A collaborative approach for solving the university course timetabling problem

Carlos Bazilio
carlosbazilio@id.uff.br / Federal Fluminense University. Rio das Ostras, Brazil

Dalessandro Soares Vianna
dalessandrovianna@id.uff.br / Federal Fluminense University. Rio das Ostras, Brazil

Thiago Jeffery Barisao de Lima
thiagojeffery@id.uff.br / Federal Fluminense University. Rio das Ostras, Brazil

Edwin Benito Mitacc Meza
emitacc@id.uff.br / Federal Fluminense University. Rio das Ostras, Brazil

Receipt: 30-6-2018 / Acceptance: 21-8-2018

Abstract. This work proposes a collaborative approach for solving the university course timetabling problem (UCTP). A prototype was developed and used for a computer science course at the Federal Fluminense University in Brazil. The main idea is that students, professors, and course coordinators contribute collaboratively to course timetabling through an app. These contributions employ heuristics, which is responsible for timetabling to improve the solution to the problem. Results and future works are described herein.

Keywords: timetabling, heuristic methods, courses, collaborative applications

Enfoque colaborativo para resolver el problema de la programación de horarios de asignaturas universitarias

Resumen. Este trabajo propone un enfoque colaborativo para resolver el problema de la programación de horarios de asignaturas universitarias (UCTP). Se desarrolló y utilizó un prototipo para un curso de ciencias de la computación de la Universidad Federal Fluminense de Brasil. La idea principal es que los estudiantes, profesores y coordinadores de asignatura contribuyan de forma colaborativa a la programación de horarios a través de un aplicativo. Estas contribuciones utilizan la heurística, responsable de la programación de horarios, a fin de mejorar la resolución del problema. Se describen los resultados alcanzados y los trabajos futuros.

Palabras clave: programación de horarios, métodos heurísticos, asignaturas, aplicaciones colaborativas

1. Introduction

Schaerf (1999) defines educational timetabling as the task of scheduling professors' and students' meetings to satisfy their constraints. Said constraints can be the availability of professors, availability of rooms, among other things. These tasks usually require substantial time to be accomplished. In many cases, a single person, as a course coordinator, is responsible for this task.

Collaboration is an activity in which many individuals cooperate in order to accomplish a task. Due to current technological advances, including smartphones and high-speed network interconnection, a lot of complex problems, such as traffic routing, time management, and shared economy, have been managed in a smoother way through blockchain technology. As educational timetabling affects professors, students, and managers, a collaborative approach can increase their satisfaction regarding time scheduling.

This work proposes a collaborative way of solving the course timetabling problem. A VNS-VND heuristic method (Lima, Martins, Vianna, Dianin, & Costa, 2015) with two neighborhood structures (i.e., k-Reschedule and k-Exchange) was used to deal with problem constraints. In k-Reschedule neighborhood, k allocations are randomly chosen and rescheduled (i.e., each of the k allocations is removed from the current day/time and inserted into another day/time.) In k-Exchange neighborhood, k allocation pairs (a1, a2) are randomly chosen. Each allocation pair is inverted (i.e., allocations a1 and a2 exchange their positions.) In the VNS disturbance procedure, neighborhoods 3-Reschedule and 3-Exchange are used. The VND local search method explores neighborhoods 1-Exchange, 2-Reschedule, and 2-Exchange.

Since these constraints differ, Multicriteria Decision Aid (MCDA) methods were applied. Despite adopted heuristic techniques greatly assist in problem solving, achieved solutions still caused dissatisfaction to system users. In order to minimize these issues, our proposal received users' suggestions which could improve heuristic solutions. Thus, an essentially collaborative system is presented. In this system, students and professors are able to interact and comment a provided timetable in order to improve it. Then, it is possible to prepare and communicate better timetables, meeting the requirements and preferences set by professors, students, and course coordinators.

This work has the following sections: Section 2 presents a research related to our approach; section 3 contains the required concepts to understand this proposal; section 4 shows the implemented prototype which has been used by a computer science course; section 5 presents some notes about a recently conducted prototype experiment; and finally section 6 includes our proposal and points out some future works.

2. Related work

Waze[1] is a mobile application that connects drivers and provides information about streets, and can be sent by other users or inferred by the application. For instance, using locations sent by drivers from time to time, Waze can infer how fast drivers are moving on streets. Then it can advise which tracks should be taken to avoid traffic jams. Routing algorithms try to solve this hard problem, which can be softened by the collaboration of drivers. This inspired us to use collaboration for dealing with the university course timetabling problem.

Authors Babaei, Karimpour, & Hadidi (2015), and Pandey & Sharma (2016) list a number of research studies which address UCTP. A lot of techniques and methodologies, such as heuristics and meta-heuristics, are employed. None of them take collaboration opportunities into account.

Collaborative Design Optimization (Parte, Auroux, Clément, Masmoudi, & Hermetz, 2013) is a research area in which collaboration allows to design processes being made by multidisciplinary teams. Although there is no direct relation between this work and our proposal, multidisciplinary teams can be seen as our system's type of users whose interests may diverge. Many other research studies cited in this text try to solve UCTP, but none of them take collaboration opportunities into account.

3. Basic concepts

The university course timetabling problem (UCTP) relates to a number of classes, a number of professors and their constraints, a number of classrooms with their features, a number of students and their desired classes, a number of an institution's business rules which must be respected, and a number of time slots to be filled. UCTP is an NP-hard problem (Babaei et al., 2015) which is often solved by heuristics. Papers proposed by S. & Turabieh (2008), and Yang & Jat (2011) applied genetic algorithms to this problem; other metaheuristics, such as Simulated Annealing, Tabu Search, and Harmony Search, are respectively used by Song, Sanya, Tang, Peng, & Chen (2018), Aladag, Hocaoglu, & Basaran (2009), and Al-Betar & Khader (2012).

Since there are more students than professors, and more professors than course coordinators, we assign in our approach weights to the suggestions depending upon the user type. In order to analyze the suggestion relevance, any user may approve, disapprove, or ignore a suggestion (like, dislike, or do not care). This information is used by heuristics to evaluate the timetable improvement.

[1] See http://www.waze.com

Let's suppose S is the total number of students, P is the total number of professors, and C is the total number of course coordinators. For a suggestion, suppose that we get s votes (number of likes minus dislikes) from students, p from professors, and c from coordinators. A suggestion weight varying between [0.1] can be calculated by:

$$\text{Suggestion weight} = (s/S + p/P + c/C) / 3$$

In this work, we do not consider classroom features since, in our institution, all classrooms are the same (i.e., classrooms have a fixed number of students.)

As said in section 1, UCTP deals with a lot of constraints. In order to allow the generation of feasible timetables, those constraints are grouped according to their relevance. Hard or strong constraints mean that requirements must be respected. For instance, a professor cannot be allocated to two classes on the same time and day of the week. Weak or soft constraints make up requirements that heuristics will try to respect. For instance, a professor may point out which time slots should not be allocated to him. When a constraint is not satisfied in a timetable, it acquires a penalty. A successful timetable acquires the smallest number of penalties.

Table 1 lists some hard and soft constraints. Due to lack of space, we listed just a few. Our model consists of 35 constraints regarding students, professors, classes, and suggestions.

Table 1

Main hard and soft constraints

Constraint	Type
Time slots in which a professor cannot be allocated.	Hard
A professor cannot be allocated to two or more classes on the same day and time.	Hard
Maximum number of classes students can attend.	Soft
Maximum number of available classrooms.	Soft

Elaborated by the authors

4. Prototype description

Figure 1 depicts the prototype's architecture. This text regards the web module (Módulo WEB) in which all interactions are made. Said module receives suggestions from Internet-subscribed users and print out system results. The finess module (Módulo Finess) runs heuristics in order to: (I) generate an initial timetable version based on users' constraints, and (ii) evaluate users' suggestions and propose a better timetable. Communication between finess and web modules is performed using JSON (JavaScript Object Notation) format for flexibility.

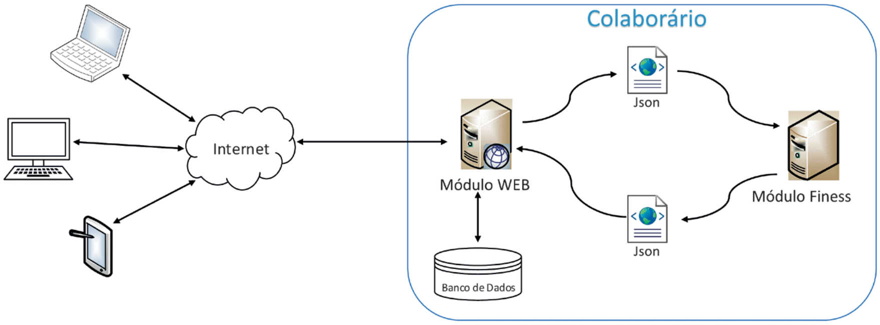


Figure 1. Prototype's architecture

Elaborated by the authors

Data used by the web module is recorded in a database (Banco de Dados in Figure 1). The database model is presented in Figure 2, where students (table alunos) and professors (table professores) are types of users. Both may give suggestions (table sugestoes) which refer to a particular allocation (table horarios_gerados). A professor may inform time constraints, which are stored at the table penalidades_professores. The table disciplinas contains information about the courses that are offered. Professors, students, and courses are aggregated in the classes (table turmas). In our case, some classes have fixed timetables. Those classes are recorded in the table horarios_fixos. All other tables are relationships between entities.

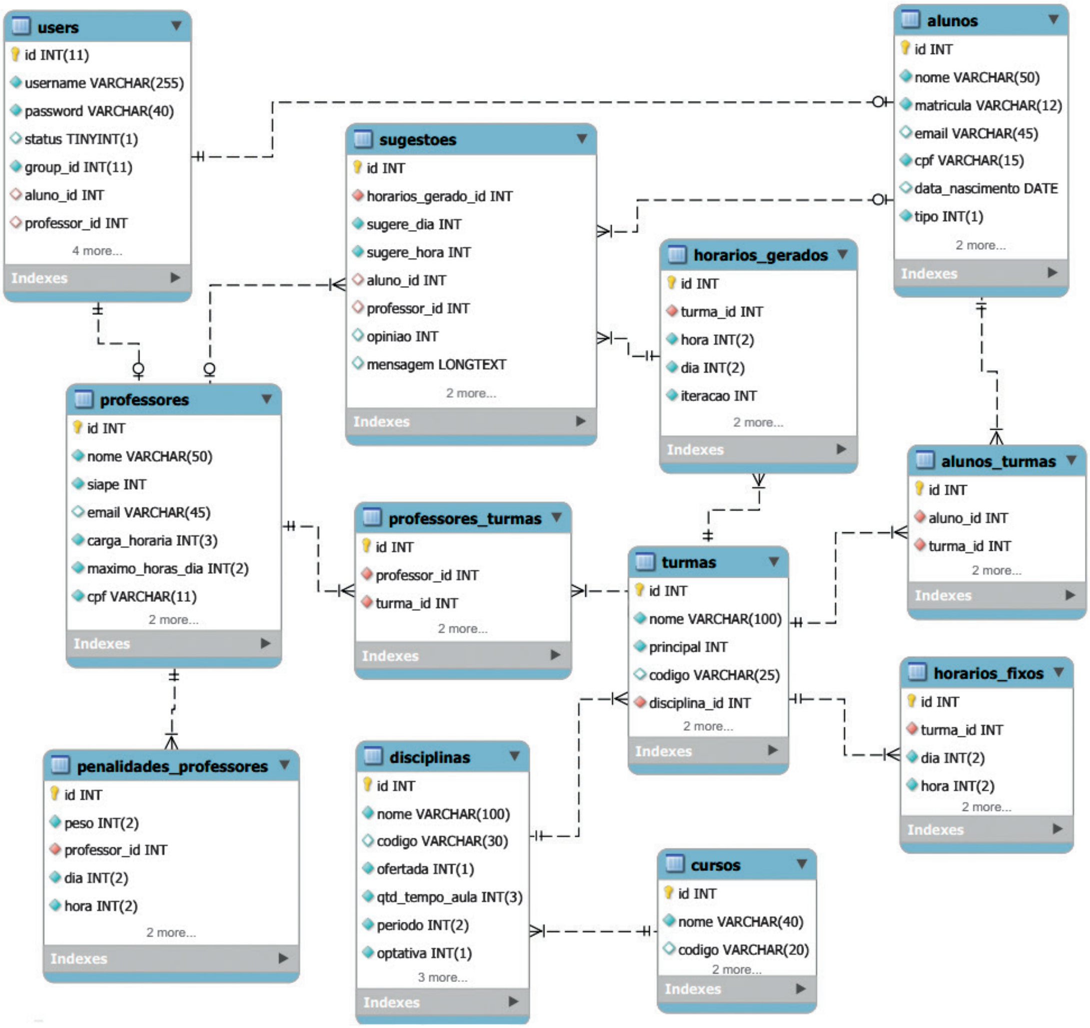
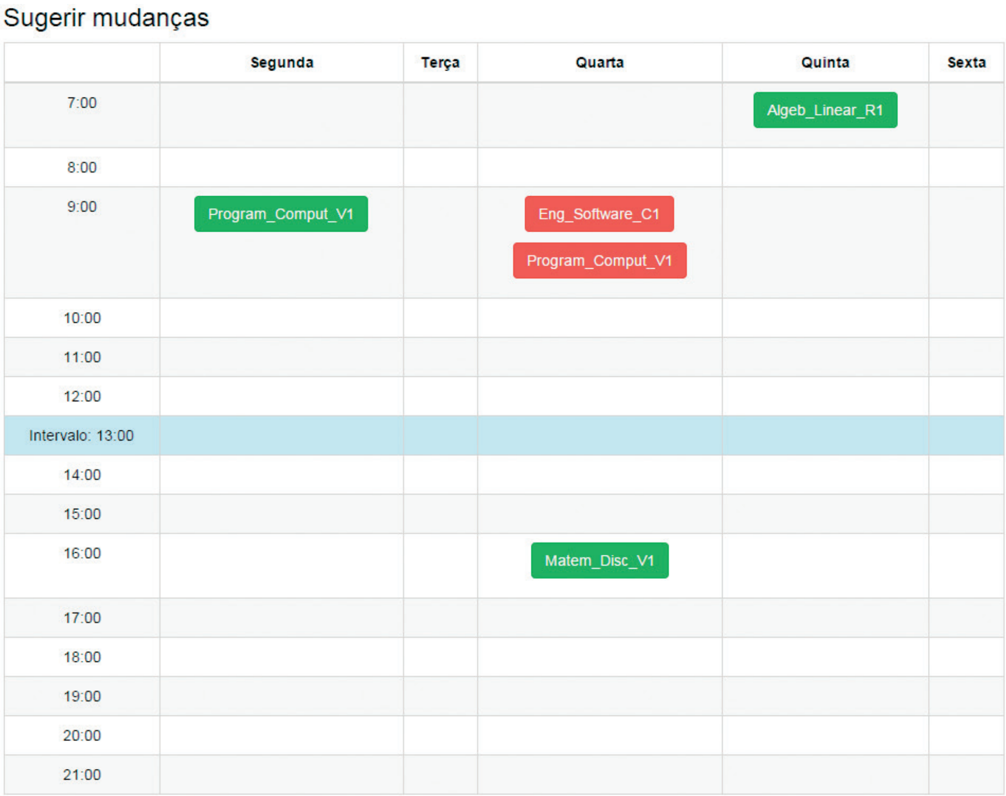


Figure 2. Web module database model
Elaborated by the authors

Regarding the prototype user interface, users may view their classes like in Figure 3 (some classes were omitted for better comprehension). At this figure, a user sees a conflict among the classes he/she desires to attend on Wednesday at 9 am: Eng_Software_C1 and Program_Comput_V1 (red rectangles). Then, they can click on one of them to suggest a change.

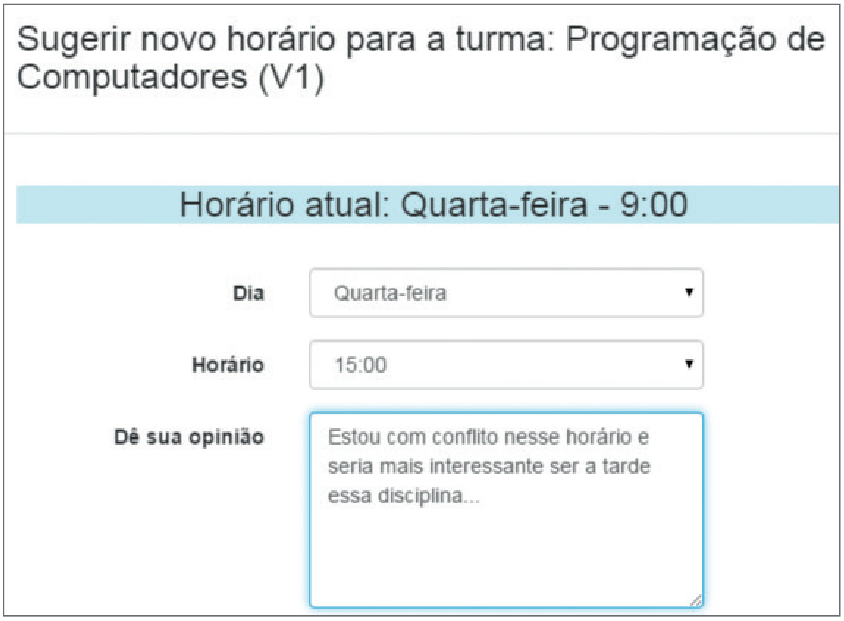


Sugerir mudanças

	Segunda	Terça	Quarta	Quinta	Sexta
7:00				Algeb_Linear_R1	
8:00					
9:00	Program_Comput_V1		Eng_Software_C1 Program_Comput_V1		
10:00					
11:00					
12:00					
Intervalo: 13:00					
14:00					
15:00					
16:00			Matem_Disc_V1		
17:00					
18:00					
19:00					
20:00					
21:00					

Figure 3. Classes window
Elaborated by the authors

Figure 4 presents a dialog where a user suggests an allocation time change. In this case, the user clicked over the Program_Comput_V1 class, and suggested moving the class from 9 am to 3 pm on Wednesday. Besides requesting the allocation time change, the user can justify his/her suggestion (Dê sua opinião at the suggestions dialog) in order to gather supporters (users approving his/her suggestion).



Sugerir novo horário para a turma: Programação de Computadores (V1)

Horário atual: Quarta-feira - 9:00

Dia: Quarta-feira

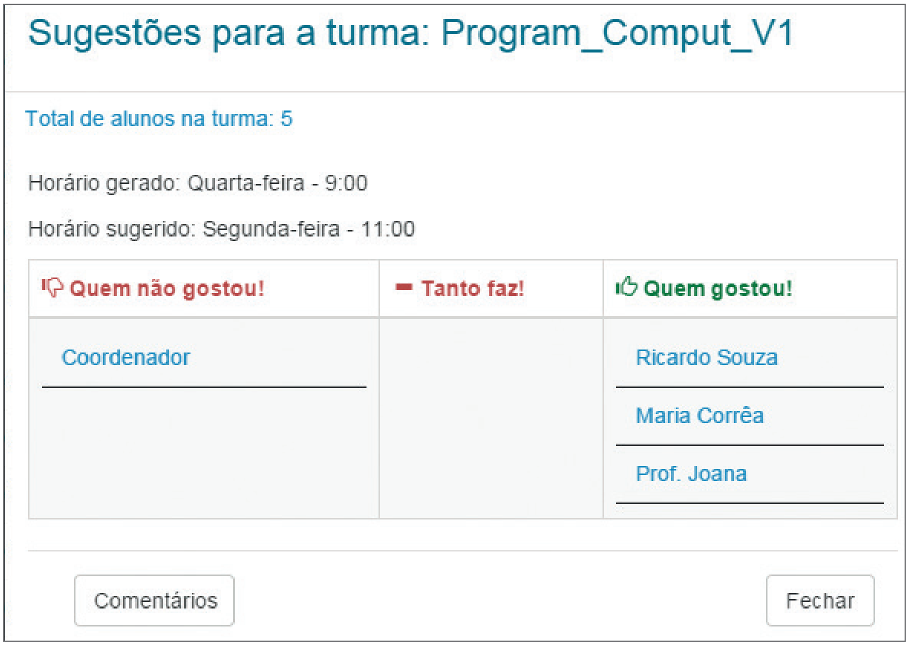
Horário: 15:00

Dê sua opinião: Estou com conflito nesse horário e seria mais interessante ser a tarde essa disciplina...

Figure 4. Suggestions dialog
Elaborated by the authors

In Figure 6, users can see a list of suggestions for each class they desire to attend. The Algeb_Linear_R1 class was scheduled at 7 am on Thursdays (Quinta-feira in the figure) and got a suggestion to be moved to 6 pm on Mondays (Segunda-feira in the figure). One user did not care about this suggestion, and another one liked it. The Eng_Software_C1 class was offered at 9 am on Wednesdays (Quarta-feira in the figure) and got a suggestion to be moved to 10 am on Thursdays (Quinta-feira in the figure). One user disliked this suggestion and no one else showed his/her opinion. Similar information is shown for the Matem_Disc_V1 and Program_Comput_V1 classes which got 2 and 7 suggestions, respectively. The fourth column of the table (Detalhes) contains a link for detailed information on each suggestion.

An example of detailed suggestion is shown in Figure 5. This class has a total of five (5) students. As seen before, the current allocated time was 9 am on Wednesdays (Quarta-feira in the figure) and a user suggested to move it to Mondays (Segunda-feira in the figure) at 11 am.



Sugestões para a turma: Program_Comput_V1

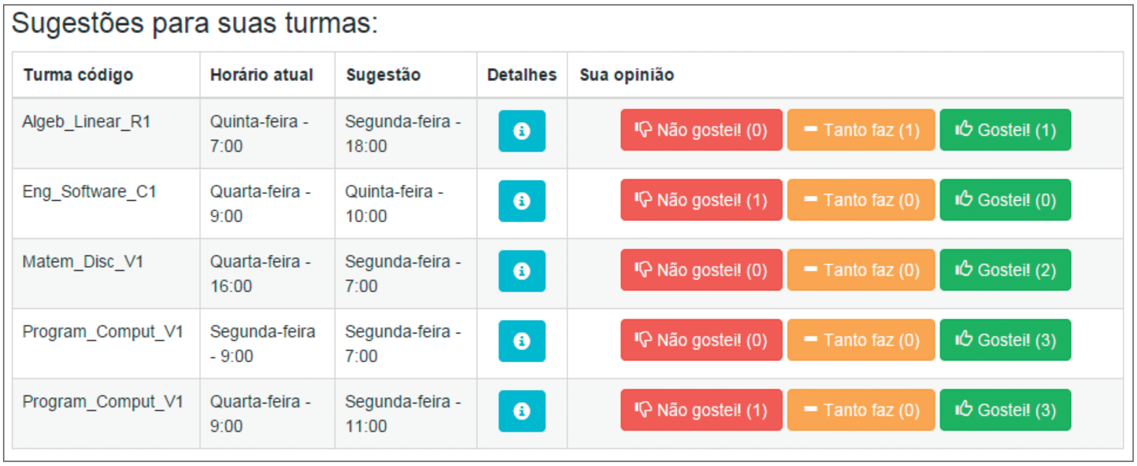
Total de alunos na turma: 5

Horário gerado: Quarta-feira - 9:00

Horário sugerido: Segunda-feira - 11:00

👎 Quem não gostou!	➖ Tanto faz!	👍 Quem gostou!
Coordenador		Ricardo Souza
		Maria Corrêa
		Prof. Joana

Figure 5. Detailed suggestion
Elaborated by the authors



Sugestões para suas turmas:

Turma código	Horário atual	Sugestão	Detalhes	Sua opinião
Algeb_Linear_R1	Quinta-feira - 7:00	Segunda-feira - 18:00	ℹ	👎 Não gostei! (0) ➖ Tanto faz (1) 👍 Gostei! (1)
Eng_Software_C1	Quarta-feira - 9:00	Quinta-feira - 10:00	ℹ	👎 Não gostei! (1) ➖ Tanto faz (0) 👍 Gostei! (0)
Matem_Disc_V1	Quarta-feira - 16:00	Segunda-feira - 7:00	ℹ	👎 Não gostei! (0) ➖ Tanto faz (0) 👍 Gostei! (2)
Program_Comput_V1	Segunda-feira - 9:00	Segunda-feira - 7:00	ℹ	👎 Não gostei! (0) ➖ Tanto faz (0) 👍 Gostei! (3)
Program_Comput_V1	Quarta-feira - 9:00	Segunda-feira - 11:00	ℹ	👎 Não gostei! (1) ➖ Tanto faz (0) 👍 Gostei! (3)

Figure 6. Suggestions report window
Elaborated by the authors

In Figure 5, there is one user who disliked the suggestion (Quem não gostou!): the Coordinator; three users who liked it (Quem gostou!): Ricardo Souza, Maria Corrêa, and Prof. Joana; and no one who did not care (Tanto faz!). Through this dialog, a user can also read all claims on the suggestion (Comentários).

Concerning technology and accessibility, nowadays mobile access is a must. Thus, this prototype was implemented in PHP with frameworks for responsiveness. Although PHP was adopted, any web technology could be used to implement these ideas.

5. Experiment

In order to evaluate the prototype, we conducted a three-day experiment. At the beginning, we had an initialization step. A system administrator entered all offered classes including their corresponding professors. Said professors registered their time constraints as time slots that should (weak constraint) or must (strong constraint) be avoided. Then, the students selected their desired classes. Finally, the course coordinator listed the classes with time allocations that could not be changed (university business rules). All these interactions were done using the web module. Our scenario had 53 classes and 20 professors, and 83 students took part in this experiment.

Next, the web module sent all input data to the finess module that used heuristics to generate an initial timetable. Then, the finess module sent back the initial timetable to be presented to the users through the web module.

Afterwards, an iteration cycle was performed. Suggestions were accepted by the web module during a day. At the end of each day, the web module sent the information to the finess module, which used heuristics to improve the initial timetable. Whenever a timetable was improved, the finess module sent back the new timetable to be analyzed by users on the next day.

In the first iteration cycle, 24 suggestions were received and just 4 were accepted. In order to include those suggestions, a lot of changes were done in the current timetable. During the second iteration, 6 suggestions were made and none was accepted. Although nothing was included, the current timetable was changed again, reinserting some allocations that were discarded in the first cycle. We guess that the execution time spent by heuristics may infer these changes.

6. Conclusion

In this work, we propose a novel approach for dealing with UCTP, which considers the collaboration of users who are affected by a timetable. This opportunity increases the timetable acceptance, since users can participate in the timetabling process. Some constraints may be hard to implement but can be inferred by users' suggestions. As far as we know, this is the first time UCTP has this kind of approach.

The prototype worked fine during the experimentation phase but requires some improvements before being released. The suggestions' weight is an issue that requires more research. Students who are at the end of a course have priority over the other students. A limit on the number of constraints provided by the professors must be set in order to allow better solutions.

Other functionalities such as the addition of suggestions (i.e., a way of suggesting a change in many time allocations at once) should be implemented. A user's reputation is another issue to consider, because it can engage other users. All professors took part but only a few provided precise constraints. Only 10% of the students participated and a quarter of them made suggestions during our tests. The educational institution must plan motivational actions in order to gain more contributors. Future works will also define a mathematical model of collaboration and the way it may change a heuristic behavior.

References

Abdullah, S., & Turabieh, H. (2008, November). Generating university course timetable using genetic algorithms and local search. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on* (Vol. 1, pp. 254-260). IEEE.

Al-Betar, A., & Khader, A. T. (2012). A harmony search algorithm for university course timetabling. *Ann Oper Res, 194*, 3-31.

Aladag, C. H., Hocaoglu, G., & Basaran, M. A. (2009). The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. *Expert Systems with Applications, 36*, 12349-12356.

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering, 86*, 43-59. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360835214003714 DOI: https://doi.org/10.106/j.cie .2014.11.010

Lima, T. J. B., Martins, C. B., Vianna, D. S., Dianin, M. F. V., & Costa, S. O. (2015). Utilização do método de análise hierárquica (ahp) para definição dos pesos de restrições fracas na re- solução de problemas de programação de horários para uma instituição federal de ensino superior. In *Simpósio brasileiro de pesquisa operacional* (pp. 1-12).

Pandey, J., & Sharma, A. K. (2016). Survey on university timetabling problem. In *3rd international conference on computing for sustainable global development (indiacom)* (pp. 1-64). IEEE. Retrieved from https://ieeexplore.ieee.org/document/7724247/

Parte, Y., Auroux, D., Clément, J., Masmoudi, M., & Hermetz, J. (2013). Collaborative optimization. *Multidisciplinary design optimization in computational mechanics*, 321-367.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review, 13,* 87-127.

Song, T., Sanya, L., Tang, X., Peng, X., & Chen, M. (2018). An iterated local search algorithm for the university course timetabling problem. (To appear in *Applied Soft Computing*).

Yang, S., & Jat, S. N. (2011). Genetic algorithms with guided and local search strategies for university course timetabling. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 41* (1), 93-106.