2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

Solving the Exam Scheduling Problem with GRA$^+$

Haibin Zhu, *Senior Member, IEEE, Zhe Yu, and Youssou Gningue*

Abstract – **Exam scheduling is a complex problem. Traditional ways are using either graph-based heuristic algorithms or mixed linear programming (MLP) plus human intelligence. The heuristics could not always guarantee an optimal solution, and MLP, due to its generality, needs human schedulers to model for special requirements.**

Role-Based Collaboration (RBC) and its Environments - Classes, Agents, Roles, Groups, and Objects (E-CARGO) model and Group Role Assignment with Constraints (GRA+) are tools well suited to tackle such a problem. With RBC E-CARGO, and GRA$^+$, human schedulers save a lot of effort to create symbols and the relationships among the components.

This paper formalizes the problem of exam scheduling as an extended GRA with Conflicting Agents on Roles (GRACAR) problem, proposes practical solutions by using a Linear Programming (LP) solver, i.e., the IBM ILOG CPLEX Optimization Package (CPLEX). The proposed solution is verified by a real-world case study, and provides technical support for human schedulers in solving similar problems.

Index Terms—**Exam scheduling, constraints, group role assignment with constraints (GRA$^+$), Role-Based Collaboration (RBC), the Environments - Classes, Agents, Roles, Groups, and Objects (E-CARGO) model.**

I. INTRODUCTION

Resource allocation is a typical management problem in the real-world. Resource allocation should have required feasibility and efficiency. Such problems are often highlighted during the assignment process related to human power such as scheduling problems.

Examination scheduling [1, 3, 5-13], a very complex problem, is to assign exams to specific time slots and venues so that different constraints and policies identified by administrators or faculties would be fully respected. It is an inevitable recourse allocation practice along with the lecture scheduling problem in educational settings. The critical challenge of the scheduling problem is to avoid any concurrent assignment against restricted resources. Meanwhile, any specific rules applied to individuals and items should also be considered towards the scheduling objective. As to the exam timetable of a university, students, as long as enrolled in certain courses, are supposed to be scheduled into exams corresponding to the related courses. In other words, all the exams must be scheduled in the timetable for students whoever registered the corresponding courses of the exams. Additionally, each individual student can only take one single exam at a time, and the venue utilized to accommodate the students of an exam cannot be over its capacity. However, the general rules mentioned above are only the basic principle every administrator in the Registrar Office of a school has to obey, the practical settings of the exams scheduling in most post-secondary institutions are far more beyond that model. This paper proposes a new method to consider exam scheduling as a role assignment problem. Fortunately, this consideration makes the problem solvable with the help of our previous work on Group Role Assignment (GRA) including GRA$^+$ where a few more constraints are provided.

Our previous work on Role-Based Collaboration (RBC) and its Environments - Classes, Agents, Roles, Groups, and Objects (E-CARGO) model [14-16] plus GRA$^+$ [14] established solid foundations to tackle such assignment problems [2].

This paper is organized as follows. Section II depicts a real-world scenario for exam scheduling. Section III introduces the Group Role Assignment with Conflicting Agents on Roles (GRACAR) model [14], which is a special case of GRA$^+$. Section IV presents the formalization of the exam scheduling problem; Section V proposes the solution by providing pertinent data structures to solve the exam scheduling problem; Section VI illustrates a case study. Section VII briefly discusses the implementation and the performance experiments and the practicability of the CPLEX based solution. Section VIII reviews the related work. This paper concludes by pointing out topics for future work in Section IX.

II. A REAL-WORLD SCENARIO

The Registrar Office of school X is going to conduct an exam session for all the students across in-class courses of the Fall term, the office administrator Lucie is responsible for this project and hopes to present an executable timetable of the exam session with conflicts free in a limited time. Since Lucie has the student enrollments information, she also needs to figure out the other fundamental information from the school administration. What is more, her manager proposes a policy that no more than two exams applied to each student per day in case of the overload, and Dr. Lee, the instructor of Course COSC 1046, particularly requests a morning slot for his course.

Besides that, Lucie knows that there are three more implied constraints she has to take into considerations:

- every exam must be scheduled into the timetable;
- every student can only take one single exam at a time; and
- every venue cannot be over its capacity.

In light of such a challenge, Lucie realizes she may not manually schedule a feasible timetable within a short time since there are many tricky constraints she has to tackle against the

This work was supported in part by Natural Sciences and Engineering Research Council, Canada (NSERC) under Grant RGPIN-2018-04818.

H. Zhu is with the Department of Computer Science and Mathematics, Nipissing University, 100 College Drive, North Bay, Ontario, P1B 8L7, Canada (phone: 705-474-3450 ext. 4434; fax: 705-474-1947; e-mail: haibinz@nipissingu.ca).

Z. Yu and Y. Gningue are with the Department of Mathematics and Computer Science, Laurentian University, 935 Ramsey Lake Rd, Sudbury, ON P3E 2C6, Canada (emails: zyu@laurentian.ca, ygningue@cs.laurentian.ca).

exam congestion. If the data size is not very large, she could handle it with trails and errors within or beyond reasonable time, e.g., a final executable optimized exam timetable, capturing all the constraints, is shown in Table I.

Table I. An Example of Exam Schedule

Items \ Day	Day 1			Day 2		
Time	9-11am	1-3pm	6-8pm	9-11am	1-3pm	6-8pm
Course 1	GEOG 2247	COSC 3426	COSC 1056	COSC 1046	STAT 2126	GEOG 2016
Course 2		ECON 2266				
Course 3		ENGR 2506				
Enrollment	6	16	11	8	12	9
Capacity	6 spots - 20 spots					

Even though the scenario above is just a miniature of the real examination scheduling practice, it clearly demonstrates the significance of the proposed problem. Note that there may be more than one optimized outcome with respect to the timetable, and the final result does not guarantee the various rules and policies. Evidently, exam scheduling is a typical GRA+ problem.

III. Brief the GRACAR Formalization

Based on RBC and the E-CARGO model [14-16] and GRACAR [15], we use agents, roles, groups and environments to express related components, then we have the following definitions. Following E-CARGO, $\mathcal{N}$ is used to denote the set of non-negative integers, i.e., {0, 1, 2, 3, …}, m (=$|\mathcal{A}|$) the size of the agent set $\mathcal{A}$, n (= $|\mathcal{R}|$) the size of the role set $\mathcal{R}$. To simplify descriptions, we suppose that $\mathcal{A}$, $\mathcal{R}$ and other derived sets are ordered sets, i, i_1, i_2, … the indices of agents, and j, j_1, j_2, … the indices of roles. Agents, roles, groups, and assignments are concentrated in this paper. The *Environment* component of the E-CARGO model is expressed by several symbols and mathematical structures.

Definition 1 [14-16]: *Role range vector L* is a n-vector, where $L[j] \in \mathcal{N}$ $(0 \le j < n)$ expresses the number of agents required for a role.

Definition 2 [14-16]: *Qualification matrix Q* is an $m\times n$ matrix to express the suitability of an agent on a role, where $Q[i, j]\in[0, 1]$ indicates agent i's qualification value for role j $(0 \le i < m, 0 \le j < n)$, 0 means the lowest and 1 the highest.

Definition 3 [14-16]: T is an $m\times n$ matrix, where $T[i, j]\in\{0, 1\}$, $T[i, j] = 1$ means that agent i is assigned to role j and $T[i, j] = 0$ means the opposite.

Definition 4 [14-16]: The *performance of a* group σ is defined as the sum of the assigned agents' qualifications, i.e.,
$$\sigma=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i, j]\times T[i, j].$$

Definition 5 [14-16]: Role j is *workable* if it is assigned with sufficient agents , i.e., $\sum_{i=0}^{m-1} T[i, j] \ge L[j]$ $(0 \le j < n)$.

Definition 6 [14-16]: T is *workable* if every role j is workable, i.e., $\sum_{i=0}^{m-1} T[i, j] \ge L[j] (0\le j<n)$. Group g is *workable* if T is workable.

Definition 7 [14-16]: An *agent conflict matrix* is an $m\times m$ matrix $A^c(A^c[i_1, i_2] \in \{0,1\}, 0 \le i_1, i_2 < m)$, where $A^c[i_1, i_2] =1$ indicates that agents i_1 and i_2 are in conflict. 0 means no conflict. We define $A^c[i, i] = 0$ $(0 \le i < m)$ to mean that an agent has no conflict with itself.

With the introduction of A^c, we may define the GRACAR problem [14].

Definition 8 [14-16]: Given $\mathcal{A}$ ($|\mathcal{A}|=m$), $\mathcal{R}$ ($|\mathcal{R}| =n$), Q, L, and A^c, *Group Role Assignment with Conflicting Agents on Roles (GRACAR)* is to find T to obtain

$$\max \sigma=\sum_{j=0}^{n-1}\sum_{i=0}^{m-1} Q[i, j]\times T[i, j]$$

subject to

$$T[i, j] \in \{0, 1\} (0 \le i < m, 0 \le j < n) \quad (1)$$

$$\sum_{i=0}^{m-1} T[i, j] = L[j] \ (0 \le j < n) \quad (2)$$

$$A^c[i_1, i_2]\times(T[i_1, j]+T[i_2, j]) \le 1$$

$$(0 \le i_1 \ne i_2 < m)(0 \le j < n), \quad (3)$$

where expression (1) is a 0-1 constraint; (2) means that each role should be workable; and (3) means no conflict due to agents being assigned to the same role.

IV. Formalize the Exam Scheduling Problem

In Exam Scheduling, we have to take care hard constraints, which are required when scheduling, and soft constrants, which are preferred. This section is to compose the exam scheduling problem into a GRACAR pattern plus appropriate constraints, to model and solve the exam scheduling problem. An exam timetable should be conflict-free and minimize the exam congestion and respect the hard and certain soft constraints as discussed Section II. Typically, the categories of the items in exam timetables vary according to institutions' preferences in the educational settings. For example, the exam timetable from RMIT University [9] includes Subject Area, Exam Title, Date, Day, Start Time, Finish Time, Duration, Venue, etc.

$\mathcal{A}$ is a derived set of exams to be scheduled, its element (also the index of exam) $i \in \{0, \dots, m-1\}$, where $m = |\mathcal{A}|$, it can be obtained from the matrix A^s which are the fact sets previously stored in the database. $\mathcal{A}$ is the agent set corresponding to the GRACAR pattern, it is also the basic unit in the rows of the timetable. To a simplified model, we will use the positive integer value m to structure the exam indices in $\mathcal{A}$ by replacing the specific course's code and section.

$\mathcal{A}^s$ is a derived set of students participating in the exams, its element (also the index of student) $s \in \{0, \dots, m^s-1\}$ where $m^s = |\mathcal{A}^s|$, it can also be obtained from the matrix A^s. $\mathcal{A}^s$ is actually an implied agent set and reveals the student-course relationship information in the rows of timetable. The positive integer value m^s will be used to structure the student indices in $\mathcal{A}^s$ by replacing the specific student number or name.

After transferring the examination scheduling practices to the GRACAR pattern regarding to the roles and agents, it enables us to merely relay on the exam index $i \in \mathcal{R}$ and spot index $j \in \mathcal{A}$ of the timetable to create the problem models.

A^s denoted as the fact sets already known, if formed in matrix, is an $m^s \times m$ sparse matrix where $A^s[s, i] \in \{0,1\} (s \in$

$\mathcal{A}^s, i \in \mathcal{A}$) indicates tht student s is registered in the exam i if 1 marked, or 0 otherwise. A^s is the fundamental student-course enrollment information demonstrating the essential relevance within between students and courses, in addition to indicating the size of the $\mathcal{A}$ and $\mathcal{A}^s$ sets and other specific information employed to the scheduling process.

Besides that, N denotes an m vector where $N[i] = \sum_{s=0}^{m^s-1} A^s[s,i]$ ($i \in \mathcal{A}$) indicates the number of students enrolled in exam i. It is used to measure the spots availability against the venue's capacity. The total enrollment number of the exams session is $\sum_{i=0}^{m-1} N[i]$, which illustrates a picture of the entire registrations across the students and exams.

A^c is an $m \times m$ exam conflict matrix, where $A^c[i,i'] \in \{0,1\}(i \neq i' \in \mathcal{A}\}$. $A^c[i,i'] = 1$ indicates whether two exames i and i' have common students, or 0 otherwise. A^c is symmetric and $A^c[i,i] = 0$, it can be derived from the student-course enrollment matrix A^s through a simple algorithm as follows:

```
Algorithm 1: ObtainConflictMatrix
-----------------------
Input: A^s
Output: A^c
  for (0≤i <m)
     B[i]= ∅;
     // B[i] is the set of registered students for exam i.
  endfor
  for (0≤s<m^s)
     for (0≤i<m)
        if ( A^s[s, i]= 1)
          B[i]= B[i]∪{ s };
        endif
     endfor
  endfor
  for (0≤i, i'<m)
     if (B[i]∩ B[i']≠ ∅)
        A^c[i, i']=1;
     endif
  endfor
----------------
```

To the examination scheduling problem, A^c is the core and basic information that is utilized to formalize the constraints, and dealing with the constraints in the scheduling process, in other words, is also meant to tactic against the exam conflict avoidance. The density of the exam conflicts indicates the degree of the exams congestion so as to influence the overall scheduling implementation.

To express the scheduling problem, we introduce the following symbols to express other constraints special for the exam scheduling problem.

$\mathcal{R}^d$ is an identified set of days to conduct all the exams, its element (also the index of date) $d \in \{0, \ldots, n^d - 1\}$ where $n^d = |\mathcal{R}^d|$. To a simplified mode, we will use the input positive integer value n^d to indicate the length of the exam session so as to structure the consecutive indices of dates in the $\mathcal{R}^d$ set regardless of the specific dates.

$\mathcal{R}^t$ is an identified set of time slots per day, its element (also the index of time slot) $t \in \{0, \ldots, n^t - 1\}$ where $n^t = |\mathcal{R}^t|$. Similarly, we will use the input positive integer value n^l to indicate the amount of exams periods within a day so that the consecutive indices of time slots get created in the $\mathcal{R}^t$ set regardless of the specific start and finish time. Upon the basis of generalization for research, we assume that every single day has the same amount of time slots, and the corresponding time slot of different days has the same length of time duration.

$\mathcal{R}^v$ is a derived set of venues utilized to accommodate the exams regardless of venue's name, location and other additional information in a simplified model, its element (also the index of venue) $v \in \{0, \ldots, n^v - 1\}$ where $n^v = |\mathcal{R}^v|$. Accordingly, C is an n^v venue capacity vector with respect to the $\mathcal{R}^v$ set, and it is an identified set user can determine. C indicates the venue's capacity of index v in $\mathcal{R}^v$, the positive integer value n^v can be obtained from C set since $n^v = |C|$, so is the $\mathcal{R}^v$ consequently. If $n^v = 1$, then $v \in \{0\}$ is the unique element and the capacity of which is $C[0]$ in $\mathcal{R}^v$. What's more, we assume that every time slot in the timetable is filled up with the identical $\mathcal{R}^v$ set.

$\mathcal{R}$ is the role set in the GRACAR model. It indicates the consecutive exam spots. As a derived set, it can be obtained from $\mathcal{R}^d$, $\mathcal{R}^t$, and $\mathcal{R}^v$. The spot element (also the index of exam spot) $j \in \{0, \ldots, n-1\}$, where $n = n^d \times n^t \times n^v$.

Through the above stretching process on the roles, we may model exam scheduling to a GRA$^+$ problem. A role $j \in \mathcal{R}$ incorporates information of *d*, *t* and *v*. With the requirement of exam scheduling, we need to introduce another *n*-vector U, which is denoted as an n upper bound vector of the exam slot $j(j \in \mathcal{R})$ corresponding to its venue's capacity, such that $U[j] = C[v]$ where $v = (j \bmod n^v)$ and $v \in \mathcal{R}^v$, C is the identified venue capacity set. U indicates the maximum seats available to each single exam spot j.

Definition 9: Given $\mathcal{A}, \mathcal{R}, \mathcal{R}^{\mathrm{d}}, \mathcal{R}^t, \mathcal{R}^v$, *Q*, A^c, *L*, *U* and *N*, the Exam Scheduling with Hard Constrants (ESHC) problem is to find a feasible matrix *T* to

$$\max \sigma = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i,j] \times T[i,j]$$

subject to (1), (3), and

$$\hat{t} = j \bmod n^v, \tag{4}$$

$$A^c[i,i'] \times (T[i,\hat{t}] + T[i',\hat{t}]) \leq 1$$
$$(i \neq i' \in \mathcal{A}, \hat{t} \in \mathcal{R}^{\hat{t}}), \tag{5}$$

$$\sum_{i=0}^{m-1} N[i] \times T[i,j] \leq U[j] \quad (j \in \mathcal{R}), \tag{6}$$

$$\begin{cases} \sum_{i=0}^{m-1} N[i] \times T[i,j] \geq L[j], & if \ \ n^v = 1 \\ \sum_{i=0}^{m-1} T[i,j] \leq 1, & if \ \ n^v > 1 \end{cases}, \tag{7}$$

where $\mathcal{R}^{\hat{t}}$=$\{0, \ldots, n^{\hat{t}} - 1\}$, as an auxiliary set on the roles, stands for the chronological time slots throughout the timetable, we use $n^{\hat{t}} = |\mathcal{R}^{\hat{t}}| = n^d \times n^t$.

Expression (Exp.) (4) gives the domain of variable $\hat{t}$. Exp. (5) regulates the condition that every two exams in any same time slot have no common students in conflicts, in other words, each student can only take one single exam at a time. Exp. (5) indicates that if i and i' have common students, then $A^c[i,i'] = 1$, meanwhile there must be no more than one exam (either i or i') assigned to the time slot $\hat{t}$, thereafter the value of the mathematical expression in (5) should be less than 1. Additionally, if $n^v = 1$, then the chronological time slot is equivalent to the corresponding exam spot, i.e., $\hat{t} \equiv j$, such that (5) is the same as (3).

Exp. (6) generally indicates that the venue(s) utilized to accommodate the students can not be over its capacity, such that all of the enrollment volumes $N[i]$ corresponding to the assigned decision variables $T[i,j]$ must be less than the venue's capacity where $T[i,j]$ s locate, and Exp. (7) specifies two situations: if $n^v = 1$ then there is only one venue utilized throughout the exam session, thus there maybe more than one exam accommodated as long as the capacity is not exceeded, meanwhile the least attendance should be guaranteed; whereas if $n^v > 1$ then (7) regulates the situation where there are more than one venues utilized to accommodate the exams in each time slot and every venue can only undertake one exam at a time. Also note that (7) alone is attributed to the soft constraint scope that users are supposed to manually identify the specific rules, the two expressions in (6) are the alternative conditions adhered to the hard constraint (6).

Please note that we did not describe the process of soft constraints in this paper to save space. Our implementations includes the soft constratins. Also please note that Definition 9 is not a plain Linear Integer Programming problem. We have to take care of it by special consideration in our implementation, i.e., using n^v as a switch to call two different CPLEX functions with different constraint expressions.

V. A SOLUTION

To solve the problem defined in Definition 9, we have to consider the following issues, i.e., exam assignment evaluation, capacity boundary, and object function.

A. Agent Evaluation

The aim of agent evaluation is to create the Q matrix. In exam scheduling, we evaluate the matching degrees between an exam to a spot, i.e., the time slot of a room on a day. A five-scale rating scheme is proposed in this research to meet the requirements, Table II indicates the possibility of an exam i assigned to an exam spot j (minimum unit in the timetable) in percentage, which is corresponding to the qualification matrix defined in GRA$^+$. If $n^v = 1$, it is simplified to the evaluation of an exam to a time slot in the timetable without the venue being involved.

Table II. Exam Evaluation Scale

Rating		Value
Assigned	★★★★★	100%
Preferred	★★★★	75%
Default	★★★	50%
Undesirable	★★	25%
Excluded	★	0%

The values converted from the ratings are manually assigned, user can actually customize the degree of rating as well as its corresponding value depending on different examination scheduling settings. To initialize Q, an $m \times n$ matrix should be set up and the default element values are 50%. Thereafter, users can modify the default value of exam i in particular exam spot(s) according to the situation, e.g., [*exam preferred to be assigned to a specific venue of a particular time slot of each day*], then a user needs to modify the values of the exams spots js for exam i from 50% to 75%, with respect to every preferred venue v within its preferred time slot t throughout all days in the timetable. User has to go through all the customized exam requests from school faculties or administrators to eventually obtain an updated version of the qualification matrix Q.

B. Role requirement

To set up L is required to solve a GRACAR problem. In exam scheduling, we have to set up the value of it and modify the corresponding constraints to express the limit of an exam spot, i.e., U.

If $n^v = 1$, then $U[j] = C[0]$ which means U is an constant positive integer n vector, meanwhile, L is applied in this situation to indicate the minimum amount of attendance that should be guaranteed for each venue v in the exam spot j (also the time slot $\hat{t}$), such that $L[j] = p \times U[j] = p \times C[0]$ where p is the utilization rate that user can identify according to different examination scheduling settings.

Both U and L are derived from C set and are considered as the component of the environment $\mathcal{E}$ in the E-CARGO model that strictly regulates the examination scheduling system.

The decision variable set T, denoted as the assignment indicator of the examination scheduling, is an $m \times n$ exam-oriented decision matrix, where $T[i,j] \in \{0,1\} (i \in \mathcal{A}, j \in \mathcal{R})$. It indicates exam i is finally scheduled in exam spot j if 1 is marked, or 0 otherwise. Along with the qualification matrix Q, decision matrix T has the identical matrix pattern corresponding to Q, both of them are within the same group g of the E-CARGO model in GRACAR and are also the basic structure of the exam timetable.

C. Objective

The objective of the exam scheduling problem is to maximize the assigned Q values. In the educational settings, the objective towards different examination scheduling tasks is not unique, this paper is particularly based on the GRACAR methodology, which is to obtain an executive and optimized decision variables set T through maximizing the objective value σ that is subject to certain constraints.

It is worth mentioning that, the objective value σ obtained after respecting certain constraints is a fixed rational number no matter how the scheduling proceeds, because the GRACAR problem is always to seek an optimal solution among others against the constraints as long as the problem is solvable. However, the optimal result may not guarantee all the customized exams from the qualification matrix Q to be assigned as users' intentions because the optimal solution is upon the overall exam timetable performance in Σ regardless of any individual particularity.

VI. CASE STUDY

The data of case *HEC92* [9] stored in the file *hec-s-92.csv*, it is a raw data simply indicating the general student-exam information that there are 2823 students in total across 81 courses, and each record (student) has 1 to 6 fields (courses) at large.

Carter et al. [9] tested a set of 13 examination scheduling problems from 3 Canadian highs schools, 5 Canadian, 1 American, 1 UK and 1 mid-east universities. The benchmark settings to the case *HEC92* are shown in Table III, there are two developed options towards the initial configuration. Option 1* is the original goal to space out the conflicting exams within a limited number of time slots, and there is no restriction either on the capacity of the rooms or any other additional conditions. To our best knowledge, the best result so far is 18 time slots respecting only the primary hard constraints described in our research. Option 2* was proposed in [9, 10], and emphasized additional soft constraints, i.e., no back-to-back exams. In addition to venues and their capacities (e.g., one venue and 1500 spots), option 2* sets up 19 days and 3 time slots per day as a presumption to schedule out an optimized timetable.

Table III. Configuration References of Benchmark Data Set

Condition	*HEC92* Benchmark Setting		*HEC92* Enhanced Scheduling Setting		
	Option 1*	Option 2*	Option 3	Option 4	Option 5
Objective	Minimum number of time slots	Avoid exams congestion	Minimum number of time slots	Avoid exams congestion	**Maximum overall performance**
Number of days	n/a	19	n/a	**18**	**18**
Number of time slots	18	3	**17**	3	3
Number of venues	unlimited	1	Unlimited	1	1
Capacities	unlimited	1500	Unlimited	**1000**	**1000**
Constraints	hard constraints only	no back-to-back exam	hard constraints only	no back-to-back exam	no back-to-back exam **exam evaluation**

In light of the proposed approach in this paper to solve such an examination scheduling problem based on the advanced GRA^+ methodology, we could be full of academic confidence to test our optimization scheduling system with harder restrictions against the literature results towards the case *HEC92* of benchmark data sets.

According to the two options of reference configuration, first of all, the following case studies will respectively test our models with correspondence to those two condition settings. The enhanced conditions are summarized in Table III. Furthermore, this case study takes advantage of the features of GRA^+ to test the scheduling system under more restricted but practical conditions in Option 5. The advanced configuration to the case *HEC92* is shown in Table III.

Now we can clearly see that Option 5 is far more complicated than the other four conditions, since it is aimed to maximize the total timetable performance respecting all the additional constraints in addition to the hard constraints. If Option 5 could be properly solved in a tolerant computational time, then other cases will be solved without too much difficulty.

VII. Implementation and Performance Experiment

Table IV shows the test environments where the scheduling implementation project properly works. This project is mainly designed with four separate Java classes, each of which is amid to deal with particular function along with the IBM ILOG CPLEX Optimization Package.

Table IV. Experiment Platform

Processor	Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
RAM	8.00 GB (7.85 usable)
OS	Windows 10 Home (version 1909)
IDE	IntelliJ IDEA 2020.1 (Ultimate Edition)
JDK	Java Release 14
API	IBM ILOG CPLEX Package, Version: 12.9.0.0

Before deploying the primary programming, original data source such like the student-exam enrollment raw data stored in external database should be converted to a certain conventional format A^s which is compatible to the scheduling programming, besides that some essential factors like exam conflict matrix A^c and enrollment volume N set should also be derived prior to the main programming. The other essential factors, i.e., customized exam evaluation matrix Q, venue capacity set C, number of venues, days, time slots and so on, are variable parameters obtained from users' dynamic input and thus can be programmed within the primary programming process.

First of all, data sources acquired from the administrative database and users' input should be preprocessed before we can generally import them to the scheduling system. The fact sets, such as, the student-exam enrollment information matrix A^s, indicate the common relationship between student and exam that we can take advantage of to extract useful information needed to the scheduling process. Therefore, we need to convert the original source file from external to a certain conventional format suitable to the scheduling system. In our implementation, we assume that the enrollment information is kind of .csv file compatible to certain databases, for each student record, the data fields store the indices of the exams in numeral strings.

With the benchmark data set *HEC92* towards the condition of Option 5, the performance indicators are listed in the Table VI with a soft constraint "no more than one exam per day".

Table VI. Performance of Case Study

Indicator	Optimal status	Runtime (second)	Overall performance σ	Gap* (%)
Value	True	15.18	41.25	0.00

Please note that the gap value is such a CPLEX built-in indicator, it shows the percent difference between the obtained solution and highest lower bound at the time the algorithm is stopped. In this paper, the gap value may always be zero to the assessment because we always pursue an optimal solution to the examination scheduling problem.

In general, if we have 3000 students, 100 courses, the exam schedule period is in 2 weeks, each day we have 3 time slots for exams, and we use one large exam room with 1000 seats, the consumed time is around 10 to 20 seconds. We can conclude that the proposed approach is practical for secndary schools, colleages, or universities in an average size.

VIII. Related Work

Exam scheduling has been a research topic for a long time [1, 3, 5-13]. Many solutions discussed here are special cases and cannot be applied widely. None of them uses the proposed way in this paper.

Akbulut and Yılmaz [1] developed a solution to the exam timetabling problem in universities. They believe that numerous students and courses make it difficult to schedule exams during the final weeks. Their system aimed to schedule different exams in same halls, similar to that of our case study. Ganguli and Roy [3] used graph coloring heuristic algorithm to build up an application and conducted the tests of hard and soft constraints through a certain amount of data sets. Their algorithm emphasizes on constraint satisfaction, even distribution, unique solutions and optimal outcome. Kasm *et al.* [5] proposed a heuristic method to facility examination scheduling problem, which requires cross-validation on both student and course levels, thus it significantly increases the computational complexity. Their method enables the timetabling system to obtain a near-optimal solution with a relatively large population of enrollments in a smaller computational time. In fact, our proposed solution also contains such a cross-validation issue with students involved. Leighton [7] proposed the renowned Recursive Largest First (RLF) algorithm, which is attributed to such a kind of heuristic graph coloring algorithm, and it only needs $O(n^2)$ time behavior to deal with most spares graphs so that it can be particularly used to address large-scale scheduling problems. Malkawi *et al.* [8] solved the examination scheduling problem by using a graph-coloring-based algorithm, i.e., an algorithm that accounted for a customized set of constraints. Their algorithm allows users to choose particular timeslots and exam venues, but the research is not capable of regulating the venue's capacity. Their algorithm assigns one exam per day for each student as presumption. If such an assignment does not work, then it releases the constraint to two exams per day if possible. This idea is not as good as our modelling since we introduced a two-day timespan and covers more constraints of the scheduling problem. Qu *et al.* [10] challenged the classical graph-based applications and proposed the new constraint-based approach, which can enable us to revoke the exams during the scheduling process. In contrast with graph-based techniques, the proposed theory handles the problem as mathematical expression consisting multiple constrains. An evaluation norm was particularly developed by Laporte and Desroches [6] to propose an automatic procedure for constructing examination timetables in universities. Rachmawati *et al.* [11] considered a situation when a course instructor has to be present during the exam. This constraint is based on the teacher-oriented view and is suitable to some small educational institutions where a course instructor may teach multiple different courses and the constraint regulates that exams from the same course instructor can not be assigned at the same time. Woumans *et al.* [13] presented a student-centric view of the examination scheduling problem and allowed for multiple versions of an exam to be scheduled to increase the spreading of exams for students. They also proposed two column generation (CG) algorithms combined with both heuristics and mathematical formulations such as the integer programming (IP) technique in order to deal with the examination scheduling problem efficiently. Our previous work [14,15,16] provides solid foundation for this work. This work is a direct application and extension of the work in [14].

IX. CONCLUSIONS

This paper proposes a new approach for exam scheduling with the modelling support by Group Role Assignment with Constraints (GRA^+). A case study with a benchmark data set *HEC92* assures that the proposed method is not only feasible but also efficient. Further performance experiments confirm our claim made by the case study. This research again demonstrates the significant modelling superiority of RBC, E-CARGO, and GRA. Future studies can be conducted in the following directions:

1. Apply GRA^+ modelling method to other scheduling problems in industry.
2. Apply the proposed method in real-world exams and conduct usability studies, and develop online scheduling services based on our GRA^+ modeling method and an optimization platform.

REFERENCES

[1] A. Akbulut and G. Yılmaz, "University Exam Scheduling System Using Graph Coloring Algorithm and RFID Technology," *Int'l J. of Innovation, Management and Technology*, vol. 4, no. 1, 2013, pp. 66–72.

[2] R. E. Burkard, M. Dell'Amico and S. Martello, *Assignment Problems, Revised Reprint*, Siam, Philadelphia, PA, 2009.

[3] R. Ganguli and S. Roy, "A Study on Course Timetable Scheduling using Graph Coloring Approach," *Int'l J. of Computational and Applied Mathematics*, vol. 12, no. 2, 2017, pp. 469-485.

[4] IBM, ILOG CPLEX Optimization Studio, avail: http://www-01.ibm.com /software/integration/optimization/cplex-optimization-studio/, 2019.

[5] O. A. Kasm, B. Mohandes, A. Diabat, and S. el Khatib, "Exam timetabling with allowable conflicts within a time window," *Computers and Industrial Engineering*, vol. 127, pp. 263–273, Jan. 2019.

[6] G. Laporte and S. Desroches, "Examination timetabling by computer," *Computers and Operations Research*, vol. 11, no. 4, Jan. 1984, pp. 351–360.

[7] F. T. Leighton, "A Graph Coloring Algorithm for Large Scheduling Problems," *J. of Research of the Notional Bureau of Standards*, vol. 84, no.6, Nov.- Dec. 1979, pp. 489-506.

[8] M. Malkawi, M. Al-Haj Hassan, O. Al, and -Haj Hassan, "A New Exam Scheduling Algorithm Using Graph Coloring," *The Int'l Arab J. of Information Technology*, vol. 5, no. 1, Jan. 2008, pp. 80-87.

[9] R. Qu., "Benchmark Exam Timetabling Datasets," May 2020, avail: http://www.asap.cs.nott.ac.uk/external/resources/ .

[10] 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," *Journal of Scheduling*, vol. 12, no. 1, pp. 55–89, Feb. 2009.

[11] H. Rachmawati, E. F. Armay, and M. H. Purnomo, "Problem solving analysis of course scheduling using graph coloring technique based on bee colony algorithm: Parameter of lecturer priority as soft constraint in Electrical Engineering Department of Sepuluh Nopember Institute of Technology," in *2012 7th International Conference on Telecommunication Systems, Services, and Applications*, Bali, Indonesia, Dec. 2012, pp. 136–141.

[12] RMIT University, "RMIT University Exam Timetable - Semester - PDF Free Download," , May 2020, Avail: https://docplayer.net/9154672-Rmit-university-exam-timetable-semester-2-2015.html .

[13] G. Woumans, L. de Boeck, J. Beliën, and S. Creemers, "A column generation approach for solving the examination-timetabling problem," *European Journal of Operational Research*, vol. 253, no. 1, Aug. 2016, pp. 178-194.

[14] H. Zhu, "Avoiding conflicts by group role assignment", *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, April 2016, pp. 535-547.

[15] H. Zhu and M.C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. on Systems, Man and Cybernetics, Part C,* vol. 36, no. 4, July 2006, pp. 578-589.

[16] H. Zhu, M.C. Zhou and R. Alkins, "Group role assignment via a Kuhn-Munkres algorithm-based solution," *IEEE Trans. on Systems, Man and Cybernetics, Part A,* vol. 42, no. 3, May 2012, pp. 739-750.