Data mining techniques for student timetable optimization

Milos Ilic, Petar Spalevic, Sinisa Ilic
Electrical and Computing Engineering
University of Pristina, Faculty of Technical Science,
Kosovska Mitrovica, Serbia
milos.ilic@pr.ac.rs, sinisa.ilic@pr.ac.rs

Mladen Veinovic
Singidunum University, Faculty of Informatics and Computing,
Belgrade, Serbia
mveinovic@singidunum.ac.rs

Zoran Milivojevic
Collage of Applied Technical Sciences,
Nis, Serbia
zoran.milivojevic@vtsnis.edu.rs

Bojan Prlincevic
Polytechnic School for vocational studies,
Zvecan, Serbia
b.prlincevic@vts-zvecan.edu.rs

Abstract — **Timetabling represents a complex problem that is a part of everyday human life and work. The key question is how to create the best timetable for some specific purpose? Student timetable scheduling is one of the challenging time consuming problems for all academic institutions. Big problem is how to automatically create timetable and optimize it. For this purpose data mining techniques can be used. Authors in this paper describe the key constrains for students timetable creation and propose data mining algorithm for timetable optimization.**

Key words - clustering; hard constrains; schedule; soft constrains; timetable;

I. INTRODUCTION

Timetabling represents a difficult optimization problem. This problem has a large range, and could not be solved efficiently with some conventional methods. More accurately, conventional methods solves timetabling problem, and such solutions are in use today, but conventional solutions are complex, not so effective, and what is most important, not an automated. With timetabling problem we are facing every day (high schools, faculties, banks, conferences, transportation and delivery). The general area of scheduling or timetabling has been the subject of intense research for a number of decades. The term scheduling covers all aspects of the activity of allocating resources and, at the same time, satisfying some predetermined objective. Scheduling problem covers a wide range, and because of that it becomes necessary to classify the scheduling problem into specialized activities such as timetabling. In the most general terms, timetabling can be described as the constrained allocation of resources to objects. This allocation depends largely on the set of constrains, and it is set so that the total cost would be minimal. Costs can be classified as financial cost, and time (temporally) costs. Time costs minimization is actually a timetable problem. Timetable construction is the allocation, subject to constraints, of given resources to objects being placed in space-time in such a way as to satisfy or nearly satisfy a desirable set of possible constrains [1]. Timetable optimization is a complex problem that occurs in the timetable creation process. For this purpose different optimization techniques can be used. Some of them provide good optimization results, but another are not so good. Recently techniques that have found a very good application in timetable optimization are techniques that fall within the domain of data mining. Data mining is a process of discovering various models, summaries, and derived values from a given collection of data. The ability to extract useful knowledge hidden in data and to act on that knowledge is becoming increasingly important. Data mining techniques can be used in timetable optimization process. Different techniques will provide different results.

For students timetable we can distinguish two different timetables. First represent courses or classes timetable, and the second represent exams timetable. For both timetables we can define different set of constrains, and these timetables can be created in different way. Here we propose optimization data mining technique for student classes timetable. Each institution have specific constrains, and based on that constrains institution creates classes and exams timetable. Constrains in all academic institutions are divide into two groups. First group represent so-called hard constrains, and the second group are so-called soft constrains. All hard constrains must be satisfied. In other hand soft constrains are more specific constrains, and in that way provides much better optimization. Soft constraints may not be all satisfied, but that is not a problem. As more soft constraints are satisfied timetable is better. To satisfy as many soft constrains, data mining techniques can be used. Research in the paper is in early stage, and here authors describe their idea of approach for timetable creation and optimization, based on artificial neural networks and clustering.

The second part describes hard and soft constrains. The third section is explanation of some optimization algorithms and in this section we described proposed data mining technique for optimization. The fourth section presents the conclusion, and in the end is list of references.

II. HARD AND SOFT CONSTRAINS

Organizations like universities and schools use timetables to schedule classes and lectures, assigning time and places to future events in a way that makes best use of the available resources. Classes timetable creation for universities are more complex process then for the schools, due to multiple constraints. In this way universities in particular increasingly have to deal with a large number of courses, and flexible degree structures [2]. A timetable which is not well designed will be inconvenient, and will be expensive in terms of wasted time and money. Dealing with classes timetable problem, requires assigning of time periods to each of the preset classes. For this assigning defined constraints must be fulfilled. Scheduling classes by rooms additionally requires rooms allocation for each class, and defined constrains for this task must be fulfilled. Assumption that is important too is that timetable is the same per week, so that each week in one semester has the same timetable. In some institutions we may have classes performed every second week in a double amount. Courses that are organized in that way could change timetable, a lot. One of the acceptable solutions for such courses can be to group two such courses in one day. In that way one professor will have whole day for classes every second week. If we have just one course, for such course we can assign all day. It that way students will have free day every second week. Here we will assume that the courses are performed every week. Courses that are performing every second week we will consider as the exception. For such courses human interaction is needed.

Workweek consists of several working days. Each working day is divided into a large number of time slots, and slots are same length. Classes need to be allocated exactly within defined slots. Duration for some classes may be greater than one time slot. Each class represent exactly one course, more precisely classes may be lectures or exercises. These are some of obvious constrains. One more assumption is that teachers and courses must be known in advance [3]. When the task is to create timetable for faculty certain problems may occurs. Some of problems are number of groups for lectures and exercises. In practice students groups for lectures are bigger then students groups for exercises, because of that we must have more groups for exercises. If we have multiple students groups for lectures and/or exercises arises a new problem. For this job more teachers and teaching assistants can be engaged to same course. Elective courses may be additional problem for timetable creation. The class timetable for students in the first year is usually simple, only limited to room, teachers and equipment constrains. But for the older year of study where only a few of the courses are mandatory and all others are elective, it is increasingly hard to find good solution for the timetables, and it is no more possible to create a static timetable valid for every year, with minor changes [4].

Depending on the number of students per courses timetable will be different from year to year. Even if it was possible to forecast the total number of students for all courses it would be still impossible to know, for each course, in advance if it would conflict other courses if the same students are enrolled to both. In fact the groups have increasing diversity, and in every semester it is different situation. For some courses that have been going on for years the total number of students per course can be forecast within some degree of accuracy. There are many courses that are new or changed or restructured within the study program. There are entities which are specific for individual schedule timetable. For example if we want to take into timetable the wishes of professors, assistants and students, it is necessary to define the entity that will represent the wishes. Timetables can allow scheduling of special lectures during the semester, or scheduling classes every other week, and the like. In the process of timetable creation certain dependencies need to be established. All we mentioned above are some constrains that need to be satisfied.

For better dealing with timetable constrains we could use meta-model of constrains. The meta-model must be specified by the user and describes the domain in which the timetable occurs. It describes the resources that are available and how they are organized, the type of events that can appear on a timetable, and the period and granularity of the time that the timetable may cover. For our model we will create a unit that represents teaching unit. For this teaching unit we have concrete group of students which listening, concrete teachers who teach and concrete classroom. We can represent our model in four dimensions, time, teachers, groups of students and classrooms respectively. Each part of our four dimensional area is restricted with classroom in which class are scheduled, with a set of teacher who teach, with a student groups that listens, and at the end with a time component. Check the availability of a term by all criteria boils down to checking whether a particular piece of space timetables is empty.

For each timetable all constrains we can classified in two types. First are constrains that must be fulfilled and constrains like these are named hard constrains. A hard constraint is a constraint that if broken for a timetable, that timetable is not a valid solution. For example, scheduling a teacher to be in two different locations at the same time would normally be considered a hard constraint. The second type of constrains are constrains that should be fulfilled. This type of constrains are named soft constrains. A soft constraint is a constraint that it is desirable not to break, but if done so does not stop the timetable from being a valid solution. It merely harms the perceived quality of the timetable. Hard or mandatory constrains conditioning the correctness of timetable.

Timetable correctness is not conditioned with soft constrains, but soft constrains defines some comfort level for the teaching staff and students. It should be noted that the same constraints for one institution can be hard constrains, and for another same constrains can be soft.

Constrains that we have mentioned above are not only one. Following are the list of hard constraints.

- Every class must be scheduled exactly once.
- In one room can be held classes just for one lectures or exercises from one course in the same time.
- Room where classes are taught must fulfill some requirements of the course, such as the number of seats, the existence of the projector, table, smart boards, etc.

- The student must be able to attend to all lectures and exercises for all courses that he listens.
- Students can be divided into groups for lectures and exercises.
- Teachers, teaching assistance and students can only be in one place at the same time.
- More teachers or teaching assistants can be engaged on same course.
- Teachers and teaching assistants may not be accessible in some part of the semester, or in some part of the week.
- The teacher can't teach two different courses at the same time.
- It is possible to request that the teacher teaches the same course to more than one group of students.
- Some classes need to be held consecutively. For example the Labs.
- Some classes require particular rooms like experiments must be held in particular laboratories.

Like we said above constrains may be different for different institutions. This is more obviously for soft constrains than for the hard constrains. Although the soft constraints are quite different in institutions, here we can try to distinguished some class of such constrains, which are frequently encountered. Soft constrains are less straight forward to define. Soft constraints however can vary in severity.

Constraints must be tagged to inform the evaluator how important it is that they are not broken so that the evaluator can take this into account when assessing the quality of the timetable. Usually, these constraints must be fulfilled as well as possible [5]. The timetable that violates these constraints is still usable, but it is not convenient for either students or teachers.

Following are the list of soft constraints:

- Teachers may prefer specific rooms, or may require that their all lectures are in same room.
- Teachers may define terms for consultative teaching for each course within the timetable.
- Teachers can define the terms in which they cannot teach.
- Some students groups may have forbidden terms (for example fourth year students should not have classes in terms of some seminars).
- Different year students can listens some course at the same time together.
- Some teachers require lectures at the same time as their colleagues (for example if they want to organize meetings before or after lectures in those days).
- If the student workload is not large, it may be required to have a day off in a week.
- Professors can require that have at least two hours lectures in the days in which they have classes, because they do not want to come to college every day.
- Some professors do not wish to have classes assigned consecutively in time.
- Most students and some professors do not wish to have empty periods in their timetables.
- Classes should be distributed evenly over the week.
- In institutions that work in shifts, sometimes it requires that teachers do not work in different shifts during the day.

Soft constraints are constraints which may be broken, but of which breaches must be minimized. Soft constraints are in fact preferences which should be followed. The materialization of any soft constraint adds more feasibility to courses schedule, and get more desirable schedule. As it can be easily seen, to get a complete solution for a particular timetabling problem with all of the constraints satisfied is very difficult, possibly even impossible to accomplish. The best practice for timetable automatic creation should be solving all hard constrains, and as much as possible soft constraints.

III. TIMETABLE OPTIMIZATION

In some cases timetabling problem can be reduced to find any solution that satisfies all specified constrains. Case like this is named searching problem. In all other cases timetabling problem is formulated as optimization problem [6]. In some papers optimization techniques are applied to the searching problem. It is notice that models that are presented in similar studies are not universal. Despite the large number of papers, some problems have not been solved. It is difficult to compare the efficiency of these algorithms, and most of them have been developed for a specific problem. In both cases basic defined problem is decision problem. For troubleshooting scheduling classes, can be used a large set of different problem approaches, and a large set of different algorithms. In general, the approaches employed to solve timetabling problems are similar in their nature and are applicable and compatible. The approaches proposed to solve the timetabling problem are categorized with regard to different criteria such as heuristics [7]. Mathematical and linear programming techniques in many ways by the literature and divided in various types [8]. Some types are:

- Particle swarm optimization.
- Ant colony optimization.
- Genetic approaches.
- Reasoning approaches.
- Mathematical programming approaches.
- Simulated annealing.
- Methods based on graph coloring algorithm.
- Constraint logic programming-CLP.

- Artificial neural networks.
- Algorithms based on data mining techniques.

Each of these optimization techniques can be used for timetable optimization. For example mathematical programing approaches for timetabling are very precise, because of that these approaches are in use just for specific applications based on papers [9,10]. This means that this model provides good timetable, but it is not universal, and because of that each institution must have own algorithm. In contrast to this approach our idea is to create model and software that could use all institutions. That means that our approach provides universality. In some papers object oriented timetable modeling and optimization is presented. This approach creates framework for class timetables. Here could be used different programing languages for implementation. In the paper [11] for one model different locally searching techniques are implemented. Research in this paper shows that small model changes could provide different algorithms implementations. This is very good approach, and provides good results. We use object oriented approach to represent constrains dependency, resource dependency and connections between atomic units.

Each of these techniques can processed some of timetable constrains. For better timetable, which fulfills as much as possible constrains some of these optimization techniques may be combined into more complex structure. We can use good sides from multiple techniques, and create some hybrid technique. Data mining has some very good techniques that can be used for timetable optimization. One data mining technique that can be used for this job is clustering. Clustering is the process of finding classes of objects that share common characteristics. Fundamental assumption that we make when using clustering is cluster hypothesis which reads “documents in the same cluster behave similarly with respect to relevance to information needs” [12]. Clustering for timetabling can be used alone or we can combine clustering and artificial neural networks. We propose hybrid method that is combination of artificial neural networks and clustering algorithms. In the first phase after all hard and soft constrains for some specific institution are written down we can start with artificial neural networks creation, and we can uses this networks to optimize timetable as much as possible.

Artificial neural networks are still little studied methods for troubleshooting timetables. For timetable optimization three models of artificial neural networks can be used. First method is *Iterative Activation and Competition – ICA.* This model uses hybrid form of artificial neural networks, introducing a structure of artificial intelligence for task scheduling to the model of neural networks. This model structure organizes lessons sequentially, starting from the region in the network with the largest number of constrains. In the same time network finds and configure the parallel resource combination that are most suitable for a given class. Here we must pay attention to the mutual interaction of complex constraints. Constrains can be adapted by weight of synapses, before being elected the next class that needs to be scaled. This model provides the amazingly slow growing of the network when the size of the problem increases. In other hand model provides flexibility in constrains coding. The second model of artificial neural networks that can be used for timetables optimization is *Potts* neural networks model [13]. In this model we can use neurons with more states instead of neurons with two states. With this benefit we can provides uses of a smaller number of neurons. With this model we can process a large number of hard constrains, and maybe some soft constrains. For example if we take that L represents number of classes that must be allocated in P time periods and R classrooms, it can be seen that for that jab are needed X(LxP) neurons for classes and Y(LxR) neurons for classrooms.

Third model or third approach, are modified *Hopfield* artificial neural networks. These networks have the potential for fast computations when there are implemented in hardware. The original *Hopfield* model must be modified in the form to provide balance between constrains, to provide fast looking for a more precisely locally minimum (immediately after the first few iterations of the network) and installation of stability mechanism that should stop lowering of the local minimum at an early stage. Results obtained by using these artificial neural networks can be compared with best heuristics methods for timetabling problem. Artificial neural networks showed that these networks can produce quality solutions for students/courses timetabling problem. These models can be used for solving not only equal size and complexity problems but also more extensive problems. With artificial neural networks we can processed almost all hard constrains. A timetable that was obtained as a result of neural networks output is very good and in that form timetable can be used. In other hand some specific hard constrains and some soft constrains are still not fulfilled, and for this purpose human interaction is needed again.

The second step or second method in our proposed hybrid model is clustering. With clustering we want to reduce human involvement in timetable creation. With artificial neural networks we got timetable that fulfilled most hard constrains and some of soft constrains depending of the specific institutional constrains. Like we said above soft constrains are more specific and there are less straight forward to define and fulfill. We can use clustering for errors discovery in generated timetables, and to fulfill more specific soft constrains. For that purpose we need to have students database. Some information like which student has chosen which courses, or in which years course is present are needed for clustering.

The key idea behind clustering is to create clusters of courses for specific year of study, students who have chosen that course and cluster for atomic units that represent one time slot. This atomic unit contains information about course, students, professors, teaching assistance and classroom for each time slot like we said above. For this task we can use k-means clustering or hierarchical agglomerative clustering algorithms. K-means objective is to minimize the average squared Euclidean distance of elements from their cluster centers. The ideal K-means cluster is a sphere with the centroid as its center of gravity. Ideally, the clusters should not overlap too. For timetable that means that each cluster represent course, students and professors in one time slot, depending for which criterion clusters are created. If clusters do not overlap, time slots, students and professors for that lectures or exercises will not be blocked, and they can teach or listen selected courses in that time slots. In that way we can find errors in timetable that is created in the first step with neural networks, and remove them. Hierarchical agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criterion is met. The various clustering criterion functions can be used to determine the pairs of clusters to be merged at each step. Here we must define good clustering criterions based on soft constrains.

Depending on the soft constrains some clusters will be merged and some will not. Initial course clusters will be merged if for example there are not in the same time, in the same classroom, or if professor or teaching assistance for that course are not blocked. Algorithm is based on two steps. Both steps have large complexity. The first step is the computation of the pairwise similarity between all the clusters in the data set. The second step is the repeated selection of the pair of clusters that best optimizes the criterion function. A naive way of performing that is to re-compute the gains achieved by merging each pair of clusters after each level of the agglomeration, and select the most promising pair. This method builds the hierarchy from the individual elements of the progressive merging clusters. Soft constrains here can be implemented easily. In this merging first step is to determine which elements should be merged into a cluster. In both cases we must modify original clustering algorithm and adapt them for timetable optimization. Like we said above modifications are reflected in clustering criterions for merging or centroid properties. These criterions must be based mainly on soft constrains. With artificial neural networks and clustering timetabling will get new dimension, and per the first time this process can be automated.

IV. Conclusion

Educational institutions employ and educate a great number of people. All of them have more obligations besides education. Educational quality depends of many facts. One fact is good courses timetable. Each institution must provide the most quality timetable. This is the key problem every year, for each faculty when the semester starts. More precisely this is the headache for employee whose job is to create courses timetable for each academic year. In many institutions employee manually create timetable. Because of manually creation process in which human is the key factor many errors can arise. Beside human errors greater number of problems can arise when employee try to fulfill all the individual desires of professors or teaching assistances. Because of all that errors and due to the real needs for quality timetable that will be good both for students and for professors we trying to create software for automated timetable generation.

Here we described some of the key fact that one automated process for timetabling must fulfill. These includes hard and soft constrains and mandatory optimization process. Optimization is very important for timetabling. Optimization provides compact timetable without for example blocked professors, or duplicated classrooms for same time slot etc. All these are very important facts for the process of timetable creation. Hybrid model that is proposed here and which consists of artificial neural networks and clustering will provide good optimization for timetable. This model combines good properties of both neural networks and clustering. The next step in our research is to implement proposed model based on constrains that are specific for some faculty or some high school and to use our model to optimize created timetable. For this purpose we first plan to define formal language for constraint representation. Some parameters for formal language are mentioned earlier, and those are atomic units, group of teachers, set a resources group, set of time slots, and connections set. Time group defines free terms, and time slot allocation during the week. Resource groups can be divided in to smaller groups. It is much easier to work with smaller groups of resources. Resources can be divided by some properties, for example based on that what concrete resource represent. Connections are represented with slots that should be assigned to different elements form resource groups.

As for the artificial neural network we plan to implement Hopfield Neural Network based on research in [14]. The first step to map an optimization problem with this artificial neural network model is to express in a single line costs and restrictions of the problem in penalty terms. Then, it is starting to define the function of the energy network, making sure they are properly determined the weights of the network and bias element. The changes that will be made to the original model of the Hopfield network is attached in finding the balance between the penalty terms, in search of more precise local minima, randomness in the network and the incorporation of the mechanism of stability, which is to stop an early descent, in the case of a local minima has been found. The results that we expect using the approach of the modified Hopfield Neural Network will proved it comparable with the best technical heuristics for the solution to the problem. In literature, the network was shown capable of producing high-quality solutions to problems of time allocation extremely difficult. This method is also superior to the other two models mentioned above, because to deal with problems of equal size and complexity as well as larger problems.

In the structure that is provided by neural networks some conflicts can arise. For conflicts resolution we will use clustering. All courses will be clustering in clusters which significantly have the similar characteristics based on their properties (year of study, mandatory or elective and etc.) and intake. Each of the clusters will have a group of courses and will be represented by a unique group. Once all subjects have been grouped in their designated clusters, they will be sorted according to the numbers of student enrolment. Based on number of students per course, rooms will be assigned to specific course. In [15] clustering is in use for general purpose, to satisfied hard event schedule per time slots. In our future research we will use clustering algorithms to find overlapping by the rooms, teachers and students that are assigned to some specific time interval. Especially clustering will provide solution for specific soft constrains and elective courses. For elective courses student database must be used. From it we will clustering students by the course that they listening, and then see if we have conflict for some student or students.

We hope that with proposed model like we said above we will fulfill all hard constrains and as much as possible soft constrains that are specific for some academic institution.

ACKNOWLEDGMENT

This paper is result of collaboration with the Ministry of Education, Science and Technological Development of Republic of Serbia within the projects TR 35026 and TR 32054.

REFERENCES

[1] S. Shatnawi, F. Albalooshi, and K. Rababa, “Generating Timetable and Students schedule based on data mining techniques”, International Journal of Engineering Research and Applications, Vol. 2, Issue 4, pp. 1638-1644, 2012.

[2] K. Kumar, Sikander, R. Sharma, and K. Mehta, “Genetic Algorithm Approach to Automate University Timetable”, International Journal of Technical Research (IJTR), Vol 1, Issue 1, pp. 47-51, 2012.

[3] A. Stojnev, “Modeling the problem of generating classes schedule at faculties”, Graduate thesis, Department for software engineering, Faculty of Electronic Engineering Nis, 2014.

[4] V. Ajanovski, “Integration of A Course Enrolment and Class Timetable Scheduling in a Students Information System”, International Journal of Database Management System (IJDMS), Vol. 5, No. 1, pp. 85-95, 2013.

[5] J. Kaur, and A. Kaur, “Timetable Scheduling using modified Clustering”, International Journal of Research in Information Technology (IJRIT), Vol. 2, Issue 7, pp. 1-8, 2014.

[6] I. Aljarah, A. Salhieh, and H. Faris, “An Automatic Course Scheduling Approach Using Instructors’ Preferences”, International Journal of Emerging Technologies in Learning (iJET), Vol. 7, No. 1, pp. 24-32, 2012.

[7] M. Carter, and G. Laporte, “Recent Developments in Practical Course Timetabling”, In the proceed of the Practice & Theory of Automated Timetabling conference, pp. 3-19, UK, 1998.

[8] S. Daskalaki, T. Birbs, and E. Housos, “An Integer programing formulation for a case study in university timetabling”, European Journal of Operational Research, Vol. 153, Issue 1, pp. 117-135, 2004.

[9] M. Kaviani, H. Shirouyehzad, and M. Sahadi, “A mathematical model for university course timetabling problems by considering multi functions”, International journal Modelling in Operations Menagement, Vol. 3, Nos. ¾, pp. 282-295, 2013.

[10] A. Gunawan, K. Poh, and K. Ng, “A mathematical programming model for Timetabling Problem”, Proceedings of the 2006 International Conference on Scientific Computing, Las Vegas, Nevada, USA, June 26-29, 2006.

[11] S. Tan, “The Complexity of Timetable Construction Problems”, International Journal of Informational Technology, Vol. 9, No. 1, pp. 31-46, 2003.

[12] M. Ilic, P. Spalevic, M. Veinovic, Z. Milkic, E. Pecanin, and D. Rancic, “Clustering Algorithms in Data Mining”, Prodcessing of 58th Conference for Electronics, Telecommunications, Computers, Automatic Control and Nuclear Engineering - ETRAN, RT 5.1, Vrnjacka banja, Serbia.

[13] M. Carraso, and M. Pato, “A Potts Neural Network Heuristic for the Class/Teacher Timetabling Problem”, Metaheuristics: Computer Decision – Making, pp. 173-186, 2003.

[14] A. SMITH, D. ABRAMSON, and D. DUKE “Hopfield neural networks for timetabling: formulations, methods, and comparative results” Computers & Industrial Engineering, Vol. 44, pp. 283-305, 2003.

[15] T. Muller, “Constraint-based Timetabling”, Ph.D. Thesis, Faculty of Mathematics and Physics, Charles University in Prague, pp. 13-22, 2005.