CREATING WEEKLY TIMETABLES TO MAXIMIZE EMPLOYEE PREFERENCES

CALEB Z. WHITE, YOUNGBAE LEE, YOONSOO KIM, REKHA THOMAS, AND PATRICK PERKINS

Abstract. We develop a method to generate optimal weekly timetables for tutors at the University of Washington Math Study Center using stated and inferred preferences. These timetables were previously generated using heuristics. We show that it is possible to create timetables using a model based on Integer Programming that are superior to the previous schedules. We compare two variants of our model. The first involves the progressive relaxation of the least important constraints and the second incorporates a penalty for constraint relaxation directly into the objective function. We find that the second approach generates better schedules. We find that using our model instead of the old system creates improvements in the value of the objective function of up to 7.2%. Less prosaically, our model creates schedules that significantly increase the satisfaction and happiness of the individual tutors.

1. Problem Statement

The Math Study Center (MSC) at the University of Washington is a tutorial facility for undergraduate students taking mathematics classes. The tutors are graduate and undergraduate students at the University. Each quarter, the MSC needs to create timetables (work schedules) for between 8 and 20 tutors. The MSC is typically open 56 hours a week, and certain times, like Monday afternoons, have historically been busy and require more tutors. Thus, for each time slot (1 hour) the MSC determines a number between 1 and 5 representing the number of tutors they need. They also guarantee each tutor a certain number of hours of employment per week. Since the tutors are all students, they are not available for work in every time slot due to conflicts with classes and other responsibilities. The tutors are allowed to rank each time slot from 1 to 3, with 3 meaning they strongly wish to be assigned that time slot and 1 meaning they would prefer not to be assigned that time slot. If the tutor has a schedule conflict and absolutely cannot work at a particular time slot, then they may place a zero in the time slots to indicate they are not available. A form on a web page is used to gather this information.

In this paper, we model and solve the problem of creating timetables for tutors at the MSC. For a survey of automated timetabling see [6]. For a general discussion of current research directions see [1]. Our formulation takes the rankings the students individually assigned various hours and creates a work schedule that maximizes the total ranking of all assigned time slots. We have chosen schedules from the Summer 2003, Winter 2003, and Autumn 2002 quarters as representative instances on which to test our model and compare it to the previous solutions. These schedules were previously created using a system of heuristics that first assigns time slots in a greedy fashion until a feasible schedule is created. It then attempts to swap time slots to obtain a higher quality schedule. Finally, it may be manually adjusted to correct any part of the schedule that was particularly poor.

The simplest approach to this problem is to model it as a *transportation problem*. The limit on the number of hours each tutor may work each week is the supply constraint and the number of tutors required at each time slot is the demand constraint. If the coefficients in the constraints are all integer and supply equals demand, then the problem can be solved as a linear programming problem and be guaranteed an integer solution [2].

Date: April 9, 2004.

Key words and phrases. Timetabling, Crew Rostering, Integer Programming.

The first and second authors are undergraduates and were partially supported by NSF VIGRE Grant DMS-9810726.

The fourth author is partially supported by NSF Grant DMS-0100141.

This project began as an optional "industrial project" in the Spring of 2003 in Math 409, an undergraduate course in Discrete Optimization at the University of Washington, taught by Rekha Thomas. The problem was provided by Patrick Perkins, the director of the MSC.

Unfortunately, tutors have additional preferences not captured by the individual ranking they gave each hour. The problem becomes considerably more complex if we attempt to optimize schedules based on these additional preferences. Specifically, we develop methods to incorporate the following inferred preferences into the problem. Tutors prefer not to be assigned more than 5 hours in one day, or more than 3 hours in a row on weekdays. They prefer not to enter and leave the MSC more than twice a day. They prefer not to tutor a single hour at a time. Finally, they prefer to have as small a spread as possible between the first hour they are at the MSC and the last hour when they leave the MSC. It is these additional constraints that make this a difficult problem. Without these additional constraints, the value of any one time slot to an individual tutor is independent of the other time slots the tutor is assigned. With the additional constraints, whether a tutor should be assigned a particular time slot becomes dependent on the other hours they may or may not have been assigned that day. This interdependency makes the problem much more complicated.

This paper is organized as follows. The methods we use are described in Section 2. Section 3 details the problem formulation. In Section 4, the implementation is discussed. Finally, our results and conclusions are explained in Section 5.

2. Methods

There are many possible approaches to solving this type of problem. The most appropriate choice is not obvious on first inspection. We develop and implement two related variants of a particular model for this problem. In both variants, we model it as a binary *integer programming* (IP) problem with many constraints. Our models use these many constraints to capture the way being assigned a particular time slot affects the value a tutor places on being assigned a different time slot. Our model can thus obtain an exact solution while simultaneously accounting for the complicated interdependencies between time slots.

The advantage of using integer programming to model the problem is that IP models of the size and type we formulate are relatively tractable and well tuned algorithms in commercially available software, e.g., ILOG's CPLEX [3], already exist for solving IP problems. Some particular characteristics of our problem lead to particularly fast solution times. If the values of the right hand side (RHS) of the constraints are generally kept to 0 or 1, the *cut and bound* methods CPLEX uses can be very effective. Unfortunately, worst case behavior may be problematic, even for relatively small instances of the problem.

Throughout the rest of this paper, we will continually refer to the two variants of our model. Henceforth, we shall refer to them as the *constraint relaxation approach* and the *penalty approach*. The constraint relaxation approach is to formulate all the constraints necessary to weed out undesirable schedules. Then, from the subset of *good* feasible schedules, we choose the one that maximizes the value of the objective function. Under this formulation, there is no guarantee of feasibility. It may be that some constraints are exclusive and cannot be mutually satisfied. In this case, we progressively relax (remove) the least important infeasible constraint until we obtain a feasible solution. Under this approach, the trade off between relaxing constraints and increasing the value of the objective function is difficult to quantify.

The penalty approach is to formulate the same set of constraints as in the constraint relaxation approach, but to add a new variable to each constraint. This new variable has a value of 0 if the original constraint is satisfied and 1 if the original constraint is not satisfied. The new variables are then incorporated into the objective function with an appropriate negative cost. Thus, maximizing the value of the objective function trades off increasing the ranking tutors gave their assigned time slots with reducing the number of violated constraints.

The penalty approach will always have a feasible solution and with the right costs, it is possible to trade-off poor schedule quality for an increased objective value in a meaningful way. The disadvantage is that it increases the size of the problem. It introduces one new variable for every constraint and as we discuss later, the number of constraints is roughly of the same order as the number of original variables. Thus, the problem roughly doubles in size.

3. Problem Formulation

Both of the approaches outlined above share some common elements. The objective function of the constraint relaxation approach is one part of the objective function used in the penalty approach. Also, the first three constraints (discussed below) are the same under both approaches. We call these first three constraints the 'hard' constraints. The additional constraints will be called 'soft' constraints. We shall assume that the 'hard' constraints are fixed for any given problem.

We shall first discuss the variables we use in our model. Next, we shall discuss all eight constraints under both the constraint relaxation approach and the penalty approach. Finally, we shall outline the two objective functions we use for the two approaches.

3.1. **Variables.** We define binary decision variables x_{jdh}, where $j = 1, 2, \ldots, J$ is the index of the tutors being scheduled, with J being the total number of tutors; $d = 1, 2, \ldots, D$ is the index of the days of the week, with D being the number of days the MSC is open each week; and $h = 1, 2, \ldots, H_d$ is the index of the hours of the day, with 1 being the first scheduled hour of the day and H_d being the last hour of day d. We assign $x_{jdh} = 1$, if tutor j is assigned hour h of day d, and 0 otherwise. The total number of variables is the total number of tutors times the total number of time slots, i.e. the number of x variables is equal to $J \cdot \left(\sum_{d=1}^{D} H_d\right)$. Additional dummy variables are introduced for certain constraints and are outlined below under the applicable constraint. In general, the number of tutors, the number of days the MSC is open a week, and how many hours it is open each day varies from quarter to quarter.

Binary Integer Variables (0 or 1) — all lower case	
x_{jdh}	1 if tutor j is assigned hour h on day d
y_{jdh}	1 if tutor j enters MSC next hour after h on day d (Constraint 6)
z_{jdh}	1 if tutor j is assigned an early hour h on day d (Constraint 8)
a^4_{jd}	1 if tutor j is assigned more than N_{jd} hours on day d (Constraint 4)
a^5_{jdh}	1 if tutor j is assigned $P_{jd} + 1$ hours in a row on a day d starting at hour h (Constraint 5)
a^6_{jd}	1 if tutor j is assigned more than 2 blocks of hours in a day (Constraint 6)
a^7_{jdh}	1 if tutor j is assigned a single hour h on day d and $V_{jdh} = 0$ (Constraint 7)
a^8_{jdh}	1 if tutor j is assigned 1 hour greater than spread S_{jd} (Constraint 8)
	all equal 0 otherwise
Indices	
j	index of tutors
d	index of days
h	index of time slots (hours)
h_0	dummy index to index time slots
Parameters — all upper case	
H_d	number of hours on day d
D	number of days MSC is open per week
J	total number of tutors at MSC
W_{jdh}	weight given by tutor j to hour h on day d
B_{dh}	number of tutors needed on day d at hour h
M_j	number of hours tutor j may be assigned each week
N_{jd}	number of hours tutor j may be assigned on day d
P_{jd}	number of hours tutor j may be assigned in a row on day d
V_{jdh}	1 if tutor j may be assigned a single hour h on day d
S_{jd}	maximum allowed spread between first hour and last hour of day d for tutor j
C_4	the cost of violating Constraint 4 (negative number)
C_5	the cost of violating Constraint 5 (negative number)
C_6	the cost of violating Constraint 6 (negative number)
C_7	the cost of violating Constraint 7 (negative number)
C_8	the cost of violating Constraint 8 (negative number)

Table 1: A summary of the variables, indices, and parameters of the model.

Next, we define the constraints. Constraints are considered in their order of importance. The first three 'hard' constraints are considered the most important and will be relaxed only in the face of infeasibility.

In the event that there is no feasible solution under the first three constraints— for example, there are not enough tutors available to fill the required time slots— it is assumed that either the supply or demand will be adjusted exogenously (by hand) to obtain feasibility and the problem will be resubmitted.

3.2. **The First Constraint: Available Hours.** Some tutors cannot be assigned particular time slots. We represent this by forcing the variable x_{jdh} to zero for those time slots, implying,

$$x_{jdh} = 0, \tag{1}$$

for every tutor j who marks a zero in hour h on day d. This constraint is identical under both the constraint relaxation approach and the penalty approach. The number of constraints is equal to the number of zeros marked by tutors. In practice, these variables are removed by the software package CPLEX in its preprocessing phase.

3.3. **The Second Constraint: Demand for Tutors.** There must be a certain number of tutors, typically between 1 and 5, in any given time slot. This implies,

$$\sum_{j=1}^{J} x_{jdh} = B_{dh} \quad \forall \quad d, h, \tag{2}$$

where $B_{dh} \in \{1, 2, \ldots, 5\}$ is the number of required tutors for the corresponding time slot. This constraint is identical under both the constraint relaxation and the penalty approaches. The number of constraints is equal to the total number of time slots.

3.4. **The Third Constraint: Supply of Tutors.** It is assumed that the supply of tutors matches the demand for tutors. The management of the MSC may preferentially adjust the number of hours per week some tutors are assigned in order to achieve this. At the MSC, a distinction is made between graduate and undergraduate tutors. Graduate tutors have a fixed number of hours they must be assigned and undergraduate tutors have a maximum number of hours they may be assigned.

$$\sum_{d=1}^{D} \sum_{h=1}^{H_d} x_{jdh} \;=\; M_{\text{grad}} \qquad \forall \quad j_{\text{grad}}, \tag{3}$$

$$\sum_{d=1}^{D} \sum_{h=1}^{H_d} x_{jdh} \;\leq\; M_{\text{undergrad}} \quad \forall \quad j_{\text{undergrad}}, \tag{4}$$

where M_{grad} is the required number of hours that a particular graduate student must tutor each week, and $M_{\text{undergrad}}$ is the maximum number of hours a particular undergraduate student may tutor in a week. If supply equals demand,

$$\sum_{j=1}^{J} M_j = \sum_{d=1}^{D} \sum_{h=1}^{H_d} B_{dh}, \tag{5}$$

then equation (4) becomes an equality. Both of these sets of constraints are identical under both approaches. The number of constraints of type (3) and (4) are equal to the number of graduate tutors, J_{grad}, and undergraduate tutors, $J_{undergrad}$, respectively.

3.5. **The Fourth Constraint: Hours per Day.** It is understood that tutors do not want to be assigned more than a certain number of hours on any given day. This number may vary depending on the student and depending on the day. For example, students typically wish to be assigned more hours per day on the weekends, as they have to come to campus just for this purpose. Under the constraint relaxation approach, the constraint is,

$$\sum_{h=1}^{H_d} x_{jdh} \leq N_{jd} \quad \forall \quad j, d, \tag{6}$$

where N_{jd} is a parameter for the maximum number of hours student j wishes to be assigned on day d. Although we use the same value of N_{jd} for all tutors in the instances we solve later, in theory these parameters could be uniquely specified by the individuals themselves. A typical value is $N_{jd} = 5$.

For constraints 4-8 (the 'soft' constraints), the penalty approach differs from the constraint relaxation approach in a very similar manner. In each case, we introduce new dummy binary integer variables that are added to the right hand side of the constraints. If the original constraint was infeasible, the new constraint will be feasible, but only if the added variable assumes a value of one. This variable is then included in the objective function with a negative cost. This negative cost results in the new variable assuming a value of one only if necessary. This is a method of relaxing the constraints, but only at the cost of reducing the value of the objective function. The precise nature of the relaxation varies from constraint to constraint, but in each case it involves the value of a binary integer variable changing from 0 to 1. At a sufficient penalty all of the 'soft' constraints may be relaxed. The following example may make this more concrete.

Under the penalty approach we add a new variable a^4_{jd} and the new fourth constraint is

$$\sum_{h=1}^{H_d} x_{jdh} \leq N_{jd} + (H_d - N_{jd}) \cdot a^4_{jd} \quad \forall \quad j, d. \tag{7}$$

This allows the number of hours a tutor is assigned on a particular day to equal H_d, which would have violated the original constraint, but this can happen if and only if $a^4_{jd} = 1$.

We assign a large negative cost to this variable in the objective function. We use a value of $C_4 = -5$. The number of constraints is equal to the number of tutors times the number of days.

3.6. **The Fifth Constraint: Hours in a Row.** Tutors do not wish to be assigned more than a certain number of hours in a row. In addition to the workers' preferences, the management of the MSC does not want tutors to work more than 5 hours in a row and would prefer they work no more than three hours in a row. Let parameter P_{jd} represent the maximum number of hours that tutor j may be scheduled in a row on day d. This is equivalent to saying that tutors may be assigned no more than P_{jd} hours in any given contiguous $(P_{jd} + 1)$ hour block on any given day. Therefore, under the constraint relaxation approach we have

$$\sum_{h=h_0}^{h_0+P_{jd}} x_{jdh} \leq P_{jd} \quad \forall \quad j, d, \text{ and } h_0 = 1, 2, \ldots, H_d - P_{jd}. \tag{8}$$

Clearly, this constraint is only applicable if P_{jd} is less than H_d, that is the number of hours a tutor may be assigned in a row is less than the number of hours the MSC is open that day. Typical values for P_{jd} are between 3 and 5.

Under the penalty approach, we again add a new variable $a^5_{jdh_0}$ and the fifth constraint becomes,

$$\sum_{h=h_0}^{h_0+P_{jd}} x_{jdh} \leq P_{jd} + a^5_{jdh_0} \quad \forall \quad j, d, \text{ and } h_0 = 1, 2, \ldots, H_d - P_{jd}. \tag{9}$$

In this case, every time a tutor is scheduled for an hour beyond P_{jd} hours in row, $a^5_{jdh_0}$ equals one. For example if P_{jd} equals three and the tutor is scheduled for the first five hours of the day, all in a row, then the two variables a^5_{jd1} and a^5_{jd2} equal one and all other $a^5_{jdh_0}$ for that day and tutor are equal to zero. We assign a moderate negative cost to this variable in the objective function. We use a value of $C_5 = -2$. The number of constraints is equal to the number of blocks in each day times the number of employees times the number of days.

3.7. **The Sixth Constraint: Blocks per Day.** Tutors prefer that they be assigned no more than two blocks of hours in each day. In other words, employees do not want to have to enter and leave the Math Study Center more than twice a day. This constraint is slightly more complicated than the constraints we have considered previously. We introduce a dummy variable y_{jdh} that is equal to 1 every time a tutor enters the MSC. Note that if a tutor will enter the MSC in the next hour, this implies they are assigned the next hour, but they are not assigned the current hour. Recall that an x variable takes value one if a tutor is assigned that hour and a value of zero if they are not. Thus, there is a pattern $\boxed{0}\boxed{1}$ in the vector of x variables every time a tutor enters the MSC to begin tutoring, as illustrated below.

$$x_{jd} \rightarrow \ldots \overset{h_0}{\boxed{0}}\overset{h_0+1}{\boxed{1}} \ldots \Rightarrow x_{jd(h_0+1)} - x_{jdh_0} = 1 \Rightarrow y_{jdh_0} = 1.$$

If we subtract the current hour from the next hour, we obtain a value that is equal to 1 only if the pairwise pattern in the vector of x variables is $\boxed{0}\boxed{1}$ and a value equal to 0 or -1 in all other cases. We do this for every pair of hours in the day h_0. There is one less pair of hours than there are hours in the day H_d. Under the constraint relaxation approach, this is,

$$x_{jd(h_0+1)} - x_{jdh_0} \leq y_{jdh_0} \quad \forall \quad j, d, \text{ and } h_0 = 1, 2, \ldots, H_d - 1. \tag{10}$$

After we check every pair of x values in the day, we then add a constraint that the sum of y_{jdh}—the number of times the tutor enters the MSC for the day—be no greater than two (the number of blocks of hours) and we obtain the constraint,

$$\sum_{h=1}^{H_d-1} y_{jdh} \leq 2 - x_{jd1} \quad \forall \quad j, d. \tag{11}$$

The variable x_{jd1} on the right hand side of the above equation is meant to account for the fact that y_{jdh} does not determine if the tutor was assigned the first hour of the day. Note, $y_{jd1} = 1$ implies that $x_{jd1} = 0$ and $x_{jd2} = 1$, that is the tutor entered the MSC on the second hour of the day. So if the tutor was assigned the first hour of the day, we must ensure that they enter the MSC only one more time by subtracting one from the right hand side of Constraint (11). This constraint adds a large number of dummy variables (the number of y variables is of the same order as the number of x variables) and a large number of inequality constraints (one for each x variable).

Under the penalty approach we add the variable a_{jd}^6 to Constraint (11) such that,

$$\sum_{h=1}^{H_d-1} y_{jdh} \leq 2 - x_{jd1} + \left(\left\lceil \frac{H_d}{2} \right\rceil - 2 \right) \cdot a_{jd}^6 \quad \forall \quad j, d. \tag{12}$$

Tutors may now enter the MSC every other hour when $a_{jd}^6 = 1$. We assign a large negative cost to this variable in the objective function. We use a value of $C_6 = -5$. The number of constraints is equal to the number of hours in each day times the number of employees times the number of days.

3.8. **The Seventh Constraint: No Single Hours.** Tutors prefer not to be assigned only one hour at a time. They prefer to work for at least two hours in a row. Again we are looking for a pattern in the values of the vector of x variables, specifically, if a tutor is assigned a single hour we see the pattern $\boxed{0}\boxed{1}\boxed{0}$ in the values of the x variables. If we want to prevent this pattern from being assigned, we add a constraint that the x value of the first plus third hour must be greater than or equal to the x value of the second hour for every triplet of hours in the day. This constraint forces every hour that is assigned to have another hour assigned beside it on one side or the other, for example $\boxed{1}\boxed{1}\boxed{0}$ or $\boxed{0}\boxed{1}\boxed{1}$.

We provide an exception to this constraint. If a tutor has ranked a time slot as 3 (very high), but marked the time slots on both sides of it as 0 (prohibited), it is often the case that the time slot fills a one hour gap in the students class schedule. This is often highly desired because it makes the student's day very compact. We define a new parameter V_{jdh} such that,

$$V_{jdh} = \begin{cases} 1, & \text{if } h = 1,\ W_{jd1} = 3 \text{ and } W_{jd2} = 0; \text{ (first hour)} \\ 1, & \text{if } W_{jdh} = 3 \text{ and } W_{jd(h-1)} = W_{jd(h+1)} = 0; \text{ (hour in the middle of the day)} \\ 1, & \text{if } h = H_d,\ W_{jdH_d} = 3 \text{ and } W_{jd(H_d-1)} = 0; \text{ (last hour)} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

This parameter is computed from the student rankings and will be used in Constraints (14)–(19) to provide an exception to the general constraints that try to prevent the scheduling of single hours. This parameter is being inferred from the student's rankings, but it could also be provided by the students directly.

The first and last hour of the day also add a slight complication. In these two cases, the pattern in the values of the x variables that we are trying to prevent is $\boxed{1}\boxed{0}$ for the first two hours of the day and $\boxed{0}\boxed{1}$ for the last two hours of the day. Constraints (14) and (16) represent the constraints on the first and last hour of the day.

This reasoning leads to the following constraints (under the constraint relaxation approach),

$$x_{jd1} \leq V_{jd1} + x_{jd2} \quad \forall \quad j,d, \tag{14}$$

$$x_{jdh_0} \leq V_{jdh_0} + x_{jd(h_0-1)} + x_{jd(h_0+1)} \quad \forall \quad j,d, \text{ and } h_0 = 2,3,\ldots,H_d-1, \tag{15}$$

$$x_{jdH_d} \leq V_{jdH_d} + x_{jd(H_d-1)} \quad \forall \quad j,d. \tag{16}$$

Each of these constraints is always satisfied in the case $V_{jdh} = 1$.

Following the usual construction under the penalty approach, we add variable a^7_{jdh} and obtain,

$$x_{jd1} \leq V_{jd1} + a^7_{jd1} + x_{jd2} \quad \forall \quad j,d, \tag{17}$$

$$x_{jdh_0} \leq V_{jdh_0} + a^7_{jdh_0} + x_{jd(h_0-1)} + x_{jd(h_0+1)} \quad \forall \quad j,d, \text{ and } h_0 = 2,3,\ldots,H_d-1, \tag{18}$$

$$x_{jdH_d} \leq V_{jdH_d} + a^7_{jdH_d} + x_{jd(H_d-1)} \quad \forall \quad j,d. \tag{19}$$

Now if either $V_{jdh_0} = 1$ or $a^7_{jdh} = 1$ the constraint is always satisfied regardless of whether tutor j was assigned a single hour h on day d. We assign a smaller negative cost to the variable a^7_{jdh} in the objective function. We use a value of $C_6 = -1$. The number of constraints is equal to the number of hours in each day times the number of employees times the number of days.

3.9. **The Eighth Constraint: No Large Spread.** Tutors prefer not to have schedules where there is a large spread between the first hour they are assigned and the last hour they are assigned. For example they do not wish to be assigned the first hour of the day and then come back 10 hours later to work during the last hour of the day. Let parameter S_{jd} be the maximum allowed spread between the first hour assigned and the last hour assigned for student j on day d. So if a tutor is assigned the first hour, they do not wish to be assigned any hours past S_{jd}. We use a technique from [4] (see also [5]) to convert or-type constraints into and-type constraints. We create a new binary integer variable z_{jdh}. Under the constraint relaxation approach we have,

$$x_{jdh_0} \leq 1 - z_{jdh_0} \quad \forall \quad j,d, \text{ and } h_0 = 1,2,\ldots,H_d - S_{jd}, \tag{20}$$

$$\sum_{h=h_0+S_{jd}}^{H_d} x_{jdh} \leq (H_d - S_{jd}) \cdot z_{jdh_0} \quad \forall \quad j,d, \text{ and } h_0 = 1,2,\ldots,H_d - S_{jd}. \tag{21}$$

Note that $z_{jdh_0} \in \{0,1\}$. Thus if the right hand side of Constraint (20) equals one, then the right hand side of Constraint (21) must equal zero. On the other hand, if the right hand side of Constraint (20) equals zero then Constraint (21) imposes no restrictions. In this way the binary variable z_{jdh} requires that if the tutor is assigned hour h_0, then they cannot be assigned any hours at or past $h_0 + S_{jd}$. Clearly this constraint is only applicable if $H_d > S_{jd}$.

Under the penalty approach, we add a new variable a^8_{jdh} to constraint (20) and obtain,

$$x_{jdh_0} \leq 1 - z_{jdh_0} + a^8_{jdh_0} \quad \forall\, j,d, \text{ and } h_0 = 1,2,\ldots,H_d - S_{jd}. \tag{22}$$

If a tutor's schedule exceeds the allowable spread S_{jd}, it forces the new variable a^8_{jdh} to equal one. We assign a moderate negative cost to this variable in the objective function. We use a value of $C_8 = -2$. The number of constraints is $(H_d - S_{jd})$ times the number of days times the number of tutors.

3.10. **The Objective Function.** We define two different objective functions. One is for the constraint relaxation approach, which we shall label *Objective(I)* and one is for the penalty approach, which we shall label *Objective(II)*.

The purpose of Objective(I)—under the constraint relaxation approach—is to maximize the collective ranking given by the tutors to their respective scheduled hours

$$\max \quad \sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{h=1}^{H_d} W_{jdh} x_{jdh}, \tag{23}$$

where $W_{jdh} \in \{0,1,2,3\}$ is the ranking tutor j gave the time slot at hour h on day d (0 is given if the tutor does not wish to be assigned the corresponding time).

Under the penalty approach, we wish to maximize Objective(II). Objective(II) includes Objective(I), but also incorporates additional negative costs for violating the constraints. Constraints 1-3 are required and thus there are no costs. The variables a^4_{jd}, $a^5_{jdh_0}$, a^6_{jd}, a^7_{jdh}, $a^8_{jdh_0}$ are associated with Constraints 4-8 and were defined above. They are equal to 1 if the respective constraint is violated and 0 if it is not. Objective(II) is then,

$$\begin{aligned}(24)\qquad \max \quad & \sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{h=1}^{H_d} W_{jdh}x_{jdh} + \sum_{j=1}^{J}\sum_{d=1}^{D} C_4 a^4_{jd} + \sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{h_0=1}^{H_d-P_{jd}} C_5 a^5_{jdh_0} \\ & + \sum_{j=1}^{J}\sum_{d=1}^{D} C_6 a^6_{jd} + \sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{h=1}^{H_d} C_7 a^7_{jdh} + \sum_{j=1}^{J}\sum_{d=1}^{D}\sum_{h_0=1}^{H_d-S_{jd}} C_8 a^8_{jdh_0},\end{aligned}$$

where C_i is the negative cost associated with violating constraint i.

4. Implementation

The two approaches used in our model were tested on three particular instances. We chose quarters of increasing complexity to test our model. The parameters used for the three instances are presented in Table 2. The Summer 2003 quarter has a reduced number of tutors and time slots. Autumn 2002 is a typical quarter. The Winter 2003 quarter had the greatest number of tutors and was the largest instance available.

The parameters used were the same for all tutors, but tailoring the parameters to individual students can be done easily enough. The schedule actually used for the Summer 2003 quarter (generated using the original system of heuristics) is shown in Figure 1.

Quarter	**Year**	**Tutors**	**Days**	**Total Hrs.**
Summer	2003	8	5	30
Parameters				
Days - d	$\mathbf{H_d}$	$\mathbf{N_d}$	$\mathbf{P_d}$	$\mathbf{S_d}$
1	9	5	3	6
2	4	5	4	4
3	9	5	3	6
4	4	5	4	4
5	4	5	4	4

Quarter	**Year**	**Tutors**	**Days**	**Total Hrs.**
Autumn	2002	15	6	56
Winter	2003	20	6	56
Parameters - Same for Both Quarters				
Days - d	$\mathbf{H_d}$	$\mathbf{N_d}$	$\mathbf{P_d}$	$\mathbf{S_d}$
1	12	5	3	8
2	12	5	3	8
3	12	5	3	8
4	12	5	3	8
5	4	5	4	4
6	4	5	4	4

Table 2: Parameters used for the three example quarters.

Figures 1 through 3 are the Summer 2003 quarter schedules generated by the original system and the two approaches we tested. Supplementary material containing the schedules for the Autumn 2002 and the Winter 2003 quarters are available at http://www.math.washington.edu/∼thomas/papers/articles.html. They all have the same basic layout. The tutors' names are abbreviated on the left. Graduate students are capitalized and undergraduates are in lowercase letters. The row of numbers under the days of the week represent how many tutors were required for that time slot, B_{dh}. The column of numbers on the far right is how many hours each tutor was assigned, M_j. The numbers in the main field represent the rankings tutors gave individual time slots. The shaded boxes are the hours that were actually scheduled. Note, no shaded box has a zero ranking, the number of shaded boxes in each column equals B_{dh}, and the number of shaded boxes in each row equals M_j. Thus, you can see visually that the 'hard' constraints were satisfied for every schedule.

To solve these instances, the models for each quarter were created using Matlab 6.5 and solved using CPLEX 8.1 on a Linux box with two 1.4 Ghz processors and 1.2 Gbytes of memory.

Each instance was modeled using Matlab to generate the objective and constraint matrices. Then, the *LP-relaxation* was checked for feasibility in the first three constraints using Linprog in Matlab. If the problem was infeasible in the 'hard' constraints, then the supply or demand was manually adjusted to ensure feasibility. Once the problem was feasible in the first three constraints, the LP-relaxation was checked with all the constraints. If some 'soft' constraints were infeasible under the constraint relaxation approach, those constraints were removed by using a binary sort to move the infeasible constraints to the bottom of the constraint matrix before removing them entirely. After the feasible LP-relaxation was found, the model was read into CPLEX and solved. It is possible that the problem is feasible in the LP-relaxation, but not feasible as an IP. If the IP problem is infeasible, CPLEX can pinpoint the constraint creating the infeasibility and that constraint can be removed manually and the process repeated.

5. Computational Results and Conclusions

The size of the models created for each quarter and the computational time required to solve them using CPLEX are shown in Table 3. It was assumed that because of the larger size of the models using the penalty approach, they would require significantly longer times to solve. This turned out not to be the case. The problem was actually solved more quickly using the penalty approach for these three particular instances.

Quarter	Approach	# Variables	# Equality Constraints	# Inequality Constraints	Time to Solve (Sec.)
Sum. 03	Relaxation	488	36	714	0.24
	Penalty	952	36	714	0.07
Aut. 02	Relaxation	1830	64	2797	12.19
	Penalty	3630	64	2797	9.72
Win. 03	Relaxation	2440	68	3728	514.67
	Penalty	4840	68	3728	156.35

Table 3: Comparison of the size of the problem and the solution time.

The solutions generated by the constraint relaxation approach and the penalty approach for the Summer 2003 quarter are included in Figures 2 and 3. A summary of the results for all three quarters, including the value of the objective functions; Objective(I) and Objective(II), appears in Table 4.

It was always possible to satisfy the first four constraints, so those columns are not shown in Table 4. The number of times constraints 5-8 were violated under each approach is shown in the respective column. The numbers equal the sum of the a^i variable for the respective constraint i. For example, the Autumn 2002 quarter schedule under the system of heuristics violated a number of the constraints. On two days there were a total of three scheduled hours that occurred immediately after a tutor had already been assigned three hours in a row. There were two times when a tutor was assigned three blocks of hours in a single day. There were five times when a tutor was assigned a single hour that did not meet the exception rule. Finally, there was a single time when a tutor was assigned a single hour that was outside the allowed spread of eight hours for that day.

In contrast, the Autumn 2002 quarter schedule generated by the Constraint approach only failed to satisfy a single constraint, yet still managed to achieve a higher value for Objective(I). In fact, the constraint that was violated was violated under all three approaches and involved a single hour that had to be scheduled in order to satisfy the more important 'hard' constraints.

			Constraints 5 through 8					
Quarter	Approach	Objective (I)	Hours in a Row $C_5 = -2$	Blocks in a day $C_6 = -5$	# Single Hours $C_7 = -1$	# Hours > Spread $C_8 = -2$	Objective (II)	% Inc. over Heuristic Approach
Sum. 03	Heuristic	**253**	13	0	0	1	**225**	
	Relaxation	**253**	10	0	1	1	**230**	2.2%
	Penalty	**257**	10	0	3	0	**234**	4.0%
Aut. 02	Heuristic	**370**	3	2	5	1	**347**	
	Relaxation	**372**	0	0	1	0	**371**	6.9%
	Penalty	**375**	0	0	3	0	**372**	7.2%
Win. 03	Heuristic	**426**	6	0	13	0	**401**	
	Relaxation	**427**	0	0	0	0	**427**	6.5%
	Penalty	**427**	0	0	0	0	**427**	6.5%

Table 4: Values of the objective functions and the 'soft' constraint violations.

The penalty approach actually scheduled three single hours (beyond the allowed exceptions), but in doing so it managed to increase the sum of the individual rankings—Objective(I)—by a value of five over the heuristic approach and three over the constraint relaxation approach. Note the cost of scheduling a single hour was -1, so even though the penalty approach schedules two more single hours than the constraint relaxation approach, because it manages to increase the value of Objective(I) by three, it comes out one point better in Objective (II).

The comparison in the Winter 2003 quarter is just as dramatic. The constraint relaxation approach and the penalty approach both managed to create a schedule with a value of Objective(I) one point better than the heuristic system, but they accomplished it without violating any of the constraints.

In reviewing the results, some important observations can be made. The two new approaches always created better schedules than the previous system of heuristics. The differences in Objective(I) were fairly small, but the two new approaches generated much *higher quality* solutions, as is indicated by the value of Objective(II). The number of times the new schedules violated the rules for generating a good schedule was much smaller. The new methods appear to do even better than the original method on the more complex problems. Our two new approaches, and the penalty approach in particular, are also much more flexible. The relative importance of the various constraints can be adjusted by choosing the appropriate costs and students can individually tailor their preferences for the various parameters.

In general, the penalty approach is faster, it attains a better solution, it is more flexible, and it does not require the overhead of sorting out and removing the infeasible constraints. Thus, the penalty approach is the preferred method of creating weekly timetables that maximize the MSC employee preferences. Since this work was done, Patrick Perkins has been using an implementation of the penalty model and CPLEX to compute work schedules in the MSC. He has used it on three different occasions, including the busiest times of the school year. The results have been very satisfactory. The implementation is easier to use than the previous method and much quicker. The average time it took CPLEX to compute a solution was about 5 seconds. The maximum was under 12 seconds. His employees have expressed satisfaction with the general improvement in the schedules.

Some additional observations may be made. The nature of the individual preferences has a lot of bearing on how the final solution turns out. Many students, especially in the Summer 2003 example, gave a positive ranking only to the exact number of hours that they were available, thus the system was forced to select the precise hours they chose. A successful model should be adjusted so as to prevent tutors from "gaming" the system. It should also be noted that this work does not address the system of rankings that the tutors use, nor does it address how the value of the parameters and the costs for violating constraints should be calculated. The method of using a summation of finite discrete rankings as the objective function to determine the preferred overall schedule is probably not optimal. Tutors should be able to indicate more precisely the intensity of their preferences for particular hours. Future work might explore the appropriate mechanism to attain individual schedule preferences and aggregate them into a single measure.

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Days	MONDAY									TUES.				WEDNESDAY									THURS.				FRIDAY				
Tutrs./ Hour	2	2	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	4	4	5	5	2	3	4	4	2	2	2	2	Hrs./ week
PAN.	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	20
RYA.	2	2	2	**2**	**2**	**3**	**3**	3	3	**3**	**3**	3	3	**3**	**3**	3	3	0	0	0	0	0	3	3	**3**	**3**	1	1	1	1	10
GAR.	0	0	**3**	**3**	**3**	**3**	3	3	2	0	0	1	1	0	0	3	3	3	**3**	**3**	**3**	2	0	0	1	1	0	0	**2**	**2**	10
TRA.	1	1	1	1	1	1	1	1	1	2	**3**	**3**	**3**	2	3	3	3	3	**3**	**2**	**2**	**2**	2	**3**	**3**	**3**	2	2	2	2	10
CAT.	**2**	**3**	**3**	3	2	2	1	1	1	2	3	**3**	**3**	2	3	**3**	**2**	**2**	0	0	**1**	**1**	2	3	3	2	2	2	2	2	10
ISH.	0	0	0	0	0	3	3	**3**	**3**	3	3	3	3	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	0	0	0	0	10
mic.	3	3	3	3	3	3	**3**	**3**	**3**	3	3	3	3	3	3	3	3	**3**	**3**	**3**	**2**	**2**	3	3	3	3	**3**	**3**	3	3	10
geo.	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	10

Figure 1: Schedule created for Summer 2003 Quarter using system of heuristics.

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Days	MONDAY									TUES.				WEDNESDAY									THURS.				FRIDAY				
Tutrs./ Hour	2	2	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	4	4	5	5	2	3	4	4	2	2	2	2	Hrs./ week
PAN.	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	20
RYA.	2	2	2	2	2	3	3	**3**	**3**	**3**	**3**	3	3	**3**	**3**	**3**	3	0	0	0	0	0	3	**3**	**3**	**3**	1	1	1	1	10
GAR.	0	0	**3**	**3**	3	**3**	**3**	3	2	0	0	1	1	0	0	3	3	3	**3**	**3**	**3**	**2**	0	0	1	1	0	0	**2**	**2**	10
TRA.	1	1	1	1	1	1	1	1	1	2	3	**3**	**3**	2	3	3	3	3	**3**	**2**	**2**	**2**	2	3	**3**	**3**	**2**	**2**	2	2	10
CAT.	2	3	3	**3**	**2**	2	1	1	1	2	**3**	**3**	**3**	2	3	**3**	**2**	**2**	0	0	**1**	**1**	2	3	3	2	2	2	2	2	10
ISH.	0	0	0	0	0	3	**3**	**3**	**3**	3	3	3	3	**3**	**3**	3	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	0	0	0	0	10
mic.	**3**	**3**	**3**	3	**3**	**3**	3	3	3	3	3	3	3	3	3	3	3	**3**	**3**	**3**	**2**	**2**	3	3	3	3	3	3	3	3	10
geo.	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	10

Figure 2: Schedule created for Summer 2003 Quarter using constraint relaxation approach.

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Days	MONDAY									TUES.				WEDNESDAY									THURS.				FRIDAY				
Tutrs./ Hour	2	2	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	4	4	5	5	2	3	4	4	2	2	2	2	Hrs./ week
PAN.	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	20
RYA.	2	2	2	2	2	**3**	**3**	3	**3**	3	3	3	3	**3**	**3**	**3**	3	0	0	0	0	0	**3**	**3**	**3**	**3**	1	1	1	1	10
GAR.	0	0	**3**	**3**	**3**	3	**3**	**3**	2	0	0	1	1	0	0	3	3	**3**	**3**	**3**	**3**	**2**	0	0	1	1	0	0	2	2	10
TRA.	1	1	1	1	1	1	1	1	1	2	3	**3**	**3**	2	3	3	**3**	3	**3**	**2**	**2**	**2**	2	**3**	**3**	**3**	2	2	2	2	10
CAT.	2	3	**3**	**3**	**2**	2	1	1	1	2	**3**	**3**	**3**	2	3	3	**2**	**2**	0	0	**1**	**1**	2	3	3	2	2	2	2	2	10
ISH.	0	0	0	0	0	**3**	3	**3**	**3**	**3**	**3**	3	3	**3**	**3**	**3**	3	0	0	0	0	0	3	3	**3**	**3**	0	0	0	0	10
mic.	**3**	**3**	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	**3**	**3**	**2**	**2**	3	3	3	3	**3**	**3**	**3**	**3**	10
geo.	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	**3**	**3**	**3**	**3**	**3**	0	0	0	0	0	0	0	0	10

Figure 3: Schedule created for Summer 2003 Quarter using penalty approach.

References

[1] E. Burke and S. Petrovic, *Recent Research Directions in Automated Timetabling*, European Journal of Operational Research **140** (2002), 266-280.

[2] William J. Cook, William H Cunningham, William R. Pulleyblank, and Alexander Schrijver, *Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, 1998.

[3] ILOG, *ILOG CPLEX 8.0 User's Manual*, 2002.

[4] V. Chandru and J.N. Hooker, *Optimization Methods for Logical Inference*, John Wiley and Sons, 1999.

[5] A. Richard, T. Schouwenaars, J. How, and E. Feron, *Spacecraft Trajectory Planning with Collision and Plume Avoidance Using Mixed-Integer Linear Programming*, AIAA Journal of Guidance, Control, and Dynamics (July, 2002).

[6] A. Schaerf, *A Survey of Automated Timetabling*, Artificial Intelligence Review **13 (2)** (1999), 87-127.

Caleb White, Department of Economics, PO Box 353330
E-mail address: zeon@u.washington.edu

Youngbae Lee, LG Chemical Ltd. in Seoul, Korea
E-mail address: rooftree@u.washington.edu

Yoonsoo Kim, Department of Aeronautics and Astronautics, PO Box 352400
E-mail address: yoonsoo@aa.washington.edu

Rekha Thomas, Department of Mathematics, PO Box 354350
E-mail address: thomas@math.washington.edu

Patrick Perkins, Director, Math Study Center, PO Box 354350
E-mail address: perkins@math.washington.edu

All (except Youngbae Lee) at: University of Washington, Seattle, WA 98195