10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

A Web-Software to handle XHSTT Timetabling Problems

George H.G. Fonseca · Thaise D. Delfino · Haroldo G. Santos

Abstract This work presents a Web-Software to handle XHSTT timetabling problems. The XHSTT format is complex and virtually only timetabling researchers are able to work with this file format. The software goal is to abstract the user from XHSTT knowledge and make any person able to specify and solve timetabling problems through XHSTT format. This software may popularize this file format and aid the new real world instances specification.

Keywords Timetabling · XHSTT format · Web-Software

1 Introduction

The diversity of School Timetabling models encountered around the world motivated the definition of an XML format (XHSTT) to express different entities and constraints considered when building timetables [6]. Based in this format definition and aiming to stimulated the research in this area, the Third International Timetabling Competition (ITC2011) occurred in 2012 [5]. It motivated the development of several efficient algorithms for Timetabling problems [2, 7, 1].

A problem faced in timetabling research is the gap between research and practice - many algorithms are proposed and validated in fictitious instances of the problem. Additionally, many of these algorithms as the ones developed for ITC2011 does not implement an intuitive user interaction mode. This makes it difficult to apply these algorithms to solve real problems. Thus, the objective of this paper is to describe a Web-Software developed to assist the specification of XHSTT timetabling problems and the solution of these problems through recognized quality solvers.

Systems and Computing Department
Federal University of Ouro Preto
Ouro Preto, Minas Gerais, Brazil 35400-000

2 XHSTT format

XHSTT format was proposed by Post *et al.* [6] to specify timetabling problems. It is split into four main entities:

Times
Set of timeslots available for assignments;

Resources
Set of resources involved in the problem (usually teachers, classes and rooms);

Events
Set of events that must be scheduled (usually lessons);

Constraints
Set of constraints that must be satisfied to a solution be feasible and/or good.

A solution of an XHSTT specified problem consists in an assignment of timeslots and resources to the events respecting the given mandatory constraints and maximizing the attendance of desirable constraints. A detailed explanation of the file format is presented by Kingston [3].

3 Software User Interface

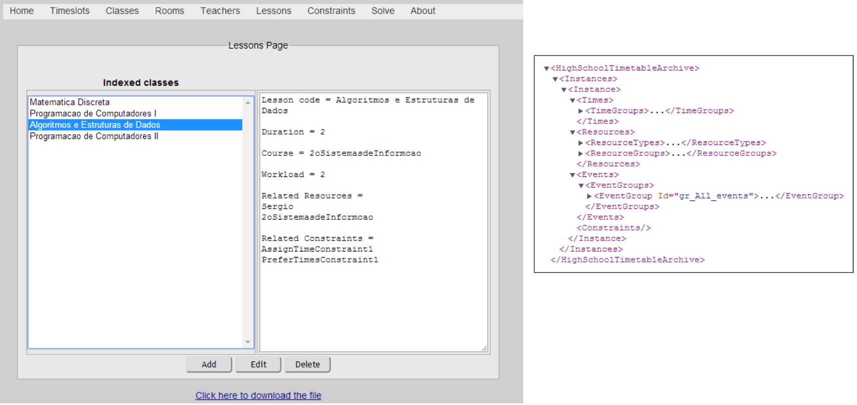
The developed software provides a HTML interface for the user to specify any instance of timetabling problem in XHSTT format. The interface design of the software was made intending to find an equilibrium point between learning curve and similarity to the XHSTT format. After specified, the XHSTT modeled problem may be submitted to the ITC2011 winner solver [2]. Figure 1 presents the general interface of our software and the automatically generated entities of XHSTT file in the instance creation.

Note that some entities creation were automated intending to make easier for the user to operate the software. As one can see in Figure 1, the elements *TimeGroups*, *ResourceGroups* and *EventGroups* are created automatically. The Times specification is made based only on the number of days in the schedule and the number of timeslots by day. The concept of Role was also abstracted from the user. Our software only allows the user to create two roles for an event: Teacher and Room, denoted respectively as 0 and 1.

After initializing an instance, the user should specify the time-slots available for the assignment. Figure 2 presents the automated *Times* entity creation through our software.

The next step in XHSTT instance generation through our software is the available resources input. The resource creation through our system is quite easy. We consider only three types of resource: classes, teachers and rooms. To create any of them, the user interface is similar as follows at Figure 3.

Defined the available time-slots and resources, one needs to specify the events to be scheduled. Figure 4 presents the automated *Event* entity creation through our software.

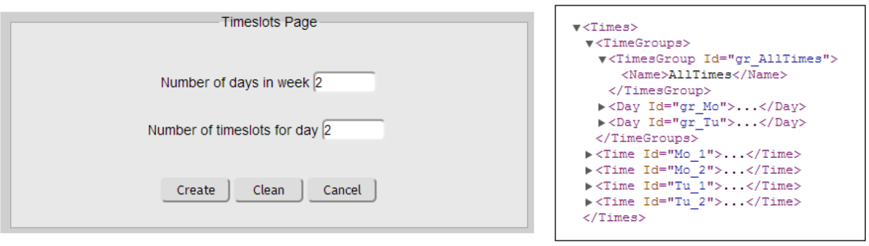


Matematica Discreta
Programacao de Computadores I
Algoritmos e Estruturas de Dados
Programacao de Computadores II

```
Lesson code = Algoritmos e Estruturas de
Dados

Duration = 2

Course = 2oSistemasdeInformcao

Workload = 2

Related Resources =
Sergio
2oSistemasdeInformcao

Related Constraints =
AssignTimeConstraint1
PreferTimesConstraint1
```

```
▼<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>
```

Fig. 1 Main user interface of software and XHSTT tags generated when a new instance is created.

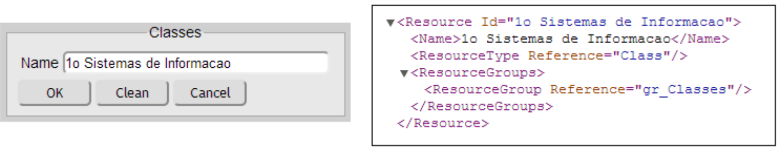


Number of days in week 2

Number of timeslots for day 2

```
▼<Times>
  ▼<TimeGroups>
    ▼<TimesGroup Id="gr_AllTimes">
        <Name>AllTimes</Name>
      </TimesGroup>
    ▶<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>
```

Fig. 2 *Times* entity through our software.



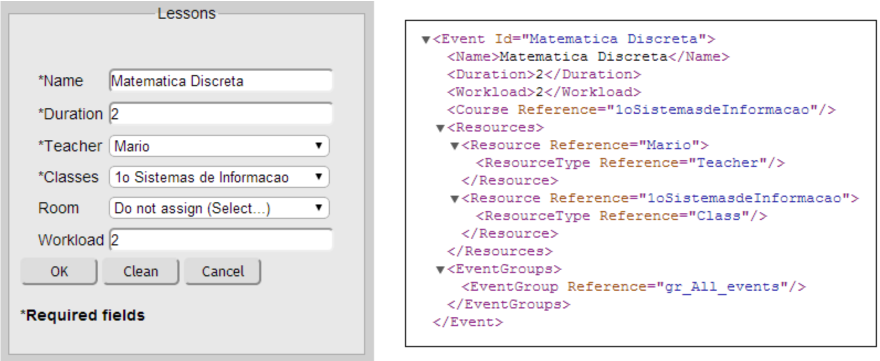
Name 1o Sistemas de Informacao

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

Fig. 3 *Resource* entity creation through our software.

After specify the time-slots, resources and events, the user should input the constraints to his problem. To handle the constraints entities, we create one window for each constraint from XHSTT format. Figure 5 presents the automated *Constraint* entity creation with our software, specifically an *Avoid Unavailable Times Constraint*.

At this point the whole instance is specified and ready to be solved. The user may also change any information at any moment. Now the user should specify the execution time limit and the solver for the instance as Figure 6 points out. After the specified running time the software will automatically generate a HTML page containing the planning timetables by resource and by event as well as the violated constraints.

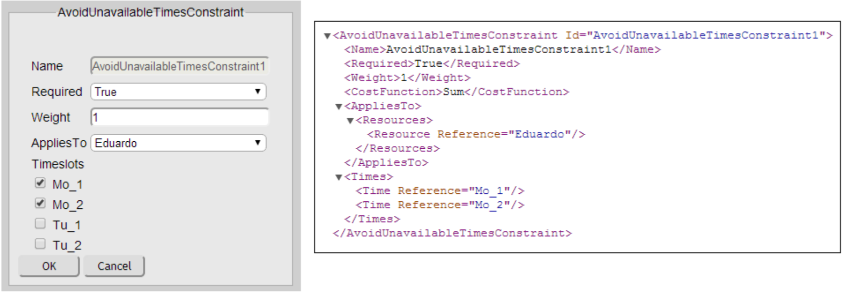


*Name: Matematica Discreta
*Duration: 2
*Teacher: Mario
*Classes: 1o Sistemas de Informacao
Room: Do not assign (Select...)
Workload: 2

*Required fields

```
▼<Event Id="Matematica Discreta">
   <Name>Matematica Discreta</Name>
   <Duration>2</Duration>
   <Workload>2</Workload>
   <Course Reference="1oSistemasdeInformacao"/>
 ▼<Resources>
  ▼<Resource Reference="Mario">
     <ResourceType Reference="Teacher"/>
    </Resource>
  ▼<Resource Reference="1oSistemasdeInformacao">
     <ResourceType Reference="Class"/>
    </Resource>
   </Resources>
 ▼<EventGroups>
    <EventGroup Reference="gr_All_events"/>
   </EventGroups>
 </Event>
```

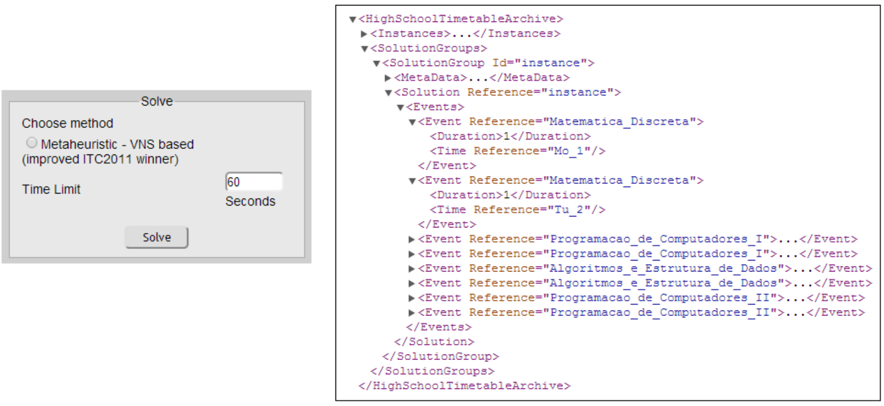
Fig. 4 *Event* entity creation through our software.



Name: AvoidUnavailableTimesConstraint1
Required: True
Weight: 1
AppliesTo: Eduardo
Timeslots
☑ Mo_1
☑ Mo_2
□ Tu_1
□ Tu_2

```
▼<AvoidUnavailableTimesConstraint Id="AvoidUnavailableTimesConstraint1">
   <Name>AvoidUnavailableTimesConstraint1</Name>
   <Required>True</Required>
   <Weight>1</Weight>
   <CostFunction>Sum</CostFunction>
 ▼<AppliesTo>
  ▼<Resources>
     <Resource Reference="Eduardo"/>
    </Resources>
   </AppliesTo>
 ▼<Times>
    <Time Reference="Mo_1"/>
    <Time Reference="Mo_2"/>
   </Times>
 </AvoidUnavailableTimesConstraint>
```

Fig. 5 *Constraint* entity creation through our software.



Choose method
○ Metaheuristic - VNS based (improved ITC2011 winner)
Time Limit: 60 Seconds

```
▼<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>
```

Fig. 6 Instance solution through our software.

4 Concluding Remarks

This paper presented a Web-Software for handling XHSTT timetabling problems. The software is available at https://sites.google.com/site/georgehgfonseca/software/timetabler. We invite the interested reader to use and contribute to our software. The proposed software aims to be easy to use, even by people who does not know about the XHSTT format. Indeed the great majority of people responsible by the timetabling in schools does not know yet about this format. In this sense the software may also help to popularize this file format to express timetabling problems.

The software may also assists researchers and practitioners to contribute with new real instances of the problem from their institutions.

As future research we suggest to:

- Introduce more solvers to the software;
- Evaluate the user experience with the software;
- Improve the usability of the software;
- Integrate this software with HSEval evaluator software [4].

References

1. Fonseca, G., Santos, H.: Variable neighborhood search based algotihms for high school timetabling. Computers and Operational Research (2014)
2. Fonseca, G., Santos, H., Toffolo, T., Brito, S., Souza, M.: A sa-ils approach for the high school timetabling problem. Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012) pp. 493–496 (2012)
3. Kingston, J.H.: A software library for school timetabling (2012). Available at http://sydney.edu.au/engineering/it/~jeff/khe/, May 2012
4. Kingston, J.H.: The hseval high school timetable evaluator (2014). Available at http://sydney.edu.au/engineering/it/~jeff/hseval.cgi, February 2014
5. Post, G., Di Gaspero, L., Kingston, J.H., McCollum, B., Schaerf, A.: The third international timetabling competition. In: Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (2012)
6. Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., Schaerf, A.: XHSTT: an XML archive for high school timetabling problems in different countries. Annals of Operations Research pp. 1–7
7. Sorensen, M., Kristiansen, S., Stidsen, T.: International Timetabling Competition 2011: An Adaptive Large Neighborhood Search algorithm, pp. 489–492 (2012)