

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

FÁBIO VIRIATO PEREZ NEUKIRCHEN

**Um estudo de caso sobre a geração de
quadros de horários nos cursos de Ciência
da Computação e Engenharia da
Computação da UFRGS**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Ciência da
Computação

Orientador: Profa. Dra. Luciana Salete Buriol
Co-orientador: Ártou Pereira Dorneles

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Ciência de Computação: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Whatever the mind of man can conceive and believe,
it can achieve.”*

— NAPOLEON HILL

AGRADECIMENTOS

Agradeço primeiramente a Deus, arquiteto deste Universo, por todas oportunidades concedidas.

Agradeço à minha família, em especial meu filho Miguel Bólivar, minha mãe Maria do Carmo e meus avós maternos Bólivar e Maria, por todo o carinho, afeto e dedicação.

Agradeço à minha namorada Carolina por todo o incentivo e motivação nas horas de necessidade.

Agradeço aos meus colegas de curso pelas experiências e pela convivência durante o período universitário.

Agradeço aos meus amigos pelo apoio quando necessário.

Agradeço aos meus orientadores Luciana Salete Buriol e Ártton Pereira Dorneles por toda ajuda, paciência e disponibilidade.

Agradeço ao coordenador Raul Fernando Weber do curso de Ciência da Computação e ao coordenador Marcelo Götz do curso de Engenharia da Computação pelo apoio fornecido e pela disponibilidade para esclarecimentos.

Agradeço à Universidade Federal do Rio Grande do Sul e ao Instituto de Informática pela infraestrutura e excelente corpo docente.

Agradeço à PETROBRAS e ao PRH PB-217 por toda ajuda e investimento durante meu período como bolsista.

RESUMO

A geração de quadros de horários para universidades é um problema clássico de otimização combinatória que leva em consideração um grande número de variáveis e requisitos. Este tipo de problema, em geral, contém restrições específicas da aplicação, gerando diversas variações do problema. Apesar do problema ser classificado como NP-Difícil, o mesmo é resolvido manualmente na maior parte das instituições, devido à dificuldade de formalizar todas as restrições que cada aplicação requer em um sistema informatizado.

Este trabalho apresenta um estudo da geração de quadros de horários no curso de Ciência da Computação e Engenharia da Computação da Universidade Federal do Rio Grande do Sul onde o problema é formalizado através de um modelo matemático e um estudo experimental é realizado em instâncias reais do problema. Para auxiliar tais experimentos, também foi desenvolvida uma ferramenta web para facilitar a criação de instâncias.

Os resultados obtidos demonstram que as instâncias consideradas podem ser resolvidas de maneira eficiente com o modelo proposto através de um resolvidor comercial de programação inteira mista, em comparação com as soluções manuais.

Palavras-chave: Quadros de horários. Programação inteira. Modelagem Matemática. Otimização combinatória.

A case of study on timetabling generation at the Courses of Computer Science and Computer Engineering of UFRGS

ABSTRACT

University timetabling is a classic combinatorial optimization problem that considers a large number of variables and requirements. That type of problem in general has specific application constraints generating a large quantity of variations of the problem. Despite the fact the problem belongs to the NP-Hard class of complexity, it is usually manually solved in most institutions, due to the difficulty to formalize all the required constraints in a computerized system.

This work presents a study about timetabling at the course of Computer Science and Computer Engineering at UFRGS where the problem is formalized by a mathematical model and an experimental study is realized using real instances of the problem. In order to assist the experiments, a web based tool was also developed to help the creation of the instances.

The results show that the instances can be efficiently solved with the proposed model using a general purpose mixed integer programming solver, in comparison to the manual solutions.

Keywords: Timetabling. Integer programming. Mathematical Modeling. Combinatorial optimization.

LISTA DE FIGURAS

Figura 2.1 Distribuição de publicações relacionadas à geração de quadros de horários (BARDADYM, 1996)	14
Figura 4.1 Parte da instância no formato YAML	29
Figura 4.2 Parte do modelo matemático representado em GMPL	30
Figura 4.3 Modelo entidade-relacionamento dos dados do INF	31
Figura 4.4 Tela da <i>Interface Gráfica</i> - Visualização da aba Cursos	32
Figura 4.5 Tela da <i>Interface Gráfica</i> - Criação de um Curso	33
Figura 4.6 Tela da <i>Interface Gráfica</i> - Visualização da aba Turmas	34
Figura 4.7 Tela da <i>Interface Gráfica</i> - Criação de uma Turma	34
Figura 4.8 Tela da <i>Interface Gráfica</i> - Visualização da aba Relação entre Turmas	35
Figura 4.9 Tela da <i>Interface Gráfica</i> - Criação de uma relação entre Turmas	36
Figura 4.10 Tela da <i>Interface Gráfica</i> - Visualização da aba Professores	37
Figura 4.11 Tela da <i>Interface Gráfica</i> - Criação de um Professor	37
Figura 4.12 Tela da <i>Interface Gráfica</i> - Visualização da aba Recursos	38
Figura 4.13 Tela da <i>Interface Gráfica</i> - Criação de um Recurso	39
Figura 4.14 Tela da <i>Interface Gráfica</i> -Gerando instância.....	40
Figura 4.15 Fluxo de execução do <i>Resolvedor</i>	41
Figura 4.16 Arquivo de saída do CPLEX	43
Figura 4.17 Exibição dos quadros de horários alocados para etapas	44
Figura 4.18 Exibição dos quadros de horários alocados para professores	45
Figura 4.19 Exibição dos quadros de horários alocados para os recursos laboratório de software e laboratório de hardware.....	46
Figura 4.20 Exibição dos quadros de horários alocados que engloba todas etapas	47
Figura 5.1 Comparação das violações de requisitos fracos entre soluções manuais e soluções obtidas via CPLEX	51

LISTA DE TABELAS

Tabela 2.1 Quadro comparativo de estudos.....	18
Tabela 3.1 Notação utilizada pelo modelo	24
Tabela 5.1 Principais características das instâncias	49
Tabela 5.2 Resultados comparativos entre as soluções manuais e as soluções obtidas pelo CPLEX.....	50

LISTA DE ABREVIATURAS E SIGLAS

UFRGS	Universidade Federal do Rio Grande do Sul
INF	Instituto de Informática
CiC	Ciência da Computação
ECP	Engenharia da Computação
PIM	Programação Inteira Mista
PI	Programação Inteira
PATAT	Practice and Theory of Automated Timetabling
GCC	GNU Compiler Collection
GMPL	GNU Mathematical Programming Language
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Motivação.....	11
1.2 Objetivo do trabalho.....	12
1.3 Organização do texto	12
2 REVISÃO BIBLIOGRÁFICA	14
2.1 Relevância do problema	14
2.2 Quadros de horários universitários.....	15
2.3 Requisitos para geração de quadros de horários universitários.....	15
2.4 Abordagens de resolução.....	16
3 DEFINIÇÃO DO PROBLEMA E MODELAGEM	19
3.1 Descrição do problema	19
3.2 Requisitos do Problema.....	21
3.2.1 Requisitos fortes.....	21
3.2.2 Requisitos fracos.....	21
3.3 Formulação matemática do problema	22
3.3.1 Função Objetivo	22
3.3.2 Restrições	25
4 IMPLEMENTAÇÃO	28
4.1 Formato das Instâncias.....	28
4.2 Modelagem GMPL.....	29
4.3 Interface Gráfica	30
4.3.1 Cursos	31
4.3.2 Turmas	33
4.3.3 Relações entre Turmas	35
4.3.4 Professores	36
4.3.5 Recursos Físicos.....	38
4.3.6 Geração do arquivo de instância	39
4.4 Resolvedor	41
4.4.1 Leitura da instância	42
4.4.2 Criação do arquivo de dados	42
4.4.3 Criação do Problema Linear	42
4.4.4 Solucionando Problema Linear.....	42
4.4.5 Leitura do arquivo com solução final.....	43
5 EXPERIMENTOS COMPUTACIONAIS	48
5.1 Instâncias	48
5.2 Ambiente Computacional.....	49
5.3 Resultados.....	49
6 CONCLUSÕES	52
REFERÊNCIAS	54

1 INTRODUÇÃO

O problema de geração de quadros de horários é um problema combinatório altamente restrito classificado como NP-completo para determinar a existência de uma grade de horários válida (problema de decisão) (COOPER; KINGSTON, 1996), e NP-Difícil para determinar a melhor solução dentre as soluções válidas (problema de otimização) (EIKELDER; WILLEMEN, 2001). O problema de geração de quadros de horários pode ser encontrado em diversas áreas, tais como no ambiente empresarial, educacional, esportes, hospitalar, entre outros (BURKE; KINGSTON; WERRA, 2004).

Os primeiros trabalhos científicos sobre geração de quadros de horários remontam da década de 60 e desde então têm demandado uma crescente atenção (GOTLIEB, 1963). Em seu estudo, (SCHAERF, 1999) classifica os problemas de geração de quadros de horários educacionais em três grupos:

- Geração de quadros de horários para escolas (*school timetabling*);
- Geração de quadros de horários para universidades (*university timetabling*);
- Geração de quadros de horários para exames (*examination timetabling*).

O desenvolvimento de métodos computacionais para solucionar estes tipos de problemas já atraiu a atenção de pesquisadores de diversas áreas da comunidade científica nos últimos 50 anos (PETROVIC; BURKE, 2004). Muitas abordagens foram propostas para tentar superar os desafios envolvidos na geração de quadros de horários educacionais, entretanto, não existe uma abordagem *de facto* para a resolução desse tipo de problema. Um dos maiores desafios para a geração de quadros de horários é a variedade de requisitos específicos encontrados, conforme o ambiente em que o problema se contextualiza (BURKE et al., 1997).

Neste trabalho será apresentada uma nova variação do problema de geração de quadros de horários para universidades que ocorre no contexto dos cursos de Ciência da Computação (CiC) e de Engenharia da Computação (ECP) oferecidos no Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

1.1 Motivação

Em muitas universidades, o grande número de eventos associados a professores e recursos, combinados a uma grande variedade de requisitos, tornam a construção de uma grade de horários uma tarefa bastante complexa. A geração de quadros de horários de baixa qualidade pode prejudicar o rendimento de alunos. Por exemplo, uma sobrecarga de aulas ocorre quando

muitas aulas são alocados em um mesmo dia da grade de horários, possivelmente, dificultando o aprendizado dos estudantes. Além disso, dado a carga de conteúdo e a eventual necessidade de deslocamento entre campus universitários, pode ocasionar atrasos para que os estudantes cheguem ao local de suas aulas.

A representação de dados do problema é outro aspecto fundamental para a geração de quadros de horários, que consideram situações reais. (MCCOLLUM; IRELAND, 2006) ressaltam a dificuldade em se obter os dados necessários para resolução do problema, processo que despende considerável quantidade de tempo para enumerar as características de uma universidade.

No Instituto de Informática (INF), os quadros de horários são resolvidos manualmente através de planilhas, processo que engloba uma considerável quantidade de esforço e tempo dos membros da coordenação. O processo pode ser dividido nas seguintes tarefas: coletar dados, definir necessidades, realizar comparações com quadros de horários antigos, desenvolver os novos quadros de horários, analisar e corrigir os quadros de horários gerados. Uma gama de dados deve ser levada em consideração para a elaboração desse processo: número de vagas para cada disciplina, professores a alocar, disponibilidade de recursos, entre outros.

1.2 Objetivo do trabalho

O objetivo deste trabalho é desenvolver uma ferramenta que resolva o problema de geração de quadros de horários do INF, de maneira mais eficiente que a atual forma manual. Para o trabalho é proposto um modelo de representação de um curso universitário com base nas instâncias fornecidas pela coordenação dos cursos do INF. Para utilizar o modelo desenvolvido, uma interface Web é proposta para facilitar o processo de coleta de dados que representem características dos cursos do INF. Para geração dos quadros de horários, é proposto a criação de uma ferramenta que sintetize os dados de instâncias e consiga gerar quadros de horários de qualidade que possam ser interpretados com facilidade através de relatórios.

1.3 Organização do texto

Os demais capítulos deste trabalho estão organizados como a seguir: O Capítulo 2 realiza a revisão bibliográfica do estudo e a definição de alguns conceitos básicos sobre a geração de quadros de horários universitários. O Capítulo 3 define formalmente o problema do Ins-

tituto de Informática e apresenta uma formulação de programação linear inteira mista para o mesmo. O Capítulo 4 apresenta as implementações realizadas, descrevendo características e o funcionamento das ferramentas. O Capítulo 5 apresenta resultados computacionais referentes à resolução do modelo proposto utilizando cinco instâncias reais do INF/UFRGS. O capítulo 6 apresenta as conclusões deste estudo e propostas de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados trabalhos já desenvolvidos sobre geração de quadros de horários universitários e que se relacionam com o assunto abordado neste estudo. A revisão bibliográfica a seguir, apresenta definições sobre o problema da geração de quadros de horários e discute algumas técnicas para a geração de quadros de horários universitários.

2.1 Relevância do problema

Problemas de geração de quadros de horários são bem conhecidos na área da otimização, devido a sua vasta área de aplicação. Fica evidente o interesse de pesquisadores nesse tipo de estudo, observando a Figura 2.1 da pesquisa de (BARDADYM, 1996).

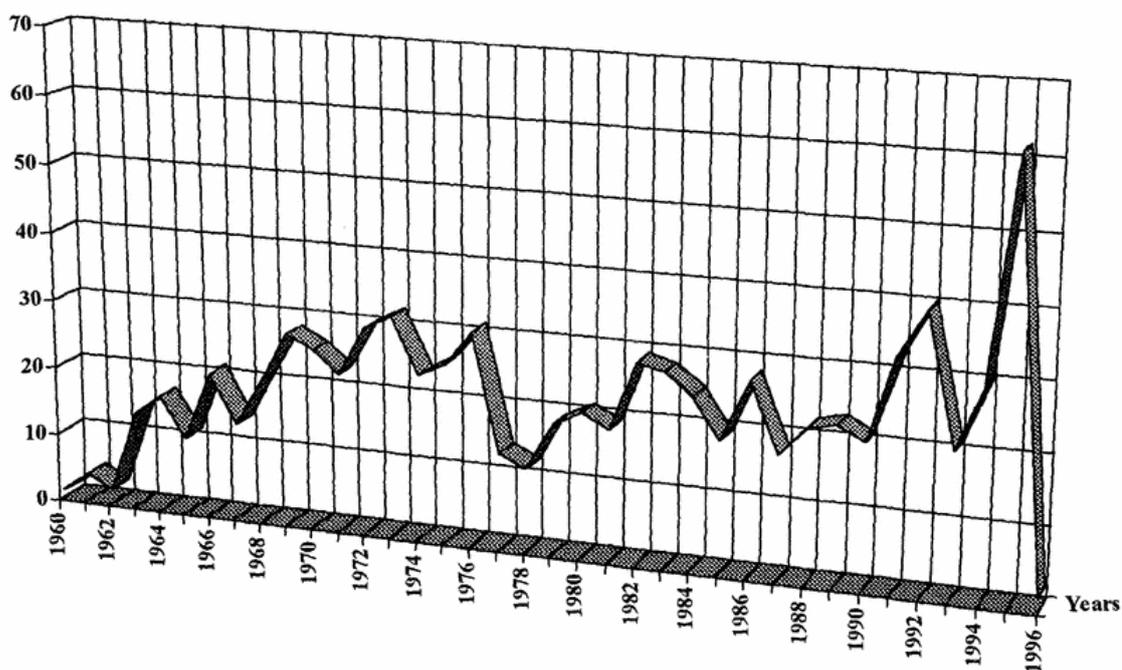


Figura 2.1 – Distribuição de publicações relacionadas à geração de quadros de horários (BARDADYM, 1996)

Outro fator que comprova o interesse de pesquisadores em problemas de geração de quadros de horários é a conferência bi-anual "Practice and Theory of Automated Timetabling"¹ (PATAT), em que estudos são publicados, promovendo a troca de conhecimento entre diversos interessados no tema, sendo a geração de quadros de horários educacionais um tema comumente

¹<http://www.patatconference.org/>

abordado.

2.2 Quadros de horários universitários

A geração de quadros de horários pode ser definida como o problema de atribuir um número de eventos a um número limitado de períodos de tempo. (WREN, 1996) define a criação de quadros de horários como: "A alocação, sujeita a restrições, de certos recursos para objetos sendo dispostos no tempo-espaço, de tal maneira que satisfaça, o tanto quanto possível, o conjunto de objetivos desejados".

Muitas pesquisas já foram realizadas sobre o tema, introduzindo inúmeros modelos e algoritmos para a formulação e resolução do problema. Apesar disso, o problema ainda é resolvido de forma manual, devido a falta de ferramentas computacionais específicas. Conforme (CARTER; LAPORTE; CHINNECK, 1994) diz: "Dependendo do tamanho do departamento e a diversidade de ofertas do curso, o tempo necessário para produzir os quadros de horários das aulas pode variar de um trabalho de uma tarde para um exaustivo e intenso trabalho que dura um mês."

Um outro aspecto que torna esse processo complexo é a necessidade de interação humana para a obtenção de uma solução de qualidade. Conforme (MCCOLLUM, 2007) nem sempre uma solução obtida por meios computacionais pode ser agradável aos responsáveis por desenvolver esse processo. Do ponto de vista de um software, qualquer solução obtida seria boa, mas a noção de solução otimizada, provavelmente, não satisfaça todos os requisitos desejados pelo encarregado desse processo. Existem fatores humanos que normalmente não são considerados por sistemas que resolvem o problema de forma automatizada, portanto, a solução final ainda passaria por um processo de avaliação humana e, eventualmente, um processo de correção realizado de forma manual.

2.3 Requisitos para geração de quadros de horários universitários

Normalmente, para o desenvolvimento de quadros de horários, os requisitos são divididos em *requisitos fortes* e *requisitos fracos*. Enquanto os requisitos fortes são aqueles que obrigatoriamente devem ser respeitados, os requisitos fracos são aqueles que preferencialmente devem ser respeitados.

Em estudos como o de (DIMOPOULOU; MILIOTIS, 2001), (DASKALAKI; BIRBAS;

HOUSOS, 2004), (BURKE et al., 2004) e (BURKE; PETROVIC, 2002) pode-se notar requisitos em comum para a geração de quadros de horários universitários:

- C1** *Colisões não podem serem permitidas.* Colisões ocorrem quando um recurso físico ou humano é alocado ao mesmo tempo para duas turmas.
- E1** *Continuidade.* Aulas de uma mesma disciplina devem ser alocados no mesmo recurso físico.
- E2** *Preferências de horários.* Algumas disciplinas devem ser ministradas antes que outras.
- E3** *Fixação de horários.* Algumas disciplinas possuem seus horários pré-estabelecidos.
- R1** *Quantidade de recursos físicos.* Não deve-se alocar um número maior de eventos do que a quantidade máxima disponível dos recursos físicos.
- R2** *Capacidade de recursos físicos.* Não deve-se exceder a capacidade de um recurso ao alocar uma turma.
- R3** *Disponibilidade de recursos.* Deve-se prever a possibilidade de um recurso, físico ou humano, não estar disponível em alguns períodos.
- Q1** *Quadros de horários completos.* Deve-se poder alocar todos os eventos desejados no quadro de horário em questão.
- Q2** *Quadros de horários compactos.* Deve-se gerar quadros de horários com o menor número possível de períodos ociosos para os estudantes e evitar dias letivos extensos.

2.4 Abordagens de resolução

Várias técnicas foram propostas para tentar suprir as dificuldades envolvidas em gerar quadros de horários universitários. Pode-se observar a utilização do método busca tabu em (LÜ; HAO, 2010) e (BURKE; KENDALL; SOUBEIGA, 2003). Outras abordagens como em (BURKE; ELLIMAN; WEARE, 1994), (WEARE; BURKE; ELLIMAN, 1995), (RICH, 1996) e (ERBEN; KEPPLER, 1996) propõem a utilização de algoritmos genéticos. Já em (LAJOS, 1996), (GOLTZ; MATZKE, 1998) e (ABDENNADHER; MARTE, 2000) observa-se a utilização de programação lógica por restrições.

Dado o avanço computacional, atualmente, formulações de programação inteira e programação inteira mista estão tornando-se uma abordagem válida para muitos problemas combinatoriais (JOHNSON; NEMHAUSER; SAVELSBERGH, 2000). Aliado ao uso de resolvidores matemáticos que conseguem resolver problemas relativamente grandes em um curto período de tempo, essa abordagem clássica de modelagem torna-se novamente atrativa. Em trabalhos como

o de (DASKALAKI; BIRBAS; HOUSOS, 2004), (SCHIMMELPFENG; HELBER, 2007) e (MIRHASSANI, 2006) são propostas modelagens matemáticas através de programação inteira descrevendo situações reais, cujo propósito é gerar quadros de horários que satisfaçam os requisitos que a instituição demanda, utilizando resolvedores matemáticos para a geração dos resultados. Outro aspecto que torna a abordagem de programação matemática interessante, é a possibilidade de ser combinada com outros métodos (DIMOPOULOU; MILIOTIS, 2001). A Tabela 2.1 apresenta um resumo de estudos sobre quadros de horários universitários, que propõem métodos semelhantes ao estudo desenvolvido neste trabalho, destacando a abordagem utilizada, o problema modelado e os requisitos considerados, conforme os requisitos descritos na Seção 2.3.

Autor	Método	Problema	Requisitos
(LAJOS, 1996)	Programação Lógica por Restrições	Universidade Alemã	C1,E2,R3
(GUÉRET et al., 1996)	Programação Lógica por Restrições	Universidade Francesa	C1,E1,E3,R2,R3
(BOIZUMAULT; DELON; PERIDY, 1996)	Programação Lógica por Restrições	Universidade Holandesa	C1,E2,E3,R2
(DERIS et al., 1997)	Programação Lógica por Restrições	Universidade Malasiana	C1,E3,R2
(DIMOPOULOU; MILIOTIS, 2001)	Programação Inteira	Universidade Grega	C1,Q1,R3
(DASKALAKI; BIRBAS; HOUSOS, 2004)	Programação Inteira	Universidade Grega	C1,E1,E2,E3,Q1
(AVELLA; VASIL'EV, 2005)	<i>Branch-And-Cut</i>	Universidade Italiana	C1,E1,E3,Q1,Q2,R2,R3
(MIRHASSANI, 2006)	Programação Inteira	Universidade Iraniana	C1,E3,Q1
(BAKIR; AKSOP, 2008)	Programação Inteira	Universidades Turcas	C1,E1,Q1,Q2
(DAMMAK et al., 2008)	Programação Inteira/Heurísticas	Universidade Tunisiana	C1, Q1,R3
(BURKE et al., 2010)	Programação Inteira	Universidade de Udine	C1,E2,E3,R3

Tabela 2.1 – Quadro comparativo de estudos

3 DEFINIÇÃO DO PROBLEMA E MODELAGEM

Neste capítulo são apresentadas as características observadas no problema de geração de quadros de horários dos cursos de CiC e ECP no INF. Na Seção 3.2 são apresentados os requisitos que fazem parte do problema. Por fim, na Seção 3.3 é apresentada uma modelagem matemática para o problema.

3.1 Descrição do problema

Os Cursos de CiC e da ECP são organizados em etapas semestrais, onde cada etapa tem seu próprio currículo de disciplinas. O INF disponibiliza a cada semestre uma gama de disciplinas eletivas que abrangem várias áreas de influência na informática, além das disciplinas obrigatórias que fazem parte do currículo essencial do estudante. Disciplinas obrigatórias são aquelas em que a coordenação do curso define como fundamental para o currículo do aluno, isto é, um aluno pode se formar, desde que seja aprovado em todas estas disciplinas. Já as eletivas são aquelas que o aluno tem a possibilidade de escolher livremente quais disciplinas ele deseja realizar, atendendo a uma quantidade mínima de créditos eletivos para que o estudante possa se formar.

O currículo de uma etapa pode envolver disciplinas oriundas de outros departamentos e que já possuem horários pré-determinados. Este é o caso, por exemplo, de disciplinas oferecidas pelos Departamentos de Matemática e Estatística. Além disso, cada disciplina normalmente é ofertada em várias turmas que compartilham professores e recursos físicos.

Além de lecionar aulas durante a semana, muitos professores são encarregados de projetos, pesquisas, tarefas administrativas e demais atividades, impossibilitando que estejam disponíveis durante todos os períodos da semana.

Alguns recursos são disponíveis durante todos períodos, o que é o caso das salas de aulas que são exclusivas para o uso dos cursos de CiC e da ECP. Enquanto outros recursos podem estar reservados por outros eventos, o que é o caso dos auditórios onde ocorre a aplicação de provas e palestras, desta maneira influenciando na disponibilidade do recurso.

Disciplinas oferecidas pelo cursos de CiC e ECP podem ser divididas em aulas teóricas e aulas práticas. Comumente, aulas teóricas são ministradas em salas de aula, enquanto as aulas práticas são ministradas em laboratórios. Eventualmente, algumas disciplinas teóricas podem necessitar de aulas em laboratório, mas nessa situação o laboratório precisa ser reservado pelo professor responsável pela disciplina. Outras disciplinas, possuem tanto aulas teóricas como

práticas, cujas aulas em sala de aula e em laboratório devem ser consideradas na geração dos quadros de horários.

A quantidade de aulas de cada disciplina é influenciada diretamente pelo o número de créditos que a disciplina representa. No caso desse estudo, cada dois créditos representam uma aula durante a semana. Particularmente, na CiC e ECP existem disciplinas com 2, 4 ou 6 créditos que correspondem, respectivamente, a 1, 2 ou 3 aulas na semana.

Durante o processo de construção da grade de horários da CiC e ECP procura-se reduzir a ocorrência de *períodos ociosos* (quando não existe atividade letiva alocada no horário e o estudante deve esperar até o começo da sua próxima aula). Também é desejado utilizar de forma eficaz os recursos físicos disponíveis no Instituto de Informática. Este último objetivo, em especial, tem ganho maior importância recentemente, devido ao aumento no número de alunos nas universidades brasileiras, inclusive no INF/UFRGS, sem o devido aumento proporcional da infraestrutura física.

O objetivo deste problema é construir quadros de horários semanais para todas as etapas dos cursos, atendendo restrições obrigatórias e minimizando a ocorrência de situações indesejadas no horário. A semana é dividida em um conjunto de D dias, os quais possuem um conjunto de horários H . Um *período* é definido pela tupla (d, h) , sendo $d \in D$ e $h \in H$, e assume-se que todos os períodos possuem a duração de uma aula. As aulas que precisam ser lecionadas no curso são definidas através de um conjunto de eventos E que é dividido em dois grupos: *eventos simples* (ES) e *eventos geminados* (EG). As aulas dos eventos simples ocorrem em dias distintos, e no mesmo período (por exemplo, segunda e quarta às 8:30, ou terça e quinta às 13:30). Enquanto as aulas dos eventos geminados ocorrem em um único dia e em períodos consecutivos (por exemplo, das 8:30 às 12:30). A cada etapa $m \in M$ está associada a um determinado conjunto de eventos E_m que compõe o currículo da etapa.

O problema ainda considera um conjunto de professores T e um conjunto de tipos de recursos R , que podem estar associados a um evento. Tanto professores como recursos, podem estar indisponíveis em alguns períodos na semana.

Alguns eventos não devem ser alocados, preferencialmente, no mesmo período da semana (por exemplo, para separar aulas teóricas e práticas de uma mesma disciplina). Quando essa preferência não é satisfeita, é dito que existe um *conflito* entre eventos.

O acúmulo de eventos alocados em um mesmo horário também deve ser evitado. É dever da coordenação garantir que os estudantes tenham a oportunidade de cursar o currículo mínimo de uma etapa, isto é, as disciplinas obrigatórias do semestre. Quando ocorre este acúmulo demasiado de eventos em um mesmo horário, ocasionando que o estudante deva escolher apenas

uma disciplina dentre as oferecidas, é dito que existem *eventos excedentes*.

3.2 Requisitos do Problema

Nas seções a seguir são definidos os requisitos fortes e fracos para o problema estudado.

3.2.1 Requisitos fortes

Quadros de horários factíveis devem atender obrigatoriamente todos os *requisitos fortes* apresentados abaixo:

- H1** Todas as aulas associadas com cada evento devem ser alocadas.
- H2** Eventos devem ser alocados respeitando a disponibilidade dos professores e recursos envolvidos no evento.
- H3** Professores devem ministrar no máximo uma aula em cada período.
- H4** Alguns eventos devem ser alocados em períodos pré-determinados.
- H5** A utilização de recursos em um dado período não pode exceder a quantidade disponível.
- H6** A cada 2 dias deve ocorrer no máximo uma aula de um evento simples.
- H7** Aulas de evento simples com carga horária igual a dois devem ser alocados com exatamente um dia de intervalo.
- H8** Eventos simples devem ter no máximo uma aula em cada dia.
- H9** Eventos simples devem ser alocados sempre no mesmo horário, obviamente em dias distintos.
- H10** Eventos geminados devem ser alocados em sequência em um único dia.
- H11** Eventos que ocorrem em Campus distintos não devem ser alocados em sequência.

3.2.2 Requisitos fracos

Além da factibilidade, é necessário satisfazer tanto quanto possível os *requisitos fracos* a seguir:

- S1** Minimizar a ocorrência de períodos ociosos em cada etapa.
- S2** Minimizar a ocorrência de conflitos entre eventos.
- S3** Deve-se evitar alocar mais de dois eventos da mesma etapa em um dado período. Os

eventos que excedem esse limite são chamados *eventos excedentes*.

S4 A utilização máxima de recursos em um período deve ser minimizada.

Para cada requisito fraco não atendido está associado um custo na função objetivo, valorado de acordo com a sua importância.

3.3 Formulação matemática do problema

Nesta seção é apresentada uma formulação de programação inteira mista que modela matematicamente todos os requisitos do problema descritos na seção anterior. Nas Seções 3.3.1 e 3.3.2 são apresentados, respectivamente, a função objetivo e a modelagem das restrições do problema. As demais notações utilizadas para este estudo são citadas na Tabela 3.1.

3.3.1 Função Objetivo

A finalidade da função objetivo é definir o custo associado à solução que será obtida. Como trata-se de um problema de minimização, quanto menor for o custo da solução, mais qualidade a mesma terá. A qualidade da função objetivo é baseada na satisfação dos requisitos fracos definidos na Seção 3.2.2.

$$\text{minimizar } \sum_{i \in E} \sum_{j \in E: i > j} \sum_{d \in D} \sum_{h \in H} \alpha c_{ijdh} C_{ij} + \sum_{m \in M} \sum_{d \in D} \sum_{h \in H} \beta b_{mdh} + \sum_{m \in M} \sum_{d \in D} \sum_{h \in H} \gamma q_{mdh} + \delta u$$

Na primeira parcela, o custo está associado à variável c_{ijdh} e ao parâmetro C_{ij} , determinando que os eventos i e j estão alocados em um mesmo horário e existe um conflito entre estes eventos. Para a segunda parcela, o custo refere-se a variável b_{mdh} , que contabiliza os períodos ociosos ao longo das etapas M . O custo da terceira parcela está associada à variável q_{mdh} , que contabiliza os eventos excedentes nas etapas M . A última parcela refere-se a variável u , que determina a quantidade de recursos utilizados.

Para o modelo em questão, cada parcela da função objetivo tem sua respectiva importância. Por exemplo, é preferível evitar conflitos entre aulas de determinadas disciplinas do que utilizar um recurso a mais do INF. Portanto, a importância de cada parcela da função objetivo está associada a um custo, os quais são definidos abaixo:

- $\alpha = 1000$: custo de cada conflito entre eventos;
- $\delta = 100$: custo pela utilização de um recurso;

- $\beta = 10$: custo de cada período ocioso;
- $\gamma = 1$: custo de cada evento excedente.

Símbolo	Definição
Conjuntos	
$d \in D$	conjunto de dias da semana.
$h \in H$	conjunto de horários do dia.
$h \in H'$	conjunto dos horários finais de cada turno do dia.
$t \in T$	conjunto de professores.
$e \in E$	conjunto de eventos.
ES	conjunto de eventos simples (aulas distribuídas na semana).
EG	conjunto de eventos geminados (aulas que ocorrem em um único dia da semana).
$r \in R$	conjunto de tipos de recurso.
$m \in M$	conjunto de etapas.
E_t	conjunto de eventos em que o professor t está envolvido.
E_r	conjunto de eventos em que o tipo de recurso r é necessário.
E_m	conjunto de eventos que fazem parte do currículo da etapa m .
Parâmetros	
$W_e \in \mathbb{N}^*$	número de aulas do evento e .
$V_{edh} \in \{0, 1\}$	indica se o evento e tem disponibilidade no período (d, h) .
$F_{edh} \in \{0, 1\}$	indica se o evento e tem uma aula pré-determinada no período (d, h) .
$Q_r \in \mathbb{N}^*$	quantidade disponível do tipo de recurso r .
$C_{ij} \in \{0, 1\}$	indica se existe um conflito entre os eventos i e j .
$L_{ij} \in \{0, 1\}$	indica se os eventos i e j ocorrem em campus diferentes.
$\alpha = 1000$	custo de cada conflito entre eventos.
$\delta = 100$	custo pela utilização de um recurso.
$\beta = 10$	custo de cada período ocioso.
$\gamma = 1$	custo de cada evento excedente.
Variáveis	
$x_{edh} \in \{0, 1\}$	define se o evento e está alocado no período (d, h) .
$f_{eh} \in \{0, 1\}$	indica que todas as aulas do evento e devem ocorrer no horário h .
$g_{edh} \in \{0, 1\}$	indica que no período (d, h) começa uma aula geminada.
$b_{mdh} \in \{0, 1\}$	indica a ocorrência de um período ocioso em (d, h) na etapa m .
$c_{ijdh} \in \{0, 1\}$	indica se os eventos i e j estão alocados no mesmo período (d, h) .
$y_{mdh} \in \{0, 1\}$	indica se existe no mínimo um evento da etapa m alocado no período (d, h) .
$q_{mdh} \in \mathbb{N}$	número de eventos excedentes da etapa m alocados no período (d, h) .
$u \in \mathbb{N}$	quantidade máxima de recursos utilizados em um período.

Tabela 3.1 – Notação utilizada pelo modelo

3.3.2 Restrições

As restrições, apresentadas nesta seção, são as representações matemáticas dos requisitos definidos nas Seções 3.2.1 e 3.2.2.

O conjunto de restrições (3.1) garante que o número de aulas de cada evento seja atendido.

$$\sum_{d \in D} \sum_{h \in H} x_{edh} = W_e \quad \forall e \in E \quad (3.1)$$

O conjunto de restrições (3.2) garante que o evento seja alocado apenas em períodos disponíveis.

$$x_{edh} \leq V_{edh} \quad \forall e \in E, d \in D, h \in H \quad (3.2)$$

O conjunto de restrições (3.3) proíbe que um professor participe de mais de um evento em cada período.

$$\sum_{e \in E_t} x_{edh} \leq 1 \quad \forall t \in T, d \in D, h \in H \quad (3.3)$$

O conjunto de restrições (3.4) garante que as aulas com períodos pré-determinados sejam alocadas.

$$x_{edh} \geq F_{edh} \quad \forall e \in E, d \in D, h \in H \quad (3.4)$$

O conjunto de restrições (3.5) garante que a quantidade disponível em cada tipo de recurso não seja excedida.

$$\sum_{e \in E_r} x_{edh} \leq Q_r \quad \forall r \in R, d \in D, h \in H \quad (3.5)$$

O conjunto de restrições (3.6) garante que aulas de um mesmo evento simples não sejam alocadas em dias consecutivos a fim de atender o requisito H6. O conjunto de restrições (3.7) impede que exista mais do que um dia livre entre as aulas de eventos simples com apenas 2 aulas. Esta restrição combinada com o conjunto de restrições (3.6) modela o requisito H7. O conjunto de restrições (3.8) determina que todas as aulas de um evento devam ocorrer exatamente no

mesmo horário durante a semana.

$$\sum_{h \in H} (x_{edh} + x_{e,d+1,h}) \leq 1 \quad \forall e \in ES, d \in D : d < |D| \quad (3.6)$$

$$\sum_{h \in H} (x_{eih} + x_{ejh}) \leq 1 \quad \forall e \in ES, i \in D, j \in D : j \geq i + 3, W_e = 2 \quad (3.7)$$

$$W_e f_{eh} = \sum_{d \in D} x_{edh} \quad \forall e \in ES, h \in H \quad (3.8)$$

Os conjuntos de restrições (3.9), (3.10) e (3.11) modelam o requisito H9. Os conjuntos de restrições (3.9) e (3.10) identificam aulas que ocorrem em sequência, enquanto, o conjunto de restrições (3.11) garante que exista apenas uma aula ocorrendo em sequência.

$$g_{edh} \leq x_{edh} \quad \forall e \in EG, d \in D, h \in H \setminus H' \quad (3.9)$$

$$g_{edh} \leq x_{e,d,h+1} \quad \forall e \in EG, d \in D, h \in H \setminus H' \quad (3.10)$$

$$\sum_{d \in D} \sum_{h \in H \setminus H'} g_{edh} = 1 \quad \forall e \in EG \quad (3.11)$$

O conjunto de restrições (3.12) garante que haja um intervalo para o deslocamento dos estudantes entre as aulas de eventos que ocorrem em campus distintos.

$$x_{idh} + x_{jdh} \leq 1 \quad \forall i \in E, j \in E, d \in D, h \in H - H' : i > j, L_{ij} > 0 \quad (3.12)$$

Os conjuntos de restrições (3.13) e (3.14) garantem que a variável y_{mdh} seja ativada caso exista no mínimo uma aula do semestre m ocorrendo no período (d, h) . Os conjuntos de restrições (3.13), (3.14) e (3.15) modelam o requisito S1 de maneira semelhante a modelagem utilizada por (AVELLA; SALERNO; VASIL, 2007) e (DORNELES; ARAÚJO; BURIOL, 2012).

$$y_{mdh} \geq x_{edh} \quad \forall m \in M, e \in E_m, d \in D, h \in H \quad (3.13)$$

$$y_{mdh} \leq \sum_{e \in E_m} x_{edh} \quad \forall m \in M, d \in D, h \in H \quad (3.14)$$

$$b_{mdh} \geq -1 + y_{mdi} + y_{mdj} - y_{mdh} \quad \forall m \in M, d \in D, h \in H, i \in H, j \in H : i < h < j \quad (3.15)$$

O conjunto de restrições (3.16) ativam a variável c_{ijdh} caso os eventos conflitantes i e j

sejam alocados no mesmo período (d, h) .

$$x_{idh} + x_{jdh} \leq 1 + c_{ijdh} \quad \forall i \in E, j \in E, d \in D, h \in H : i > j, C_{ij} = 1 \quad (3.16)$$

O conjunto de restrições (3.17) identificam na variável q_{mdh} o número de eventos excedentes ocorrendo no semestre m no período (d, h) .

$$\sum_{e \in E_m} x_{edp} \leq 2 + q_{mdh} \quad \forall m \in M, d \in D, h \in H \quad (3.17)$$

O conjunto de restrições (3.18) identifica na variável u a quantidade máxima de recursos utilizados em um dado período.

$$\sum_{r \in R} \sum_{e \in E_r} x_{edh} \leq u \quad \forall d \in D, h \in H \quad (3.18)$$

Os demais conjuntos de restrições, listados abaixo, são restrições de integralidade que definem o domínio das variáveis.

$$\begin{aligned} x_{edh} &\in \{0, 1\} & \forall e \in E, d \in D, h \in H \\ f_{eh} &\in \{0, 1\} & \forall e \in E, h \in H \\ g_{edh} &\in \{0, 1\} & \forall e \in EG, d \in D, h \in H \setminus H' \\ c_{ijdh} &\in \{0, 1\} & \forall i \in E, j \in E, d \in D, h \in H : i > j \\ y_{mdh} &\in \{0, 1\} & \forall m \in M, d \in D, h \in H \\ b_{mdh} &\in \{0, 1\} & \forall m \in M, d \in D, h \in H \\ q_{mdh} &\in \mathbb{N} & \forall m \in M, d \in D, h \in H \\ u &\in \mathbb{N} \end{aligned}$$

Utilizando linguagens de programação matemática é possível representar o modelo matemático proposto nesta seção e resolvê-lo de forma automatizada.

4 IMPLEMENTAÇÃO

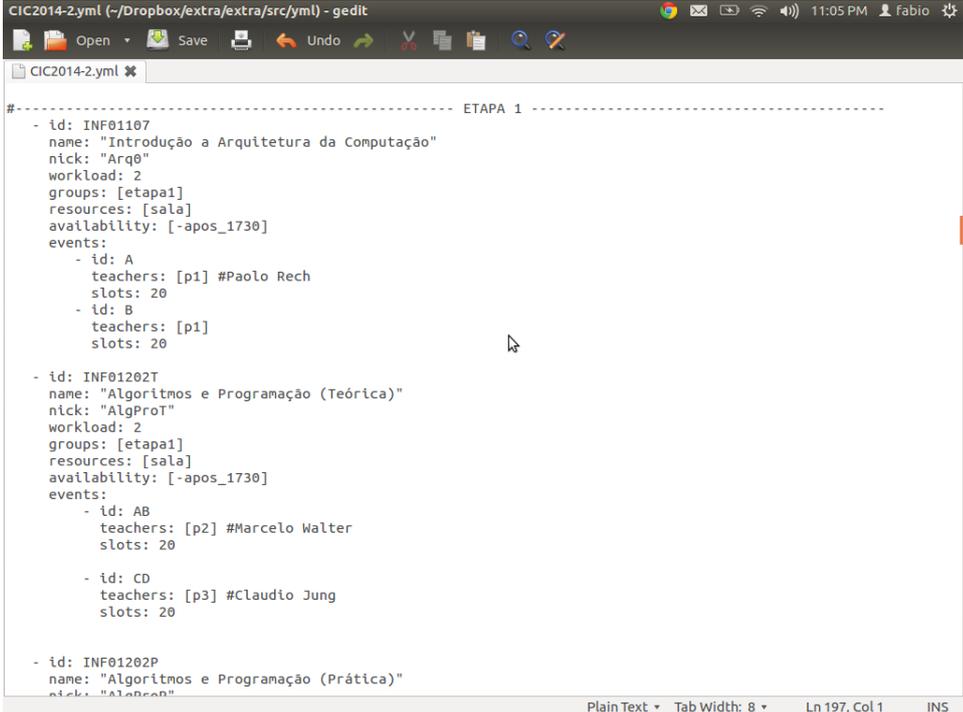
Este capítulo detalha a criação das ferramentas desenvolvidas para este trabalho. Primeiramente será apresentado o modelo de instância desenvolvido para representar os cursos de CiC e ECP. Em seguida é apresentado o modelo matemático no formato GNU Mathematical Programming Language (GMPL), definido na Secção 3.3. Por fim, são apresentadas as implementações de uma interface gráfica e, finalmente, uma ferramenta resolvidora que gera quadros de horários com base no modelo matemático.

A implementação consiste na criação de duas ferramentas:

- **Interface Gráfica:** é responsável pela manipulação de informações úteis ao problema e geração de arquivos de entrada para a segunda ferramenta.
- **Resolvidor:** é responsável por gerar os quadros de horários com base no arquivo de instância gerado pela interface gráfica e no modelo matemático do problema.

4.1 Formato das Instâncias

Para esse estudo, foi elaborado um formato de instância para representar os dados da CiC e ECP do INF, utilizando a linguagem YAML Ain't Markup Language (YAML). Uma das principais características do YAML é a representação de dados (ERIKSSON; HALLBERG, 2011). O YAML, ao contrário do formato Extensible Markup Language (XML), é mais simples de ler, editar e produzir, portanto, é ideal para representar as características diversas do problema de geração de quadros de horários do INF. O objetivo deste formato é padronizar os dados do INF, permitindo que pesquisadores testem outras abordagens de resolução com base nos arquivos de instâncias desenvolvidos. Na Figura 4.1 é possível observar a representação de duas disciplinas do curso da CiC, em ambas são ofertadas duas turmas.



```

CIC2014-2.yml (~/.Dropbox/extra/extra/src/yml) - gedit
CIC2014-2.yml x
#----- ETAPA 1 -----
- id: INF01107
  name: "Introdução a Arquitetura da Computação"
  nick: "Arq0"
  workload: 2
  groups: [etapa1]
  resources: [sala]
  availability: [-apos_1730]
  events:
    - id: A
      teachers: [p1] #Paolo Rech
      slots: 20
    - id: B
      teachers: [p1]
      slots: 20
- id: INF01202T
  name: "Algoritmos e Programação (Teórica)"
  nick: "AlgProT"
  workload: 2
  groups: [etapa1]
  resources: [sala]
  availability: [-apos_1730]
  events:
    - id: AB
      teachers: [p2] #Marcelo Walter
      slots: 20
    - id: CD
      teachers: [p3] #Claudio Jung
      slots: 20
- id: INF01202P
  name: "Algoritmos e Programação (Prática)"
  nick: "AlgProP"

```

Figura 4.1 – Parte da instância no formato YAML

4.2 Modelagem GMPL

A linguagem GMPL permite representar o modelo definido na Seção 3.3. Utilizando a linguagem consegue-se resolver problemas de otimização e análise combinatória através do resolvidor GLPK (MAKHORIN, 2008) ou permite exportar um arquivo LP¹ que pode ser resolvido por outros resolvidores. Optou-se pelo uso do formato GMPL, pois é uma linguagem de alto nível e de fácil manutenção. Outra possibilidade seria representar o problema diretamente em um formato LP, no entanto, o processo para gerar um arquivo LP não é trivial e a manutenção do mesmo não é uma tarefa simples. Na Figura 4.2 é possível observar a representação do função objetivo e de algumas restrições no formato GMPL.

¹<http://www-01.ibm.com/support/knowledgecenter/SSSA5P12.6.1/ilog.odms.cplex.help/CPLEX/FileFormats/topics/LP.html>

```

tt-ufrgs.mod (-/Dropbox/extra/extra/src/mods) - gedit
CIC2014-2.yml tt-ufrgs.mod
#Função Objetivo-----
minimize Z:
  sum{i in E, j in E, s in S: EREL[i,j] > 0} c[i,j,s]*EREL[i,j] +
  sum{m in M, d in D, p in P} q[m,d,p] +
  sum{m in M, d in D, p in P} b[m,d,p] * 10+
  u * 100;

#Restrições HARD-----
s.t. H1 {e in E}:
  sum{s in S} x[e,s] = W[e]; #Carga horária do evento

s.t. H2 {e in E, s in S}: #Disponibilidade do Evento
  x[e,s] <= V[e,s];

s.t. H3 {t in T, d in D, p in P}: #Permitir no máximo 1 aula do professor por posição
  sum{e in Et[t]} x[e,SS[d,p]] <= 1;

s.t. H4 {e in E, s in S}: #Aulas fixas dos eventos
  x[e,s] >= F[e,s];

s.t. H5 {r in R, d in D, p in P}:
  sum{e in Er[r]} x[e,SS[d,p]] <= Q[r]; #Conflito de recurso

|
s.t. H6 {e in A, d in D: d < card(D)}: #Proibe aulas do mesmo evento em dias consecutivos (Apenas aulas não
join_all)
  sum{p in P} (x[e,SS[d,p]] + x[e,SS[d+1,p]]) <= 1;

s.t. H7a {e in A, d in D: d < card(D) and W[e] <= 2}: #Proibe que as aulas sejam colocadas na Segunda E na
Sexta
  sum{p in P} (x[e,SS[d,p]] + x[e,SS[card(D),p]]) <= 1;

s.t. H7b {e in A: W[e] <= 2}: #Proibe que as aulas sejam colocadas na Segunda E na quinta
  sum{p in P} (x[e,SS[1,p]] + x[e,SS[card(D)-1,p]]) <= 1;

```

Figura 4.2 – Parte do modelo matemático representado em GMPL

4.3 Interface Gráfica

A interface gráfica é uma ferramenta web cuja finalidade é facilitar a criação de arquivos que são fornecidos a ferramenta Resolvedor. Desenvolvida em um sistema operacional Linux, codificada na linguagem PHP com auxílio do framework web Yii2, é executada em um servidor Apache e seus dados são armazenados em um banco de dados MySQL. A ferramenta gera arquivos YAML no formato proposto na Seção 4.1. Para utilizar a ferramenta, o usuário deve efetuar login para poder realizar funções básicas de criação, leitura, edição e deleção de dados. A codificação da interface gráfica está disponibilizada em <http://www.inf.ufrgs.br/fvpneukirchen/tt-ufrgs>. A Figura 4.3 apresenta o modelo entidade-relacionamento do banco de dados (CHEN, 1976) onde são representadas as principais entidades do problema.

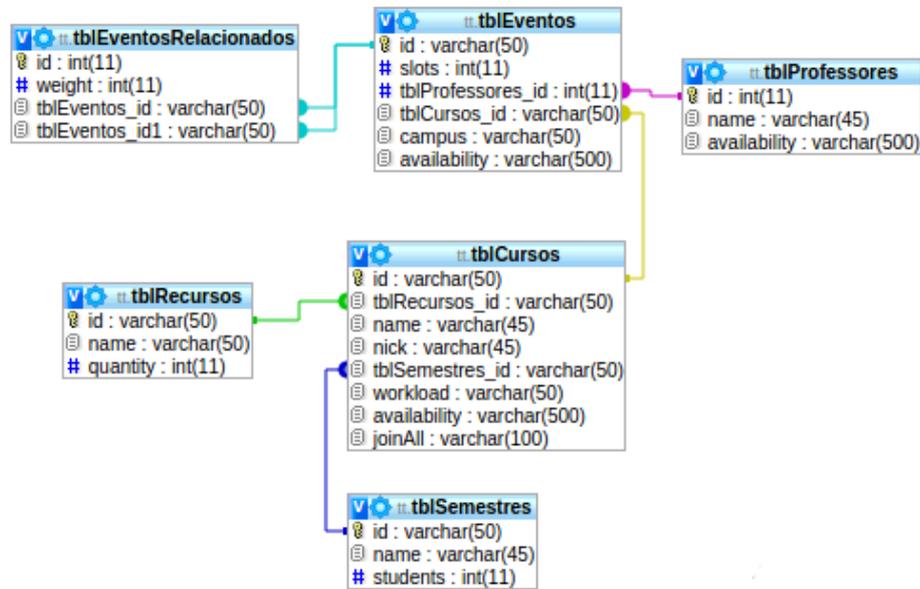


Figura 4.3 – Modelo entidade-relacionamento dos dados do INF

A interface permite manipular dados de:

- Cursos
- Turmas
- Conflitos entre turmas
- Professores
- Recursos Físicos

Nas seções seguintes são apresentadas as principais características de cada elemento.

4.3.1 Cursos

Um curso representa as disciplinas ministradas nos currículos dos cursos de CiC e ECP. Acessando a aba Disciplinas e selecionando Cursos, é possível gerenciar os cursos cadastrados (Figura 4.4) ou clicar no botão *Criar Curso* e realizar a criação de um novo curso (Figura 4.5).

Os principais dados de um curso são:

- Código do curso
- Recurso físico utilizado
- Nome da curso
- Quantidade de créditos
- Se tem aulas em sequência

- Abreviação
- Semestre
- Indisponibilidade

Quadro de Hóraros do Instituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas Professores Logout (admin)

Cursos

[Criar Curso](#)

Exibindo 1-20 de 38 itens.

#	Código	Nome	Semestre	Espaço Físico Utilizado	Indisponibilidade	
1	INF01032	Empreendimento em Informática	sem6	sala		 
2	INF01043	Interação Homem Computador	sem6	sala		 
3	INF01046	Fundamentos de Processamento de Imagens	sem4	sala		 
4	INF01047	Fundamentos de Computação Gráfica	sem5	sala		 
5	INF01048	Inteligência Artificial	sem5	sala		 
6	INF01058	Circuitos Digitais	sem3	sala		 
7	INF01107	Introdução a Arquitetura da Computação	sem1	sala	seg5,ter5,qua5,qui5,sex5	 
8	INF01108	Arquitetura e Organização I	sem2	sala		 
9	INF01112	Arquitetura e Organização II	sem3	sala		 
10	INF01113	Organização de Computadores B	sem4	sala		 
11	INF01120	Técnicas de Construção de Programas	sem4	sala		 
12	INF01121	Modelos de Linguagens de Programação	sem5	sala		 
13	INF01124	Classificação e Pesquisa de Dados	sem3	sala		 

Figura 4.4 – Tela da *Interface Gráfica* - Visualização da aba Cursos

Quadro de Horários do Instituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas Professores Logout (admin)

Criando um Curso...

Código

Espaço Físico Utilizado

Nome

Créditos

Aulas em Sequência?

Abreviação

Semestre

Indisponibilidade

Seg 08:30
 Seg 10:30
 Seg 13:30
 Seg 15:30
 Seg 17:30
 Ter 08:30
 Ter 10:30
 Ter 13:30
 Ter 15:30
 Ter 17:30
 Qua 08:30
 Qua 10:30
 Qua 13:30
 Qua 15:30
 Qua 17:30
 Qui 08:30
 Qui 10:30
 Qui 13:30
 Qui 15:30
 Qui 17:30
 Sex 08:30
 Sex 10:30
 Sex 13:30
 Sex 15:30
 Sex 17:30

Figura 4.5 – Tela da *Interface Gráfica* - Criação de um Curso

4.3.2 Turmas

Acessando a aba Disciplinas e selecionando Turmas, é possível gerenciar as turmas cadastradas (Figura 4.6) ou clicar no botão *Criar Turma* e realizar a criação de uma nova turma. (Figura 4.7). Para criar uma turma, é necessário que exista um curso, para que seja feita uma associação curso-turma.

Os principais dados de uma turma são:

- Código da turma
- Quantidade de vagas
- Professor ministrante
- A qual curso pertence
- Campus onde ocorre a aula
- Horário a ser fixado

Quadro de Hórgarios do Insituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas Professores Logout (admin)

Turmas

[Criar Turmas](#)

Exibindo 1-20 de 80 itens.

#	Turma	Vagas	Pertence ao Curso	Fixado em	
1	INF01032-U	50	INF01032		 
2	INF01043-U	30	INF01043		 
3	INF01046-A	35	INF01046		 
4	INF01046-B	35	INF01046		 
5	INF01047-U	35	INF01047		 
6	INF01048-U	40	INF01048		 
7	INF01058-A	1	INF01058		 
8	INF01058-B	18	INF01058		 
9	INF01058-C	18	INF01058		 
10	INF01107-A	20	INF01107		 
11	INF01107-B	20	INF01107		 
12	INF01108-A	39	INF01108		 
13	INF01108-B	10	INF01108		 

Figura 4.6 – Tela da *Interface Gráfica* - Visualização da aba Turmas

Quadro de Hórgarios do Insituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas Professores Logout (admin)

Criando Turma...

Turma

Vagas

Professor

Pertence ao Curso

Campus

Fixar Horário
 Seg 08:30 Seg 10:30 Seg 13:30 Seg 15:30 Seg 17:30 Ter 08:30 Ter 10:30 Ter 13:30 Ter 15:30 Ter 17:30 Qua 08:30
 Qua 10:30 Qua 13:30 Qua 15:30 Qua 17:30 Qui 08:30 Qui 10:30 Qui 13:30 Qui 15:30 Qui 17:30 Sex 08:30 Sex 10:30
 Sex 13:30 Sex 15:30 Sex 17:30

[Criar](#)

© Quadro de Horário INF-UFRGS 2015 Powered by Yii Framework

Figura 4.7 – Tela da *Interface Gráfica* - Criação de uma Turma

4.3.3 Relações entre Turmas

Um conflito, como definido em 3.1, é quando deve-se evitar que turmas sejam alocadas no mesmo horário. Um curso representa as disciplinas ministradas nos currículos dos cursos de CiC e ECP. Acessando a aba Disciplinas e selecionando Relação entre Turmas, é possível gerenciar as relações entre turmas cadastradas (Figura 4.8) ou clicar no botão *Criar Relação* e realizar a criação de uma nova relação. (Figura 4.9). Para criar uma relação entre turmas, é necessário que existam pelo menos duas turmas cadastradas, para que seja feita uma associação turma-turma.

Os principais dados de uma relação entre turmas são:

- Código da 1ª turma
- Código da 2ª turma

Quadro de Horários do Instituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas Professores Logout (admin)

Relação entre Turmas

Criar nova Relação

Exibindo 1-1 de 1 item.

#	Turma 1	Turma 2	
1	INF01032-U	INF01043-U	 

© Quadro de Horário INF-UFRGS 2015

Powered by Yii Frame

Figura 4.8 – Tela da *Interface Gráfica* - Visualização da aba Relação entre Turmas

The screenshot shows a web interface for creating a relationship between classes. At the top, there is a dark header with the text 'Quadro de Hórgios do Instituto de Informática da UFRGS' and a navigation menu with items: 'Início', 'Sobre', 'Gerador', 'Recursos', 'Disciplinas', 'Professores', and 'Logout (admin)'. The main content area has the title 'Criando Relação entre Turmas...'. Below the title, there are three dropdown menus: 'Proximidade' (set to 'Provavelmente Afastados'), 'Turma 1' (set to 'Escolha...'), and 'Turma 2' (set to 'Escolha...'). A green 'Criar' button is positioned below the 'Turma 2' dropdown. At the bottom of the page, there is a footer with '© Quadro de Horário INF-UFRGS 2015' on the left and 'Powered by Yii Framework' on the right.

Figura 4.9 – Tela da *Interface Gráfica* - Criação de uma relação entre Turmas

4.3.4 Professores

Um professor representa o Corpo Docente dos cursos de CiC e ECP. Acessando a aba Professores, é possível gerenciar os professores cadastrados (Figura 4.10) ou clicar no botão *Criar Professor* e realizar o cadastro de um novo professor (Figura 4.11).

Os principais dados de um professor são:

- Nome do professor
- Indisponibilidades

Quadro de Hóraros do Insituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas **Professores** Logout (admin)

Professores

[Criar Professor](#)

Exibindo 1-20 de 56 itens.

Nome	Indisponibilidade	
Não Definido/Não ministra as aulas no INF		 
Luciana Buriol		 
Taisy Weber		 
Marcelo Pimenta		 
Professor 4	seg1,seg2,seg3,seg4,seg5,qua1,qua2,qua3,qua4,qua5,sex1,sex2,sex3,sex4,sex5	 
Paolo Rech		 
Marcelo Walter		 
Claudio Jung		 
Egon		 
Leila Ribeiro		 
Lucio Duarte		 
Ana Bazzan		 
Raul Weber		 

Figura 4.10 – Tela da *Interface Gráfica* - Visualização da aba Professores

Quadro de Hóraros do Insituto de Informática da UFRGS

Início Sobre Gerador Recursos Disciplinas **Professores** Logout (admin)

Criando Professor...

Nome

Indisponibilidade

Seg 08:30
 Seg 10:30
 Seg 13:30
 Seg 15:30
 Seg 17:30
 Ter 08:30
 Ter 10:30
 Ter 13:30
 Ter 15:30
 Ter 17:30
 Qua 08:30
 Qua 10:30
 Qua 13:30
 Qua 15:30
 Qua 17:30
 Qui 08:30
 Qui 10:30
 Qui 13:30
 Qui 15:30
 Qui 17:30
 Sex 08:30
 Sex 10:30
 Sex 13:30
 Sex 15:30
 Sex 17:30

[Criar](#)

© Quadro de Horário INF-UFRGS 2015 Powered by Yii Framework

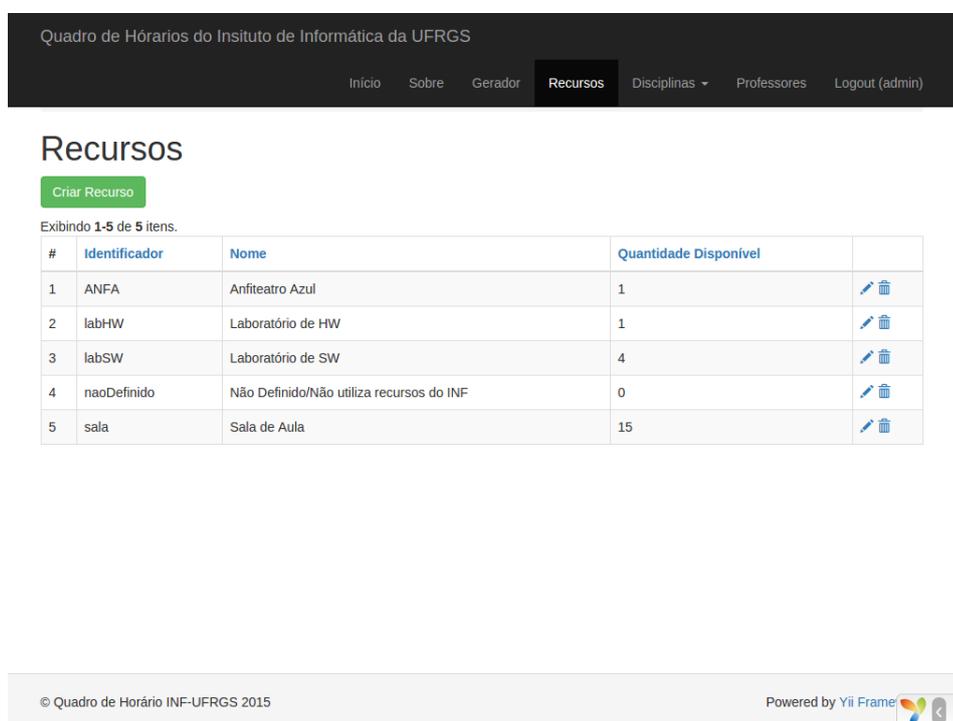
Figura 4.11 – Tela da *Interface Gráfica* - Criação de um Professor

4.3.5 Recursos Físicos

Um recurso representa os recursos físicos do INF. Acessando a aba Recursos, é possível gerenciar os recursos cadastrados (Figura 4.12) ou clicar no botão *Criar Recursos* e realizar a criação de um novo recurso (Figura 4.13).

Os principais dados de um recurso são:

- Abreviação
- Nome do recurso
- Quantidade Disponível



Quadro de Horários do Instituto de Informática da UFRGS

Início Sobre Gerador **Recursos** Disciplinas Professores Logout (admin)

Recursos

[Criar Recurso](#)

Exibindo 1-5 de 5 itens.

#	Identificador	Nome	Quantidade Disponível	
1	ANFA	Anfiteatro Azul	1	✎ 🗑️
2	labHW	Laboratório de HW	1	✎ 🗑️
3	labSW	Laboratório de SW	4	✎ 🗑️
4	naoDefinido	Não Definido/Não utiliza recursos do INF	0	✎ 🗑️
5	sala	Sala de Aula	15	✎ 🗑️

© Quadro de Horário INF-UFRGS 2015 Powered by Yii Framework

Figura 4.12 – Tela da *Interface Gráfica* - Visualização da aba Recursos

The screenshot shows a web application interface for creating a resource. At the top, there is a dark header with the text 'Quadro de Hóraros do Instituto de Informática da UFRGS' and a navigation menu with links: 'Início', 'Sobre', 'Gerador', 'Recursos', 'Disciplinas', 'Professores', and 'Logout (admin)'. Below the header, the main content area is titled 'Criando Recurso...'. It contains three input fields: 'Abreviação', 'Nome', and 'Quantidade'. A green 'Criar' button is positioned below the 'Quantidade' field. At the bottom of the page, there is a footer with the text '© Quadro de Horário INF-UFRGS 2015' on the left and 'Powered by Yii Framework' on the right, accompanied by the Yii Framework logo.

Figura 4.13 – Tela da *Interface Gráfica* - Criação de um Recurso

4.3.6 Geração do arquivo de instância

Para gerar-se a instância, deve-se acessar a aba Gerador e clicar no link *Clique aqui para realizar o download da instância gerada do problema* (Figura 4.14). A aplicação carrega todos os dados armazenados no banco de dados e gera uma instância no formato YAML, conforme definida na Seção 4.1, que será utilizada como entrada pela ferramenta Resolvedor.

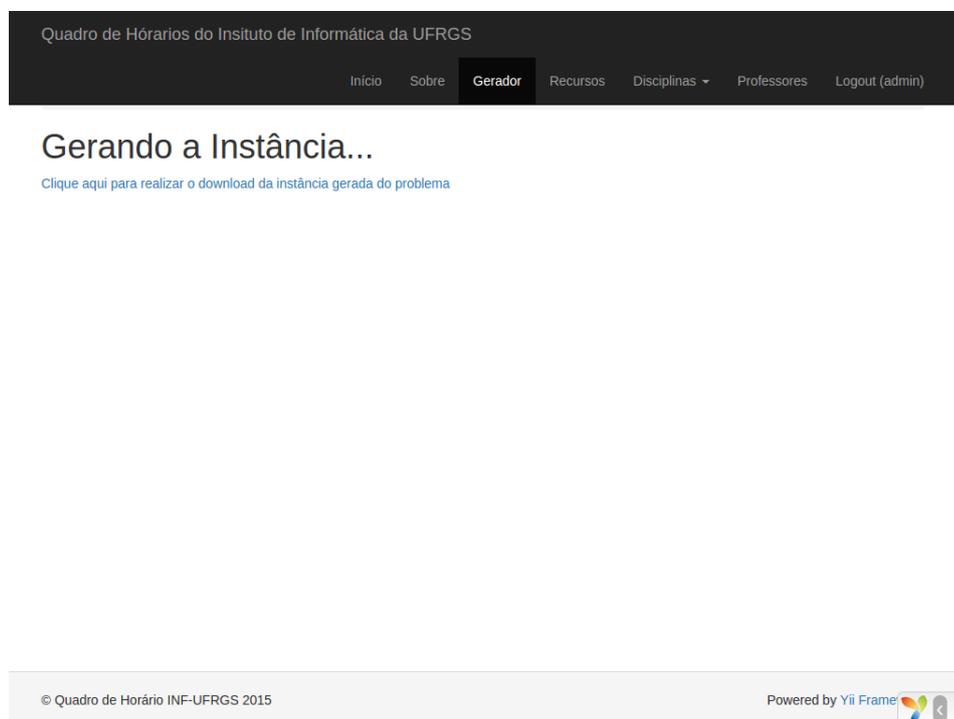


Figura 4.14 – Tela da *Interface Gráfica*-Gerando instância

4.4 Resolvedor

A ferramenta Resolvedor é responsável por gerar os quadros de horários. Desenvolvida na linguagem C++, ela utiliza bibliotecas `yaml.h` (YAML, 2009), `glpk.h` (GLPK, 2009) e `cplex.h` (CPLEX, 2013) para auxiliar no processamento dos dados. É executado via linha de comando em um sistema operacional Linux. O fluxo de execução do resolvedor pode ser observado na Figura 4.15 e é detalhado nas seções a seguir. A codificação do resolvedor está disponibilizada em <<http://www.inf.ufrgs.br/fvpneukirchen/tt-ufrgs>>

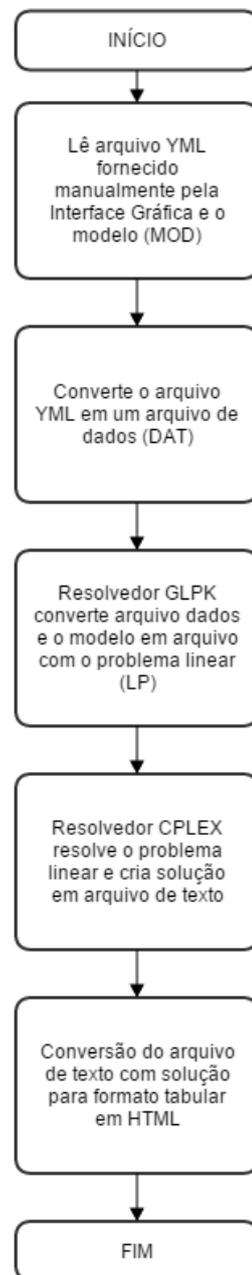


Figura 4.15 – Fluxo de execução do *Resolvedor*

4.4.1 Leitura da instância

A leitura da instância na linguagem YAML ocorre através de funções da biblioteca `yaml.h`. Os dados são armazenados em estrutura de dados que representam Dias, Semestres, Professores, Recursos, Disciplinas, Turmas e Relações entre Turmas.

4.4.2 Criação do arquivo de dados

O arquivo de dados (formato `.DAT`) é criado realizando interações nas estruturas criadas na Seção 4.4.1. No arquivo de dados são criados os Conjuntos e os Parâmetros requisitados pelo modelo definido na Seção 3.3. Tanto o arquivo de dados quanto o modelo matemático são representados na linguagem GMPL.

4.4.3 Criação do Problema Linear

Tendo como entrada o arquivo de dados (`.DAT`) e o modelo matemático (`.MOD`), utiliza-se a função `GMPL2LP` da biblioteca `glpk.h`, para leitura dos arquivos e criação de um arquivo contendo um problema linear (formato `.LP`) que é um dos formatos utilizados como entrada do Resolvedor CPLEX.

4.4.4 Solucionando Problema Linear

Para solucionar o problema linear criado na Seção 4.4.3 é utilizado o resolvedor comercial CPLEX. Utilizando funções da biblioteca `cplex.h`, o arquivo LP é carregado e resolvido, gerando um arquivo de texto com a solução (Figura 4.16) como saída. No arquivo são listadas as variáveis utilizadas no modelo e os valores que compõem a solução. Além disso, é exibido o custo obtido, o custo esperado, a diferença percentual entre o custo esperado e custo obtido, o número de variáveis e de restrições do modelo e, finalmente, o estado da solução.

```

CIC2014-2.txt (-/Dropbox/extra/extra/src/yml) - gedit
Instance=yml/CIC2014-2.lp Obj=910.000000 LB=910.000000 Gap=0.000000 Cols=7576 Rows=15937 Status=Optimal
q(7,5,2): 3
q(7,5,3): 4
q(7,5,4): 3
u: 9
x(1,2): 1
x(1,12): 1
x(2,9): 1
x(2,19): 1
x(3,3): 1
x(3,13): 1
x(4,8): 1
x(4,18): 1
x(5,23): 1
x(6,22): 1
x(7,22): 1
x(8,23): 1
x(9,3): 1
x(9,13): 1
x(10,9): 1
x(10,19): 1
x(11,6): 1
x(11,16): 1
x(12,1): 1
x(12,11): 1
x(12,21): 1
x(13,6): 1
x(13,16): 1
x(14,7): 1
x(14,17): 1
x(15,3): 1
x(15,13): 1
x(16,2): 1
x(16,12): 1
x(17,4): 1
x(17,14): 1
x(18,2): 1

```

Figura 4.16 – Arquivo de saída do CPLEX

4.4.5 Leitura do arquivo com solução final

Converte-se a solução fornecida pelo CPLEX em relatórios HTML no formato de tabelas. São criados quatro tipos de relatórios: um relatório que contém aulas de cada semestre (Figura 4.17), um com informações de professores (Figura 4.18), outro de recursos físicos (Figura 4.19) e, por fim, um relatório que apresenta informações de aulas em todos os semestres (Figura 4.20).

Etapa 1

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30	MAT01353_(A3)	INF05008_(C) MAT01375_(A)	MAT01353_(A3)	INF05008_(C) MAT01375_(A)	MAT01353_(A3)
(2) 10:30	INF01107_(A)	MAT01375_(B)	INF01107_(A)	MAT01375_(B)	INF01202P_(B) INF01202P_(C)
(3) 13:30	INF01202T_(AB) INF05008_(A)	INF01202T_(CD)	INF01202T_(AB) INF05008_(A)	INF01202T_(CD)	INF01202P_(A) INF01202P_(D)
(4) 15:30	.	INF01107_(B) INF05008_(B)	.	INF01107_(B) INF05008_(B)	.
(5) 17:30

Etapa 2

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30	MAT01354_(A3)	MAT01355_(A1)	MAT01354_(A3)	MAT01355_(A1)	MAT01354_(A3)
(2) 10:30	INF01108_(B) INF01203_(B)	INF05508_(B) MAT01355_(B2)	INF01108_(B) INF01203_(B)	INF05508_(B) MAT01355_(B2)	.
(3) 13:30	INF01108_(A) INF05508_(A)	INF05512_(A)	INF01108_(A) INF05508_(A)	INF05512_(A)	.
(4) 15:30	INF01108_(C) INF01203_(C)	INF01203_(A) INF05512_(B)	INF01108_(C) INF01203_(C)	INF01203_(A) INF05512_(B)	.
(5) 17:30

Figura 4.17 – Exibição dos quadros de horários alocados para etapas

p1

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30
(2) 10:30	.	Arq0_(B)	.	Arq0_(B)	.
(3) 13:30
(4) 15:30	Arq0_(C)	.	Arq0_(C)	.	.
(5) 17:30

p2

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30
(2) 10:30	.	FCG_(A)	.	FCG_(A)	.
(3) 13:30
(4) 15:30	.	AlgProT_(EF)	.	AlgProT_(EF)	.
(5) 17:30

p3

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30
(2) 10:30
(3) 13:30	AlgProT_(GH)	.	AlgProT_(GH)	.	.
(4) 15:30
(5) 17:30

Figura 4.18 – Exibição dos quadros de horários alocados para professores

Laboratório de Informática SW

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30
(2) 10:30	AlgProP_(B) AlgProP_(C)
(3) 13:30	AlgProP_(A) AlgProP_(D)
(4) 15:30
(5) 17:30

Laboratório de Informática HW

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30	RedesP_(C)
(2) 10:30	RedesP_(D)
(3) 13:30	RedesP_(B)
(4) 15:30	RedesP_(A)
(5) 17:30

Figura 4.19 – Exibição dos quadros de horários alocados para os recursos laboratório de software e laboratório de hardware.

Todas Etapas

HORÁRIO	SEG	TER	QUA	QUI	SEX
(1) 8:30	CalcI (A3) CalcI (A4) CalcII (A3) CD (C) CPD (A) Teoria (BC) FPI (U) FCG (B) SisOpl (B) IHC (B) COMP (A) ProtoComu (U) CG (U)	FundAlg (B) MatDisc (A) Algebra (A1) CPD (B) Formais (A) Prob (K) OrgB (A) OrgB (B) Otimização (B) FTF (A) PDP (U) ConcepCI-I (U)	CalcI (A3) CalcI (A4) CalcII (A3) CD (C) CPD (A) Teoria (BC) FPI (U) FCG (B) SisOpl (B) IHC (B) COMP (A) ProtoComu (U) CG (U)	FundAlg (B) MatDisc (A) Algebra (A1) CPD (B) Formais (A) Prob (K) OrgB (A) OrgB (B) Otimização (B) FTF (A) PDP (U) ConcepCI-I (U)	CalcI (A3) CalcI (A4) CalcII (A3) EMP (U) RedesP (D) Desafios (U) TECXXVI (C)
(2) 10:30	FundAlg (A) CalcI (B2) ArqI (B) TGeAC (B) ArqII (B) CD (A) Complex (A) MLP (B) EngSoft (A) FTF (B) HistComp (U)	Arq0 (B) MatDisc (B) Log (A) CD (E) Prob (F) Complex (B) FCG (A) SisOpl (A) SisOpl (A) RedesT (CD) EnSWII (U) ArqAvanComp (U)	FundAlg (A) CalcI (B2) ArqI (B) TGeAC (B) ArqII (B) CD (A) Complex (A) MLP (B) EngSoft (A) FTF (B) HistComp (U)	Arq0 (B) MatDisc (B) Log (A) CD (E) Prob (F) Complex (B) FCG (A) SisOpl (A) SisOpl (A) RedesT (CD) EnSWII (U) ArqAvanComp (U)	AlgProP (A) AlgProP (D) AlgProP (C) CalcI (B2) EMP (U) RedesP (A) Desafios (U) TECV (U)
(3) 13:30	AlgProT (GH) CalcI (C4) ED (A) CalcII (C2) ArqII (A) CD (F) FBD (U) MLP (A) Otimização (A) IHC (A) SisOpl (B) EspFormN (U)	Arq0 (A) AlgProT (AB) ED (B) Algebra (C2) CD (D) Numerico (A) TCP (A) TCP (B) EngSoft (B) Semantica (A) Fotografia (U) TECXII (U)	AlgProT (GH) CalcI (C4) ED (A) CalcII (C2) ArqII (A) CD (F) FBD (U) MLP (A) Otimização (A) IHC (A) SisOpl (B) EspFormN (U)	Arq0 (A) AlgProT (AB) ED (B) Algebra (C2) CD (D) Numerico (A) TCP (A) TCP (B) EngSoft (B) Semantica (A) Fotografia (U) TECXII (U)	AlgProP (F) CalcI (C4) CalcII (C2) RedesP (B) TECXXVI (B)
(4) 15:30	Arq0 (C) AlgProP (E) FundAlg (C) ArqI (A) Log (B) Teoria (A) Formais (B) Semantica (B) COMP (B) RedesT (AB)	AlgProT (CD) AlgProT (EF) TGeAC (A) TGeAC (C) CD (B) Numerico (B) IA (U) ArDesBD (U) CompEvo (U)	Arq0 (C) FundAlg (C) ArqI (A) Log (B) Teoria (A) Formais (B) Semantica (B) COMP (B) RedesT (AB)	AlgProT (CD) AlgProT (EF) TGeAC (A) TGeAC (C) CD (B) Numerico (B) IA (U) ArDesBD (U) CompEvo (U)	AlgProP (B) AlgProP (C) AlgProP (H) RedesP (C) IntroPesInf (U) TECXXVI (A)
(5) 17:30	CalcI (E1) CalcII (F1) Prob (J)	Algebra (E1) Prob (I)	CalcI (E1) CalcII (F1) Prob (J)	Algebra (E1) Prob (I)	CalcI (E1) CalcII (F1)

Figura 4.20 – Exibição dos quadros de horários alocados que engloba todas etapas

5 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo, é apresentada uma avaliação experimental do modelo matemático proposto na Seção 3.3. O objetivo deste capítulo é responder as seguintes questões:

1. O modelo proposto é apropriado para resolver o problema abordado?
2. Como as soluções obtidas pelo modelo se comparam com as soluções geradas manualmente?

Este capítulo é dividido como segue: Na Seção 5.1 são apresentados dados das instâncias utilizadas nos experimentos. Na Seção 5.2 é descrito o ambiente computacional em que são realizados os experimentos. Por fim, na Seção 5.3 são apresentados os resultados obtidos e a análise dos mesmos.

5.1 Instâncias

Para avaliar a modelagem proposta foi utilizado um conjunto formado por cinco instâncias reais fornecidas pelo departamento do curso de Ciência da Computação e pelo departamento do curso de Engenharia da Computação, ambos pertencentes ao Instituto de Informática da Universidade Federal do Rio Grande do Sul.

As instâncias referem-se aos anos de 2011 (2º semestre), 2013 (2º semestre), 2013 (2º semestre), 2014 (1º semestre) e 2014 (2º semestre). As suas principais características são apresentadas na Tabela 5.1. A primeira coluna apresenta a identificação das instâncias. As colunas 2-7 apresentam, respectivamente, o número de dias, períodos, professores, eventos, eventos fixos, tipos de recursos e etapas. As últimas colunas apresentam, respectivamente, o total de aulas da instância, o número de eventos conflitantes e a quantidade total de recursos disponíveis. Todas as instâncias estão disponibilizadas em <<http://www.inf.ufrgs.br/~fvpneukirchen/tt-ufrgs>>.

A instância ECP2014-2 não possui dados sobre professores ou conflitos entre turmas, pois os dados coletados não continham as informações desejadas para estas características.

Instância	$ D $	$ H $	$ T $	$ E $	$ EF $	$ R $	$ M $	$\sum_{e \in E} W_e$	$\sum_{i,j \in E: i > j} C_{ij}$	$\sum_{r \in R} Q_r$
CIC2011-2	5	5	49	96	14	3	7	180	106	16
CIC2013-2	5	5	55	103	11	3	7	191	69	18
CIC2014-1	5	5	57	114	19	3	7	221	85	18
CIC2014-2	5	5	54	108	14	3	7	202	75	17
ECP2014-2	5	5	-	98	39	3	10	201	-	17

Tabela 5.1 – Principais características das instâncias

5.2 Ambiente Computacional

Os resultados dos experimentos foram obtidos em um computador equipado com um processador Intel® Pentium(R) Dual CPU T3400 de 2.16GHz, com 4GB de memória RAM, executando um sistema operacional Linux de 64 bits. Para resolver as instâncias foi utilizado o resolvidor comercial CPLEX na versão 12.5 (ILOG, 2012) com suas configurações padrão. Para a compilação do Resolvedor foi utilizado o compilador GNU Compiler Collection (GCC) para programas escritos em C++.

5.3 Resultados

Na Tabela 5.2 são apresentadas as soluções geradas manualmente pelos departamentos de Ciência da Computação e Engenharia da Computação da UFRGS em comparação com os resultados obtidos pelo CPLEX utilizando a formulação proposta na Seção 3.3. As colunas obj_m e obj_c apresentam, respectivamente, o valor objetivo da solução manual e da solução obtida pelo CPLEX. Os requisitos fracos não atendidos na solução são apresentados nas colunas $\sum b$, $\sum c$, $\sum q$ e u que informam, respectivamente, a quantidade de períodos ociosos, o total de conflitos entre eventos, o total de eventos excedentes e o total de recursos utilizados. A coluna $tempo$ apresenta o tempo de execução utilizado pelo CPLEX em segundos. A coluna gap_c representa o desvio percentual entre a solução obtida pelo CPLEX e o limitante inferior da solução fornecido pelo CPLEX. A coluna gap_m representa o desvio percentual entre a solução obtida pelo CPLEX e a solução encontrada manualmente. Os valores tabelados nesta coluna são obtidos por $100 * (obj_c - obj_m) / obj_c$. Finalmente, as colunas $\#rest$ e $\#var$ apresentam, respectivamente, o número de restrições e de variáveis do modelo informados pelo CPLEX.

Instância	Soluções Manuais					CPLEX									
	obj _m	$\sum b$	$\sum c$	$\sum q$	u	tempo(s)	obj _c	gap _c (%)	gap _m (%)	#rest	#var	$\sum b$	$\sum c$	$\sum q$	u
CIC2011-2	1239	3	0	9	12	3,51	904	0	-37,1	15790	8406	0	0	4	9
CIC2013-2	5372	3	4	22	13	4,50	905	0	-493,6	14987	7076	0	0	5	9
CIC2014-1	11367	4	10	27	13	5,98	1028	0	-1005,7	16229	7566	2	0	8	10
CIC2014-2	9294	7	8	24	12	5,04	910	0	-921,4	15937	7576	0	0	10	9
ECP2014-2	1066	16	0	6	9	2,84	723	0	-47,5	10464	3891	2	0	3	7

Tabela 5.2 – Resultados comparativos entre as soluções manuais e as soluções obtidas pelo CPLEX

Analisando os resultados da Tabela 5.2, é possível perceber que o CPLEX obteve soluções factíveis ótimas para as cinco instâncias consideradas, utilizando menos de 10 segundos de processamento. Além disso, o custo das soluções obtidas pelo CPLEX são significativamente menores em comparação ao custo das soluções geradas manualmente. Em média, as soluções obtidas pelo método proposto são 501% melhores que a solução manual. Este ganho expressivo, é devido principalmente à redução do número de eventos em posição de conflitos e da redução de períodos ociosos. Pode-se perceber também que houve uma significativa redução no número total de recursos utilizados e eventos excedentes. Observando o gráfico na Figura 5.1 fica evidente que as soluções obtidas via CPLEX satisfazem um maior número de requisitos fracos. Percebe-se que o resolvidor CPLEX prioriza a satisfação de alguns requisitos fracos conforme a importância do requisito (como visto na Seção 3.3.1).

Conforme apresentado na Tabela 5.2, percebe-se que as instâncias apresentam diferentes dimensões, principalmente, em relação ao número de professores e eventos. As instâncias CIC2011-2, CIC2013-2, CIC2014-1, CIC2014-2 e ECP2014-2 estão organizadas em ordem crescente de tamanho. Neste contexto, observa-se que, enquanto a qualidade das soluções obtidas pelo resolvidor não foi influenciada significativamente com o aumento do tamanho das instâncias, a qualidade das soluções obtidas manualmente piorou visivelmente. Este comportamento, no entanto, é esperado, visto que o aumento no tamanho da instância reflete no aumento do espaço de busca, o que torna cada vez mais complexa a criação do horário, em especial, quando realizada por humanos.

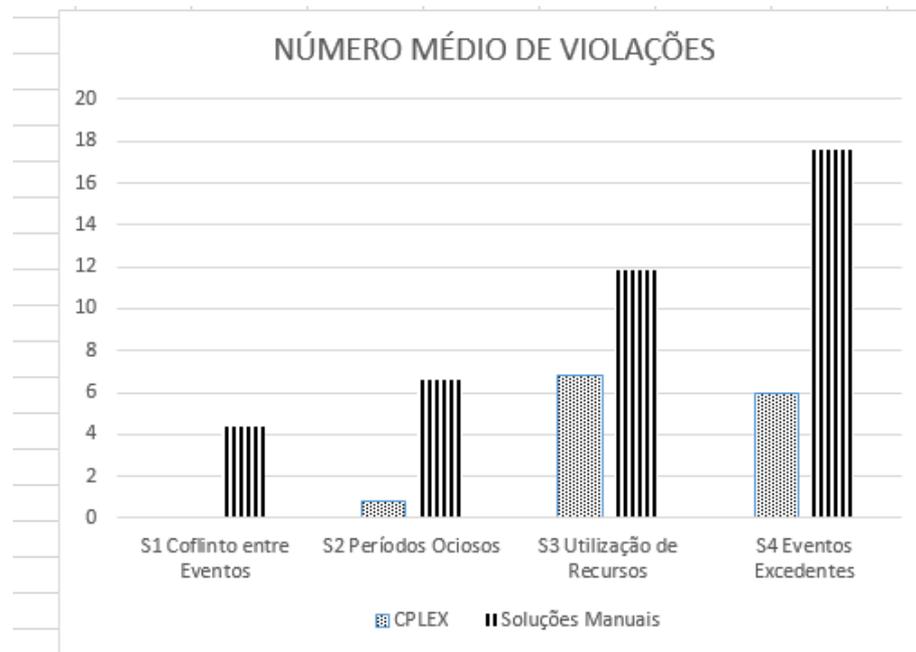


Figura 5.1 – Comparação das violações de requisitos fracos entre soluções manuais e soluções obtidas via CPLEX

6 CONCLUSÕES

Neste trabalho, foi proposto um modelo de programação inteira mista que resolve uma variação do problema de geração de quadros de horários em universidades adaptado ao contexto dos cursos de Ciência da Computação e Engenharia da Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul. O modelo proposto foi avaliado através de cinco instâncias reais utilizando um resolvidor comercial de programação inteira mista. Os resultados experimentais demonstraram que é possível resolver o problema de forma bastante eficiente, sendo possível obter soluções ótimas para todas as instâncias em pouco tempo. Em comparação com soluções geradas manualmente, a abordagem proposta apresenta soluções significativamente melhores. Portanto, esta abordagem poderá ser usada para resolver novas instâncias desse problema, possibilitando a geração automática de quadros de horários para o Instituto de Informática.

Em contato com a Coordenação dos Cursos, verificou-se que, em termos práticos, os resultados obtidos pelo modelo proposto são muito semelhantes aos obtidos manualmente e poderiam perfeitamente ser utilizados em uma divulgação oficial de horários do curso. Todas as soluções geradas pelo método proposto foram avaliadas qualitativamente pela Coordenação dos cursos e não foi encontrada nenhuma grade de horários que não satisfizesse os critérios atualmente utilizados para a geração manual dos horários, o que permite afirmar que os conjuntos de restrições utilizados foram corretamente implementados e que são suficientes para atender os critérios solicitados.

Apesar do modelo de instância desenvolvido no formato YAML conseguir representar as diversas características presentes para resolução do problema de geração de quadros de horários do INF, criar manualmente uma instância nesse formato pode ser uma tarefa complexa, dada a extensão que o arquivo de instância pode atingir. Assim, a utilização da interface gráfica proposta na Seção 4.3 torna-se uma alternativa para simplificar o processo de criação e coleta de instâncias, além de permitir a reutilização dados de instâncias já criadas e a manipulação dos dados do problema.

Existe espaço para expansão dos assuntos tratados neste trabalho, como a modelagem da capacidade de recursos físicos em relação com a quantidade de alunos em uma determinada turma, evitando que turmas com grande quantidade de estudantes sejam alocados em ambientes com pouca capacidade.

Observa-se a possibilidade de trabalhos futuros como a criação de uma ferramenta única que integre as duas funcionalidades implementadas pelas ferramentas vistas na Seções 4.3 e 4.4,

assim simplificando o processo atual. Melhorar a usabilidade da ferramenta, permitindo a interação de professores, para que os mesmos consigam editar as suas preferências de horários. Outra possibilidade seria a manipulação da solução final, permitindo a aplicação de correções e alterações pontuais, conforme critérios particulares da coordenação. Por fim, avaliar as ferramentas desenvolvidas através de testes com usuários, para que estes validem a usabilidade e a qualidade das mesmas.

REFERÊNCIAS

- ABDENNADHER, S.; MARTE, M. University course timetabling using constraint handling rules. **Applied Artificial Intelligence**, Taylor & Francis, v. 14, n. 4, p. 311–325, 2000.
- AVELLA, P.; SALERNO, S.; VASIL, I. A computational study of local search algorithms for Italian high-school timetabling. **Journal of Heuristics**, p. 543–556, 2007.
- AVELLA, P.; VASIL'EV, I. A computational study of a cutting plane algorithm for university course timetabling. **Journal of Scheduling**, Springer, v. 8, n. 6, p. 497–514, 2005.
- BAKIR, M. A.; AKSOP, C. A 0-1 integer programming approach to a university timetabling problem. **Hacettepe Journal of Mathematics and Statistics**, v. 37, n. 1, 2008.
- BARDADYDYM, V. A. Computer-aided school and university timetabling: The new wave. In: **Practice and theory of automated timetabling**. [S.l.]: Springer, 1996. p. 22–45.
- BOIZUMAULT, P.; DELON, Y.; PERIDY, L. Constraint logic programming for examination timetabling. **The Journal of Logic Programming**, Elsevier, v. 26, n. 2, p. 217–233, 1996.
- BURKE, E.; ELLIMAN, D.; WEARE, R. A genetic algorithm based university timetabling system. In: CITESEER. **Proceedings of the 2nd east-west international conference on computer technologies in education**. [S.l.], 1994. v. 1, p. 35–40.
- BURKE, E. et al. Automated university timetabling: The state of the art. **The computer journal**, Br Computer Soc, v. 40, n. 9, p. 565–571, 1997.
- BURKE, E.; KINGSTON, J.; WERRA, D. D. 5.6: Applications to timetabling. **Handbook of graph theory**, CRC Press, p. 445, 2004.
- BURKE, E. K. et al. The State of the Art of Nurse Rostering. **Journal of Scheduling**, v. 7, n. 6, p. 441–499, nov. 2004. ISSN 1094-6136.
- BURKE, E. K.; KENDALL, G.; SOUBEIGA, E. A tabu-search hyperheuristic for timetabling and rostering. **Journal of Heuristics**, Springer, v. 9, n. 6, p. 451–470, 2003.
- BURKE, E. K. et al. Decomposition, reformulation, and diving in university course timetabling. **Computers & Operations Research**, Elsevier, v. 37, n. 3, p. 582–597, 2010.
- BURKE, E. K.; PETROVIC, S. Recent research directions in automated timetabling. **European Journal of Operational Research**, Elsevier, v. 140, n. 2, p. 266–280, 2002.
- CARTER, M. W.; LAPORTE, G.; CHINNECK, J. W. A general examination scheduling system. **Interfaces**, INFORMS, v. 24, n. 3, p. 109–120, 1994.
- CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. **ACM Transactions on Database Systems (TODS)**, ACM, v. 1, n. 1, p. 9–36, 1976.
- COOPER, T. B.; KINGSTON, J. H. **The complexity of timetable construction problems**. [S.l.]: Springer, 1996.
- CPLEX. 2013. <http://www-304.ibm.com/ibm/university/academic/pub/page/academic_initiative>. Online; Acessado em: 20-Maio-2015.

- DAMMAK, A. et al. Course timetabling at a tunisian university: A case study. **Journal of Systems Science and Systems Engineering**, Springer, v. 17, n. 3, p. 334–352, 2008.
- DASKALAKI, S.; BIRBAS, T.; HOUSOS, E. An integer programming formulation for a case study in university timetabling. **European Journal of Operational Research**, Elsevier, v. 153, n. 1, p. 117–135, 2004.
- DERIS, S. et al. University timetabling by constraint-based reasoning: A case study. **Journal of the Operational Research Society**, JSTOR, p. 1178–1190, 1997.
- DIMOPOULOU, M.; MILIOTIS, P. Implementation of a university course and examination timetabling system. **European Journal of Operational Research**, Elsevier, v. 130, n. 1, p. 202–213, 2001.
- DORNELES, Á. P.; ARAÚJO, O. C. B.; BURIOL, L. S. The impact of compactness requirements on the resolution of high school timetabling problem. In: **Proceedings of XLIV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2012)**. [S.l.: s.n.], 2012. p. 3336–3347.
- EIKELDER, H. M. ten; WILLEMEN, R. Some complexity aspects of secondary school timetabling problems. In: **Practice and Theory of Automated Timetabling III**. [S.l.]: Springer, 2001. p. 18–27.
- ERBEN, W.; KEPLER, J. A genetic algorithm solving a weekly course-timetabling problem. In: **Practice and Theory of Automated Timetabling**. [S.l.]: Springer, 1996. p. 198–211.
- ERIKSSON, M.; HALLBERG, V. Comparison between json and yaml for data serialization. **Bachelor's thesis**, 2011.
- GLPK. 2009. <<ftp://ftp.gnu.org/gnu/glpk/>>. Online; Acessado em: 04-Junho-2015.
- GOLTZ, H.-J.; MATZKE, D. University timetabling using constraint logic programming. In: **Practical aspects of declarative languages**. [S.l.]: Springer, 1998. p. 320–334.
- GOTLIEB, C. The construction of class-teacher timetables. In: **IFIP**. Amsterdam: North-Holland, 1963. p. 73–77.
- GUÉRET, C. et al. Building university timetables using constraint logic programming. In: **Practice and Theory of Automated Timetabling**. [S.l.]: Springer, 1996. p. 130–145.
- ILOG, I. **CPLEX 12.5 reference manual and software**. [S.l.]: Sunnyvale: IBM, 2012.
- JOHNSON, E. L.; NEMHAUSER, G. L.; SAVELSBERGH, M. W. Progress in linear programming-based algorithms for integer programming: An exposition. **INFORMS Journal on Computing**, INFORMS, v. 12, n. 1, p. 2–23, 2000.
- LAJOS, G. Complete university modular timetabling using constraint logic programming. In: **Practice and Theory of Automated Timetabling**. [S.l.]: Springer, 1996. p. 146–161.
- LÜ, Z.; HAO, J.-K. Adaptive tabu search for course timetabling. **European Journal of Operational Research**, Elsevier, v. 200, n. 1, p. 235–244, 2010.
- MAKHORIN, A. **GLPK (GNU linear programming kit)**. 2008.

MCCOLLUM, B. A perspective on bridging the gap between theory and practice in university timetabling. In: **Practice and theory of automated timetabling VI**. [S.l.]: Springer, 2007. p. 3–23.

MCCOLLUM, B.; IRELAND, N. University timetabling: Bridging the gap between research and practice. **E Burke, HR, ed.: PATAT**, p. 15–35, 2006.

MIRHASSANI, S. A computational approach to enhancing course timetabling with integer programming. **Applied Mathematics and Computation**, Elsevier, v. 175, n. 1, p. 814–822, 2006.

PETROVIC, S.; BURKE, E. K. University timetabling. In: **Handbook of scheduling: algorithms, models, and performance analysis**. Nottingham: UK, 2004. p. 1–23.

RICH, D. C. **A smart genetic algorithm for university timetabling**. [S.l.]: Springer, 1996.

SCHAERF, A. A survey of automated timetabling. **Artificial Intelligence Review**, Springer, v. 13, n. 2, p. 87–127, 1999.

SCHIMMELPFENG, K.; HELBER, S. Application of a real-world university-course timetabling model solved by integer programming. **Or Spectrum**, Springer, v. 29, n. 4, p. 783–803, 2007.

WEARE, R.; BURKE, E.; ELLIMAN, D. A hybrid genetic algorithm for highly constrained timetabling problems. In: **Proceedings of the Sixth International Conference on Genetic Algorithms**, ed. LJ Eshelman. [S.l.: s.n.], 1995. p. 605–610.

WREN, A. Scheduling, timetabling and rostering—a special relationship? In: **Practice and theory of automated timetabling**. [S.l.]: Springer, 1996. p. 46–75.

YAML. 2009. <<http://yaml-cpp.googlecode.com/files/yaml-cpp-0.5.1.tar.gz>>. Online; Acessado em: 10-Junho-2015.