Research Article

Mixed-Integer Linear Programming Models for Teaching Assistant Assignment and Extensions

Xiaobo Qu,[1] Wen Yi,[2] Tingsong Wang,[3] Shuaian Wang,[4] Lin Xiao,[5] and Zhiyuan Liu[6]

[1]School of Civil and Environmental Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia
[2]Department of Building and Real Estate, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[3]School of Economics and Management, Wuhan University, Wuhan 430072, China
[4]Department of Logistics & Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[5]National Research Council of the National Research Academies of Science, Engineering, and Medicine, Washington, DC 20001, USA
[6]Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies,
 School of Transportation, Southeast University, Jiangsu, China

Correspondence should be addressed to Tingsong Wang; emswangts@whu.edu.cn, Shuaian Wang; wangshuaian@gmail.com,
and Zhiyuan Liu; leakeliu@163.com

In this paper, we develop mixed-integer linear programming models for assigning the most appropriate teaching assistants to the tutorials in a department. The objective is to maximize the number of tutorials that are taught by the most suitable teaching assistants, accounting for the fact that different teaching assistants have different capabilities and each teaching assistant's teaching load cannot exceed a maximum value. Moreover, with optimization models, the teaching load allocation, a time-consuming process, does not need to be carried out in a manual manner. We have further presented a number of extensions that capture more practical considerations. Extensive numerical experiments show that the optimization models can be solved by an off-the-shelf solver and used by departments in universities.

1. Introduction

Teaching assistants (TAs) are essential in many departments in universities because of limited numbers of lecturers. TAs are mainly postgraduate students, though in some universities TAs can also be undergraduate students. TAs are usually in charge of teaching tutorials in small classes. TA allocation, that is, which TA teaches which tutorial, is an essential task that is carried out in a department every semester. It has been well recognized that appropriate teaching staff allocation contributes to the equality of teaching [1, 2]. The teaching assistants are usually reallocated every semester. This is due to the following two reasons: (1) changes of staff (some TAs are no longer available because they have graduated or have other commitments and new TAs are available) and (2) changes of programs (some tutorials are cancelled and some new tutorials are developed).

There are many factors to consider in TA assignment. First, all tutorials must be taught in order to satisfy the needs of the teaching programs. Second, a TA should not be allocated a higher teaching load other than what is required by her/his role because of her/his limited available time. Third, TAs should teach tutorials that they are proficient with; otherwise it will cost them a lot of preparation time and the teaching outcomes will not be ideal. Fourth, we should consider whether two tutorials are delivered at the same time, and if so, they cannot be taught by the same TA. Fifth, it is easy to see that if a TA teaches several tutorials of the same lecture, in other words, repeating the tutorials, then it is easier for the TA to save time for preparation. Sixth, it might be difficult for a lecturer to manage if there are too many TAs for the lecturer and hence it may be desirable to control the number of TAs for a lecturer.

Our personal experiences show that almost all departments allocate teaching assistants manually in a trial-and-error manner. As we all know, this allocation is very time-consuming process and may lead to unfavorable results which may jeopardize the teaching quality. In this regard, it is imperative and of vital significance to develop a systematic tool in order to better allocate the teaching assistants in a more efficient manner.

1.1. Literature Review. It is universally acknowledged that improving the teaching quality is the number one concern in all teaching activities. Academics have developed a number of new educational theories and pedagogical techniques to meet the teaching requirements in view of the changing society (e.g., [3, 4]). Some studies are related to the timetabling of educational organizations. Valouxis and Housos [5] applied a constraint programming approach to solve a high school timetabling problem considering various practical constraints. Beligiannis et al. [6] proposed an adaptive algorithm to address the timetabling problem of an educational institute. Pillay and Banzhaf [7] examined the use of genetic algorithms to address an examination timetabling problem. A category of research that is more closely related to ours is optimization models proposed to better allocate teaching load for full time academic teaching staff in a teaching unit. For example, Breslaw [8] was among the first to develop a linear programming model to address a teaching staff assignment problem. His objective was to maximize the preference of the faculty members. Schniederjans and Kim [9] adopted a different mathematical approach—a goal programming model—to optimize the overall benefit to the teaching department. Badri [10] developed a more complex model that consists of a two-stage multiobjective scheduling approach. The objectives are twofold: first, the preference for tutorials by faculty members is incorporated; second, the time slots of the tutorials that are preferred by faculty members are also considered. Qu et al. [11] developed a very simple model for assignment lecturers to classes and tested the model with data from a university in Australia. As discussed in Badri [10], the abovementioned mathematical approaches could improve teaching quality by maximizing the preference of the faculty members. However, two key issues are neglected in the abovementioned studies. First, as the focus of teaching activities is to improve or guarantee the teaching quality, the teaching quality should be considered as the objective to maximize, rather than a constraint in the optimization model. In other words, teaching quality is much more important than faculty preference as the former is the number one concern in all teaching activities. Second, the abovementioned studies are mainly focused on teaching load allocation for full time academic staff, while the arrangement for teaching assistants is mostly treated in a very simple way, which may lead to unreliable results.

In most teaching units, teaching assistants are Ph.D. or senior students and they are working as teaching assistants on a part time basis. As a result, it is not appropriate to assign heavy teaching loads to them, due to the fact that they need to focus on their own studies/research. In this regard, the workload equity, namely, the workload of a teaching assistant should be in line with his or her availability, is also very important. Burgess [12] did a comparison study regarding different methods for allocating teaching tasks to faculty members. Vardi [13] carried out a similar work with the objective of analyzing the impacts of workload allocation on the satisfaction of working life of faculty members. Bentley and Kyvik [14] compared the teaching workload allocation statuses in a number of countries. There are a number of works related to the relations between research and teaching for academic staff (e.g., [15–18]). It should be stressed that all of the studies realized that equity has a vital impact on academics' satisfaction. Unfortunately, two issues are not properly addressed. First, the equity issue is largely neglected in the above teaching load allocation optimization models. Second, to the best of our knowledge, there is no research which focuses on the equity issue for teaching assistants.

In this study, we aim to develop teaching assistant assignment models in order to optimize the teaching quality by taking into account various constraints such as equality and availability of teaching assistants. The model can improve the "fitness" between teaching assistants and tutorials without jeopardizing the equity issue of teaching assistants. Further, this proposed model will reduce the workload for school heads, managers, or program coordinators by automatically selecting the optimal solutions.

1.2. Objectives and Contributions. The objective of this research is to develop mixed-integer programming models [19–23] that are able to generate the optimal allocation plan of teaching assistants. The contribution of the paper is threefold: First, we develop a new approach that is able to enhance the overall teaching performance or quality of a teaching unit essentially at no cost. This is due to the fact that our model enables more teaching assistants to teach tutorials that they are competent at. As a result, the teaching assistants' satisfaction can be improved and teaching quality is accordingly improved. The basic assumption is very reasonable: the teaching quality is positively related to the "fitness" between teaching assistants and tutorials that they teach. However, it is very challenging if we want to reach the optimal solution manually: one cannot guarantee that the optimal decision is made. By contrast, mathematical programming approaches could obtain optimal decisions for department heads to assign the teaching tasks.

Second, this allocation is usually done by senior academics (e.g., head of school, deputy head of school, department chair, school manager, and program coordinator). They are naturally very busy with their teaching, research, supervision, and other commitments. Our model can save the valuable time for these academics.

Third, the model considers the maximum teaching load of each teaching assistant. Teaching assistants are usually on a part-time basis. Imposing too much teaching to them will jeopardize their performance in their full time commitment (i.e., study/research). If this allocation is carried out manually by senior academics, it is likely that the maximum teaching is violated due to the fact that manual checking is very time-consuming. This deficiency can be completely overcome using our models.

Fourth, we also consider whether two tutorials are delivered at the same time, and if so, they cannot be taught by the same TA.

Fifth, it is easy to see that if a TA teaches several tutorials of the same lecture, in other words, repeating the tutorials, then the TA can save time for preparation. Our model thus aims to increase the number of tutorial repetitions.

Sixth, it might be difficult for a lecturer to manage if there are too many TAs for the lecturer and hence it may be desirable to control the number of TAs for a lecturer. This factor is also formulated in our model.

The remainder of this paper is organized as follows. Section 2 describes considerations in teaching assistant allocation. Section 3 builds a basic integer linear optimization model. Section 4 reports a number of practical extensions to the basic model. The results of numerical experiment are reported in Section 5. Conclusions are presented in Section 6.

2. Problem Description

This study examines the allocation of teaching assistants at a department or school. In a department, many tutorials for undergraduates and postgraduates are taught by teaching assistants. A teaching assistant has limited time that can be spent on teaching. Different teaching assistants have different maximum numbers of teaching hours because (i) they have different availability and (ii) they have different willingness to teaching. In reality, most teaching assistants are Ph.D. students in a teaching unit. Some students have research scholarships which limit the number of hours that they can spend in teaching, while the others have no scholarships so that the number of hours that they can spend in teaching is only limited by the visa requirement (for international Ph.D. students). As different teaching assistants have different expertise, we classify the relation between a tutorial and a teaching assistant into three types. (i) In the first type, the teaching assistant is unable to teach that tutorial. For example, in a department of mathematics, a teaching assistant with expertise in statistics cannot teach the tutorial for "Group Theory." (ii) In the second type, the teaching assistant can teach a tutorial, but it is not the most suitable person. For example, a teaching assistant with expertise in the structural engineering can teach the elective tutorial "Introduction to Geotechnical Engineering" for civil engineering students, but she/he is not the most suitable person for teaching it, as a teaching assistant with the expertise in geotechnical engineering is more suitable. (iii) In the third type, the teaching assistant is one of the most suitable persons to teach a tutorial. For example, a teaching assistant with expertise in transport engineering should teach the tutorial "Traffic Flow Theory" or "Traffic Engineering Fundamentals" if possible. Another example is that if the teaching assistant taught this tutorial and was highly appraised last year, she or he should also teach it this year if possible.

We can now see that the purpose of optimizing the allocation of teaching assistants includes constraints that all tutorials are taught and that the available teaching time of teaching assistants is satisfied and an objective of maximizing

the total number of tutorials delivered by the most appropriate teaching assistant. At the same time, some other practical factors must also be taken into account. For instance, we should consider whether two tutorials are delivered at the same time, and if so, they cannot be taught by the same TA; it is easy to see that if a TA teaches several tutorials of the same lecture, in other words, repeating the tutorials, then the TA can save time for preparation; it might be difficult for a lecturer to manage if there are too many TAs for the lecturer and hence it may be desirable to control the number of TAs for a lecturer.

3. Basic Optimization Model

To address the teaching assistant (TA) assignment problem, we develop an integer optimization model similar to Qu et al. [11]. However, we will present rich extensions to the basic optimization in the next section. The notations used in the basic optimization model are listed below.

Sets

I: set of tutorials

I_j: set of tutorials that TA $j \in J$ may teach; $I_j = \{i \in I \mid j \in J_i\}$

J: set of TAs

J_i: set of TAs who can deliver tutorial $i \in I$; $J_i = J_i^1 \cup J_i^2$

J_i^0: set of TAs who cannot deliver tutorial $i \in I$

J_i^1: set of TAs who can teach but are not the most suitable for delivering tutorial $i \in I$

J_i^2: set of TAs who are the most suitable for delivering tutorial $i \in I$

Indices

i: an index $i \in I$ that refers to a particular tutorial

j: an index $j \in J$ that refers to a particular TA

Parameters

$n_j^{\min}$: the minimum number of tutorials that must be delivered by TA $j \in J$: it can be set at 0 if there is no such requirement

$n_j^{\max}$: the maximum number of tutorials that can be delivered by TA $j \in J$

t_i: the number of contact hours required for tutorial $i \in I$: in practice, t_i is usually 1, 2, or 3 hours

$t_j^{\min}$: the minimum number of available hours per week for TA $j \in J$

$t_j^{\max}$: the maximum number of available hours per week for TA $j \in J$

Decision Variables

  x_{ij}: a binary decision variable which equals 1 if TA $j \in J_i^1 \cup J_i^2$ delivers tutorial $i \in I$ and 0 otherwise

The above sets and parameters well capture the real decision process. For instance, the sets J_i^0, J_i^1, and J_i^2 can be determined by a survey of the TAs. The survey can have one question for each tutorial, in which each TA must choose one from the following three answers: (a) I like to teach this tutorial very much; (b) This tutorial is not my favorite, but I can teach it if required; (c) I cannot teach this tutorial. Sets J_i^0, J_i^1, and J_i^2 are mutually exclusive and collectively exhaustive; $J = J_i^0 \cup J_i^1 \cup J_i^2$. The teaching load allocation problem for TAs can be formulated as an integer linear programming model:

$$[\text{M0}]: \max \quad \sum_{i \in I} \sum_{j \in J_i^2} x_{ij} \tag{1}$$

$$\text{subject to:} \quad \sum_{j \in J_i} x_{ij} = 1, \quad i \in I \tag{2}$$

$$t_j^{\min} \le \sum_{i \in I_j} t_i x_{ij} \le t_j^{\max}, \quad j \in J \tag{3}$$

$$n_j^{\min} \le \sum_{i \in I_j} x_{ij} \le n_j^{\max}, \quad j \in J \tag{4}$$

$$x_{ij} \in \{0, 1\}, \quad i \in I, \ j \in J_i. \tag{5}$$

The objective function (1) maximizes the total number of tutorials that are taught by the most suitable TAs. Equation (2) imposes that all tutorials are taught. Equations (3) and (4) take into account the available time of each TA. Finally, (5) defines the domains of the decision variables.

Proposition 1. *The above integer linear programming model cannot be solved as a linear program.*

Proof. We construct an example to show that relaxing the integrality constraints may not lead to a correct solution. Suppose that there are two TAs and one tutorial; the tutorial requires two hours; the first TA is the most suitable but has only one available hour; the second TA, with two available hours, can teach the tutorial but is not the most suitable. It is evident that the only feasible solution to the problem is to let the second TA teach the tutorial. However, solving a linear program will require the first TA to teach for one hour and the second TA to teach for one hour. □

The size of the above integer linear programming model is usually not large for several reasons. First, many courses do not have tutorials. Second, even in one department, there are several areas of specialization and TAs in one area of specialization can usually only teach tutorials for courses in this area. For example, in a mathematics department, there are at least three areas: pure mathematics, applied mathematics, and statistics. Even in the area of applied mathematics, there is, for instance, financial mathematics, chemical mathematics, medical mathematics, and operations

research. TAs in one subarea may not be able to teach tutorials in other subareas. As a result, the overall model for a department can be decomposed for each area or subarea. Our numerical experiments in Section 5 show that the model can be solved by off-the-shelf solvers such as CPLEX.

4. Extensions to the Basic Integer Programming Model

The above basic integer linear programming model captures the most essential features of the TA assignment problem. However, there are many other realistic factors that must be addressed before the above model can be put into use directly or put into use after minimal manual adjustment. We elaborate these factors and present approaches on how to model these factors.

4.1. Time Conflict of Two Tutorials. Usually the schedules of the tutorials are a priori determined. For instance, if the course "Basics of Calculus" has 3 tutorials, then the 3 tutorials are usually scattered uniformly in a week. As a result, it is likely that one tutorial (say, tutorial i is from 4:00 pm to 5:00 pm on Monday) has time overlap with another tutorial (say, tutorial k is from 3:00 pm to 5:00 pm on Monday). In this case, the two tutorials cannot be taught by the same TA.

To address this difficulty, we define a new binary parameter θ_{ik}, which equals 1 if and only if tutorial i and tutorial k have no time conflict, meaning that they could be taught by the same TA. We add the following constraints to the basic model to formulate the time constraints:

$$[\text{M1}]: x_{ij} + x_{kj} \le 1 + \theta_{ik}, \tag{6}$$
$$j \in J_i \cap J_k, \ i \in I, \ k \in I, \ i < k.$$

Equation (6) means that, for each TA $j \in J$, if two tutorials i and k have time conflict, that is, $\theta_{ik} = 0$, then the TA can teach at most one of them because $x_{ij} + x_{kj} \le 1 + \theta_{ik} = 1$.

Proposition 2. *In constraints (6) if there are a set of tutorials denoted by Ω such that $\theta_{ik} = 0$, $\forall i \in \Omega$, $k \in \Omega$, and $i < k$, then we can strengthen the constraints by combining some of them to one constraint:*

$$\sum_{i \in \Omega} x_{ij} \le 1, \quad j \in J. \tag{7}$$

Proof. The results hold trivially based on observation. □

4.2. Repetition of Tutorials. Some tutorials may belong to the same course, for example, "Basics of Calculus," and in these tutorials TAs deliver the same contents to students. The tutorials are repeated for two reasons. First, the tutorial class can have a small size of students to facilitate interactions among the students and between students and the TA. Second, repeating the tutorials could provide students with the flexibility of choosing the time slot, choosing the TA, and possibly attending the tutorial twice to improve the learning outcomes. It is convenient for a TA to teach several tutorials

that belong to the same course so that she could save time for preparation.

To incorporate the advantage of assigning several tutorials that belong to the same course to one TA, we first define a weight α to represent the benefit of assigning one more same tutorial to a TA. The weight α should be understood to be the value of the saved preparation time by TA. We further define a set H to be the set of courses with at least two tutorials. The set of tutorials that belong to course $h \in H$ is defined to be $\bar{I}_h$. We define new decision variables y_{hj} to be the number of tutorials for course $h \in H$ that are assigned to TA $j \in J$. The objective function (1) should be revised to

$$[M2]: \ \max \sum_{i \in I} \sum_{j \in J_i^2} x_{ij} + \alpha \sum_{h \in H} \sum_{j \in J} \max\left(y_{hj} - 1, 0 \right) \tag{8}$$

and the following constraints should be added:

$$y_{hj} = \sum_{i \in \bar{I}_h} x_{ij}, \quad h \in H, \ j \in J. \tag{9}$$

Equation (9) counts y_{hj} and the new objective function (8) has the extra term $\alpha \sum_{h \in H} \sum_{j \in J} \max(y_{hj} - 1, 0)$.

It should be noted that the extra term $\alpha \sum_{h \in H} \sum_{j \in J} \max(y_{hj} - 1, 0)$ has the "max" operator that makes the model nonlinear. We propose the following method to linearize the objective function (8). To make the model clear, we explicitly define the new sets, parameters, and decision variables.

Newly Defined Sets

H: set of courses with at least two tutorials

I_h: set of tutorials that belong to course $h \in H$

Newly Defined Parameters

α: benefit of assigning one more same tutorial to a TA

Newly Defined Decision Variables

y_{hj}: number of tutorials for course $h \in H$ that are assigned to TA $j \in J$

z_{hj}: an intermediate binary variable

u_{hj}: an intermediate continuous variable for linearization

The new model is

$$\left[M2' \right]: \ \max \ \sum_{i \in I} \sum_{j \in J_i^2} x_{ij} + \alpha \sum_{h \in H} \sum_{j \in J} u_{hj} \tag{10}$$

$$\text{subject to:} \ \ z_{hj} \leq y_{hj}, \quad h \in H, \ j \in J \tag{11}$$

$$u_{hj} \leq y_{hj} - z_{hj}, \quad h \in H, \ j \in J \tag{12}$$

$$u_{hj} \leq M z_{hj}, \quad h \in H, \ j \in J \tag{13}$$

$$z_{hj} \in \{0, 1\}, \quad h \in H, \ j \in J \tag{14}$$

and constraints (2) to (5) and (9). Note that M in constraints (13) is a large positive number.

Proposition 3. *In constraints (13) the value of M could be set to $|\bar{I}_h|$.*

Proof. The upper bound of u_{hj} is $|\bar{I}_h|$. Therefore, it is sufficient to set M to $|\bar{I}_h|$ in constraints (13). $\square$

4.3. Controlling the Number of TAs for One Course. Some lecturers may want to control the number of TAs for his course. For instance, no lecturer wants to have 12 TAs to teach his tutorials. To reflect this requirement, we define parameter b_h as the maximum number of TAs that are assigned to the tutorials for course $h \in H$. We further define intermediate binary decision variables π_{hj} which equal 1 if and only if TA $j \in J$ teaches at least one tutorial for course $h \in H$. We then could add the following constraints:

$$[M3]: \ M\pi_{hj} \geq y_{hj}, \quad h \in H, \ j \in J \tag{15}$$

$$\sum_{j \in J} \pi_{hj} \leq b_h, \quad h \in H \tag{16}$$

$$\pi_{hj} \in \{0, 1\}, \quad h \in H, \ j \in J, \tag{17}$$

where in constraints (15) the value of M could be set to $|\bar{I}_h|$ because the upper bound of y_{hj} is $|\bar{I}_h|$.

Note that constraints (16) could be changed to the following ones:

$$\sum_{j \in J} \pi_{hj} \leq b_h, \quad h \in H, \ b_h \leq \left| \bar{I}_h \right| - 1 \tag{18}$$

because if $b_h = |\bar{I}_h|$, then constraints (18) are always satisfied.

4.4. Controlling the Number of Days a TA Works. A TA may like to teach five tutorials, but she may not like to teach one tutorial every day. To reflect this requirement, we define parameter κ_j as the maximum number of days TA $j \in J$ would like to teach. Evidently, $\kappa_j = 1, 2, 3, 4, 5$. We further define set Θ_m to be the set of tutorials taught on day $m = 1, 2, 3, 4, 5$, where 1 means Monday, 2 means Tuesday, and so on. We define intermediate binary decision variable λ_{mj} that equals 1 if and only if TA $j \in J$ needs to teach on day $m = 1, 2, 3, 4, 5$. The following constraints could capture the requirement regarding the number of days a TA works:

$$[M4]: \ \lambda_{mj} \geq x_{ij}, \quad j \in J, \ m = 1, 2, 3, 4, 5, \ i \in \Theta_m$$

$$\lambda_{mj} \in \{0, 1\}, \quad m = 1, 2, 3, 4, 5, \ j \in J \tag{19}$$

$$\sum_{m=1}^{5} \lambda_{mj} \leq \kappa_j, \quad j \in J.$$

5. Numerical Experiments

We carry out extensive numerical experiments to demonstrate the computational efficiency. The experiments are implemented on a PC with 3.30 GHz of Intel Core i5 CPU and 4 GB of RAM. The algorithm is coded in Matlab 2011b, calling

TABLE 1: Computation time (s) of the models.

Number of tutorials	Number of TAs	[M0]	[M1]	[M2]	[M3]	[M4]
20	10	0.0127	0.0119	0.0139	0.0130	0.0140
	15	0.0139	0.0146	0.0291	0.0158	0.0155
	20	0.0156	0.0148	0.0300	0.0169	0.0176
25	15	0.0163	0.0152	0.0276	0.0189	0.0186
	20	0.0167	0.0164	0.03675	0.0195	0.0200
	25	0.0188	0.0181	0.0433	0.0221	0.0225
30	20	0.0192	0.0190	0.0448	0.0233	0.0239
	25	0.0194	0.0196	0.0558	0.0245	0.0242
	30	0.0210	0.0215	0.0669	0.0272	0.0281

CPLEX12.3 to solve mixed-integer linear programming models.

Different combinations of the number of tutorials and number of TAs (I, J) are considered: $(20, 10)$, $(20, 15)$, $(20, 20)$, $(25, 15)$, $(25, 20)$, $(25, 25)$, $(30, 20)$, $(30, 25)$, and $(30, 30)$, as shown in Table 1. All of the five models [M0], [M1], [M2], [M3], and [M4] are evaluated for each combination. Ten random instances are generated for each model in each combination, and therefore we have a total of $10 \times 5 \times 9 = 450$ instances. The instances are generated as follows. In [M0], for each tutorial-TA combination, there is 1/3 chance that the TA cannot deliver the tutorial, 1/3 chance that the TA can but is not the most suitable, and 1/3 chance that the TA is the most suitable for delivering tutorial; the minimum number of tutorials that must be delivered by a TA is 0; the maximum number of tutorials that can be delivered by a TA is an integer uniformly drawn between 1 and 3; the minimum number of available hours per week for a TA is 0; the maximum number of available hours per week for TA is an integer uniformly drawn between 1 and 5; the number of contact hours required for a tutorial is an integer uniformly drawn between 1 and 2. In [M1], five pairs of tutorials are randomly chosen to have time conflict. In [M2], the number of courses is equal to half the number of tutorials and each tutorial is randomly assigned to a course; the weight α is set at 0.33. In [M3], the maximum number of TAs for a course is an integer uniformly drawn between 1 and 3. In [M4], the maximum number of days a TA works is an integer uniformly drawn between 1 and 5. We report the average CPU time required to solve one instance for each model over the ten random instances in Table 1. It can be seen that all of the models can efficiently be solved. This demonstrates the practical relevance of the proposed models.

6. Conclusions

In this paper, an integer programming model is developed to maximize the teaching quality by assigning most appropriate teaching assistants to their tutorials. The model is very useful for teaching units as (1) it can improve teaching quality by allocating suitable teaching assistants to teach the tutorials; (2) the teaching allocation does not need to be manually allocated, which is a time-consuming process; and (3) the maximum teaching load of each staff will not be violated as it is modeled as hard constraints. Some extensions, which are formulated as mixed-integer linear programming models, are further addressed to consider more practical factors. This includes the following ones: (1) we consider whether two tutorials are delivered at the same time, and if so, they cannot be taught by the same TA; (2) we increase the number of repetitions of tutorials so that it is easier for the TAs to save time for preparation; (3) a lecturer can control the number of TAs for her/his class. Numerical experiments show that these models can efficiently be solved by off-the-shelf solvers, demonstrating the practical relevance of the proposed models.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

Acknowledgments

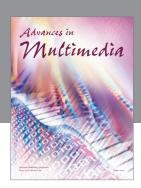## References

[1] R. H. McClure and C. E. Wells, "Modeling multiple criteria in the faculty assignment problem," *Socio-Economic Planning Sciences*, vol. 21, no. 6, pp. 389–394, 1987.

[2] F. Y. Partovi and B. Arinze, "A knowledge based approach to the faculty-course assignment problem," *Socio-Economic Planning Sciences*, vol. 29, no. 3, pp. 245–256, 1995.

[3] G. Hu and J. Lei, "English-medium instruction in Chinese higher education: a case study," *Higher Education*, vol. 67, no. 5, pp. 551–567, 2014.

[4] A. Oleson and M. T. Hora, "Teaching the way they were taught? Revisiting the sources of teaching knowledge and the role of prior experience in shaping faculty teaching practices," *Higher Education*, vol. 68, no. 1, pp. 29–45, 2014.

[5] C. Valouxis and E. Housos, "Constraint programming approach for school timetabling," *Computers & Operations Research*, vol. 30, no. 10, pp. 1555–1572, 2003.

[6] G. N. Beligiannis, C. Moschopoulos, and S. D. Likothanassis, "A genetic algorithm approach to school timetabling," *Journal of the Operational Research Society*, vol. 60, no. 1, pp. 23–42, 2009.
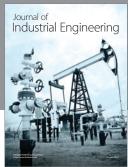
[7] N. Pillay and W. Banzhaf, "An informed genetic algorithm for the examination timetabling problem," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 457–467, 2010.
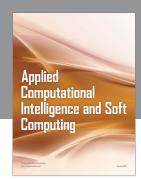
[8] J. A. Breslaw, "A linear programming solution to the faculty assignment problem," *Socio-Economic Planning Sciences*, vol. 10, no. 6, pp. 227–230, 1976.

[9] M. J. Schniederjans and G. C. Kim, "A goal programming model to optimize departmental preference in course assignments," *Computers & Operations Research*, vol. 14, no. 2, pp. 87–96, 1987.

[10] M. A. Badri, "A two-stage multiobjective scheduling model for [faculty-course-time] assignments," *European Journal of Operational Research*, vol. 94, no. 1, pp. 16–28, 1996.

[11] W. Qu, Y. Zhao, M. Wang, and B. Liu, "Research on teaching gamification of software engineering," in *Proceedings of the 9th International Conference on Computer Science & Education (ICCSE '14)*, Vancouver, Canada, August 2014.

[12] T. F. Burgess, "Planning the academic's workload: different approaches to allocating work to university academics," *Higher Education*, vol. 32, no. 1, pp. 63–75, 1996.

[13] I. Vardi, "The impacts of different types of workload allocation models on academic satisfaction and working life," *Higher Education*, vol. 57, no. 4, pp. 499–508, 2009.

[14] P. J. Bentley and S. Kyvik, "Academic work from a comparative perspective: a survey of faculty working time across 13 countries," *Higher Education*, vol. 63, no. 4, pp. 529–547, 2012.

[15] C. Halse, E. Deane, J. Hobson, and G. Jones, "The research—teaching nexus: what do national teaching awards tell us?" *Studies in Higher Education*, vol. 32, no. 6, pp. 727–746, 2007.

[16] J. C. Shin, "Teaching and research nexuses across faculty career stage, ability and affiliated discipline in a South Korean research university," *Studies in Higher Education*, vol. 36, no. 4, pp. 485–503, 2011.

[17] S. Hornibrook, "Policy implementation and academic workload planning in the managerial university: understanding unintended consequences," *Journal of Higher Education Policy and Management*, vol. 34, no. 1, pp. 29–38, 2012.

[18] J. D. J. Kenny and A. E. Fluck, "The effectiveness of academic workload models in an institution: a staff perspective," *Journal of Higher Education Policy and Management*, vol. 36, no. 6, pp. 585–602, 2014.

[19] F. Li, Z. Gao, K. Li, and D. Z. W. Wang, "Train routing model and algorithm combined with train scheduling," *Journal of Transportation Engineering*, vol. 139, no. 1, pp. 81–91, 2013.

[20] Z. Liu, S. Wang, W. Chen, and Y. Zheng, "Willingness to board: a novel concept for modeling queuing up passengers," *Transportation Research Part B: Methodological*, vol. 90, pp. 70–82, 2016.

[21] D. Z. W. Wang, H. Liu, and W. Y. Szeto, "A novel discrete network design problem formulation and its global optimization solution algorithm," *Transportation Research E*, vol. 79, pp. 213–230, 2015.

[22] H. Liu and D. Z. W. Wang, "Global optimization method for network design problem with stochastic user equilibrium," *Transportation Research B*, vol. 72, pp. 20–39, 2015.

[23] B. Du and D. Z. W. Wang, "Continuum modeling of park-and-ride services considering travel time reliability and heterogeneous commuters—a linear complementarity system approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 58–81, 2014.

Advances in
Multimedia

**The Scientific
World Journal**

International Journal of
**Distributed
Sensor Networks**

Journal of
Industrial Engineering

Applied
**Computational
Intelligence and Soft
Computing**

Advances in
**Fuzzy
Systems**

**Modelling &
Simulation
in Engineering**

Journal of
**Computer Networks
and Communications**

Advances in
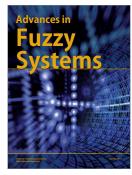**Artificial
Intelligence**

![Hindawi logo]

Hindawi

Submit your manuscripts at
https://www.hindawi.com

Advances in
Computer Engineering

International Journal of
**Computer Games
Technology**

International Journal of
Biomedical Imaging

Advances in
**Artificial
Neural Systems**

Advances in
Software Engineering

Journal of
Robotics

Advances in
**Human-Computer
Interaction**

**Computational
Intelligence and
Neuroscience**

International Journal of
**Reconfigurable
Computing**

Journal of
**Electrical and Computer
Engineering**