
**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Curso Sistemas de Informação**



UFOP
Universidade Federal
de Ouro Preto

**Software Web para Problemas de
Agendamento de Horários
Modelados em XHSTT**

Thaise Daiara Delfino

**TRABALHO DE
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:
Prof. Msc. George Henrique Godim da Fonseca

**Outubro, 2014
João Monlevade/MG**

Thaise Daiara Delfino

**Software Web para Problemas de Agendamento
de Horários Modelados em XHSTT**

Orientador: Prof. Msc. George Henrique Godim da
Fonseca

Monografia apresentada ao Curso de
Sistemas de Informação do Instituto de
Ciências Exatas e Aplicadas, como requisito
parcial para aprovação na Disciplina
Trabalho de Conclusão de Curso II.

Universidade Federal de Ouro Preto
João Monlevade
Outubro de 2014

Software Web para Problemas de Agendamento de Horários Modelados em XHSTT

Thaise Daiara Delfino
Universidade Federal de Ouro Preto

Orientador: Prof. Msc. George Henrique Godim da Fonseca



Curso Sistemas de Informação

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Software Web para Problemas de Agendamento de Horários Modelados em XHSTT

Thaise Daiara Delfino

Monografia apresentada ao Departamento de Computação e Sistemas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CEA499 – Trabalho de Conclusão de Curso II, do curso de Bacharelado em Sistemas de Informação, e aprovada pela Banca Examinadora abaixo assinada:

Prof. Me. George Henrique Godim da Fonseca
Mestre em Ciência da Computação - UFOP
Orientador
Departamento de Computação e Sistemas - UFOP

Prof^a. Me. Elton Máximo Cardoso
Mestre em Ciência da Computação - UFMG
Examinador
Departamento de Computação e Sistemas - UFOP

Prof. Me. Tiago França Melo de Lima
Mestre em Análise e Modelagem de Sistemas Ambientais - UFMG
Examinador
Departamento de Computação e Sistemas - UFOP

Prof. Matheus Guedes Vilas Boas
Bacharel em Sistemas de Informação - UFOP
Examinador
Departamento de Computação e Sistemas - UFOP

João Monlevade, 10 de outubro de 2014

TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado “**Software Web para Problemas de Agendamento de Horários Modelados em XHSTT**” é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos referidos autores.

João Monlevade, 10 de Outubro de 2014

Thaise Daiara Delfino

Dedico carinhosamente este trabalho aos meus pais Maria Aparecida e Antônio, aos meus irmãos Thássia e Thales, e à minha tia Lúcia da Luz Aparecida da Silva (in memoriam) e avó Maria Celeste Delfino (in memoriam).

Software Web para Problemas de Agendamento de Horários Modelados em XHSTT

Resumo

O Problema do Agendamento de Horários pertence à categoria dos problemas de otimização difíceis de resolver. Esse problema consiste na alocação de espaços de tempo e recursos a um conjunto de eventos, com o objetivo de chegar a uma solução eficiente. A *Third International Timetabling Competition* (ITC2011) empregou o XHSTT como formato padrão para a modelagem do problema, e estimulou o desenvolvimento de resolvedores eficientes para essa categoria. Porém, esses resolvedores não possuem interface gráfica, o que torna seu uso restrito a pesquisadores da área, além de sua especificação manual ser uma tarefa complexa e trabalhosa. O presente trabalho apresenta um Software Web capaz de especificar problemas de agendamento de horários no formato XHSTT, e através do resolvedor vencedor da competição chegar numa solução eficiente. O software auxilia a criação e a disponibilização de novas instâncias, e permite que usuários comuns possam utilizá-lo. Como teste, desenvolveu-se e disponibilizou-se uma instância da Universidade Federal de Ouro Preto para a literatura. Como resultado, foi obtida uma solução excelente, e como trabalhos futuros pode-se explorar a usabilidade, além estudos de caso de aplicação do software.

Web-software to Timetabling Problems Modeled in XHSTT

Abstract

The High School Timetabling Problem is considered a hard problem in the context of combinatorial optimization. This problem consists in the assignment of timeslots and resources to a set of events, with the goal of reaching an efficient solution. The Third International Timetabling Competition (ITC2011) used the XHSTT format to model instances of this problem and stimulated the development of efficient solvers for this category of problems. However, these solvers have no graphical interface, which makes their use restricted to researchers in the field, furthermore, its manual specification is a complex and laborious task. The software supports the creation and provision of new instances, and enables ordinary users to use it. As a test, we developed an instance of the Federal University of Ouro Preto with the aid of this software. As result, an optimal solution was obtained. For future work we suggest explore the usability of this software and its application in case studies.

Agradecimentos

Em primeiro lugar, agradeço a *Deus* por essa grande conquista. Ele é quem me capacitou para a realização deste grande sonho, e esteve comigo em todos os momentos dando-me forças para não desistir e seguir em frente.

Agradeço a minha mãe *Cida*, por todo carinho, motivação e apoio durante todos esses anos de muito estudo e dedicação. A meu pai *Toninho*, também pelo apoio e por todos os dias que abriu mão de seu tempo para estar ao meu lado rumo à universidade.

Aos meus irmãos *Thássia* e *Thales*, sou grata pelo apoio, pela paciência, por toda ajuda e força que me deram durante esse período.

Agradeço ao professor *Theo Silva Lins* que dedicou gentilmente seu tempo para auxiliar na hospedagem online do software.

E por fim, agradeço a meu orientador *George Fonseca*, pela excelente e dedicada orientação, pela confiança que fora depositada em mim, por toda atenção e ajuda concedida durante o desenvolvimento deste trabalho.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Nomenclatura	1
1 Introdução	2
1.1 Objetivos	3
1.2 Organização do Trabalho	3
2 Problema do Agendamento de Horários	4
2.1 Formato XHSTT	5
3 Metodologias Utilizadas	9
3.1 Tecnologias	10
3.1.1 HTML 4	10
3.1.2 XML	10
3.1.3 PHP 5	11
4 Software Desenvolvido	13
4.1 Interface Gráfica	13

4.2	Resolvedor Incorporado	19
4.3	Instância Proposta	20
5	Considerações Finais	22
A	Apêndices	23
A.1	Publicações	23
	Referências Bibliográficas	24

Lista de Figuras

2.1	Exemplo de alto nível de arquivo especificada no formato XHSTT	6
3.1	Exemplo de código HTML	10
3.2	Exemplo de marcações XML	11
3.3	Exemplo de código PHP	12
4.1	Interface de Login e Formulário de Cadastro de Usuário	14
4.2	Interface principal e tags XHSTT geradas quando da criação de nova instância	14
4.3	Criação da entidade Times.	15
4.4	Criação da entidade Resource.	15
4.5	Criação da entidade Event.	16
4.6	Criação da restrição.	16
4.7	Interface para solução da instância.	17
4.8	Solução da instância.	18
4.9	Exemplo de tabela de horários gerada pelo Timetabler.	19

Lista de Tabelas

4.1	Características da Instância <i>Brazil_UFOP_2014_2.xml</i>	21
-----	--	----

Nomenclatura

ITC2011 *Third International Timetabling Competition*

VNS *Variable Neighborhood Search*

XHSTT *eXtensible High School TimeTabling*

XML *eXtensible Markup Language*

HTML *HyperText Markup Language*

SGML *Standard Generalized Markup Language*

WWW *World Wide Web*

W3C *World Wide Web Consortium*

PHP *HyperText Preprocessor*

Capítulo 1

Introdução

O Problema de Agendamento de Horários, que é alvo de diversas pesquisas na área de Inteligência Artificial e Pesquisa Operacional, consiste na alocação de horários e recursos com o objetivo de chegar numa solução eficiente para essa categoria de problemas (Fonseca 2013b).

Na *Third International Timetabling Competition* (ITC2011), foi considerado um formato de instância que permite a modelagem desse tipo problema, o XHSTT. A competição motivou o desenvolvimento de diversas soluções de otimização eficientes para problemas dessa classe (of Twente 2012).

Em contrapartida, nenhum dos resolvedores desenvolvidos utiliza alguma forma de interação gráfica com o usuário, o que dificulta e até mesmo impossibilita que essas soluções possam ser utilizadas por usuários comuns. Ademais, a especificação manual do formato XHSTT é complexa e trabalhosa até mesmo para pesquisadores da área, pois o arquivo é extenso e muito detalhado. Conforme (of Twente 2011), pelos anos de discussão que levou à formação abstrata desse formato, o mesmo é difícil de entender. Por esses motivos, surge a necessidade de um aplicativo que permita trabalhar com esse formato.

O presente trabalho propõe um Software Web para manusear e resolver problemas de agendamento de horários modelados em XHSTT, além de permitir que usuários comuns possam utilizá-lo no gerenciamento de seus recursos. O software proposto conta com o resolvedor vencedor da ITC2011 (Fonseca, Santos, Toffolo, Brito and Souza 2014).

O formato de arquivo é baseado em XML (eXtensible Markup Language), criado com

o objetivo de unificar as representações de problemas de agendamento de horários em um formato de entrada/saída único. Assim, é possível que uma solução de otimização que trata esse formato de arquivo possa ser aplicada a diversos outros problemas do gênero, desde que especificadas de acordo com o formato em questão.

1.1 Objetivos

O principal objetivo do presente trabalho é desenvolver um Software Web que abstraia o usuário do conhecimento do formato XHSTT de modo que qualquer interessado - pesquisador ou usuário comum - seja apto a especificar e resolver problemas de agendamento de horários.

Esse software irá auxiliar ainda na popularização do formato XHSTT para especificação de problemas de agendamento, visto que o formato é genérico e pode ser aplicado a uma vasta gama de problemas reais. Adicionalmente, novas instâncias poderão ser criadas através do software e enriquecer a base de dados de problemas *benchmark* para resolvedores de agendamento de horários que lidam com o formato. Nesse sentido, o presente trabalho introduz uma instância do problema na literatura, geradas com o auxílio do software.

1.2 Organização do Trabalho

O restante do trabalho está organizado da seguinte maneira:

- Capítulo 2: disserta sobre o problema do agendamento de horários, apresenta o formato XHSTT para especificação de problemas dessa classe;
- Capítulo 3: apresenta a metodologia utilizada para desenvolver o software;
- Capítulo 4: apresenta o Software Web desenvolvido bem como o resolvidor incorporado e a instância sob a qual foi testado;
- Capítulo 5: apresenta as considerações finais sobre o trabalho e sugestões de trabalhos futuros.

Capítulo 2

Problema do Agendamento de Horários

O Problema do Agendamento de Horários consiste, em alocar horários e recursos para a realização de tarefas em um dado horizonte de planejamento. As restrições podem ser fortes (essenciais) ou fracas (desejáveis). O objetivo é encontrar a melhor solução conforme as restrições aplicadas.

Nas instituições de ensino, o problema em questão se refere a alocação dos horários das disciplinas em cada dia da semana com seus respectivos recursos (professores, classes e salas). Por isso, o agendamento de horários é uma tarefa complexa cuja solução adequada pode melhorar a satisfação dos funcionários bem como tornar as organizações mais eficientes na gestão de seus recursos.

Por se tratar ainda de um problema da classe \mathcal{NP} -Difícil (Fonseca 2013b), a programação automática de horários tem sido alvo de diversas pesquisas nas áreas de Pesquisa Operacional e Inteligência Artificial. Diversos problemas se enquadram nessa categoria, como o Problema do Agendamento de Enfermeiros (Santos, Toffolo, Ribas and Gomes 2012), o Problema da Programação de Horários Escolares (Santos, Uchoa, Ochi and Maculan 2012) e o Problema do Agendamento de Tripulações (Jutte and Thonemann 2012).

Atualmente, existem softwares populares que trabalham com o agendamento de horários na plataforma desktop, utilizando algoritmos eficientes para problemas fáceis e difíceis. Como exemplo, tem-se o Mimosa (Mimosa 2014), o ascTimeTables (AsCTimetables 2014), e o FET (Fet 2014). Sendo este último gratuito e, o algoritmo utilizado é uma heurística nomeada "*Recursive Swapping*" (Hartog 2007, Marte 2002).

Na ITC2011 (Terceira Competição Internacional de Agendamento de Horários), para a modelagem de problemas difíceis do mundo real (of Twente 2012), adotou-se um formato de arquivo como padrão, baseado na linguagem XML, o XHSTT (eXtensible High School TimeTabling), para a criação de instâncias interpretadas pelos resolvedores para chegar a uma solução.

2.1 Formato XHSTT

O formato XHSTT foi criado por (Post 2011) e consiste de um formato de arquivo baseado no XML (eXtensible Markup Language), com o objetivo de unificar as representações de problemas de agendamento de horários em um formato de entrada/saída único. Assim, é possível que uma solução de otimização que trata esse formato de arquivo possa ser aplicada a diversos outros problemas do gênero, desde que especificadas de acordo com esse formato.

O código da Figura 2.1 apresenta um exemplo em alto nível de arquivo no formato XHSTT. Cada instância é dividida em quatro entidades principais: Tempos, Recursos, Eventos e Restrições. Os tópicos a seguir expandem a descrição sobre essas entidades. Para mais detalhes sobre a especificação de instâncias e soluções no formato XHSTT, veja (Post 2011) e (Kingston 2012).

- **Tempos:** A entidade Tempos (*Times*) consiste de um espaço de tempo (*Time*) ou de um conjunto de espaços de tempo (*TimeGroup*) que estão disponibilizados para a realização das alocações. Comumente cada *Time* representa um slot de tempo para alocação e cada *TimeGroup* agrupa entidades *Time* que pertencem ao mesmo dia da semana.
- **Recursos:** A entidade Recursos (*Resources*) consiste de um recurso em específico (*Resource*), de um grupo de recursos (*ResourceGroup*) ou de um tipo de recurso (*ResourceType*) e expressa informações sobre os recursos disponíveis para alocação. Cada recurso possui os seguintes atributos:
 - **Name.** Nome do recurso, usado também como identificador (ID) no presente software;
 - **ResourceType.** Representa a categoria à qual o recurso pertence, por exemplo: Professor, Classe e Sala;

```

<HighSchoolTimetableArchive>
  <Instances>
    <Instance Id="Sudoku4x4">
      <Times>
        ...
      </Times>
      <Resources>
        ...
      </Resources>
      <Events>
        ...
      </Events>
      <Constraints>
        ...
      </Constraints>
    </Instance>
  </Instances>
  <SolutionGroups>
    ...
  </SolutionGroups>
</HighSchoolTimetableArchive>

```

Figura 2.1: Exemplo de alto nível de arquivo especificada no formato XHSTT

- **ResourceGroups.** Especifica os grupos de recursos aos quais o recurso pertence, por exemplo: Turmas do curso de Sistemas de Informação;
- **Eventos:** Um evento (*Event*) é a unidade básica de alocação, representa uma aula simples ou um conjunto de aulas, gerando um grupo de eventos (*EventGroup*). Um evento possui os seguintes atributos:
 - **Name.** Nome do evento, usado também como identificador (ID) no presente software;
 - **Duration.** Representa o número de espaços de tempo que têm que ser alocados ao evento;
 - **Resources.** Especifica os recursos que deverão se alocados ao evento;
 - **Workload.** Carga de trabalho que irá ser adicionada à carga de trabalho total dos recursos alocados ao evento (opcional);
- **Restrições:** (Post 2010) agrupa as restrições em três grupos: restrições básicas do problema de agendamento, restrições para os eventos e restrições para os recursos. A função objetivo (função matemática que define a qualidade da solução)

calcula a violação de cada uma das restrições, que é penalizada de acordo com seu peso. Cada instância pode ainda definir se uma restrição específica é forte (obrigatória) ou fraca (desejável) em seu contexto. A seguir, apresentamos cada um dos conjuntos de restrições.

Restrições Básicas de Agendamento

1. ASSIGN TIME. Alocar espaços de tempo a cada evento;
2. ASSIGN RESOURCE. Alocar os recursos a cada evento;
3. PREFER TIMES. Indica que determinado evento tem preferência por determinado(s) timeslot(s);
4. PREFER RESOURCES. Indica que determinado evento tem preferência por determinado(s) recurso(s).

Restrições para os Eventos

1. LINK EVENTS. Agendar os grupos de eventos no mesmo tempo de início;
2. SPREAD EVENTS. Agendar os eventos de cada grupo de eventos para cada grupo de tempo entre um número mínimo e máximo de vezes. Essa restrição pode ser usada, por exemplo, para definir um limite diário de aulas;
3. AVOID SPLIT ASSIGNMENTS. Para cada grupo de eventos, alocar um determinado recurso a todos os eventos do grupo. Com essa restrição, pode-se forçar que, por exemplo, todos as alocações de um evento ocorram na mesma sala;
4. DISTRIBUTE SPLIT EVENTS. Para cada evento, alocar entre um número mínimo e máximo de aulas consecutivas de uma duração dada. Essa restrição se justifica pois um número grande de aulas consecutivas do mesmo assunto pode prejudicar o aprendizado;
5. SPLIT EVENTS. Limita o número de aulas não consecutivos em que um evento será agendado e sua duração. Um exemplo de aplicação dessa restrição é forçar que um dado evento com quatro aulas seja alocado em dois blocos de aulas pareadas, uma vez que uma aula em separado não seria suficiente para se dar andamento ao conteúdo.

Restrições para os Recursos

1. **AVOID CLASHES.** Agendar os recursos sem conflitos, ou seja, sem alocar o mesmo recurso a mais de um evento ao mesmo tempo;
2. **AVOID UNAVAILABLE TIMES.** Evitar alocar os recursos nos horários em que estão indisponíveis. Com essa restrição é possível, por exemplo, evitar que uma sala seja alocada num horário em que está reservada para limpeza ou evitar que um professor seja alocado num horário em que participa de outro evento; ou horário indisponível
3. **LIMIT WORKLOAD.** A cada evento está associada uma carga de trabalho. Essa carga é somada à carga de trabalho do recurso que estiver a ele designado. Essa restrição objetiva manter a carga de trabalho dos recursos entre um limite mínimo e máximo. Essa restrição pode ser utilizada, por exemplo, para limitar o número de disciplinas que um professor irá lecionar;
4. **LIMIT IDLE TIMES.** O número de horários ociosos em cada grupo de espaços de tempo deve estar entre um limite mínimo e máximo para cada recurso selecionado. Tipicamente, um grupo de espaços de tempo consiste dos *timeslots* de um dia da semana. Essa restrição é utilizada para evitar horários sem atividade entre horários ativos na agenda de cada recurso;
5. **LIMIT BUSY TIMES.** O número de horários ocupados em cada grupo de espaços de tempo deve estar entre um limite mínimo e máximo para cada recurso selecionado. Um número grande de alocações num mesmo dia pode prejudicar o desempenho dos estudantes e dos professores, o que justifica o uso dessa restrição;
6. **CLUSTER BUSY TIMES.** O número de grupos de tempo com um *timeslot* alocado a um recurso deve figurar entre um limite mínimo e máximo. Tipicamente, o grupo de tempo são dias e, por exemplo, um professor requer no máximo três dias com aulas.

Capítulo 3

Metodologias Utilizadas

O projeto deu início com base no estudo dos problemas de Agendamento de Horários, sua aplicação em problemas da vida real, funcionamento e objetivo dessa classe de problemas, e emprego do formato XHSTT como um padrão estabelecido para essa categoria na ITC2011. A indisponibilidade de uma interface gráfica que permitisse a utilização do formato por usuários comuns motivou o desenvolvimento do software, que utiliza o resolvidor vencedor da competição internacional.

Posteriormente, foi feita uma análise de forma detalhada em relação à estrutura e o relacionamento entre as entidades no arquivo XHSTT através de instâncias disponibilizadas no site da ITC2011, no link <http://www.utwente.nl/ctit/hstt/archives/XHSTT-2011/>, para a abstração de ideias. Para desenvolver uma interface coerente com o formato, o HTML (*Hypertext Markup Language*) foi utilizado na estruturação da página.

A linguagem de alto nível utilizada para desenvolver o software deste trabalho foi PHP (*Hypertext Preprocessor*) versão 5.3.8, que permitiu a manipulação do XML (*eXtensible Markup Language*) através da biblioteca DOM. As principais operações executadas foram a criação de arquivos, edição, remoção, exclusão, além da consistência dos dados armazenados. Pela complexidade do arquivo XHSTT, muitos testes foram feitos a partir de cada instância gerada pelo software, para ajustar possíveis incompatibilidades do arquivo com o resolvidor. Esses testes foram feitos em computadores com sistema operacional Linux antes da hospedagem nos servidores da UFOP (Universidade Federal de Ouro Preto).

3.1 Tecnologias

Neste tópico serão descritas as tecnologias utilizadas.

3.1.1 HTML 4

HTML 4 (*Hypertext Markup Language* - Linguagem de Formatação de Hipertexto) é uma aplicação SGML (*Standard Generalized Markup Language*) e está mundialmente reconhecida como a linguagem padrão para publicação *World Wide Web* (WWW). SGML é uma linguagem para descrever linguagens de marcação utilizadas em gestão e publicação eletrônica de documentos. O HTML é um exemplo de linguagem definida em SGML (Gonçalves 2007).

A concepção original da linguagem HTML, conforme (Gonçalves 2007), foi destinada ao tráfego de documentos de caráter científico e técnico, adaptada para ser utilizada por não especialistas naqueles tipos de documentos. HTML especifica um conjunto de tags com características estruturais e semânticas adaptadas para o gerenciamento de documentos simples. Posteriormente, foram acrescentados suporte para HiperTexto e facilidades de multimídia. Na Figura 3.1 é mostrado um exemplo de código.

```
<html>
<head>
<title>Documento sem título</title>
</head>
<body>
<h1>Cabeçalho</h1>
<h2>Cabeçalho médio</h2>
<p>Parágrafo</p>
<hr>
  Isto é um link:
  <a href="http://200.239.152.5/timetabler/index.htm">
Abrir Aplicativo Timetabler</a>
</body>
</html>
```

Figura 3.1: Exemplo de código HTML

3.1.2 XML

XML (*eXtensible Markup Language*) é uma linguagem de marcação desenvolvida pelo W3C (*World Wide Web Consortium*) que ultrapassa as limitações do HTML, permitindo

que novas tipologias de aplicações Web sejam criadas. O armazenamento de dados no XML é padronizado em formato de texto simples (Medina 2014).

Conforme (Almeida 2002), o XML é utilizado para gerenciar melhor a grande quantidade de informações disponíveis na Internet. As principais características da linguagem é permitir ao autor do documento a definição de suas próprias marcas, e a flexibilidade às aplicações Web. Temos um exemplo a seguir de código XML na Figura 3.2.

```
▼<HighSchoolTimetableArchive>
  ▼<Instances>
    ▼<Instance Id="instances">
      ▼<MetaData>
        <Name>thaisedd</Name>
        <Contributor>Generated by Timetabler</Contributor>
        <Date>Aug 02,2014 16:54:34</Date>
        <Country>Brazil</Country>
        <Description>Test instance</Description>
        <Remarks/>
      </MetaData>
      ▶<Times>...</Times>
      ▶<Resources>...</Resources>
      ▶<Events>...</Events>
      <Constraints/>
    </Instance>
  </Instances>
</HighSchoolTimetableArchive>
```

Figura 3.2: Exemplo de marcações XML

3.1.3 PHP 5

PHP 5 (*Hypertext Preprocessor*) é uma linguagem híbrida pela sua sintaxe possuir características de outras linguagens como C, Shell, Perl e até mesmo Java. É baseada em sequência de comandos (script), e muito utilizada em servidores Web (Gutmans, Bakken and Rethans 2005).

Segundo (Gutmans, Bakken and Rethans 2005), o PHP permite aos desenvolvedores irem além de aplicações HTML simples. Suporta metodologias orientadas a objetos, atualmente é a melhor linguagem disponível para processar XML, e é a plataforma ideal para criar e usar serviços Web. Um exemplo de uso da linguagem é apresentado na Figura 3.3.

```
<?php

$userHidden = $_POST['userHidden'];
$dom = new DOMDocument('1.0','UTF-8');

$dom->preserveWhiteSpaces = false;
$dom->formatOutput = true;

$HighSchoolTimetableArchive =
$dom->createElement('HighSchoolTimetableArchive');
.
.
$dom->saveXML();
$dom->save('xmlfiles/'.$userHidden.'.xml');

?>
```

Figura 3.3: Exemplo de código PHP

Capítulo 4

Software Desenvolvido

O objetivo do software é permitir a modelagem do formato XHSTT de maneira automatizada através da entrada de dados especificados pelo o usuário. A análise de requisitos envolve o cadastramento de recursos (professor, classe e sala), eventos e restrições no arquivo XHSTT. Por conseguinte, encaminhar a instância gerada ao resolvidor para encontrar uma solução para o problema de agendamento de horários de uma determinada instituição.

O desenvolvimento da interface gráfica objetivou encontrar um ponto de equilíbrio entre a facilidade de uso, aprendizado, e a similaridade com o formato XHSTT. Para tal, nossa implementação se abstraiu de algumas características do referido formato, representando através de formulários as entidades tempos, recursos, eventos e restrições. O software encontra-se disponível através do link <http://200.239.152.5/timetable/>. Convidamos o leitor interessado a testar o software e apresentar sugestões de melhoria.

4.1 Interface Gráfica

O sistema possui Login e permite a criação de cadastro de usuários, como apresenta a Figura 4.1.

A Figura 4.2 apresenta a interface principal (Home) do software. À esquerda da tela tem-se instruções sobre o software, e à direita a descrição do arquivo gerado. É possível a criação de um novo arquivo no formato XHSTT, além das funcionalidades de importação e exportação de arquivo.

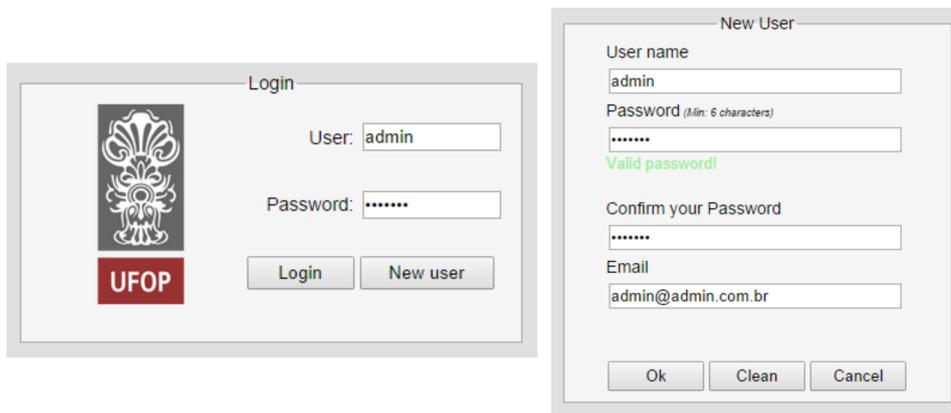


Figura 4.1: Interface de Login e Formulário de Cadastro de Usuário

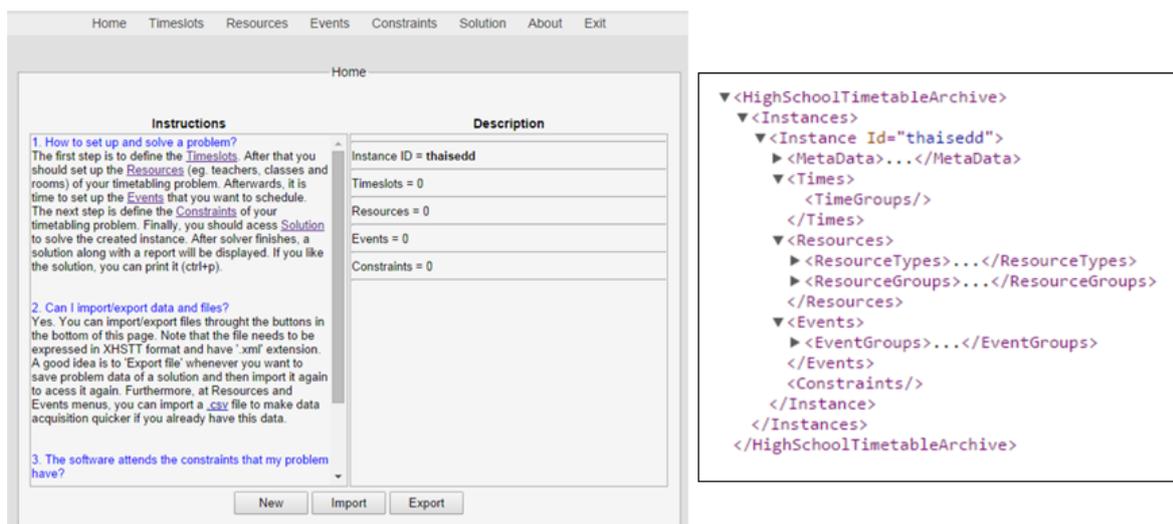


Figura 4.2: Interface principal e tags XHSTT geradas quando da criação de nova instância

Note que a criação de algumas entidades foi automatizada de modo a tornar mais fácil o uso do software. Como exemplo, note que na Figura 4.2, que os elementos TimeGroups, ResourceGroups e EventGroups são criados automaticamente.

A Figura 4.3 apresenta a criação da entidade Times através do presente software.

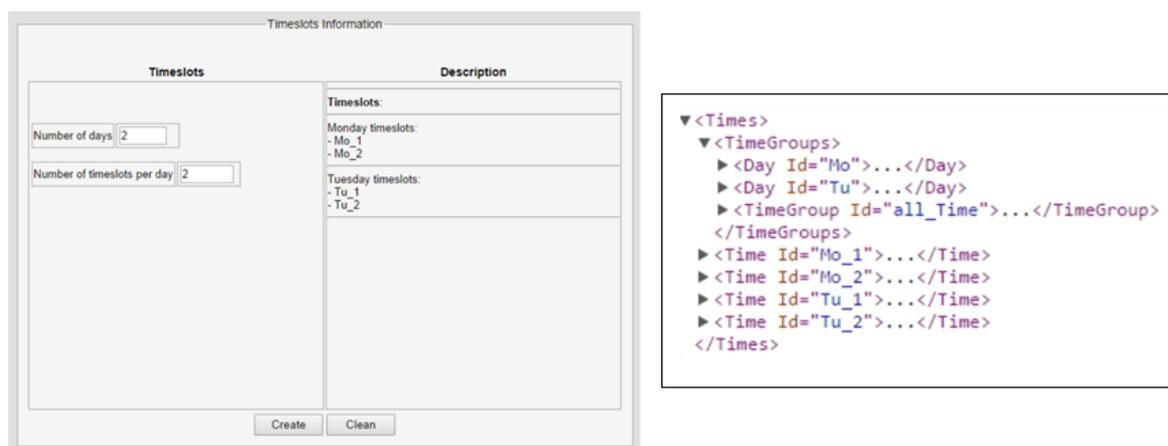


Figura 4.3: Criação da entidade Times.

O cadastro de recursos através do Timetabler é trivial. Necessita-se apenas do nome do recurso como identificador. No presente trabalho considerou-se apenas três tipos possíveis de recurso: Classe, Professor e Sala. Para criar qualquer tipo de recurso, utiliza-se a interface apresentada na Figura 4.4.

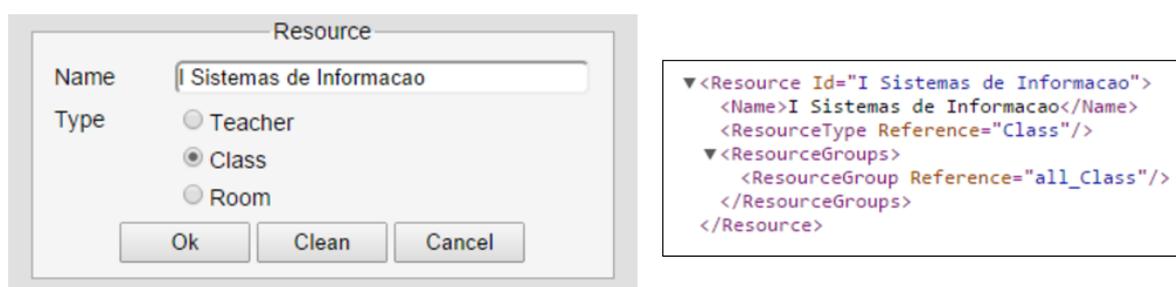


Figura 4.4: Criação da entidade Resource.

A Figura 4.5 apresenta a criação da entidade Event através do presente software.

Para tratar a criação de distintas restrições previstas pela entidade Constraints, criou-se uma janela para cada restrição prevista no formato XHSTT. A Figura 4.6 apresenta a criação automatizada da entidade Constraint através do presente software. Especificamente, foi criada uma restrição do tipo Avoid Unavailable Times. A restrição criada foi

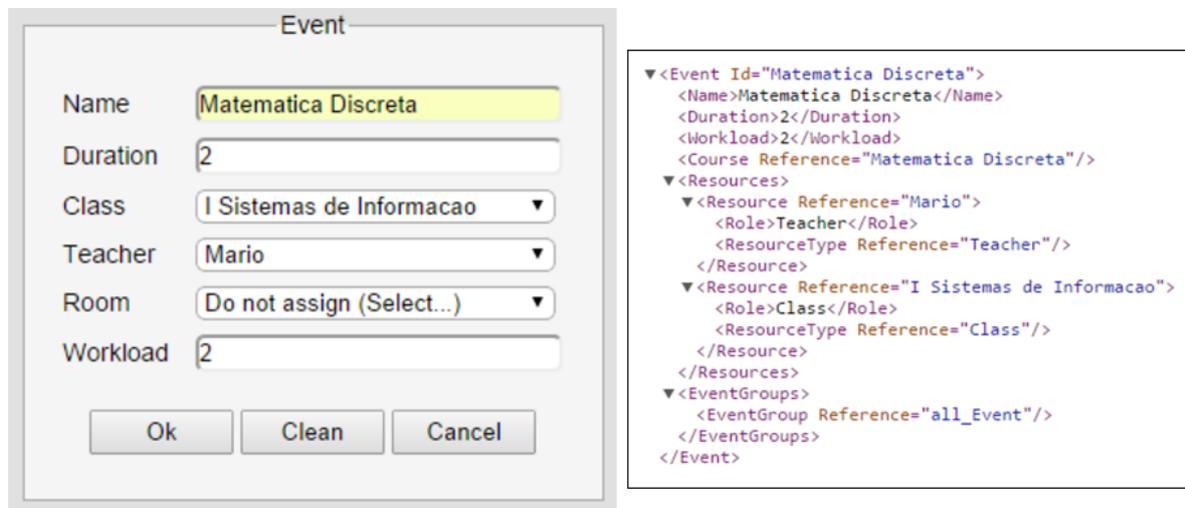


Figura 4.5: Criação da entidade Event.

uma restrição forte (de atendimento obrigatório) com peso 1 a ser adicionado na função objetivo caso violada que se aplica ao recurso Eduardo dizendo que o mesmo não está disponível nos timeslots Mo_1 e Mo_2.

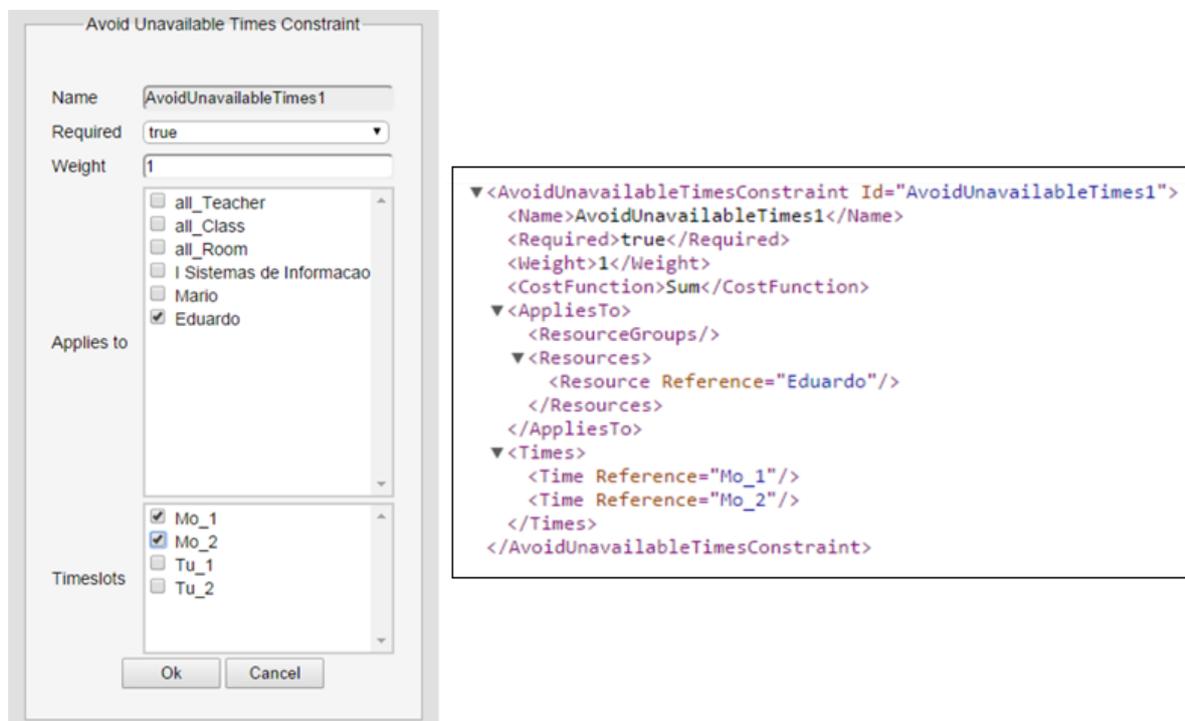


Figura 4.6: Criação da restrição.

Após definidos os espaços de tempo disponíveis, os recursos, os eventos a serem

alocados e as restrições a serem consideradas, o usuário deve submeter a instância a um resolvidor para obter uma solução. A Figura 4.7 apresenta a interface que permite a solução da instância criada pelo presente software.

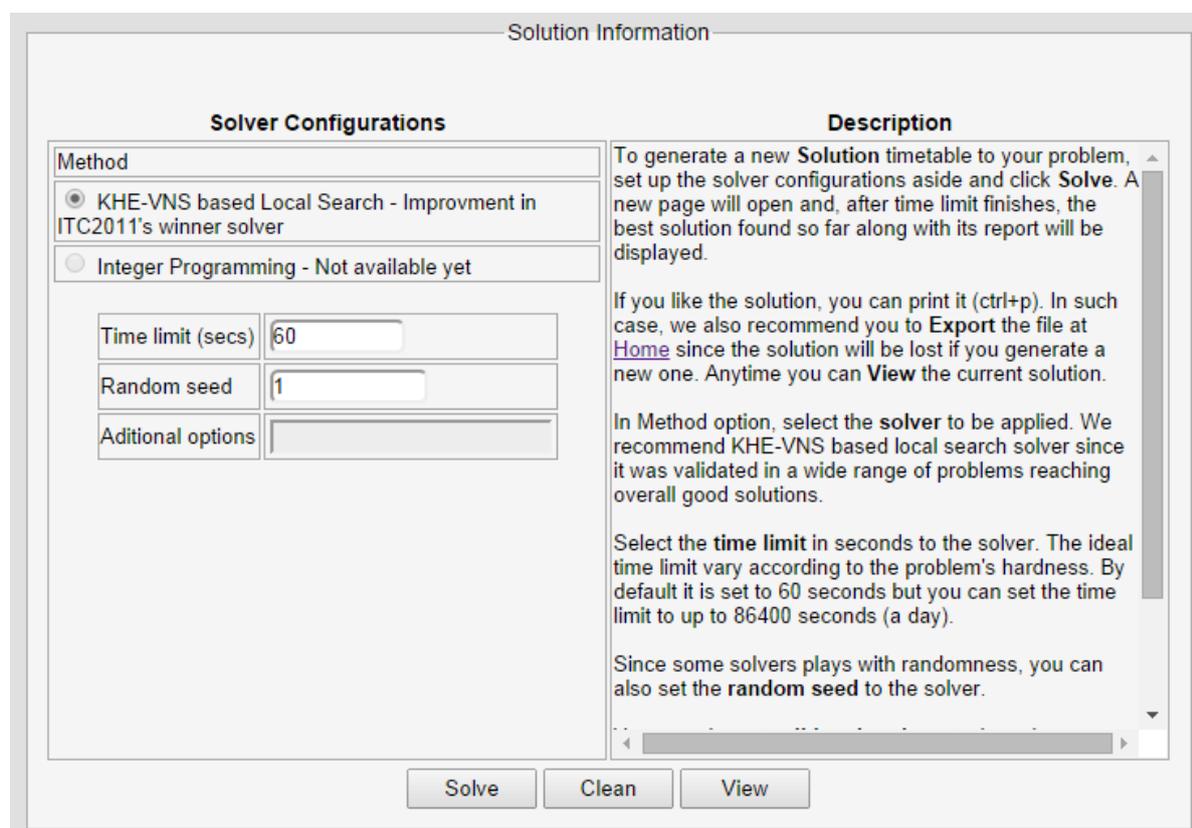


Figura 4.7: Interface para solução da instância.

Note que atualmente o software conta apenas com o resolvidor composto por uma melhoria no algoritmo vencedor da ITC2011 (Fonseca and Santos 2014). Como trabalhos futuros pretende-se incorporar novos resolvidores ao software, em especial resolvidores baseados em Programação Inteira (Daskalaki, Birbas and Housos 2004). O usuário deve especificar o tempo limite para a execução do resolvidor.

Após solucionada, a instância conterá uma entidade Solution contendo, para cada evento, o espaço de tempo e os recursos a ele alocados. Note pela Figura 4.8 que o evento Matematica_Discreta terá uma aula alocada no horário Mo_1 e outra aula no horário Tu_2 e que o evento empregará apenas recursos pre-alocados (os recursos são omitidos quando pre-alocados).

Solucionada a instância exibir-se-à uma página HTML contendo tabelas de horários de cada recurso (e.g. horários do Professor Eduardo ou horários da turma I de Sistemas

```
▼<HighSchoolTimetableArchive>
  ▶<Instances>...</Instances>
  ▼<SolutionGroups>
    ▼<SolutionGroup Id="instance">
      ▶<MetaData>...</MetaData>
      ▼<Solution Reference="instance">
        ▼<Events>
          ▼<Event Reference="Matematica_Discreta">
            <Duration>1</Duration>
            <Time Reference="Mo_1"/>
          </Event>
          ▼<Event Reference="Matematica_Discreta">
            <Duration>1</Duration>
            <Time Reference="Tu_2"/>
          </Event>
          ▶<Event Reference="Programacao_de_Computadores_I">...</Event>
          ▶<Event Reference="Programacao_de_Computadores_I">...</Event>
          ▶<Event Reference="Algoritmos_e_Estrutura_de_Dados">...</Event>
          ▶<Event Reference="Algoritmos_e_Estrutura_de_Dados">...</Event>
          ▶<Event Reference="Programacao_de_Computadores_II">...</Event>
          ▶<Event Reference="Programacao_de_Computadores_II">...</Event>
        </Events>
      </Solution>
    </SolutionGroup>
  </SolutionGroups>
</HighSchoolTimetableArchive>
```

Figura 4.8: Solução da instância.

de Informação) bem como as informações de violação de cada restrição caso ocorra. O horário pode ser impresso e/ou salvo no formato HTML para visualização em qualquer navegador Web. A Figura 4.9 apresenta uma tabela de horários.

Solution				
Overview				
Instance name: teste				
Infeasibility value: 0				
Objective value: 0				
Description: Improvement of ITC2011 Winner: VNS based Algorithms for High School Timetabling - Computers and OR				
Contributed by: GOAL team: Fonseca, G.H.G. and Santos, H.G., Sun Aug 24 12:34:14 2014				
Timetables by resource				
Eliane				
Monday	Tuesday	Wednesday	Thursday	Friday
				CEA001 Lin Algebra Analytic Geom.
		CEA001 Lin Algebra Analytic Geom.		

Figura 4.9: Exemplo de tabela de horários gerada pelo Timetabler.

4.2 Resolvedor Incorporado

O resolvedor é oriundo de uma melhoria no resolvedor vencedor da Third International Timetabling Competition (Fonseca, Santos, Toffolo, Brito and Souza 2014). Desde o período que o resolvedor ganhou a competição, alterações foram feitas em virtude de trabalhos acadêmicos do autor.

O resolvedor utiliza-se da plataforma KHE (Kingston 2006, Kingston 2012) para gerar soluções iniciais e é baseado em uma abordagem metaheurística empregando o algoritmo *Skewed Variable Neighborhood Search* (Hansen and Mladenovic 2000). A heurística considera seis movimentos para a geração de novas soluções (soluções vizinhas) (Fonseca and Santos 2014):

1. Trocar Evento. Dois eventos e_1 e e_2 têm seus espaços de tempo trocados.
2. Mover Evento. Um evento e é movido do espaço de tempo ot_1 para um novo espaço de tempo t_2 .
3. Trocar Bloco de Evento. Similar ao Trocar Evento, porém, se os eventos estão em horários contíguos e possuem durações diferentes, a troca mantém a contiguidade das alocações.
4. Trocar Recurso. Dois eventos e_1 e e_2 têm seus recursos r_1 e r_2 trocados. Os recursos r_1 e r_2 devem ser do mesmo tipo (e.g. ambos professores).
5. Mover Recurso. A um evento e é alocado um novo recurso r_2 em detrimento de r_1 previamente alocado.
6. Movimento Kempe. Dois espaços de tempo t_1 e t_2 são selecionados e procura-se pelo melhor caminho em um grafo bipartido de conflitos contendo todos os eventos alocados a t_1 e a t_2 . Arestas ligam eventos conflitantes. Dois eventos são ditos conflitante se estão em espaços de tempo diferentes e empregam recursos em comum. O custo de cada aresta é o custo supondo a troca de horário entre os eventos que liga.

Para mais informações sobre o resolvidor incorporado, leia (Fonseca, Santos, Toffolo, Brito and Souza 2014), (Fonseca and Santos 2014) e (Fonseca 2013a).

4.3 Instância Proposta

No intuito de testar e encontrar melhorias à interface desenvolvida, foi desenvolvida uma instância real com o auxílio do software. Foi considerado o problema de agendamento de horários do Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto no período letivo de 2014/2. A instância desenvolvida foi denominada *Brazil_UFOP_2014_2.xml* e está disponível no link <https://sites.google.com/site/georgegfonseca/software/timetabler>.

O referido instituto conta com aproximadamente 70 professores que ministram aulas para 4 cursos de graduação - Engenharia de Produção, Engenharia Elétrica, Engenharia de Computação e Sistemas de Informação, sendo 10 períodos letivos (turmas) para as engenharias e 8 períodos letivos no curso de Sistemas de Informação. Ao todo, tem-se

aproximadamente 380 aulas a serem alocadas. As aulas ocorrem em dois horários à tarde e dois à noite de segunda a sexta, totalizando 20 espaços de tempo. O problema considerado não prevê a alocação de recursos - a alocação de professores a disciplinas ocorre *à priori*. Além das restrições elementares de agendamento (Avoid Clashes e Assign Times), restrições importantes para a alocação de horários do instituto são: não alocar aulas aos professores nos horários em que os mesmos estão envolvidos com atividades de doutorado (Avoid Unavailable Times) e limitar o número de dias em que os professores estão ministrando aulas (Cluster Busy Times). Eventualmente deseja-se forçar a alocar disciplinas optativas no mesmo horário (Link Events). A Tabela 4.1 apresenta as principais características da instância criada.

Tabela 4.1: Características da Instância *Brazil_UFOP_2014_2.xml*

<i>Brazil_UFOP_2014_2</i>	
Características	
Professores	72
Timeslots	20
Classes	38
Eventos	170
Solução atual	(0, 5) ¹
Restrições (Peso)	
Assign Times	Hard (10)
Avoid Clashes	Hard (10)
Spread Events	Hard (10)
Split Events	Hard (1)
Prefer Times	Hard (1)
Avoid Unavailable Times	Soft (10)
Cluster Busy Times	Soft (1)

As restrições são classificadas em hard (essenciais) e soft (desejáveis). As do tipo hard tem prioridade sobre as soft. A solução (0,5) significa que nenhuma das restrições hard foram violadas, e cinco professores ficaram com atividades em mais de três dias (violação da Cluster Busy Times). Conclui-se que a solução gerada foi excelente.

¹O custo da solução é dada pelo par (hard, soft).

Capítulo 5

Considerações Finais

O trabalho apresentou um software Web para a criação e resolução de problemas de agendamento de horários no formato XHSTT. Esse formato foi criado no intuito de unificar os diversos formatos para especificação de problemas dessa classe e vem sendo amplamente aplicado na literatura desde a Third International Timetabling, competição que adotou o formato.

O software incorpora um resolvidor sucessor ao vencedor da competição e constitui uma potencial ferramenta para solucionar problemas reais de agendamento de horários através da interface desenvolvida. Além disso, o software permite a criação e a disponibilização de novas instâncias na literatura para estudos futuros na área de agendamento de horários.

Como trabalhos futuros, sugere-se: (1) incorporar mais resolvidores ao software como alternativa para o usuário (e.g. um resolvidor baseado em Programação Inteira pode ser mais interessante que um resolvidor meta-heurístico dependendo das características da instância); (2) realizar avaliações e melhorias de usabilidade no software; (3) desenvolver um manual do usuário e suporte para outros idiomas; e (4) realizar estudos de caso de aplicação do software em instituições de ensino ou outras em que se faz necessário o agendamento de horários.

Apêndice A

Apêndices

A.1 Publicações

Neste apêndice são listados os trabalhos publicados e apresentados em eventos científicos desenvolvidos durante o período de realização da presente pesquisa.

1. FONSECA, G. H. G. ; DELFINO, T. D. . **Software Web para Problemas de Agendamento de Horários Modelados em XHSTT**. In: *XLIV Simpósio Brasileiro de Pesquisa Operacional*, 2014, Salvador, Brasil.
2. FONSECA, G. H. G. ; DELFINO, T. D. ; SANTOS, H. G. . **A Web-Software to handle XHSTT Timetabling Problems**. In: *10th International Conference on the Practice and Theory of Automated Timetabling*, York, United Kingdom. Tue. 26th ? Fri. 29th August 2014.

Referências Bibliográficas

- Almeida, M. B.: 2002, Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares, *Ciência da Informação* **31**, 5 – 13.
- AsCTimetables, T.: 2014, ascTimetables - características. Disponível em <http://www.asctimetables.com/>, Acessado em Agosto / 2014.
- Daskalaki, S., Birbas, T. and Housos, E.: 2004, An integer programming formulation for a case study in university timetabling., *European Journal of Operational Research* **153**(1), 117–135.
- Fet: 2014, Free timetabling software description. Disponível em <http://lalescu.ro/liviu/>, Acessado em Agosto / 2014.
- Fonseca, G. H. G.: 2013a, *Metodos de Busca Heuristica para Problemas de Programacao de Horarios Modelados em XHSTT*, dissertacao de Mestrado em Ciencia da Computacao, Universidade Federal de Ouro Preto, Ouro Preto, Brasil.
- Fonseca, G. H. G. d.: 2013b, Métodos de busca heurística para problemas de programação de horários modelados em xhstt.
- Fonseca, G. and Santos, H. G.: 2014, Variable neighborhood search based algorithms for high school timetabling, *Computers and Operations Research* .
- Fonseca, G., Santos, H. G., Toffolo, T. M., Brito, S. S. and Souza, M. J. F.: 2014, Internation timetabling competition: GOAL team solver description, *To Appear in Annals of Operations Research* .
- Gonçalves, E.: 2007, *Desenvolvendo aplicações web com JSP, Servlets, JavaServer Faces, Hibernate, EJB 3 Persistence e AJAX* /, Ciência Moderna,, Rio de Janeiro :.
- Gutmans, A., Bakken, S. and Rethans, D.: 2005, *PHP 5: programação poderosa*, Série Open Source Bruce Perens, Alta Books.
- Hansen, P. and Mladenovic, N.: 2000, *Variable Neighborhood Search: A Chapter of Handbook of Applied Optimization.*, Les Cahiers du GERAD G-2000-3. Montreal, Canada, chapter 8.

- Hartog, J.: 2007, *Timetabling on Dutch High Schools. Satisfiability versus gp-Untis*, Master thesis, University of Technology – TU, Delft, The Netherlands. <http://www.st.ewi.tudelft.nl/sat/theses/jantien.pdf> – last visited 9th september 2007.
- Jutte, S. and Thonemann, U.: 2012, Divide-and-price: A decomposition algorithm for solving large railway crew scheduling problems, *European Journal of Operational Research*, pp. 219.2: 214–223.
- Kingston, J. H.: 2006, Hierarchical timetable construction, *Problems, Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling*.
- Kingston, J. H.: 2012, A software library for school timetabling. Disponível em <http://sydney.edu.au/engineering/it/jeff/khe/>, Acessado em Abril de 2012.
- Marte, M.: 2002, *Models and Algorithms for School Timetabling*, PhD thesis.
- Medina, R. D.: 2014, A linguagem xml. Notas de aula. Universidade Federal de Santa Maria. Disponível em <http://www-usr.inf.ufsm.br/~rose/curso3/cafe/XML-Cap1-Linguagem.pdf>, Acessado em Setembro / 2014.
- Mimosa, S. L.: 2014, Mimosa scheduling software. Disponível em <http://www.mimosasoftware.com/>, Acessado em Agosto / 2014.
- of Twente, U.: 2011, Tutorial. Available at <http://www.utwente.nl/ctit/hstt/tutorial/>, Accessed in October / 2014.
- of Twente, U.: 2012, International Timetabling Competition 2012. Available at <http://www.utwente.nl/ctit/hstt/itc2011/welcome/>, Accessed in October / 2014.
- Post, G.: 2010, An xml format for benchmarks in high school timetabling, *Annals of Operations Research DOI 10.1007/s10479-010-0699-9*, pp. 3867 : 267–279.
- Post, G. e. a.: 2011, XHSTT: an XML archive for high school timetabling problems in different countries, *Annals of Operations Research* pp. 1–7.
- Santos, H. G., Toffolo, T. M., Ribas, S. and Gomes, R. A. M.: 2012, Integer programming techniques for the nurse rostering problem, *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling. SINTEF, Son, Norway.*, pp. 258–282.
- Santos, H. G., Uchoa, E., Ochi, L. and Maculan, N.: 2012, Strong bounds with cut and column generation for class-teacher timetabling, *Annals of Operations Research*, pp. 194 : 399–412.