A new convex optimization framework for approximately solving timetabling problems that can be described as integer linear programs is proposed. The method is based on converting the timetabling problem into a cardinality constrained problem while viewing the operational constraints of the timetable as noisy measurements with an unknown average slack. The problem is then iteratively solved using weighted Lasso with weights that are updated using a simple linear program to satisfy the cardinality constraint while respecting the operational constraints in the least squares sense. Compared with previous convex relaxations for solving timetabling problems, our solution technique will converge to a binary solution without the need for subsequent randomized rounding. Moreover, the number of Lasso iterations required are in the order of the number of events to be scheduled and hence the method can handle very large timetabling problems using efficient Lasso solvers. We provide the assumptions required on the linear timetabling model and establish the associated error bound of the technique upon convergence assuming restricted strong convexity and uniqueness. We also study the effect of the Lasso regularization parameter and the effect of relaxing the objective on the extent of constraint satisfaction through several simulation experiments. Our experiment on one of the benchmark datasets for university examination timetabling improved the best documented result by more than 30% with 99.9% of all constraints satisfied.

INDEX TERMS Automated timetabling, integer linear programming, convex optimization, lasso, examination timetabling, cardinality constrained optimization, combinatorial optimization, graph coloring.

I. INTRODUCTION

Timetabling is the assignment of events to a finite number of time-space resources while respecting a set of feasibility and operational constraints and meeting a certain desired objective [1]. Enhancing the efficiency and optimality of automated solutions of timetabling problems is receiving considerable attention; see for example the annual timetabling competitions in [2]–[6] and the surveys in [7]–[9].

Often timetabling problems can be modeled as integer linear programs that involve very large number of binary decision variables and linear constraints [10], [11]. These problems are known to be NP-complete and are difficult to solve for optimality [12]. Heuristics are commonly used to provide feasible solutions without any guarantee on global optimality of the solutions, see for example the recent surveys in [6] and a good summary review of heuristics in [9], [10], [13]. Alternatively, these problems can be also solved using exact methods, including cutting plane and/or branch and bound techniques that can be terminated with a certificate proving sub-optimality [14]. However, these techniques can take considerable amount of time and computational resources to solve, especially for timetabling problems that involve very large number of decision variables and constraints [15]. Efforts to improve the efficiency and size of integer programming methods have been previously studied using different techniques, including column generation [16], cutting plane [17], [18], decomposition methods [19] and multi-stage techniques [10].

As a compromise solution to exact methods, techniques that are based on convex relaxations of timetabling problems have been studied for developing efficient approximate solutions. These techniques include Linear Programming (LP) relaxations, Semidefinite Programming (SDP) relaxations and Lagrangian relaxations (LG). The linear programming relaxations are the basis for exact branch-and-bound based approaches and can sometimes provide a tight lower bound (for a minimization problem) of the solution (or even the...
optimal solution) when the constraint matrix is uni-modular [20]. However, this special requirement on the constraint matrix is seldom satisfied in timetabling problems [21].

SDP relaxations is a generalization of LP relaxations and can often provide tighter approximate solutions of timetabling problems. SDP relaxations for timetabling problems has been previously studied in [22]–[24]. However, as noted in [22], SDP relaxations can be too large to solve using existing SDP algorithms in view of the expansion in the number of variables involved. Moreover, the method often does not lend to a binary solution and requires randomized rounding which can devalue the obtained SDP solution and/or render it infeasible [22].

Lagrangian dual relaxations are based on solving the convex dual problem and has been used in solving school timetabling problems in [25] and large train timetabling problems, see for example [26]–[29]. There are different duals that can be formed from the original primal problem and each dual will have different characteristics [30]. These techniques are usually combined with exact methods like branch and bound and cutting plane approaches and can be used for both linear and nonlinear timetabling problems [27]. Although these techniques can handle large timetabling problems they also require randomized rounding techniques after obtaining the solution (similar to SDP relaxations). For very large timetabling problems, randomized rounding techniques on obtained solutions from LG relaxations can result in many infeasibilities as noted in the experiments conducted in [27] for large train timetabling problems.

An important observation related to integer linear program models developed for timetabling problems is that a cardinality constraint is implicit in these models where the cardinality is related to the number of binary 1 assignment variables in the solution that corresponds, in turn, to the number of events to be scheduled. In this study, we exploit this implicit cardinality constraint together with convex optimization methods for solving cardinality constrained problems. Particularly, we exploit sparse convex optimization techniques for approximately solving timetabling problems by reformulating the timetabling problem into a cardinality constraint problem while satisfying all linear inequality constraints in the least squares sense. Weighted Lasso (Least absolution shrinkage and selection operator) is used as the vehicle for solving timetabling problems that can be described as cardinality constrained problems. The technique will require converting operational constraints into soft constraints while enforcing constraints that are essential. The weights on Lasso are updated through a solution of a simple linear program with the objective of incrementally obtaining a binary solution with desired cardinality while satisfying all operational constraints in the least squares sense. Figure 1 depicts a summary block diagram explaining the new technique developed in this study.

Compared with previous convex relaxations for solving very large timetabling problems our technique does not require randomized rounding as the technique will converge to a binary solution. Most importantly, the method exhibits predictable convergence since the number of Lasso iterations required is in the order of the number of events to be scheduled as verified in all study experiments. Finally, due to the existence of efficient convex programming algorithms that are parallelizable for computing globally optimal solutions for Lasso, a convex Lasso formulation of the timetabling problem can offer tremendous efficiency and scalability improvement compared to semi-definite programming relaxation [31]. This is in contrast to parallelization efforts in integer linear programming that have met with only relatively modest success as mentioned in [31].

Hence, the contribution of this study can be summarized in the following points:

1) We presented the necessary assumptions on the timetabling integer linear program formulation to enable conversion of the timetabling problem into a cardinality constraint problem. We also demonstrated the applicability of the assumptions on the general graph coloring problem which is considered a basis for many linear timetabling problems.

2) We presented two algorithms for approximately solving the cardinality constraint problems originating from timetabling problems using iterative re-weighted Lasso. The first algorithm considers all problem constraints in each iteration while the second algorithm monitors constraint satisfaction to enable problem size reduction in subsequent iterations.

3) We bound the error of the technique upon convergence using a restricted strong convexity assumption [32]. The error bound derived was found to be a function of the variance of slack and the extent of convexity of the linear mapping imposed by the problem.

4) We study the effect of the Lasso regularization parameter and the effect of relaxing the objective on the extent of constraint satisfaction through several simulation experiments. Our experiments demonstrate that the Lasso regularization parameter need to be sufficiently large to ensure a final binary solution and constraint satisfaction can be significantly improved if a lower bound on the optimal value of the objective is known.

5) Finally, we demonstrate the potential of the new formulation through several numerical experiments including the approximate solution of a benchmark timetabling problem from the 2007 International timetabling competition. Our experiments demonstrated that the number of Lasso iterations required to solve the timetabling problem corresponds to the number of events to be scheduled. Moreover, our experiment on one of the benchmark problems in the 2007 International timetabling competition improved the best documented result by more than 30% with 99.9% of all constraints satisfied obtained using 1821 LASSO iterations.

The following is the outline of the study: Section II will introduce the problem formulation of the problem with the necessary assumptions. Section III will provide the
The original timetabling problem is described as a binary integer linear program (ILP) that is reformulated into a cardinality constraint problem. A reweighted Lasso sequence is then used to satisfy the cardinality constraint incrementally while the Linear operational constraints of the timetable are satisfied in the least squares sense. The weights of the Lasso program are updated using a simple linear program that optimally selects the weights based on the previous solution. Upon convergence of the technique, a binary solution is obtained without any subsequent randomized rounding.

**Additional Assignment Variables**

Additional assignment variables may also be part of $x_{\text{ass}}$ that are used to facilitate modeling. The binary decision vector $x_{\text{sup}} \in \{0, 1\}^{\sup}$ includes surplus binary decision variables to model deviation from the satisfaction of soft constraints. For example, if $x_{\text{ass}} + x_{\text{ass}} \leq 1$ is a linear constraint imposed by the problem, where $x_{\text{ass}}$ and $x_{\text{ass}}$ are assignment variables and $x_{\text{sup}}$ is a surplus variable, then $x_{\text{sup}} = 0$ when the constraint $x_{\text{ass}} + x_{\text{ass}} \leq 1$ is satisfied and $x_{\text{sup}} = 1$ if the constraint is not satisfied. Finally, auxiliary binary variables may be present to facilitate modeling and are represented by the subvector $x_{\text{aux}} \in \{0, 1\}^{\text{aux}}$. In the following, we will introduce the three main assumptions needed in the linear timetabling problems analyzed in this study.

**Assumption 1:** All feasible solutions of (1) will satisfy the cardinality condition $\|x_{\text{ass}}\| = n_s$, where $n_s$ is the number of events to be scheduled (timetabled) or a multiple of this number.

The first assumption in this study is related to the number of events to be scheduled in the timetabling problem. In order to schedule all events in a timetabling problem, the number of assignment variables that are equal to binary 1 in the solution should correspond to the number of events to be scheduled. In other words, the vector $x_{\text{ass}}$, defined earlier, which is a vector that stacks together all assignment variables (and auxiliary assignment variables), should have a number of elements equal to 1 corresponding to the number of events to be scheduled and zero for the remaining elements. This can be translated into a cardinality constraint on the vector $x_{\text{ass}}$, represented by $\|x_{\text{ass}}\| = n_s$, where $n_s$ is the number of events to be scheduled or a multiple of this number if auxiliary assignment variables are used. The following additional assumption will be required.
Assumption 2: The linear equality constraints $Ax = b$ in (1) include $n_s$ uniqueness constraints of the form:
\[ \sum_{i \in S_j} x_{\text{ass},i} = 1, \quad j = 1, \cdots, n_g \quad (2) \]
where $S_j$ represents the set of all assignment variables related to event $j$ and $x_{\text{ass},i}$ denotes the $i$th element of vector $x_{\text{ass}}$. Additional equality constraints linking assignment variables to auxiliary assignment variables can be also present in the model in (1).

The linear equality constraints of the form given in (2) ensure that each event $j$ is assigned to only one unique set of timetabling resource. Consequently, assumptions 1 and 2 combined will enable replacing the binary constraint $x_{\text{ass}} \in \{0, 1\}^{n_a}$ in (1) with a cardinality constraint $\|x_{\text{ass}}\|_0 = n_s$.

To see how this is possible, consider, for example, the set of binary vectors $u$ of dimension $n_a$ with cardinality equal to one; i.e., $\|u\|_0 = 1$. Then we can describe this set in two different ways:
\[ \{ \sum_{i=1}^{n_a} u_i = 1, \quad u \in \{0, 1\}^{n_a} \} = \{ \sum_{i=1}^{n_s} u_i = 1, \quad \|u\|_0 = 1 \} \]
where, $u_i$ represents the $i$th element of vector $u$. In other words, a binary vector with the summation of its elements equal to 1 can be expressed as a cardinality 1 vector with summation of its elements equal to 1. By induction, assumptions 1 and 2 allow us to replace the binary constraints on the assignment variables with a cardinality constraint through the following equivalence:
\[ \{ x_{\text{ass}} \mid \sum_{i \in S_j} x_{\text{ass},i} = 1, \quad j = 1, \cdots, n_s, \quad x_{\text{ass}} \in \{0, 1\}^{n_a} \} = \{ x_{\text{ass}} \mid \sum_{i \in S_j} x_{\text{ass},i} = 1, \quad j = 1, \cdots, n_s, \quad \|x_{\text{ass}}\|_0 = n_s \} \]

As demonstrated in appendix A and B, uniqueness constraints of the form given in (2) are used in the graph coloring formulation of timetabling problems which is a basis for many timetabling problems. We now present the third and final assumption needed in the timetabling ILP formulation.

Assumption 3: The binary constraints on surplus and auxiliary variables $x_{\text{sup}} \in \{0, 1\}^{n_{sup}}$ and $x_{\text{aux}} \in \{0, 1\}^{n_{aux}}$ in (1) can be replaced by convex box constraints $0 \leq x_{\text{sup}} \leq 1$ and $0 \leq x_{\text{aux}} \leq 1$.

In other words, the binary state of surplus and auxiliary variables are derived from the binary state of assignment variables and convex box constraints are sufficient to ensure that these variables are binary in the solution. To demonstrate with examples how can this be possible, let $x_{\text{ass},1}, x_{\text{ass},1} \in \{0, 1\}^2$ be two binary assignment variables and let $x_{\text{sup},1}$ be a surplus variable that is minimized in the objective and constrained as:
\[ 0 \leq x_{\text{sup},1} \leq 1, \quad x_{\text{ass},1} + x_{\text{ass},2} \leq x_{\text{sup},1} + 1 \]
Consequently, since $x_{\text{sup},1}$ is minimized in the objective, the optimal solution $\hat{x}_{\text{sup},1}$ will be either $\hat{x}_{\text{sup},1} = 0$ (when $\hat{x}_{\text{ass},1} = 1$ or $\hat{x}_{\text{ass},2} = 1$) or $\hat{x}_{\text{sup},1} = 1$ (when both $\hat{x}_{\text{ass},1} = 1$ and $\hat{x}_{\text{ass},2} = 1$). As a result, a binary constraint on $x_{\text{sup},1}$ is unnecessary and can be relaxed using the convex box constraint.

To generalize this simple example, if $c_{\text{sup}} > 0$ in (1), where $c = [c_{\text{ass}}^T c_{\text{sup}}^T c_{\text{aux}}^T]^T$, where $c_{\text{ass}}, c_{\text{sup}}$ and $c_{\text{aux}}$ are subvectors of $c$ associated with $x_{\text{ass}}, x_{\text{sup}}$ and $x_{\text{aux}}$, respectively and the linear inequality constraints in (1) include inequalities of the form:
\[ f_i(x_{\text{ass}}, x_{\text{aux}}) \leq x_{\text{sup},i}, \quad i = 1, \cdots, n_{\text{sup}} \quad (3) \]
where, $f_i(x_{\text{ass}}, x_{\text{aux}}) \in \{0, 1\}$, then $x_{\text{sup}} \in \{0, 1\}^{n_{sup}}$ can be relaxed with $0 \leq x_{\text{sup}} \leq 1$ in (1). In other words, when the range of $f_i(x_{\text{ass}}, x_{\text{aux}})$ is binary, then $x_{\text{sup},i}$ do not require binary constraints. Inequalities of the form (3) are often used to model soft constraints in timetabling problems as demonstrated in Appendix A in this study for the general graph coloring problem and in Appendix B for the university examination timetabling problem.

As another example, consider four binary assignment variables $x_{\text{ass},1}, x_{\text{ass},2}, x_{\text{ass},3}$ and $x_{\text{ass},4}$ and let $0 \leq x_{\text{aux},1} \leq 1$ be an auxiliary variable with the following constraints:
\[ x_{\text{ass},1} + x_{\text{ass},2} \leq x_{\text{aux},1}, \quad x_{\text{ass},1} + x_{\text{ass},2} \leq 1 \]
\[ x_{\text{ass},3} + x_{\text{ass},4} \leq x_{\text{aux},1}, \quad x_{\text{ass},3} + x_{\text{ass},4} \leq 1 \]
\[ x_{\text{aux},1} \leq x_{\text{ass},1} + x_{\text{ass},2} + x_{\text{ass},3} + x_{\text{ass},4} \]

Then the optimal solution $\hat{x}_{\text{aux},1}$ will be either $\hat{x}_{\text{aux},1} = 0$ (when all four assignment variables are zero) or $\hat{x}_{\text{aux},1} = 1$ (when any single or multiple assignment variables are 1). Hence, there is no effect of relaxing the binary constraint on $x_{\text{aux},1}$ on the final optimal solution. To generalize this example, if the linear inequality constraints in (1) include inequalities of the form:
\[ h_j(x_{\text{ass}}, x_{\text{sup}}) \leq x_{\text{aux},i}, \quad j = 1, \cdots, n_{\text{aux}} \]
\[ x_{\text{aux},i} \leq \sum_{j} h_j(x_{\text{ass}}, x_{\text{aux}}), \quad i = 1, \cdots, n_{\text{aux}} \quad (4) \]
where, $h_j(x_{\text{ass}}, x_{\text{sup}}) \in \{0, 1\}$, then $x_{\text{aux}} \in \{0, 1\}^{n_{aux}}$ can be relaxed with $0 \leq x_{\text{aux}} \leq 1$ in (1). Inequalities of the form (4) are often used to model auxiliary binary variables that capture special timetabling scenarios. The above two examples for linear inequalities are two of possibly many other conditions that allow relaxation of the binary constraints on the surplus and auxiliary variables.

Following assumptions 1, 2 and 3, problem (1) can now be reformulated as a cardinality constrained problem as:
\[
\begin{align*}
\min \quad c^T x \\
\text{s.t.} \quad Ax &= b, \quad Dx \leq g, \quad x = [x_{\text{ass}}^T x_{\text{sup}}^T x_{\text{aux}}^T]^T \\
\|x_{\text{ass}}\|_0 &= n_s, \quad 0 \leq x \leq 1
\end{align*}
\]
(5)

To simplify presentation, the convex box constraints were written for the entire vector $x$, although convex box constraints on assignment variables are unnecessary following assumptions 1 and 2. In the appendix A, we demonstrate the
applicability of the above mentioned assumptions to the general graph coloring problem that is commonly used as a basis for modeling timetabling problems. We also demonstrate the applicability of the above assumptions on the university examination timetabling problem in detail in appendix B.

III. ALGORITHM DESIGN

At this point, we would like to focus our attention on solving (5) instead of (1). However, the constant cardinality constraint in (5) is non-convex and known to be NP-hard [33]. Dattorro in [34] proposed a convex optimization approach to cardinality-constrained feasibility problems of the form:

$$\text{find } x \quad \text{such that } Ax = b, \quad \|x\|_0 = n_s, \quad x \geq 0$$

(6)

It was suggested in [34] that cardinality constraint problems, such as (6), be solved through iteration of a sequence of linear programs as follows:

$$\hat{x} = \arg \min_x \hat{y}^T x \quad \text{s.t. } Ax = b, \quad x \geq 0 \quad \quad (7a)$$

$$\hat{y} = \arg \min_y \hat{y}^T \hat{x} \quad \text{s.t. } 0 \leq y \leq 1, \quad \sum_i y_i = n - n_s \quad \quad (7b)$$

Here, the direction vector $y$ is calculated in (7b) such that its elements are equal to one corresponding to the smallest $n - n_s$ elements of $\hat{x}$ and zero for the remaining entries. Ultimately, when $\hat{y}^T \hat{x}$ vanishes to zero after several iterations of the sequence (7a) and (7b), then $n - n_s$ entries of $\hat{x}$ will sum to zero, which is only possible when these entries are themselves zero since the elements of $\hat{x}$ must be positive. In other words, $\|\hat{x}\|_0 = n_s$ when $\hat{y}^T \hat{x}$ converges to zero.

Finding $y$ using repeated iteration of (7a) and (7b) provides choice of direction generally better than the $\ell_1$ norm as discussed in [34]. However, global optimality can not be guaranteed and the method can experience stalling at a local minimum due to active constraints blocking further reduction in $\hat{y}^T x$ [34].

To overcome stalling, Dattorro in [34] proposed to use a heuristic escape technique for the sequence in (7a) and (7b) that manipulates $y$ randomly until the penalty $\hat{y}^T \hat{x}$ in (7a) converges to zero. However, with the use of heuristics there is no guarantee when the method will escape stalling and when will the solution of (7a) and (7b) converge to zero.

An extension to the technique given in [34] to our problem in (5) will result in the repeated iterations of the following two convex minimization problems:

$$\hat{x} = \arg \min_x \hat{w}^T x \quad \text{s.t. } Ax = b, \quad D x \leq g, \quad 0 \leq x \leq 1 \quad \quad (8a)$$

$$\hat{y} = \arg \min_y \hat{y}^T \hat{x} \quad \text{s.t. } 0 \leq y \leq 1, \quad \sum_i y_{ass,i} = n_{ass} - n_s \quad \quad (8b)$$

where, $w = c + \hat{y}$ and $y = [y_{ass}^T, y_{sup}^T, y_{aux}^T]^T$. Here, $y_{ass}$ are the weights corresponding to $x_{ass}$ and so on. Consequently, when $\hat{y}_{ass} x_{ass} = 0$ then $\|x_{ass}\|_0 = n_s$ and $x_{ass}$ will be binary following assumption 1 and 2. Note, that the solution of (8b) will result into $\hat{y}_{sup} = 0$ and $\hat{y}_{aux} = 0$ since these variables are unconstrained. Consequently, when $\hat{y}_{ass} x_{ass} = 0$ then $\hat{y}^T \hat{x} = 0$ as well. Note also that the objective $c^T x$ was included in the cardinality constrained problem and it may be removed and replaced by an upper bound on the objective as will be discussed later.

Our experiments using the iterations of (8a) and (8b) for solving timetabling problems resulted in stalling at a local minimum due to one or more of the inequality constraints $D x \leq g$ becoming active and hence the above technique will generally fail. This leads us to the following proposed solution to overcome stalling of the technique.

A. PROPOSED SOLUTION

In this study, we propose relaxing the inequality constraints to prevent stalling of the sequence (8a) and (8b) at a local minimum. The following is the relaxation used for the linear inequality constraints in (5):

$$g = Dx* + d^* x + \zeta$$

(9)

where, $x^*$ represents the optimal solution of (5) (assuming the existence of a unique optimal), $d^* \in \mathbb{R}^n_+$ is a strictly positive scalar representing the average slack value at the optimal solution $x^*$ and $1$ is a vector of ones of dimension $n_{ineq}$. The vector $\zeta \in \mathbb{R}^{n_{ineq}}$ is a vector that captures any positive or negative deviation from the average slack value. Consequently, we may view vector $g$ as a vector of “noisy measurements” with an unknown bias $d^*$ and deterministic noise $\zeta$. Although $g$ is not related to any real measurements, we describe it here as noisy measurements only to relate to sparse recovery estimation techniques that are typically described in the context of estimating a sparse vector from available noisy measurements [32].

Using the terminology used in sparse inference techniques, we may now view the timetabling problem in (1) as follows: given the linear measurements $b$ and $g$ and the measurement matrices $A$ and $D$, we seek to find an estimate of the binary sparse vector $x^*$ and the mean slack $d^*$. Based on the above discussion and assuming that assumptions 1-3 are satisfied, we propose solving the following sequence of convex programs until convergence of $\hat{y}^T \hat{x}$ to zero for estimating $x^*$:

$$\hat{x}, \hat{d} \in \arg \min_{x, d} \frac{1}{2h_{min}}\|g - D x - d^* \hat{x} - \lambda w^T x\|_2^2 \quad \text{s.t. } Ax = b, \quad 0 \leq x \leq 1, \quad \hat{d} \geq 0, \quad \quad (10a)$$

$$\hat{y} = \arg \min_y \hat{y}^T \hat{x} \quad \text{s.t. } 0 \leq y \leq 1, \quad \sum_i y_{ass,i} = n_{ass} - n_s \quad \quad (10b)$$

where, $w = c + \hat{y}$ as before. The following is a summary of the rational behind this proposed algorithm for solving (5):
1) The additional penalty $\hat{y}^T x$ in (10a) was introduced to ensure that the cardinality constraint $|\text{ass}|_0 = n_s$ is satisfied upon convergence of $\hat{y}^T \hat{x}$ to zero. Upon vanishing of $\hat{y}^T \hat{x}$, the solution $\hat{x}$ will essentially be binary following assumptions 1, 2 and 3.

2) The linear equality constraints are necessary for including the uniqueness constraints to convert the problem into a cardinality constrained problem as required by assumption 2.

3) The Lasso minimizes the uncertainty term $\zeta$ in the least squares sense which helps to satisfy the inequality constraints $Dx \leq g$ since it will minimize deviation from the average positive slack $d_c$. This will also help to avoid stalling of the technique, as discussed earlier, since all inequality constraints are essentially relaxed.

4) The tuning parameter $\lambda$ helps to trade-off optimality of the objective with constraint satisfaction. In other words, the higher the value of $\lambda$, the more emphasis the technique gives to minimizing the objective and the less emphasis to satisfying the constraints.

5) The box constraints $0 \leq x \leq 1$ are needed to ensure that the surplus and auxiliary decision variables are binary in the solution following assumption 3.

6) The parameter $d_c$ was set positive to ensure estimation of a positive mean slack for all inequalities.

As suggested in [34], if $n_s$ is unknown, we may substitute the constraint $\sum_i y_{\text{ass},i} = n_{\text{ass}} - n_s$ in (10b) with $\sum_i y_{\text{ass},i} = n_{\text{ass}} - k$, where $k$ is an index that increases in value until convergence of the term $\hat{y}^T \hat{x}$ to zero. In this case, $\hat{y}$ will decrease in cardinality in every iteration by one unit. The implicit cardinality constraint will be satisfied as soon as $\hat{y}^T \hat{x}$ vanishes to zero.

Our experiments in Section V demonstrate that this incremental technique (i.e. using $\sum_i y_{\text{ass},i} = n_{\text{ass}} - k$ instead of $\sum_i y_{\text{ass},i} = n_{\text{ass}} - n_s$ in (10b), where $k$ is an integer that increases in value in each iteration) also helps to obtain better solutions with less constraint violations. The number of iterations required in this case are $n_s$ iterations; i.e. the number of events to be scheduled or a multiple of this number. Loosely speaking, a zero is added in one of the elements of the vector $y_{\text{ass}}$ in each iteration of (10b) that subsequently relieves one of the suppressed assignment variables in $x_{\text{ass}}$ in the next iteration. Algorithm 1 is a summary of the proposed method in this study for solving timetabling problems of the form given in (1) after adapting this change.

Ideally, Algorithm 1 will converge to a binary solution that satisfies all linear equality and inequality constraints and the solution will be suboptimal with respect to (1). The technique cannot, in general, guarantee global optimality even when all constraints are satisfied. However, under certain assumptions, the error can be bounded as described in Section IV.

When the solution does not satisfy all the important operational constraints, then the following additional measures can be made to minimize constraint violations:

1) The objective can be converted to a soft constraint with a user defined upper bound on the objective as

Algorithm 1

**Given:** $A$, $b$, $D$, $g$, $n_{\text{ass}}$, $n_{\text{in}}$, $\lambda > 0$

**Initialization:** $\hat{y} \leftarrow 1$, $k = 0$.

**repeat**

**solve:**

$$\hat{x}, \hat{d}_c = \arg \min \limits_{x,d_c} \frac{1}{2n_{\text{in}}} \|g - Dx - d_c1\|_2^2 + \lambda \hat{x}^T x$$

s.t. $Ax = b$, $0 \leq x \leq 1$, $d_c \geq 0$  \hspace{1cm} (11)

**update:**

$$\hat{y} = \arg \min \limits_{y} \hat{y}^T \hat{x}$$

s.t. $0 \leq y \leq 1$, $\sum_i y_{\text{ass},i} = n_{\text{ass}} - k$  \hspace{1cm} (12)

$k \leftarrow k + 1$

until $\hat{y}^T \hat{x} \approx 0$

**follows:**

$$c^T x \leq z_{\text{min}} \rightarrow z_{\text{min}} = c^T x^* + d^* + \zeta$$

where, $z_{\text{min}}$ is the user defined upper bound for the objective and $\zeta$ is the deviation from the estimated value. This additional “noisy measurement” can then be added to be part of the inequalities in (9) and will be part of the soft constraints that we seek to satisfy in the least squares objective of (11).

2) Estimating multiple mean slack for multiple groups of inequality constraints. As an example, for the case of two groups of inequality constraints we will have two equations of the form:

$$g_1 = D_1 x^* + d_{c,1}^* 1 + \zeta_1, \quad g_2 = D_2 x^* + d_{c,2}^* 1 + \zeta_2$$

where, $g_1$ and $D_1$ are the parameters related to the first set of inequalities ($D_1 x \leq g_1$) and $g_2, D_2$ are the parameters related to the second group of inequalities ($D_2 x \leq g_2$). In this case, there will be two different slack means $d_{c,1}$ and $d_{c,2}$ to be estimated instead of one. Estimating multiple slack means helps to reduce the variance of the uncertainty terms $\zeta_1$ and $\zeta_2$. Consequently, for two groups of inequality constraints, the least squares penalty in (11) can be replaced by:

$$\frac{1}{2n_{\text{in}}} \left[ \frac{y_1(g_1 - D_1 x_1 - d_{c,1} 1_{\text{nun}})}{y_2(g_2 - D_2 x_2 - d_{c,2} 1_{\text{nun}})} \right]_2^2$$

where, $1_{\text{nun},1}$ and $1_{\text{nun},2}$ are vectors with ones of dimensions corresponding to the number of inequalities associated with the first group and second group, respectively and $y_1$ and $y_2$ are scaling factors to make the different penalties of equal weight in the objective. Another idea is to predetermine the average slack value for each group of inequalities from problem data and use it in the formulation instead of estimating it.
The effect of the two additional measures in reducing constraint violations will be demonstrated in the simulation section later in this study.

B. REDUCING THE SIZE OF LARGE PROBLEMS

For large timetabling problems, solving large Lasso iterations repeatedly can be computationally expensive. For reducing the size of individual Lasso iterations we propose eliminating constraints that are satisfied and retaining constraints that are violated in each iteration of Algorithm 1. We select the constraints with greater than a predefined threshold $s_{th}$ and assume that these constraints will remain satisfied in the next iteration allowing for elimination of these constraints in the next iteration. All constraints are reexamined and evaluated for elimination, retaining and restoring in each iteration. Algorithm 2 shows the proposed method.

As a result, the size of the minimization problem in (11) is reduced by reducing the set of constraints to be satisfied in the least squares penalty. The index of constraints is updated in every iteration as shown in Algorithm 2. Our experiments in Section V (Experiment 1c) demonstrate that this simple technique helps to reduce computation time at the expense of violating slightly more constraints compared to Algorithm 1.

**Algorithm 2**

**Given:** $A, b, D, g, n_{ass}, n_{in}, \lambda > 0, s_{th}$

**Initialization:** $\hat{y} \leftarrow 1, \text{ind} = \{1, \ldots, n_{in}\}, k = 1$

repeat

solve:

\[
\hat{x}, \hat{d}_c = \text{arg min}_{x, d_c} \frac{1}{2n_{in}} \sum_{i \in \text{ind}} (g_i - D_i^\top x - d_{c_i})^2 + \lambda \vert \hat{x} \vert
\]

s.t. $Ax = b$

\[
0 \leq x \leq 1, \quad d_c \geq 0, \quad w = c + \hat{y}
\]

update:

\[
\text{ind} = \{i | D_i^\top \hat{x} > s_{th}, i = 1, \ldots, n_{in} \}
\]

\[
\hat{y} = \text{arg min}_{y} \hat{y}^\top \hat{x}
\]

s.t. $0 \leq y \leq 1, \quad \sum_i (y_{ass})_i = n_{ass} - k$

$k \leftarrow k + 1$

until $\hat{y}^\top \hat{x} \approx 0$

**IV. ESTIMATION PERFORMANCE AT CONVERGENCE**

In the following, we are interested in analyzing the performance of the method described in (11) and (12) for solving (1) when $\hat{y}^\top \hat{x}$ converges to zero. We will view problem (11) as an estimation problem where we need to estimate $x^*$ and the associated average slack $d_c^*$ from the given “noiseless measurements” $Ax^* = b$ in (1) and “noisy measurements” $g = Dx^* + d_c^* 1 + \zeta$ defined in (9). We will make use of the following additional assumption.

**Assumption 4:** The binary linear program (1) has a unique global optimal solution $x^*$.

Our approach for establishing an error bound will make use of methods developed in [35]. First, to eliminate the equality constraints in (11) we need the following substitutions:

\[
\begin{align*}
Ax &= b \quad \rightarrow \quad x = A^\dagger b + N_A \alpha \\
g &= Dx + d_c 1 + \zeta = DN_A \alpha + d_c 1 + DA^\dagger b + \zeta
\end{align*}
\]

Here, $A^\dagger$ is the Moore-Penrose pseudoinverse of matrix $A$, $N_A$ is a matrix with columns spanning the null space of $A$ and $\alpha$ is the new decision variable vector that lives in the null space of $A$. Consequently, (11) can be written as:

\[
\begin{align*}
\min_{\beta} & \frac{1}{2n_{in}} \| \tilde{g} - X\beta \|_2 + \lambda \tilde{w}^\top \beta \\
\text{s.t.} & \quad 0 \leq A^\dagger b + N_A \alpha \leq 1, \quad d_c \geq 0 \\
& \text{where,} \quad \beta = \begin{bmatrix} N_A \alpha \\ d_c \end{bmatrix}, \quad \tilde{w} = \begin{bmatrix} \hat{y} + c \\ 0 \end{bmatrix}
\end{align*}
\]

\[
\tilde{g} = g - DA^\dagger b, \quad X = \begin{bmatrix} D & 1 \end{bmatrix}
\]

Here, we eliminated the constant term $\tilde{y}^\top A^\dagger b$ since it will not effect the solution. We approach for performance evaluation will be bounding the $\ell_1$ norm error of the difference $\hat{\beta} - \beta^*$, where $\beta^* = (N_A \alpha^*)^T d_c^*$ and $d_c^*$ is the true mean of all inequality constraints in (1). A small $\ell_1$ norm error will indicate the recovery of the unique optimal timetable solution $x^*$ determined from the solution of the integer linear program in (1). We first present an important assumption required to bound the error.

**Assumption 5 (Restricted Strong Convexity):** The regression matrix defined by $X = \begin{bmatrix} D & 1 \end{bmatrix}$ satisfies the following:

\[
\frac{1}{2n_{in}} \| Xv \|_2^2 \geq \kappa^* \| v \|_1, \quad \kappa^* > 0, \quad \forall v \in C, \quad \text{where,}
\]

\[
C = \{ v | v_{S^*} \in (-1, 0)^{n_S^*}, v_{S_c^*} \in \{0, 1\}^{n_{S_c^*}} \}
\]

where, $S^*$ and $S_c^*$ are the true and complement supports of $x^*$ and the vectors $v_{S^*}$ and $v_{S_c^*}$ are the subvectors of $v$ with elements corresponding to the support index $S^*$ and $S_c^*$, respectively.

We note that the above assumption can only be verified if the true support $S^*$ and the complement support $S_c^*$ are known in advance. Since this information essentially is the solution to the problem, normally $S^*$ and $S_c^*$ are unknown. However, this assumption is needed to bound the error of the technique as given in the following theorem.

**Theorem 6:** Suppose that $\tilde{g} = X\beta^* + \zeta$ and $\hat{y}^\top \hat{x} = 0$ upon convergence of Algorithm 1. Suppose also that assumptions 4 and 5 are satisfied for some $\kappa^* > 0$ and $\lambda \geq \frac{1}{2n_{in}} \| X^T c \|_\infty$, then the following error bound applies:

\[
\| \hat{\beta} - \beta^* \|_1 \leq \frac{2\kappa^*}{\kappa^*}
\]

**Proof:** See Appendix C.

**Theorem 6** is applicable for both low dimensional settings, when $n_{in} \gg n$ and high dimensional settings; i.e. when $n \gg n_{in}$. The result demonstrates that the error can be made
small, for example, by (1) increasing the value of $\kappa^*$, which is imposed by the problem itself and (2) reducing the magnitude of projected “noise term” $\|X^T \zeta\|_\infty$ which is possible when the variance of the slack in the solution is small. This variance can be made small if the slack for all groups of constraints is approximately known.

V. SIMULATION STUDY
Three experiments were conducted for testing the technique. The first experiment is for a relatively small artificial examination timetabling problem to demonstrate the efficiency of the new technique in comparison with integer linear programming solutions at the expense of violating a small percentage of constraints. The second simulation experiment solves the same problem with different values of the tuning parameter $\lambda$ to analyze the sensitivity of the solution to changes in this parameter. The third simulation experiment solved a benchmark problem presented in the second international timetabling problem in [3] demonstrating improved value of the objective compared to documented results at the expense of violating a very small percentage of operational constraints. All experiments also demonstrate predictive convergence of the technique; i.e. the number of Lasso iterations required to reach a solution corresponds to the number of events to be scheduled or a multiple of this number.

A. FIRST EXPERIMENT
The experiment involves 50 exams, 7 rooms, 20 periods and 3581 students. The model used is described in appendix B and given in [3]. The conflict set $A_C$ contained 154 pair of exams. The parameters of the model are identical to the parameters given in [3] for the first benchmark dataset. The model includes room capacity constraints, period constraints, student conflict constraints and has all other model constraints except for the non-mixed duration soft constraints, the coincidence constraints and the exclusion constraints for simplicity.

The matrices $A$, $D$ and vectors $b$ and $g$ were formed from the equations and inequalities of the model using Matlab CVX [36]. The resulting number of binary decision variables were $n = 8812$ and the number of inequalities were $n_{in} = 34821$. We split the inequality constraints into two groups. The first group contained all the hard constraints except the student conflict constraints (item 3 in the hard constraints of the model in appendix B) while the second group contained the student conflict constraints and all model soft constraints.

Algorithm 1 was then implemented iteratively using Matlab CVX [36] with $\lambda = 1$ and with the least squares objective (13) using $\gamma_1 = 1$ for the first group of inequalities and $\gamma_2 = 50$ for the second group to approximately make the least squares penalties of equal weight. We first solved this problem with no objective by setting the vector $c$ to be zero (Experiment 1a). The solution was obtained after 132 Lasso iterations which took 252 seconds on a Mac-mini with 6 cores Intel 5 processor with 6GB Ram. The obtained solution satisfied all constraints, including the binary constraints on all variables, which demonstrates that the technique can provide feasible solutions when no objective is used. The value of the first bias was $\hat{d}_{c,1} = 0.8892$ and the second bias was $\hat{d}_{c,2} = 190$. However, the objective value for this case was relatively large as shown in Table 1.

We then repeated the experiment using the objective function $c^T x$ (Experiment 1b) and obtained a solution after 132 Lasso iterations that took 245 seconds. The objective obtained was zero but with 4 students having conflict exams and 3 exams exceeding room capacity constraints. The value of the first bias was $\hat{d}_{c,1} = 190$ and the second bias was $\hat{d}_{c,2} = 0.89$. All remaining constraints were satisfied, including the binary constraints on variables.

For comparison, we repeated the experiment using Algorithm 2 using a threshold value $s_{th} = 0.1$ (Experiment 1c). The solution was obtained after 85.6 seconds with an objective of zero as before showing 66% reduction in computation time. The solution however, had 34 more additional students with conflict exams representing 0.9% of all students. Hence, this experiment shows that the solution time can be improved significantly using Algorithm 2 at the expense of violating more constraints (0.8% more constraints were violated in this experiment).

Finally, we solved the integer linear program (1) using Mosek integer programming solver version 8.0.0.6 in Matlab CVX [36] (Experiment 1d). The best value of the objective function obtained was 90 after 14 minutes with all constraints satisfied. No additional improvement was observed in the objective after running the program for several hours. Hence, this experiment shows that our proposed technique can provide efficient solutions with improved optimality at the expense of violating a small percentage of constraints. Table 1 gives a summary of the results for the above mentioned experiments showing the execution time, the value of the objective evaluated at the solution, the percentage of constraints violated and the number of Lasso iterations required to reach the final solution.

B. SECOND EXPERIMENT
The objective of this experiment is to test the sensitivity of the technique to changes in the tuning parameter $\lambda$ used in Algorithm 1 and the effect of replacing the objective with an upper bound constraint. In this experiment, the same model described in experiment 1 was used with the same experimental setup, including the splitting of the inequality constraints into two groups.

| Exp# | time (sec) | Objective | % Violation | Iterations |
|------|------------|-----------|-------------|------------|
| 1a   | 252        | 1166      | 0%          | 132        |
| 1b   | 245        | 0         | 0.22%       | 132        |
| 1c   | 85.6       | 0         | 0.46%       | 132        |
| 1d (ILP) | 840 | 90       | 0%          | N/A        |
A grid of 100 values for $\lambda$ between $\lambda_{\text{max}} = 1000$ and $\lambda_{\text{min}} = 0.1$ generated using Matlab’s “logspace” function was used in each experiment. Algorithm 1 was then solved iteratively using Matlab CVX [36] until convergence for each value of $\lambda$ in the grid. In each run we recorded the solution, the number of constraints violated, the number of Lasso iterations to reach convergence and the value of the objective function. The results are shown in Figure 2 (Experiment 2a). All obtained solutions between $0.5337 \leq \lambda \leq 1000$ satisfied the binary constraint on decision variables, whereas the solutions obtained for $\lambda < 0.5337$ did not satisfy the binary constraint which shows that $\lambda$ must be greater than a threshold value to obtain a binary solution upon convergence. Figure 2 shows, from left, the objective value (left plot), the number of students with conflict exams (second plot) the number of exams with assigned rooms exceeding room capacity (third plot) and the number of Lasso iterations before convergence and the value of the objective function.

The results are shown in Figure 2 (Experiment 2a). All obtained solutions between $0.5337 \leq \lambda \leq 1000$ satisfied the binary constraint on decision variables, whereas the solutions obtained for $\lambda < 0.5337$ did not satisfy the binary constraint which shows that $\lambda$ must be greater than a threshold value to obtain a binary solution upon convergence. Figure 2 shows, from left, the objective value (left plot), the number of students with conflict exams (second plot) the number of exams with assigned rooms exceeding room capacity (third plot) and the number of Lasso iterations before convergence (forth plot) all versus the value of the tuning parameter $\lambda$ used. There were no duration constraints violated in all experiments. The figure demonstrates the trade-off effect introduced by increasing the tuning parameter $\lambda$ in decreasing the value of the objective for $\lambda \leq 0.5337$ only, when the binary constraints were not satisfied. On the other hand, for the remaining range of values for $\lambda$, the method overall was slightly sensitive to changes in the value of $\lambda$ in both the objective function and the number of constraints violated. More interestingly, the number of Lasso iterations was exactly 132 Lasso iterations in all experiments.

For the second experiment (Experiment 2b), we used an upper bound for the objective as 10; i.e. $z_{\text{min}} = 10$ as a measure to reduce constraint violations. Consequently, Algorithm 2 was implemented using the same range of values of $\lambda$ as before. The results are shown in Figure 3. The experiment demonstrated again that binary solutions were obtained only when $\lambda > 0.5337$. Moreover, the experiments also demonstrated that relaxing the objective helped to reduce the number of constraint violations for both conflict constraints and room capacity constraints. However, the experiment again demonstrated that the technique is slightly sensitive to changes in the tuning parameter $\lambda$ beyond the threshold of $\lambda = 0.5337$. The number of Lasso iterations in all experiments was exactly 132 iterations as before. The best solution obtained had an objective value of 9.69 with 1 student having a conflict exam and one exam exceeding a room capacity constraint using a value of $\lambda = 1.3219$.

**C. THIRD EXPERIMENT**

The third experiment solved a benchmark examination problem appearing in the 2007 international timetabling
The problem involves 607 exams, 7 rooms, 54 periods and 7883 students. The complete model description is given in Appendix B. The conflict set $A_C$ contained 9277 pair of exams. The parameters of the model are identical to the parameters given in [3] for the first dataset. All constraints of the original model were considered.

The matrices $A$, $D$ and vectors $b$ and $g$ were formed from all model equations and inequalities using Matlab CVX [36]. The number of binary decision variables for this problem was $n = 300, 352$ and the number of inequalities $n_{in} = 5, 958, 430$. We split the inequality constraints into three groups. The first group contained all the room capacity and period constraints; the second group contained the student conflict constraints while the third group contained all the remaining soft constraints of the model as described in Appendix B.

For the first test (Experiment 3a) we introduced the objective $e^T x$ directly in the problem. Algorithm 2 was implemented using Matlab CVX [36] with $\lambda = 1$ and $S_{th} = 0.1$. In relation to equation (13), the individual scaling of the groups was $\gamma_1 = 1$, $\gamma_2 = 10$ and $\gamma_3 = 10$. The result of this implementation is shown in Figure 4 as image plots showing the index of the exam versus the assigned period (first image) the index of the exam vs. the assigned room number (second image) and the index of the period versus the room showing the number of additional students in a room (third image). The solution was obtained in 12.7 hours using a Mac-mini with 6 Cores Intel 5 processor with 6GB Ram. The number of iterations required for $\hat{y}^T \hat{x}$ to converge to zero was equal to 1821 iterations ($3 \times 607$); i.e. the number of exams to be scheduled multiplied by 3 since there are 2 additional auxiliary assignment variables for each main assignment variable. The value of the first bias was $\hat{d}_{c,1} = 205.7$, the second bias was $\hat{d}_{c,2} = 0$ and the third bias $\hat{d}_{c,3} = 0$ also.

The solution satisfied all equality and inequality constraints except for one room capacity constraint and 1381 students with conflict exams representing 17.5% of all students as shown in Figure 2. The single room capacity constraint violation required 24 additional students in a single exam held in a single room. The value of the objective function was equal to 0. This experiment shows ignoring student conflicts the lowest possible objective of zero can be obtained using the proposed technique.

For the second test (Experiment 3b), we used an upper bound for the objective as 3000; i.e. $z_{min} = 3000$ as a measure to reduce constraint violations. Consequently, Algorithm 2
was implemented using Matlab CVX [36] with $\lambda = 1$ and $s_h = 0.1$ as before. The result of this implementation is shown in Figure 3. The solution was obtained in 19.6 hours. The value of the first bias was $d_{c,1} = 205.75$, the second bias was $d_{c,2} = 0$ and the third bias $d_{c,3} = 0$ also.

The solution satisfied all equality and inequality constraints except for one room capacity constraint and 24 students with conflict exams representing 0.3% of all students as shown in Figure 5. The single room capacity constraint violation required 33 additional students in a single exam held in a single room. The value of the objective function was equal to 3020 which is 31% less than the best documented result in the 2007 international timetabling competition at the expense of violating a very small percentage of constraints (0.0928 % of all constraints). This experiment demonstrates that the technique improves optimality significantly compared to other solution techniques at the expense of violating a small percentage of constraints. Most importantly, the convergence of the technique is predictable; i.e. equal to a multiplication factor of the number of events to be scheduled. The experiment also demonstrates the importance of relaxing the objective if constraint satisfaction is important.

It is worth noting that we could not solve the ILP using MOSEK solver version 8.0 when using a XEON processor IMAC pro 2018 computer with 32Gb, as the computer ran out of memory.

VI. CONCLUSION

A new convex optimization framework for solving automated timetabling problems described as binary linear programs is presented. The method is based on converting the binary linear program into a cardinality constrained problem while relaxing the operational constraints. A weighted Lasso is repeatedly solved with weights that are updated using a linear program to direct the solution towards a binary solution that satisfies the cardinality constraint. The number of Lasso iterations required is approximately equal to the number of events to be timetabled or a multiple factor of this number. Required assumptions were established and an error bound using restricted strong convexity was derived. In summary, the advantages of the proposed technique are as follows:

1) The technique can provide efficient solutions for timetabling problems compared with integer linear programming methods at the expense of violating a small percentage of constraints as demonstrated in Experiment 1d.

2) The convergence of the technique is predictable since the number of convex programming iterations required for convergence of Algorithm 1 are equal to $n_s$ or a factor of this number depending on the existence of auxiliary assignment variables in the model. This has been verified in all the simulation experiments conducted. It is worth noting here that the number of Lasso iterations could have been reduced to correspond exactly to the number of events to be scheduled using minor adjustments to the algorithm in 2 considering the increments in the number of nonzero elements of the weighting vector.

3) The method can trade off optimality with constraint satisfaction using a soft upper bound constraint on the objective. Our simulation experiments in Section V demonstrated that the method can provide solutions satisfying all constraints when no objective is used.

4) Upon convergence of the technique a binary solution will be obtained. Hence, no additional randomized rounding steps are required that could be detrimental to the solution as compared to previous convex relaxations, including Lagrange dual techniques and semidefinite programming.

5) Finally, compared to both integer programming techniques and semidefinite program relaxations, the method can solve very large timetabling problems since only a predictable number of Lasso iterations are required. This is possible through efficient Lasso solvers and other techniques including distributed optimization using, for example, the alternating direction method of multipliers as discussed in [37].

Although the technique was only demonstrated on examination timetabling problems, the problem assumptions were verified on the general graph coloring problem which is commonly used as a basis for formulating timetabling problems. The method can be used to solve timetabling problems that comply with assumptions 1-3, including, for example, the university course timetabling problem. Extension to the technique can be made to other timetabling problems that are modeled using convex objectives and constraints. Also, efficiency of the technique can be improved using specialized Lasso solvers and distributed optimization. Possible future studies will address ways for finding the best regularization parameter $\lambda$ and ways for improving the technique to satisfy all constraints, including the effect of predetermining the average slack for all constraints on the solution and adaptive adjustment of the number of nonzero elements of the weighting vector so that all constraints are satisfied in each iteration.

APPENDIX A GRAPH COLORING FORMULATION

Graph coloring problems can be used as a basis for describing the university timetabling problem, examination timetabling problem, frequency assignment in cellular networks and registry allocation in compilers [38]. In the following, we will demonstrate applicability of assumptions 1, 2 and 3 to the general integer programming formulation of graph coloring problems with soft constraints.

The graph coloring problem can be described as follows: Given an undirected graph $G = (V, E)$, that may or may not be connected, assign $|K|$ colors to vertices $v \in V$ such that no two connected vertices $(u, v) \in E$ are assigned the same color. Furthermore, the same color can not be repeated more than $r$ times. Here, $V$ is the set of vertices, $E$ is the set of connected pair of vertices and $K$ is the set of colors. The
objective is to find a colored graph that satisfies the above requirements while minimizing deviation from additional desired properties. The following standard ILP formulation for general graph coloring can be used:

\[
\min \sum_{i=1}^{|V|} \sum_{c=1}^{|K|} w_{vc}x_{vc} + \sum_{c \in K, (u,v) \in \tilde{E}} \tilde{w}_{uv} s_{uv} \tag{17}
\]

subject to Uniqueness constraints: \(\sum_{c=1}^{|K|} x_{vc} = 1\), \(\forall v \in V\) (18)

Hard constraints: \(x_{vc} + x_{vc} \leq 1\), \(\forall c \in K, \forall (u,v) \in E\) (19)

\(|V|\) \(\sum_{c=1}^{K} x_{vc} \leq r\), \(\forall c \in K\) (20)

Soft constraints: \(x_{uc} + x_{vc} \leq 1 + s_{uv}, c \in K, \forall (u,v) \in \tilde{E}\) (21)

where, \(x_{vc} = \begin{cases} 1, & \text{if vertex } v \text{ has color } c \\ 0, & \text{otherwise} \end{cases}, s_{uv} = [0, 1]\)

The uniqueness equality constraints in (18) ensure that all vertices are colored; the hard constraints in (19) ensure that no two connected vertices have the same color while the constraints in (20) ensure that each color is repeated at most \(r\) times in the entire graph. The soft constraints in (21) reflect special coloring rules that we desire to have in the colored graph, where, \(K\) is a set of colors and \(\tilde{E}\) is a set of pairs of vertices that are different than \(K\) and \(E\), respectively. Finally, the objective in (17) is to obtain a feasible colored graph while minimizing a weighted sum of the assignment variables \(x_{vc}\), reflecting our coloring preferences and another weighted sum of surplus variables \(s_{uv}\) to minimize violations of the soft constraints, where \(\tilde{w}_{uv} > 0\).

As an example of how graph coloring problems can be used as a basis for timetabling problems we briefly describe the university course timetabling problem, as given in [39]. In this problem, vertices represent events of courses, two vertices are connected if the corresponding events are part of a single curriculum and assignment of events to periods and rooms is represented by assignment of \(|K|\) colors to \(|V|\) vertices such that connected vertices are assigned different colors and each color is used at most \(r\) times, where \(r\) represents the number of classrooms available in a single period. Other equality and inequality constraints may be present, like room capacity constraints, precedence/coincidence constraints and exclusion constraints, as explained for example in [13]. The objective is to minimize \(1\) the number of students left without a seat at an event; \(2\) the number of deviations from the desired spread of events for each course over distinct weekdays; \(3\) the number of isolated events in daily timetables of individual curricula and \(4\) the number of rooms allocated for a single course. These and similar objectives can be reflected as soft constraints (special coloring rules) of the form given in (3) with the utility of auxiliary variables defined as in (4) (Refer to inequalities 30-36 in [39] that are similar in form to (3) and (4)).

To see how the above graph coloring problem satisfies assumptions 1, 2 and 3, we can first redefine the problem variables and parameters as follows:

\[
x_{ass} = \text{vec}(x_{vc}), \quad x_{sup} = \text{vec}(s_{uv}), \quad x = \begin{bmatrix} x_{ass} \\ x_{sup} \end{bmatrix}, \quad c = \begin{bmatrix} \text{vec}(w_{vc}) \\ \text{vec}(\tilde{w}_{uv}) \end{bmatrix}
\]

where, \(x_{vc}\) represents the matrix of decision variables with \(|V|\) rows and \(|K|\) columns and \(s_{uv}\) is a 3-dimensional matrix of surplus variables. The parameter matrices \(w_{vc}\) and 3-D matrix \(\tilde{w}_{uv}\) are defined similarly. The operator \(\text{vec}(\cdot)\) concatenates all the columns of a matrix in one vector column-wise. The set of equality constraints and inequality constraints can be represented in matrix form as \(Ax = b\) and \(Dx \leq g\), respectively, after defining matrices \(A\) and \(D\) and vectors \(b\) and \(g\) appropriately.

Consequently, assumption 1 is satisfied since \(\|x_{ass}\|_0 = |V|\); i.e. the number of binary 1 elements in the vector \(x_{ass}\) should be exactly the number of vertices \(|V|\). Assumption 2 is also satisfied by the existence of \(|V|\) uniqueness equality constraints (18). Finally, assumption 3 is satisfied since the soft constraints in (21) are of the form given in (3). Hence, the graph coloring problem described above in (17)-(21) may be represented as the cardinality constraint problem in (5) without auxiliary variables using the above definitions of vectors and matrices.

For demonstration purposes, we also validated the study assumptions in detail for the university examination timetabling problem described in [3] in Appendix B, which is a practical problem that serves as an example for similar timetabling problems.

**APPENDIX B EXAMINATION TIMETABLEING MODEL**

In the context of examination timetabling, the objective is to find a feasible examination timetable that will respect all operational constraints while minimizing the inconveniences to the parties involved in conducting and taking the exams. Feasibility constraints include assigning only one exam per time period per room, while operational constraints include respecting room and time period limitations and not assigning more than one exam for a student at a certain time period. The inconveniences that sought to be minimized include, having two or more exams in a row or in the same day for a student, or in two consecutive days or within a certain period spread [3], [10]. Here, we briefly describe the integer linear program model developed in [3] and show how assumptions 1, 2 and 3 are satisfied in this model.

The formulation takes as input sets as given in Table 2. The binary decision variables of this formulation are given in Table 3 and the model parameters are as given in Table 4. We may represent the subvectors of the decision vector \(x\) as...
Hence, the cardinality of $x$ can be represented by assignment variables present. Consequently, the implicit exams since for each assignment variable two redundant linear constraints and auxiliary assignment variables are given by the following model. The relation between main assignment variables $x$ and auxiliary assignment variables introduced to facilitate modeling. The relation between main assignment variables and auxiliary assignment variables are give by the following linear constraints:

$$
x_{ass} = \begin{bmatrix}
\text{vec}(x_{ipr}) \\
\text{vec}(x_{ip}) \\
\text{vec}(x_{pr})
\end{bmatrix}, \quad x_{sup} = \begin{bmatrix}
\text{vec}(c_{2R}) \\
\text{vec}(c_{2D}) \\
\text{vec}(c_{ij}) \\
\text{vec}(n_{MD})
\end{bmatrix}
$$

where, $\text{vec}(\cdot)$ is an operator that concatenates the columns of a matrix in one vector column-wise. Here, we use boldface variables to denote matrices; e.g. $x_{ipr}$ is the matrix of variables $x_{ipr}$, $\forall i \in E$, $p \in P$ and $r \in R$ and so on. We note that the decision variables represented by $x_{ip}^P$ and $x_{ip}^R$ are auxiliary assignment variables introduced to facilitate modeling. The relation between main assignment variables and auxiliary assignment variables are given by the following linear constraints:

$$
x_{ip}^P = \sum_{r \in R} x_{ipr}, \quad \forall p \in P, \forall i \in E
$$

$$
x_{ip}^R = \sum_{p \in P} x_{ipr}, \quad \forall i \in E, \forall r \in R
$$

Hence, the cardinality of $x_{ass}$ is three times the number of exams since for each assignment variable two redundant assignment variables are present. Consequently, the implicit cardinality constraint can be represented by $\|x_{ass}\|_0 = 3n_E$, where $n_E$ is the number of exams to be timetabled.

1) **UNIQUENESS EQUALITY CONSTRAINTS**

Constraints for ensuring that all the exams are allocated to a unique period and a unique room are as follows:

$$
\sum_{p \in P} x_{ip}^P = 1, \quad \sum_{r \in R} x_{ip}^R = 1, \quad \forall i \in E
$$

Hence, assumption 2 is satisfied since the equality constraints include uniqueness constraints on the assignment variables. In this study, all equality constraints are represented in matrix-vector form as $Ax = b$.

2) **HARD INEQUALITY CONSTRAINTS**

1- Constraints for ensuring room capacities are respected:

$$
\sum_{i \in E} s_{ip}^E \leq s_{pr}^E, \quad \forall p \in P, \forall r \in R
$$

where, $s_{ip}^E$ and $s_{pr}^E$ denote the number of students sitting exam $i$ and the seating capacity of room $r$, respectively. 2- The duration hard constraints.

$$
d_{ip}^E x_{ip}^P \leq d_{ip}^P, \quad \forall i \in E, \forall p \in P
$$

where, $d_{ip}^E$ and $d_{ip}^P$ denote the duration of exam $i$ and the duration of period $j$, respectively.

3- The conflict constraints that enforce the constraint that at any period, a student will be sitting at most one exam.

$$
\sum_{i \in E} t_{is} x_{ip}^P \leq 1, \quad \forall p \in P, \forall s \in S
$$

The parameter matrix $t_{is} = 1$ if student $s$ is sitting exam $i$.

3) **SOFT CONSTRAINTS WITH SURPLUS VARIABLES**

1- Two exams in a row soft constraint:

$$
\lambda_{ip}^P + \lambda_{jq}^P \leq 1 + c_{ij}^R
$$

$\forall(i,j) \in A_C, \forall p, q \in P$ with $|p - q| = 1, y_{pq} = 1$

Here, $c_{ij}^R$ is a surplus variable that is one if a student has two exams in a row.

2- Two in a Day soft constraint:

$$
\lambda_{ip}^P + \lambda_{jq}^D \leq 1 + c_{ij}^2D
$$

$\forall(i,j) \in A_C, \forall p, q \in P$ with $|p - q| \geq 2, y_{pq} = 1$

variable $c_{ij}^2D$ is a surplus variable that is one if a student has two exams in a day.

3- Period Spread soft constraint:

$$
\lambda_{ip}^P + \lambda_{jq}^R \leq 1 + c_{ij}^PS
$$

$\forall(i,j) \in A_C, \forall p, q \in P$ with $1 \leq |q - p| \leq g^{ps}$

variable $c_{ij}^PS$ is a surplus variable that is one if a student has two exams within the period spread given by $g^{ps}$. We note that the above soft constraints are of the form given in (3) which partially verifies assumption 3 for the surplus variables. All the above inequality constraints and soft constraints can be represented in matrix-vector form as $Dx \leq g$.  

**TABLE 2. Model sets.**

| Set | Description |
|-----|-------------|
| $E$ | Set of exams |
| $R$ | Set of rooms |
| $P$ | Set of periods |
| $S$ | Set of students |
| $E^{FL}$ | Set of exams subject to front load penalty |
| $A_C$ | Set of pair of exams $(i,j)$ if there is at least one student enrolled in exams $i$ and $j$ |
| $T$ | Set of exam duration |
| $E^t$ | The set of exams with duration $t$ |

**TABLE 3. Model binary variables.**

| Variable | Description |
|----------|-------------|
| $x_{ip}^P$ | 1 if exam $i$ is assigned to period $p$ |
| $x_{ip}^R$ | 1 if exam $i$ is assigned to period $r$ |
| $x_{ipr}$ | 1 if exam $i$ is assigned to period $p$ and room $r$ |
| $c_{ipr}^{2R}$ | 1 if $2$ in a row constraint is violated $(i,j) \in A_C$ |
| $c_{ipr}^{2D}$ | 1 if $2$ in a day constraint is violated $(i,j) \in A_C$ |
| $c_{ipr}^{PS}$ | 1 for a period spread constraint is violated $(i,j) \in A_C$ |

**TABLE 4. Model parameters.**

| Parameter | Description |
|-----------|-------------|
| $w_{ij}^{CS}$ | Number of students taking the two exams $(i,j) \in A_C$ |
| $w_{ij}^{2R}$ | weight on Two in a Row penalty |
| $w_{ij}^{2D}$ | weight on Two in a Day penalty |
| $w_{ij}^{PS}$ | weight on Period Spread penalty |
| $y_{pq}$ | 1 if periods $p$ and $q$ are in the same day |
| $g_{ps}^{2R}$ | desired number of period spread |
| $s_{ip}$ | number of students sitting exam $i$ |
| $s_{pr}$ | room capacity of room $r$ |
| $g^{ps}$ | period number from which front load penalty occurs |
Finally, the objective to be minimized is a weighted sum of variables, as follows:

$$z := C^{2R} + C^{2D} + C^{PS} + C^R + C^P + C^{FL}$$

where,

$$C^{2R} = w_{22}^{2R} \sum_{(i,j) \in A} w_{ij}^C c_{ij}^{2R}$$
$$C^{2D} = w_{22}^{2D} \sum_{(i,j) \in A} w_{ij}^C c_{ij}^{2D}$$
$$C^{PS} = \sum_{(i,j) \in A} w_{ij}^C c_{ij}^{PS}$$
$$C^R = \sum_{i \in E} \sum_{r \in R} w_{r,i}^R x_{r,i}^R$$
$$C^P = \sum_{i \in E} \sum_{p \in P} w_{p,i}^P x_{p,i}^P$$
$$C^{FL} = w_{FL}^{2R} \sum_{i \in E^{FL}} \sum_{p \in S^{FL}} \lambda x_{ip}^{FL}$$

Vector $c$ will be defined according to the weights appearing in the objective function. We note here that $c_{sup} > 0$; i.e. the surplus variables are minimized in the objective. Consequently, assumption 1 is verified since the soft constraints are of the form given in (3), as mentioned earlier, and the surplus variables are minimized in the objective. As a result and following the verification of assumptions 1, 2 and 3, the model can be represented as a cardinality constrained problem as given in (5).

**APPENDIX C PROOF OF THEOREM 5.3**

From optimality of the solution of (11) we may write:

$$\frac{1}{2n_{\text{in}}} \|\tilde{g} - X\tilde{\beta}\|_2^2 + \lambda \hat{\omega}^T \tilde{\beta} \leq \frac{1}{2n_{\text{in}}} \|\tilde{g} - X\beta^*\|_2^2 + \lambda \hat{\omega}^T \beta^*$$

Defining $\hat{v} := \tilde{\beta} - \beta^*$ and expanding the above inequality, after some algebra we obtain the following:

$$\frac{1}{2n_{\text{in}}} \left( \|X\hat{v}\|_2^2 - 2\tilde{\beta}^T X\hat{v} - 2\|X\beta^*\|_2^2 + 2(\beta^*)^T X^T X\hat{\beta} \right) + \lambda \hat{\omega}^T \hat{v} \leq 0$$

Using the equation $\tilde{g} = X\beta^* + \zeta$ and considering that $\tilde{y}^T (A^T b + N\tilde{\alpha}) = \tilde{y}^T \tilde{x} = 0$ and $c^T \tilde{x}^* \leq c^T \tilde{x}$ following assumption 4, then we obtain after some algebra:

$$\frac{1}{2n_{\text{in}}} \|X\hat{v}\|_2^2 \leq \frac{1}{n_{\text{in}}} \|\tilde{\beta}\|_2^2 + \lambda \hat{\omega}^T \hat{v}$$

where in the last expression Holder’s inequality was used and the fact that $\tilde{y}^T x^* \leq \|\tilde{x} - x^*\|_1 \leq \|\tilde{y}\|_1$. Note, that the term $\lambda c^T N\tilde{\alpha} (\tilde{\alpha} - \alpha^*)$ was eliminated due to global optimality of $x^*$, which makes this term non-negative. Now, following the assumption in 5, which states that:

$$\lambda \geq \frac{\|X^T \zeta\|_\infty}{n_{\text{in}}}$$

then we can rewrite the above inequality as:

$$\frac{1}{2n_{\text{in}}} \|X\hat{v}\|_2^2 \leq 2\lambda \|\hat{v}\|_1$$

Finally, from the restricted strong convexity assumption in 5, we obtain:

$$\|\tilde{\beta} - \beta^*\|_1 \leq \frac{2\lambda}{\kappa^*}$$

The restricted set $C$ can be inferred from the constraints in (14) as follows:

$$\tilde{x}, x^* \in \{0, 1\}^n$$
$$\tilde{w}_{S^*} - \tilde{w}_{S^*_e} \in [-1, 0]^{n_{in}}$$
$$\tilde{x}_{S^*} - \tilde{x}_{S^*_e} \in \{0, 1\}^{n_{in}}$$
$$N_{\lambda}(\tilde{\alpha}_{S^*} - \alpha_{S^*_e}) \in [-1, 0]^{n_{in}}$$
$$N_{\lambda}(\tilde{\alpha}_{S_{S^*}} - \alpha_{S^*}) \in \{0, 1\}^{n_{in}}$$

where, $n_{in}$ and $n_{in}$ are the dimensions of the true and complement support of $x^*$, respectively. As a result, $\tilde{w}_{S^*} \in \{0, 1\}^{n_{in}}$, $\tilde{w}_{S^*} \in \{0, 1\}^{n_{in}}$ and $\tilde{w}_{S_{S^*}} \in \mathbb{R}$, where $\tilde{w}_{S_{S^*}}$ is the last element of $\tilde{v}$ which is associated with the error in $d_l$. We note that the same error bound will result when the objective $c^T x$ is relaxed as a soft constraint given as: $z_{min} = c^T x^* + d_{sup}^* + \zeta$.

**REFERENCES**

[1] A. Wren, “Scheduling, timetabling and rostering—A special relationship?” in Proc. Int. Conf. Pract. Theory Automated Timetabling, Berlin, Germany: Springer, 1995, pp. 46–75.

[2] E. Burke and W. Erben, Practice and Theory of Automated Timetabling III: Third International Conference, PATAT 2000 Konstanz, Germany, August 16–18, 2000 Selected Papers, vol. 2079. Berlin, Germany: Springer, 2003.

[3] B. McCollum, P. McMullan, E. K. Burke, A. J. Parkes, and R. Qu, “The second international timetabling competition: Examination timetabling track,” Queen’s Univ., Kingston, ON, Canada, Tech. Rep. QUB/IEEE/Tech/ITC2007/-Exam/v4.0/17, 2007.

[4] B. McCollum, P. McMullan, A. J. Parkes, E. K. Burke, and R. Qu, “A new model for automated examination timetabling,” Ann. Oper. Res., vol. 194, no. 1, pp. 291–315, Apr. 2012.

[5] B. McCollum and E. K. Burke, “The practice and theory of automated timetabling,” Ann. OR, vol. 218, no. 1, pp. 1–2, 2014.

[6] E. Ozcan, E. K. Burke, B. McCollum, D. Kjenstad, and A. Riise, “The second practice and theory of automated timetabling (2012),” Ann. Oper. Res., vol. 230, no. 1, pp. 1–2, 2016.

[7] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, “A survey of search methodologies and automated system development for examination timetabling,” J. Scheduling, vol. 12, no. 1, pp. 55–89, Feb. 2009.

[8] A. S. Uyar, E. Ozcan, and N. Urquhart, Automated Scheduling and Planning: From Theory to Practice, Berlin, Germany: Springer, 2013, vol. 505.

[9] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hilleberg, “Practices in timetabling in higher education institutions: A systematic review,” Ann. Oper. Res., vol. 275, no. 1, pp. 145–160, Apr. 2019.

[10] A. Cataldo, J.-C. Ferret, J. Miranda, P. A. Rey, and A. Sauré, “An integer programming approach to curriculum-based examination timetabling,” Ann. Oper. Res., vol. 258, no. 2, pp. 369–393, Nov. 2017.
[11] T. Arbabio, “Modeling and solving university timetabling,” Ph.D. dissertation, Dept. Gnie Informatique, Université de Technologie de Compigne, Heudiasyc, Compiègne, France, Dec. 2014.

[12] S. Eben, A. Ifai, and A. Shahr, “On the complexity of time table and multi-commodity flow problems,” in Proc. 16th Ann. Symp. Found. Comput. Sci. (sfcs), Oct. 1975, pp. 184–193.

[13] G. H. Fosseca, H. G. Santos, E. G. Carrano, and T. J. Stidsen, “Integer programming techniques for educational timetabling,” Eur. J. Oper. Res., vol. 262, no. 1, pp. 28–39, 2017.

[14] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” Oper. Res., vol. 14, no. 4, pp. 699–719, 1966.

[15] W. Zhou and H. Teng, “Simultaneous passenger train routing and timetabling using an efficient train-based lagrangian relaxation decomposition,” Transp. Res. B, Methodol., vol. 94, pp. 409–439, Dec. 2016.

[16] M. L. Fisher, “The lagrangian relaxation method for solving integer programming problems,” Manage. Sci., vol. 27, no. 1, pp. 1–18, Jan. 1981.

[17] T. Ralphs, Y. Shinano, T. Berthold, and T. Koch, “Parallel solvers for mixed integer linear optimization,” in Handbook Parallel Constraint Reasoning. Cham, Switzerland: Springer, 2018, pp. 283–336.

[18] A. Schrijver, “A computational study of a cutting plane algorithm for university course timetabling,” J. Scheduling, vol. 8, no. 6, pp. 497–514, Dec. 2005.

[19] H. G. Santos, E. Uchoa, L. S. Ochi, and N. Maculan, “Strong bounds with cut and column generation for class-teacher timetabling,” Ann. Oper. Res., vol. 194, no. 1, pp. 399–412, Apr. 2012.

[20] T. K. Ralphs and M. V. Galati, Decomposition Methods for Integer Programming, Hoboken, NJ, USA: Wiley, 2011.

[21] E. K. Burke, J. Mareček, A. J. Parkes, and H. Rudová, “A supernodal branch-and-price: Column generation for solving huge integer programs,” Oper. Res., vol. 46, no. 3, pp. 316–329, Jun. 1998.

[22] D. de Werra, “Restricted coloring models for timetabling,” Discrete Math., vol. 165–166, pp. 161–170, Mar. 1997.

[23] D. de Werra, “Restricted coloring models for timetabling,” Discrete Math., vol. 165–166, pp. 161–170, Mar. 1997.

[24] E. K. Burke, J. Mareček, and A. J. Parkes, “Semidefinite programming relaxations in timetabling,” in Proc. PATAT, 2011, p. 481.

[25] M. L. Fisher, “The lagrangian relaxation method for solving integer programs,” Oper. Res., vol. 27, no. 1, pp. 1–36, Jan. 1979.

[26] T. Hastie, R. Tibshirani, and M. Wainwright, Statistical Learning With Sparsity: The Lasso and Generalizations. Boca Raton, FL, USA: CRC Press, 2015.

[27] D. Ge, X. Jiang, and Y. Ye, “A note on the complexity of lp minimization,” Math. Program., vol. 129, no. 2, pp. 285–299, Oct. 2011.

[28] A. Schrijver, “A survey of approaches to the break minimization problem,” in Proc. 8th Workshop Algorithmic Approaches Transp. Modeling, Optim., Syst. (ATMOS), Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

[29] E. Hassannayebi, S. H. Zegordi, and M. Yaghini, “Train timetabling for an urban rail transit line using a lagrangian relaxation approach,” Appl. Math. Model., vol. 40, nos. 23–24, pp. 9892–9913, Dec. 2016.

**ALI AHMED AL-MATOUQ** received the B.Sc. and M.Sc. degrees (Hons.) in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, in 2002 and 2007, respectively, and the Ph.D. degree in electrical engineering from the Colorado School of Mines, in 2014. He has five years of industrial experience as a Process Control and Automation Engineer and is an ISA certified automation professional. He is currently an Assistant Professor with the Department of Engineering Management, Prince Sultan University, Riyadh. His research interests include process control and convex optimization.