Correspondence

Solving the Group Multirole Assignment Problem by Improving the ILOG Approach

Haibin Zhu, *Senior Member, IEEE*, Dongning Liu, Siqin Zhang, Shaohua Teng, and Yu Zhu

***Abstract*—Role assignment is a critical element in the role-based collaboration process. There are many different requirements to be considered when undertaking this task. This correspondence paper formalizes the group multirole assignment (GMRA) problem; proves the necessary and sufficient condition for the problem to have a feasible solution, provides an improved IBM ILOG CPLEX optimization package solution, and verifies the proposed solution with experiments. The contributions of this paper include: 1) the formalization of an important engineering problem, i.e., the GMRA problem; 2) a theoretical proof of the necessary and sufficient condition for GMRA to have a feasible solution; and 3) an improved ILOG solution to such a problem.**

***Index Terms*—Agents, group role assignment (GRA), IBM ILOG CPLEX optimization (ILOG) package, roles.**

I. Introduction

Role assignment has been revealed as a complex process throughout the life cycle of role-based collaboration (RBC) [24], i.e., agent evaluation, role assignment, role playing, and role transfer. Group role assignment (GRA) seeks an optimal role-to-agent assignment based on the results of agent evaluations [25] and greatly affects collaboration efficiency and the degree of satisfaction of members involved in RBC.

GRA is itself a complex problem where the exhaustive search algorithm has an exponential increase in complexity. In our previous work, an efficient algorithm was developed by using the Hungarian algorithm (also called Kuhn–Munkres algorithm [8], [10]) [26].

The GRA solution solves a role assignment problem [26] considering the following constraints: 1) the number of agents required for each role and 2) only one role assigned to an agent.

After the GRA problem is solved, role assignment becomes a straightforward process if there are no additional variations and constraints. However, there can be many variations in role assignment. Group multirole assignment (GMRA) is such a variation, and by GMRA, we mean assigning a limited number of different roles to each agent within the scenario of GRA.

GMRA is, in fact, a GRA problem while considering the current role and potential roles together, based on our previous work in solving role transfer problems [20]–[26].

GMRA is common in the real-world. For example, in a team, different people (agents) may be assigned with a limited number of jobs (roles). In a university, a professor (agent) is assigned with at most three different courses (roles) to teach and a student (agent) can register at most six (of course, different) courses (roles) in a semester. In transportation, a group of cargo (agents) may be moved by a limited number of different containers (roles). In services computing, a server (agent) may provide a limited number of different services (roles).

Similar to the GRA problem, the GMRA problem is a complex problem. A primary objective of this correspondence paper is to provide a practical solution to the GMRA problem. Although such problems arise in the research of RBC, they are also important and challenging in the domains of administration, production, and engineering.

The contributions of this paper include the following.

1) The concise formalization of the GMRA problem that is easily categorized as an extended integer linear programming (x-ILP) problem [12], [13], [17], [20].
2) A theoretical proof of the necessary and sufficient condition for GMRA to have a feasible solution.
3) An improvement to the IBM ILOG CPLEX optimization package (ILOG) solution by applying the proved theory.

This paper is organized as follows. It first discusses the related work in Section II, Then, it describes a real-world scenario related to the proposed problem in Section III and formally specifies the GMRA problem with the revised Environments-Classes, Agents, Roles, Groups, and Objects (E-CARGO) model in Section IV. Section V presents an ILOG solution, and Section VI explains the drawback of the ILOG solution, proves the necessary and sufficient conditions for the GMRA problem to have a feasible solution, and implements the improved ILOG solution. Comparisons between the improved ILOG solution and the ILOG one are illustrated in Section VII. Section VIII describes another real world example to inform potential applications of the proposed solution. This paper concludes and points out future works in Section IX.

II. Related Work

Although GMRA is an important and challenging problem in organizational performance [5], system construction [14], [15] and system management [5], [14]. To the authors' knowledge, there is a lack of fundamental and comprehensive research on this issue.

Manuscript received August 6, 2015; revised January 19, 2016; accepted April 30, 2016. Date of publication May 31, 2016; date of current version November 16, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council, Canada, under Grant RGPIN262075-2013 and in part by the National Natural Science Foundation of China under Grant 61402118 and Grant 61370229. This paper was recommended by Associate Editor L. Cao. *(Corresponding author: Dongning Liu.)*

H. Zhu is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China, and also with the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

D. Liu, S. Zhang, and S. Teng are with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: liudn@gdut.edu.cn; 1032235096@qq.com; shteng@gdut.edu.cn).

Y. Zhu is with the Faculty of Mathematics, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: davisyzhu@yahoo.ca).

Digital Object Identifier 10.1109/TSMC.2016.2566680

Some related research work focuses on assignment problems in different applications [2], [6], [9], [19], and role assignment for agents in multiagent systems [2], [5], [11] or nodes in networked systems [1].

At first, it is valuable to mention that [2] is a textbook on assignment problems. Such a book presents and discusses all the discovered typical assignment problems before 2009 in the conventional research. Not surprisingly, our proposed GMRA problem is not covered in this paper. This result unveils a fact that RBC provides an innovative way to investigate abstract assignment problems in the sense of collaboration based on the E-CARGO model.

Bhardwaj and Chandrakasan [1] presented a real-world problem that requires role assignment. They introduce a framework for role assignment and use the framework to obtain the bounds of the lifetime of a sensor network for a variety of data gathering scenarios. Such work can be categorized as agent evaluation in RBC [20], [24], [26], and then can be used in role assignment algorithms.

Dastani *et al.* [3] investigated the determination of conditions under which an agent can adopt a role and the meanings for the agent to play a role. They define the relationships between roles and agents and then discuss architectural and functional changes that an agent must deal with when the agent enters an open system. Their work proposes a framework to support role negotiation and agent evaluation in RBC.

Durfee *et al.* [4] proposed a formulation of the team formation by modeling the assignment and scheduling of expert teams as a hybrid scheduling problem. Their work clarifies the significance and complexity of the problem of assignment and scheduling in the team organization of experts. It also illustrates that the assignment problem in teams can be described as an ILP or a mixed integer linear programming problem and that such a team formation problem can be solved by a standard mathematical programming package such as ILOG used in this paper.

Hui and Zhang [6] presented a new control-theoretic framework to design efficiently balanced coordinated resource allocation algorithms in a network based on semi-stabilization theory for discrete-time stochastic linear systems together with compartmental modeling. They convert this optimal control-based design problem into a constrained, nonlinear optimization problem to look for possible numerical solutions to the original balanced coordinated resource allocation problem. They also proposed a class of randomized swarm optimization-based numerical algorithms called multiagent coordination optimization to solve the constrained, nonlinear optimization problem. Their work demonstrates that the investigation of assignment problems is valuable to engineering applications.

Li *et al.* [9] discussed the problem of intercell scheduling with single processing machines and batch processing machines. This problem involves an assignment subproblem, a sequencing subproblem, and a batch formation subproblem. They propose a solution based on an ant colony optimization-based hyperheuristic. They claim that their solution is better than the one based on the IBM ILOG CPLEX platform. Their research method is similar to that of the authors of this paper, i.e., to propose a solution to a problem in the category of problems that can be solved by a general optimization platform and compare the new solution with the one based on the platform.

Odell *et al.* [11] pointed out that the roles played by an agent may change over time. They describe a case study where such role changes are required, analyzing and classifying the various kinds of role changes that may occur. Their contribution focuses on the third step of role assignment, i.e., role transfer.

Shen *et al.* [14] presented a multicriteria assessment model capable of evaluating the suitability of individual workers for a specified task according to their capabilities, social relationships, and existing tasks. Candidates are ranked based on their suitability scores to support workflow administrators in selecting appropriate workers to perform the tasks assigned to a given role. Their work can be applied into making the Q matrix for GRA, i.e., agent evaluation [20], [24], [26].

TABLE I
REQUIRED POSITIONS

Position	Project Manager	System Analyst	Software Developer	Tester
Required Number	1	2	4	2

Stone and Veloso [15] took periodic team synchronization domains as time-critical environments in which agents act autonomously. They state that dynamic role adjustment (the third step, i.e., role transfer [25]) makes possible formation changes allowing for a group of agents to collaborate. They verify their method using a soccer team of robots.

Wolsey and Nemhauser [17] extended Stone and Veloso's work [15] in role assignment and coordination in multirobot systems, especially in highly dynamic tasks. They propose an approach to sharing sensed information and effective coordination through the introduction of shared potential fields. Such fields are based on positions (roles) of other robots on the team and the ball. Robots position themselves on a field by following the gradient to a minimum of the potential field.

Zhang [18] proposed a teamwork language role-based multiagent logic language for encoding teamwork to support multiagent collaboration. She designs rules and related algorithms to regulate the selection of roles and the assignment of roles to agents. Her major objective is to provide a technique for task assignment based on decision trees.

Zheng *et al.* [19] provided an analysis and practical solution to the problem of designing and implementing a human–robot team for simple conversational interactions. They propose models for operation timing, customer satisfaction, and customer–robot interaction. Based on these models, they introduce techniques for managing interaction flow and operator task assignment. This paper informs the importance of role (task) assignment in collaboration.

The research above indicates a strong need to investigate fundamentally role assignments including the problem discussed in this paper.

III. REAL-WORLD SCENARIO

In company X, Ann, the Chief Executive Officer, has just signed a contract the value of which is half million dollars. She asks Bob, the Human Resources (HR) officer, to organize a team from employees of the company. Bob drafts a position list shown as Table I for the team and a candidate staff shortlist shown as Table II. Then, Bob initiates an evaluation process and asks the branch officers to evaluate employees for each possible position (Table II). After that, Bob informs Ann that he must assign some people to more than one position because the total number of staff members is less than that of the required positions. Ann agrees with Bob but asks Bob that if someone must be assigned with more than one job, then the jobs must be different and each person should be limited to at most three to avoid too much overloaded. Ann's demand is reasonable because one person can shift among different jobs but not the same ones and each person should work on limited jobs to guarantee the quality of their work. Based on Ann's request, Bob composes a table (Table III) to express the limits. Then, Ann instructs Bob to assign the most qualified candidates to jobs while satisfying the mentioned conditions. After some consideration, Bob suggests that a satisfactory solution, in light of such a challenge, may require a significant amount of time. Fortunately, Ann, as an experienced administrator, understands the complexity of the problem and does not demand an unreasonable response timeframe.

TABLE II
CANDIDATES AND POSITION EVALUATIONS

Positions / Candidates	Project Manager	System Analyst	Software Developer	Tester
Adam	0.18	**0.82**	0.29	0.01
Brian	0.35	0.80	**0.58**	0.35
Chris	0.84	**0.85**	**0.86**	0.36
Doug	0.96	0.51	**0.45**	**0.64**
Edward	0.22	0.33	**0.68**	0.33
Fred	**0.96**	0.50	0.10	**0.73**

TABLE III
LIMITED NUMBERS OF JOBS

Names	Adam	Brian	Chris	Doug	Edward	Fred
Limited Number of Jobs	1	2	3	2	2	2

From the above scenario, Ann and Bob follow the initial steps of RBC and Bob encounters a variation of the GRA problem that considers both current and potential roles [20]–[22]. The final optimized solution, is shown as the bold and underlined ones in Table II, i.e., a tuple set as: {<Adam, {System Analyst}>, <Brian, {Software Developer}>, <Chris, {System Analyst, Software Developer}>, <Doug, {Software Developer, Tester}>, <Edward, {Software Developer}>, <Fred, {Project Manager, Tester}>}. The total sum of assigned evaluation values is 6.57. This scenario clearly demonstrates the significance of the proposed problem. It appears that Doug's assignment may not be the best use of his talents. However, it can be supported if we consider the overall team performance.

IV. PROBLEM FORMALIZATIONS WITH THE REVISED E-CARGO MODEL

To be self-contained, we concisely describe the required concepts and definitions of the E-CARGO model. With E-CARGO [20]–[26], a system $\sum$ can be described as a 9-tuple $\sum ::= <\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H}>$, where $\mathcal{C}$ is a set of classes, $\mathcal{O}$ is a set of objects, $\mathcal{A}$ is a set of agents, $\mathcal{M}$ is a set of messages, $\mathcal{R}$ is a set of roles, $\mathcal{E}$ is a set of environments, $\mathcal{G}$ is a set of groups, s_0 is the initial state of the system, and $\mathcal{H}$ is a set of users. In such a system, $\mathcal{A}$ and $\mathcal{H}$, $\mathcal{E}$ and $\mathcal{G}$ are tightly-coupled sets. A human user and his/her agent perform a role together. Every group should work in an environment. An environment regulates a group.

In discussing role assignment problems [20]–[26], environments and groups are simplified into vectors and matrices, respectively. Furthermore, we use nonnegative integers m ($=|\mathcal{A}|$, where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$) to express the size of the agent set $\mathcal{A}$, n ($= |\mathcal{R}|$) the size of the role set $\mathcal{R}$, i, i_1, $i_2, \ldots$ the indices of agents, and j, $j_1, j_2, \ldots$ the indices of roles. Note: we use $\mathcal{N}$ to denote the set of non-negative integers.

Definition 1: A role range vector L [20]–[24], [26] is a vector of the lower bound of the ranges of roles in environment e of group g.

Definition 2: A qualification matrix Q [20], [21], [26] is an $m \times n$ matrix, where $Q[i, j] \in [0, 1]$ expresses the qualification value of agent $i \in \mathcal{N}$ $(0 = i < m)$ for role $j \in \mathcal{N}$ $(0 = j < n)$. $Q[i, j] = 0$ indicates the lowest value and 1 the highest.

Note that, a Q matrix can be obtained by comparing all the Ⓠ s of agents with all the Ⓡ s of roles [20].

Definition 3: A role assignment matrix T [20]–[24], [26] is defined as an $m\times n$ matrix, where $T[i, j] \in \{0, 1\}$ $(0 = i < m, 0 = j < n)$ indicates whether agent i is assigned to role j or not. $T[i, j] = 1$ means yes and 0 no.

As a matter of fact, we consider both the current and the potential roles, i.e., r_c. and $\mathcal{R}_p$ [23]–[26]. From the problem discussed in Section III, we need one agent to play more than one role. With the above new requirement, we need to introduce a new definition.

$$\begin{bmatrix} 0.18 & 0.82 & 0.29 & 0.01 \\ 0.35 & 0.80 & 0.58 & 0.35 \\ 0.84 & 0.85 & 0.86 & 0.36 \\ 0.96 & 0.51 & 0.45 & 0.64 \\ 0.22 & 0.33 & 0.68 & 0.33 \\ 0.96 & 0.50 & 0.10 & 0.73 \end{bmatrix} \qquad \begin{bmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,1\,1\,0 \\ 0\,0\,1\,1 \\ 0\,0\,1\,0 \\ 1\,0\,0\,1 \end{bmatrix}$$

(a) (b)

Fig. 1. Matrix examples: (a) Qualification matrix Q and (b) an assignment matrix T.

Definition 4: An ability limit vector is m-vector L^a, where $L^a[i]$ tells how many roles can be assigned to agent i at most, $(0 = i < m)$.

From Definition 4, we actually introduce a new constraint, i.e., $|a_i \cdot \mathcal{A}_p| + 1 \leq L^a[i]$, $(0 \leq i < m)$.

Definition 5: The group performance σ [20], [21], [26] of group g is defined as the sum of the assigned agents' qualifications, that is

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

Definition 6: Role j is workable [20], [21], [26] in group g if it has been assigned enough agents, i.e., $\sum_{i=0}^{m-1} T[i, j] \geq L[j]$.

Definition 7: T is workable [20], [21], [26] if each role j is workable, i.e., $\forall(0 \leq j < n)\sum_{i=0}^{m-1} T[i, j] \geq L[j])$. Group g is workable if T is workable.

From the above definitions, group g can be expressed by a Q, an L, L^a, and T.

For example, Fig. 1(a) is a qualification matrix for Table II. Fig. 1(b) is an assignment matrix that makes the group (Table II) work with vectors $L = [1, 2, 4, 2]$ (Table I) and $L^a = [1, 2, 3, 2, 2, 2]$ (Table III). The sum of the assigned values is 6.57.

With the brief introduction of the E-CARGO model, it is exciting to formalize the proposed problem with a very concise way as presented in Definition 8.

Definition 8: GMRA is to find a workable T to

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

subject to

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

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

$$\sum_{j=0}^{n-1} T[i, j] \leq L^a[i] (0 \leq i < m) \tag{3}$$

where, expression (1) is a 0–1 constraint, (2) makes the group workable, (3) makes an agent be assigned with a limited number of roles.

For example, Fig. 1(b) is a T following the constraints L = [1, 2, 4, 2] and L^a = [1, 2, 3, 2, 2, 2]. The sum of the assigned evaluation values is 6.57.

Note that, the GMRA problem is fundamentally different from the GRA with conflicting agents on roles or in a group (GRACAR/G) problem [20], [22], [23], because GRACAR/G discusses the role assignment with considerations of possible conflicting agents (i.e., the relationships among agents), i.e., the constraint expressed by a conflict m*m matrix whose elements belong to {0, 1}. The GMRA problem deals with the role assignment with limited numbers of agents on roles (i.e., the relationships among roles), the constraint (3) expressed by an m vector whose elements belong to positive integers. Compared with the GRA problem, GMRA replaces one constraint in GRA, and GRACAR/G adds a new constraint to GRA.

Evidently, GMRA is different from GRA and cannot be solved with the solution to GRA. However, from this formalization, GMRA becomes a typical x-ILP problem if we transfer the Q and T matrices into vectors [12], [13], [17], [20]. Then, it can be solved by an optimization tool in the industry, such as ILOG platform or package.

V. ILOG Solution and Its Performance Experiments

We decide to use the ILOG package to solve the GMRA problem, which is different from the original usage of the optimization programming language (OPL) language of the IBM ILOG CPLEX optimization studio [7]. Using the package by designing a Java program would result in a better performance by bypassing the compiler of OPL. On the other hand, such a program can be used to solve a list of problems automatically. This requirement comes directly from the industry, such as processor scheduling in supercomputing and service allocations in a cloud environment.

To provide a solution with the ILOG package, the major job is to collect the four required elements and transfer the objective and the constraints into the forms required by ILOG.

Step 1: Find out the four elements (i.e., objective function coefficients, constraint coefficients, right-hand side constraint values; and upper and lower bounds) required by the ILOG package, i.e., use Q, L, L^a, and T to define an LP problem in ILOG. In this case, matrix Q expresses the objective function coefficients; and T the variables. The upper and lower bounds of T are 1 and 0.

Step 2: Add the objective and constraint expressions.

The objective of the GMRA problem can be expressed by a formula of the 1-D array forms of matrices Q and T. In the ILOG package, we can maximize this formula based on the objective.

To add the optimization objective, we invoke the following methods in Java:

*IloIntVar[]X = cplex.intVarArray(m*n, 0, 1);*

for T constraint (1),

cplex.addMaximize(cplex.scalProd(X, V));

for σ,

where $X[i\times n+j]=T[i, j]$ and $V[i\times n+j]=Q[i, j]$ $(0 \leq i < m, 0 \leq j < n)$.

To add the constraints to ILOG, we follow three steps.

1) Declare expression objects by calling:
   IloLinearNumExpr expr1 = cplex.linearNumExpr();
   IloLinearNumExpr expr2 = cplex.linearNumExpr().
2) Add all the terms by invoking the methods:
   for $(0 \leq j < n)$ *expr1.addTerm*(1, $X[j+i*n]$);
   for $L[j]$ $(0 \leq j < n)$ constraint (2);
   for $(0 \leq i < m)$ *expr2.addTerm*(1, $X[j+i*n]$);
   for $L^a[i]$ $(0 \leq i < m)$ constraint (3).
3) Add the constraint expressions to ILOG by invoking:
   cplex.addEq(expr1, L);
   for $L[j]$ $(0 \leq j < n)$ constraint (2);
   cplex.addLe(expr2, L^a);
   for $L^a[i]$ $(0 \leq i < m)$ constraint (3).

To check the applicability of the ILOG solution, we conduct experiments with random groups by different scales, where m is from 20 to 600 with an increment of 20, $n = m/2$, $L[j]$ is between 1 and 10, and $L^a[j]$ is between 1 and 5. For each $m(n)$, we produce 100 random groups and collect the maximum, minimum, and average times. The performance experiments are based on the platform shown in Table IV. The performance result is shown in Fig. 2. The results show that the ILOG solution is practical.

TABLE IV
Test Platform Configuration

Hardware	
CPU	Intel core i7-4650U CPU @1.7GHz 2.3 GHz
MM	8GB
Software	
OS	Windows 7 Enterprise
Eclipse	Version: Luna Release (4.4.0)
JDK	Java 8 Update (45)

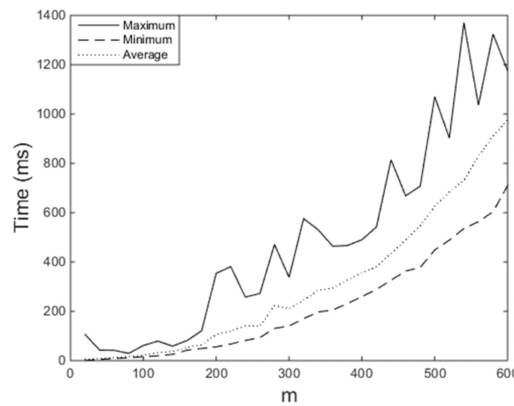

Fig. 2. Performance of the ILOG solution.

VI. Improvement of the ILOG Solution

By looking into the GMRA problems, we find that there are many cases for the problem to have no feasible solutions. When we observe the performance results of the ILOG solutions, we notice that each problem with a similar scale consumes a similar time including those without a feasible solution. We believe that we can improve the ILOG solution by providing a simple way to check if a GMRA problem has a feasible solution. With our improvement, ILOG can solve more problems in a limited time than before.

Theorem 1: The necessary condition for the GMRA problem to have a feasible solution is that $\sum_{i=0}^{m-1} L^a[i] \geq \sum_{j=0}^{n-1} L[j]$.

Proof: To have a feasible solution means that there is a T, such that $T[i,j] \in \{0,1\}(0 \leq i < m, 0 \leq j < n)$, $\sum_{i=0}^{m-1} T[i,j] = L[j](0 \leq j < n)$, and $\sum_{j=0}^{n-1} T[i,j] \leq L^a[i](0 \leq i < m)$.

If $\sum_{i=0}^{m-1} L^a[i] < \sum_{j=0}^{n-1} L[j]$, i.e., $\sum_{i=0}^{m-1} L^a[i] = \sum_{j=0}^{n-1} L[j] - k$, where k is a positive integer. To satisfy $\sum_{j=0}^{n-1} T[i,j] \leq L^a[i]$, $\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} T[i,j] \leq \sum_{i=0}^{m-1} L^a[i] = \sum_{j=0}^{n-1} L[j] - k$, i.e., $\sum_{j=0}^{n-1}\sum_{i=0}^{m-1} T[i,j] \leq \sum_{j=0}^{n-1} L[j] - k$.

That is to say, there are at least k roles, such that

$$\sum_{i=0}^{m-1} T[i,j] = L[j] - 1.$$

This case does not satisfy the constraint

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

Therefore, the assignment matrix T is not a feasible solution. ■

To check if the necessary condition is satisfied, the complexity is $O(m)$ if m and n are in the same scale because it only needs to obtain $\sum_{i=0}^{m-1} L^a[i]$ and $\sum_{j=0}^{n-1} L[j]$.

Theorem 1 can help discard those problems that do not meet the necessary condition to save the search times. However, there are still some problems that have no feasible solutions and make the ILOG system try in vain. We need a necessary and sufficient condition.

To find the necessary and sufficient condition for the GMRA problem to have a feasible solution, we need to introduce some symbols

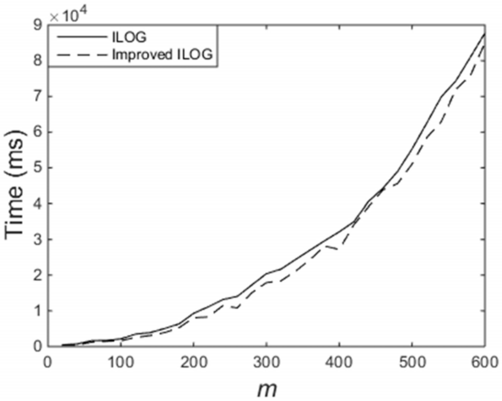

Fig. 3. Times used by processing 100 problems ($n = m/2$, $1 \leq L[j] \leq 10$, and $1 \leq L^a[j] \leq 5$).

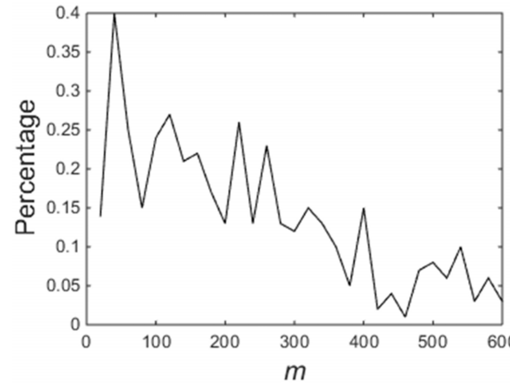

Fig. 4. Percentage of saved time ($n = m/2$, $1 \leq L[j] \leq 10$, and $1 \leq L^a[j] \leq 5$).

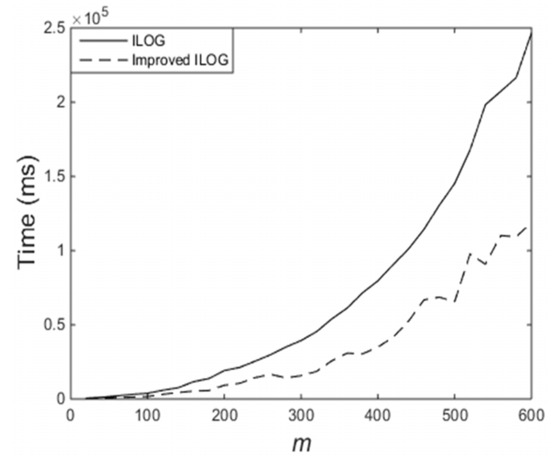

Fig. 5. Times used by processing 100 problems ($n = m$, $1 \leq L[j] \leq 10$, and $1 \leq L^a[j] \leq 10$).

to prove the theorem. Given vector V, $|V|$ is the cardinality of V; $\max^k V$ is the set of the k biggest elements in V ($k \geq 0$), where $|V| > k$; V^* is the first element in V; especially, V^* of an empty set is defined as 0; sub($\max^k V$) is the function such that every k biggest elements in V minus 1; W^* is the initial W, W_i is W_{i-1}-{ $W_{i-1}{}^*$} ($0 < i \leq |W|$); and WV_0 is the initial V, WV_i is sub($\max^{Li-1*} V_{i-1}$) ($0 < i \leq |V|$).

Theorem 2 (Decidable Theorem): GMRA (with Q, T, L, and L^a) has a feasible solution if and only if there exists an integer q, such that L_q is empty and for each i, $L_i{}^* \leq |L_{i-1}L^a{}_i|$ ($0 < i \leq q$).

Proof:

1) *Left to Right (The Necessary Condition):*
Proof by Contradiction: If there is not an integer q, such that L_q is empty and for each of i, $L_i{}^* \leq |L_{i-1}L^a{}_i|$($0 \leq i \leq q$), then it is equivalent to that for any integer q, L_q is not empty or there exists an integer i, such that $L_i{}^* > |L_{i-1}L^a{}_i|$.
Case 1 (L_q Is Not Empty): The construction of a sequence of L_i and $L_{i-1}L^a{}_i$, is equal to assigning roles to agents that can play the most roles every time. This construction guarantees that $|L_{q-1}L^a{}_q|$is the biggest after the current assignment, which can save the most agents for the next assignment. For any q, that L_q is not empty means that there always exists at least one role that has not enough agents to play it.
Case 2 ($L_i^* > |L_{i-1}L_i^a|$): If there exists an integer i, such that $L_i{}^* > |L_{i-1}L^a{}_i|$. It means that, in the ith time, there are not enough agents to play role $L_i{}^*$.
In summary, GMRA has no a feasible solution.

2) *Right to Left (The Sufficient Condition):* If there exists an integer q, such that L_q is empty and for each of i, $L_i{}^* \leq | L_{i-1}L^a{}_i$ |($0 < i \leq q$), then we can construct a solution by assigning the role to agents that can play the most roles, i.e., choosing agent i, where $L^a[i] = \max\{L^a\}$. Therefore, we can obtain a workable T. ■

Now, based on Theorem 2, we can construct a cardinality constraint detection (CCD) algorithm as follows.

Input:

- *An n-*vector *L with the above preliminaries*; and
- *An* m-vector L^a *with the above preliminaries.*

Output:

- *True or False.*

```
CCD(L, L^a)
{
  while (!empty(L)) {
    if (L* > |L^a|) return false;
    else {
      L^a = sub(max^{L*} L^a);
      L = L- L*;
    }
  }
  return true;
}
```

Note that n is the cardinality of L and m is that of L^a, i.e., n and m represent the numbers of elements in L and L^a, respectively. Hence, the complexity of the outside loop (while loop) is $O(n)$. The complexity of $L^a = \text{sub}(\max^{L*} L^a)$ is $O(m\log m)$, regarding that a quick sort method can be used in the sort of L^a. Then, the complexity of the CCD algorithm is $O(n \times m \log m)$. Suppose that m and n are in the same scale, we can simplify the complexity as $O(m^2 \log m)$). The two theorems are implemented in the improved ILOG solution with Java on Eclipse (Section V).

VII. Comparisons

To verify the proposed approach, we conduct comparisons between the improved ILOG solution and the ILOG one [7]. The platform is the same as the one shown in Table IV. In our experiments, we use the case presented in Section V and other three different cases. Here, we collect the total times used by the improved solution and the ILOG one to process the same 100 random problems at each scale (m) (Figs. 3, 5, 7, and 9). To present the improvement, we collect the percentage of the saved time, i.e., $(t_1 - t_2)/t_1$, where t_1 is the time used by the ILOG solution to process 100 problems and t_2 is the time used by the improved solution (Figs. 4, 6, 8, and 10).

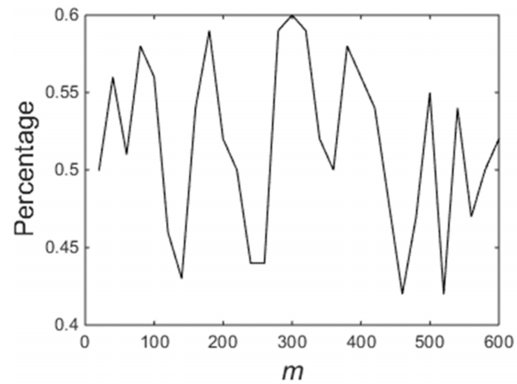

Fig. 6. Percentage of saved time ($n = m$, $1 \leq L[j] \leq 10$, and $1 \leq L^a[j] \leq 10$).

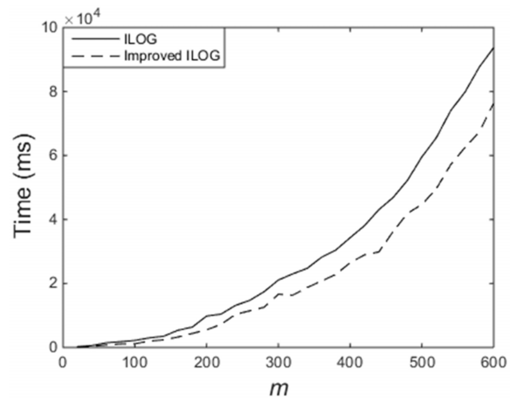

Fig. 7. Times used by processing 100 problems ($n = m/2$, $1 \leq L[j] \leq 20$, and $1 \leq L^a[j] \leq 10$).

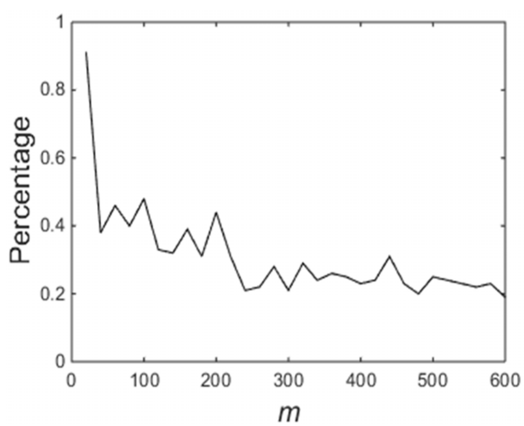

Fig. 8. Percentage of saved time ($n = m/2$, $1 \leq L[j] \leq 20$, and $1 \leq L^a[j] = 10$).

From Figs. 4, 6, 8, and 10, the percentage of the saved time fluctuates because of random group configurations (Qs, Ls, and $L^a s$). That more problems do not have feasible solutions makes the improved solutions use less time than the ILOG one to process the 100 random problems at different scales (m). Figs. 3–10 reflect the fact that the improved solution takes advantages of Theorems 1 and 2.

It is needed to state that GMRA is a new abstract assignment problem under the category of linear programming (LP). Based on the authors' knowledge, there has not been a specific solution to this problem other than the LP solutions similar to the ILOG solution presented in this paper.

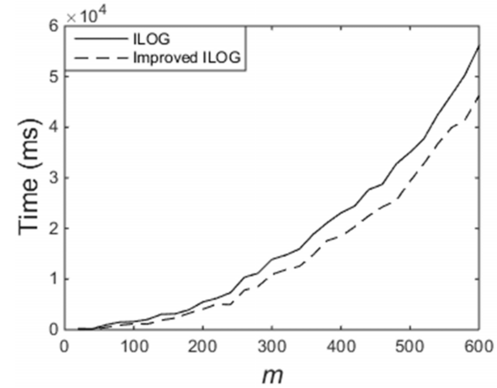

Fig. 9. Times used by processing 100 problems ($n = m/3$, $1 \leq L[j] \leq 30$, and $1 \leq L^a[j] \leq 10$).

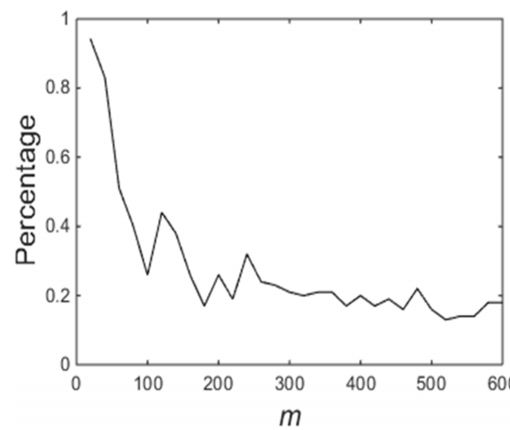

Fig. 10. Percentage of saved time ($n = m/3$, $1 \leq L[j] \leq 30$, and $1 \leq L^a[j] \leq 10$).

TABLE V
OFFERED COURSES

Course	Software Engineering	Computer Networking	Data Mining	Optimization
Number of Sections	2	2	3	2

VIII. ANOTHER REAL-WORLD EXAMPLE

Because GMRT is an abstract problem, it is not hard to find a real-world example. In Summer School SS, the School Manager Anna hopes to call for instructors. In the advertisement, Brett, the HR officer drafts the course list and related numbers of sections for each course (Table V). After two weeks, Brett uses an evaluation process and obtains a shortlist (Table VI). Brett informs Anna that he must assign some people to more than one section because the total number of qualified candidates is less than that of the required sections.

Anna agrees with Brett but asks Brett that if someone must be assigned to more than one section, then the sections must be different and each instructor should be limited to at most three to keep the quality of the course offering. Anna's demand is reasonable because one instructor can shift among different courses but not the same ones because the same course is offered at the same time, and each person should work on limited sections to guarantee the quality of their work. The reason why sections must be at the

TABLE VI
CANDIDATES AND POSITION EVALUATIONS

Course / **Candidates**	Software Engineering	Computer Networking	Data Mining	Optimization
Adolf	0.78	**0.82**	0.29	0.23
Betty	0.68	0.65	**0.80**	0.35
Christen	**0.86**	**0.80**	**0.89**	0.36
Danna	**0.92**	0.55	0.45	0.64
Erlene	0.22	0.33	0.68	**0.73**
Frank	0.84	0.50	**0.96**	**0.73**

TABLE VII
LIMITED NUMBERS OF SECTIONS

Names	Adolf	Betty	Christen	Danna	Erlene	Frank
Limited Number of Sections	2	2	3	2	2	2

same time is that the rental facilities are charged for hours. Based on Anna's request and the preferences of the candidates, Brett composes a table (Table VII) to express the limits. Then, Anna instructs Brett to assign the most qualified candidates to course sections while satisfying the mentioned conditions. This is clearly an instance of the proposed GMRT problem. The result is shown as bold and underlined numbers in Table VI.

IX. CONCLUSION

GMRA is an important, interesting, and complex problem. We formalize the GMRA problem at first. Next, we implement an ILOG solution and verify its performance. Then, we prove the necessary and sufficient condition for GMRA to have a feasible solution, and provide an improved ILOG solution. After that, we compare the improved ILOG solution and the ILOG one. Experiments indicate that the improved ILOG solution can use shorter time to process the same number of GMRA problems than the ILOG one, i.e., the improved ILOG solution is effective.

From this correspondence paper, it is clear that further investigations may be required along the following directions.

1) Although the proposed algorithm improves ILOG solution, it is still possible to find a better solution other than the Linear Programming method used in ILOG.
2) As mentioned in the section of related work, there may be more abstract assignment problems with additional constraints that require further investigations around the problem of GRA.
3) Investigations in new ways to solve the GMRA problem with other constraint logic programming environments, declarative approaches, metaheuristics, and other optimization methods/environments are meaningful.
4) The last but not the least is to conduct empirical studies to apply the proposed method into real-world software project management.

ACKNOWLEDGMENT

The authors would like to thank M. Brewes of Nipissing University for his assistance in proofreading this paper. The authors would also like to thank the anonymous reviewers for their constructive comments that improved the presentation of this paper.

REFERENCES

[1] M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. 21st Annu. Joint Conf. IEEE Comput Commun. Soc. (INFOCOM)*, vol. 3. New York, NY, USA, Jun. 2002, pp. 1587–1596.

[2] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems (Revised Reprint)*. Philadelphia, PA, USA: SIAM, 2012.

[3] M. Dastani, V. Dignum, and F. Dignum, "Role-assignment in open agent societies," in *Proc. 2nd Int. Joint Conf. Auton. Agents Multiagent Syst.*, Melbourne, VIC, Australia, 2003, pp. 489–496.

[4] E. H. Durfee, J. C. Boerkoel, Jr., and J. Sleight, "Using hybrid scheduling for the semi-autonomous formation of expert teams," *Future Gen. Comput. Syst.*, vol. 31, pp. 200–212, Feb. 2014.

[5] J. Ferber, O. Gutknecht, and F. Michel, "From agents to organizations: An organizational view of multi-agent systems," in *Agent-Oriented Software Engineering (AOSE) IV* (LNCS 2935), P. Giorgini, J. Müller, and J. Odell, Eds. Melbourne, VIC, Australia: Springer, 2004, pp. 214–230.

[6] Q. Hui and H. Zhang, "Optimal balanced coordinated network resource allocation using swarm optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 5, pp. 770–787, May 2015.

[7] *IBM, ILOG CPLEX Optimization Studio*. (2013). [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[8] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistic Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[9] D. Li, M. Li, X. Meng, and Y. Tian, "A hyperheuristic approach for intercell scheduling with single processing machines and batch processing machines," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 315–325, Feb. 2015.

[10] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[11] J. J. Odell, H. Van Dyke Parunak, S. Brueckner, and J. Sauter, "Changing roles: Dynamic role assignment," *J. Object Technol.*, vol. 2, no. 5, pp. 77–86, Sep./Oct. 2003.

[12] R. L. Rardin, *Optimization in Operations Research*. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[13] A. Schrijver, *Theory of Linear and Integer Programming*. Chichester, U.K.: Wiley, 2000.

[14] M. Shen, G.-H. Tzeng, and D.-R. Liu, "Multi-criteria task assignment in workflow management systems," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2003, pp. 1–9.

[15] P. Stone and M. Veloso, "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork," *Artif. Intell.*, vol. 110, no. 2, pp. 241–273, 1999.

[16] D. Vail and M. Veloso, "Multi-robot dynamic role assignment and coordination through shared potential fields," in *Multi-Robot Systems*, A. Schultz, L. Parkera, and F. Schneider, Eds. Washington, DC, USA: Kluwer, 2003, pp. 87–98.

[17] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley, 1999.

[18] Y. Zhang, "A role-based approach in dynamic task delegation in agent team work," *J. Softw.*, vol. 3, no. 6, pp. 9–20, 2008.

[19] K. Zheng, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Designing and implementing a human–robot team for social interactions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 4, pp. 843–859, Jul. 2013.

[20] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 46, no. 4, pp. 535–547, Apr. 2016.

[21] H. Zhu and R. Alkins, "Group role assignment," in *Proc. Int. Symp. CTS*, Baltimore, MD, USA, May 2009, pp. 431–439.

[22] H. Zhu and L. Feng, "An efficient approach to group role assignment with conflicting agents," in *Proc. IEEE 17th Int. Conf. Comput. Support. Cooperative Work Design (CSCWD)*, Whistler, BC, Canada, Jun. 2013, pp. 145–152.

[23] H. Zhu, Y. Sheng, and X. Zhou, "The benefits of conflict avoidance in collaboration," in *Proc. IEEE 14th Int. Conf. Cogn. Informat. Cogn. Comput. (ICCI*CC)*, Beijing, China, Jul. 2015, pp. 221–229.

[24] H. Zhu and M. C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[25] H. Zhu and M. Zhou, "M–M role-transfer problems and their solutions," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 2, pp. 448–459, Mar. 2009.

[26] H. Zhu, M. C. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 739–750, May 2012.