

RESEARCH ARTICLE

A Matching Game-Based Laboratory Learning System for Resources Management in Remote Laboratories

AHMED M. ABD EL-HALEEM¹, MOHAMED G. ANANY²,
MAHMOUD M. ELMESALAWY¹, (Member, IEEE), AND EMAN SERAJ EL-DIN BAKR^{2,3}

¹Department of Electronics and Communications Engineering, Helwan University, Cairo 11792, Egypt

²Department of Electronics and Communications, Canadian International College, Cairo 11241, Egypt

³Network Planning Department, National Telecommunication Institute (NTI), Cairo 11432, Egypt

Corresponding author: Mahmoud M. Elmesalawy (melmesalawy@h-eng.helwan.edu.eg)

This work was supported by the Academy of Scientific Research and Technology (ASRT), Egypt, from 2021 to 2023, under Program RDI/P4/1/20-21.

ABSTRACT Recently, increasing demand for remote laboratories is noticed, due to the shifting of the educational systems from face-to-face education to online education through the internet. A Remote-controlled laboratory is based on the use of real instrumentations and components, in which experiments are completely conducted and controlled remotely by the students through the internet. Thus, it provides a hands-on learning experience for students on an online basis. In order to realize this increasing demand for remote laboratories, efficient management of the laboratory resources should be considered. In this work, a novel laboratory learning system is proposed, which provides the flexibility of deployment on any physical infrastructure, integrability with other different physical laboratory resources, and scalability. Moreover, a mathematical model for the laboratory learning system is formulated, to provide the basis for defining the optimization problem. The problem of minimizing the number of blocked time slots per acquiring student, by optimizing the association between the reserving students and the available resources at different time slots is also formulated, which reflects the maximization of resources utilization. Then, a matching game-based framework is proposed to optimally solve the formulated problem. Simulation results show the significance of the proposed matching game-based framework compared to other students to resources association schemes, in minimizing the number of blocked time slots per acquiring student and maximizing the resources utilization efficiency.

INDEX TERMS Remote laboratories, laboratory learning system, matching game, resource management.

I. INTRODUCTION

Nowadays, the remote laboratory system is an important and supplementary part of learning to bridge the gap between theory and practice. New technologies have provided the laboratories with a different form of observation, experimentation, and investigation whereby distance learning garnered the attention of researchers [1], [2]. Remote laboratories provide opportunities and flexibilities for students to conduct experiments by the level of ability and pace of

learning each student, anytime and anywhere [3], [4], [5], [6]. Hence, remote laboratory systems have been designed to perform the remote access platform for learners from faraway places [7], [8].

Historically, physical laboratory requirements have limited the ability to support flexible usage and sharing of resources. Nowadays, the rapid development of computer technologies and networking have supported the transformation to remote laboratories. Remote laboratories are important tools for both student learning and scientific research. They do represent a significant investment that needs to be developed and maintained [9], [10], [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai¹.

In order to operate the remote laboratories, and manage the user's accessibility, laboratory learning systems are required. Nonetheless, one of the major challenges in laboratory learning systems is the lack of standardization [13]. Which means that laboratory learning systems that runs and controls the remote laboratory may not be modular, portable, flexible, or scale. The most important feature of laboratory learning systems is to support sharing expensive resources across the world [14], [15]. In this regard, some benefits such as flexible access, sharing and integrating remote laboratories' resources, and modular architecture, have been shown to increase the attention of users [16], [17], [18], [19], [20], [21]. Regarding a flexible and integrable platform, resource allocations are considered a significant issue. Thus, proper scheduling can respond to the demand for having access to the remote laboratory.

Generally, remote laboratories have grown in sophistication to provide a number of benefits such as flexibility and scalability. Current remote laboratories can be categorized into two types: batch laboratories and interactive laboratories [22], [23]. Batches are for submitting a specification for the experiment and submitting it to run asynchronously at a later time when the apparatus is available [24]. When using batches, the user does not interact with the lab at all and overall utilization of the apparatus is limited by algorithms managing the queue [25]. Interactive laboratories, in contrast, allow users to interact with their experiments to give them full control over what's going on in real-time [26], [27]. Thus, they allow you to monitor and adjust an experiment as it executes, so every user must be allocated to a laboratory resource at a time when it is available. This process represents a resource scheduling and allocation task, and since interactive laboratories use laboratory infrastructure in different ways and provide some interesting opportunities and challenges, it is different from other types of scheduling tasks.

The specific design of the laboratory resources scheduling and reservation system will depend on a number of factors, including the number of users, the number of available laboratory resources, the type of experiment for each user, the required resources for each experiment, and guaranteeing the access for each reserving user. Moreover, the design of the resources scheduling algorithm has a significant impact on the number of available timeslots on the reservation system for each user, and how the user's access is managed and guaranteed. Using a reservation system, a user can often schedule an access time slot in advance, which in turn gives the user assured access at a predetermined time. It can be noted that the scheduling techniques that have been applied in laboratory learning systems up to this point have been either nonexistent or extremely basic. Most of the systems currently in use that does enable scheduling have chosen a basic reservation system.

II. LITERATURE REVIEW

A. RELATED WORK

In [28], the authors described the hardware and software architectures as a platform of a hybrid online laboratory developed at the TU Ilmenau. They adopted a dynamic booking system, such that resources are accessed dynamically, and a time slot is blocked, only if all the devices of the same type needed for the experiment are equipped. However, they did not consider different experiments that require multiple resources, and how resources are allocated to the users in that specific case, in order to minimize the number of blocked time slots.

The work in [29] developed an application that is used in evaluating the quality of their remote laboratory system. The considered remote laboratory system adopted a basic static reservation system, where students must reserve their timeslots in advance, in order to avoid conflict between students using the same resource at the same time.

The work in [30] proposed a low-cost, scalable remote FPGA laboratory architecture based on remote access mechanism for digital circuit design. nevertheless, the proposed remote laboratory architecture adopted a simple booking system for reserving the users' time slots.

In paper [31], the authors designed and implemented a unified and flexible web-based online laboratory framework for experimentations of control education. In this work, the used access mechanism was not described considering multiple users requiring the same resource.

The authors in [32] proposed a remote laboratory management system named REMLABNET, for managing experiments in the university level and school level as well. They used a basic reservation system based on students' registration.

Moreover, the authors in [33] proposed remote laboratory architecture that focuses on experimentations of embedded systems. The proposed remote laboratory architecture provides high scalability, and cost-efficiency based on a modular design, and hardware sharing techniques. The authors adopted a queuing system based on first come first serve in the remote laboratory management system layer. However, at congestion time, a user might take a long access time, which is the time taken by the user until it accesses the experiment, which in turn affects the quality of experience.

The work in [34] proposed an approach to integrate remote laboratories in a grid and service-oriented architecture-based environment, considering advanced reservation, and service level agreements, to set up, configure, and manage remote laboratories. They adopted a basic scheduling service that queries the availability of the required resources and negotiates the reservation across all the involved scheduling systems.

In addition, the authors in [35] developed an in-house solution to reserve resources via the internet, in order to

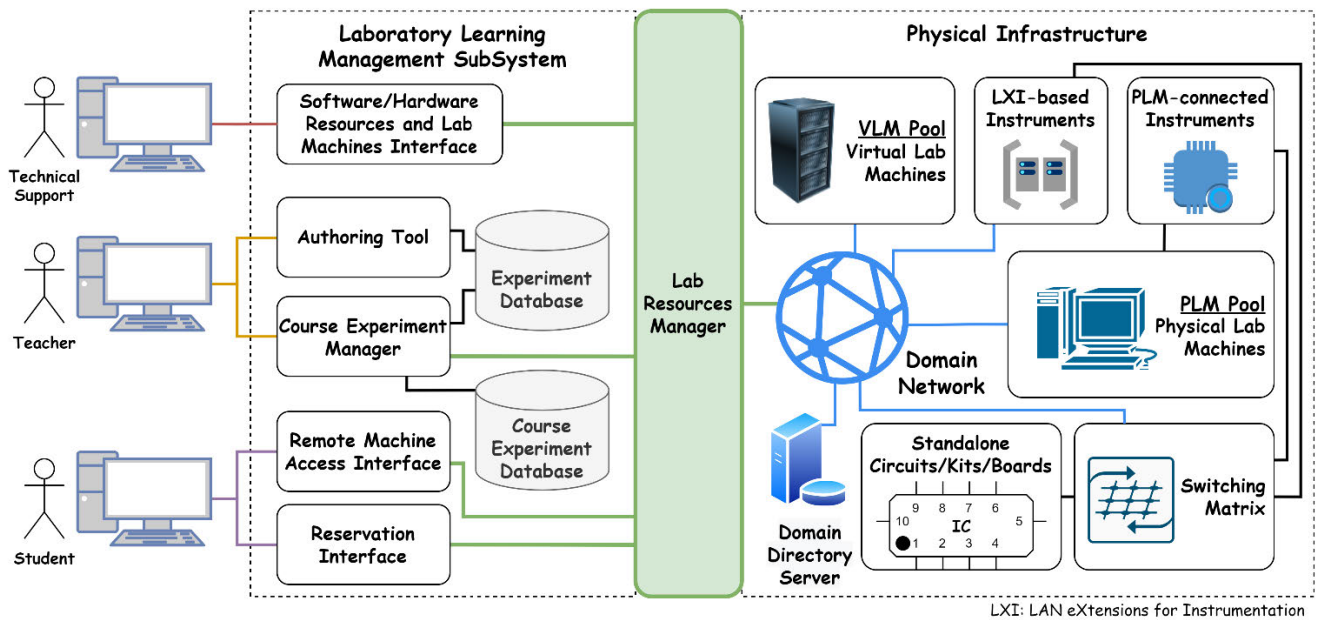


FIGURE 1. Proposed laboratory learning system architecture.

conduct their experiments and exercises at their convenience. They adopted a basic reservation system based on students booking, that permits the student to use the required remote laboratory resource without conflict with other students.

In [36], the authors proposed a new method for students to accessing multiple instances of remote laboratories. They used a server-side scheduler service as a central point of control to all access requests. For each access request, the scheduling service consults the in-memory database in order to find free rigs. This basic methodology does not take into consideration the resources required for different experiments, and it can not efficiently utilize the available resources in diverse experiments scenario.

The work in [37] designed and developed a remote laboratory system based on embedded system and user management server, that can be used by mobile devices having PC-level capabilities, for experiments on electronic telecommunication courses. The authors used a basic reservation method based on students' bookings.

Furthermore, In [38], the authors analyzed the scheduling algorithm used by Sahara remote laboratory system, they investigated its limitations, and proposed some modifications to address them. They used a hybrid calendar-based reservation and priority queuing mechanism, which takes advantage of early finished reservation sessions, in order to utilize the reserved resources by other users using the priority queuing. However, they did not consider experiments that require multiple resources, and how these resources can be managed dynamically to increase the availability of the overall resource.

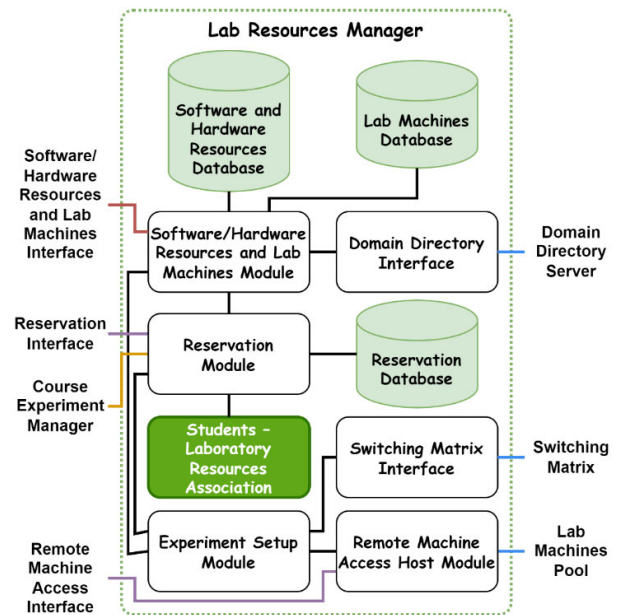


FIGURE 2. Laboratory resource manager.

The rest of the paper is organized as follows: Section III describes the proposed system architecture, while section IV presents a mathematical modeling for the proposed system until reaching an optimization problem formulation, which aims at minimize the students' time slots blocking rate. Then, section V presents matching game-based framework for students-resources association to solve the aforementioned optimization problem. After that, the performance of

the proposed framework is evaluated in section VI, followed by the conclusion in section VII.

III. SYSTEM ARCHITECTURE

In this section, the architecture of the laboratory learning system is proposed with the aim of minimizing the time slots' blocking rate for students acquiring experiments at the experiment reservation phase. In the proposed system architecture, a generalized physical infrastructure is considered, which reflects the flexibility of deploying the proposed laboratory learning system on any physical infrastructure. It also provides smooth integration between different laboratory infrastructures used in various fields at diverse spaces (e.g., An electronics laboratory can be integrated with a computer laboratory at different universities to be managed by the proposed laboratory learning system), which reflects its integrability and scalability.

Figure 1 shows an overview of the proposed system. The proposed laboratory learning system consists of two main subsystems, besides the connected physical infrastructure. Generally, we consider a physical laboratory infrastructure that consists of *virtual lab machines* and *physical lab machines (PLM)* (i.e., desktop computers), that may contain all the required software installed on it, and also are connected to the hardware lab resources through the domain network. It also may contain *PLM-connected Instruments*, which require a direct connection to a desktop computer for remote accessibility, and *LXI-based Instruments*, which are laboratory devices and instruments that have an LXI interface to be remotely accessed through the Internet. The physical infrastructure also may contain a *switching matrix*, which controls all the hardware resources interconnectivity according to control signals from the *Lab Resource Manager* to build the experiment topology.

A *domain directory server* is needed in the physical infrastructure to store information about the lab resources connected to the network and makes this information easy for administrators and users to find and use.

A. LABORATORY LEARNING MANAGEMENT SUBSYSTEM

The first subsystem in the proposed laboratory learning system is the laboratory learning management subsystem, which is the management software that helps teachers to create the laboratory learning object (experiment) and manage the way of experiment execution, it also provides an interface between the student and the physical resources, meanwhile, it allows technical support to manage the hardware and software resources. The laboratory learning management subsystem is composed of an *authoring tool*, *course experiment manager*, *remote machine access interface*, *reservation interface*, and *software/hardware resources and lab machine interface*.

The *authoring tool* which is the teachers' tool, it allows the teachers to create or edit experiments as a laboratory learning object, and store them in the *experiment database*. This includes the laboratory activities, contents, assessment, and evaluation mechanism. It also allows teachers to link an

experiment to the required software resources (i.e., Matlab, Labview, ...etc.) and hardware resources needed (i.e., LXI instruments, PLM connected instruments, standalone circuits, kits, boards, ...etc.), which are already stored as available resources in the *software and hardware resources database*.

The *course experiment manager* which is also a teachers' tool, it allows the teachers to access all the created experiment objects stored in the *experiment database*, through which the teachers can associate the created experiments objects to a course he/she is teaching. It provides the teachers with the availability to assign the experiment to a cohort or a group of students and configure the experiment time duration, start and due date, and the maximum number of attempts. The relations between the experiment, the course, and the students' group, are stored in the *course experiment database*.

On the other hand, the *remote machine access interface* is an interface between the students and the laboratory machine resources used during the experiment execution to fulfill his experiment objectives. It allows the student to start the experiment in his reserved time slot by initiating a remote access connection with a reserved virtual or physical machine through the *Lab resources manager*. The *Lab resources manager* delivers the remote access session connection information, and the shared session connection is sent to the student's browser.

Moreover, the *reservation interface* is an interface that allows a student to reserve a time slot to execute his assigned experiment during the experiment active period. The available time slots are obtained by the lab resources manager by running the matching game algorithm, through which all the required experiment resources are available. The student is then allowed to select his preferred time slot. After reserving the time slot, the related information is saved in the *reservation database*.

Furthermore, the *software/hardware resources and lab machine interface* is a technical support interface, which allows the technical support to install a new software and hardware control programs (i.e., software and hardware resources) on the virtual and physical laboratory machines. It allows the technical support to configure installed software, and to connect hardware for each virtual and physical laboratory machine. It provides the technical support to configure and edit the laboratory machines' access time (e.g., laboratory machines are only available at night, since day times are reserved for on-campus activities). All the information about the laboratory virtual and physical machines, and the software and hardware resources installed or connected to them, are stored in the *software and hardware resources database*, and the *Lab Machines database*.

B. THE LABORATORY RESOURCE MANAGER

The second main component in the proposed laboratory learning system is the *Laboratory Resource Manager*. The *Laboratory Resource Manager* is considered the interface that connects the *laboratory learning management subsystem*

with the physical infrastructure. It manages the physical infrastructure to ensure smooth and reliable students' connectivity to the remote laboratory. It provides an intelligent association mechanism, which aims at increasing the remote laboratory students' capacity, by minimizing the time slots blocking rate at the students' reservation phase and increasing the number of available time slots for reservation. Besides, it acts as a laboratory learning system decoupler, where it can be used as a standalone software that enables an intelligent management for diverse remote laboratory physical infrastructures, using different *laboratory learning management subsystem* designs.

Figure 2 shows the main components of the *Laboratory Resource Manager*. The first component is the *software/hardware resources and lab machines module*, which contains all the information about the lab physical and virtual machines and the software and hardware resources installed or connected to the physical laboratory domain network. It acts as an interface to the *software and hardware resources database*, *lab machines database*, and the *domain directory server*, to be accessed and edited by the technical support.

The second component is the *reservation module*, which is used to provide the *reservation interface* with the available time slots so that a student can book his machine reservation for a specific experiment. When a student accesses the laboratory learning system to book an experiment through the *reservation interface*, the reservation module requests the experiment required resources from the *course experiment manager*. According to the available hardware and software resources, the machines' available time slots, and the reserved resources, the matching game algorithm runs with the given criteria to deliver the available time slots for booking. The student is then allowed to select his preferred time slot. After reserving the time slot, the related information is stored in the *reservation database*.

The next component is the *students – laboratory resources association*, which is the brain of the *laboratory resource manager*. It runs the proposed matching algorithm in section VI, in order to find the optimum students to physical and virtual machines association, which aims at increasing the remote laboratory students' capacity, by minimizing the time slots blocking rate. The *students – laboratory resources association* gets its required information (i.e., the required hardware and software resources in each time slot stored on the *reservation database*, the resources available on each machine table stored on the *lab machine database*) from the *reservation module*. It also sends the available time slots to the *reservation module* for student reservations.

Moreover, the *experiment setup module* is responsible for preparing the experiment resources for the student at the beginning of the reserved time slot. Prior to the experiment execution, the *experiment setup module* requisites the ID of the experiment's associated laboratory physical or virtual machine from the *reservation module*, in order to request its activation from the *remote machine access host module*. It also requisites the required hardware resources IDs from

the *reservation module*, in order to configure the experiment topology, by sending a control signal to the *switching matrix* to connect the required hardware resources and the laboratory physical or virtual machine to set up the experiment.

Furthermore, the *remote machine access host module* is a module that creates a session sharing URL with control access, which is delivered to the student's *remote access interface*. This module is a remote access gateway, which allows session sharing and granting access control. For a student to be able to connect seamlessly and control the assigned remote lab machine, this module will open a session with the student's credentials on the remote lab machine's name using Remote Desktop Protocol (RDP). The RDP protocol provides a secure authentication through Network Level Authentication (NLA), and communication through Secure Sockets Layer (SSL) and Transport Layer Security (TLS).

IV. SYSTEM MODEL

In this section, we propose a mathematical formulation for the presented system architecture. Let $\mathcal{M} = \{1, 2, \dots, m, \dots, N^M\}$ be the set of the available pool of laboratory virtual machines and physical machines, such that N^M is the total number of virtual and physical machines. Besides the virtual and physical machines, the laboratory available resources are comprised of different software and hardware instruments and can be denoted by the set $\mathcal{R} = \{1, 2, \dots, r, \dots, N^R\}$, such that N^R is the number of available software and hardware resources, and N_r^M is the number of physical and virtual machines that has resource r (e.g. software that can be installed on different physical or virtual machines with licenses, or hardware replicas that are connected to different physical or virtual machines). Hence, the total number of resources in the system is $N_{total}^R = \sum_{r=1}^{N^R} N_r^M$. Also, the set of available resources for each machine $m \in \mathcal{M}$ are denoted by $\mathcal{M}_m \subseteq \mathcal{R}$, and N_m^R denotes the number of available resources in machine m . Moreover, the created experiments are denoted by the set $\mathcal{E} = \{1, 2, \dots, e, \dots, N^E\}$, where N^E is the total number of created experiments. Each created experiment $e \in \mathcal{E}$ has a set of required resources $E_e \subseteq \mathcal{R}$, such that there is at least a set $\mathcal{M}_m \supseteq E_e$. Also, each experiment $e \in \mathcal{E}$ is assigned a duration time t_e from the set of available time durations $\mathcal{T} = \{1, 2, \dots, t_d, \dots, N^T\}$, where N^T is the total number of available time durations. Besides, $N_e^{T_d}$ is the number of time durations that can be reserved by a student for experiment e . This number can be calculated by knowing the start scheduled time T_e^{start} and due to scheduled time T_e^{due} of the experiment, the virtual and physical machines' off-time duration for physical and virtual machines t_m^{OFF} , and the duration time t_e , as follow:

$$N_e^{T_d} = \frac{(T_e^{due} - T_e^{start}) - t_m^{OFF}}{t_e} \quad (1)$$

On the other hand, the set of students reserving experiments at time slot t are denoted by $\mathcal{S}_t = \{1, 2, \dots, s_t, \dots, N^{S_t}\}$, such that N^{S_t} is the total number of

students reserving time slot t . an association matrix X_t^e at time slot t represents the association between the set of students S_t and the set of virtual and physical machines M , such that X_t^e is an $N^{S_t} \times N^M$ matrix, and $N^M \geq N^{S_t}$. An element of the matrix X_t^e has the value of $x_{s_t,m}^e \in \{0, 1\}$, where $x_{s_t,m}^e = 1$ represents the reservation of student s_t to machine m for experiment e , while $x_{s_t,m}^e = 0$ means that machine m is not reserved by student s_t for experiment e , this can be formulated mathematically as follow:

$$x_{s_t,m}^e = \begin{cases} 1, & E_e \subseteq M_m, \text{ and } \sum_{s_t} x_{s_t,m}^e = 0 \\ 0, & E_e \neq E_e \cap M_m, \text{ or } \sum_{s_t} x_{s_t,m}^e = 1 \end{cases} \quad (2)$$

A student acquiring experiment e at time slot t that needs a duration time t_e is denoted by s_t^e , such that $s_t^e \notin S_t$ and $s_t^e = N^{S_t} + 1$. The availability of time slot t for experiment e to be reserved by student s_t^e is denoted by a_t^e , and can be represented as follow:

$$a_t^e = \begin{cases} 1, & \sum_{s_t} \sum_m x_{s_t,m} = N^{S_t} + 1 \\ 0, & \sum_{s_t} \sum_m x_{s_t,m} < N^{S_t} + 1 \end{cases} \quad (3)$$

where $a_t^e = 1$ if all the reserved students S_t and the acquiring student s_t^e for experiment e at time slot t are associated with machines from the machines pool \mathcal{M} . While $a_t^e = 0$ if only some of the acquiring and reserved students are associated with machines from the machine pool \mathcal{M} .

The availability matrix A^e for any student acquiring experiment, e is an $N_e^{T_d} \times 1$ matrix that contains the elements a_t^e where $t = [1, N_e^{T_d}]$. The number of available time slots $N_{s^e}^A$ for student s^e is equal to:

$$N_{s^e}^A = \sum_{t=1}^{N_e^{T_d}} a_t^e \quad (4)$$

While the number of blocked time slots $N_{s^e}^B$ for student s^e can be calculated as follow:

$$N_{s^e}^B = N_e^{T_d} - N_{s^e}^A \quad (5)$$

In this work, our problem is to minimize the blocking rate which is defined as the number of blocked time slots per acquiring student, by optimizing the association matrix X^e between all the reserved students and the available physical and virtual machines at different time slots, where $X^e = \left\{ X_1^e, X_2^e, \dots, X_t^e, \dots, X_{N_e^{T_d}}^e \right\}$. This problem can be formulated as follow:

$$\text{OPT} : \min_{X^e} N_{s^e}^B \quad (6)$$

$$\text{s.t. } x_{s_t,m}^e \in \{0, 1\}, \quad (7)$$

$$x_{s_t,m}^e = 1 : E_e \subseteq M_m, \text{ and } \sum_{s_t} x_{s_t,m}^e = 0, \quad (8)$$

$$x_{s_t,m}^e = 0 : E_e \neq E_e \cap M_m, \text{ or } \sum_{s_t} x_{s_t,m}^e = 1, \quad (9)$$

$$\sum_m x_{s_t,m} \leq 1, \quad (10)$$

$$N^{S_t} \leq \sum_m \sum_{s_t} x_{s_t,m} \leq N^{S_t} + 1, \quad (11)$$

In optimization problem OPT, constraint (7) ensures that the association index $x_{s_t,m}^e$ is equal only to 0 or 1; and, constraint (8) and (9) ensures that a student is associated with a specific physical or virtual machine m , if only the machine can provide the experiment's required resources E_e , and no student is associated with this machine; Also, constrain (10) ensures that a student cannot associate with more than one machine; Moreover, constraint (11) ensures that at least all the reserved students in a specific time slot t are associated to physical or virtual machines.

V. MATCHING GAME-BASED FRAMEWORK FOR STUDENTS – LABORATORY RESOURCES ASSOCIATION

The aforementioned formulated problem OPT can be posed as a matching problem, where students should be matched optimally to the physical and virtual machines according to their available resources and each student's experiment required resources. It can be considered as a one-to-one matching problem, where each student is matched to one physical or virtual machine. The matching theory is a Nobel prize winning framework that provides mathematical tractable solutions for combinatorial matching problems [39], [41], [42].

In order to solve this problem, we propose a framework based on matching game algorithm, where the optimization problem OPT is divided into $N_e^{T_d}$ matching sub problems. For a student acquiring an experiment e , the matching sub problems should be solved to determine the availability of each specific timeslot t . If a stable matching between the students (the acquiring student, and the reserving students at time slot t) occurs, then this timeslot is considered available, if not, then the timeslot is considered unavailable. Solving the matching sub problem is applied to each of the available timeslots for this student.

A. MATCHING GAME DEFINITION FOR STUDENTS' ASSOCIATION

The basic solution concept of a matching problem is defined as two-sided stable matching. The two-sided stable matching occurs only if there are no blocking pairs. A pair (s_t', m') is defined as a blocking pair if student s_t' prefers to be matched to machine m' instead of its currently matched machine m , and machine m' prefers to be matched to student s_t' instead of its matched student s_t .

The work in [39] introduced a matching game algorithm that can find a stable matching between two sets, which is called the differed acceptance matching algorithm. In this algorithm each student ranks the physical and virtual machines based on a utility function in order to obtain their preference relation, on the other side, each physical and virtual machine should also rank the students based on a utility function to obtain their preference relations.

The one-to-one matching game can be formulated by the tuple $(S_t, \mathcal{M}, \succ_{s_t}, \succ_{\mathcal{M}})$, where the first players set is the

students set \mathcal{S}_t , the second players set is the physical and virtual machines set \mathcal{M} , while $\succ_{\mathcal{S}_t}$, and \succ_m are the preference relations of the students' set and the machines set, respectively, such that $\succ_{\mathcal{S}_t} = \{\succ_{s_t}\}_{s_t \in \mathcal{S}_t}$, and $\succ_{\mathcal{M}} = \{\succ_m\}_{m \in \mathcal{M}}$. A matching between the students and the machines is denoted by $\mu : \mathcal{S}_t \rightarrow \mathcal{M}$, where μ is the matching game student's association outcome. Moreover, a utility function for each student is denoted by $U_{s_t}(m)$, which is used by the students to rank the physical and virtual machines, in order to build the preference relations \succ_{s_t} . On the other hand, the utility function for the physical and virtual machines are denoted by $U_m(s_t)$, which is used by the physical and virtual machines to rank the students, in order to build the preference relations \succ_m . The utility functions are described in the following subsections.

B. STUDENTS' UTILITY FUNCTION

In order to minimize the blockage rate $N_{s_t}^B$ in the optimization problem OPT, we designed a utility function for the students' that aims at maximizing the utilization of the available resources M_m for each machine m . The utility function $U_{s_t}(m)$ is formulated as follow:

$$U_{s_t}(m) = \min(N_r^{\mathcal{M}} : r \in M_m) - N_m^R, \forall m \in \mathcal{M}, M_m \supseteq E_e \quad (12)$$

In this utility function, the first term ($\min(N_r^{\mathcal{M}} : r \in M_m)$) is the minimum of the number of physical and virtual machines that has resource r , for all the resources on machine m . In other words, for each machine m , we find the minimum number $N_r^{\mathcal{M}}$ for all the resources on that particular machine. The aim of finding this minimum number $N_r^{\mathcal{M}}$ is that it reflects the most rarely resource r in machine m , and a student is preferred to be matched with a machine that has a higher rare value, instead of matching with a machine with a rare resource, which could block other students.

On the other hand, the second term (N_m^R) is the number of resources on physical or virtual machine m . This number reflects resources occupancy of machine m . A student is preferred to be matched with a machine that has the lowest resources occupancy, in order to leave the highest possible number of resources available for other students. This will provide a better utilization of the available resources.

C. PHYSICAL AND VIRTUAL MACHINES' UTILITY FUNCTION

In this subsection, we design an efficient physical and virtual machine utility function that serves the matching game algorithm in solving the optimization problem OPT. The utility function $U_m(s_t^e)$ is formulated as follow:

$$U_m(s_t^e) = -\min(N_r^{\mathcal{M}} : r \in E_e, e \rightarrow s_t), \forall s_t^e \in \mathcal{S}_t, M_m \supseteq E_e \quad (13)$$

A physical and virtual machine m rank the students based on the resources required by the experiments in which the

students are mapped to. For a particular resource r required by experiment e that student s_t should perform, the number $N_r^{\mathcal{M}}$ is the number of physical and virtual machines that has resource r . Thus, for a student s_t , if an experiment e requires multiple resources, machine m will consider the resource that has the minimum number $N_r^{\mathcal{M}}$, which indicates the rarity of that resource. The utility function $U_m(s_t^e)$ implies that machine m will prefer matching with the student that needs the rarest resource, which decreases the probability that this student is blocked, and will provide a better utilization of the available resources.

Algorithm 1 One-to-One Matching Game for Students Association

Initialization: $\mathcal{S}_t, \mathcal{M}$.

Discovery and utility function computation:

1: Every student s_t^e construct $\succ_{s_t^e}$ using $U_{s_t^e}(m)$

Find stable Matching:

2: **While** $\sum_{\forall s_t, m} b_{s_t \rightarrow m} \neq 0$ **do:**

3: For each unassociated student:

4: Find $m = \arg \max_{m \in \succ_{s_t}} U_{s_t^e}(m)$.

5: Send a request $b_{s_t \rightarrow m} = 1$ to machine m .

6: For all machines ($m \in \mathcal{M}$):

7: Update $\mathcal{S}_m^{req} \leftarrow \{s_t : b_{s_t \rightarrow m} = 1, s_t \in \mathcal{S}_t$.

8: Construct \succ_m based on $U_m(s_t^e)$.

9: Accept $s_t = \arg \max_{m \in \succ_m} U_m(s_t^e)$.

10: Update $\mathcal{S}_m^{rej} \leftarrow \{\mathcal{S}_m^{req} \setminus \mathcal{S}_m\}$.

11: Remove machines $m \in \succ_{s_t^e}, \forall s_t^e \in \mathcal{S}_m^{rej}$

12: **end while**

13: **Result:** A stable matching μ^* .

D. PROPOSED MATCHING GAME ALGORITHM

Algorithm 1 presents the details of the proposed matching game algorithm. After initialization, each student s_t use the utility function $U_{s_t}(m)$ in (12) to construct its preference relation \succ_{s_t} (step 1). Then it sends a bid request $b_{s_t \rightarrow m}$ to machine m that has the maximum $U_{s_t}(m)$ value (steps 2-5). On the other side, the physical and virtual machines update the requesting students set \mathcal{S}_m^{req} according to the bidding received (steps 6-7). Then, each machine constructs its preference relations \succ_m based on (13), and accepts the student with the highest $U_m(s_t^e)$ value (steps 8-9). After that, each machine sends a rejection to the not accepted bidding students according to the rejection list \mathcal{S}_m^{rej} (step 10). Finally, for each student received a rejection from machine m , he should remove machine m from his preference relation \succ_{s_t} . This process (steps 2-12) is repeated until there is no bidding request left.

VI. PERFORMANCE EVALUATION

A. SIMULATION SETUP

In order to evaluate the performance of the proposed laboratory management system, a simulation setup is used where 5 experiments are created, and each experiment is assigned a

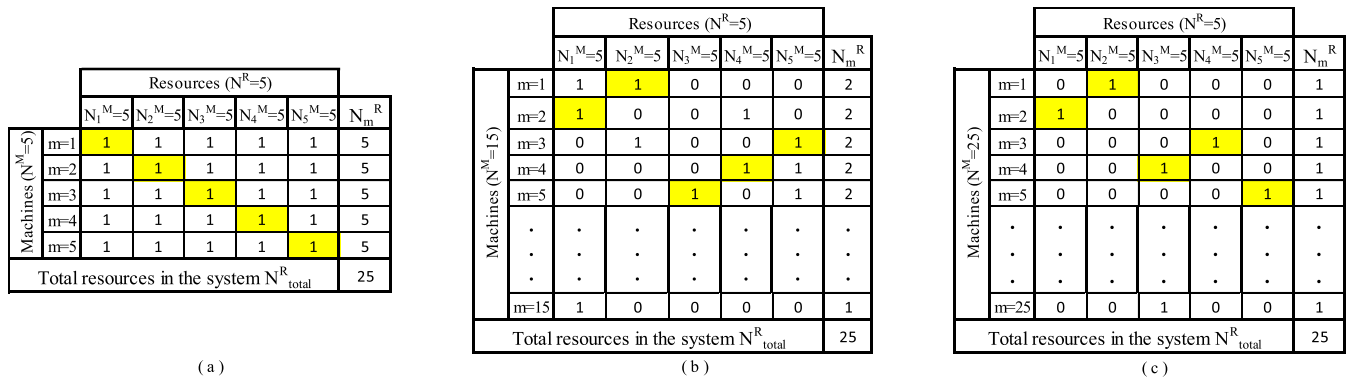


FIGURE 3. Resources to physical and virtual machines mapping. In (a), the total number of physical and virtual machines is $N^M = 5$. In (b), the total number of physical and virtual machines is $N^M = 15$. In (c), the total number of physical and virtual machines is $N^M = 25$. Note that, the total resources in the system $N_{total}^R = 25$, and the number of different resource types $N^R = 5$, and the number of resources used in each resource type N_i^M , are fixed in (a), (b), and (c).

unique resource randomly from the set of available resources. The students are randomly assigned one experiment from the created experiments in each iteration. In each simulation run, 1000 iterations are conducted, and average values of the iterations are calculated.

Moreover, two performance metrics are adopted to show the effectiveness of the proposed laboratory management system under different parameters changes. The first performance metric is the average percentage of blocked timeslots $N_{se}^B / N_e^{T_d}$, which is the ratio of the number of blocked timeslots to the total number of available timeslots, where $0 \leq N_{se}^B / N_e^{T_d} \leq 1$. The lower the percentage of blocked timeslots, the better the resource management scheme is.

The second performance metric is the average resource utilization efficiency, which is the ratio of the maximum number of students that can reserve in a specific timeslot to the total number of physical and virtual machines. Note that this performance metric also has a minimum of 0 and a maximum of 1. Whenever this ratio increases, this indicates the effectiveness of the used resource management scheme in utilizing the available resources.

In this context, and in order to evaluate the average percentage of blocking while increasing the number of students, we consider fixing the following parameters, the number of physical and virtual machines, the total number of timeslots, and the total number of resources in the system. On the other hand, in order to evaluate the average percentage of blocking while increasing the number of machines, we consider fixing the total number of students, the total number of timeslots, and the total number of resources in the system parameters. Furthermore, to evaluate the average resource utilization efficiency while increasing the number of physical and virtual machines, we consider fixing the total number of timeslots, and the total number of resources in the system parameters.

In addition, we fix the number of physical and virtual machines, and the total number of timeslots parameters in evaluating the average resource utilization efficiency while increasing the total number of resources in the system.

TABLE 1. Parameters values for different simulation runs.

Simulation Parameter	Value
Number of experiments N^E	5
Number of resources needed for an experiment $ E_e $	1
Percentage of blocking vs number of students	
Number of machines N^M	5, 15
Number of timeslots $N_e^{T_d}$	10
Total number of resources in the system N_{total}^R	25
Percentage of blocking vs number of machines	
Number of students N^S	40, 80
Number of timeslots $N_e^{T_d}$	10
Total number of resources in the system N_{total}^R	25
Resource utilization efficiency vs number of machines	
Number of timeslots $N_e^{T_d}$	1
Total number of resources in the system N_{total}^R	25
Resource utilization efficiency vs total number of resources in the system	
Number of machines N^M	5, 15
Number of timeslots $N_e^{T_d}$	1

A summary of the different parameters changes for each simulation run is presented in Table 1. Moreover, a detailed explanation of how each physical and virtual machine is assigned its resources is described in the next subsection.

B. RESOURCES TO PHYSICAL AND VIRTUAL MACHINES MAPPING

An important parameter that directly affects the performance of the laboratory management system and represents different case scenarios is the number of available physical and virtual machines, and how the total available resources in the system are mapped to it. In order to distribute and map

the available resources to a specific number of the available virtual and physical machines N^M , the total number of available resources N_{total}^R should be given. In each iteration, the number of resources N_r^M for each resource type r is assigned randomly, such that $\sum_{r=1}^{N^R} N_r^M = N_{total}^R$. Then, the resources of each resource type r is mapped randomly to the available physical and virtual machines, under the condition that each machine should be assigned at least one resource from the different resource types, and considering that there is always at least one combination where each resource type is mapped to a different machine, to reflect the possibility of conducting all the available experiments at the same timeslot.

It should be noticed that the total number of available resources N_{total}^R should always be greater than or equal to the number of virtual and physical machines N^M , in order to make sure that every machine is mapped at least to one resource. Besides, the total number of available resources N_{total}^R should not exceed the number of virtual and physical machines N^M multiplied by the number of available resource types N^R , to ensure that there are no extra resources wasted and not assigned to a machine. Thus, the following inequality must be satisfied:

$$N^M \leq N_{total}^R \leq N^M \times N^R \quad (14)$$

For illustration, Figure 3 shows three resources to physical and virtual machines mapping tables with different number of machines in each table. Note that, the values in each cell of the table is binary, to indicate whether a resource r is mapped to machine m or not, where 1 indicates a mapping between resource r to machine m , and 0 otherwise. The reason that the cell value should be binary, and does not have values more than 1, is that a resource can only be installed once on a physical or virtual machine (i.e., it is not logical to have more than one license for a specific software on a physical or virtual machine). In this figure, the total number of resources in the system N_{total}^R is fixed, and is equal to 25. Also, the number of available resource types N^R is fixed, and is equal to 5. Moreover, the number of resources N_r^M for each resource type r is fixed, and is equal to 5, such that $\sum_{r=1}^{N^R} N_r^M = N_{total}^R = 25$. The highlighted cells in each table show that there is always at least one combination where each resource type is mapped to a different machine, to ensure that there is at least one way to work all the experiments at the same time, and this condition must be considered while randomly assigning the number of resources for each resource type to the physical and virtual machines.

In Figure 3-a, the number of physical and virtual machines N^M is 5. This table represents the extreme scenario of (14) where $N_{total}^R = N^M \times N^R$. This scenario is the most cost-efficient scenario where all the physical and virtual machines are fully utilized, and each machine is mapped to all the available resource types. However, only 5 students, requiring any of the available experiments, can be associated with the available machines in one timeslot. By increasing

the number of physical and virtual machines N^M to be equal to 15 as shown in Figure 3-b, the number of resources for each resource type is mapped randomly to the available physical and virtual machines. Thus, each physical or virtual machine is assigned some of the available resource types. Although the number of machines increased, and accordingly the number of students that can be associated with machines in one timeslot increased, the resource types availability per machine decreases, hence, all the students cannot acquire the same experiment at the same timeslot. On the other side, increasing the number of machines increases the laboratory cost as well. By increasing the number of physical and virtual machines to the maximum to be $N^M = N_{total}^R = 25$ satisfying (14) as shown in Figure 3-c, the number of resources for each resource type is mapped randomly to the available physical and virtual machines, while considering that each machine should be mapped to at least one resource of the available resource types.

Thus, each machine will be mapped to only one resource type. In this scenario, the number of machines is maximized, hence, the number of students that can be associated with the available machines is also maximized, and this is considered the ideal scenario in terms of the associated number of students. However, the laboratory cost is also maximized, and the resource types availability per machine is decreased to the minimum. These different scenarios show the tradeoff between cost, and the resource types availability per machine, which if exploited by a well-designed management scheme, it can offer a significant performance enhancement.

C. SIMULATION RESULTS

In this subsection, we evaluate the performance of the proposed remote laboratory management system without the matching game framework, where a student's reservation to a physical or virtual machine for specific experiment is fixed (i.e., the student will be always associated to the same reserved machine until working the experiment). Then we will compare it with proposed remote laboratory management system while applying the matching game framework, where a student's reservation to a physical or virtual machine for specific experiment is dynamic (i.e., the student will be associated to any machine that is mapped to the needed resources for his experiment). Both the proposed remote laboratory management system with and without the matching game framework are compared to a third remote laboratory where each physical and virtual machine is assigned only one experiment, such that all the experiments are mapped to physical and virtual machines. In order to ease the comparing between the three remote laboratory management systems, the proposed remote laboratory management system without the matching game framework, which indicates a Fixed Student to Machine Reservation, will be denoted by (FSMR). The proposed remote laboratory management system with matching game framework, which indicates a Matching Game based Resource Association, will be denoted by (MGRA).

While the remote laboratory system where each physical and virtual machine is assigned only one experiment, which indicates a Fixed Resource to Machine Assignment, will be denoted by (FRMA).

In Figure 4, a comparison between the different remote laboratory management schemes in terms of the average resource utilization efficiency while changing the total number of resources in the system is presented. As shown, the proposed MGRA and FSMR management schemes outperform the FRMA scheme at $N^M = 5$ and $N^M = 15$. Also, it can be shown that the proposed MGRA has higher average resource utilization efficiency compared to the proposed FSMR management scheme at $N^M = 5$ and $N^M = 15$. This is because the FRMA scheme has a fixed physical and virtual machine to experiment assignment, thus, if the number of students acquiring an experiment is greater than the physical and virtual machines assigned to that experiment, only some of the students (equal to the number of machines assigned to that experiment) will be associated to the physical and virtual machines, while others will remain unassociated. On the other hand, the proposed MGRA and FSMR management schemes look for the needed resources of the acquired experiment by a student on all the available machines, thus, they are not bounded by a number of physical or virtual machines that are assigned to that experiment, they are bounded by the number of resources on all the machines that are acquired by the experiment.

Moreover, the proposed MGRA scheme outperforms the FSMR scheme because the MGRA scheme manages the students to machines association dynamically according to the experiments acquired by the students (i.e. it looks for the best association combination to fit all the students), while in the FSMR scheme, the student that comes first chooses the machine to be associated with, according to the machine's available resources and the needed resources by the experiment, and this association remains fixed.

Furthermore, it can be noticed that the FRMA scheme's average resource utilization efficiency does not change with the change in the total number of resources, and has almost similar values at a different number of physical and virtual machines. This is because increasing the total number of resources will increase the available resource types per machine, despite that, the FRMA scheme will assign a permanent experiment to a machine, and the available resource types per machine will not be useful in the students to machine association. On the other side, it can be noticed that the proposed MGRA and FSMR management schemes increase with the increasing total number of resources.

This is because increasing the total number of resources will increase the available resource types per machine, and the proposed schemes exploit the number of available resource types per machine, as it will increase resource types availability per machine, which in turn can handle a diverse students' experiments.

Figure 5 illustrates the effect of increasing the number of physical and virtual machines for the different schemes

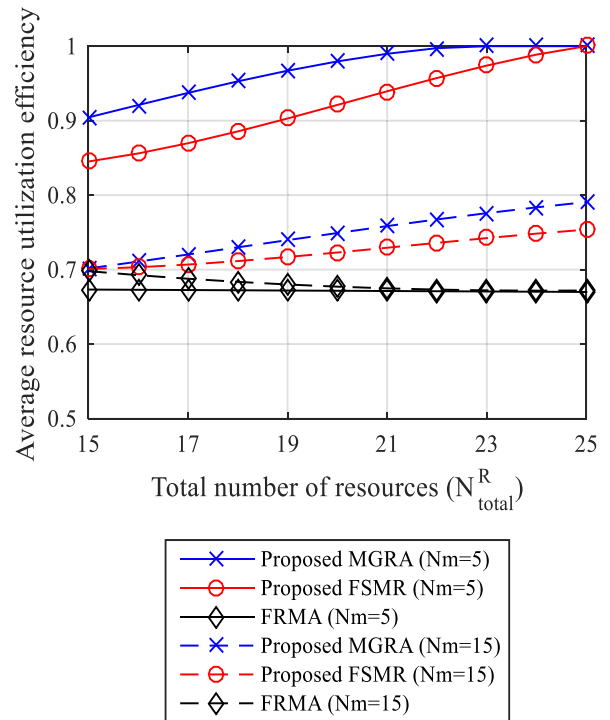


FIGURE 4. Average resource utilization efficiency for different remote laboratory management schemes with the change in the total number of resources at number of machines = 5 and number of machines = 15.

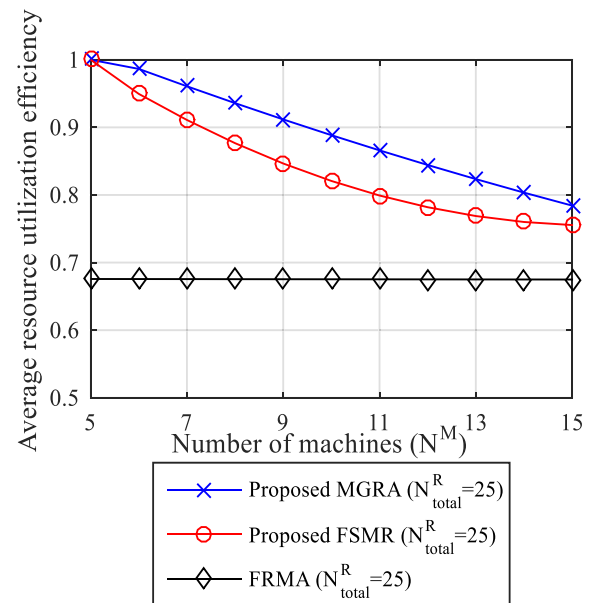


FIGURE 5. Effect of changing the number of physical and virtual machines for the different management schemes in terms of average resource utilization efficiency.

in terms of average resource utilization efficiency. It can be shown that both the proposed MGRA and FSMR management schemes outperform the FRMA scheme. This is because both schemes exploit the number of available resource types per machine, while the FRMA scheme has a fixed machine to experiment assignment, thus, wasting the number of available

resource types per machine. Moreover, it can be noticed that the FRMA scheme has an almost constant average resource utilization efficiency, this is because increasing the number of machines does not affect the resource utilization efficiency, since the number of students that should be associated increases as well. Also, it can be noticed that both the proposed MGRA and FSMR management schemes decreases with the increase of the number of machines, this because the total number of resources is constant and is equal to 25, and by increasing the number of machines, the resource type availability per machine decreases, until reaching the ideal scenario shown in Figure 3-c (when the number of machines reaches 25), at which all the schemes will achieve the same average resource utilization efficiency, as each machine will be assigned only one resource. Furthermore, the proposed MGRA scheme outperforms the proposed FSMR scheme because of the dynamic management of the students to physical and virtual machines association according to the required resources needed for the acquired experiment.

An interesting observation is that both the proposed MGRA and FSMR management schemes achieves the same and the highest average resource utilization efficiency when the number of machines is equal to 5. This is because the total number of resources is 25, and the number of resource types is 5, thus, each machine will have all the resource types available for it, as shown in Figure 3-a. In this situation the resource type availability per machine is at its maximum, thus, both schemes have an average resource utilization efficiency equal to 1.

In order to study the effect of increasing the number of students on the remote laboratory management schemes, Figure 6 shows the performance of the different management schemes, in terms of the average percentage of blocked timeslots at a different number of machines, while changing the number of students in the system. It is noteworthy that the total number of resources in the system is fixed at 25 for this simulation run. This figure shows that both the proposed MGRA and FSMR scheme outperform the FRMA scheme at a different number of machines. This is because of the dynamic experiment to machine assignment according to the needed resources by the experiment in the FSRMA, and the dynamic student to machine association according to the needed resources by the experiments acquired by all the students reserving the same timeslot in the MGRA scheme, compared to a static and fixed experiment to machine assignment in the FRMA scheme. For a number of machines equal to 5, and since the number of timeslots is 10, all the schemes reach a percentage of 100% at a number of students equal to 50, since all the machines in all the timeslots are reserved. The same happens when the number of machines is 15, and the number of students reaches 150.

Also, it can be noticed that both the proposed MGRA and FSMR management schemes have the same average percentage of blocked timeslots at a number of machines equal to 5 while increasing the number of students. This is because of the case scenario discussed in Figure 3-a, where the resource

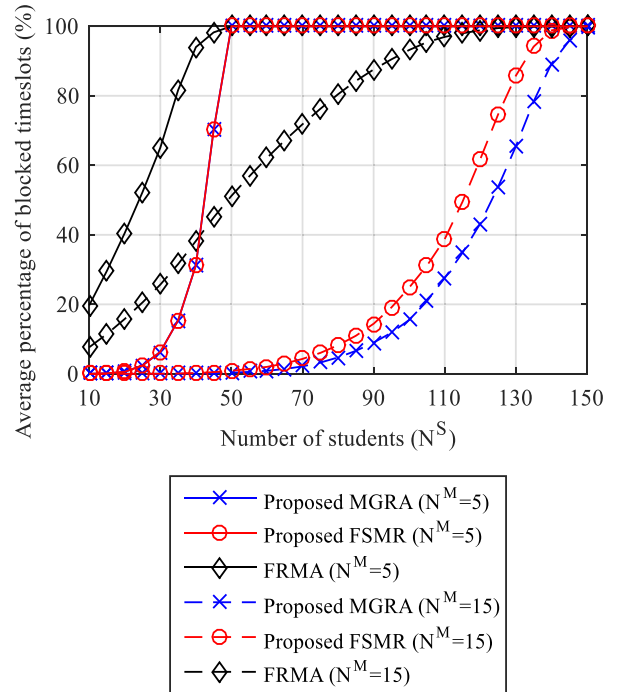


FIGURE 6. Average percentage of blocked timeslots for laboratory management schemes at different number of machines while changing the number of students.

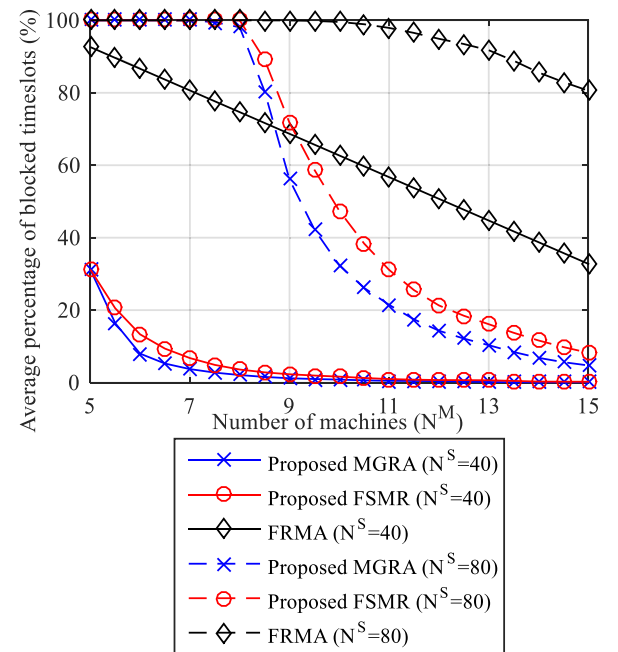


FIGURE 7. Average percentage of blocked timeslots for laboratory management schemes at different number of students while changing the number of machines.

type availability per machine is at its maximum, and all the resource types are available for each machine, thus both the proposed schemes achieve the same performance. For a number of machines equal to 15, the proposed MGRA achieves better performance compared to the proposed FSMR, where

TABLE 2. Percentage of improvement between different remote laboratory management schemes.

Performance metric	Changing parameter	Changing parameter value	Fixed parameter	Performance improvement percentage		
				Proposed FSMR and FRMA	Proposed MGRA and FRMA	Proposed MGRA and Proposed FSMR
Average resource utilization efficiency	Total resources in the system	$N_{total}^R=15$	$N^M=5$	26%	34%	7%
	Total resources in the system	$N_{total}^R=25$	$N^M=5$	49%	49%	0%
	Total resources in the system	$N_{total}^R=15$	$N^M=15$	0%	0%	0%
	Total resources in the system	$N_{total}^R=25$	$N^M=15$	12%	18%	5%
	Number of machines	$N^M=5$	$N_{total}^R=25$	49%	49%	0%
	Number of machines	$N^M=12$	$N_{total}^R=25$	16%	25%	8%
	Number of machines	$N^M=15$	$N_{total}^R=25$	12%	18%	5%
Average percentage of blocked timeslots	Number of students	$N^S=10$	$N^M=5$	100%	100%	0%
	Number of students	$N^S=35$	$N^M=5$	81%	81%	0%
	Number of students	$N^S=50$	$N^M=5$	0%	0%	0%
	Number of students	$N^S=10$	$N^M=15$	100%	100%	0%
	Number of students	$N^S=125$	$N^M=15$	25%	46%	28%
	Number of students	$N^S=150$	$N^M=15$	0%	0%	0%
	Number of machines	$N^M=5$	$N^S=40$	66%	66%	0%
	Number of machines	$N^M=6$	$N^S=40$	85%	91%	40%

the resource type availability per machine decreases, hence, the students to machines association optimization done in the MGRA scheme shows significant performance improvement.

Furthermore, the effect of increasing the number of machines on the performance of the different remote laboratory management schemes in terms of the average percentage of blocked timeslots at a different number of students in the system is studied in Figure 7. It is noteworthy to know that the total number of resources in the system is 25 in this simulation run.

Generally, it can be shown that by increasing the number of machines the average percentage of blocked timeslots decreases for all the remote laboratory management schemes, since the machines' availability increases, hence increasing the capacity of students to machine association at one timeslot. Also, it can be noticed that the proposed MGRA scheme achieves better performance at a number of students equal to 40 and 80, this is because of the students to machines association optimization used in the MGRA scheme to minimize the number of blocked timeslots.

Moreover, it can be noticed that both the proposed MGRA and FSMR schemes achieves the same performance at a number of machines equal to 5, this is due to the scenario described in Figure 3-a, where the resource type availability per machine is at its maximum. It can also be shown that all the schemes achieve an average percentage of blocked timeslots equal to 100%, when the number of students is 80, until the number of machines reaches 8. This is because

when the number of machines is less than 8, the total number of machines available for reservation in all the 10 available timeslots is less than the number of students in the system, therefore, there will always be no timeslots available.

In order to precisely compare the different laboratory management schemes, Table 2 presents the performance improvement percentage of the proposed MGRA and FSMR schemes compared to the FRMA scheme, and the improvement percentage of the proposed MGRA and FSMR schemes themselves, considering different parameter values. The table shows that the performance improvement percentage of both the proposed MGRA and FSMR schemes compared to the FRMA scheme can reach up to 49% in the average resource utilization efficiency metric, when the total number of resources is 25 and the number of machines is 5. This is because of the case scenario discussed in Figure 3-a. Also, their performance improvement percentage can reach up to 100% in the average percentage of blocked timeslots metric, when the number of students is 10 and the number of machines is 5 and 15. This is because the number of students is low, and the proposed laboratory management schemes can achieve a very low average percentage of blocked timeslots, while due to the static nature of the FRMA scheme, it could not achieve such a low average percentage of blocked timeslots.

Moreover, Table 2 shows that the proposed MGRA scheme has a performance improvement percentage that can reach 8% compared to the proposed FSMR scheme in the average

TABLE 3. Comparing the proposed MGRA with the optimum solution in terms of average percentage of blocked timeslots.

Optimality gap for the proposed MGRA			
	$N^S = 40$	$N^S = 80$	$N^S = 120$
$N^M = 5$	0%	0%	0%
$N^M = 7$	0.1%	0.9%	0%
$N^M = 9$	0%	0.9%	0.2%
$N^M = 11$	5%	10.9%	11%

resource utilization efficiency metric, and the performance improvement percentage that can reach 40% in the average percentage of blocked timeslots. This is because the MGRA scheme can manage the students to machines association dynamically as discussed before.

Furthermore, Table 2 presents some special cases, such as in the average resource utilization efficiency, when the total number of resources is 15 and the number of machines is 15, where both the proposed MGRA and the FSMR schemes cannot achieve any performance improvement. This case represents the scenario discussed in Figure 3-c, where each resource has a unique machine to serve it, in which the FRMA scheme is in its ideal case, however, this scenario is costly compared to others, since each virtual or physical machine has only one resource to operate. Also, the case in the average resource utilization efficiency, when the total number of resources is 25 and the number of machines is 5, where the proposed MGRA scheme cannot achieve any performance improvement compared to the proposed FSMR scheme. This case represents the scenario discussed in Figure 3-a, where all the machines operate all the available resources, thus, there will be no advantage of the MGRA scheme over the FSMR scheme. Also, the cases in the average percentage of blocked timeslots, when the number of machines is 5 and 15, while the number of students is 50 and 150, respectively. In both cases, there is no performance improvement for the MGRA or the FSMR over the FRMA. This is because all the laboratory management schemes reach their maximum student's occupation, since in the first case, there are 5 machines and 10 available timeslots, thus the maximum occupation is 50, and in the second case, there are 15 machines and 10 available timeslots, thus the maximum occupation is 150.

Finally, to evaluate the performance of the proposed MGRA algorithm in solving the proposed optimization problem, we compare it with surrogate optimization algorithm, that attempts to find a global solution for expensive integer programming optimization problems, using a few objective function evaluations [43]. Table 3 shows the optimality gap by comparing the MGRA with the optimum solution, which is computed by the surrogate optimization algorithm. The comparison is in terms of average percentage of blocked timeslots. It can be shown that, with the low complexity of the proposed MGRA algorithm, it achieves a near optimal solutions for most of the cases, and it can reach the optimal solutions in some scenarios. Also, it can be noticed that the optimality gap reaches 11% at some instants, where the number of students increases, and the number of machines

increases as well. On the other side, the complexity of finding the optimal solution at these scenarios increases dramatically, and becomes untraceable, compared to the proposed matching theory algorithm complexity [42].

VII. CONCLUSION

In this paper, we proposed a laboratory learning system that provides flexibility, integrability, and scalability for deployment on multiple different physical infrastructures. Moreover, we formulated a mathematical model for the laboratory learning system, which is followed by formulating an optimization problem that aims at minimizing the number of blocked time slots per acquiring student, by optimizing the association between the reserving students and the physical and virtual machines at different time slots, which reflects the maximization of the resources utilization. Then, a matching game-based framework was developed to optimally solve the aforementioned problem.

The performance of the proposed matching game framework was investigated. Simulation results showed the significance of the proposed matching game framework compared to other schemes on resources utilization efficiency, and the number of blocked time slots per acquiring student.

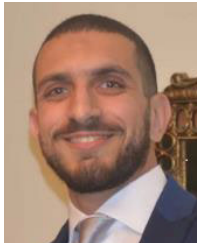
REFERENCES

- [1] S. B. Nolen and M. D. Koretsky, "Affordances of virtual and physical laboratory projects for instructional design: Impacts on Student engagement," *IEEE Trans. Educ.*, vol. 61, no. 3, pp. 226–233, Aug. 2018.
- [2] I. Chirikov, T. Semenova, N. Maloshonok, E. Bettinger, and R. F. Kizilcec, "Online education platforms scale college STEM instruction with equivalent learning outcomes at lower cost," *Sci. Adv.*, vol. 6, no. 15, pp. 1–11, Apr. 2020.
- [3] W. G. Maroquin, M. Fernandez, and W. M. Avila, "The joint training, a SENA learning model for Latin America," *IEEE Latin Amer. Trans.*, vol. 14, no. 6, pp. 2997–3002, Jun. 2016.
- [4] F. Schauer, M. Krbecsek, P. Beno, M. Gerza, L. Palka, and P. Spilakova, "REMLABNET II—Open remote laboratory management system for university and secondary schools research based teaching," in *Proc. 12th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, Feb. 2015, pp. 109–112.
- [5] S. AbuShanab, M. Winzker, R. Bruck, and A. Schwandt, "A study of integrating remote laboratory and on-site laboratory for low-power education," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2018, pp. 405–414.
- [6] A. Maiti, D. G. Zutin, H.-D. Wuttke, K. Henke, A. D. Maxwell, and A. A. Kist, "A framework for analyzing and evaluating architectures and control strategies in distributed remote laboratories," *IEEE Trans. Learn. Technol.*, vol. 11, no. 4, pp. 441–455, Oct. 2018.
- [7] L. D. L. Torre, L. T. Neustock, G. K. Herring, J. Chacon, F. J. G. Clemente, and L. Hesselink, "Automatic generation and easy deployment of digitized laboratories," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7328–7337, Dec. 2020.
- [8] G. S. Machado, Y. M. L. R. Silva, and V. F. de Lucena, "A tool for the automatic selection of mechatronics remote laboratories based on their actual effective costs," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2018, pp. 1–5.
- [9] N. M. Mamani, F. J. Garcia-Penalvo, M. A. Conde, and J. Goncalves, "A systematic mapping about simulators and remote laboratories using hardware in the loop and robotic: Developing STEM/STEAM skills in pre-university education," in *Proc. Int. Symp. Comput. Educ. (SICE)*, Malaga, Spain, Sep. 2021, pp. 23–24.
- [10] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.
- [11] V. J. Harward et al., "The iLAB shared architecture: A web services infrastructure to build communities of internet accessible laboratories," *Proc. IEEE*, vol. 96, no. 6, pp. 931–950, Jun. 2008.

- [12] D. Lowe, S. Conlon, S. Murray, L. Weber, M. De La Villefromoy, E. Lindsay, A. Nafalski, W. Nageswaran, and T. Tang, "LabShare: Towards cross-institutional laboratory sharing," in *Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Disciplines*, A. Azad, M. Auer, J. Harward, Eds., 1st ed. Hershey, PA, USA: IGI Global, 2012, pp. 453–467.
- [13] L. F. Z. Rivera and M. M. Larrondo-Petrie, "Design of a Latin American and Caribbean remote laboratories network," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2016, pp. 1–5.
- [14] R. Heradio, L. de la Torre, and S. Dormido, "Virtual and remote labs in control education: A survey," *Annu. Rev. Control*, vol. 42, pp. 1–10, Aug. 2016.
- [15] Z. Lei, H. Zhou, W. Hu, Q. Deng, D. Zhou, and Z.-W. Liu, "HTML5-based 3-D online control laboratory with virtual interactive wiring practice," *IEEE Trans. Ind. Inform.*, vol. 14, no. 6, pp. 2473–2483, Jun. 2018.
- [16] Z. Lei, H. Zhou, W. Hu, Q. Deng, D. Zhou, Z.-W. Liu, and J. Lai, "Modular web-based interactive hybrid laboratory framework for research and education," *IEEE Access*, vol. 6, pp. 20152–20163, 2018.
- [17] A. Maiti, A. A. Kist, and A. D. Maxwell, "Real-time remote access laboratory with distributed and modular design," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3607–3618, Jun. 2015.
- [18] L. Tobarra, R. Pastor, R. Hernández, M. Castro, A. Al-Zoubi, M. Dmour, A. Robles-Gomez, A. C. Caminero, and J. Cano, "An integrated example of laboratories as a service into learning management systems," *Int. J. Online Eng.*, vol. 12, no. 9, pp. 32–39, Sep. 2016.
- [19] M. Tawfik, C. Salzmann, D. Gillet, D. Lowe, H. Saliyah-Hassane, E. Sancristobal, and M. Castro, "Laboratory as a service (LaaS): A novel paradigm for developing and implementing modular remote laboratories," *Int. J. Online Eng.*, vol. 10, no. 4, pp. 13–21, Jun. 2014.
- [20] Y. Verbelen, P. Taelman, A. Braeken, and A. Touhafi, "Reconfigurable and modular mobile robotics platform for remote experiments," *Int. J. Online Eng.*, vol. 9, no. 3, pp. 19–26, Jul. 2013.
- [21] A. Bagnasco, G. Parodi, D. Ponta, and A. M. Scapolla, "A modular and extensible remote electronic laboratory," *Int. J. Online Eng.*, vol. 1, no. 1, pp. 1–6, 2005.
- [22] G. Esposito, D. Mezzogori, D. Reverberi, G. Romagnoli, M. Ustenko, and F. Zammori, "Non-traditional labs and lab network initiatives: A review," *Int. J. Online Biomed. Eng.*, vol. 17, no. 5, pp. 4–32, 2021.
- [23] P. Orduña, P. H. Bailey, K. DeLong, D. López-de-Ipiña, and J. García-Zubia, "Towards federated interoperable bridges for sharing educational remote laboratories," *Comput. Hum. Behav.*, vol. 30, pp. 389–395, Jan. 2014.
- [24] D. Lowe, "MOOLS: Massive open online laboratories: An analysis of scale and feasibility," in *Proc. 11th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, Feb. 2014, pp. 1–6.
- [25] P. Orduna, L. Rodriguez-Gil, D. Lopez-de-Ipina, and J. Garcia-Zubia, "Sharing the remote laboratories among different institutions: A practical case," in *Proc. 9th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, Bilbao, Spain, Jul. 2012, pp. 1–4.
- [26] T. Richter, D. Boehringer, and S. Jeschke, "LiLa: A European project on networked experiments," in *Proc. 6th Int. Conf. Remote Eng. Virtual Instrum.*, 2009, pp. 307–317.
- [27] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the art about remote laboratories paradigms-foundations of ongoing mutations," *Int. J. Online Eng.*, vol. 4, no. 1, p. 19, 2008.
- [28] P. Orduña, "Scheduling schemes among internet laboratories ecosystems," in *Proc. 8th Int. Conf. Remote Eng. Virtual Instrum.*, 2011, pp. 1–6.
- [29] K. Henke, J. Nau, R. N. Bock, and H. D. Wuttke, "A hybrid online laboratory for basic STEM education," in *Smart Education and E-Learning 2021*, vol. 240. Cham, Switzerland: Springer, 2021, pp. 29–39.
- [30] E. Taoufik, C. Sahbi, and S. Wafik, "Quality assurance for remote-lab systems by new reporting tool," in *Proc. Int. Conf. Remote Eng. Virtual Instrum.*, in *Lecture Notes in Networks and Systems*, vol. 298. Cham, Switzerland: Springer, Feb. 2021, pp. 237–243.
- [31] A. Bekasiewicz, B. Pankiewicz, M. Wojcikowski, M. Klosowski, and S. Koziel, "Application of open-hardware-based solutions for rapid transition from stationary to the remote teaching model during pandemic," *IEEE Trans. Educ.*, vol. 64, no. 3, pp. 299–307, Aug. 2021.
- [32] Z. Lei, H. Zhou, W. Hu, and G.-P. Liu, "Unified and flexible online experimental framework for control engineering education," *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 835–844, Jan. 2022.
- [33] F. Schauer, M. Krbecsek, P. Beno, M. Gerza, L. Palka, and P. Spilakova, "REMLABNET—Open remote laboratory management system for e-experiments," in *Proc. 11th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, Feb. 2014, pp. 268–273.
- [34] A. Villar-Martínez, L. Rodríguez-Gil, I. Angulo, P. Orduña, J. García-Zubia, and D. López-De-Ipiña, "Improving the scalability and replicability of embedded systems remote laboratories through a cost-effective architecture," *IEEE Access*, vol. 7, pp. 164164–164185, 2019, doi: 10.1109/ACCESS.2019.2952321.
- [35] P. Wieder, O. Waldrich, and W. Ziegler, "Advanced techniques for scheduling, reservation, and access management for remote laboratories," in *Proc. 2nd IEEE Int. Conf. E-Sci. Grid Comput.*, Dec. 2006, pp. 128–134.
- [36] M. Domínguez, J. J. Fuertes, M. A. Prada, S. Alonso, and A. Morán, "Remote laboratory of a quadruple tank process for learning in control engineering using different industrial controllers," *Comput. Appl. Eng. Educ.*, vol. 22, no. 3, pp. 375–386, Sep. 2014.
- [37] J. M. Andujar, A. Mejias, and M. A. Marquez, "Augmented reality for the improvement of remote laboratories: An augmented remote laboratory," *IEEE Trans. Educ.*, vol. 54, no. 3, pp. 492–500, Aug. 2011.
- [38] F. Y. Limpraptono, E. Nurcahyo, and A. Faisol, "The development of electronics telecommunication remote laboratory architecture based on mobile devices," *Int. J. Online Biomed. Eng.*, vol. 17, no. 3, p. 26, Mar. 2021.
- [39] D. Lowe, "Integrating reservations and queuing in remote laboratory scheduling," *IEEE Trans. Learn. Technol.*, vol. 6, no. 1, pp. 73–84, Jan. 2013.
- [40] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.
- [41] M. Anany, M. M. Elmesalawy, and A. M. A. El-Haleem, "Matching game-based cell association in multi-RAT HetNet considering device requirements," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9774–9782, Dec. 2019.
- [42] A. I. Abdulshakoor, M. G. Anany, and M. M. Elmesalawy, "Outage-aware matching game approach for cell selection in LTE/WLAN multi-RAT HetNets," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107596.
- [43] Y. Wang and C. A. Shoemaker, "A general stochastic algorithmic framework for minimizing expensive black box objective functions based on surrogate models and sensitivity analysis," 2014, *arXiv:1410.6271*.



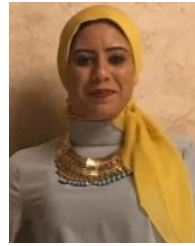
AHMED M. ABD EL-HALEEM received the B.Sc., M.Sc., and Ph.D. degrees in electronics and communications engineering from Helwan University, Cairo, Egypt, in 2001, 2006, and 2012, respectively. He is currently an Associate Professor with the Department of Electronics and Communications Engineering, Faculty of Engineering, Helwan University. He is also a Secondment with the Department of Electrical Engineering, Faculty of Engineering, British University, Egypt. He has supervised bachelor's and master's students and worked on projects in areas related to device-to-device communication, the Internet of Things (IoT), and software-defined networks (SDN), including mobility and resource management. His current research interests include mobile/vehicular ad-hoc communication networks, 5G and 6G radio access networks, cognitive radio networking, device to device communication, the Internet of Things (IoT), reconfigurable intelligent surface (RIS), and AI application in wireless communication. This includes mobility management techniques and routing schemes for mobile ad-hoc networks (MANET), security and secure routing algorithms for ad-hoc wireless networks, and routing protocols for cognitive radio communication networks. His research was well received and has been published in several prestigious journals and conferences in the field of electrical and communication engineering, including IEEE, Springer, the *International Journal of Network Security and Its Applications* (IJNSA), the *International Journal of Corrosion and Scale Inhibition* (IJCSI), and Hindawi and Wiley journals with more than 27 journal and conference papers. He is a member of a research teams that awards several applied research projects funded by the national and international funding agencies in the field of wireless communication, the IoT and its applications, and smart education systems.



MOHAMED G. ANANY was born in Asyut, Egypt. He received the dual B.Sc. degrees in electronics and communication engineering from the Canadian International College (CIC), Cairo, Egypt, and in electronics and control engineering from Cape Breton University (CBU), Canada, in 2011, and the dual M.Sc. degrees in electronics and communications engineering from Helwan University, Cairo, and in systems engineering and engineering management from the South Westphalia University of Applied Sciences (SOEST), Germany, in 2017. He is currently pursuing the Ph.D. degree in electronics and communications engineering with Helwan University. Since 2017, he has been an Assistant Lecturer with the Department of Electronics and Communication, CIC. His research interests include 5G, B5G, and 6G access networks, the IoT, unmanned aerial vehicle (UAV) communications, AI-based applications in wireless communication systems, and WLAN resource management. He is a member of the research team that awarded several applied research projects funded by national and international funding agencies in the field of wireless communication, the Internet of Things (IoT) and its applications, and smart education systems.



MAHMOUD M. ELMESALAWY (Member, IEEE) was born in Cairo, Egypt, in 1981. He received the B.S., M.S., and Ph.D. degrees in electronics and communications engineering from Helwan University, Cairo, in 2002, 2005, and 2010, respectively. He is currently working as a Professor of wireless communications with the Department of Electronics and Communications Engineering and the Dean of the Faculty of Engineering, Helwan University. He led the effort to build the cellular-based communication infrastructure of the Egyptian wide area monitoring system (EWAMS) that is developed on the Egyptian grid and operated by Helwan University. His current research interest includes B5G and 6G mobile communication technologies. This includes the Internet of Things (IoT) communications, unmanned aerial vehicle (UAV) communications, reconfigurable intelligent surface (RIS), smart grid communication and AI-based applications in wireless communication systems.



EMAN SERAJ EL-DIN BAKR was born in Cairo, Egypt. She received the Ph.D. degree from the Faculty of Engineering, Ain-Shams University, in 2013. She is currently an Assistant Professor with the Department of Electronics and Communications Engineering, Faculty of Engineering, Canadian International College (CIC), Egypt. She is also a Doctor Engineer with the Networks Planning Department, National Telecommunication Institute, Ministry of Communication and Information Technology, Cairo. Her research interests include modern mobile wireless communications networks, networking, performance evaluation techniques, quality of services techniques, optimization, modeling, planning techniques. Her current research interests include B5G and 6G mobile communication technologies. This includes the Internet of Things (IoT) communications, automotive communications systems, digital transformation systems, and AI-based applications in wireless communication systems.

...