

SOFTWARE WEB PARA PROBLEMAS DE AGENDAMENTO DE HORÁRIOS MODELADOS EM XHSTT

George Henrique Godim da Fonseca

Universidade Federal de Ouro Preto
Rua 36, 115 B. Vera Cruz – João Monlevade / MG
george@decea.ufop.br

Thaise Daiara Delfino

Universidade Federal de Ouro Preto
Rua 36, 115 B. Vera Cruz – João Monlevade / MG
thaisedd@yahoo.com.br

RESUMO

O presente trabalho apresenta um software Web capaz de especificar e resolver problemas de agendamento de horários especificados no formato XHSTT. O formato XHSTT foi proposto com o objetivo de unificar os diversos formatos da literatura para essa classe de problemas, de modo que um resolvidor seja aplicável a uma vasta gama de problemas. A Terceira Competição Internacional de Agendamento de Horários empregou o formato XHSTT e estimulou o desenvolvimento de resolvidores eficientes para problemas dessa classe. Porém, esses resolvidores não possuem interface gráfica, assim, seu uso, atualmente é praticamente restrito a pesquisadores da área. O presente trabalho propõe um software Web de fácil uso para interessados especificarem e resolverem problemas de agendamento de horários através do resolvidor vencedor da última competição de agendamento de horários. O software auxiliará ainda na criação de novas instâncias dessa classe de problemas para a literatura. Como exemplo, desenvolveu-se e disponibilizou-se instâncias da Universidade Federal de Ouro Preto para a literatura.

PALAVRAS CHAVE. Agendamento de Horários. Formato XHSTT. Software Web.

Pesquisa Operacional na Educação; Otimização Combinatória.

ABSTRACT

This work presents a Web-software able to specify and solve XHSTT modeled timetabling problems. The XHSTT format was proposed aiming to unify several I/O formats in literature for this class of problems in such way that a solver can be applied to a wide range of problems. The Third International Timetabling Competition employed the XHSTT format and stimulated the development of efficient solvers for this class of problems. Therefore, those solvers has not user graphical interface, thus, its use, is practically restricted to timetabling researchers. The present work proposes an easy to use software letting the user to specify and solve timetabling problems through the solver winner of the last timetabling competition. This software will also be useful to create new timetabling instances for researchers. As example, we specified timetabling instances from Federal University of Ouro Preto with our software.

KEYWORDS. Timetabling. XHSTT Format. Web Software.

Operational Research in Education; Combinatorial Optimization.

1. Introdução

O presente trabalho propõe um Software Web para manusear e resolver problemas de agendamento de horários modelados em XHSTT. O formato XHSTT é um formato de arquivo 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 esse formato.

Esse formato foi considerado na ITC2011 (*Third International Timetabling Competition*). A competição motivou o desenvolvimento de diversas soluções de otimização eficientes para essa categoria de problemas (of Twente, 2012). Em contrapartida, nenhum dos softwares desenvolvidos para a competição emprega alguma forma de interação gráfica com o usuário, o que dificulta/impossibilita essas soluções de serem utilizadas por instituições de ensino. Ademais, a especificação manual do formato XHSTT é complexa e trabalhosa até mesmo para pesquisadores em agendamento de horários.

Assim, o principal objetivo do presente trabalho é apresentar um Software Web que abstraia o usuário do conhecimento do formato XHSTT de modo que qualquer interessado - seja pesquisador ou gestor de instituição de ensino - esteja apto a especificar e resolver problemas de agendamento de horários. O software proposto conta com o resolvidor vencedor da *Third International Timetabling Competition* (Fonseca *et al.*, 2014) para solucionar o problema após especificado pelo usuário.

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 resolvidores de agendamento de horários que lidam com o formato. Nesse sentido, o presente trabalho introduz também duas novas instâncias do problema na literatura, geradas com o auxílio do software apresentado.

O restante do trabalho está organizado da seguinte maneira. A Seção 2 disserta sobre o problema do agendamento de horários e apresenta o formato XHSTT para especificação de problemas dessa classe. A Seção 3 apresenta detalhadamente o software desenvolvido e a Seção 4 apresenta resumidamente o resolvidor incorporado no presente software. A Seção 5 apresenta um conjunto de instâncias reais geradas com o auxílio do software desenvolvido. Por fim, a Seção 6 apresenta as considerações finais.

2. Problema do Agendamento de Horários

O Problema do Agendamento de Horários consiste elementarmente em alocar horários e recursos para a realização de tarefas em um dado horizonte de planejamento. 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 de problemas *NP*, 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 *et al.*, 2012), o Problema da Programação de Horários Escolares (Santos, Uchoa e Ochi, 2012) e o Problema do Agendamento de Tripulações (Jutte e Thonemann, 2012).

2.1. Formato XHSTT

O formato XHSTT foi criado por Post *et al.* (2011) e consiste de um formato de arquivo baseado no XML (*eXtensible Markup Language*). O formato foi 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 seja aplicada a diversos outros problemas do gênero, desde que especificadas de acordo

com esse formato.

O código da Figura 01 apresenta um exemplo em alto nível de arquivo no formato XHSTT. Cada instância no formato XHSTT é dividida em quatro entidades principais: Tempos, Recursos, Eventos e Restrições. As subseções a seguir expandem a descrição sobre essas entidades. Para mais detalhes sobre a especificação de instâncias e soluções no formato XHSTT, veja Post *et al.* (2011) e Kingston (2012).

```

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

```

Figura 01: Exemplo de alto nível de instância especificada no formato XHSTT.

2.1.1. 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.

2.1.2. 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;
- **ResourceGroups.** Especifica os grupos de recursos aos quais o recurso pertence, por exemplo: Turmas do curso de Sistemas de Informação;

2.1.3. Eventos

Um evento (*Event*) é um conjunto de encontros (aulas) sobre um determinado assunto. Eventos podem também ser agrupados, 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 devem ser alocados ao evento;
- **Resources.** Especifica os recursos que deverão ser alocados ao evento;
- **Workload.** Carga de trabalho que será adicionada à carga de trabalho total dos recursos alocados ao evento (opcional);
- **Time.** Espaço de tempo em que o evento está pre-alocado (opcional).

2.1.4. Restrições

Post *et al.* (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 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 ou fraca em seu contexto. As subseções a seguir apresentam cada um dos conjuntos de restrições bem como a avaliação de função objetivo.

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 um 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, todas 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 consecutivas 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 duas aulas consecutivas, uma vez que uma aula em separado não seria suficiente para lecionar o 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 algum curso de capacitação;

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 a 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, os grupos de tempo são dias e, por exemplo, um professor requer no máximo três dias com aulas.

3. Software Desenvolvido

O software desenvolvido provê uma interface HTML (*HyperText Markup Language*) para o usuário especificar qualquer instância de agendamento de horários no formato XHSTT. A interface foi desenvolvida objetivando a encontrar um ponto de equilíbrio entre a facilidade de uso e aprendizado e a similaridade com o formato XHSTT. Para tal, nossa implementação abstraiu algumas características do referido formato. O software encontra-se disponível através do link <https://sites.google.com/site/georgehfonseca/software/timetabler>. Convidamos o leitor interessado a testar nossa aplicação e apresentar sugestões de melhoria.

A Figura 02 apresenta a interface geral do nosso software e a Figura 03, as entidades geradas automaticamente no arquivo XHSTT quando da criação da instância.

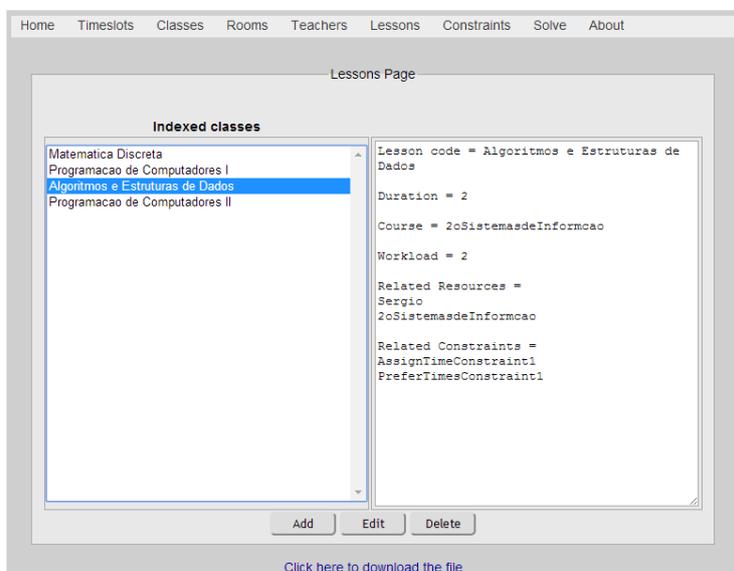


Figura 02: Interface principal do software.

```

▼<HighSchoolTimetableArchive>
  ▼<Instances>
    ▼<Instance>
      ▼<Times>
        ▶<TimeGroups>...</TimeGroups>
      </Times>
      ▼<Resources>
        ▶<ResourceTypes>...</ResourceTypes>
        ▶<ResourceGroups>...</ResourceGroups>
      </Resources>
      ▼<Events>
        ▼<EventGroups>
          ▶<EventGroup Id="gr_All_events">...</EventGroup>
        </EventGroups>
      </Events>
      <Constraints/>
    </Instance>
  </Instances>
</HighSchoolTimetableArchive>
  
```

Figura 03: 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 na Figura 03 que os elementos *TimeGroups*, *ResourceGroups* e *EventGroups* são criados automaticamente.

A Figura 04 apresenta a criação da entidade *Times* através do presente software.

Timeslots Page

Number of days in week

Number of timeslots for day

```

▼<Times>
  ▼<TimeGroups>
    ▼<TimeGroup Id="gr_AllTimes">
      <Name>AllTimes</Name>
    </TimeGroup>
    ▶<Day Id="gr_Mo">...</Day>
    ▶<Day Id="gr_Tu">...</Day>
  </TimeGroups>
  ▶<Time Id="Mo_1">...</Time>
  ▶<Time Id="Mo_2">...</Time>
  ▶<Time Id="Tu_1">...</Time>
  ▶<Time Id="Tu_2">...</Time>
</Times>
  
```

Figura 04: Criação da entidade *Times* através do presente software.

A edição de recursos através do nosso software é trivial. Necessita-se apenas do nome do recurso como identificador. No presente trabalho considerou apenas três tipos possíveis de recurso: Classe, Professor e Sala. Para criar qualquer tipo de recurso, a interface é similar à apresentada na Figura 05.

Classes

Name

```

▼<Resource Id="1o Sistemas de Informacao">
  <Name>1o Sistemas de Informacao</Name>
  <ResourceType Reference="Class"/>
  ▼<ResourceGroups>
    <ResourceGroup Reference="gr_Classes"/>
  </ResourceGroups>
</Resource>
  
```

Figura 05: Criação da entidade *Resource* através do presente software.

A Figura 06 apresenta a criação da entidade *Event* através do presente software.

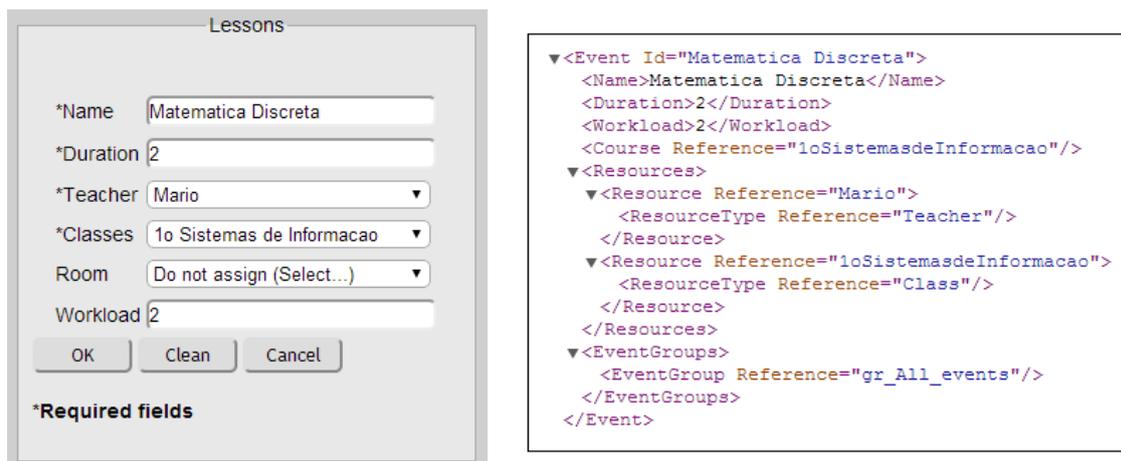


Figura 06: Criação da entidade Event através do presente software.

Para tratar a criação de distintas restrições previstas pela entidade Constraints, criou-se uma janela para cada restrição prevista no formato XHSTT. A Figura 07 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 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.

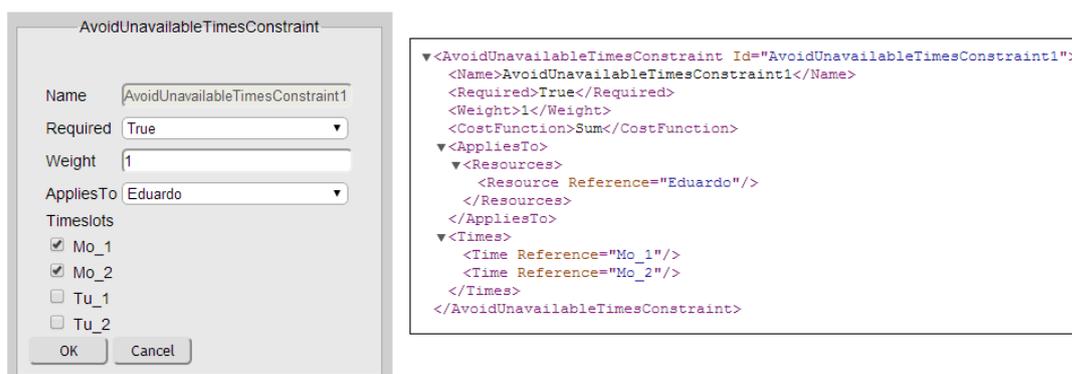


Figura 07: Criação de restrições, em específico *Avoid Unavailable Times*, através do presente software.

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 08 apresenta a interface que permite a solução da instância criada pelo presente software.

Note que atualmente nosso software conta apenas com o resolvidor composto por uma melhoria no algoritmo vencedor da ITC2011 (Fonseca e Santos, 2014). Futuramente pretende-se incorporar novos resolvidores ao software, em especial resolvidores baseados em Programação Inteira (Daskalaki, Birbas e 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 ainda pela Figura 08 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).

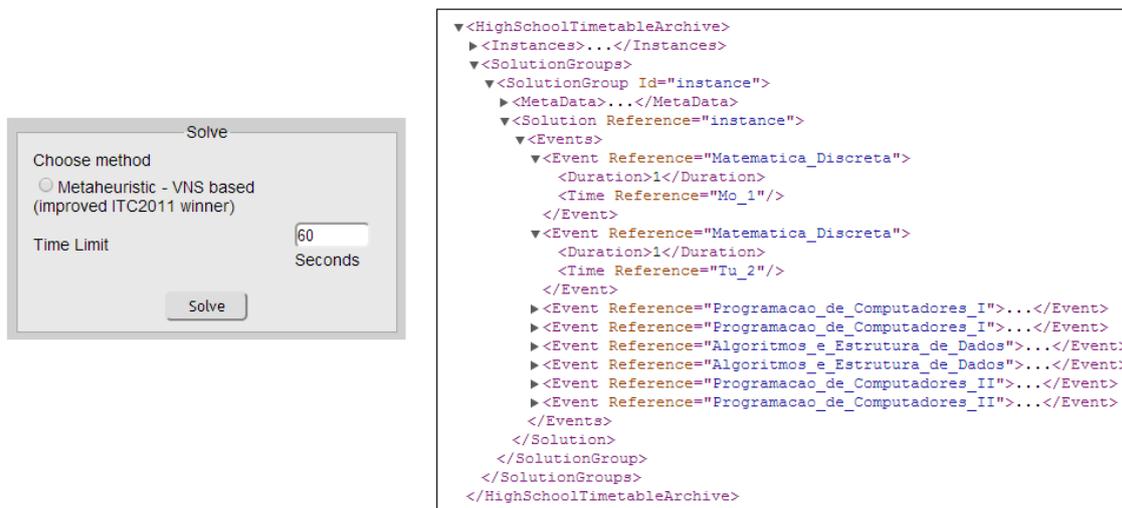


Figura 08: Solução da instância através do presente software.

Solucionada a instância exibir-se-á uma página HTML contendo tabelas de horários de cada recurso (e.g. horários do Professor Eduardo ou horários da turma 1o Período de Sistemas 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.

4. Resolvedor Incorporado

O presente software incorpora o resolvedor oriundo de uma melhoria no resolvedor vencedor da *Third International Timetabling Competition* (Fonseca *et al.*, 2014). 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 e Mladenovic, 2000). A heurística considera seis movimentos para a geração de novas soluções (soluções vizinhas) (Fonseca e 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 t_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 resolvedor incorporado, leia Fonseca *et al.* (2014), Fonseca e Santos (2014) e Fonseca (2013).

5. Instâncias Propostas

No intuito de testar a aplicação e encontrar melhorias à interface desenvolvida, foram desenvolvidas instâncias reais 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 nos períodos letivos de 2014/1 e 2013/2.

As instâncias desenvolvidas, denominadas *Brazil_UFOP_2013_2.xml* e *Brazil_UFOP_2014_1.xml* estão disponíveis no link <https://sites.google.com/site/georgehgfonseca/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. Os cursos de engenharia possuem 10 períodos letivos (turmas) e o curso de Sistemas de Informação, 8 períodos letivos. 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 doutoramento (*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*).

6. 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* que adotou o formato.

O software incorpora um resolvedor sucessor ao vencedor da competição e constitui uma potencial ferramenta para solucionar problemas reais de agendamento de horários através de sua interface intuitiva. Além disso, o software permitirá que novas instâncias sejam criadas e disponibilizadas na literatura para estudos futuros na área de agendamento de horários.

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

Referências

- Daskalaki, S., Birbas, T., and Housos, E.** *An integer programming formulation for a case study in university timetabling*. European Journal of Operations Research, pages 153(1) : 117–135, 2004.
- Fonseca, G. and Santos, H. G.** *Variable neighborhood search based algorithms for high school timetabling*. Computers and Operations Research, 2014a.
- Fonseca, G., Santos, H. G., Toffolo, T. M., Brito, S. S., and Souza, M. J. F.** *International timetabling competition: GOAL team solver description*. To Appear in Annals of Operations Research, 2014b.
- Fonseca, G. H. G.** *Metodos de Busca Heuristica para Problemas de Programacao de Horarios Modelados em XHSTT*. Dissertação de Mestrado em Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, Brasil, 2013.
- Hansen, P. and Mladenovic, N.** *Variable Neighborhood Search: A Chapter of Handbook of Applied Optimization.*, chapter 8. Les Cahiers du GERAD G-2000-3. Montreal, Canada, 2000.
- Jutte, S. and Thonemann, U.** *Divide-and-price: A decomposition algorithm for solving large railway crew scheduling problems*. In European Journal of Operational Research, pages 219.2: 214–223, 2012.
- Kingston, J. H.** *Hierarchical timetable construction*. In Problems, Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling, 2006.
- Kingston, J. H.** *A software library for school timetabling*. Disponível em <http://sydney.edu.au/engineering/it/jeff/khe/>, Acessado em Abril de 2012.
- of Twente, University.** *International Timetabling Competition 2012*. Disponível em <http://www.utwente.nl/ctit/hstt/itc2011/welcome/>, Acessado em de 2012.
- Post, G., Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngas, J., Nurmi, C., and Ranson, D. (2010).** *An xml format for benchmarks in high school timetabling*. In Annals of Operations Research DOI 10.1007/s10479-010-0699-9., pages 3867 : 267–279, 2010.
- Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., and Schaerf, A.** *XHSTT: an XML archive for high school timetabling problems in different countries*. Annals of Operations Research, pages 1–7, 2011.
- Santos, H. G., E., U., L., O., and N., M.** *Strong bounds with cut and column generation for class-teacher timetabling*. In Annals of Operations Research, pages 194 : 399–412, 2012a.
- Santos, H. G., Toffolo, T. M., Ribas, S., and Gomes, R. A. M.** *Integer programming techniques for the nurse rostering problem*. In Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling. SINTEF, Son, Norway., pages 258–282, 2012b.