Author's Accepted Manuscript

www.elsevier.com/locate/caor

Mathematical Models and Algorithms for a High school timetabling problem

Salem M. Al-Yakoob, Hanif D Sherali

PII: S0305-0548(15)00042-8
DOI: http://dx.doi.org/10.1016/j.cor.2015.02.011
Reference: CAOR3732

To appear in: *Computers & Operations Research*

Mathematical Models and Algorithms for a High School Timetabling Problem

Salem M. Al-Yakoob [a], Hanif D. Sherali [b]

[a] Department of Mathematics, College of Science, Kuwait University, P.O. Box: 5969, Safat 13060, State of Kuwait, Tel: (965) 90904500, Fax: (965) 4817201, E-mail: salem@al-yakoob.com

[b] Grado Department of Industrial and Systems Engineering (0118), Virginia Tech, Blacksburg, VA 24061, USA, E-mail: hanifs@vt.edu

Abstract

This paper investigates a high school timetabling problem in a case study related to Kuwait's public educational system, which is concerned with assigning teachers to classes and time-slots. Because a direct solution to an initially formulated comprehensive mixed-integer programming model for generating weekly teacher schedules was found to be untenable for practical-sized realistic test instances, we propose in this paper two decomposition approaches to the underlying problem. A two-stage modeling solution approach is presented first, where the initial stage determines weekly time-slots for the classes, based on which, the second stage then assigns teachers to classes. Instead of generating weekly schedules within the model itself, we propose another mixed-integer programming formulation that selects valid combinations of weekly schedules from the set of all feasible schedules, and we design a column generation solution framework to exploit its inherent special structure. Computational results are presented for the proposed solution approaches using several real as well as simulated realistic test problems pertaining to high schools in Kuwait.

Keywords: Timetabling, scheduling, mixed-integer programming, two-stage approach, column generation, linear programming-based heuristic.

1. Introduction

This section presents an overview of the studied problem in Subsection 1.1, and then reviews the related research available in the open literature in Subsection 1.2. Finally, Subsection 1.3 discusses the contribution and organization of this paper.

1.1. Problem specifics

In this research effort, we consider a challenging high school timetabling problem that is ubiquitous in many countries worldwide. The specific problem investigated in this paper is concerned with generating class-teacher timetables for high schools in Kuwait's public educational system. In this context, there are three grade levels: tenth, eleventh, and twelfth. The class requirements for the tenth grade (*i.e.*, subjects and hours for each subject) are fixed for all students and are known in advance. For the eleventh grade, the students have the choice of specializing in Science or Liberal Arts, where again, the class requirements are fixed and known. A student who has completed the eleventh grade in Science proceeds to the twelfth grade with the same specialization, and likewise for a student who has completed the eleventh grade in Liberal Arts. We aim to find weekly class-teacher schedules without violating any of the problem's (hard) constraints that include: (a) avoiding the assignment of more than one teacher to a class during any given time-slot; (b) avoiding the assignment of a teacher to more than one class during any given time-slot, and (c) not exceeding the daily and weekly teaching loads of teachers.

1.2. Related Literature

Academic institutions worldwide face complex timetabling problems related to scheduling courses, exams, faculty members, teaching assistants, and the like. The generation of effective and robust timetabling tools for concerned decision makers requires efficient quantitative and algorithmic approaches due to the highly combinatorial nature of such problems, especially for relatively large institutions. Timetabling problems that incorporate multiple critical features are typically challenging and theoretically NP-hard [19, 23, 29, 31, 33]. Several surveys related to academic timetabling problems have been presented in the literature; see for example, [20, 23, 48, 57, 61, 68]. Also, McCollum [50] and McCollum *et al*. [51] have provided insightful discussions that focus on bridging the gap between research and practice in this area. In this context, increasing attention has been dedicated to timetabling problems in general, as evidenced from: a) the series of biannual international conferences on the Practice and Theory of Automated Timetabling (PATAT); b) the establishment of prominent research groups, such as WATT and the EURO Working Group on Automated Timetabling, which were established to "discuss, promote, and perform research related to automated timetabling issues and methods"; and c) the vast number of graduate theses and dissertations related to timetabling problems, e.g., [38, 41, 44, 47, 49, 52, 63, 79].

High school timetabling problems vary depending on the country and type of educational system [61]. Country-specific literature includes that related to Australian schools [52], Dutch schools [28, 38, 41, 69, 79], Finnish schools [49], German schools [39, 40], Greek schools [14, 15, 17, 56], Hong Kong schools [43], Italian schools [7], South African schools [62], and Spanish schools [6, 8]. This body of research has produced valuable theoretical, conceptual, and practical contributions. Moreover, to facilitate comparative studies, a *School Benchmarking Project* was launched in 2007 (see Post *et al*. [58, 59, 60]), where a general file format was introduced based on typical common features, and where researchers from all over the world have contributed instances pertaining to their local schools. The website dedicated for this project (http://opt-kd.cse.dmu.ac.uk/www/information.php) also contains insightful and important information about high school timetabling such as terminology, references, history, links, algorithms, and most importantly, a data repository. This project is still active and the continued contributions of researchers in this domain is valuable.

Several algorithmic approaches have been used in the literature to tackle high school and related timetabling problems, including ant colony algorithms [32, 70], constraint programming [36, 46, 77], evolutionary algorithms [14], genetic algorithms [15, 34, 62, 63, 76], graph theory-based procedures [16], mixed-integer programming methods [17, 18, 25, 26, 45, 56, 65, 75], local search schemes [10, 67], metaheuristics [24, 27], neural network algorithms [21, 22, 35], programmed search techniques [37], simulated annealing [2, 3, 4, 74, 81], and tabu search heuristics [1, 5, 7, 8, 11, 39, 64, 66]. A number of high school related survey papers are also available in the literature-for example, Junginger [40] has surveyed high schools in Germany that use computer software for their timetabling, and Kwok *et al*. [43] have surveyed approaches that pertain to the timetabling process for high schools in Hong Kong. Also, Schaerf [68] has presented a general survey on automated timetabling, along with a special section devoted to high school timetabling. Given these extensive discussions, we focus in the remainder of this section on some relatively recent work related to high school timetabling.

Wood and Whitaker [80] formulated a nonlinear goal programing model for a secondary school timetabling problem, where students freely choose their courses of study from a list of subjects. Similar to the problem dealt with in the present paper, Birbas *et al*. [17] considered a timetabling problem faced by high schools in Greece. The authors presented an integer programming model for scheduling a large number of classes, teachers, courses, and classrooms with respect to a number of time-slots. Computational results for a typical Greek high school were presented to demonstrate the effectiveness of the developed model in satisfying both hard and soft problem constraints. Valousix and Housos [77] also addressed this same type of problem and solved it using a constraint programming (CP) approach. Local search techniques were used to assist the CP search process by effectively reducing the solution search space, and specialized algorithms were developed to compute lower bounds for providing performance guarantees. Computational results for a number of specific test problem instances were presented. A column generation approach for the problem investigated in [17] was proposed by Papoutsis *et al*. [56] based on a 0-1 integer programming model in which each column represents a valid schedule of a given teacher. The authors solved the linear programming relaxation of the resulting model by sequentially generating promising new columns via a network-based heuristic, and accordingly designed an overall heuristic procedure to solve the original discrete problem. Birbas *et al*. [18] proposed an integer programming approach for a high school timetabling problem related to the Hellenic secondary educational system in Greece. A shift assignment problem that optimally assigns shifts to teachers was solved first, and then tasks within shifts were assigned to the corresponding teachers. Using this combined approach, the authors were able to generate conflict free, complete, fully compact, and well balanced timetables for students. Daskalaki *et al*. [25] and Daskalaki and Birbas [26] developed mixed-integer programming approaches for a timetabling problem related to Greek universities. In the latter paper, a two-stage relaxation procedure was designed, where the computationally heaviest constraints were initially relaxed in the first stage, and were resurrected in the second stage for generating timetables on a day-by-day basis in order to construct a good quality solution for the original problem. Tripathy [75] formulated an MIP model for a timetabling problem and presented a solution method based on coordinating Lagrangean relaxation and subgradient optimization with a branch-and-bound procedure using special ordered sets of variables.

Carrasco and Pato [21, 22] explored the application of a Potts neural network-based metaheuristic to a class-teacher timetabling problem. Avella *et al*. [10] proposed a local search algorithm for a high school timetabling problem taking into account both hard and soft constraints. The authors employed simulated annealing with a very large-scale neighborhood search approach, where this neighborhood was explored by solving an integer programming problem. Haan *et al*. [28] presented a three-step tabu search approach for generating a timetable for a Dutch high school. Beligianni *et al*. [15] developed a genetic algorithmic procedure for high schools in Greece and tested it using exhaustive computational experiments along with real-world data. Moura *et al*. [53] examined a constrained high school timetabling problem and applied a Greedy Randomized Adaptive Search Procedure (GRASP) heuristic, combined with a path-relinking improvement step. Zhang *et al*. [81] used a simulated annealing-based heuristic with a newly-designed neighborhood structure to approximately solve a high school timetabling problem, where the procedure performs a sequence of swaps between pairs of time-slots, instead of between pairs of assignments as in traditional simulated annealing approaches for this problem. Ramirez [63] used genetic algorithms to solve two high school timetabling problems, with and without a fixed master schedule, while employing smart operators during the mutation process. Kingston [42] presented several algorithms for the assignment of teachers and rooms to meetings, assuming that the meeting time-slots are known. These algorithms were tested on real-world data, the best of which is currently implemented within a free public software for high school timetabling. Raghavjee and NePillay [62] designed a two-phase genetic algorithmic approach for solving a high school timetabling problem in South Africa.

Dorneles *et.al* [30] proposed a novel framework for solving a variant of the high school timetabling problem for the Brazilian schools. The authors formulated a mixed-integer programming model and developed a fix-and-optimize heuristic approach combined with a variable neighborhood descent method to solve the developed model. Odeniyi *et.al.* [55] investigated the school timetabling problem of the Fakunle Comprehensive High School (FCHS) in Osogbo, Nigeria. The authors utilized a modified simulated annealing (MSA) approach for solving the underlying timetabling problem. Sørensen and Dahms [71] adopted a two-stage decomposition approach of an IP model formulated for a high school timetabling problem in Denmark. This particular timetabling problem aims to assign each lecture to both a time-slot and a classroom. The solution for the Stage I model is given as input to the Stage II model, where certain irreversible decisions are fixed in Stage I. Tassopoulos and Beligiannis [72, 73] developed and implemented swarm optimization based methods to solve the timetabling problem of high schools in Greece. Also, Valouxis *et.al.* [78] adopted and implemented a two phase approach to solve the Greek High School problem. In the first phase, a sequence of IP problems are iteratively solved in order to create a schedule for one day at a time. The second phase systematically selects pairs of days and attempts to move teaching events between days. Finally an archive of high school timetabling problems in different countries is given in [60].

1.3. Contribution and organization

The present paper contributes to the foregoing body of literature by developing for the first time specialized mixed-integer programming models and algorithms for the high school timetabling problem faced by the public educational system in Kuwait, which deals with many problem-specific constraining issues Initially, we formulated a comprehensive mixed-integer programming model for generating weekly schedules for teachers, but we found that the popular commercial package CPLEX was not able to solve even its linear programming relaxation for some realistically sized test problems due to the highly combinatorial nature of the problem and the large number of constraints and integer variables.

Consequently, we propose in this paper two decompositions schemes. The first of these is a two-stage solution approach developed to derive approximate solutions. In the first stage, we develop a mixed-integer programming model to determine the weekly time-slots of classes for all the class-sections, and in the second stage, we formulate another mixed-integer programming model to assign teachers to classes based on the already fixed time-slots of classes as determined from Stage I. The second proposed approach formulates an alternative model that is amenable to a column generation solution process for which we design an effective sequential fixing heuristic procedure. Unlike the original mixed-integer programming formulation that was found to be untenable, we demonstrate the practical utility of both of these approaches (particularly the latter) using a set of realistic test problems.

Moreover, from both modeling and algorithmic perspectives, several aspects of our methodology could benefit timetabling efforts in general. We stress here that the problem considered in this paper differs from that solved in [17, 56], where the authors assumed that each teacher is pre-assigned certain courses and class-sections, whereas in our case, these are determined within the developed models themselves. This affords a more comprehensive and practical solution framework.

The remainder of this paper is organized as follows. In Section 2, we delineate the basic assumptions of the problem and specify certain preliminary general notation that will be used throughout the paper. Other notation, indices, parameters, decision variables that will be needed to formulate the specific models developed in this paper are introduced in the respective sections. The proposed two-stage solution approach (which involves the formulation of two models, denoted by M1 and M2) is presented in Section 3. Another mixed-integer program (denoted M3), which selects valid combinations of weekly schedules for teachers from the set of all feasible weekly schedules is formulated in Section 4, where each column in this model corresponds to a valid teacher schedule. Due to the excessive number of such schedules that can be possibly generated, we design in Section 5 a specialized column generation method for solving its linear programming relaxation, based on which, a sequential variable-fixing heuristic is devised to solve Model M3. Computational results are presented in Section 6 for a number of test instances, including three real cases pertaining to high schools in Kuwait along with several simulated realistic test problems. Section 7 concludes the paper with a summary and some closing remarks.

2. Problem Assumptions and Preliminary Notation

This section presents the basic problem assumptions and certain generic notation that will be used throughout the paper, where other pertinent modeling constructs will be introduced in the respective sections.

2.1. Problem Assumptions

This section presents the assumptions that collectively describe the studied problem under consideration, and that also delineate some practical considerations specified by the Ministry of Education in Kuwait (MOE).

a) There are three grade-levels: tenth, eleventh, and twelfth. For the eleventh and twelfth grade, there are two specializations: Science and Liberal Arts. Thus there are five grade-specialization combinations in the studied high school system as delineated in Table 1.

b) For each grade-specialization combination, students are partitioned into groups of (approximately) the same size. Each group is called a *class-section* and resides in a particular classroom for all instructional sessions.

c) For each grade-specialization combination, there exists a fixed set of *classes* (or *subjects*) that must be taken by all enrolled students

Table 1. Grade levels and specializations.

Grade Level	Tenth	Eleventh	Eleventh	Twelfth	Twelfth
Specialization	General	Science	Liberal Arts	Science	Liberal Arts

d) A school week in Kuwait starts on Sunday and ends on Thursday. There are seven time-slots per school day, each lasting for forty minutes. Table 2 displays the daily schedule in terms of time-slots and breaks.

Table 2. Daily time-slots and breaks.

Time	7:40-8:20	8:25-9:05	9:10-9:50	9:50-10:10	10:10-10:50	10:55-11:35
Activity	Time-slot 1	Time -slot 2	Time -slot 3	Break 1	Time -slot 4	Time -slot 5
Time	11:35-11:55	11:55-12:35	12:40-1:20			
Activity	Break 2	Time -slot 6	Time -slot 7			

e) There are 35 available time-slots for any class-section during the week, 34 of which need to be assigned to classes (as requested by MOE) in order to cover all the required classes (subjects), where the students are required to attend the first six time-slots of each school day.

f) The number of time-slots that have to be allocated for teaching a particular class to a class-section during the week is known.

g) The teaching workforce is not fixed. Certain teachers are *a priori* selected by the administration, as for example, if such teachers have the desired teaching and administrative experience. However, other teachers are hired only on an as-needed basis during any year. The teaching load for teachers is specified by the school administration.

h) A feasible schedule for any teacher must satisfy the following restrictions: (1) minimum and maximum daily loads as well as the maximum weekly load requirements;
(2) at most one class can be assigned to this teacher during any time-slot of any school day; (3) the classes assigned to the teacher belong to the given subset of classes that he/she can teach; and (4) at most three classes are taught over consecutive time-slots by the teacher, if so desired.

i) Subjects are partitioned into 12 groups as displayed in Table 3, where this partitioning scheme is specified by the Ministry of Education, and where the teachers are also likewise partitioned into 12 corresponding groups based on their qualifications and certificates.

Remark 1

Even though Table 2 specifies seven time-slots as mandated by MOE, we define the set H (with $|H|$ time-slots), for use in our formulated models below in order to maintain the generality of the proposed approach. For our test instances and computational results presented in Section 6, we restore the actual value of $|H| = 7$, in which case we will have a total 35 available time-slots; however, only 34 of these need to be occupied as alluded to in Assumption (e) above.

Table 3. Partitions of subjects

Group 1	Group 2	Group 3	Group 4
Geology, Chemistry	Social Studies, Sociology, Geography, Philosophy, Psychology, Economics, History	French	Physical Education
Group 5	**Group 6**	**Group 7**	**Group 8**
Computer	English	Arabic	Religion
Group 9	**Group 10**	**Group 11**	**Group 12**
Biology, Scientific Culture	Mathematics	Home Economics	Physics

2.2. Preliminary notation and formulated models

This subsection presents preliminary sets, indices and input data that will be used through the paper and also briefly describe the three models that will be formulated in this paper. Note that other model-specific constructs and decision variables will be introduced in the pertinent sections. For this purpose, Tables 4 and 5 respectively present various sets and indices that will be used throughout the paper. Constants pertaining to input data are specified in Tables 6, and Table 7 summarizes the models that will be formulated in the subsequent sections along with associated subproblems and the definitions of feasible regions used to describe teachers schedules.

Table 4. General Sets

$G \equiv \{1,\ldots,5\}$	Set of grade-specialization combinations (hereinafter referred to as *grades*) as delineated in Table 1.
S^g for $g \in G$	Set of all the class-sections related to g .
C^g for $g \in G$	Set of all classes that are required to be taken by grade g.
$D = \{1,\ldots,5\}$	Set of school days of the week from Sunday to Thursday.
$H = \{1,\ldots,\lvert H \rvert\}$	Set of the daily time-slots, for a total of $\lvert H \rvert$ such slots. (see Remark 1 above).
T	Set of available teachers.
$\Phi_1 \subseteq T$	Subset of teachers who are selected for employment in advance.
$\Phi_2 \subseteq T$	The complement of Φ_1 in T.
$A_t^g \subseteq C^g$	Set of classes in C^g that can be taught by teacher $t \in T$.
$A_t \equiv \{(g,c) : g \in G,\ c \in A_t^g\}$	Set of grade-class pairs, so that a class in this pair can be taught by teacher $t \in T$.
$B_c^g \equiv \{t : (g,c) \in A_t\}$	Set of teachers who can teach class $c \in C^g$, for each $g \in G$.
$D_t \subseteq D$	Set of available days associated with teacher $t \in T$.
$H_{t,d} \subseteq H$	Set of available time-slots associated with teacher $t \in T$ on day $d \in D_t$.
$\Psi_1 \subseteq T$	Set of teachers who are willing to teach two classes consecutively.
$\Psi_2 \subseteq T$	Set of teachers who are willing to teach three classes consecutively, noting that Ψ_2 is the complement of Ψ_1 in T .
For each $g \in G$, $E_1^g = \{c \in C^g : O_c^g \le 5\}$ and $E_2^g = \{c \in C^g : O_c^g > 5\}$	These sets are defined to separately account (via E_2^g) for a teacher having to possibly teach a class more than once in a day for any group of students.

Table 5. General indices

$g \in G$	Indices for the grade-specialization combinations.
$s \in S^g$, for $g \in G$	Indices for the class-sections related to g.
$c \in C^g$, for $g \in G$	Indices for the classes that are required to be taken by grade g.
$d \in D$	Respective indices for the days from Sunday to Thursday.
$h \in H$	Indices for the time-slots given in Table 2.
$t \in T$	Indices for the available teachers.

Table 6. Input data

$O_c^g : c \in C^g, g \in G$	Number of time-slots that have to be allocated for teaching the class c to each class-section $s \in S^g$ during the week.
$N^g = \sum_{c \in C^g} O_c^g, g \in G$	Total number of time-slots, where $N^g = 34$.
$DL^{t,d}$ and $DU^{t,d}$ for $t \in T$ and $d \in D$	Respectively, the minimum and maximum teaching load of any selected teacher $t \in T$ during a given day $d \in D$ of the school week.
L^t	The maximum number of time-slots that may be assigned to teacher $t \in T$ during a school week as specified by the school administration.

Table 7. Formulated models

M1	The Stage I model for the two-stage approach, denoted TSA. This is a feasibility model that determines the weekly time-slots of classes for all the class-sections.
M2_i, for $i \in I$	The Stage II models for the two-stage approach TSA, where for each $i \in I$, Model M2_i assigns teachers from the set T_i to classes from the set C_i based on the already fixed time-slots of classes as determined from the Stage I model M1.
M3	An exact model to select a feasible weekly schedule from all valid schedules for each employed teacher. This model will be solved using column generation approach
FR_t for $t \in T$	Feasible region that characterizes, via defined set of constraints, all the feasible schedules for any given teacher $t \in T$.
SP^t for $t \in T$	Auxiliary pricing *subproblem* used within the proposed column generation approach in Section 5.
$\overline{\text{M}}$	The linear programming relaxation of any model M.

3. A Two-Stage Solution Approach

In this section, we present a two-stage solution approach, denoted by TSA, for our high school timetabling problem. In Stage I, a mixed-integer programming model is solved to determine the weekly time-slots of classes for all the class-sections, and in Stage II, another mixed-integer programming model is solved to assign the teachers to classes based on the already fixed time-slots of classes as determined from Stage I. The sets, parameters, decision variables, and input data used to define the proposed models M1 and M2 of the two-stage approach are delineated in Tables 8 and 9, and Models M1 and M2 are presented subsequently.

Table 8. Sets related to the two-stage approach

$C \equiv \{(g, c): g \in G, c \in C^g\}$	Set of pairs (g, c), the first element of which is the grade level and the second element is a specific class associated with that grade level.
$I = \{1,...,\alpha\}$	Set containing the class-teacher partitions as discussed in Assumption (i) above.
$T_1 ,..., T_\alpha$	Partitions of the set of teachers T based on grade-subject specializations.
$C_1 ,..., C_\alpha$	Partitions of the set C corresponding to $T_1,...,T_\alpha$, where grade-class combinations from C_i may only be taught by teachers from T_i, $\forall i \in I$.

Remark 2.

A) The partitioning of T and C specified in Table 8 satisfies the following restrictions:

a) $T_i \cap T_j = \varnothing$ and $C_i \cap C_j = \varnothing, \forall i, j \in I$ with $i \neq j$.

b) $\bigcup_{i \in I} T_i = T$ and $\bigcup_{i \in I} C_i = C$.

c) For each $i = \{1,...,\alpha\}$, a teacher in T_i may only be assigned to class-grade combinations from the set C_i, and class-grade combinations from the set C_i may only be taught by teachers from T_i.

B) The parameter $\lambda_{g,s,c,d,h}$ in Table 9 is defined based on the assignment of classes to time-slots as determined by the solution obtained in Stage I of the two-stage approach, and indicates whether a specific class is taught during a given time-slot of a day or not.

Table 9. Parameters, decision variables and input constants related to Models M1 and M2 of the two-stage approach

Parameters	
$\lambda_{g,s,c,d,h}$	$\begin{cases} 1 & \text{if the class specified by } (g,\, c) \in C_i \text{ is offered for the class-section } s \in S^g \\ & \text{during time-slot } h \text{ of day } d, \\ 0 & \text{otherwise.} \end{cases}$
Decision Variables	
$x_{g,s,c,d,h}$	$\begin{cases} 1 & \text{if class } c \in C^g \text{ of class-section } s \in S^g, \text{for } g \in G, \\ & \text{is assigned to time-slot } h \in H \text{ of day } d \in D, \\ 0 & \text{otherwise.} \end{cases}$
$y_{t,g,s,c}$	$\begin{cases} 1 & \text{if teacher } t \in T_i \text{ teaches the class prescribed by the pair } (g,c) \in C_i \\ & \text{for the class-section } s \in S^g, \\ 0 & \text{otherwise.} \end{cases}$
For $t \in \Phi_2$ w_t	$\begin{cases} 1 & \text{if teacher } t \in T \text{ is selected in the teaching workforce,} \\ 0 & \text{otherwise,} \end{cases}$ where we have $W_t \equiv 1$, $\forall\, t \in \Phi_1$.
	Input Data
$\Pi^i_{d,h}$ for $i \in I$, $d \in D$, and $h \in H$	Number of teachers from the set T_i who are available (for teaching duties) during time-slot $h \in H$ of day $d \in D$.

3.1. Stage I Model

The following *class-to-time-slot assignment model* (**M1**) aims to find a feasible solution to the following set of constraints:

M1: Feasibility problem, subject to**:**

$$\sum_{d \in D} \sum_{h \in H} x_{g,s,c,d,h} = O_c^g, \ \forall\, g \in G,\, s \in S^g, c \in C^g, \tag{3.1}$$

$$\sum_{c \in C^g} x_{g,s,c,d,h} = 1, \ \forall\ g \in G,\, s \in S^g,\, d \in D,\, h \in H, \tag{3.2}$$

$$\sum_{h\in H} x_{g,s,c,d,h} \le 1,\ \forall\ g\in G,\ s\in S^g,\ c\in E_1^g,\ d\in D, \tag{3.3}$$

$$1\le \sum_{h\in H} x_{g,s,c,d,h} \le 2,\ \forall\ g\in G,\ s\in S^g,\ c\in E_2^g,\ d\in D, \tag{3.4}$$

$$\sum_{(g,c)\in C_i}\sum_{s\in S^g} x_{g,s,c,d,h} \le \Pi_{d,h}^i,\ \forall\ i\in I,\ h\in H,\ d\in D, \tag{3.5}$$

$$x_{g,s,c,d,h}\in\{0,1\},\quad \forall\ g\in G,\ s\in S^g,\ c\in C^g,\ d\in D,\ h\in H. \tag{3.6}$$

Constraint (3.1) assures that for each $g\in G$ and $s\in S^g$, class $c\in C^g$ is assigned to the required number of time-slots given by O_c^g over a school week. Constraint (3.2) assures that exactly one class $c\in C^g$ is assigned to the time-slot $d\in D$ and $h\in H$ specified for class-section $g\in G,\ s\in S^g$. Moreover, note that any time-slot can accommodate at most one class-teacher combination for each $s\in S^g,\ g\in G$. For each $s\in S^g, g\in G$, a class $c\in C^g$ may be assigned to more than two time-slots per school day only if $O_c^g>5$ (*i.e.*, if $c\in E_2^g$). In this case, such a class must be assigned at least one time-slot and at most two time-slots for any day. On the other hand, a class $c\in E_1^g$, can be assigned at most one time-slot on any day. These requirements are guaranteed via Constraints (3.3) and (3.4). For a given $i\in I$, Constraint (3.5) assures that the number of classes from the set C_i, over all the grades and corresponding class-sections that are offered during a given time-slot $h\in H$ of some day $d\in D$, does not exceed the number of correspondingly available teachers from the set T_i (as denoted by $\Pi_{d,h}^i$).

3.2. Formulation of the Stage II Model

Stage I has thus far determined the time-slots of the classes that need to be offered for all the class-sections. In Stage II, for each $i=I$, we formulate a mixed-integer programming model that assigns teachers in the set T_i to class-sections pertaining to grade-class pairs (g,c) in C_i. Hence, for each $i\in I$, the corresponding Stage II *teacher-to-class-and-time-slot* assignment model ($\mathbf{M}2_i$) is formulated as follows:

M 2_i **:**

Minimize $\sum_{t\in T_i\cap\Phi_2} w_t$

subject to

$$\sum_{t\in T_i} y_{t,g,s,c} = 1,\ \forall (g,c)\in C_i,\ s\in S^g, \tag{3.7}$$

$$\sum_{(g,c)\in C_i}\sum_{s\in S^g} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le 1,\ \forall\ t\in T_i,\ d\in D_t,\ h\in H_{t,d}, \tag{3.8}$$

$$DL^{t,d} \le \sum_{(g,c)\in C_i}\sum_{s\in S^g}\sum_{h\in H_{t,d}} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le DU^{t,d},\ \forall\ t\in T_i\cap\Phi_1,\ d\in D_t, \tag{3.9}$$

$$DL^{t,d}\ w_t \le \sum_{(g,c)\in C_i}\sum_{s\in S^g}\sum_{h\in H_{t,d}} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le DU^{t,d}\ w_t,\quad \forall\ t\in T_i\cap\Phi_2,\ d\in D_t, \tag{3.10}$$

$$\sum_{(g,c)\in C_i}\ \sum_{d\in D_t}\ \sum_{s\in S^g}\ \sum_{h\in H_{t,d}} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le L^t,\ \forall\ t\in T_i, \tag{3.11}$$

$$\sum_{(g,c)\in C_i}\sum_{s\in S^g}\sum_{h=h'}^{h'+2} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le 2,$$
$$\forall\ t\in \Psi_1\cap T_i,\ d\in D_t,\ h'\in H_{t,d}: \{h'+1, h'+2\}\subseteq H_{t,d}, \tag{3.12}$$

$$\sum_{(g,c)\in C_i}\sum_{s\in S^g}\sum_{h=h'}^{h'+3} \lambda_{g,s,c,d,h}\ y_{t,g,s,c} \le 3,$$
$$\forall\ t\in \Psi_2\cap T_i,\ d\in D_t,\ h'\in H_{t,d}: \{h'+1,\ldots,h'+3\}\subseteq H_{t,d}, \tag{3.13}$$

$$y_{t,g,s,c} \equiv 0 \text{ if } \lambda_{g,s,c,d,h} = 0 \text{ for all } (d,h): d\in D_t,\ h\in H_{t,d}, \forall t\in T_i,\ (g,c)\in C_i, s\in S^g,$$
$$y_{t,g,s,c}\in\{0,1\},\ \forall\ t\in T_i,\ (g,c)\in C_i,\ s\in S^g. \tag{3.14}$$

For $i\in I$, the objective function of Model M2_i aims to minimize the number of teachers from the set $T_i\cap\Phi_2$. Constraint (3.7) assures that each class-section pertaining to the pair $(g,c)\in C_i$ is assigned to a single suitable teacher from the set T_i. For any given time-slot of a school day, a teacher may teach at most one class as enforced by Constraint (3.8). The teaching load of a selected teacher must lie within some specified minimum and maximum daily teaching loads. However, if a teacher is not selected, then the daily teaching load is naturally zero. These restrictions are enforced by Constraints (3.9) and (3.10). Note that the teachers from the set Φ_1, along with the subset of teachers from Φ_2 that is determined via the w-variables in Constraint (3.10), constitute the required teaching workforce. The maximum weekly teaching load for each teacher is restricted by Constraint (3.11). Constraint (3.12) assures that teachers in the set Ψ_1 may teach at most two classes consecutively, while teachers in the set Ψ_2 may teach at most three classes consecutively as enforced via Constraint (3.13).

4. An Exact Model with Special Column Structures

In this section, we present an exact high school timetabling model (M3) based on selecting a feasible weekly schedule from all valid schedules for each employed teacher, instead of implicitly representing the teachers' schedules within a directly formulated model. The parameters and decision variables used to define Model M3 are listed next in Table 10.

Table 10. Parameters and decision variables related to Model M3.

Parameters and decision variables	
$\alpha^1_{t,k,g,s,c}$	$\begin{cases} 1 & \text{if teacher } t \in T \text{ is assigned to class } c \in A^g_t \text{ for class-section } s \in S^g, \\ & \text{of grade } g \in G, \text{ based on schedule } k \in K_t, \\ 0 & \text{otherwise.} \end{cases}$
$\alpha^2_{t,k,g,s,c}$	$\begin{cases} 1 & \text{if teacher } t \in T \text{ is assigned to a class from the set } A^g_t \\ & \text{for class-section } s \in S^g, \text{ of grade } g \in G \text{ during time-slot } h \in H_{t,d} \\ & \text{of day } d \in D_t, \text{ based on schedule } k \in K_t, \\ 0 & \text{otherwise.} \end{cases}$
$\beta_{t,k}$	$\begin{cases} 1 & \text{if } t \in \Phi_2, \\ 0 & \text{if } t \in \Phi_1. \end{cases}$
$z_{t,k}$	$\begin{cases} 1 & \text{if schedule } k \in K_t \text{ is selected for teacher } t \in T, \\ 0 & \text{otherwise.} \end{cases}$
$u_{t,g,s,c,d,h}$	$\begin{cases} 1 & \text{if teacher } t \in T \text{ teaches class } c \in A^g_t \text{ for class-section } s \in S^g \\ & \text{at the grade level } g \in G \text{ during time-slot } h \in H_{t,d} \text{ of day } d \in D_t, \\ 0 & \text{otherwise.} \end{cases}$
$v_{t,g,s,c}$	$\begin{cases} 1 & \text{if teacher } t \in T \text{ teaches class } c \in A^g_t \text{ for class-section } s \in S^g \\ & \text{at the grade level } g \in G, \\ 0 & \text{otherwise.} \end{cases}$

Note that the values of the α-and β-parameters of Table 10 are known *a priori* for any given teacher's schedule; these parameters define each column in Model M3 as described later in Section 5. The proposed model M3 is given as follows:

M3:

Minimize $\sum_{t\in T} \sum_{k\in K_t} \beta_{t,k} \; z_{t,k}$

subject to

$$\sum_{t\in B_c^g} \sum_{k\in K_t} \alpha^1_{t,k,g,s,c} \; z_{t,k} = 1, \;\; \forall g \in G,\; s\in S^g,\; c\in C^g, \tag{4.1}$$

$$\sum_{t\in T} \sum_{k\in K_t} \alpha^2_{t,k,g,s,d,h} \; z_{t,k} = 1, \;\; \forall \; g \in G,\; s\in S^g,\; d\in D,\; h\in H, \tag{4.2}$$

$$\sum_{k\in K_t} z_{t,k} = 1, \;\; \forall \; t\in T \bigcap \Phi_1, \tag{4.3}$$

$$\sum_{k\in K_t} z_{t,k} \le 1, \;\; \forall \; t\in T \bigcap \Phi_2, \tag{4.4}$$

$$z_{t,k} \in \{0,1\}, \;\; \forall \; t\in T,\; k\in K_t.$$

Also, for a given $t\in T$, $g \in G$, and $s\in S^g$, if $c \notin A_t^g$, then $\alpha^1_{t,k,g,s,c}$ (defined in Table 10 above) is set to zero for all $k\in K_t$.

Objective function and constraints

The objective function in Model M3 aims to minimize the number of selected teachers from the optional set Φ_2. Constraint (4.1) guarantees that every class that is required to be taken by a class-section must be taught by an appropriate teacher. Each class-section must be taught exactly one class during the specified time-slots of any school day as enforced by Constraint (4.2). A *selected* teacher must follow exactly one feasible weekly schedule, as enforced by Constraints (4.3) and (4.4).

5. Analysis of Model M3 and a Column Generation Approach

This section is composed of three parts. In Subsection 5.1, we formulate a set of constraints for a given teacher *t* whose feasible region characterizes all valid schedules for this teacher. Subsection 5.2 presents a column generation approach to solve the linear programming relaxation of Model M3, based on which, we devise in Subsection 5.3 a column generation heuristic to derive a solution for Model M3 (see [12, 13, 54] for a general discussion on column generation). Let FR_t, for $t \in T$, denote the feasible region that characterizes the set of columns of the coefficient matrix of M3 that are associated with the $z_t \equiv (z_{t,1},\ldots,z_{t,|K_t|})$-variables. This region is described in Subsection 5.1 below. Tables 11 and 12 summarizes specific sets, indices, dual solutions associated with model $\overline{\mathrm{M3}}$, and other related notation pertinent the proposed column generation approach.

5.1. Characterization of the columns of Model M3 via a set of constraints

The feasible region FR_t, which characterizes all the feasible schedules for any given teacher $t \in T$, is presented next in terms of the *u*-and *v*-variables defined in Table 10.

FR*t*:

$$\sum_{d \in D_t} \sum_{h \in H_{t,d}} u_{t,g,s,c,d,h} = O_c^g \; v_{t,g,s,c}, \; \forall \; g \in G, s \in S^g, c \in A_t^g, \tag{5.1}$$

$$\sum_{h \in H_{t,d}} u_{t,g,s,c,d,h} \le v_{t,g,s,c}, \; \forall g \in G, s \in S^g, c \in A_t^g \cap E_1^g, \; d \in D_t, \tag{5.2}$$

$$v_{t,g,s,c} \le \sum_{h \in H_{t,d}} u_{t,g,s,c,d,h} \le 2\, v_{t,g,s,c}, \; \forall g \in G, s \in S^g, c \in A_t^g \cap E_2^g, \; d \in D_t, \tag{5.3}$$

$$u_{t,g,s,c,d,h} \le v_{t,g,s,c}, \; \forall \; g \in G, s \in S^g, c \in A_t^g \cap E_2^g, \; d \in D_t, h \in H_{t,d}, \tag{5.4}$$

$$\sum_{g \in G} \sum_{s \in S^g} \sum_{c \in A_t^g} u_{t,g,s,c,d,h} \le 1, \; \forall \; d \in D_t, h \in H_{t,d}, \tag{5.5}$$

$$DL^{t,d} \le \sum_{g \in G} \sum_{s \in S^g} \sum_{c \in A_t^g} \sum_{h \in H_{t,d}} u_{t,g,s,c,d,h} \le DU^{t,d}, \; \forall \; d \in D_t, \tag{5.6}$$

$$\sum_{g \in G} \sum_{s \in S^g} \sum_{c \in A_t^g} \sum_{d \in D_t} \sum_{h \in H_{t,d}} u_{t,g,s,c,d,h} \le L^t, \; \forall \; t \in T, \tag{5.7}$$

$$\sum_{g\in G}\sum_{s\in S^g}\sum_{c\in A_t^g}\sum_{h=\bar{h}}^{\bar{h}+2} u_{t,g,s,c,d,h} \le 2,\ \forall\ t\in \Psi_1,\ d\in D_t,\ \bar{h}\in H_{t,d} : \{\bar{h}+1, \bar{h}+2\} \subseteq H_{t,d}, \tag{5.8}$$

$$\sum_{g\in G}\sum_{s\in S^g}\sum_{c\in A_t^g}\sum_{h=\bar{h}}^{\bar{h}+3} u_{t,g,s,c,d,h} \le 3,\ \forall\ t\in \Psi_2,\ d\in D_t,\ \bar{h}\in H_{t,d} : \{\bar{h}+1, \ldots, \bar{h}+3\} \subseteq H_{t,d}, \tag{5.9}$$

$$u_{t,g,s,c,d,h} \in \{0,1\} \text{ and } v_{t,g,s,c} \in \{0,1\},\ \forall\, g\in G,\ s\in S^g,\ c\in A_t^g,\quad d\in D_t,\ h\in H_{t,d}. \tag{5.10}$$

Constraints

Constraint (5.1) assures that if the given teacher t is assigned to class $c\in A_t^g$ of grade $g\in G$ for section $s\in S^g$, then this teacher is responsible for teaching all the required hours (credits) of this class, as given by O_c^g. Class offering restrictions related to a maximum offering either of one or two sections are, respectively, enforced by Constraints (5.2) and (5.3). The valid inequality given by Constraint (5.4) serves to tighten the linear programming relaxation, where this inequality is implied by Constraint (5.2) for the case of $c\in A_t^g \bigcap E_1^g$. Teacher t may teach at most one class $c\in A_t^g$, $g\in G$, to any class-section $s\in S^g$ during any time-slot $h\in H_{t,d}$ of day $d\in D_t$, as enforced Constraint (5.5). If $t\in \Phi_1$, the teaching load of t must lie within some specified minimum and maximum daily teaching loads. However, if this teacher is not selected, then the daily teaching load is naturally zero. These restrictions are enforced via Constraint (5.6). The maximum weekly teaching load for each teacher is restricted via Constraint (5.7). Constraint (5.8) assures that teachers in the set Ψ_1 may teach at most two classes consecutively. Also, teachers in the set Ψ_2 may teach at most three classes consecutively as modeled by Constraint (5.9). Constraint (5.10) enforces the binary restriction requirements on the u-and v-variables; however, the following proposition proves that with the u-variables being binary-valued, we may relax this restriction for the v-variables.

Table 11. Sets related to the column generation approach

$K_1 = \{1,\ldots,\lvert K_1\rvert\}$	Index set of the potential feasible schedules of teacher 1.
$K_2 = \{\lvert K_1\rvert+1,\ldots,\lvert K_1\rvert+\lvert K_2\rvert\}$	Index set of the potential feasible schedules of teacher 2, where the index sets for the remaining teachers are defined similarly.
$K = \{1,\ldots, p\}$	Index set for all the potential feasible schedules of the teachers, where $p \equiv \sum_{t\in T}\lvert K_t\rvert$.
S_{nb}	Index set for the nonbasic variables associated with an optimal basic feasible solution to the linear programming relaxation (denoted by $\overline{\mathrm{M3}}$) at any particular iteration of the revised simplex method.
S_b	Index set for the basic variables associated with an optimal basic feasible solution to $\overline{\mathrm{M3}}$ at any particular iteration of the revised simplex method.
S_b^1 and S_b^f	Index sets of basic variables $(\subseteq S_b)$ that are equal to one and that are fractional-valued, respectively, in the optimal solution obtained for Model $\overline{\mathrm{M3}}$.
J	A set consisting of the indices corresponding to the teachers' schedules that are selected for implementation (this set is constructed iteratively by the proposed column generation algorithm).
$T^J \equiv \{t \in T:$ there exists an index j in the set J that corresponds to teacher $t\}$.	
$\overline{T}^J$	Complement of T^J in T.
$\overline{J}$	Set of indices in $\{1,\ldots, p\}$ that are associated with teachers in $\overline{T}^J$.

Proposition 1. Given that the u-variables are binary-valued in FR$_t$, for any $t \in T$, the v-variables in FR$_t$ will automatically take on binary values for any feasible solution, when restricted to be simply nonnegative.

Proof

This can be verified by noting that if any of the u-variables corresponding to the combination (g, s, c) takes a value of one, then Constraints (5.2) and (5.4) force $v_{t',g,s,c}$ to have a value of one for $c \in E_1^g$ and $c \in E_2^g$, respectively. □

Note that the values of the α-parameters based on a solution to FR_t are accordingly given as follows for all relevant index values:

$$\alpha^1_{t,k,g,s,c} = v_{t,g,s,c} \text{ and } \alpha^2_{t,k,g,s,d,h} = \sum_{c \in A_t^g} u_{t,g,s,c,d,h}.$$

Table 12. Indices, complementary dual solution associated with Model $\overline{\mathrm{M3}}$, and other related notation

Indices	
$j \in K_t$	Indices for all schedules associated with teacher $t \in T$.
$j \in K$	Indices for all the potential feasible schedules of the teachers.
$\zeta^1, \ldots, \zeta^4$ complementary dual solution at any iteration of the revised simplex method for Model $\overline{\mathrm{M3}}$ associated respectively with Constraints (4.1)-(4.4), where	
$\zeta^1 \equiv \left(\zeta^1_{g,s,c},\ g \in G,\ s \in S^g,\ c \in C^g\right)$.	
$\zeta^2 \equiv \left(\zeta^2_{g,s,d,h},\ g \in G,\ s \in S^g,\ d \in D,\ h \in H\right)$.	
$\zeta^3 \equiv \left(\zeta^3_t,\ t \in T \cap \Phi_1\right)$.	
$\zeta^4 \equiv \left(\zeta_t,\ t \in T \cap \Phi_2\right)$.	
$z_j \equiv z_{t,k}$ and $c_j \equiv c_{t,k}$, where the index j corresponds to the combination (t, k).	
$\bar{z}$: A solution of Model $\overline{\mathrm{M3}}$ that is obtained using the column generation procedure described in Subsection 5.2.	
$\hat{j} \in \underset{j \in S_b^{fr}}{\mathrm{arglexmax}}\{\bar{z}_j, -c_j\}$	

5.2. A column generation method CGM to solve Model $\overline{\mathrm{M3}}$

In this subsection, we present a *column generation method* (CGM) to solve the continuous relaxation $\overline{\mathrm{M3}}$ of Model M3, which will then be used in Subsection 5.3 within a column generation heuristic to derive a solution for Model M3. Toward this end, suppose that at some iteration of the revised simplex method as applied to solve $\overline{\text{M3}}$, we have a basic feasible solution with S_{nb} representing the index set for the nonbasic variables, and let ζ denoting the corresponding complementary dual solution, with components $(\zeta^1,...,\zeta^4)$ associated with Constraints (4.1)-(4.4) as defined Table 12 above. We first explicitly price the nonbasic slack variables associated with Constraint (4.4) and identify a most enterable candidate nonbasic variable. If any exists, we enter such a variable into the basis and accordingly update the basis and repeat. Hence, suppose that no such variable is enterable. We next implicitly price the (nonbasic) z-variables to find a candidate nonbasic z-variable that has the smallest (most negative) reduced cost to enter the basis by solving the following subproblem SP^t for each $t \in T$, noting that $c^t \equiv 0$ if $t \in \Phi_1$ and $c^t \equiv 1$ if $t \in \Phi_2$:

SP^t: Minimize

$$c^t - \left[\sum_{g \in G} \sum_{s \in S^g} \sum_{c \in A_t^g} \zeta^1_{g,s,c} \; v_{t,g,s,c} + \sum_{g \in G} \sum_{s \in S^g} \sum_{d \in D_t} \sum_{h \in H_{t,d} \cap H} \zeta^2_{g,s,d,h} \left(\sum_{c \in A_t^g} u_{t,g,s,c,d,h} \right) + \zeta^\delta_t \right],$$

subject to $(u^t, v^t) \in \text{FR}_t$, where $\delta = 4$ if $t \in T \cap \Phi_1$, and $\delta = 5$ if $t \in T \cap \Phi_2$.

Letting opt(M) denote the optimal objective function value of any Model M, then for any given teacher t, if $\text{opt}(\text{SP}^t) \geq 0$, we have that none of the $z_{t,k}$-variables are enterable into the basis; on the other hand, if $\text{opt}(\text{SP}^t) < 0$, then we will have obtained a candidate entering variable $z_{t,k}$ for Model $\overline{\text{M3}}$ from the optimal solution obtained for Model SP^t. We thus enter each of the identified enterable variables for the different teachers sequentially (for teachers in the set Φ_1 first, and then for teachers in the set Φ_2), provided they remain enterable with respect to the updated basis at each step, and we terminate this process when no enterable variable exists.

Remark 3. For the sake of convenience and efficiency, we use a set of initial columns derived from a manually generated schedule to compose a basis along with artificial columns as necessary. This construct enabled us to solve Model $\overline{\text{M3}}$ via column generation relatively easily, without having to resort to any dual stabilization techniques (see [11], for example). Note that there is no standard existing method for generating manual solutions, and administrators in different schools adopt different approaches. Typically, a manual solution is obtained in two-stages similar to that adopted in Section 3 of the paper, except that each stage is handled manually.

5.3. A Column Generation Heuristic CGH for Model M3

In this subsection, we discuss a column generation-based sequential variable-fixing heuristic procedure for constructing a good quality feasible solution to Model M3. This method recursively applies CGM as described in the foregoing section in order to derive a valid schedule for each teacher.

Consider the solution $\bar{z}$ to Model $\overline{\text{M3}}$ that is obtained using the CGM procedure. Let S_b denote the index set for the basic variables in this solution, and let S_b^1 and S_b^f be the index sets of basic variables that are equal to one and that are fractional-valued, respectively. Note that if $S_b^f = \varnothing$, we have at most $|T|$ schedules at hand, and we can stop with this solution as optimal to Model M3. Otherwise, we initialize a set $J \equiv S_b^1$, where each index in the set J corresponds to a teacher's schedule.

Let $\hat{j} \in \underset{j \in S_b^{fr}}{\text{arglexmax}}\{\bar{z}_j, -c_j\}$, and augment the set $J \leftarrow J \bigcup \{\hat{j}\}$. Note that no two indices in this updated set J could correspond to the same teacher, because otherwise, Constraint (4.3) or (4.4) would be violated by the solution $\bar{z}$. Let $T^J \equiv \{t \in T :$ there exists an index j in the set J that corresponds to teacher $t\}$, and let $\overline{T}^J$ denote the complement of T^J in T. Let $\overline{J}$ be the set of indices in $\{1,\ldots, p\}$ that are associated with teachers in $\overline{T}^J$. Consider a modified version of M3, denoted by M3_{T^J}, which is stated as follows:

$\mathbf{M3}_{T^J}$ **:**

$$\text{Minimize } \left\{ \sum_{j \in J \bigcup \overline{J}} c_j z_j : (4.1)\text{-}(4.4),\ \ z_j = 1,\ \forall j \in J;\ \ z_j \in \{0,1\},\ \forall j \in \overline{J} \right\}.$$

We can now solve the linear programming relaxation $\overline{\mathrm{M3}_{T^J}}$ of Model $\mathrm{M3}_{T^J}$ using the foregoing CGM procedure, where for each $t \in \overline{T}^J$, the feasible region FR_t is updated by setting $\{ u_{t,g,s,c,d,h} = 0, \forall d \in D_t, h \in H_{t,d}$, and $v_{t,g,s,c} = 0 \}$, $\forall\, t \in \overline{T}^J$, and for each $c \in C^g$ associated with class-section $s \in S^g$ of grade $g \in G$ that is already assigned to some teacher $\bar{t} \in T^J$. Denote this updated feasible region as FR_t^J for the current set J, $\forall\, t \in \overline{T}^J$. The overall proposed column generation heuristic (**CGH**) procedure for selecting schedules for teachers then proceeds as follows:

Procedure CGH

Initialization

- Set $J = \varnothing$, $\overline{J} = \{1,..., p\}$, $T^J = \varnothing$, $\overline{T}^J = T$, and $\mathrm{FR}_t^J \equiv \mathrm{FR}_t, \forall t \in T$.

Main Step

- Solve Model $\overline{\mathrm{M3}_{T^J}}$ using CGM and let $\bar{z}$ denote the resulting solution.
- Determine the index sets S_b^1 and S_b^f as defined above.
- If $S_b^f = \varnothing$, then stop; the required schedules for the selected teachers are collectively described by the set J. Otherwise, proceed to the next step.
- Let $\hat{j} \in \underset{j \in S_b^f}{\text{arglex max}}\{\bar{z}_j, -c_j\}$, and update $J \leftarrow J \cup S_b^1 \cup \{\hat{j}\}$.
- Update $\overline{J}$, T^J, $\overline{T}^J$, and FR_t^J, $\forall\, t \in \overline{T}^J$, as discussed above, and repeat the Main Step.

Remark 4. Observe that we are using a "diving" heuristic in CGH with no backtracking because of the complexity of the problem. However, one issue that might cause infeasibility is the lower-bounding restriction of Constraint (5.6), whereby at some iteration, it is possible that the subproblems SP^t are infeasible for all $t \in \overline{T}^J$, whence no entering column would be generated, and so, the overall problem has no feasible completion. In such a case, we could relax these lower bounding restrictions. Also, at each iteration of our heuristic, we can add artificial variables, representing adjunct teachers, within Model $M3_{T^J}$ to ensure its feasibility. If some of the artificial variables remain positive at termination of the heuristic, administrators would need to consider hiring appropriate adjunct teachers to cover schedules assigned to such artificial variables. (Alternatively, we can resort to the two-stage approach of Section 3 in order to investigate the possibility of generating a feasible solution using the available mix of teachers).

6. Computational Results

In this section, we present computational results related to solving the defined high school timetabling problem via the two-stage approach (TSA) and the proposed column generation heuristic (CGH). We used the software package CPLEX-MIP (version 12.0) with its default settings to solve the different subproblems within the developed solution procedures. All runs were made on a Pentium IV, CPU 3.00 GHz computer having 1.99GB of RAM with coding in Java. We consider six practical test problem instances as indicated in Table 13, where the first three are from different high schools in Kuwait, and the other three are simulated realistic test cases based on similar data that involves more grade-sections than those in the first three test problems. A manually generated solution obtained via an *ad hoc* procedure implemented in practice was used for each of the test cases to initiate CGH, where the number of teachers needed (NT) for each case in this initial solution is specified in the final column of Table 14. Basic data related to the class requirements for each grade-level is provided in Table 15, while selected manually generated initial solutions, as well as selected schedules generated via the proposed modeling approaches can be gleaned at www.al-yakoob.com. This section is composed of two parts. Subsection 6.1 presents particular implementation issues that are required by MOE, and then Subsection 6.2 provides the pertinent computational results.

6.1. Implementation issues

Recall that, to maintain the generality of our modeling approach, we assumed that there are $|H|$ time-slots during each day that are available for each class-section. However for our particular problem, as indicated in Assumption (e) and Remark 1 of subsections 2.1 and 2.2, respectively, we have that $|H|$=7 has been specified by the Ministry of Education. Although this induces 35 available time-slots, only 34 of these need to be assigned to classes, in a manner such that the first six time-slots for each day are occupied, while only four of the daily seventh time-slots are occupied (*i.e.*, one of the seventh daily time-slots will be unoccupied). This requires splitting Constraint (3.2) of Model M1 into two constraints as follows:

$$\sum_{c\in C^g} x_{g,s,c,d,h} = 1,\ \forall\ g\in G,\ s\in S^g,\ d\in D,\ h\in\{1,\ldots,6\}, \tag{3.2.1}$$

$$\sum_{c\in C^g} x_{g,s,c,d,7} \le 1,\ \forall\ g\in G,\ s\in S^g,\ d\in D, \tag{3.2.2}$$

In this context, each class-section must be taught exactly one class during the first six time-slots of any school day as enforced by Constraint (3.2.1), and at most one class during the seventh time-slot as guaranteed by Constraint (3.2.2).

Similarly, Constraint (4.2) of Model M3 must be split into two constraints as follows:

$$\sum_{t\in T}\ \sum_{k\in K_t} \alpha^2_{t,k,g,s,d,h}\ z_{t,k} = 1,\ \forall\ g\in G,\ s\in S^g,\ d\in D,\ h\in\{1,\ldots,6\}, \tag{4.2.1}$$

$$\sum_{t\in T}\ \sum_{k\in K_t} \alpha^2_{t,k,g,s,d,7}\ z_{t,k} \le 1,\ \forall\ g\in G,\ s\in S^g,\ d\in D. \tag{4.2.2}$$

This splitting of Constraint (4.2) requires the partitioning of the complementary dual solution ζ^2 into $\zeta^{2.1}$ and $\zeta^{2.2}$ as follows:

$$\zeta^{2.1} \equiv \left(\zeta^2_{g,s,d,h},\ g\in G,\ s\in S^g,\ d\in D,\ h\in\{1,\ldots,6\}\right),$$

$$\zeta^{2.2} \equiv \left(\zeta^3_{g,s,d},\ g\in G,\ s\in S^g,\ d\in D\right).$$

Based on this, the objective function of Problem SP^t becomes

$$c^t - \left[\sum_{g\in G}\sum_{s\in S^g}\sum_{c\in A_t^g}\zeta^1_{g,s,c}\ v_{t,g,s,c} + \sum_{g\in G}\sum_{s\in S^g}\sum_{d\in D_t}\ \sum_{h\in H_{t,d}\cap\{1,\ldots,6\}} \zeta^{2.1}_{g,s,d,h}\left(\sum_{c\in A_t^g} u_{t,g,s,c,d,h}\right) + \right.$$

$$\left.\sum_{g\in G}\sum_{s\in S^g}\sum_{d\in D_t} \zeta^{2.2}_{g,s,d}\left(\sum_{c\in A_t^g} u_{t,g,s,c,d,h}\right) + \zeta^{\delta}_t\right].$$

6.2. Comparison of results obtained via TSA, CGH, and a manual approach

In this subsection, we present computational results related to applying our two modeling solution approaches TSA and CGH, and we compare the results obtained against a manual used timetabling approach for generating timetables. Table 13 summarizes the notation that will be used throughout this section.

Table 13. Notation for computational results

$\text{opt}(\text{M})$	Optimal objective function value for any model M.
$\text{ofv}^{\text{TSA}} = \sum_{i=1}^{\alpha} \text{opt}(\text{M2}_i)$	The total objective function value obtained using the two-stage approach (TSA).
$\text{ofv}_{\overline{\text{M3}}}^{\text{CGM}}$	Objective function value of the linear programming relaxation of Model M3 obtained using Procedure CGM.
$\text{ofv}_{\text{M3}}^{\text{CGH}}$	Objective function value of Model M3 obtained using Heuristic CGH.
$\tau_1(lb, ub) = 100\left(\frac{ub - lb}{ub}\right)\%$	Optimality gap percentage, where lb and ub are respectively lower and upper bounds obtained for the particular model.
$\tau_2(\text{ofv}_1, \text{ofv}_2) = 100\left(\frac{\text{ofv}_2 - \text{ofv}_1}{\text{ofv}_2}\right)\%$	Percentage improvement provided by objective value ofv_1 attained by some specified approach over the objective value ofv_2 produced by another approach. Note that the ofv_1- and ofv_2-values represent the required number of teachers prescribed by the respective approaches.
NT	Number of teachers required for the manually generated schedules.
RT	Run-time in CPU seconds.

Tables 16-19 present computational results related to solving Models M1 and M2_i, for $i \in I$, directly via CPLEX, and for solving Model M3 via the proposed column generation heuristic CGH. Note that the only available lower bound for the problem is given by $\text{opt}(\overline{\text{M3}})$, and hence the optimality gaps reported in these tables for the solutions obtained via TSA and CGH are both calculated with respect to $\text{opt}(\overline{\text{M3}})$.

As evident from Table 16, solutions for Model M1 were readily obtained, noting that M1 is a simple feasibility model that does not involve any objective function. The two-stage approach (TSA) derived feasible solutions for the six test problems in relatively short CPU times, but assured only an optimality gap of 18.48% on average as displayed in Table 18, with the lowest gap of 7.14% obtained for the instance P_1 and the highest of 27.45% obtained for P_5. Also, as evident from Tables 17 and 18, solutions to Models $\overline{M3}$ and M3 for all the six test problems were also readily obtained in relatively short CPU times using CGM and CGH, respectively, with almost identical run-times. Furthermore, the relative tightness of Model $\overline{M3}$ is evident from the respective values obtained for $\text{opt}(\overline{M3})$ and M3, which are identical for test problems P_2 and P_4, substantiating the effectiveness and robustness of the proposed column generation method for solving the underlying problem. Heuristic CGH outperformed TSA with respect to CPU times and the total number of teachers needed (see Table 17), as well as achieved a much improved optimality gap average of 2.48% as seen from Table 18, which is a seven-fold reduction from that obtained for TSA (18.48%). In particular, the solutions obtained for test problems P_2 and P_4 using CGH were optimal as evident from the matching values of their respective lower and upper bounds, whereas the optimality gaps for these problems using TSA are given by 11.43% and 27.45%, respectively. The percentage improvement of CGH over TSA varies from 3.57% for test problem P_1 to 27.45% for test problem P_4. Also, given the satisfactory performance of the implemented column generation procedure, no dual stabilization technique was found necessary to enhance the solvability of Model $\overline{M3}$ (see, for example, Bazaraa *et al*. [13] for a general discussion on such techniques). Furthermore, it turned out that no artificial variables were needed to ensure the feasibility of Model $\overline{M3}$ or of its restricted versions $\overline{M3_{T^j}}$.

Furthermore, the average number of teachers needed using TSA was about 46, which produced an average daily load of 3.1 time-slots per teacher, while that obtained via CGH was 38, resulting in an average daily load of 3.8 time-slots per teacher (Table 19). On the other hand, the manual *ad hoc* approach resulted in an average of 50 teachers to meet the school requirements, thus yielding the lowest average daily load (i.e., the lowest utilization rate) of 2.9 time-slots per teacher. On average, therefore, the percentage improvement in objective value of CGH over TSA was about 16.39% (see Table 18), while that of TSA and CGH over the manual approach were, respectively, 7.06% and 22.32%. Moreover, CGH required, on average, less than a third of the CPU time consumed by TSA to solve the problems. Hence, the overall effectiveness and speed of execution of CGH over TSA, makes it a preferred approach for solving this problem.

7. Summary and Conclusions

This paper addresses a high school timetabling problem that is concerned with assigning teachers to classes and time-slots. In our preliminary research on this problem, we had formulated a comprehensive mixed-integer programming model that directly generates weekly schedules for teachers, but we found that solving even its linear programming relaxation with a competitive commercial package (CPLEX-MIP-version 12.0) was out-of-reach for the six test problems related to a number of high schools in Kuwait. This motivated the development of a two-stage solution approach as described in this paper, which partitions the underlying problem such that classes are assigned to time-slots in Stage I (Model M1), and then teachers are assigned to classes in Stage II (Model M2) based on the fixed time-slots for classes as determined at Stage I.

We next proposed an alternative comprehensive mixed-integer programming model (M3), which composes feasible sets of schedules for teachers in order to assign classes for the sections of the different grade levels. Due to the exponential number of variables in Model M3, a column generation method was designed to solve its linear programming relaxation, based on which, a heuristic was devised to derive a good quality feasible solution. Both the two-stage approach (TSA) and the column generation heuristic (CGH) produced solutions for the six test problems in relatively short CPU times, but the latter approach outperformed the former with respect to CPU time and the number of teachers needed. In particular, CGH generated schedules that required on average about 16.39% fewer teachers than those needed by schedules generated via TSA, and moreover, CGH consumed less than a third of the CPU time as that required by TSA to solve the problems. The robustness of CGH over TSA is further substantiated by noting that it yielded an optimality gap of about 2.48% on average, which is a seven-fold reduction in the optimality gap obtained for TSA. Both TSA and CGH produced solutions that satisfied the school requirements while using a fewer number of teachers than those obtained via a manual approach that is implemented in practice, with average percentage reductions respectively given by 7.06% and 22.32%. Therefore, we intend to incorporate the proposed approaches (particularly, CGH) within a user-friendly software and introduce it to the Ministry of Education in Kuwait for potential implementation.

Table 14. Test problems

Test Problem	$g=1$	$g=2$	$g=3$	$g=4$	$g=5$	**Total**	**NT**
	$\lvert S^g \rvert$ (Number of class-sections of grade g)						
P_1 (Fatima bint Assad High School)	4	2	3	2	2	**13**	**30**
P_2 (Um Alala Ansaria High School)	5	2	2	2	3	**14**	**40**
P_3 (Sabahia High School)	6	3	4	3	3	**19**	**50**
P_4 (Simulated)	7	4	5	4	4	**24**	**55**
P_5 (Simulated)	8	5	6	5	5	**29**	**60**
P_6 (Simulated)	9	6	7	6	6	**34**	**65**

Accommodating additional modeling considerations related to quantifying teachers' preferences for specific classes and time-periods, and related to overall satisfaction levels of teachers along with associated equity issues would disrupt the special column structure of Model M3, which facilitated the design of an efficient column generation heuristic in this paper. Given the relative performances of the two modeling approaches, it is of interest to investigate techniques for incorporating such preferences and equity issues within the column generation framework. One way to accomplish this might be to retain all the columns generated during the process of solving $\overline{\text{M3}}$ via the CGM procedure, and then to solve M3 using only this restricted set of columns in addition to equity-driven side-constraints. We propose this investigation for future research.

Table 15. Basic information for the test problems

Course Index (c)	Subject	10^{th} Grade O_c^g	11^{th} Grade Science O_c^g	11^{th} Grade Liberal Arts O_c^g	12^{th} Grade Science O_c^g	12^{th} Grade Liberal Arts O_c^g
1	Geology		2			
2	Chemistry	2	3		4	
3	Social Studies			1		
4	Sociology	3				
5	Geography			2		2
6	Philosophy					2
7	Psychology					2
8	Economics			2		
9	History			2		2
10	French			5		5
11	Physical Education	2	1	1	1	1
12	Computer	2				
13	English	5	6	7	5	7
14	Arabic	6	5	7	5	8
15	Religion	3	3	3	3	3
16	Biology	2	2		3	
17	Scientific Culture			1		
18	Math	5	6	1	6	
19	Home Economics	2	2	2	2	2
20	Physics	2	4		5	
Total		**34**	**34**	**34**	**34**	**34**

Table 16. Results for solving M1 directly using CPLEX

Test Problem	Rows	Columns	RT $\overline{\text{CTM}}$	RT CTM
P_1	1803	4830	0.48	0.50
P_2	1897	5145	0.45	0.48
P_3	2436	7035	5.25	5.32
P_4	2963	8855	0.62	0.68
P_5	3490	10675	2.64	2.69
P_6	4017	12495	6.46	6.48
Avg	**2767.67**	**8172.5**	**2.65**	**2.69**

Table 17. Results related to implementing TSA and CGH

Test Prob-lem	$\mathrm{ofv}^{\mathrm{TSA}}$	Total RT TSA	$\mathrm{ofv}^{\mathrm{CGM}}_{\overline{\mathrm{M3}}}$	Total RT CGM	$\mathrm{ofv}^{\mathrm{CGH}}_{\mathrm{M3}}$	Total RT CGH	No. of Iterations in CGH
P_1	28	1.58	26	0.39	27	1.06	29
P_2	35	1.94	31	1.25	31	1.51	33
P_3	46	6.94	34	1.33	35	1.54	36
P_4	51	2.87	37	1.43	37	1.57	39
P_5	58	5.25	43	1.55	46	1.62	48
P_6	62	9.22	54	1.59	55	1.68	56
Avg	**46.66**	**4.63**	**37.5**	**1.25**	**38.5**	**1.49**	**40.16**

Table 18. Optimality gaps and percentage improvements

Test Prob-lem	$\tau_1(\mathrm{ofv}^{\mathrm{TSA}}, \mathrm{ofv}^{\mathrm{CGA}}_{\overline{\mathrm{M3}}})$	$\tau_1(\mathrm{ofv}^{\mathrm{CGH}}_{\mathrm{M3}}, \mathrm{ofv}^{\mathrm{CGM}}_{\overline{\mathrm{M3}}})$	$\tau_2(\mathrm{ofv}^{\mathrm{CGH}}_{\mathrm{M3}}, \mathrm{ofv}^{\mathrm{TSA}})$	$\tau_2(\mathrm{ofv}^{\mathrm{TSA}}, \mathrm{NT})$	$\tau_2(\mathrm{ofv}^{\mathrm{CGH}}_{\mathrm{M3}}, \mathrm{NT})$
P_1	7.14	3.70	3.57	6.66	10.00
P_2	11.42	0	11.43	12.50	22.50
P_3	26.08	2.86	23.91	8.00	30.00
P_4	27.45	0	27.45	7.27	32.72
P_5	25.86	6.52	20.69	3.33	23.33
P_6	12.90	1.82	11.29	4.61	15.38
Avg	**18.48**	**2.48**	**16.39**	**7.06**	**22.32**

Acknowledgement: The authors gratefully acknowledge the assistance of Ms. Renju Lekshmi in implementing the developed procedures, and thank an anonymous referee who provided several constructive and insightful suggestions for improving the presentation of this paper.

Table 19. Average daily loads for teachers

Test Problem	Manual Approach	TSA	CGH
P_1	2.94	3.25	3.37
P_2	2.38	2.72	3.07
P_3	2.58	2.87	3.6
P_4	2.86	3.2	4.4
P_5	3.28	3.4	4.2
P_6	3.55	3.7	4.20
Avg	**2.93**	**3.19**	**3.80**

References

[1] Abdullah S, Ahmadi S, Burke EK, Dror M, McCollum B. A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem. J. Oper. Res. Soc. 2007;58(11):1494-1502.

[2] Abramson D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. Manag. Sci. 1991;37(1):98-113.

[3] Abramson D, Dang H. School timetables: a case study in simulated annealing. In: Vidal V (Eds.). Applied simulated annealing, lecture notes in Economics and Mathematics Systems. Berlin, Springer; 1993;(Chapter 5):103–124.

[4] Abramson D, Krishnamoorthy M, Dang H. Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific J. Oper. Res. 1999;16(1):1-22.

[5] Aladaga CH, Hocaoglua G, Basaranb MA. The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. Expert. Syst. Appl. 2009;36(10):12349-12356.

[6] Alvarez-Valdes R, Martin G, Tamarit JM. Hores: a timetabling system for Spanish secondary schools. O. R. Pract. 1995;3(1):137-144.

[7] Alvarez-Valdes R, Crespo E, Tamarit JM. Design and implementation of a course scheduling system using tabu search. Eur. J. Oper. Res. 2002;137(3):512-523.

[8] Alvarez-Valdes R, Parreno F, Tamarit JM. A tabu search algorithm for assigning teachers to courses. TOP: J. Span. Soc. Stat. Oper. Res. 2002;10(2):239-259.

[9] Al-Yakoob, SM, Sherali HD. Mathematical programming models and algorithms for a class-faculty assignment problem. Eur. J. Oper. Res., 2006:173(2):488-507.

[10] Avella P, D'Auria B, Salerno S, Vasil'ev I. A computational study of local search algorithms for Italian high-school timetabling. J. Heur. 2007;13(6):543-556.

[11] Azimi ZN. Hybrid heuristics for examination timetabling problems. Appl. Math. Comput. 2005;163 (2):705-33.

[12] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MW, Vance PH. Branch-and-price: column generation for solving huge integer programs. Oper. Res. 1998;46(3):316-329.

[13] Bazaraa MS, Jarvis JJ, Sherali HD. Linear Programming and Network Flows. 4th ed. Hoboken, New Jersey, John Wiley and Sons; 2010.

[14] Beligiannis GN, Moschopoulos CN, Kaperonis GP, Likothanassia SD. Applying evolutionary computation to the school timetabling problem: the Greek case. Comput. Oper. Res. 2008;35(4):1265-1280.

[15] Beligiannis GN, Moschopoulos CN, Likothanassis SD. A genetic algorithm approach to the school timetabling problem. J. Oper. Res. Soc. 2009;60(1):23-42.

[16] Bello GS, Rangel MC, Boeres S. An approach for the class/teacher timetabling problem using graph coloring. In Burke EK and Gendreau M (Eds). The Practice and Theory of Automated Timetabling VII, August 2008.

[17] Birbas T, Daskalaki S, Housos E. Timetabling for Greek high schools. J. Oper. Res. Soc.1997;48(2):1191-1200.

[18] Birbas T, Daskalaki S, Housos E. School timetabling for quality student and teacher schedules. J. Sched. 2009;12(2):177-197.

[19] Burke EK, Ross P. Practice and theory of automated timetabling. Lecture Notes in Computer Science. Springer, 1996;1153: 283-295.

[20] Burke EK, Petrovic S. Recent research directions in automated timetabling. Eur. J. Oper. Res. 2002;140(2):266-280.

[21] Carrasco MP, Pato MV. A Potts neural network heuristic for the class/teacher timetabling problem. In Applied Optimization Metaheuristics: Computer Decision-Making. Kluwer Academic Publishers; 2004a:173-186.

[22] Carrasco MP, Pato MV. A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. Eur. J. Oper. Res. 2004;153(1):63-79.

[23] Carter MW, Laporte G. Recent developments in practical course timetabling. Practice and Theory of Automated Timetabling II (Second International Conference, PATAT'97, University of Toronto, August 1997, selected papers), Springer Lecture Notes in Computer Science, 1998;1408:3-19.

[24] Colorni A, Dorigo M, Maniezzo V. Metaheuristics for high school timetabling. Comput. Optim. Appl. 1998;9(3): 275-298.

[25] Daskalaki S, Birbas T, Housos E. An integer programming formulation for a case study in university timetabling. Eur. J. Oper. Res. 2004;153(1):117-35.

[26] Daskalaki S, Birbas T. Efficient solutions for a university timetabling problem through integer programming. Eur. J. Oper. Res. 2005;160(1):106-120.

[27] De Causmaecker P, Demeester P, Vanden Berghe G. A decomposed metaheuristic approach for a real-world university timetabling problem. Eur. J. Oper. Res. 2009; 195(1):307-318.

[28] de Haan P, Landman R, Post G, Ruizenaar H. A case study for timetabling in a Dutch secondary school. In Burke EK, Rudova H (Eds). Proceedings of PATAT 2006, Lecture Notes in Computer Science, Springer-Verlag, 2006;3867:267-279.

[29] de Werra D, Asratian AS, Durand S. Complexity of some special types of timetabling problems. J. Sched. 2002;5(2):171-183.

[30] Dorneles AP, de Araújo OCB, Buriol LS. A fix-and-optimize heuristic for the high school timetabling problem. Comput. Oper. Res. 2014:52:29-38.

[31] Eikelder HM, Willemen RJ. Some complexity aspects of secondary school timetabling problems. Computer Science Practice and Theory of Automated Timetabling III, Lecture Notes in Computer Science, 2001;2079:18-27.

[32] Eley E. Ant algorithms for the exam timetabling problem. In: Burke EK, Rudova H. (Eds). Practice and Theory of Automated Timetabling VI: Selected Papers from the 6th International Conference, (LNCS 3867), 2007; 364-382.

[33]] Even S, Itai A, Shamir A. On the complexity of timetable and multicommodity flow problems. SIAM J. Comput. 1976;5:691-703.

[34] Fernandez BC, Santos M. A non-standard genetic algorithm to solve constrained school timetabling problems. In Eurocast, Springer, Berlin, 2003;26-37.

[35] Gislen L, Peterson C, Soderberg B. Teachers and classes with neural networks. Int. J. Neural Syst. 1989;1(2):167-176.

[36] Goltz H, Matzke D. University timetabling using constraint logic programming. In Practical Aspects of Declarative Languages, (LNCS 1551), Springer, 1999;320–334.

[37] Gonzalez-del-Campo R, Saenz-Perez F. Programmed search in a timetabling problem over finite domains. Electronic Notes in Theoretical Computer Science. 2007;177:253–267.

[38] Hartog J. Timetabling on Dutch high schools. PhD thesis, TU Delft, The Netherlands, March 2007.

[39] Jacobsen F, Bortfeldt A, Gehring H. Timetabling at German secondary schools: tabu search versus constraint programming. In Burke EK, Rudova H (Eds.), Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006);439-442.

[40] Junginger W. Timetabling in Germany - a survey. Interfaces. 1986;16(4):66–74.

[41] Kesteren van. The clustering problem in Dutch high schools: changing metrics in search space. Masters Thesis, Leiden University, The Netherlands, 1999.

[42] Kingston JF. Resource assignment in high school timetabling. Ann. Oper. Res. 2010;1-4, D. Online First™, 4 February 2010.

[43] Kwok L, Kong S, Kam Y. Timetabling in Hong Kong secondary schools. Compu. Educa. 1997; 28(3):173-183.

[44] Landman R. Creating good-quality timetables for Dutch high schools. Masters Thesis, University of Twente, The Netherlands, March 2005.

[45] Lawrie NL. An integer linear programming model of a school timetabling problem. Compu. J. 1969;12(4):307-316.

[46] Legierski W. Search strategy for constraint-based class-teacher timetabling. Practice and Theory of Automated Timetabling IV, Lecture Notes in Computer Science. 2003;2740:247-261.

[47] Lewandowski G. Practical implementations and applications of graph coloring. PhD Thesis, University of Wisconsin, Madison, USA, 1994.

[48] Lewis R. A survey of metaheuristic-based techniques for university timetabling problems. OR Spect. 2007;30(1):167–190.

[49] Marte M. Models and algorithms for school timetabling - a constraint programming approach. PhD thesis, Ludwig-Maximilians-Universität, München, Germany, July 2002.

[50] McCollum B. A perspective on bridging the gap between research and practice in university timetabling. In Burke E, Rudova H (Eds). Practice and Theory of Automated Timetabling VI, (LNCS 3867), Springer-Verlag, Berlin, 2007; 3- 23.

[51] McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, Di Gaspero L, Qu R, Burke EK. Setting the research agenda in automated timetabling: the second international timetabling competition. Informs J. Compu. 2009;22(1):120-130.

[52] Merlot LT. Techniques for academic timetabling. PhD Thesis, University of Melbourne, Australia, 2005.

[53] Moura AV, Augusto R, Scaraficci A. GRASP strategy for a more constrained school timetabling problem. Int. Trans. Oper. Res. 2010;I7(22):152-170.

[54] Nemhauser GL, Wolsey LA. Integer and Combinatorial Optimization. 2nd ed. New York, NY; 1999.

[55] Odeniyi OA, Omidiora EO, Olabiyisi SO, Aluko. Development of a modified simulated annealing to school timetabling problem. Int. J. Appl. Inform. Sys.(JAIS). 2015:8(2):16-24.

[56] Papoutsis K, Valouxis C, Housos E. A column generation approach for the timetabling problem of Greek high schools. J. Oper. Res. Soc. 2003;54(3):230-238.

[57] Petrovic S, Burke EK. University timetabling, handbook of scheduling: algorithms, models, and performance analysis, CRC Press, Boca Raton, FL,2004 (Chapter 45).

[58] Post G, Ahmadi S, Daskalaki S, Kingston JH, Kyngas J, Nurmi C, Ruizenaar H, Ranson D. An XML format for benchmarks in high school timetabling. In Proceeding of the 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT, 2008, Montreal, Canada.

[59] Post G, Ahmadi S, Daskalaki S, Kingston J, Kyngas J, Nurmi C, Ranson D. An XML format for benchmarks in high school timetabling. Ann. Oper. Res. 2010;1-13.

[60] Post G, Kingston JH, Ahmadi S, Daskalaki S, Gogos C, Kyngas J, Nurmi C, Musliu N, Pillay N, Santos H, Schaerf A. "XHSTT: an XML archive for high school timetabling problems in different countries. Ann. Oper. Res. 2014:218:295-301.

[61] Qu R, Burke EK, McCollum B, Merlok LT, Lee SY. A survey of search methodologies and automated approaches for examination timetabling. J. Sched. 2009;12(1):55-89.

[62] Raghavjee R, Pillay N. An informed genetic algorithm for the high school timetabling problem. Proceedings of the SAICSIT '10 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, ACA, New York, NY, USA, 2010.

[63] Ramirez ER. Using genetic algorithms to solve high school course timetabling problems. Masters' Thesis, Computer Science Department, San Diego State University, USA, 2010.

[64] Santos HG, Luiz S, Ochi LS, Souza M. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. ACM J. Exp. Algorithm. 2005;10: article no. 2.9.

[65] Santos HG, Uchoa E, Ochi LS, Maculan N. Strong bounds with cut and column generation for class-teacher timetabling. In the Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT, 2008.

[66] Schaerf A. Tabu search techniques for large high school timetabling problems. Technical Report CS-R9611 1996, Computer Science Department of Interactive Systems, Centrum Voor Wiskunder en Informatica (CWI), ISSN 0169-118X.

[67] Schaerf A. Local search techniques for large high-school timetabling problems. IEEE T. Syst. Man. Cy. A. 1999;29(4):368-377.

[68] Schaerf A. A survey of automated timetabling. Artif. Inte. Rev. 1999;13(2):87-127.

[69] Schreuder JAM, Van der Velde JA. Timetables in Dutch high schools. In Operational Research '84 (Brans JP, Eds), Elsevier, Amsterdam, 1984;601-612.

[70] Socha K, Sampels M, Manfrin M. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In Proceedings of EvoCOP 2003, 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization (LNCS 2611), 2003; 334–345.

[71] Sørensen M, Dahms FHW. A two-stage decomposition of high school timetabling applied to cases in Denmark. Comput. Oper. Res. 2014:43:36-49.

[72] Tassopoulos IX, Beligiannis GN. Solving effectively the school timetabling problem using particle swarm optimization. Expert. Syst. Appl. 2012:39:6029-6040.

[73] Tassopoulos IX, Beligiannis GN. A hybrid particle swarm optimization based algorithm for high school timetabling problems. Appl. Soft. Comput. 2012:12: 3472-3489.

[74] Thompson J, Dowsland KA. General cooling schedules for a simulated annealing based timetabling system. In Burke E, Ross P. (Eds). Practice and Theory of Automated Timetabling I, (LNCS 1153), Springer-Verlag, Berlin. 1996;345-363.

[75] Tripathy A. School timetabling - a case in large binary integer linear programming, Manag. Sci. 1984;30(12):1473-1489.

[76] Ueda H, Ouchi D, Takahashi K, Miyahara T. Comparisons of genetic algorithms for timetabling problems. Syst. Comput. Jpn. 2004;35(7):691-701.

[77] Valouxis C, Houso E. Constraint programming approach for school timetabling. Comput. Oper. Res. 2003;30(10):1555-572.

[78] Valouxis C, Gogos C, Alefragis P, Housos E. Decomposing the high school timetable problem. Practice and Theory of Automated Timetabling (PATAT), Son, Norway. August 2012: 209-221.

[79] Willemen RJ. School timetable construction: algorithms and complexity. PhD Thesis, Technische Universiteit Eindhoven, The Netherlands, May 2002.

[80] Wood J, Whitaker D. Student centered school timetabling, J. Oper. Res. Soc. 1998;49:1146-1152.

[81] Zhang D, Liu Y, M'Hallah R, Leung S. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. Eur. J. Oper. Res. 2010;203(3):550-558.

Paper Highlights

- The problem studied is related to high schools in Kuwait.
- A comprehensive mixed-integer programming model is initially developed; however, it was found to be untenable for practical-sized realistic test instances.
- A column generation heuristic (CGH) is proposed.
- Computational results substantiate the effectiveness of the adopted CGH .