Course Scheduling

Tolga Genç *
Department of Industrial Engineering
Bilkent University
TR-06533 ANKARA
tolgag@bilkent.edu.tr

May 4, 1999

Abstract

Course scheduling problems consist of scheduling a certain number of resources, such as classes, teachers, courses and the classrooms to a number of time-slots on a daily basis. The feasible combinations for these resources are the ones that avoid the conflicts between the teachers, classes, rooms and other specified requirements. Models formulated for scheduling problems are also case-sensitive and generally quite different from each other. Therefore, to establish the common features of the course scheduling problems is crucial. This paper presents a general view of course scheduling problems, formulate a high-school course scheduling problem which includes the common elements and the features of most of the course scheduling problems.

1 Introduction

The course scheduling problem consists of scheduling a certain number of resources, such as classes, teachers, courses and the classrooms to a number of time-periods on a daily basis. The feasible combinations for these resources are the ones that avoid the conflicts between the teachers, classes, rooms, and other specified requirements. The variety of the course scheduling problems is very large due to the different characteristics of regulations in the educational systems. Another reason may be the frequent changes occurring in the educational methods. Therefore, it is really hard to state a general model for course scheduling problems applicable for most of the cases but the elements used in these problems seem to be similar [2].

*In partial fulfillment of the requirements for the course IE 672 Spring 1999

The basic elements for the course scheduling problems are : Day, Time - Slot , Class, Teacher and Course. Problems are constructed by using 0-1 variables and defining appropriate constraints to satisfy the feasibility requirements and the rules of the educational system and staffing regulations.

School course schedules have been constructed by hand for many years. But these schedules resulted in dissatisfaction of faculty with their teaching assignments and poor utilization of the classrooms. Schedules have to improve classroom utilization and satisfy faculty preferences. To reach this goal, computerized scheduling methods which certainly save time and ensure the schedule feasibility have to be utilized.

There has been an extensive study of various course scheduling problems and different approaches have been suggested for formulation and solution of the problems [3]. Mathematical programming is the mostly used modeling technique for these problems. Also, it plays a important role in describing the problem clearly. Therefore, it is necessary to model the problem by using mathematical programming before deciding which modeling technique is appropriate for the current case. With the entrance of the mathematical programming technique in course scheduling activities, the automation drew attention but in spite of the complexity of the problems also caused some unexpected solutions where important constraints are violated. To reduce the complexity, unsolvable huge problems were divided and solved consecutively. The results then seemed satisfactory but now require manual implementation. In this paper, the basic structure of a course scheduling problem will be described, a mathematical programming model will be given and the solution techniques will be summarized [11].

2 Problem Description

Although course scheduling problems vary according to the type of the educational institutions, their particular rules and characteristics, we can group them into two major categories : High school course scheduling problems and university course scheduling problems. University course scheduling problems are much more complicated than the high school course scheduling problems. The main difference between them occurs in the assignment of classes to classrooms and instructors to classes. In universities, instructors are assigned to the courses and classes don't have to be assigned to the predetermined classrooms. But in high schools, each class takes the courses in one classroom and teacher to class assignments are also available. It can be considered that high school course scheduling problem possesses the main frame of the complex university course scheduling problem and it seems to be a simplified version.

The weekly schedule of a high school provides for each class a sequence of courses to be taught every day of the week. The curriculum of a school is fixed for a given class and scheduling problem involves scheduling of the different subjects, whose required number of meetings per week is given for each course, according to the feasibility and system rules.

Objective function of the model depends on the preferences and the needs of the educational institutions. Scheduler must attach weights to the objective function elements. The objective function elements involve with the instructor preferences related with the assignment of their teaching hours to the specified time-slots, balancing the distribution of the lectures over a week period, campus related assignments, limiting staff workload.

The general high-school model has the following elements, followed by their respective set of values:
Day : day on which a course takes place;
Day = {Monday, Tuesday,Wednesday, Thursday, Friday}
Time-slot : time-period of a day in which course can be assigned;
Time-slot = {8-9 am, 9-10 am,... }
Class : group of students who attend same courses which are defined in class curriculum in the same class-room;
Class = {1A,1B,...,2A,2B,...,3A,3B,...}
Teacher : person teaching a certain course;
Teacher = {ID1, ID2,..}
Course : subject taught in a class;
Course = {Course1, Course2,...}

It is assumed that value sets of the structure elements Day, Time-Slot, Class, Teacher, Course are finite and strictly ordered. This structure is applicable to the most of the course scheduling problems. The size and the complexity of the model depend on the size of the value set of each structural element.

The formulation of the mathematical programming model is based on the variable $X_{a,b,c,d,e}$ with the five indices a, b, c, d, e. Each index of variable x corresponds to a different structural element explained above. Thus index a takes values from the set Day, b from the set Time-Slot,c from the set Class, d from the set Teacher, and e from the set Course. Since the indices of a variable must be integers, then to get the one-to-one mapping between the value set of each structural element and integer values, the variable $X_{a,b,c,d,e}$ is directly transformed to variable $X_{i,j,k,l,m}$ where $i \in I = \{1, 2, ..., i1\}, j \in J = \{1, 2, ..., j1\}, k \in K = \{1, 2, ..., k1\}, l \in L = \{1, 2, ..., l1\}$ and $m \in M = \{1, 2, ..., m1\}$ with i1,j1,k1,l1,m1 corresponding to the cardinalities of the sets Day, Period, Class, Teacher and Course. Total number of the variables is the multiplication of the cardinalities of the sets.

2.1 Constraints

Mainly, there are two sets of constraints which are grouped according to their importance. The first set includes the hard constraints which ensure feasibility of the model. The other set of constraints are called as soft constraints and a solution violating such constraints is not considered as infeasible but such violations are undesirable.

Hard Constraints:

1. The subjects of the courses listed in the curriculum of each class must be thought.
2. Each teacher is pre-assigned to certain courses and certain classes.
3. A teacher can not be assigned more than one subject of course at the same period.
4. Each class is taught in the same class-room. So no class-room assignments have to be made.
5. The total number of teaching hours for each class is fixed.
6. Each teacher must be assigned at least one-time period per day for all days.

Soft Constraints

1. The time-slots assigned to a course should be evenly distributed over week days.
2. Certain courses are preferred to be taught at the beginning of the day, some are preferred to be taught at the end of the day.
3. The preferences of the teachers related to their teaching times must be satisfied.

2.2 Model Formulation

The variable $X_{i,j,k,l,m}$ is defined previously, it is a binary variable, if course m of class k is taught by teacher l in day i, at time-slot j, then variable x is 1; otherwise it is 0.

Constraint sets in the model are as follows [4]:

$$\sum_{k \in K_l} \sum_{m \in M_{kl}} X_{i,j,k,l,m} \leq 1 \quad i \in I, j \in J \tag{1}$$

Where K_l= $\{k \in K$: k = class assigned to teacher l $\}$, l$\in L$
M_{kl} = $\{$m$\in M$: m = subject taught in class k by teacher l $\}$

These constraints prevent the cases where a teacher is assigned to more than one class at the same time-slot.

$$\sum_{l \in L_k} \sum_{m \in M_{kl}} X_{i,j,k,l,m} = 1 \quad i \in I, j \in J \tag{2}$$

where $L_k = \{l \in L : \mathrm{l} = \text{a teacher assigned to teach in class k}\}$, $k \in K$.

These constraints ensure that for each day and the time-slot, all classes will be assigned a teacher and a subject of a course.

$$\sum_{l \in K_l} \sum_{j \in J} \sum_{m \in M_{kl}} \sum_{i \in I} X_{i,j,k,l,m} = a_k \quad k \in K \tag{3}$$

where a_k is the total number of time-slots per week required for class k. These constraints ensure that all classes will satisfy their weekly requirements in terms of total number of time-slots for presence in class.

$$\sum_{k \in K_l} \sum_{m \in M_{kl}} \sum_{j \in J} X_{i,j,k,l,m} \geq 1 \quad i \in I, l \in L \tag{4}$$

These constraints ensure that every teacher must be assigned at least one hour for all days.

The constraints that are explained above are the hard constraints. Soft constraints depend on some specifications and preferences, therefore is is difficult to express them in a standard form. Since there is no requirement to satisfy soft constraints, it is needless to keep them in the model and reduce the feasible region. So, it is better to move them to the objective function assigning weights to them as cost terms. For example, a soft constraint set which is related with the instructor preferences concerned with the teaching assignments can be moved to the objective function by indicating relative weights to the preferred course to time-slot assignments for each instructor. Such an reduction in the constraint set saves the computation time and prevents unexpected infeasible solutions.

3 University Course Scheduling Case

University course scheduling problems are more complicated than the high school course scheduling problems and the variety is increased as the structure of the problem is becoming more complex. The definitions made for the elements in high school course scheduling problem are generally valid but in some cases, they must be modified according to the characteristics of the system. In universities, courses have several sections and each course section is assigned to an instructor. Classes or groups are defined as the group of people following a fixed curriculum together. In spite of this definition, the courses taken by each student can be different within a class. Because there is no restriction for students in taking the courses which are not listed in the curriculum (such as elective courses). Class to classroom assignments are not performed on class/group basis, that is at each time-slot, every course section is assignable to the classrooms whose size fits the course section limit. Limits show the maximum number of the students that can register to the specified course section and these are pre-determined. Therefore, at each time-slot, the number of classrooms that will be used is determined by the model and then course section to classroom assignments are accomplished manually.

Subjects of course sections can be given within pre-determined blocks of hours to prevent unexpected distribution over the scheduling period. For example, in Bilkent University, such blocks are called as sessions whose numbers and sizes change according to the total requirement hours of that course section within a week. Hard constraints that are indicated previously are also valid for university course scheduling problems and it is obvious that clash(instructor, curriculum) and course section requirement constraints must exist in every course scheduling problem. We have additionally classroom availability constraint set which ensures the assignment of course sections to each time-slot in a such way that the number of the assignments must not exceed the number of the available classrooms for that course section.

4 Solution Approaches

There are several solution approaches for course scheduling problems. The basic approaches are:

1. Mathematical Programming
2. Graph Coloring
3. Tabu Search Algorithm
4. Other techniques

4.1 Mathematical Programming

It is the most popular technique used in course scheduling activities. A number of articles are based on mathematical programming methods such as linear and integer programming. In Akkoyunlu[1] and Csima and Gotlieb[5], a linear programming problem has been used for the computation of an optimal solution for the simplified version of the problem. This is simply an assignment problem between a set of courses and the total number of time-slots of weekly university course scheduling problem has been solved. These are based on prevention of conflicts however no further rules regarding educational systems are applied. In Lawrie[10], a linear programming model with combinatorial extensions has been used in order to solve the school course scheduling problem. This paper does not attempt the scheduling of instructors, classes and rooms but instead creates schedules for elective courses and various groupings of students. Birbas[4] attempted to solve a Greek high school timetabling problem with mathematical programming model. This paper presented an efficient and a compact model with significant number of rules and regulations. In Wood[12], student centred school timetabling is formulated where students freely choose their courses of study form a complete list.

4.2 Graph Coloring

Graph coloring is the formulation of the problem that is described as mathematical programming on a graph. Since graph coloring is a manual technique, it is not preferred to be used in course scheduling problems, Beside this, it is impossible to express complex models on a graph. Therefore, it only works in cases where the course scheduling problem can be reduced to simple assignment problem. Werra[11] has presented graph coloring technique to solve simple class to instructor assignment problem which is suitable for high school case. The nodes in the graph are the classes and instructors : node c_i and t_j are linked by r_{ij} parallel edges. If each period corresponds to a color, the problem consists in finding an assignment of one among p colors to each edge of the graph in such a way that no two adjacent edges have the same color ; so X_{ijk} will be 1 if some edge (c_i,t_j) gets color k. After a feasible solution is obtained, by using heuristic approach or network flow techniques other edge coloring solutions are obtained.

4.3 Tabu Search

Tabu search is an iterative heuristic approach used in large sized course scheduling problems. The requirements or constraints are partitioned into two as essential and relaxed requirements. These are similar to the classification as hard and soft constraints. A schedule is feasible if it satisfies the essential requirements and cost function is defined for the violation of the relaxed constraints for each feasible schedule in the feasible schedule set. The aim is to find a feasible solution with minimum cost function.[8]

4.4 Other Techniques

In addition to the techniques explained above, due to the case-dependent structure of course scheduling problems, different heuristic methods are developed to solve specific problems. They do not share common properties in solving various combinations of problems. In most of the cases, the course scheduling problems are divided into two or more assignment problems such as class to instructor, course to time-slot or class to classroom assignment problems, each of these assignment problems solved heuristically by graph coloring or tabu search algorithms or if it is possible mathematical programming is utilized. Then manual implementation takes place to create a feasible solution by combining the solutions of each sub-problem. This is a very time consuming process and only applicable for small sized course scheduling problems. In literature, many papers have been published including heuristic methods. Eglese and Rand[7] have proposed a heuristic algorithm to create the timetable of a typical seminar series. Their algorithm consists of two parts: an initial feasible solution finder and improvement procedure that uses simulated annealing. Kiaer and Yellen[9] have also examined the university course scheduling problem. A weighted graph is introduced to model the problem, where the edge weights quantify the severity of the conflict among the constraints and a heuristic algorithm is used to find feasible solutions to this problem.

5 Conclusion

In this paper, a typical course scheduling problem is described with its substantial elements for both high school and university cases, a mathematical model for a high school scheduling problem is presented and finally solution approaches developed in the literature are summarized. The mathematical model presented in this paper is generic and flexible while it captures all the specific characteristics of the problem. The size of the model can be enlarged upon the need of required detail and complexity. Although solution techniques are dispersed in a large area, the mathematical programming technique keeps its popularity in automated scheduling. Other solution techniques such as graph coloring and tabu search algorithm give good results when the size of the problem is small.

References

[1] Akkoyunlu EA, *A Linear Algorithm for Computing the Optimum University Timetable*, Computer J 16 (1973), 347-350.

[2] Alvarez R., Martin G., Tarmarit J.M, *Constructing Good Solutions for the Spanish School Timetabling*,Journal of Operational Research 47 (1996), 1203-1215.

[3] Badri Masood et al, *A Multi Objective Course Scheduling Model:Combining Faculty References for Courses and Times*, Computers Ops Res 25 (1998), 303-316.

[4] Birbas T.,Daskalaski S. and Housos E., *Timetabling for Greek High Schools*, Journal of the Operational Research Society 48 (1997), 1191-1200.

[5] Csima J. and Gotlieb CC, *Tests on a Computer Method for Construction of School Timetables*, Comm ACM 7 (1961), 160-163.

[6] Drexl A., Salewski F., *Distribution Requirements and Compactness in school timetabling*, European Journal of Operational Research 102 (1997), 193-214.

[7] Eglese R.W., Rand G.K., *Conference Seminar Timetabling*, Journal of Operational Research Society 38 (1987), 591-598.

[8] Hertz, A., *Tabu Search for Large Scale Timetabling Problems*, European Journal of Operational Research 54 (1991), 39-47.

[9] Kiaer, L., Yellen, J., *Weighted Graphs and University Course Timetabling*, Computers Operations Research 19 (1992), 59-67.

[10] Lawrie N.L., *An Integer Programming Model of a School Timetabling Problem*, Computer Journal 12 (1969), 307-316.

[11] Werra, D. , *An Introduction to Timetabling*, European Journal of Operation Research 19 (1985), 151-162.

[12] Werra D. , Chahal, *An Interactive System for Constructing Timetables on a PC*, European Journal Operation Research 40 (1989), 32-33.

[13] Wood, DC., *A System for Computing University Timetables*, Computer Journal 11 (1968), 41-44.