aulas contíguas para algumas disciplinas¹. Novas restrições e variáveis poderiam ser incluídas no modelo para

¹Em especial, quadros de horários com essas características foram observados em escolas agrotécnicas, onde algumas disciplinas são realizadas fora da escola, requerendo um turno semanal completo para a sua realização.

avaliar o atendimento de outros tipos de distribuições de aulas.

Ainda, considerando que somente aulas duplas 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:

$$\sum_{h_1 \in \hat{G}_{pd} \setminus \{h, h-1\}} y_{ptdh_1} + x_{ptdh} \leq v_{pd} \quad (5.3)$$

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 [6], que considera o número mínimo de dias que cada professor deve comparecer à escola:

$$\sum_{d \in D} v_{pd} \geq \max \left\{ \left\lceil \frac{\sum_{t \in T} \tilde{r}_{pt}}{|H|} \right\rceil, \max_{t \in T} \left\lceil \frac{\tilde{r}_{pt}}{\tilde{m}_{pt}} \right\rceil \right\} \quad \forall p \in P \quad (5.4)$$

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:

$$\begin{aligned} \sum_{d \in D} v_{pd} &\geq \bar{k} \quad \forall p \in P, \bar{k} \in D \\ \left| \sum_{k=1}^{\bar{k}} \dot{p}_{pk} \geq \sum_{t \in T} \tilde{r}_{pt} \wedge (\bar{k} - 1 = 0 \vee \sum_{k=1}^{\bar{k}-1} \dot{p}_{pk} < \sum_{t \in T} \tilde{r}_{pt}) \right. & \quad (5.5) \end{aligned}$$

Claramente, a utilização do lado direito de 5.5 na primeira parcela da função max em 5.4 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 ati-

vidades 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:

$$\begin{aligned} \sum_{d \in D} v_{pd} &\geq \sum_{t \in T} (\bar{k} + 1) x_{ptd'h} \\ &\quad \forall p \in P, d' \in D \\ \left| \sum_{d''=1}^{\bar{k}} \dot{p}_{pd''} \geq \sum_{t \in T} \tilde{r}_{pt} \wedge (\bar{k} - 1 = 0 \vee \sum_{d''=1}^{\bar{k}-1} \dot{p}_{pd''} < \sum_{t \in T} \tilde{r}_{pt}), \right. & \quad (5.6) \\ &\quad \left. \sum_{k''=1}^{\bar{k}-1} \dot{p}_{k''} + \sum_{h \in H} \tilde{p}_{pd'h} < \sum_{t \in T} \tilde{r}_{pt} \right) \end{aligned}$$

Como exemplo ilustrativo de um corte que pode ser gerado a partir de 5.6, 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 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:

$$\tilde{m}_{pt} v_{pd} \geq \sum_{h \in H} x_{ptdh} \quad \forall p \in P, t \in T, d \in D \quad (5.7)$$

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 2.2), 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 :

$$\check{b}_{pdh} \geq \sum_{t \in T} (x_{ptdh} + x_{ptdh+2} - x_{ptdh+1}) - 1 \quad \forall p \in P, d \in D, h \in \widehat{B}_{pd} \quad (5.8)$$

$$b_{pd} \geq \sum_{h \in \widehat{B}_{pd}} \check{b}_{pdh} \quad \forall p \in P, d \in D \quad (5.9)$$

5.2 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 [52]. 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 [53], 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 2.2, 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 para o professor p na turma t no padrão de alocação j . A variável de decisão λ_{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:

$$\text{minimizar} \quad \sum_{p \in P} \sum_{d \in D} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot c_{pj} + \sum_{p \in P} \sum_{t \in T} w_p''' \cdot g_{pt} \quad (5.10)$$

sujeito a

$$\sum_{j \in \check{P}_{pd}} \lambda_{pjd} = 1 \quad \forall p \in P, d \in D \quad (5.11)$$

$$\sum_{p \in P} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot \check{x}_{pjth} \leq 1 \quad \forall t \in T, d \in D, h \in H \quad (5.12)$$

$$\sum_{j \in \check{P}_{pd}} \sum_{d \in D} \sum_{h \in D} \lambda_{pjd} \cdot \check{x}_{pjth} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \quad (5.13)$$

$$\sum_{d \in D} \sum_{j \in \check{P}_{pd}} \lambda_{pjd} \cdot \check{g}_{pjt} + g_{pt} \geq \tilde{g}_{pt} \quad \forall p \in P, t \in T \quad (5.14)$$

$$\lambda_{pjd} \in \{0, 1\} \quad \forall p \in P, d \in D, j \in \check{P}_{pd} \quad (5.15)$$

$$g_{pt} \in \mathbb{N} \quad \forall p \in P, t \in T \quad (5.16)$$

Onde 5.11, 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 ², 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 [43]. 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.

²Em 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.

5.2.1 Geração de Colunas

Inicialmente, para a resolução de \mathcal{F}_{2r} são inseridas somente colunas (variáveis λ) 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 alguma disponibilidade, o padrão de alocação com menor custo reduzido. Essa computação utiliza variáveis duais associadas às restrições envolvidas. Denotam-se aqui por μ , ν , π e κ as variáveis duais associadas com as restrições 5.11, 5.12, 5.13 e 5.14, 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}_{th} ($t \in T, h \in H$), sendo que $\check{x}_{th} = 1$ caso o professor p esteja alocado para uma aula com a turma t no dia d e período h , $\check{x}_{th} = 0$ 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:

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

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 \tilde{m}_{pt} , por questões pedagógicas, visto que as aulas são distribuídas no decorrer da semana.

5.2.2 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, 5.17, 5.18 e 5.19, controlam a seleção única de uma carga de trabalho diária para os professores e a amarração de tal escolha com as colunas λ , já as restrições 5.20, 5.21 e 5.22 garantem a seleção da carga horária para encontros diários entre professores e turmas e a relação dessas escolhas com as variáveis λ , como segue:

$$\sum_{h \in \{0, \dots, |H|\}} \check{a}_{pdh} = 1 \quad \forall p \in P, d \in D \quad (5.17)$$

$$\sum_{j \in \check{P}_{pdh}} \lambda_{pjd} = \check{a}_{pdh} \quad \forall p \in P, d \in D, h \in H \quad (5.18)$$

$$\check{a}_{pdh} \in \{0, 1\} \quad \forall p \in P, d \in D, h \in H \quad (5.19)$$

$$\sum_{h \in \{0, \dots, |H|\}} \check{a}_{ptdh} = 1 \quad \forall p \in P, t \in T, d \in D \quad (5.20)$$

$$\sum_{j \in \check{P}_{ptdh}} \lambda_{pjd} = \check{a}_{ptdh} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (5.21)$$

$$\check{a}_{ptdh} \in \{0, 1\} \quad \forall p \in P, t \in T, d \in D, h \in H \quad (5.22)$$

As restrições 5.17 e 5.20 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 λ aparecem agora em duas restrições adicionais. \mathcal{P}'_{pd} consiste em \mathcal{P}_{pd} aumentado das variáveis $\check{a}_h \in \{0, 1\} \forall h \in \{0, \dots, |H|\}$, que indicam a seleção de uma carga horária total h para um professor p em um dia d ($\check{a}_h = 1, \check{a}_h = 0$ c.c.), bem como as variáveis $\check{a}_{th} \forall t \in T, h \in \{0, \dots, |H|\}$, que indicam a seleção de uma carga horária diária com h aulas para o professor p em uma turma t em um dia d ($\check{a}_{th} = 1, \check{a}_{th} = 0$ c.c.). As novas

variáveis são amarradas com a escolha do padrão de trabalho com as seguintes restrições:

$$\sum_{h \in H} h \cdot \check{a}_h = \sum_{h \in H} \sum_{t \in T} \check{x}_{th} \quad (5.23)$$

$$\sum_{h \in \{0, \dots, H\}} \check{a}_h = 1 \quad (5.24)$$

$$\check{a}_h \in \{0, 1\} \quad \forall h \in \{0, \dots, |H|\} \quad (5.25)$$

$$\sum_{h \in H} h \cdot \check{a}_{th} = \sum_{h \in H} \check{x}_{th} \quad \forall t \in T \quad (5.26)$$

$$\sum_{h \in \{0, \dots, H\}} \check{a}_{th} = 1 \quad \forall t \in T \quad (5.27)$$

$$\check{a}_{th} \in \{0, 1\} \quad \forall t \in T, h \in \{0, \dots, |H|\} \quad (5.28)$$

Finalmente, em \mathcal{P}'_{pd} a computação da porção \check{d}_{pd} do custo reduzido, referente às variáveis duais, deve incorporar as equações 5.18 e 5.21. Desse modo, definimos \check{d}_{pd} para \mathcal{P}'_{pd} , agora considerando as variáveis duais ι e ϑ , relacionadas às restrições 5.18 e 5.21, respectivamente:

$$\begin{aligned} \check{d}_{pd} = & \mu_{pd} + \sum_{t \in T} \sum_{h \in H} \nu_{tdh} \cdot \check{x}_{th} + \sum_{t \in T} \sum_{h \in H} \pi_{pt} \cdot \check{x}_{th} + \\ & \sum_{t \in T} \sum_{h \in \hat{G}_{pd}} \check{y}_{th} \cdot \kappa_{pt} + \sum_{h \in H} \check{a}_h \cdot l_{pdh} + \sum_{t \in T} \sum_{h \in H} \check{a}_{th} \cdot \vartheta_{ptdh} \end{aligned} \quad (5.29)$$

Um dos cortes possíveis válidos para a formulação \mathcal{F}'_2 são cortes 0-1 *knapsack cover* [54, 55, 56]. Considere as restrições implícitas existentes em \mathcal{F}'_2 :

$$\sum_{h \in H} \sum_{d \in D} h \cdot \check{a}_{pdh} = \sum_{t \in T} \tilde{r}_{pt} \quad \forall p \in P \quad (5.30)$$

$$\sum_{h \in H} \sum_{d \in D} h \cdot \check{a}_{ptdh} = \tilde{r}_{pt} \quad \forall p \in P, t \in T \quad (5.31)$$

Como exemplo de um corte de *cover* levando em conta as restrições 5.30 considere um professor p e um conjunto de pares ordenados \hat{C} com elementos $(d_1, h_1), \dots, (d_{\hat{c}}, h_{\hat{c}})$, $d_i \in D \forall i \in \{1, \dots, \hat{c}\}$, $h_i \in H \forall i \in \{1, \dots, \hat{c}\}$ tal que:

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

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

$$\sum_{i \in \{1, \dots, \dot{c}\}} \check{a}_{pd_i h_i} \leq \dot{c} - 1 \quad (5.32)$$

Apesar da possibilidade de utilização de uma implementação específica para a separação dos cortes de *cover*, como a descrita em [57], foi utilizada a abordagem mais genérica dos planos de cortes por enumeração, os *Fenchel Cuts*, descritos por Boyd et al. em [58, 59], 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 5.17, 5.30 e 5.19, já no segundo foram considerados os pontos extremos do poliedro definido pelas equações 5.20, 5.31 e 5.22. 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 fracionária³ 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 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.

³Solução fracionária considerando as variáveis \check{a}

5.2.3 O Algoritmo de Geração de Colunas e Cortes

O algoritmo de Geração de Colunas e Cortes para o PPTC (Algoritmo 3), 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 [53]. 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 [53], 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  $\bar{c}_{pd} < 0$  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      para cada  $t \in T$  faça
26        Resolva  $\mathcal{S}_{pt}$ ;
27        se Corte violado encontrado então
28          Adicione corte em  $\mathcal{PL}$ ;
29           $novosCortes \leftarrow novosCortes + 1$ ;
30        fim
31      fim
32    fim
33  fim
34 fim

```

Algoritmo 3: Algoritmo de Geração de Colunas e Cortes GCCPPTC

Formulação	Cortes incluídos		
	{5.1, 5.2, 5.3}	{5.4, 5.5, 5.7}	{5.9}
$\mathcal{F}_{1,0}$			
$\mathcal{F}_{1,1}$	✓		
$\mathcal{F}_{1,2}$		✓	
$\mathcal{F}_{1,3}$			✓
$\mathcal{F}_{1,4}$	✓	✓	✓

Tabela 5.1: Diferentes variações da formulação \mathcal{F}_1 avaliadas.

5.2.4 Experimentos Computacionais

Experimentos computacionais foram realizados para avaliação dos melhoramentos propostos para a formulação \mathcal{F}_1 , bem como para a avaliação da nova formulação estendida \mathcal{F}'_2 .

Todos os códigos foram implementados em C++, utilizando a *callable library* [60] do pacote CPLEX 10 [61] para a resolução dos problemas lineares. A solução do problema de *pricing* na implementação corrente é realizada pela enumeração dos padrões. Os códigos foram compilados com o compilador GCC [47] 4 com a opção `-O2`. Os testes foram executados em um computador Dell Optiplex G620, com um processador Pentium D 3.0Ghz e 2GB de memória RAM, rodando o sistema operacional Linux.

Os resultados apresentados nesta seção incluem os limites inferiores e superiores, indicados por \mathcal{LI} e \mathcal{LS} , respectivamente, bem como a distância percentual relativa entre ambos, indicada por $gap(\%)$. Dado um limite inferior li e um superior ls o $gap(\%)$ é: $\frac{ls-li}{li} * 100$. Os melhores limites são formatados em negrito. O tempo de execução dos experimentos foi medido utilizando-se o tempo de CPU em segundos e é indicado por `t.cpu(s)`.

Primeiramente avaliou-se o desempenho da formulação \mathcal{F}_1 considerando a qualidade dos limites duais fornecido pela relaxação linear, bem como o tempo computacional necessário para a resolução do problema linear relaxado. Diferentes variações da formulação \mathcal{F}_1 foram consideradas, caracterizadas por terem diferentes composições de cortes, como detalhado na Tabela 5.1. Os resultados são apresentados na Tabela 5.2.

Como pode-se observar, os cortes propostos mostraram-se efetivos em melhorar os limites duais que são obtidos com a formulação \mathcal{F}_1 , visto que os limites inferiores foram

inst.	$\mathcal{F}_{1,0}$			$\mathcal{F}_{1,1}$			$\mathcal{F}_{1,2}$			$\mathcal{F}_{1,3}$			$\mathcal{F}_{1,4}$			\mathcal{LS}
	\mathcal{LI}	$gap(\%)$	t.cpu(s)	\mathcal{LI}	$gap(\%)$	t.cpu(s)	\mathcal{LI}	$gap(\%)$	t.cpu(s)	\mathcal{LI}	$gap(\%)$	t.cpu(s)	\mathcal{LI}	$gap(\%)$	t.cpu(s)	
1	135,0	49,6	0,2	138,6	45,7	0,5	189,0	6,9	0,2	135,0	49,6	0,2	189,0	6,9	0,5	202
2	270,0	23,3	1,2	270,0	23,3	3,2	333,0	0,0	1,9	270,0	23,3	1,4	333,0	0,0	5,3	333
3	360,0	17,5	0,5	360,0	17,5	0,6	414,0	2,2	1,0	360,0	17,5	0,5	414,0	2,2	1,7	423
4	600,0	8,8	3,1	605,0	7,9	6,9	639,0	2,2	4,2	600,0	8,8	4,2	643,0	1,6	7,8	653
5	585,0	30,9	6,4	596,0	28,5	23,0	756,0	1,3	13,4	585,0	30,9	7,7	756,0	1,3	47,0	766
6	630,0	20,6	8,0	630,4	20,6	29,7	738,0	3,0	17,6	630,0	20,6	8,9	738,0	3,0	53,0	760
7	900,0	14,3	23,7	900,0	14,3	90,1	999,0	3,0	41,1	900,0	14,3	25,6	999,0	3,0	160,0	1.029

Tabela 5.2: Resultados da formulação \mathcal{F}_1

melhorados sempre que cortes foram inseridos, com exceção da formulação $\mathcal{F}_{1,3}$ que incluía somente os cortes 5.9. O conjunto de cortes que sozinho oferece os melhores limites aparece na formulação $\mathcal{F}_{1,2}$. Tal resultado já era esperado visto que para as instâncias consideradas o componente com maior peso na função objetivo é a capacidade de horários para professores.

A Tabela 5.3 apresenta os resultados considerando a formulação estendida com geração de colunas e cortes. Os limites duais e o tempo computacional são apresentados para diferentes formulações, sendo que os resultados da versão mais completa são apresentados na coluna CCGCTTPC, que corresponde a execução do Algoritmo 3. As colunas \mathcal{F}_2 e \mathcal{F}_2^* correspondem aos resultados da formulação \mathcal{F}_2 sem a adição de nenhum corte e a formulação \mathcal{F}_2 incluindo somente os cortes 5.4 e 5.5, respectivamente. Essas duas formulações são resolvidas através de versões mais simples do Algoritmo 3. Como pode-se observar, o algoritmo CCGCTTPC produziu os melhores limites duais conhecidos até o momento, superando a melhor formulação baseada em \mathcal{F}_1 em todos os casos, exceto na instância 2, onde os limites inferiores produzidos foram iguais. Em três casos o limite inferior ótimo foi obtido, sendo que no pior caso obteve-se um *gap*(%) de apenas 1,2. O custo computacional de resolução da formulação com geração de colunas e cortes foi até 16 vezes maior do que da formulação $\mathcal{F}_{1,4}$.

inst.	$\mathcal{F}_{1,4}$			\mathcal{F}_2			\mathcal{F}_2^*			CCGCTTPC		
	\mathcal{LI}	gap(%)	t.cpu(s)	\mathcal{LI}	gap(%)	t.cpu(s)	\mathcal{LI}	gap(%)	t.cpu(s)	\mathcal{LI}	gap(%)	t.cpu(s)
1	189,0	6,9	0,5	189,0	6,9	0,2	196,8	2,7	0,3	202,0	0,0	0,5
2	333,0	0,0	5,3	333,0	0,0	1,9	333,0	0,0	10,2	333,0	0,0	11,4
3	414,0	2,2	1,7	414,0	2,2	1,2	419,3	0,9	8,8	423,0	0,0	4,8
4	643,0	1,6	7,8	639,0	2,2	4,2	643,0	1,6	110,6	652,0	0,2	91,2
5	756,0	1,3	47,0	756,0	1,3	11,9	756,0	1,3	93,9	762,0	0,5	179,4
6	738,0	3,0	53,0	738,0	3,0	16,8	738,0	3,0	119,8	756,0	0,5	231,1
7	999,0	3,0	160,0	999,0	3,0	43,3	999,0	3,0	321,5	1.017,0	1,2	2.678,3

Tabela 5.3: Resultados da formulação estendida com geração de colunas e cortes

Capítulo 6

Métodos Híbridos para o PPTC

A modelagem do PPTC como um problema de programação linear inteira permite que através da utilização de um programa resolvidor de PLIM se navegue de forma inteligente através do espaço de busca: resolvidores modernos implementam técnicas de geração automática de cortes, pré-processamento e estratégias de direcionamento através da árvore de busca que aceleram o processo de resolução. Mesmo assim, a prova da otimalidade para problemas como o PPTC pode não ser possível em tempos computacionais factíveis para muitas instâncias reais. Em muitos casos, como o autor verificou em experimentos preliminares, mesmo a obtenção de soluções factíveis para o PPTC pode ser extremamente custosa se a única ferramenta utilizada for PLIM. De qualquer modo, métodos de resolução que produzam soluções boas no início do processo de busca são preferíveis à métodos que apresentam soluções boas apenas nos últimos estágios do processo, com tempos computacionais potencialmente inactíveis. Utilizando essa premissa foi proposto o método de *Local Branching* [10], com o objetivo de melhorar o desempenho heurístico¹ dos resolvidores PLIM. O componente fundamental dessa técnica é o corte *local branching*, que define o sub-espaço onde irá se procurar melhorar uma solução de referência.

Dada uma solução de referência \bar{x} , com variáveis binárias, seja \bar{S} como o conjunto dos índices das variáveis x ativas ($\bar{S} = \{j : x_j = 1\}$) o corte *local branching* para o caso de quando $|\bar{S}|$ é constante para qualquer solução \bar{x} pode ser definido como:

$$\sum_{j \in \bar{S}} (1 - x_j) \leq k \tag{6.1}$$

¹Neste capítulo palavra desempenho será relacionada com o comportamento heurístico dos resolvidores PLIM, isto é, não irá se considerar o tempo de prova da otimalidade, mas sim a qualidade das soluções produzidas considerando-se limites de tempo reduzidos.

Seja $\mathcal{N}_{lb}(\bar{x}, k)$ como o conjunto de soluções factíveis que podem ser alcançadas através da modificação de no máximo k variáveis ativas na solução \bar{x} , a adição do corte 6.1 ao modelo define que a busca será realizada em $\mathcal{N}_{lb}(\bar{x}, k)$, que corresponde a idéia clássica de vizinhança k -optimal. Fischetti e Lodi [10] apresentam a desigualdade 6.1 como uma maneira de diminuir o espaço de busca através de uma *fixação fraca*, em contraposição à *fixação forte*, que consistiria em fixar o valor de algumas variáveis binárias em 1.

A idéia é que mesmo que o resolvidor não consiga realizar a busca em todo o espaço de soluções, a busca em $\mathcal{N}_{lb}(\bar{x}, k)$ pode ser realizada de modo eficiente para valores de k em que a busca exaustiva seria inviável. Desse modo, a definição do parâmetro k é ponto crítico no método: k deve ser suficientemente pequeno para que a vizinhança seja explorada de forma eficiente e suficientemente grande para permitir que soluções melhores sejam alcançadas.

Outra abordagem para a exploração de sub-espacos também foi considerada em [11], para a melhora do limite primal de uma instância do Problema de Alocação Generalizada. Nesse caso, ao invés da utilização de uma solução de referência como em [10] utilizam-se duas soluções de referência. Pretende-se desse modo investigar uma vizinhança elipsoidal ao redor de duas soluções guia, ao invés da pesquisa na vizinhança em esfera de [10]. Sejam \bar{x}_1 e \bar{x}_2 duas soluções de referência suficientemente distintas e \bar{S}_1 e \bar{S}_2 o conjunto dos índices das variáveis ativas em \bar{x}_1 e \bar{x}_2 , o **corte elipsoidal** é definido como:

$$\sum_{j \in (\bar{S}_1 \cap \bar{S}_2)} 2x_j + \sum_{j \in \bar{S}_1 \setminus \bar{S}_2} x_j + \sum_{j \in \bar{S}_2 \setminus \bar{S}_1} x_j \geq \alpha(\bar{x}_1, \bar{x}_2) - k \quad (6.2)$$

sendo que $\alpha(\bar{x}_1, \bar{x}_2)$ é o valor obtido através da utilização de \bar{x}_1 ou \bar{x}_2 no lado esquerdo de 6.2:

$$\alpha(\bar{x}_1, \bar{x}_2) = 2 \cdot |\bar{S}_1 \cap \bar{S}_2| + |\bar{S}_2| - |\bar{S}_1 \cap \bar{S}_2| \quad (6.3)$$

Já o parâmetro k indica a folga para a definição da vizinhança elipsoidal. Define-se como $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$ a vizinhança contendo as soluções factíveis que também satisfazem 6.2. A idéia básica na definição 6.2 é de que atributos comuns em ambas as soluções são mais fortemente fixados do que atributos que não são consenso.

Claramente, essa definição de vizinhança apresenta um paralelo com a técnica de Reconexão de Caminhos (RC) [62, 63], no sentido em que soluções combinando componentes de duas soluções de referência são exploradas. Uma execução de exemplo de uma busca

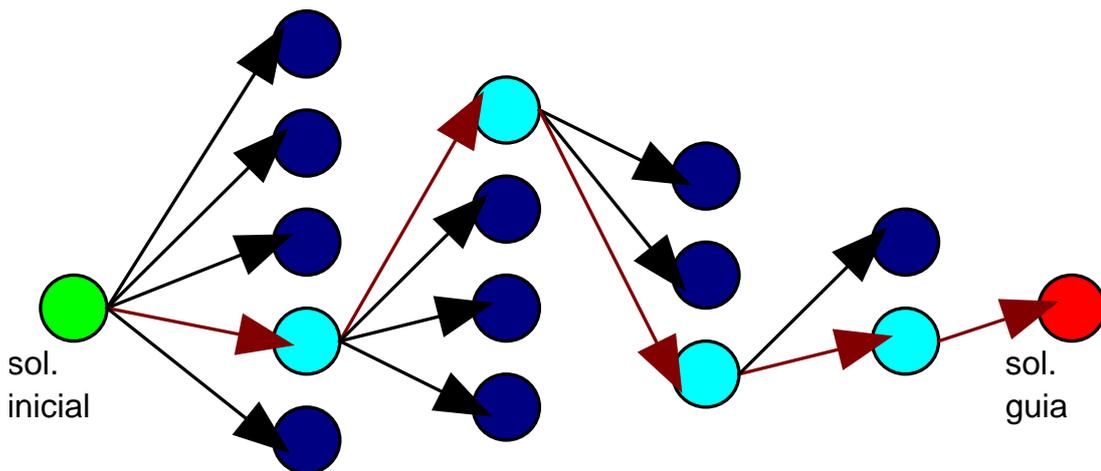


Figura 6.1: Exemplo de execução do método de reconexão de caminhos, um caminho entre uma solução base e uma solução guia é explorado.

com RC é ilustrada na Figura 6.1. Nesse método, inicialmente é computada a diferença simétrica entre duas soluções e, passo-a-passo, realiza-se a incorporação gulosa de um atributo de uma solução guia em uma solução corrente. Ao final do processo retorna-se a melhor solução intermediária visitada. O método de RC tem se mostrado eficiente como estratégia de intensificação considerando sua utilização combinada com metaheurísticas que mantêm um conjunto elite de soluções. Maiores detalhes sobre essa técnica podem ser obtidos em [64].

A relação entre a vizinhança elipsoidal e o método de RC pode ser analisada do seguinte modo: primeiramente, considerando que $\mathcal{RC}(\bar{x}_1, \bar{x}_2)$ seja a vizinhança de RC como descrita no parágrafo anterior e \bar{x}_1 e \bar{x}_2 sejam duas soluções factíveis, pode-se observar que $\mathcal{RC}(\bar{x}_1, \bar{x}_2) \subset \mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, 0)$, ou seja, a melhor solução existente em $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, 0)$ é melhor ou igual à melhor solução em $\mathcal{RC}(\bar{x}_1, \bar{x}_2)$. A utilização de valores não nulos para k em $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$ implica que $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$ contém todas as soluções vizinhas das soluções intermediárias (que consistem em qualquer combinação de atributos de \bar{x}_1 e \bar{x}_2), com uma vizinhança de tamanho k . Um exemplo de uma busca em uma vizinhança $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, 0)$ é apresentado na Figura 6.2.

Em [10], os autores do método de *Local Branching* afirmam que idéias da área de metaheurísticas, especificamente os métodos de Busca Tabu e Busca em Vizinhança Variável (VNS) [65], podem contribuir significativamente para a melhora do desempenho dos resolvidores de PLIM, algo fundamental para a utilização de PLIM em aplicações práticas. Parte desse potencial foi explorado por Hansen e seus colegas em [66]. Nesse trabalho propõe-se uma heurística genérica para PLIM baseada em VNS, com estratégias de intensificação e diversificação. Os resultados forneceram melhores limites primais para

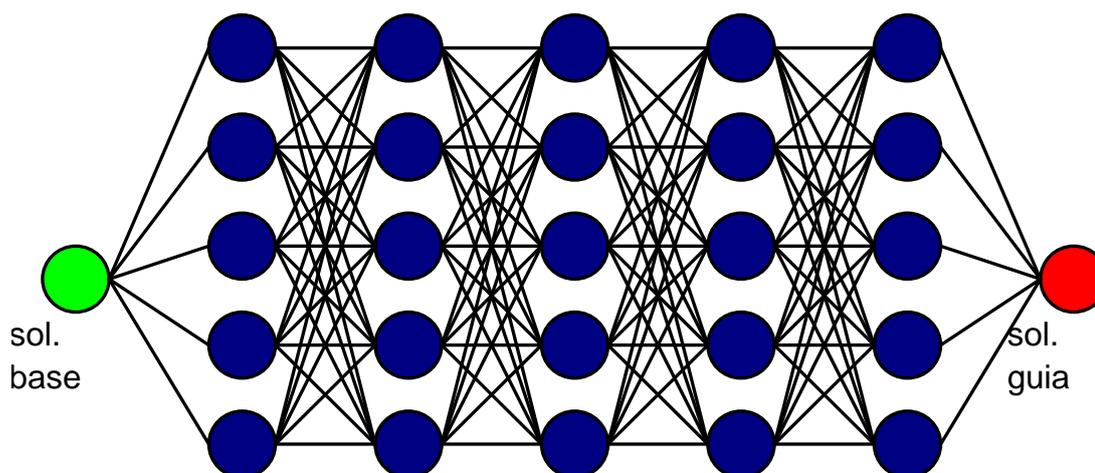


Figura 6.2: Exemplo de execução da busca na vizinhança elipsoidal com $k = 0$, todas as soluções com qualquer combinação de atributos das soluções base e guia são consideradas.

14 de 29 instâncias da literatura.

Extensões ao método de *Local Branching* também foram propostas por Fischetti e seus colegas [67], com o objetivo de melhorar o desempenho da técnica em problemas com dois níveis de variáveis, comuns na área de projeto de redes de telecomunicações. Nesses problemas a ativação de alguns subconjuntos de variáveis é condicionada a ativação de outras variáveis, através da ocorrência dos coeficientes *big-M* em restrições específicas. No caso do PPTC as variáveis v_{pd} seriam as variáveis do primeiro nível, enquanto as variáveis x_{ptdh} corresponderiam às variáveis do segundo nível.

Neste capítulo o uso de resolvidores de PLIM genéricos é considerado para a busca em vizinhança de soluções para o PPTC, considerando propostas dos trabalhos anteriormente citados. Ao invés de reportar resultados de um método híbrido, que incorporasse a heurística BT-PPTC-TR e a busca em vizinhança usando PLIM, por exemplo, optou-se por exibir experimentos mais detalhados considerando em separado os resultados da busca local usando PLIM, em diferentes vizinhanças e com diferentes configurações de parâmetros e modos de execução. Acredita-se que dessa forma será mais fácil a compreensão do comportamento heurístico dos resolvidores PLIM na melhora de soluções do PPTC e a definição de algumas tendências que foram observadas. Ademais, pode-se considerar que de um modo simples a integração dos métodos citados é trivial, considerando-se a busca em uma grande vizinhança utilizando PLIM como um processo de pós-otimização do método BT-PPTC-TR, por exemplo.

Experimentos preliminares realizados indicaram algumas questões importantes que deviam ser consideradas para a implementação eficiente de busca em vizinhança com

PLIM:

1. Que formulação utilizar ?

forte - melhor limite dual, menor árvore de busca mas processamento de nós mais lento;

fraca - pior limite dual, maior árvore de busca mas processamento de nós mais rápido.

2. Quais valores de k são mais apropriados para a busca em $\mathcal{N}_{lb}(\bar{x}, k)$ e $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$, considerando soluções produzidas por BT-PPTC-TR ?
3. Ao realizar uma busca em vizinhança com PLIM, no momento em que uma nova melhor solução \bar{x}' é encontrada, deve-se reiniciar a busca com a definição de uma nova vizinhança ao redor de \bar{x}' ou pesquisar exaustivamente a vizinhança corrente ?

Os experimentos apresentados na seção seguinte não pretendem oferecer respostas definitivas para essas questões, mas sim indicar algumas tendências observadas durante a realização desta pesquisa.

6.1 Experimentos computacionais

Experimentos computacionais foram realizados para a avaliação do potencial heurístico das idéias apresentadas neste capítulo. Em especial, considerando o PPTC, pretende-se verificar o quão eficientemente a busca nas vizinhanças definidas pelos cortes de *local branching* e pelos cortes elipsoidais pode ser realizada para diferentes valores de k utilizando-se um resolvidor de PLIM. Para os experimentos com buscas em $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$ e $\mathcal{N}_{lb}(\bar{x}, k)$ foram consideradas soluções geradas por BT-PPTC-TR com alguma distância da solução ótima, de modo que sempre fosse possível a melhora das soluções. Na Tabela 6.1 são apresentados os custos das 5 soluções de partida consideradas para cada instância. As soluções foram selecionadas de modo que não existam duas soluções muito similares (cada solução deve ter no mínimo 5% das alocações em períodos distintos). O ambiente computacional e as ferramentas utilizadas foram as mesmas descritas no capítulo anterior.

Primeiramente foi realizado um experimento para avaliar o alcance das vizinhanças $\mathcal{N}_{lb}(\bar{x}, k)$ e $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$, do seguinte modo: dado um conjunto de soluções factíveis não ótimas, que valores de k são necessários para que $\mathcal{N}_{lb}(\bar{x}, k)$ e $\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$ contenham soluções melhores. Na Tabela 6.2 são exibidos os resultados, sendo que t.cpu(s) indica o

instância	pior	médio	melhor
1	205	204,0	203
2	353	349,2	344
3	439	437,2	435
4	683	680,6	677
5	796	787,8	780
6	790	787,0	783
7	1072	1063,4	1055

Tabela 6.1: Custos das soluções de partida para cada instância

k	$\mathcal{N}_{lb}(\bar{x}, k)$		$\mathcal{N}_{eps}(\bar{x}_1, \bar{x}_2, k)$	
	t.cpu(s)	sucesso(%)	t.cpu(s)	sucesso(%)
1	7,8	0,0	47,6	14,3
2	84,5	14,3	182,7	57,1
3	191,1	28,6	488,1	80,0
4	279,0	85,3	870,5	94,1

Tabela 6.2: Comparação da otimização na vizinhança definida pelos cortes de *local branching* e pelo corte elipsoidal

tempo médio de CPU. Esse tempo corresponde ao tempo para que uma solução melhor seja alcançada, ou o tempo para provar que a vizinhança em questão não contém uma solução melhor. Como esperado, os mesmos valores de k definem vizinhanças com diferentes alcances dependendo do corte utilizado. Um resultado interessante é que para a obtenção do mesmo sucesso na tarefa de melhorar soluções, no caso 14,3%, a busca na vizinhança elipsoidal consumiu consideravelmente menos tempo. Considerando outros valores de sucesso a comparação não é tão simples. De qualquer maneira, esse resultado parece indicar que o desempenho heurístico da busca na vizinhança com cortes elipsoidais é melhor do que o desempenho obtido na busca na vizinhança dos cortes de *local branching*.

Considerando a pergunta 1 da seção anterior, experimentos foram realizados com duas variações da formulação \mathcal{F}_1 : a mais fraca ($\mathcal{F}_{1,0}$) e a mais forte ($\mathcal{F}_{1,4}$). Foram executadas buscas em $\mathcal{N}_{lb}(\bar{x}, 12)$ para as instâncias da Tabela 6.1. Estipulou-se um limite de tempo de 10 minutos para que uma solução melhor seja encontrada. A cada solução melhor encontrada a solução de referência \bar{x} era atualizada e o contador de tempo reinicializado. Na Figura 6.3 é apresentado um gráfico que considera o número de melhoras na solução de referência que foi alcançado com cada uma das formulações. Para esse experimento, o resolvidor de PLIM CPLEX foi configurado com a ênfase 4, que prioriza a obtenção de soluções factíveis. Como pode-se observar, a formulação $\mathcal{F}_{1,4}$, apesar de ser mais pesada (maior número de linhas), apresenta um desempenho heurístico superior.

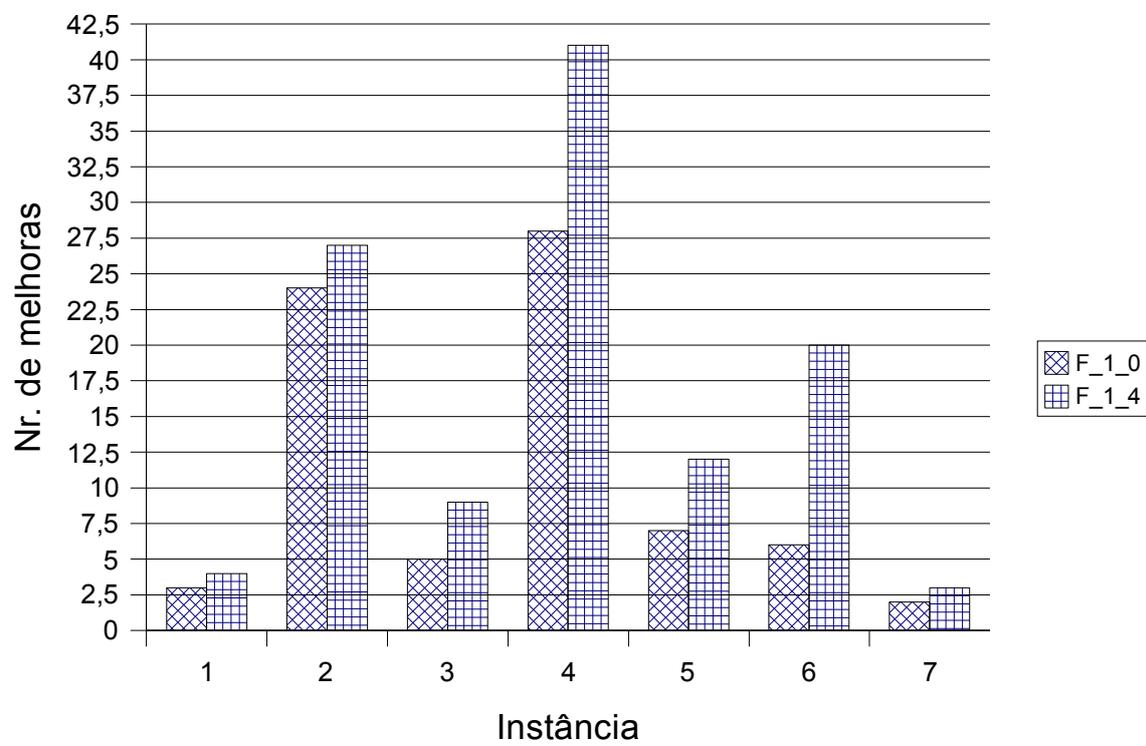


Figura 6.3: Número de melhoras obtidas com a busca na vizinhança *Local Branching*, formulações fraca e forte, execuções com limite de tempo.

Observando o andamento das execuções dos experimentos realizados, um padrão recorrente foi notado: com muita freqüência a melhora de soluções ocorria logo no início da busca, na abertura dos primeiros nós da árvore, ou mesmo durante fase de pré-processamento do resolvedor utilizado. Desse modo, a decisão de redefinir a vizinhança em cada melhora na solução de referência (considerando a pergunta 3 da seção anterior), implica uma freqüente re-execução da fase de pré-processamento do resolvedor de PLIM, uma fase bastante custosa computacionalmente para as instâncias consideradas. Nesse sentido, uma alternativa mais atrativa do ponto de vista heurístico pode ser a redefinição da vizinhança somente depois que a solução de referência for atualizada e um certo número de nós da árvore de busca foi visitado, possibilitando assim duas melhoras consecutivas em um menor espaço de tempo.

Capítulo 7

Conclusões e Trabalhos Futuros

Acreditamos que os resultados apresentados nos capítulos anteriores nos permitem dizer que os objetivos desta tese foram em grande parte alcançados. De modo específico a obtenção da solução ótima para problemas em aberto encontrados na literatura, bem como a definição de melhores limites superiores e inferiores e o desenvolvimento de uma heurística que permite a obtenção rápida de quadros de horário de qualidade foram os resultados do trabalho que gostaríamos de ressaltar.

Acreditamos também que os resultados positivos obtidos com a formulação estendida com geração de cortes e colunas podem ser repetidos em vários problemas relacionados, uma vez que a aplicação deles requer apenas a existência de uma estrutura bastante comum em problemas de programação de horários, que é a divisão das atividades em diferentes intervalos, no caso dias. Nesse sentido, deve-se observar que a melhora dos limites inferiores depende também da função objetivo, a qual no caso do problema considerado é bastante relacionada com a distribuição das atividades nesses intervalos. Uma possibilidade bastante interessante é a utilização da formulação \mathcal{F}'_2 para a busca em grandes vizinhanças, utilizando as técnicas descritas no Capítulo 6. Os experimentos computacionais realizados neste capítulo parecem indicar que resultados melhores podem ser obtidos desse modo, visto que os melhores resultados foram obtidos com a formulação mais forte experimentada, no caso $\mathcal{F}_{1,4}$.

Nesta tese a colaboração entre (i) metaheurísticas e (ii) técnicas baseadas em programação linear foi explorada considerando-se o fornecimento de informações de i para ii, no caso a melhor solução encontrada em i. Como trabalho futuro um caminho promissor é a inversão dos papéis: ii fornecendo informações para i, no espírito da técnica RINS (*Relaxation Induced Neighborhood Search*) [68]. Nesse sentido, soluções da relaxação linear da formulação forte \mathcal{F}_2 podem fornecer valiosas informações para guiar procedimentos

heurísticos de construção ou de melhoramento.

Um tema que certamente pode ser considerado é o tratamento de outras variantes do PPT, com a inclusão de novas restrições e preferências. No entanto, deve-se observar que a tese apresentada de modo algum esgota a possibilidade de melhoramento dos métodos de otimização para o PPTC. A determinação de limites melhores para instâncias do PPTC de maior dimensão permanece um desafio. Ainda, a solução média apresentada pelas heurísticas desenvolvidas considerando um tempo limitado ainda apresenta uma distância considerável da solução ótima ou dos limites inferiores. Nesse sentido, acreditamos que o desenvolvimento de novos métodos em trabalhos futuros pode ser beneficiado pela avaliação comparativa com os métodos que já foram desenvolvidos e avaliados computacionalmente neste trabalho e em outros. Nesse sentido será disponibilizado um banco de instâncias para o PPTC na Internet com detalhadas informações dos resultados exatos e heurísticos obtidos para essas instâncias até o momento.

Considerando o melhoramento da heurística desenvolvida, uma possibilidade interessante é a incorporação de mecanismos especializados de busca em vizinhanças grandes, levando em conta as propostas de Souza [8].

Em se tratando da formulação estendida proposta caminhos promissores parecem ser o estudo de novos mais cortes que considerem a estrutura do problema bem como o estudo de técnicas para aceleração da geração de colunas. Nesse sentido, técnicas de estabilização devem ser consideradas, bem como maneiras mais rápidas de inserção de colunas no início do processo, possivelmente através da utilização de heurísticas nas primeiras iterações.

APÊNDICE A - Publicações

Neste apêndice são listados os trabalhos publicados em periódicos ou apresentados em eventos científicos desenvolvidos durante o período de realização desta pesquisa.

Publicações em Periódicos

Título: A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem

Co-autores: Luiz S. Ochi e Marcone J.F. Souza

Periódico: Journal of Experimental Algorithmics

Editora: ACM Press

ISSN: 1084-6654

Volume: 10 **Ano:** 2005

Título: Combining an Evolutionary Algorithm with Data Mining to solve a Vehicle Routing Problem

Co-autores: Luiz S. Ochi, Euler H. Marinho e Lúcia M.A. Drummond

Periódico: Neurocomputing

Editora: Elsevier

ISSN: 0925-2312

Volume: 70 **Ano:** 2006

Páginas: 70-77

Trabalhos Apresentados em Eventos Internacionais

Título: Combining Metaheuristics and Integer Programming on School Timetabling Problems

Co-autores: Eduardo Uchoa e Luiz S. Ochi

Evento: Matheuristics 2006: 1st Workshop on Mathematical Contributions to Metaheuristics

Local: Bertinoro (Forlì - Cesena), Italy

Período: 27-30 de Agosto de 2006

Título: An Efficient Tabu Search Heuristic for the School Timetabling Problem

Co-autores: Luiz S. Ochi e Marcone J.F. Souza

Evento: III Workshop on Efficient and Experimental Algorithms

Local: Angra dos Reis, Rio de Janeiro

Período: 25 a 28 de maio, 2004

ISSN: 0302-9743

Editora: Springer Berlin / Heidelberg

Série: Lecture Notes in Computer Science

Volume: 3059 **Páginas:** 468-481

Ano: 2004

Trabalhos Apresentados em Eventos Nacionais

Título: GRASP with Data Mining for The School Timetabling Problem

Co-autores: Luiz Satoru Ochi, Lúcia M.A. Drummond e Marcone M.J.F. Souza

Evento: Two Days on Combinatorial Optimization: A Bridge Between Rio and Niterói

Local: Rio de Janeiro

Período: 11 e 12 de Dezembro de 2003

-
- Título:** Metaheurística Busca Tabu para o Problema de Coloração de Grafos
- Co-autores:** Felipe N. Aguiar (primeiro autor), Gustavo S. C. Honorato e Luiz Satoru Ochi
- Evento:** XXXVII Simpósio Brasileiro de Pesquisa Operacional (XXXVII SBPO)
- Local:** Gramado, RS
- Período:** 27 a 30 de setembro de 2005
- Páginas:** 2497-2504 (em CD-ROM)
-

APÊNDICE B - Quadros de Horários Ótimos e Sub-ótimos

Neste apêndice são apresentadas algumas das melhores soluções obtidas para as sete instâncias do PPTC consideradas neste trabalho. Três dessas soluções são provadamente ótimas. Nas Figuras [B.1](#), [B.2](#), [B.3](#) e [B.4](#) cada linha representa o horário semanal completo de um professor. Em cada célula é indicada turma que o professor está lecionando em um determinado período. Células vazias indicam inatividade e células com “×” indicam períodos de indisponibilidade do professor.

Professor\Período	Dia 1					Dia 2					Dia 3					Dia 4					Dia 5					
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
1	2	2		3	1	1	1	1	3	2	×	×	×	×	×											
2		1	1	2	2						2	2	1						2	1	1	×	×	×	×	×
3	3		2	1	3											×	×	×	×	×	1	2	3	2	1	
4						×	×	×	×	×			3	1	2	1	2	3						2	1	3
5	×	×	×	×	×	3	3	2	2							3	3		2	2	2	2	3			
6						2	2	1	1	3	×	×	×	×	×	2	1	1	3	3				3	2	
7	1	3	3			×	×	×	×	×	1	1		3	3						3	1	1			
8										1	3	3	2	2	1	×	×	×	×	×						

Figura B.1: Solução ótima - instância 1 - avaliação: 202

Professor\Período	Dia 1					Dia 2					Dia 3					Dia 4					Dia 5									
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
	1						3	3	1	2	2						3	2	3	1	1	1								
2	6	6	4	4	5					6						5	5	4	2	6	×	×	×	×	×					
3			6	5	4																									
4	3	3				2	2	3	1	4						2	1	1	4	4	×	×	×	×	×					
5						6	6	5	5		6	6	5																	
6	4	4	5	2	2						3	2	4	1	1	1	1	6	3	3	6	5	5							
7	1	2	2			1	1	4	4	5	4	3	3	6	6						2	3	6	5	5					
8				1	1						5	5	1	3	3	4	4	2	6	5	3	2	2	4	6					
9																														
10	2	1	1	3	3																									
11	5	5	3	6	6									2	4						4	1	3	2	1					
12						4	5	6	6	3	1	4	2	5	2															
13						5	4	2	3	1	2	1	6	4	5	6	3													
14	×	×	×	×	×						×	×	×	×	×	×	×	×	×	×	5	4	1	3	2					

Figura B.2: Solução ótima - instância 2 - avaliação: 303

Professor\Período	Dia 1					Dia 2					Dia 3					Dia 4					Dia 5				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1	3	2	3	2	3	2	3	1	1	×	×	×	×	×	3	1	2	2	1					
2	2	1	3	2	3	×	×	×	×	×						1	2	3	1	3	2	1	2	1	3
3									2	2	1	1	2	3	3	×	×	×	×	×	1	3	1	3	2
4	3	2	1								3	1	2	1	2	×	×	×	×	×					
5	×	×	×	×	×	2	1	7	3	8	3	2	1	7	6						3	2	6	8	1
6	4	8	7	1	5	1	4	6			5	8	7	6	1						×	×	×	×	×
7						6	5	1	7	3	2	4	8			×	×	×	×	×					
8						1	8	3	2	4	5	×	×	×	×	6	7	1	3	2					
9	×	×	×	×	×						4	3				2	3				5	4	3	2	
10						×	×	×	×	×						4	4	5	4	5	4	5			4
11	5	4	5	4							×	×	×	×	×										4
12						7	8	5	6	4	7	6	5	8	4	×	×	×	×	×	7	6	4	5	8
13	7	6	6	8	7						×	×	×	×	×	7	6	8	6	8	6	8	8	7	7
14	8	7	8	6	6	×	×	×	×	×						8	8	7	7	6	8	7	7	6	6
15	6	5	4	7	8	4	6	8	5	7	8	5	6	4	7	×	×	×	×	×					
16						5	7	4	8	6	6	7	4	5	8	4	5	6	8	7	×	×	×	×	×

Figura B.3: Solução ótima - instância 3 - avaliação: 423

Professor \ Período	Dia 1					Dia 2					Dia 3					Dia 4					Dia 5					
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
	1	×	×	×	×	×	6	6	7	8	8			7	7	2	1	6	6	8	8	1	1	2	2	7
2	10	11	9	9	9	12	12	3	3	5	11	11	5	12	12	2	10	10	5	5	10	3	3	9	9	
3	×	×	×	×	×	4					×	×	×	×	×	4	4		×	×	×	×	×	×	×	
4				1	4																×	×	×	×	×	
5	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×						5	8	7	6	6	
6	×	×	×	×	×	9	5	11	12	6			12	1	7	11	7	8	10	10	5	8	9	6	1	
7	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	3	2	2	4	4	3	
8				3	2	8	3	4	8	1	10	12	9	10	4	11	9	2	1	12	11	×	×	×	×	×
9	6	5	6	5	7																7					
10	1	4	5	3	1						4	2	2	3	5						×	×	×	×	×	
11	8	6	7	12	11						6	12	9	9	10	10	8	7	11		×	×	×	×	×	
12	3	1	1	6	6	7	7	4	4	2	5	5	1	6	6	3	3		2	2	4	7	7	5	5	
13	×	×	×	×	×	8	8	9	11	11	10	10	8	8	9	12	12	11	9	9	12	12	11	10	10	
14	2	12	4	11	12						1	1	3	2	3			4	3	1	11	4	12	11	2	
15	7	10	9	8	5	5	9	10	10	7						6	9	5	6	7			6	8	8	
16	5	2	×	×	×	2	1	2	6	3	3	4	4	×	×	5	1	3	×	×	6	6	5	1	4	
17	9	8	12	7	10						9	7	11	10	8	8	11	12	7							
18	4	7	2	4	3	1	3	5	7	1	2	3	6	5	1	×	×	2	4	6	×	×	×	×	×	
19	×	×	×	×	×	×	×	×	×	×	8	8		×	×	×	×	×	×	×	×	×	×	×	×	
20	×	×	×	×	×	11	11	12	9	12	×	×	×			×	×	×	×		9	10	10			
21	11	9	10	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	12	×	×	×	×	×	
22	×	×	×	×	×						×	×	×	×	×	×	×	×	×	×	3	9		12	11	
23	12	3	8	10	2	10	2	6	5	9	7	6		11	4	7	5	9	1	4	8	11	1	3	12	

Figura B.4: Solução sub-ótima - instância 4 - avaliação: 653 - limite inferior: 652

Referências

- [1] GOTLIEB, C. The construction of class-teacher timetables. In: POPPLEWELL, C. M. (Ed.). *Proceeding IFIP congress 1962*. Amnsterdam: [s.n.], 1963. p. 73–77.
- [2] CSIMA, J.; GOTLIEB, C. Tests on a computer method for constructing school timetables. *Communications of the ACM*, v. 7, n. 3, p. 160–163, 1964.
- [3] BURKE, E.; ROSS, P. (Ed.). *Practice and Theory of Automated Timetabling*, (Lecture Notes in Computer Science). [S.l.]: Springer, 1996.
- [4] BURKE, E.; CARTER, M. (Ed.). *Practice and Theory of Automated Timetabling II*, (Lecture Notes in Computer Science). [S.l.]: Springer, 1997.
- [5] BURKE, E.; ERBEN, W. (Ed.). *Practice and Theory of Automated Timetabling III*, (Lecture Notes in Computer Science). [S.l.]: Springer, 2000.
- [6] SOUZA, M. *Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas*. Tese (Doutorado) — Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [7] SOUZA, M.; MACULAN, N.; OCHI, L. Melhorando quadros de horários de escolas através de caminhos mínimos. *Tendências em Matemática Aplicada e Computacional*, v. 1, n. 1, p. 515–524, 2000.
- [8] SOUZA, M.; OCHI, L.; MACULAN, N. A GRASP-tabu search algorithm for solving school timetabling problems. In: _____. *Metaheuristics: Computer Decision-Making*. [S.l.]: Kluwer Academic Publishers, 2003. p. 659–672.
- [9] MATA, S. *O Problema do Horário na Escola do Segundo Grau: Modelagem e Implementação*. Dissertação (Mestrado) — Progama de Engenharia de Sistemas e Computação, COPPE, UFRJ, Rio de Janeiro, Brasil, 1989.
- [10] FISCHETTI, M.; LODI, A. Local branching. *Mathematical Programming*, v. 98, p. 23–47, 2003.
- [11] PIGATTI, A.; ARAGÃO, M. Poggi de; UCHOA, E. Stabilized branch-and-cut-and-price for the generalized assignment problem. In: *2nd Brazilian Symposium on Graphs, Algorithms and Combinatorics, 2005, Angra dos Reis*. Amsterdam, Holanda: Elsevier Science, 2005. v. 19, p. 385–395.
- [12] SCHAERF, A. A survey of automated timetabling. *Artificial Intelligence Review*, v. 13, n. 2, p. 87–127, 1999.
- [13] WERRA, D. An introduction to timetabling. *European Journal or Operational Research*, v. 19, p. 151–162, 1985.

- [14] WERRA, D. The combinatorics of timetabling. *European Journal of Operational Research*, Elsevier, v. 96, n. 3, p. 504–513, 1997.
- [15] EVEN, S.; ITAI, A.; SHAMIR, A. On the complexity of timetable and multicommodity flow problems. *SIAM Journal of Computing*, v. 5, n. 4, p. 691–703, dez. 1976.
- [16] COOPER, T.; KINGSTON, J. H. The complexity of timetable construction problems. In: _____. *The Practice and Theory of Automated Timetabling (selected papers from Proceedings of the First International Conference, Practice and Theory of Automated Timetabling, Edinburgh, 1995)*. London, UK: Springer-Verlag, 1996. (Springer Lecture Notes in Computer Science, v. 1153), p. 283–295.
- [17] COSTA, D. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, v. 76, p. 98–110, 1994.
- [18] SCHAERF, A. Local search techniques for large high school timetabling problems. *IEEE Transactions on Systems, Man and Cybernetics Part A: systems and Humans*, v. 29, n. 4, p. 368–377, jul. 1999.
- [19] ALVAREZ-VALDÉS, R.; MARTIN, G.; TAMARIT, M. Constructing good solutions for the spanish school timetabling problem. *Journal of the Operational Research Society*, v. 47, p. 1203–1215, 1996.
- [20] COLORNI, A.; DORIGO, M.; MANIEZZO, V. Metaheuristics for high-school timetabling. *Computational Optimization and Applications*, v. 9, n. 3, p. 277–298, 1998.
- [21] CALDEIRA, J.; AGOSTINHO, C. School timetabling using genetic search. In: *Practice and Theory of Automated Timetabling (1997)*. Toronto: [s.n.], 1997. p. 115–122.
- [22] STEFANO, C. D.; TETTAMANZI, A. An evolutionary algorithm for solving the school time-tabling problem. In: AL., E. B. et (Ed.). *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*. London, UK: Springer-Verlag, 2001. v. 2037, p. 452–462.
- [23] HERTZ, A. Tabu search for large scale timetabling problems. *European Journal of Operational Research*, v. 54, p. 39–47, 1991.
- [24] AVELLA, P.; VASIL'EV, I. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, v. 8, p. 497–514, 2005.
- [25] DASKALAKI, S. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, v. 153, p. 117–135, 2004.
- [26] PAPOUTSIS, K.; VALOUXIS, C.; HOUSOS, E. A column generation approach for the timetabling problem of greek high schools. *Journal of the Operational Research Society*, v. 54, p. 230–238, 2003.
- [27] DREXL, A.; SALEWSKI, F. Distribution requirements and compactness constraints in school timetabling. *European Journal of Operational Research*, v. 102, n. 1, p. 193–214, out. 1997.
- [28] BIRBAS, T.; DASKALAKI, S.; HOUSOS, E. Timetabling for greek high schools. *Journal of the Operational Research Society*, v. 48, p. 1191–1200, 1997.

- [29] EISELT, H. A.; LAPORTE, G. Combinatorial optimization problems with soft and hard requirements. *Journal of Operations Research*, v. 38, n. 9, p. 785–795, 1987.
- [30] SANTOS, H. G.; OCHI, L.; SOUZA, M. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *Journal of Experimental Algorithmics (JEA)*, ACM Press, New York, NY, USA, v. 10, 2005. ISSN 1084-6654.
- [31] FERLAND, J.; LAVOUE, A. Exchange procedures for timetabling problems. *Discrete Applied Mathematics*, p. 237–253, 1992.
- [32] GLOVER, F. Future paths for integer programming and artificial intelligence. *Computers & Operations Research*, v. 13, p. 533–549, 1986.
- [33] HANSEN, P. The steepest ascent mildest descent heuristic for combinatorial programming. In: *Congress on Numerical Methods in Combinatorial Optimization*. Capri, Italy: [s.n.], 1986.
- [34] HERTZ, A.; WERRA, D. Using tabu search techniques for graph coloring. *Computing*, v. 39, n. 4, p. 345–351, dez. 1987.
- [35] GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. *Management Science*, v. 40, n. 10, p. 1276–1290, out. 1994.
- [36] RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures. In: *Handbook of Metaheuristics*. [S.l.]: Kluwer, 2003. p. 219–249.
- [37] TORREÃO, J. R. A.; ROE, E. Microcanonical optimization applied to visual processing. *Physics Letters*, v. 205, p. 377–382, fev. 1995.
- [38] KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, p. 671–680, 1983.
- [39] ABRAMSON, D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, v. 37, p. 98–113, 1991.
- [40] CUNG, V. et al. Strategies for the parallel implementation of metaheuristics. In: _____. *Essays and Surveys in Metaheuristics*. [S.l.]: Kluwer Academic Publishers, 2002. p. 263–308.
- [41] ABRAMSON, D.; ABELA, J. A parallel genetic algorithm for solving the school timetabling problem. In: *Proceedings of the 15th Australian Computer Science Conference*. [S.l.: s.n.], 1992.
- [42] LAWRIE, N. An integer linear programming model of a school timetabling problem. *Computer Journal*, v. 12, p. 307–316, 1969.
- [43] DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. *Operations Research*, v. 8, n. 1, p. 101–111, 1960.
- [44] GANS, O. D. A computer timetabling system for secondary schools in the netherlands. *European Journal of Operational Research*, v. 7, p. 175–182, 1981.
- [45] BRESINA, J. Heuristic-biased stochastic sampling. In: *AAAI/IAAI, Vol. 1*. [S.l.: s.n.], 1996. p. 271–278.

- [46] GLOVER, F.; LAGUNA, M. *Tabu Search*. Boston: Kluwer, 1997.
- [47] GRIFFITH, A. *GCC: The Complete Reference*. [S.l.]: McGraw-Hill, 2002.
- [48] AIEX, R. M.; RESENDE, M. G. C.; RIBEIRO, C. C. Probability distribution of solution time in GRASP: an experimental investigation. *Journal of Heuristics*, v. 8, p. 343–373, 2002.
- [49] T., L. J.; RALPHS, T. K. Noncommercial software for mixed-integer linear programming. In: KARLOF, J. (Ed.). *Integer Programming: Theory and Practice*. [S.l.: s.n.], 2005, (Operations Research Series, v. 3).
- [50] ATAMTÜRK, A.; SAVELSBERGH, M. Integer-programming software systems. *Annals of Operations Research*, v. 140, p. 67–124, 2005.
- [51] WOLSEY, L. *Integer Programming*. [S.l.]: Wiley, 1998. (Wiley-Interscience series in discrete mathematics and optimization).
- [52] BARNHART, C. et al. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, v. 46, p. 316–329, 1998. Disponível em: <citeseer.ist.psu.edu/barnhart96branchprice.html>.
- [53] aO, M. Poggi de A.; UCHOA, E. Integer program reformulation for robust branch-and-cut-and-price. In: *Annals of Mathematical Programming in Rio*. Buzios, Brazil: [s.n.], 2003. p. 56–61.
- [54] BALAS, E. Facets of the knapsack polytope. *Mathematical Programming*, Springer Berlin, v. 8, n. 1, p. 146–164, 1975.
- [55] HAMMER, P.; JOHNSON, E.; PELED, U. Facets of regular 0-1 polytopes. *Mathematical Programming*, Springer Berlin, v. 8, n. 1, p. 179–206, 1975.
- [56] WOLSEY, L. Faces for a linear inequality in 0-1 variables. *Mathematical Programming*, v. 8, p. 165–178, 1975.
- [57] ZONGHAO, G.; NEMHAUSER, G.; SAVELSBERGH, M. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, v. 10, n. 4, p. 427–437, 1998.
- [58] BOYD, E. Fenchel cutting planes for integer programming. *Operations Research*, v. 42, p. 53–64, 1992.
- [59] BOYD, E. Solving 0/1 integer programs with enumeration cutting planes. *Annals of Operations Research*, v. 50, p. 61–72, 1994.
- [60] ILOG S.A. *ILOG CPLEX 10.0 Callable Library*. [S.l.], 2006.
- [61] ILOG S.A. *ILOG CPLEX 10.0 User's Manual*. [S.l.], 2006.
- [62] GLOVER, F. Tabu search and adaptive memory programming - advances, applications and challenges. In: BARR, R.; HELGASON, R.; KENNINGTON, J. (Ed.). *Advances in Metaheuristics, Optimization and Stochastic Modeling Technologies*. Boston, M.A.: Kluwer, 1997. p. 1–75.

-
- [63] GLOVER, F.; LAGUNA, M.; MARTÍ, R. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, v. 39, n. 3, p. 653–684, 2000.
- [64] RESENDE, M.; RIBEIRO, C. GRASP with path-relinking: Recent advances and applications. In: IBARAKI, T.; NONOBE K. E YAGIURA, M. (Ed.). *Metaheuristics: Progress as Real Problem Solvers*. [S.l.]: Kluwer Academic Publishers, 2005. p. 29–63.
- [65] HANSEN, P.; MLADENOVIC, N. Variable neighborhood search. *Computers and Operations Research*, v. 24, n. 11, p. 1097–1100, 1997.
- [66] HANSEN, P.; MLADENOVIC, N.; UROSEVIC, D. Variable neighborhood search and local branching. *Comput. Oper. Res.*, Elsevier, Oxford, v. 33, n. 10, p. 3034–3045, 2006. ISSN 0305-0548.
- [67] FISCHETTI, M.; POLO, C.; SCANTAMBURLO, M. A local branching heuristic for mixed-integer programs with 2-level variables. *Networks*, v. 44, n. 2, p. 61–72, July 2004.
- [68] DANNA, E.; ROTHBERG, E.; PAPE, C. L. *Exploring relaxation induced neighborhoods to improve MIP solutions*. [S.l.], 2003.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)