Decomposing the High School Timetable Problem

Christos Valouxis[1,4], Christos Gogos[2,4], Panayiotis Alefragis[3,4] and Efthymios Housos[4]

[1]Secondary Education Office, Patras Branch, Greece

[2]Department of Finance and Auditing, Technological Educational Institute of Epirus, Psathaki, Preveza, Greece

[3]Department of Telecommunication Systems and Networks, Technological Educational Institute of Mesolonghi, Varia, Nafpaktos, Greece

[4]Department of Electrical and Computer Engineering, University Of Patras, Rio Patras, Greece

Abstract: The process of timetable construction is a common and repetitive task for High Schools worldwide. In this paper a generic approach is presented for Greek High Schools organized around the idea of solving a significant number of tractable Integer Programming problems. Variables of the underlying mathematical model correspond to daily teacher schedules while a number of hard and soft constraints are included so as for the model to handle practical aspects that manifest themselves in Greek High Schools. By selecting better teacher schedules that exist in sub-problems the quality of the overall solution gradually improves. The collected results which are obtained within reasonable time are most promising. The strength of the approach is supported by the fact that it managed to find the best known results for two public instance problems included in the Benchmarking Project for High School Timetabling (XHSTT-2012[1]).

Keywords: high school timetabling, integer programming

1. Introduction

Timetabling problems manifest themselves across various domains of practice and research and can be described as the task of allocating resources to available time slots so as a set of constraints are satisfied. Additionally, a set of quality features renders alternative timetables superior or inferior with respect to each other. Timetabling problems in their general form belong to the class of NP-complete problems giving little hope of finding an algorithm that produces an optimal solution in polynomial bounded time. Nevertheless, specific timetabling problems with great practical interest can be solved satisfactorily. Therefore, a wealth of solution approaches has been studied originating mainly from Mathematical Programming, Computational Intelligence and Metaheuristics.

[1] http://www.utwente.nl/ctit/hstt/archives/XHSTT-2012/

Educational Timetabling problems are a specialization of timetabling problems. They have been studied in detail and even before the new millennium a plethora of approaches have been proposed as can be consulted in (Schaerf, 1999). The volume of papers published since then about the subject has been increased steadily. Educational Timetabling problems can be broadly classified in three main types:

- High School Timetabling problems: Sets of students forming classes have to be scheduled in teaching hours involving specific availability and specialization of teachers so as to generate a feasible and balanced timetable. The underlying model of High School Timetabling has been formulated as an edge coloring problem on a bipartite graph. Under this formulation the abstract problem can be polynomially solvable. Nevertheless, the addition of real life constraints makes it NP-complete. The edge colouring formulation of the problem can be traced in (Csima, 1971) and (Bondy and Murty, 1976) with the latter having references to (de Werra, 1970) and (Dempster, 1971).
- University Course Timetabling problems: This problem can be seen as a specialization of the previous type with the difference of students that can belong to more than one classes. The main objective is to minimize lecture overlaps involving the same students.
- Examination Timetabling problems: This problem involves exams undertaken by sets of students and the goal is to reduce instances of students having to take part in more than one examinations simultaneously and evenly spread the exams over the whole exam period.

A significant number of papers about the High School problem in general and the Greek variant of it in particular have been published. Some representative papers are (Valouxis and Housos, 2003), (Burke et al, 2007), (Beligiannis et al, 2008), (Birbas et al, 2009) and (Liu et al, 2009). In this contribution we address the Greek High School Problem through a two phase process that incrementally solves parts of the problem. A feasible solution is progressively formed using solutions obtained through mathematical programming. Then, better solutions are located by keeping parts of the schedule fixed and changing the remaining schedule. The process gradually converges to better schedules. A similar methodology has been applied by our team for the nurse rostering problem with exceptionally good results. The interested reader is forwarded to the related paper (Valouxis et al, 2012).

The paper is structured as follows. Section 2 presents the problem as it occurs in Greek High schools. Constraints are categorized in hard and soft ones and a case study of a specific High school is documented. Section 3 gives the details of our approach by presenting the mathematical model used in the two phases. Then both phases are analyzed. Section 4 discusses implementation issues of our approach. Section 5 analyzes the experimental results obtained. Finally, section 6 provides conclusions drawn from our approach.

2. Problem Description

In Greek High schools, students have to attend a three year curriculum and in each year students are grouped in class sections alphabetically. Lectures are given from Monday to Friday and the teaching daily hours must be six or seven without idle hours in between for all students. The lectures that must be given to every class are predefined and common across Greece. Most of the lectures are given to students belonging to the original class sections by a single teacher for each subject. However, class sections can be split so as each group to attend a different course or to be merged with groups from other class sections in order to form temporary class sections attending certain courses. This usually occurs in courses like foreign languages where class sections are partitioned subject to the knowledge level of the students or subject to each student preference amongst the foreign language options provided by their school (usually English and French but also other languages in some schools). Class splitting also occurs for lessons that have to be attended in rooms with smaller capacity than that of the full number of a class section. Such courses are chemistry, technology and informatics which take place in laboratories. In order to have a full day schedule for all students of a class section that is split in two groups for a lesson like chemistry the first group might attend chemistry while the second group might attend different lesson like gymnastics or career guidance. Then at another time slot courses taught to each group appear to be swapped. Every teacher is qualified to teach a subset of the courses while the teaching load is different across teachers depending on seniority, motherhood and other conditions. It is possible for a teacher not to teach exclusively in a specific school and to have to complement his weekly work duty by teaching to more than one school. If that is the case then he is available for teaching to each school for certain days only.

In the model of the problem examined every class has its own predefined room but certain rooms or other resources like video projectors might be shared by more than one classes. Teachers are assigned for certain hours to each class section before the creation of the timetable. As, usually is the case in combinatorial optimization problems two categories of constraints can be identified: hard constraints and soft constraints. Hard constraints must hold for a solution to be considered valid while soft constraints violations degrade the quality of the proposed solution. Hard constraints of the problem are the following:

- HC1: Each teacher should teach a specified number of hours to each class.
- HC2: Each teacher should teach only at days that he is present at school.
- HC3: Each teacher should teach only to one class per time period.
- HC4: Each teacher should teach at least one hour for the days that he is present at school.
- HC5: Each class should have no more than one lesson per time period.
- HC6: Lectures of the same lessons should not be positioned in the same day.
- HC7: Empty time periods for each class section should only be positioned at the last period of each day.

- HC8: A resource should not be consumed more times than its availability in each time period.

On the other hand soft constraints are:

- SC1: Each teacher should have the smallest possible number of idle periods in his schedule. An idle period for a teacher is defined as a not busy period with a busy period earlier in the day and another busy period later in the day.
- SC2: Each teacher should have a balanced daily work schedule.
- SC3: Teacher preferences for teaching in early or in late hours should be respected to the highest possible degree. This constraint can also serve as an indirect way of giving preference to certain courses to be positioned in early or late time periods (e.g. Mathematics in the early hours).

Throughout the paper a specific problem instance depicted in Tables 1 and 2 will be presented and analyzed so as to better grasp the underlying concepts of the problem. This problem comes from the “3nd Gymnasium” High School in the city of Patras, Greece for the period from September 2010 to June 2011. It is about a medium size by Greek standards school with 9 classes, 29 teachers giving 340 lessons either to entire classes or to classes formed from student groups coming from different classes. Each class has to attend 7 teaching hours per day. For those teachers that have preferences for teaching early or late hours in some days this is marked in Table 1 with the letter E or L respectively. The letter X implies that the teacher is not available in this day. When a teacher prefers early teaching in a day this means that the solution will be penalized if he has to teach in hours 5 to 7 of this day. Likewise when a preference is late teaching assignments in hours 1 to 3 gets penalized. Furthermore, each teacher should have a balanced scheduled across the days that he is present at school. So lower and upper bounds, shown under column “Low” and column “High” in Table 1, are imposed. Columns A1A2, A3A4, B1B2, C1C2 and C2C3 denote class formations generated by joining groups of students between classes. For example, teachers T15 and T16 teach to mutually exclusive groups of students from classes A1 and A2 subject “English Language” for 3 hours each week.

	Week					Hours per day		Classes														
Teacher	M	T	W	T	F	Low	High	A1	A2	A3	A4	A1A2	A3A4	B1	B2	B1B2	C1	C2	C3	C1C2	C2C3	Total
T0	E	E	X		E	1	1		2		2											4
T1	L	L	L	E		1	2										2	2	2			6
T2	E	L	E	E	X	2	4	2	2	2	2			2	2							12
T3	L	E		L	E	2	4		3	10	2											15
T4		E	L	L	L	2	4										3	12				15
T5	E		E	E	L	2	4	2							5		9					16
T6		L	L	L	E	2	4		2						4				9			15
T7	E			L		2	4		5		3			9								17
T8		E	E	E	E	2	4	8			5								3			16
T9	X				L	1	1								4							4
T10	L	L	L		L	2	4							4			4	4	4			16
T11	L	L	L	L		2	4	4	4	4	4											16
T12	E	L		L	E	2	4	2		2				3	3		1	1	1			13
T13	E	E	E		L	1	3							2	2		2	2	2			10
T14	L	E		X	L	3	5	2	2	2	2						2	2	2			14

T15	L	L	L	E		1	3					3	3			2			2	2		12
T16	E		E	L	E	2	4					3	3	2	2	2			2	2		16
T17		E	E		E	2	4	4	4	2	2			2							2	16
T18	E	L		X	E	1	3				2				2		2				2	8
T19	L	L	E	E		1	2			2	2				2							6
T20	X	X	L	X	L	1	3		2								2					4
T21		X	X	E		2	4	1	1	1	1			1	1		1	1	1			9
T22	L	X	L	L		3	5	1	1	1	1			2	2		2	2	2			14
T23		L	L	E	L	3	5	3	3	3	3			3			2	2				19
T24	E	E	E	L		1	1								3				2			5
T25		E		L	E	1	3	1	1	1	1			1	1		1	1	1			9
T26	L		E	E	L	2	4	2	2	2	2			2	2		1	1	1			15
T27	L		L	E	L	1	3	2	2	2	2						1	1	1			11
T28	E	E	E		E	1	2							2	2		1	1	1			7

Table 1. Teacher availabilities, preferences and teaching hours per class

Table 2 presents a compact form of all "lessons" that have to be taught to all classes or combinations of classes. Based on the observation that a timetable consists of meetings between teachers and classes for specific durations each table value represents a number of such meetings. For example, value "T2;TH;2" in the second row under column A1 means that the entire class A1 will have 2 lessons of 1 hour duration with teacher T2 teaching subject TH and these meetings should not be scheduled in the same day.

A different situation occurs in column A1 and row 15 having value "T26,T27;TE,PL;1". This means that class A1 will be split into two groups and teacher T26 will teach subject TE to one group while teacher T27 will simultaneously teach possibly a different subject (in this case subject PL) to the other group. This will happen 2 times: one corresponding to row 15 and a second one corresponding to row 16. These lessons can coexist in the same day but certainly in different hours. Note that if those two cells were replaced by a single one with "T26,T27;TE,PE;2" this would have implied that the lesson taught by T26 and the lesson taught by T27 could not have coexisted in the same day. This is the case with row 17, column A2 having value "T20,T17;GL,FL;2". So, teacher T26 teaches "Technology" while teacher T27 teaches "Informatics" in subclasses formed from class A1. On the other hand teacher T20 teaches "German Language" and teacher T17 teaches "French Language" to subclasses originated from class A2. The conclusion is that while "Informatics" and "Technology" lessons can be positioned in the same day for the same group of students, this could not happen for lessons "Germany Language" and "French Language" of class A2. This occurs because in the first case all students will be taught "Informatics" and "Technology" while in the latter case students will be taught either "Germany language" or the "French Language" based on their preference.

Another situation manifests itself with value "T15,T16;AG,AG;3" of merged columns A1 and A2 in row 17. This value means that class A1 and class A2 are joined and then split into 2 groups of students. Teacher T15 takes over one group while teacher T16 takes over the other group. During the week the 3 lessons of teacher T15 should be scheduled to different days and start at the same time with the corresponding lessons of teacher T16.

	A1	A2	A3	A4	B1	B2	C1	C2	C3
1	T2;TH;2	T2;TH;2	T2;TH;2	T2;TH;2	T2;TH;2	T2;TH;2	T1;TH;2	T1;TH;2	T1;TH;2
2	T5;NE;2	T6;NE;2	T3;NE;2	T8;NE;2	T7;NE;2	T6;NE;2	T5;NE;2	T4;NE;2	T6;NE;2
3	T8;AM;2	T7;AM;2	T3;AM;2	T3;AM;2	T7;AM;2	T5;AM;2	T5;AM;2	T4;AM;2	T6;AM;2
4	T8;AR;3	T7;AR;3	T3;AR;3	T7;AR;3	T7;AR;3	T5;AR;3	T5;AR;3	T4;AR;3	T6;AR;3
5	T8;GD;3	T3;GD;3	T3;GD;3	T8;GD;3	T7;GD;2	T6;GD;2	T5;GD;2	T4;GD;2	T6;GD;2
6	T11;MA;4	T11;MA;4	T11;MA;4	T11;MA;4	T10;MA;4	T9;MA;4	T10;MA;4	T10;MA;4	T10;MA;4
7	T12;GE;2	T0;GE;2	T12;GE;2	T0;GE;2	T12;GE;2	T12;GE;2	T26;SE;1	T26;SE;1	T26;SE;1
8	T14;BI;2	T14;BI;2	T14;BI;2	T14;BI;2	T12;XH;1	T12;XH;1	T12;XH;1	T12;XH;1	T12;XH;1
9	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1	T21;KA;1
10	T22;OO;1	T22;OO;1	T22;OO;1	T22;OO;1	T22;OO;2	T22;OO;2	T22;OO;2	T22;OO;2	T22;OO;2
11	T23;GY;3	T23;GY;3	T23;GY;3	T23;GY;3	T23;GY;3	T24;GY;3	T23;GY;2	T23;GY;2	T24;GY;2
12	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1	T25;MO;1
13	T17;IS;2	T17;IS;2	T17;IS;2	T17;IS;2	T16;IS;2	T16;IS;2	T4;IS;3	T4;IS;3	T8;IS;3
14	T17;FL;2	T26,T27;TE,PL;1	T19;2	T26,T27;TE,PL;1	T13;PH;2	T13;PH;2	T13;PH;2	T13;PH;2	T13;PH;2
15	T26,T27;TE,PL;1	T26,T27;TE,PL;1	T26,T27;TE,PL;1	T26,T27;TE,PL;1	T17;FL;2	T26,T28;TE,PL;1	T14;BI;2	T14;BI;2	T14;BI;2
16	T26,T27;TE,PL;1	T20,T17;GL,FL;2	T26,T27;TE,PL;1	T18,T19;HI,IL;2	T26,T28;TE,PL;1	T26,T28; TE,PL;1	T28,T27;TE,PL;1	T28,T27;TE,PL;1	T15,T16;AG,AG;2
17	T15,T16;AG,AG;3		T15,T16;AG,AG;3		T26,T28;TE,PL;1	T18,T19;HI,IL;2	T20,T18;GL,HI;2		T27,T28;TE,PL;1
18					T15,T16;AG,AG;2		T15,T16;AG,AG;2		
19								T17;T18;FL,GL;2	
Total	35	35	35	35	35	35	35	35	35

Table 2. Lessons that have to be taught. Value between semicolons refers to the subject been taught

3. The Algorithm

The algorithm can be considered as a two phase approach. During the first phase IP problems are iteratively solved in order to create a schedule for one day at a time. The first phase ends when all days are solved and after a number of improvements considering each day in isolation have been tried. The second phase systematically selects pairs of days and attempts to move teaching events between days. Again a series of IP problems are generated and gradually better solutions are reached. Firstly, a formal description of the mathematical model employed in solving each problem is presented and subsequently the two phases are analyzed. The second phase can be seen as a very large-scale neighbourhood (VLSN) search. More techniques about VLSN in timetabling problems can be consulted in (Meyers and Orlin, 2007).

3.1 The Mathematical Model

The basic sets used in the problem's model definition are the following:

- T is the set of teachers.
- C is the set of classes.
- D is the set of days in the timetable, usually 5 working days of a single week.
- H is the set of teaching hours in a day.
- E is the set of events.
- R is the set of resources that might be shared by more than one lesson. A resource can be a room of a certain type, a room with special equipment, a video projector etc.

From those sets the following subsets are derived:

- D_t is the set of days that teacher t is available.
- E_t is the set of events that involve teacher t.
- W_t is the set of legal daily schedules for teacher t for all days that he or she is available for work at the school.

Each legal daily schedule of a teacher is a combination of events and idle hours having length $|H|$. The set of legal daily schedules for a teacher is formed by creating all the combinations of events selected from set E_t and idle hours. Events that should not be scheduled in the same day (e.g. events of the same course) cannot exist in the same combination. The selected daily schedules for all teachers should cover the teaching workload of all days so the problem can be categorized as a Set Covering Problem.

Let x_{tdw} be a binary variable that assumes value 1 if teacher t at day d is assigned the daily schedule w and 0 otherwise. Let c_{tdw} be the cost of the daily schedule w for teacher t at day d. This cost comprises from:

- Penalty for idle hours between work in teacher schedules.
- Penalty for deviation from the lower or upper limit of the desired total teaching hours per day.
- Penalty for teaching in an hour that is not in the preferences of the teacher.

The mathematical model of the problem is presented below:

$$Minimize \sum_{t=1}^{|T|} \sum_{d=1}^{|D|} \sum_{w=1}^{|Wt|} c_{tdw} x_{tdw} \qquad [1]$$

Subject to:

$$\sum_{t=1}^{|T|} \sum_{d=1}^{|D|} x_{tdw} = 1 \quad t \in T, d \in D, w \in W_t \qquad [2]$$

$$\sum_{t=1}^{|T|} \sum_{d=1}^{|D|} \sum_{w=1}^{|Wt|} a_{tdwe} x_{tdw} = 1 \quad e \in E \qquad [3]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|Wt|} a_{tdwhc} x_{tdw} = 1 \quad d \in D, h \in H, c \in C \qquad [4]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|Wt|} a_{tdwhe} x_{tdw} = k_e y \quad d \in D, h \in H, e \in E with T > 1 \qquad [5]$$

$$\sum_{t=1}^{|T|} \sum_{w=1}^{|Wt|} a_{tdwhr} x_{tdw} = MaxQ_r \quad d \in D, h \in H, r \in R \qquad [6]$$

Eq. 1 is the objective function and represents the total cost of a solution that should be minimized. It is the sum of the cost over all selected daily schedules for all teachers and for all days of the teaching week.

Constraint shown in Eq. 2 states that one daily schedule should be selected for each teacher.

Eq. 3 states that all events must be assigned meaning that each event must exist at exactly one selected daily schedule of a teacher. In this equation parameter a_{tdwe} assumes value 1 if daily schedule w of teacher t in day d includes event e or 0 otherwise.

Eq. 4 guarantees that the schedule of each class will have no idle times. So, for each day and hour 1 event must exist that involves each class. It should be noted that a_{tdwhc} is a parameter that assumes value 1 if the daily schedule w of teacher t in day d and hour h includes an event that involves class c and 0 otherwise.

Eq. 5 handles events with more than 1 teacher and instructs them to be included in the same day and hour in the daily schedules of the affected teachers. Binary variable y assumes value 1 when k_e teachers should teach concurrently and value 0 when no concurrent teaching occurs. Parameter a_{tdwhe} assumes value 1 if the daily schedule of teacher t in day d and hour h includes event e and 0 otherwise.

Eq. 6 appears in case room or other potentially shared resources exist. $MaxQr$ is the available quantity for resource R while parameter a_{tdwe} is the number of the required quantities for resource r if daily schedule w of teacher t in day d and hour h includes an event that requires resource r and 0 otherwise.

The size of the problem is too big to be solved including all possible legal daily work schedules for all teachers and all days. So, an approximation method is employed where a preliminary solution is initially created and subsequently gets improved. Problem sizes in each step of the method are relatively small and can be solved in reasonable time.

3.2 First Phase: Solve by Day

An initial solution is generated by solving the mathematical model of the problem considering a single day at a time. Eq. 3 of the mathematical model is no longer needed because each single day can have a subset only of the events from E scheduled in it. Each column of the mathematical model represents a work schedule of a teacher for the day under consideration.

The order of days that will be considered is determined by estimating how difficult each day timetable construction might be. A day is considered more difficult than another when less teachers with a lot of teaching hours are available.

After the day is selected a list of yet unscheduled lessons that are legitimate to be taught is constructed for each teacher. Then, all possible combinations of each list's lessons that could have been scheduled in the available hours are generated. So, for each teacher a set of legitimate work patterns is assembled with the presupposition that the size of the set is manageable. Otherwise, subsets of combinations unlikely to be included in the final solution are prematurely rejected. For each teacher's set of patterns only one will be included in the schedule of the day under question.

In order to solve the first day, daily work schedules of each teacher for this day are generated. For each teacher t all events E_t are considered. After each day gets solved daily schedules for all teachers are re-computed considering events that still remain unscheduled. In solving each day every legitimate daily work schedule for a teacher t should contain all events characterized as mandatory. If not then infeasible problems will result in solving next days.

Nevertheless, for teachers with many events the number of possible daily work schedules might be overwhelming. So, an artificial upper bound to the number of generated daily work schedules that contain a specific event at a specific hour is imposed. When this bound is reached, by counting work patterns, this event is removed from the list of possible events for this hour. So, gradually the number of combinations diminishes while daily work schedules of each teacher having each event in every possible hour do exist.

When the daily work schedules of all teachers are finished generated the linear relaxation of the mathematical model can be solved. The rather small problem size results in fast solve times for each relaxed problem. When the solution of the relaxed problem shows that a margin for better solutions does exist, the integrality constraints are enforced and the problem gets solved again this time as an IP problem. A time limit of 1 minute is allowed which might be less than the time needed for solving the problem to optimality. When results are obtained from solving the IP problem, even if they are suboptimal, they are used to define the daily work schedule for each teacher.

The process continues by generating a new problem for the same day that is examined to give better cost values. The solution derived from the relaxed problem of the previous step is exploited in two ways. Firstly, a column is selected that corresponds to the daily work schedule of a single teacher with the smallest cost contribution that has also the greatest value in the solution of the relaxed LP. The set of events that are included in the selected daily schedule are decided to "freeze" meaning that these events should be scheduled thereafter in the day and hour dictated by the relaxed LP. Such decisions ease subsequent daily work schedule generations by lowering the total number of combinations that have to be considered. Secondly, dual values derived from solving the LP problem are exploited in calculating the reduced cost value of each column if included in the solution. Reduced cost of a variable is the amount by which its coefficient in the objective function has to be decreased so as to be included in the solution.

Legitimate teacher day schedules that have negative reduced cost are stored so as to be included in the next problem that will be solved. This result in gradual decrease of the cost value found by the LP solver or in the worst case achieving the same cost value as before. By selecting columns from the LP problem that lowers the value of its objective function we hope to discover better values for the IP problem too. This process repeatedly occurs during our approach. A column, that represents a teacher day schedule, is stored if its cost is better than the cost of the best solution found so far for this day. Naturally, a column with cost equal or greater cannot be included in a solution better than the solution that we already have.

The cost of the best LP solution found so far is inserted as upper bound for the objective function in the IP problem. The new problem contains columns forming the basis of the LP model, columns of the best integer solution and the new columns that have been generated. The procedure repeats until no more columns can be found that can be frozen or when no more teacher daily schedule programs can be generated.

3.3 Second Phase: Solve by day pairs

The previous phase results in an initial solution where all days have complete schedules for all teachers and classes. The second phase tries to find a better solution by selecting two days and trying to move teacher events between them. So, for every pair of days under investigation new daily teacher schedules are examined that improve the cost of the two day solution and therefore the cost of the full planning period solution. Teacher schedules in the days not belonging to the selected pair remain unaffected.

The same mathematical model described previously is used but in this case it includes two instead of one day. The set of events *Et* that involve teacher t is formed from the original set *Et* by removing events that are scheduled in the unaffected days of the planning period. The solution procedure is the same as that of the previous phase and it continues until no improvement can be found by any two days combination.

4. Implementation

The implementation of our approach was undertaken in Java using the open source mathematical solver GLPK (http://www.gnu.org/software/glpk/). Three key modules of the software can be identified that facilitated the process undertaken in the phases described in the previous paragraphs. These are: CostEvaluator, MandatoryFinder and CombinationsGenerator. A brief description of each module's role follows.

CostEvaluator is the component that calculates the cost of each solution which might be partial or complete. Furthermore, it computes the cost of every teacher daily schedule that is subsequently used as a coefficient in the mathematical problem.

On the other hand, given a partial solution MandatoryFinder calculates the events that have to be scheduled in the day under consideration so as to avoid infeasibilities that would occur latter in case that these events were left to be scheduled later. For example, suppose that a lesson have to be taught 4 times in a week and after solving the first day all events of this lesson are still unscheduled. Then, for all subsequent days an event of this lesson has to be taught. MandatoryFinder also handles more complex situations like when an event involves more than one teacher.

CombinationsGenerator is a heavily used module that is responsible for computing the combinations of events that should be scheduled in the hours available in each day. In order to reduce the number of combinations generated that occurs when the number of possible events of a teacher is large, heuristics are used so as to select a representative subset of all available combinations.

5. Experimental Results

The datasets used in our experiments originate from High schools in Greece. We tested our approach in several problem instances with different characteristics and in all cases a feasible solution of High quality was able to be achieved. Execution times even for bigger schools (~50 teachers, ~15 classes) were less than 20 minutes in our test computer which was an Intel i3 380M (2.53GHz) with 3GBytes RAM running Windows 7 64 bit.

Among datasets used two of them, presented in Table 3, belong to the benchmarking project for High School Timetabling. These datasets alongside with solutions achieved by our and other approaches are publicly available in (http://www.utwente.nl/ctit/hstt/). The benchmarking project for High School Timetabling is a joint effort from researchers across several countries so as to create a common XML standard for exchanging datasets. Description of the XML format can be consulted in (Post et al, 2012) and (Post et al, 2011). In both datasets our approach found the optimal solution in less than 10 minutes. Dataset GreeceThirdHighSchoolPatras2010 and GreeceThirdHighSchoolPreveza2008 don't have room constraints. Nevertheless, other custom datasets including such type of constraints were solved.

Dataset	Periods	Teachers	Classes	Events	Duration
GreeceThirdHighSchoolPatras2010	35	29	9	178	340
GreeceThirdHighSchoolPreveza2008	35	29	9	164	340

Table 3. Datasets included in the benchmarking project for High School Timetabling

The solution schedule that has been produced and satisfied all constraints for dataset GreeceThirdHighSchoolPatras2010 is presented in Table 4. Each asterisk represents a scheduled teaching for the associated teacher and period. It can be easily observed that the generated daily schedule of each teacher consists of consecutive busy periods.

```
TEACHER |  Mon  |  Tue  |  Wed  |  Thu  |  Fri  |
T0      |*------|*------|-------|*------|*------|
T1      |------*|------*|-*-----|-*-----|-----**|
T2      |**-----|----***|***----|****---|-------|
T3      |---****|****---|-***---|-----**|**-----|
T4      |--***--|***----|-----**|----***|---****|
T5      |****---|-***---|***----|**-----|---****|
T6      |-----**|----***|---****|---****|--**---|
T7      |****---|****---|---**--|----***|---****|
T8      |--****-|***----|***----|****---|**-----|
T9      |-------|-*-----|--*----|--*----|------*|
T10     |----***|---****|---****|-***---|-----**|
T11     |-----**|----***|---****|---****|--***--|
T12     |**-----|----***|-**----|----***|***----|
T13     |--*----|--**---|*------|--***--|***----|
T14     |---****|****---|----***|-------|----***|
T15     |----**-|----***|---***-|***----|---*---|
T16     |----***|---****|---****|***----|--**---|
T17     |***----|***----|***----|---****|***----|
T18     |-*-----|----**-|**-----|-------|***----|
T19     |-----*-|----**-|------*|-------|-**----|
T20     |-------|-------|-**----|-------|**-----|
T21     |-***---|-------|-------|**-----|-****--|
T22     |----***|-------|---****|---****|***----|
T23     |*****--|---****|---****|***----|----***|
T24     |*------|*------|*------|---*---|-----*-|
T25     |------*|--**---|*------|----***|**-----|
T26     |****---|**-----|***----|-**----|---****|
T27     |---*---|*------|***----|***----|----***|
T28     |-**----|-*-----|--*----|*------|---**--|
```

Table 4. Optimal schedule for GreeceThirdHighSchoolPatras2010

6. Conclusions

Quality schedules for High Schools are vital for the success of the education effort. Obtaining manually such a schedule is unlikely to occur so computer generated solutions seem to be the only logical option. Nevertheless, practical issues often prohibit schools of operating based on an optimal schedule for students and teachers. In this contribution we presented an approach for the Greek case of the High School problem and we showed its ability to generate very good and in some cases optimal results. The approach is based on a two phase process that incrementally solves parts of the problem, using mathematical programming. Better schedules are unmasked while portions of the schedule remain fixed. The process gradually converges to better schedules.

We acknowledge the fact that adaptation of new automated timetabling solutions by High Schools is a target hard to achieve but we believe that the quality of the schedules that we produce should entice schools in Greece to test the proposed approach. It is in our plans to offer a web service that schools in Greece will be able to use so as to enter the schedule problem as it occurs in their specific case and our application will propose a High quality schedule based on this data. We also plan to offer a re-schedule service that a High School could use so as to make changes throughout the year to an existing schedule while keeping most of it unchanged.

References

Beligiannis, Grigorios N., Moschopoulos N. Charalampos, Kaperonis P. Georgios, and Likothanassis D. Spiridon. "Applying Evolutionary Computation to the School Timetabling Problem: The Greek Case." *Computers & Operations Research* 35, no. 4 (2008): 1265–1280.

Birbas, T., S. Daskalaki, and E. Housos. "School Timetabling for Quality Student and Teacher Schedules." *Journal of Scheduling* 12, no. 2 (October 28, 2008): 177–197.

Bondy J.A. and Murty U.S.R., Graph theory with applications, North-Holland (1976).

Burke, Edmund K., Barry McCollum, Amnon Meisels, Sanja Petrovic, and Rong Qu. "A Graph-based Hyper-heuristic for Educational Timetabling Problems." *European Journal of Operational Research* 176, no. 1 (2007): 177–192.

Csima J. Investigations on a Time-Table Problem, PhD Thesis, Institute of Computer Science, University of Toronto (1965).

de Werra, D. On some combinatorial problems arising in scheduling. INFOR, 8, 165-175.

Dempster, M. A. H. (1971). Two algorithms for the time-table problem, in Combinatorial Mathematics and its Applications (ed. D. J. A. Welsh), Academic Press, New York, pp. 63-85.

Liu, Yongkai, Defu Zhang, and Stephen C.H. Leung. "A Simulated Annealing Algorithm with a New Neighborhood Structure for the Timetabling Problem." In *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 381–386. GEC '09. New York, NY, USA: ACM, 2009. http://doi.acm.org/10.1145/1543834.1543885.

Meyers, Carol, and James Orlin. "Very Large-Scale Neighborhood Search Techniques in Timetabling Problems." In Practice and Theory of Automated Timetabling VI, edited by Edmund Burke and Hana Rudová, 3867:24–39. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007.

Post, Gerhard, Jeffrey H. Kingston, Samad Ahmadi, Sophia Daskalaki, Christos Gogos, Jari Kyngas, Cimmo Nurmi, et al. "XHSTT: An XML Archive for High School Timetabling Problems in Different Countries." Annals of Operations Research (November 10, 2011).

Post, Gerhard, Samad Ahmadi, Sophia Daskalaki, Jeffrey Kingston, Jari Kyngas, Cimmo Nurmi, and David Ranson. "An XML Format for Benchmarks in High School Timetabling." Annals of Operations Research 194, no. 1 (2012): 385–397.

Schaerf, A. “A Survey of Automated Timetabling.” Artif. Intell. Rev. 13, no. 2 (April 1999): 87–127.

Valouxis, Christos, and Efthymios Housos. “Constraint Programming Approach for School Timetabling.” *Computers & Operations Research* 30, no. 10 (2003): 1555–1572.

Valouxis, Christos, Christos Gogos, George Goulas, Panayiotis Alefragis, and Efthymios Housos. “A Systematic Two Phase Approach for the Nurse Rostering Problem.” *European Journal of Operational Research* 219, no. 2 (2012): 425–433.