Group Role Assignment With Cooperation and Conflict Factors

Haibin Zhu, *Senior Member, IEEE*, Yin Sheng, Xianzhong Zhou, *Member, IEEE*, and Yu Zhu

***Abstract*—Collaboration is complex. To solve a problem occurring in collaboration, using computers, we must first define and specify the problem. This paper presents a challenging problem in collaboration, called group role assignment with cooperation and conflict factors (GRACCFs). This problem's solution aims at creating a high-performance group by role assignment with consideration of cooperation and conflicts between agents. The contribution of this paper is the formalization of the proposed problem, a confirmation of the complexity of the problem, a practical solution that uses the IBM ILOG CPLEX optimization package (ILOG), a verification of the benefits of solving the GRACCF problem by simulations and a practical way of collecting the required factors to support decision makers in solving such a problem within a real-world scenario. Experiments are used to verify the efficiency of the proposed ILOG solution.**

***Index Terms*—Conflict, cooperation, E-CARGO model, role assignment, role-based collaboration (RBC).**

I. Introduction

COLLABORATION is complex, because it involves many aspects that are difficult to specify and, even after tackling those specifications, remains a challenging problem to solve. The fundamental way to deal with complex problems is to "divide and conquer." From an administrative viewpoint, essential work in managing collaboration must include task distribution and task execution. In most cases, execution requires interactions and coordination among the people involved in the collaboration. The total administrative effort applied to collaboration can also be divided into these two parts. It is understandable that more effort in one part reduces the effort needed for the other. In this paper, the authors believe that the latter issue is more complex than the former one, because problems in task distribution are easier to formalize than those in task execution. From this viewpoint, it is acceptable that a problem formalized by mathematical expressions is less complex than a problem, in the same area, which cannot be formalized in this manner. Therefore, it is beneficial that doing the less complex work reduces the effort required in doing the more complex work. Role-based collaboration (RBC) is one such an approach to solving problems in collaboration.

RBC is a promising computational methodology that uses roles as a primary underlying mechanism to facilitate collaborative activities [44]–[50]. In RBC, task execution is called role playing and task distribution is called role assignment. The flowchart of RBC [44] clearly indicates that role playing is highly dependent on role assignment. Furthermore, careful role assignment can help avoid complex role playing problems, such as agent conflicts in role playing. This paper contributes a practical way to conduct role assignment in consideration of possible cooperation and conflicts (CCs) during role playing.

Besides being a computational methodology [44]–[50], RBC has been developed into a discovering methodology that is used to identify and specify unknown problems. Many significant challenges have been revealed in the research of RBC. Interestingly, these challenges have numerous corresponding engineering problems in the real world. Solving such problems meets the requirements of engineering practice. Our previous research has solved several important problems in role assignment and role transfer. General assignment problems have been investigated systematically for many years [2], [3], [5], [6], [22], [27], [32]. However, role assignment problems seem to be ignored by the researchers other than those of RBC. Role assignment is one such challenge and has been revealed as a complex task through the life cycle of RBC [44], i.e., role negotiation, agent evaluation, role assignment, and role transfer, where role negotiation specifies the environment to form a group; agent evaluation creates a matrix the element of which expresses the qualification value of an agent for a role; role assignment chooses the best match between roles and agents from the viewpoint of the whole group; and role transfer changes some agents' roles based on requirements.

Group role assignment (GRA) aims at finding an optimal assignment from roles to agents based on agent evaluation results [50] and it largely affects the efficiency of collaboration and the satisfaction degree of members involved in RBC. There are many factors that need to be considered while performing role assignment. CC factors (CCFs) are two such

Manuscript received March 8, 2016; revised July 14, 2016; accepted November 14, 2016. Date of publication December 13, 2016; date of current version May 15, 2018. This work was supported in part by the Natural Sciences and Engineering Research Council, Canada, under Grant RGPIN262075-2013, in part by the National Natural Science Foundation of China under Grant 71171107, and in part by the Guangzhou City Science and Technology Project under Grant 2013J4500028. This paper was recommended by Associate Editor K. W. Hipel.

H. Zhu is with the Department of Control and System Engineering, Nanjing University, Nanjing 210093, China and the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

Y. Sheng is with Huawei Technologies Company Ltd., Nanjing 210012, China (e-mail: shengyin1218@qq.com).

X. Zhou is with the Department of Control and System Engineering, Nanjing University, Nanjing 210093, China (e-mail: zhouxz@nju.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.2016.2633244

important factors [7], [18]–[21], [26], [29], [35], [40], [41]. Ignorance of such factors in assignment may lead to a significant decrease of the actual team performance during the role playing phase. In this paper, we assume that agent evaluation is done in certain ways [36], [37], i.e., obtaining the qualification matrix [50].

This paper discusses a new GRA problem that considers both CCFs (GRACCF) between agents. Intuitively, when an administrator assigns tasks to people, she/he may need to consider many factors, e.g., individual/collective performance, preferences, future conflicting, and cooperation potentials. Task assignment can be conducted without considering such factors. However, such assignments may lead to difficulties or performance losses when people execute tasks. Our proposed method is to avoid future problems when doing assignments, i.e., to assign tasks to people with consideration of possible influence from people's conflicts or cooperation factors.

To solve such a problem is to maximize the group performance by role assignment in consideration of the mentioned factors. In fact, the proposed problem is a significant extension from our previously solved problem, i.e., GRA with conflicting agents on roles/groups (GRACAR/G) [44] that avoids conflicting agents being assigned to the same role/group. GRACCF considers both conflicts and cooperation, and covers a situation where conflicting agents may be assigned in optimization. For example, the conflict resolution efforts [4] in role playing can be taken as the negative factors in role assignment. Note that, "simulations" are the major research method to verify our proposed solution in this paper. We use "experiments" to mean the simulations that consider performance, i.e., the process time used to solve concrete problems.

This paper is organized as follows. Section II describes a real-world scenario related to the proposed problem. It condenses the environments—classes, agents, roles, groups, and objects (E-CARGO) model in Section III; and formally defines and specifies the GRACCF problem in Section IV. Also in Section IV, the complexity of the GRACCF problem is confirmed, and a practical solution based on the IBM ILOG CPLEX optimization package (ILOG) is proposed by transferring the GRACCF problem into a linear programming (LP) problem. We verify the practicability and the benefits of solving the GRACCF problem through simulations in Sections V and VI. Section VII offers a feasible way to collect the CCFs, which is the prerequisite for solving the GRACCF problem. The related work is discussed in Section VIII. This paper concludes by pointing out topics for future work in Section IX.

II. Real-World Scenario

In company X, Ann, the Chief Executive Officer, has just signed a contract the value of which is a million dollars. She asks Bob, the Human Resources officer, to organize a team from the employees of the company. Bob drafts a position list shown in Table I for the team and a candidate staff shortlist shown in Table II, where the numbers in parentheses are indices of positions and people. Then, Bob initiates an evaluation process and asks the branch officers to evaluate the employees for each possible position (Table II). Even though employee evaluation is not a primary concern of this paper, there are indeed ways to accomplish such a task, e.g., multicriteria decision making [36], [37].

TABLE I
Required Positions

Position	Project Manager	Senior Programmer	Programmer	Tester
Required Number	1	2	4	2

TABLE II
Candidates and Evaluations on Positions

Positions / **Candidates**	Project Manager (0)	Senior Programmer (1)	Programmer (2)	Tester (3)
Adam(0)	0.18	**0.82**	0.29	0.01
Bret (1)	0.35	**0.80**	0.58	0.35
Chris(2)	0.84	0.85	**0.86**	0.36
Doug(3)	0.96	0.51	0.45	**0.64**
Edward(4)	0.22	0.33	**0.68**	0.33
Fred(5)	0.96	0.50	0.10	**0.73**
George(6)	0.25	0.18	0.23	0.39
Harry(7)	0.56	0.35	**0.80**	0.62
Ice(8)	0.49	0.09	0.33	0.58
Joe(9)	0.38	0.54	**0.72**	0.20
Kris(10)	**0.91**	0.31	0.34	0.15
Larry(11)	0.85	0.34	0.43	0.18
Matt(12)	0.44	0.06	0.66	0.37

After that, when Ann and Bob have a meeting with the branch officers to assign the positions to the candidate employees, the officers report that from historical experience, some employees have cooperation intentions and conflicts for different reasons, such as personalities, working styles, emotional issues, and political beliefs. Ann asks the branch officers to evaluate the degrees of all the significant CC effects and the evaluation results are shown in Table III.

In Table III, the 0s means no effect exists; the numbers less than 0 means that there is conflict, but those larger than 0 indicate cooperation. For example, the number means the changing degree of employees' qualification values, i.e., the cross point of "Edward (SP)(4, 1)" and "Bret (T)(1, 3)" is 0.3 meaning that the qualification value of Edward as a Senior Programmer (SP) increases by 30% if Bret takes the position of a Tester (T). That is to say, employees in good relationships will increase their performance if they are doing related jobs and those in conflict may decrease their performance.

Then, Ann tells Bob to try his best to assign the most qualified candidates to positions in consideration of the effects. Bob considers this for a while, and then tells Ann that such a problem may require a significant amount of time to obtain a satisfactory solution. Fortunately, Ann, as a highly experienced administrator, understands the situation and allows reasonable response time.

From the above scenario, Ann and Bob actually follow the initial steps of RBC and Bob encounters a problem of GRA [50] that requires consideration of additional constraints, i.e., the CCFs.

TABLE III
COOPERATION OR CONFLICT FACTORS*

Person (Position) \ Person (Position)	Adam (SP) (0,1)	Adam (P) (0,2)	Bret (SP) (1,1)	Bret (P) (1,2)	Bret (T) (1,3)	Edward (SP) (4,1)	Edward (P) (4,2)	Edward (T) (4,3)	Fred (PM) (5,0)	Fred (SP) (5,1)	Fred (T) (5,3)	Larry (PM) (11,0)	Larry (P) (11,2)	Matt (P) (12, 2)	Matt (T) (12, 3)
Adam (SP)(0,1)	0	0	-0.3	0.35	0.35	0	0	0	-0.4	0	0	0	0	0.8	0.9
Adam (P)(0,2)	0	0	-0.2	-0.2	0.2	0	0	0	-0.5	0	0	0	0	0.5	0.6
Bret (SP)(1,1)	-0.2	0.2	0	0	0	0.2	0.2	0.3	0	0	0	-0.3	0.35	0.7	0.6
Bret (P)(1,2)	-0.35	-0.2	0	0	0	0.2	0.2	0.3	0	0	0	-0.2	-0.2	0	0
Bret (T)(1,3)	0.35	0.4	0	0	0	0.2	0.2	0.2	0	0	0	-0.3	0.35	0	0
Edward (SP)(4,1)	0	0	0.2	0.2	0.3	0	0	0	-0.5	-0.4	-0.3	0	0	0.6	0.7
Edward (P)(4,2)	0	0	0.2	0.2	0.3	0	0	0	-0.4	-0.45	-0.3	0	0	0	0
Edward (T)(4,3)	0	0	0.2	0.2	0.2	0	0	0	-0.2	-0.2	-0.3	0	0	0	0
Fred (PM)(5,0)	0	0	0	0	0	0.3	0.2	0.3	0	0	0	0	0	0.8	0.7
Fred (SP)(5,1)	0	0	0	0	0	-0.4	-0.45	-0.2	0	0	0	0	0	0.6	0.5
Fred (T)(5,3)	0	0	0	0	0	-0.2	-0.2	-0.3	0	0	0	0	0	0	0
Larry (PM)(11,0)	0.3	0.2	-0.3	0.35	0.35	0.3	0.2	0.1	-0.5	-0.4	-0.3	0	0	0.7	0.7
Larry (P)(11,2)	0	0	-0.2	-0.2	0.2	0	0	0	0	0	0	0	0	0.6	0.6
Matt (P)(12,2)	0	0	0	0	0	0	0	0	-0.4	0	0	0.8	0.6	0	0
Matt (T)(12,3)	0	0	0	0	0	0	0	0	-0.3	0	0	0.8	0.6	0	0

*Note: 1) The other person (position) pairs not in the table are all 0s; 2) PM: Project Manager, SP: Senior Programmer, P: Programmer, and T: Tester; 3) (i, j) means an assignment, e.g., (4, 2) means person 4 is assigned with position 2; and 4) The values are produced according to the same evaluation criteria.

III. CONDENSED E-CARGO MODEL

With the E-CARGO model [44]–[50], a system Σ can be described as a nine-tuple $\Sigma ::= <\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H}>$, 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, while an environment regulates a group.

To be self-contained and concise, the definitions of the components of E-CARGO are described and simplified as follows to address the proposed problems: *classes*, *objects,* and *messages* are ignored in this paper, because they are not highly relevant to the proposed problems. Many items in the major components, such as *roles* and *agents*, of the original E-CARGO model are simplified to concentrate on assignments.

In order to state the related concepts, we use conventional notations in definitions and algorithm descriptions. If a and b are objects, "$a ::= b$" denotes that a is defined as b; "$a := b$" denotes that a is assigned with b; $a.b$ denotes b of a or a's b; and $\{a, b, \ldots\}$ denotes a set of enumerated elements of a, b, and others. If a and b are integers, a/b is the integer quotient; $a\%b$ is the remainder of the division of a/b (e.g., $12/5 = 2$ and $12\%5 = 2$). If a and b are real numbers, $[a, b]$ and $[a, b)$ denote the set of all the real numbers between a and b, where the former includes both a and b but the latter includes a but not b. If $\mathcal{Y}$ is a vector, $\mathcal{Y}[i]$ denotes the element at its ith position. If $\mathcal{Y}$ is a matrix, $\mathcal{Y}[i, j]$ denotes the element at the intersection of row i and column j in $\mathcal{Y}$. We use $\mathcal{N}$ to denote the set of natural numbers or, more exactly, non-negative integers, i.e., $\{0, 1, 2, 3, \ldots\}$.

Definition 1 [49], [50]: A *r*ole is defined as $r ::= <id, Ⓡ>$, where

- id *id*entification of the role;
- Ⓡ set of *r*equirements for agents to play r.

Definition 2 [49], [50]: An *a*gent is defined as $a ::= <id, Ⓠ>$, where

- id identification of a;
- Ⓠ set of abilities (or *q*ualifications) possessed by a.

Note: The term *agent* can be any entity that is described in the way of the definition, such as, human being, software agent, machine, or commodity [4], [11], [26], [43].

Definition 3 [49], [50]: An environment is defined as $e ::= <id, \mathcal{R}_e, Ⓢ, \mathcal{B}>$, where

- id identification of the environment;
- $\mathcal{R}_e$ finite set of *r*oles (in e);
- Ⓢ *s*hared object for $\mathcal{R}_e$;
- $\mathcal{B}$ (symbolized from *b*ase) finite set of tuples consisting of roles and their ranges, i.e., $< r, q >$, where $r \in \mathcal{R}_e$.

The role range (also called cardinalities) q (it is symbolized from re*q*uirements, because $\boldsymbol{r}$ and $\boldsymbol{e}$ have been used) is expressed by $<l, u>$ and tells how many agents must (l, i.e., *l*ower bound) and may (u, *u*pper bound) play r in this environment.

Definition 4 [49], [50]: A *group* is defined as $g ::= <id, e, \mathcal{A}_g, \mathcal{J}>$, where

- id identification of the group;
- e environment for the group to work in;
- $\mathcal{A}_g$ finite set of agents (in g);
- $\mathcal{J}$ (symbolized from *j*unction) finite set of tuples consisting of agents and roles, i.e., $\mathcal{J} = \{<a, r> | a \in \mathcal{A}_g, r \in e.\mathcal{R}_e\}$.

Definition 5 [49], [50]: For a group g, a tuple $< a, r >$ of $g.\mathcal{J}$ is called a *role assignment*, also called *agent assignment*.

In formalizing role assignment problems, only agents and roles are emphasized. In the following discussions, current agents or roles are our focus. Environments and groups are simplified into vectors and matrices, respectively. Compared with the definitions in Definitions 1–4 are simplified. Furthermore, we use non-negative integers $m(= |\mathcal{A}|)$ 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.

Definition 6 [45], [50]: A *role range vector* L is a vector of the lower bound of the ranges of roles in environment e of group g. Suppose that roles in $g.e$ are numbered as $j \in N(0 \le j < n)$ and $\mathcal{B}[j]$ means the tuple for role j, then $L[j] = g.e.\mathcal{B}[j].q.\ell$. The role range vector is denoted as $L[j] \in \mathcal{N}$.

Definition 7 [45], [50]: A *qualification matrix* Q is an $m \times n$ matrix, where $Q[i, j] \in [0, 1]$ expresses the qualification value of agent $i \in \mathcal{N}(0 \le i < m)$ for role $j \in N(0 \le j < n)$.$Q[i, j] = 0$ means the lowest and 1 the highest.

Note: A Q matrix can be obtained by comparing all the ⓠs of agents with all the Ⓡs of roles.

Definition 8 [45], [50]: A *role assignment matrix* T is defined as an $m \times n$ matrix, where $T[i, j] \in \{0, 1\}(0 \le i < m, 0 \le j < n)$ expresses if agent i is assigned to role j (i.e., $<a_i, r_j> \in g.\mathcal{J}$) or not (i.e., $<a_i, r_j> \notin g.\mathcal{J}$). $T[i, j] = 1$ means yes and 0 no.

Definition 9 [45], [50]: The group performance σ(named after the sigma Σ that is normally used to denote a sum, this symbol has been used as a system in the E-CARGO model) *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 10 [45], [50]: Role j is *workable* in group g if it is assigned with enough agents, i.e., $\sum_{i=0}^{m-1} T[i, j] = L[j]$.

Definition 11 [45], [50]: T is *workable if* each role j is workable, i.e., $\forall(0 \le j < n) \sum_{i=0}^{m-1} T[i, j] = L[j]$). Group g is *workable* if T is workable.

From the above definitions, group g can be expressed by a Q, an L, and a T. In the following discussions, we assume that $m \ge \sum_{i=0}^{n-1} L[j]$ if we do not clearly state special cases.

Definition 12 [45], [50]: Given Q, and L, GRA is to find a matrix T to

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

$$\text{subject to } T[i, j] \in \{0, 1\} \quad (0 \le i < m, 0 \le j < n) \tag{2}$$

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

$$\sum_{j=0}^{n-1} T[i, j] \le 1 \quad (0 \le i < m) \tag{4}$$

where constraint (2) tells that an agent can only be assigned or not; (3) makes the group workable; and (4) means that each agent can only be assigned to one role. By GRA, we emphasize the viewpoint of group performance other than that of individual agents. Therefore, GRA pursues the best group performance but role assignment may pursue the best individual performances.

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 vector $L = [1\ 2\ 4\ 2]$ in Table I with GRA. The sum of the assigned values (bold in Table II) is 6.96.

$$\begin{bmatrix} 0.18 & 0.82 & 0.29 & 0.01 \\ 0.35 & 0.80 & 0.58 & 0.35 \\ 0.84 & 0.85 & 0.86 & 0.36 \\ 0.96 & 0.51 & 0.45 & 0.64 \\ 0.22 & 0.33 & 0.68 & 0.33 \\ 0.96 & 0.50 & 0.10 & 0.73 \\ 0.25 & 0.18 & 0.23 & 0.39 \\ 0.56 & 0.35 & 0.80 & 0.62 \\ 0.49 & 0.09 & 0.33 & 0.58 \\ 0.38 & 0.54 & 0.72 & 0.20 \\ 0.91 & 0.31 & 0.34 & 0.15 \\ 0.85 & 0.34 & 0.43 & 0.18 \\ 0.44 & 0.06 & 0.66 & 0.37 \end{bmatrix} \qquad \begin{bmatrix} 0\,1\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 1\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix}$$

(a) (b)

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

IV. Problem, Complexity, and Practical Solution

Actually, Definitions 8–11 are concerned with the state of a group after role assignment that is highly dependent on various conditions. Therefore, we need to specify the constraints before the assignment is done. In our opinions, cooperative and conflicting agents can be formally defined as follows. In this specification, a reasonable and acceptable assumption is that cooperative agents increase the performance of all the related agents on all the roles, but conflicting ones decrease the performance.

A. Problem and Its Complexity

Some agents may be in conflict when they play certain roles in a group. By contrast, the performance of other agents may be promoted because of these agents' cooperation.

Definition 13: A *CCF matrix* C^f is an $(m \times n) \times (m \times n)$ matrix: $(A \times R) \times (A \times R) \to [-1, +1]$, where $C^f[i, j, i', j'] \in [-1, +1]$ expresses the changing degree of agent i's qualification when agent i plays role j and agent i' plays role $j'(0 = i, i' = m - 1, 0 = j, j' = n - 1)$, $C^f[i, j, i', j'] > 0$ (<0) means cooperation (conflict), i.e., an increase (decrease) of the group qualification.

From the definition, the group qualification increases/decreases by a percentage of agent i's qualification when agent i plays role j and agent i' plays role j', agents i and i' have effects on each other when they play roles j and j'. The range of $[-1, +1]$ assumes that an agent can obtain up to double its original working performance with cooperation, or may lose the whole working performance due to conflicts.

The above assumption is reasonable and there are many such cases in reality. For example, in a software development team shown in Table III, if Edward takes the position of SP (row), he would be happy and increasing his performance by 20% for Bret to take a position of *programmer* (column), and would be more than happy and increasing his performance by 30% if Bret plays the role of tester. However, if Bret takes the position of *tester*, he would be happy and increasing his performance by 20% if Edward plays a position of SP, *programmer*, or *tester*. This case also tells that matrix C^f is not symmetric.

Definition 14: Given Q, L, and C^f, the GRACCF is to find a T to

$$\max \left\{ \sigma = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,j] \times T[i,j] + \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\sum_{i'=0}^{m-1}\sum_{j'=0}^{n-1} Q[i,j] \times C^f[i,j,i',j'] \times T[i,j] \times T[i',j'] \right\} \tag{5}$$

subject to (2)–(4).

The above GRACCF presents a real-world problem that expresses the important aspect of cooperation/conflict, i.e., with the CCFs introduced, an individual may contribute to a team more than what they can contribute without to form a team. For example, to play role j individually, the performances of agents i_1 and i_2 are 0.7 and 0.8, respectively, i.e., $Q[i_1, j] = 0.7, Q[i_2, j] = 0.8$, and the simple sum is 1.5. If the cooperation factors of i_1 with i_2 on role j and vice versa are 0.3 and 0.4, respectively, i.e., $C^f[i_1, j, i_2, j] = 0.3$ and $C^f[i_2, j, i_1, j] = 0.4$, then the sum of the real performances of the two individual agents is $0.7+0.8+0.7\times 0.3+0.8\times 0.4 = 2.03$. This formal definition reveals the true benefit of cooperation, i.e., "$1 + 1 > 2$" or in the opposite, "$1 + 1 < 2$" in conflict.

The GRACCF problem is a nonlinear 0−1 programming problem [2], [3], [12]–[14], [23], [25], [32], [34], [42] and it is also a special case of the quadratic assignment problem (QAP) [6], [29]. The Sahni and Gonzalez theorem [6] states that the QAP is strongly NP-hard. Therefore, there is no general way to solve the GRACCF problem.

Definition 15 [6]: The QAP is to

$$\min \left\{ \sigma = \sum_{\substack{i,k=0 \\ i\neq k}}^{n-1}\sum_{\substack{j,l=0 \\ j\neq l}}^{n-1} C[i,j,k,l] \times x[i,k] \times x[j,l] + \sum_{i,j=0}^{n-1} B[i,j] \times x[i,j] \right\} \tag{6}$$

$$\text{subject to} \quad x[i,j] \in \{0,1\} \quad (0 \le i,j < n) \tag{7}$$

$$\sum_{i=0}^{n-1} x[i,j] = 1 \quad (0 \le j < n) \tag{8}$$

$$\sum_{j=0}^{n-1} x[i,j] = 1 \quad (0 \le i < n) \tag{9}$$

where C is an $(n\times n)\times(n\times n)$ matrix of cost values, $C[i,j,k,l]$ and $B[i,j]$ $(0 \le i,j,k,l < n)$ are constants, and x is an $n \times n$ variable matrix.

Theorem 1: The GRACCF problem is strongly NP-hard.

Proof: Theorem 1 is proved if we can transfer the GRACCF problem to the QAP form by restrictions and equivalent transformations [12].

Restriction 1: Let $m = n$.

Restriction 2: Let $L[j] = 1 (0 \le j < n)$.

Restriction 3: Let "$\le$" be restricted to $=$ in (4).

Let $C[i,j,k.l] = -C^f[i,j,k,l] \times Q[i,j] (0 \le i,j,k,l < n)$, the time complexity is about $O(n^4)$.

Let $B[i,j] = -Q[i,j] (0 \le i,j < n)$, the time complexity is about $O(n^2)$.

That is, the total time to transform the GRACCF problem to the QAP is within polynomial time, $O(n^4)$.

Because min (x) can be transferred to max(-x), the objective of the QAP, i.e., (6) is equivalent to the objective of the GRACCF problem, i.e., (5).

Now, the GRACCF problem becomes a QAP. Because the GRACCF problem becomes the QAP by restriction, the GRACCF problem is more complex than the QAP. Note the range of the coefficients in C does not affect the complexity of the problem.

Therefore, the GRACCF problem is strongly NP-hard.

Theorem 1 is proved. ■

As a matter of fact, we tried the IBM ILOG CPLEX optimization studio (ILOG) with Definition 14 but failed, because ILOG does not support such a nonlinear expression (5). It is worth noting that in most situations, a nonlinear objective function cannot be transformed to a linear one, and there is not a general efficient way to solve it. However, we may find a special way to transfer the objective of the GRACCF problem into a linear one.

Because the variables in the objective of the GRACCF problem only take 0 or 1 as their values, i.e., $T[i,j] \in \{0, 1\}$, it is possible for us to transfer the objective into a linear one by logical expression transformations.

$T[i,j] \times T[i',j'] = 1$ if both $T[i,j]$ and $T[i',j']$ are 1 and $T[i,j] \times T[i',j'] = 0$, otherwise. We may introduce additional variables $\overline{T}[i,j,i',j'] = T[i,j] \times T[i',j']$ and the constraint can be expressed as a linear expression. The constraints are shown as follows:

$$\overline{T}\left[i,j,i',j'\right] \in \{0,1\}\left(0 \le i,i' < m, 0 \le j,j' < n\right) \tag{10}$$

$$2\overline{T}\left[i,j,i',j'\right] \le T[i,j] + T\left[i',j'\right] \le \overline{T}\left[i,j,i',j'\right] + 1 \quad \left(0 \le i,i' < m, 0 \le j,j' < n\right). \tag{11}$$

Now, we have a new expression of GRACCF.

Definition 16: The new expression of GRACCF is to

$$\max \left\{ \sigma = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,j] \times T[i,j] + \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\sum_{i'=0}^{m-1}\sum_{j'=0}^{n-1} C^f\left[i,j,i',j'\right] \times Q[i,j] \times \overline{T}\left[i,j,i',j'\right] \right\} \tag{12}$$

subject to (2)–(4), (10), and (11).

Therefore, we make the objective function linear by adding some variables that are related to C^f. The number of variables is determined by the number of nonzero values in C^f plus $m \times n$ that is the number of variables in T.

Theoretically, C^f is a matrix that has a large dimension, i.e., $(m \times n) \times (m \times n)$. This dimension will increase the complexity of any solution. We may estimate the complexity of solving the GRACCF problem with an LP solver.

B. Practical Solution

From the practical point of view, C^f is a sparse matrix. That is to say, only a limited number of agents may have significant factors on cooperation or conflicts. We use n_c to express the number of elements in C^f that are not zeros. Now we define a new matrix to replace C^f.

Definition 17: A *Compact CCF matrix* C^{cf} is an $n_c \times 5$ matrix, where $C^{cf}[k, 4] \in [-1, 0) \cup (0, 1] (0 \leq k < n_c)$ expresses that the changing degree of agent $C^{cf}[k, 0]$'s qualification on role $C^{cf}[k, 1]$ while agent $C^{cf}[k, 2]$ plays role $C^{cf}[k, 3]$, where n_c is called *the significance number*, i.e., the total number of nonzero elements in C^f, i.e., $n_c = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{i'=0}^{m-1} \sum_{j'=0}^{n-1} \lceil |C^f[i, j, i', j']| \rceil$.

Here, n_c is actually the total number of significant agent-role opinions about other agent-role assignments from all the team members.

From Definition 17, we may simplify the $(m \times n) \times (m \times n)$ $\overline{T}$ matrix into an n_c vector $\overline{T'}$.

Now, the GRACCF becomes

$$\max \left\{ \sigma(T) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j] + \sum_{k=0}^{n_c-1} C^{cf}[k, 4] \times Q[C^{cf}[k, 0], C^{cf}[k, 1]] \times \overline{T'}[k] \right\} \tag{13}$$

subject to (2)−(4), and

$$\overline{T}'[k] \in \{0, 1\} \quad (0 \leq k < n_c) \tag{14}$$

$$2\overline{T}'[k] \leq T\left[C^{cf}[k, 0], C^{cf}[k, 1]\right] + T\left[C^{cf}[k, 2], C^{cf}[k, 3]\right] \quad (0 \leq k < n_c) \tag{15}$$

$$T\left[C^{cf}[k, 0], C^{cf}[k, 1]\right] + T\left[C^{cf}[k, 2], C^{cf}[k, 3]\right] \leq \overline{T}'[k] + 1 \quad (0 \leq k < n_c). \tag{16}$$

To obtain a practical solution, we use the ILOG package to solve the GRACCF problem, which is different from the original usage of the optimization programming language (OPL) of the IBM ILOG CPLEX optimization studio [17]. Using the package by designing a Java program would result in better performance by bypassing the compiler of OPL.

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, C^{cf}, T, and $\overline{T'}$ to define an LP problem in ILOG. In this case, matrices Q and C^{cf} express the objective function coefficients; and T and $\overline{T'}$ the variables. The upper and lower bounds of T and $\overline{T'}$ are 1 and 0.

Step 2: Add the objective and constraint expressions.

The objective of the GRACCF problem can be expressed by a formula of the 1-D array forms of matrices Q and T and a linear expression of C^{cf} and $\overline{T'}$. 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.

For expressions (2), (13), and (14), we have codes

```
IIoIntVar [] X = cplex.intVarArray(m*n, 0, 1);
IIoIntVar [] Y = cplex.intVarArray(nc, 0, 1);
```

For σ, we have:

```
double [] QNC=new double [nc];
for (int k = 0; k<nc; k++){
QNC[k]=CCF[k][4]*Q[(int)(CCF[k][0]*n+CCF[k][1]);
}
IIoLinearNumExpr obj2 = cplex.linearNumExpr();
for (int k = 0; k<nc; k++){
obj2.addTerm(QNC[k], Y[k]);
}
cplex.addMaximize(cplex.sum(cplex.scalProd(X, V), obj2))
```

where $X[i \times n + j] = T[i, j]$, $V[i \times n + j] = Q[i, j]$ $(0 \leq i < m, 0 \leq j < n)$, and obj2 is the second part of the objective, $\sum_{k=0}^{n_c-1} C^{cf}[k, 4] \times Q[C^{cf}[k, 0], C^{cf}[k, 1]] \times \overline{T'}[k]$.

To add the constraints to ILOG, we iteratively add each constraint expression into ILOG.

1) Constraint (3)

```
for (0 ≤ j < n) {
    IIoLinearNumExpr exp1 = cplex.linearNumExpr();
    for (0 ≤ i < m) exp1.addTerm(1, X[ j+i*n]) ;
    cplex.addEq(exp1, L[i]);
}.
```

2) Constraint (4)

```
for (0 ≤ i < m) {
    IIIoLinearNumExpr exp2 = cplex.linearNumExpr();
    for (0 ≤ j < n) exp2.addTerm(1, X[ j+i*n] ;
    cplex.addLe(exp2, 1);
}.
```

3) Constraint (15)

```
for (0 ≤ i < nc) {
    exp3.addTerm(1, X[(int)CCF[k][0]*n+(int)CCF[k][1]]);
    exp3.addTerm(1, X[(int)CCF[k][2]*n+(int)CCF[k][3]]);
        exp3.addTerm(-1, Y[k]);
        cplex.addLe(exp3, 1);
}.
```

4) Constraint (16)

```
for (0 ≤ i < nc) {
        exp4.addTerm(1, X[(int)CCF[k][0]*n+(int)CCF[k][1]]);
        exp4.addTerm(1, X[(int)CCF[k][2]*n+(int)CCF[k][3]]);
        exp4.addTerm(-2, Y[k]);
        cplex.addGe(exp4, 0);
}.
```

With the above ILOG solution, the final optimized assignment, considering cooperation or conflict factors, is shown in Table II (underlined), i.e., a tuple set as: {<Adam,

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)

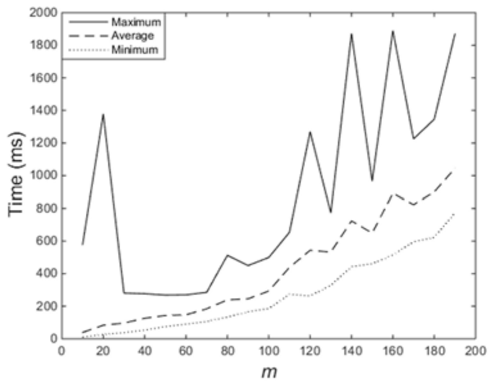


Fig. 2. Performance ($n = m/3, n_c = 5m, 0 \leq L[j] \leq 3$, and $0 \leq j < n$).

Senior Programmer>, <Bret, Tester>, <Chris, Programmer>, <Doug, Tester>, <Edward, Senior Programmer>, <Harry, Programmer>, <Joe, Programmer>, <Larry, Project Manager>, and <Matt, Programmer>}. From Table II, the total sum of the assigned evaluation values (underlined) plus the effect is 9.45. However, by GRA (bold), the total sum (6.96) plus the CCFs (negative) is 6.5. This scenario clearly demonstrates the significance of solving the proposed problem. From this comparison, the CCFs affect the group performance evidently, i.e., $(9.45 - 6.5)/6.5 = 45\%$.

V. PERFORMANCE EXPERIMENTS

To verify the practicability of the proposed approach, we conduct performance experiments on the platform shown in Table IV. The first experiment is conducted to present the time trends. In each step we repeat the test for 100 rounds. In each round, Q, L, and C^{cf} are randomly generated (uniform distributions), where m changes from 10 to 200 with a step of 10. To compare the impact of the ratio of *n/m* on performance, we form two groups of tests whose *n/m* ratios are as 1/3 and 1/4, respectively. To determine the influence of the number of the significant CCFs, we use n_c as 5 m and 7 m, respectively. We only present two figures as Figs. 2 and 3.

From the figures, we notice that the proposed solution based on ILOG is practical. Comparing Fig. 2 with Fig. 3, it is clear that n_c affects the performance significantly, i.e., a larger n_c consumes more time in obtaining the result. Note that the configurations, i.e., the ranges for n, n_c, and $L[j](0 = j < n)$, are set to guarantee a feasible solution for each random group.

From Figs. 2 and 3, we observe that the most important parameters that affect the performance of the proposed solution are m and n_c, i.e., m and n_c determine the complexity scale of the GRACCF problem.

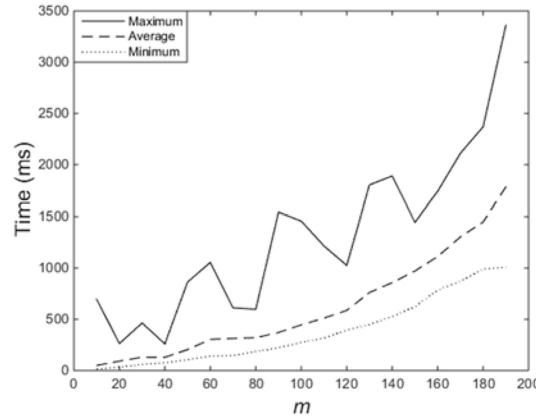


Fig. 3. Performance ($n = m/3, n_c = 7m, 0 \leq L[j] \leq 3$, and $0 \leq j < n$).

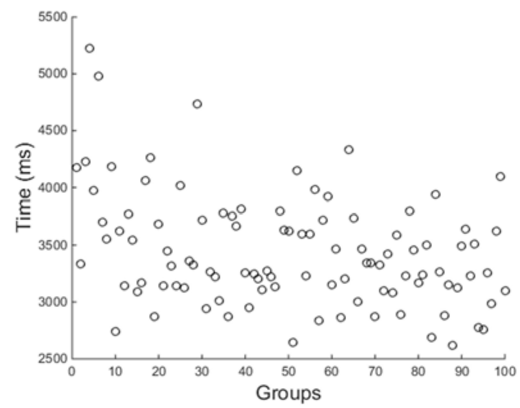


Fig. 4. Performance ($m = 400, n = m/6, n_c = m \times 8, 0 \leq L[j] \leq 66$, and $0 \leq j < n$).

From Figs. 2 and 3, we may also observe fluctuations in the maximum time curve. Here, the maximum time curve indicates values used by one specific case out of 100 random groups of different sizes. The fluctuations in the maximum curve mean that solving a problem of a group with a larger m may use less time than that of a group with a smaller m. These fluctuations are possible because the corresponding groups have different configurations including m, n, L, Q, and C^{cf}.

To present the distribution of the times used to deal with different groups in the same scale, we make an experiment (Fig. 4) with 100 random groups with $Q[i, j] \in [0, 1](0 \leq i < m, 0 \leq j < n), C^{cf}[k, 4] \in [-1, -0.01] \cup [0.01, 1]$ $(0 \leq k < n_c)$.

The second experiment is to see the processing time for a problem with $m = 50$, but the number of rows of the compact CCF matrix, i.e., n_c, is from 500 to 2000 with steps of 250.

From all the above experiments, we know that the proposed approach is practical. From the analysis in Section IV-A, we understand the significant effect of n_c to the ILOG-based solution shown in Fig. 5.

VI. BENEFITS

To quantitatively present the benefits of considering the CCFs, we conduct simulations on the same platform as mentioned in Section V using random groups. In the simulation,

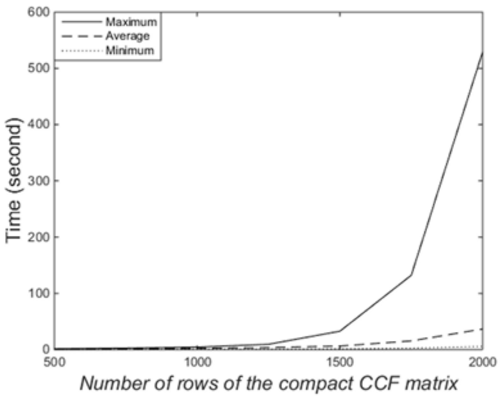


Fig. 5. Performance ($m = 50, n = m/4, 0 \le L[j] \le 12$, and $0 \le j < n$).

TABLE V
SIMULATION AVERAGES ($m = 30, n = 5, 1 \le L[j] \le 6$, AND $0 \le j < 5$)

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	15.23	15.24	15.79	30	16	4%
2	15.79	15.76	16.91	60	17	7%
3	15.65	15.64	17.23	90	17	10%
4	16.09	16.08	18.12	120	17	13%
5	15.81	15.93	18.32	150	17	15%
6	15.90	15.69	18.59	180	17	19%
7	16.01	15.94	19.08	210	17	20%
8	15.52	15.60	18.92	240	17	22%
9	15.80	16	19.55	270	17	23%
10	15.20	15.13	18.82	300	16	25%

we choose two typical sizes, i.e., ($m = 30, n = 5$) and ($m = 100, n = 16$), without loss of generality. We concentrate only on benefit but not efficiency. These are very common in small and middle-sized companies or factories. To simplify the description, we use n_a to denote the number of the required agents, i.e., $n_a = \sum_{j=0}^{n-1} L[j]$.

For each group, we randomly choose the following data within the range of the definitions.

1) $Q[i, j] \in [0, 1](0 \le i < m, 0 \le j < n)$, i.e., the qualification value of each agent for each role.
2) $C^{cf}[k, 4] \in [-1, -0.01] \cup [0.01, 1](0 \le k < n_c)$, i.e., the CCFs.
3) $L[j](0 \le j < n)$, $1 \le L[j] \le m/n$ to make $m \ge n_a$.

For each group, we collect several data items.

1) σ_1 the maximum group performance with GRA.
2) σ_2 the maximum group performance with GRA after considering CCFs, i.e., the real group performance with GRA.
3) σ_3 the maximum group performance with GRACCF.

The benefit λ is calculated as $(\sigma_3 - \sigma_2)/\sigma_2$.

To observe the differences among different configurations, we also change *the significance number* $n_c = m \times x$, where x is from 1 to 10 with a step of 1. To make the data convincing, we create 100 groups for each n_c. Finally, we compute averages of the 100 numbers for each of the data items. Tables V and VI show the simulation results.

By observing the simulation results, it is evident that CCFs do significantly affect group performance. Group performance increases between 2% and 38% in the different types of groups with different configurations (Tables V–XII). Every assignment in these groups introduces CCFs randomly corresponding to the significance number n_c.

TABLE VI
SIMULATION AVERAGES ($m = 100, n = 16, 1 \le L[j] \le 6$, AND $0 \le j < 16$)

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	53.79	53.77	55.16	100	55	2%
2	54.64	54.66	57.00	200	56	4%
3	53.78	53.71	57.17	300	55	6%
4	53.68	53.70	57.87	400	55	7%
5	54.50	54.53	59.55	500	56	9%
6	54.95	54.96	60.67	600	56	10%
7	54.47	54.46	60.86	700	56	11%
8	53.83	53.82	60.74	800	55	12%
9	54.14	54.17	61.51	900	55	13%
10	54.65	54.68	62.61	1000	56	14%

TABLE VII
SIMULATION AVERAGES* ($m = 30, n = 5, 1 \le L[j] \le 6$, AND $0 \le j < 5$)

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	14.94	14.94	15.49	30	16	4%
2	15.64	15.60	16.77	60	17	8%
3	15.81	15.77	17.39	90	17	11%
4	15.15	15.20	17.03	120	16	12%
5	15.07	15.07	17.36	150	16	15%
6	15.51	15.49	18.25	180	17	18%
7	15.45	15.43	18.33	210	17	19%
8	15.38	15.40	18.68	240	17	22%
9	15.62	15.73	19.27	270	17	23%
10	15.72	15.78	19.49	300	17	24%

*Note: $C^{cf}[k,4] \in [-0.9, -0.2] \cup [0.2, 0.9]$ $(0 \le k < n_c)$.

TABLE VIII
SIMULATION AVERAGES* ($m = 100, n = 16, 1 \le L[j] \le 6$, AND $0 \le j < 5$)

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	54.00	53.99	55.25	100	55	2%
2	54.30	54.25	56.70	200	56	4%
3	54.90	54.90	58.19	300	56	6%
4	55.08	55.10	59.24	400	56	7%
5	55.29	55.37	60.40	500	57	9%
6	54.05	54.02	59.80	600	55	10%
7	54.49	54.51	60.91	700	56	11%
8	53.90	53.88	60.61	800	55	12%
9	53.91	53.93	61.36	900	55	13%
10	53.35	53.42	61.29	1000	55	14%

*Note: $C^{cf}[k,4] \in [-0.9, -0.2] \cup [0.2, 0.9]$ $(0 \le k < n_c)$.

To check the effect of the range of the CCFs, we make a new experiment with $C^{cf}[k, 4] \in [-0.9, -0.2] \cup [0.2, 0.9](0 \le k < n_c)$ (Tables VII and VIII) under the same configurations of Tables V and VI, the results are almost the same as those in Tables V and VI. It does follow a common sense that the random creation of C^{cf} makes the elements of CCFs be evenly distributed and the GRA algorithm may evenly assign those in cooperation and in conflicts.

From Tables IX–XII, we know that if only cooperation factors are considered, higher cooperation factors mean better performance.

From Tables V–XII, it is clear that n_c affects the benefits significantly, i.e., a larger n_c means more benefits regardless of the configuration. That is to say, if you allow more people to present their opinions in role assignment, the group

TABLE IX

SIMULATION AVERAGES $\{m = 30, n = 5, 1 \le L[j] \le 6 (0 \le j < 5),$ AND $C^{cf}[k, 4] \in [-1, -0.3] (0 \le k < n_c)\}$

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	15.74	15.50	15.72	30	17	1%
2	15.44	14.88	15.39	60	17	3%
3	15.43	14.72	15.35	90	17	4%
4	15.61	14.61	15.50	120	17	6%
5	15.37	14.20	15.23	150	17	7%
6	15.78	14.35	15.62	180	17	8%
7	15.43	13.83	15.25	210	17	10%
8	15.64	13.64	15.44	240	17	13%
9	15.24	12.99	14.99	270	17	15%
10	15.88	13.51	15.58	300	17	15%

TABLE X

SIMULATION AVERAGES $\{m = 30, n = 5, 1 \le L[j] \le 6 (0 \le j < 5),$ AND $C^{cf}[k, 4] \in [0.1, 0.6] (0 \le k < n_c)\}$

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	15.96	16.12	16.58	30	17	2%
2	15.30	15.65	16.51	60	16	5%
3	15.75	16.22	17.52	90	17	8%
4	15.73	16.35	18.02	120	17	10%
5	15.67	16.40	18.27	150	17	11%
6	15.96	16.81	19.13	180	17	13%
7	15.76	16.86	19.37	210	17	15%
8	14.66	15.76	18.30	240	16	16%
9	16.12	17.52	20.44	270	17	16%
10	15.46	16.96	20.23	300	17	19%

TABLE XI

SIMULATION AVERAGES $\{m = 30, n = 5, 1 \le L[j] \le 6 (0 \le j < 5),$ AND $C^{cf}[k, 4] \in [0.3, 1] (0 \le k < n_c)\}$

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	15.37	15.68	16.85	30	17	7%
2	16.29	16.80	19.22	60	18	14%
3	16.04	16.72	19.63	90	17	17%
4	15.36	16.30	19.70	120	17	20%
5	15.78	17.00	21.05	150	17	24%
6	15.68	17.20	21.78	180	17	27%
7	16.08	17.83	22.69	210	18	28%
8	16.05	18.11	23.43	240	17	30%
9	15.09	16.95	22.56	270	16	34%
10	15.27	17.76	23.47	300	17	33%

TABLE XII

SIMULATION AVERAGES $\{m = 30, n = 5, 1 \le L[j] \le 6 (0 \le j < 5),$ AND $C^{cf}[k, 4] \in [0.5, 1] (0 \le k < n_c)\}$

$n_c = m\times$	σ_1	σ_2	σ_3	n_c	n_a	λ
1	15.62	15.88	17.41	30	17	9%
2	15.79	16.37	18.92	60	17	16%
3	15.50	16.38	19.59	90	17	19%
4	15.74	16.94	20.93	120	17	24%
5	15.75	17.05	21.75	150	17	28%
6	15.67	17.39	22.53	180	17	29%
7	16.08	17.90	23.80	210	17	33%
8	16.00	18.51	24.49	240	17	33%
9	16.13	18.76	25.37	270	18	36%
10	15.25	17.79	24.50	300	17	38%

performance increases more. However, it is not a trivial matter to collect CCFs based on a large n_c. A larger n_c increases the complexity of the GRACCF problem (Figs. 3–5) and increases the burden of the work in collecting the CCFs.

Name: __(#3)__.

If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.
If I work in __(#1)__, I __(#2)__ to cooperate with __(#3)__ in __(#1)__.

#1: a) Position A b) Position B c) Position C d) Position D
#2: a) strongly like b) like c) weakly like d) weakly dislike e) dislike f) strongly dislike
#3: a) Adam b) Brett c) Chris d) Doug e)Edward f) Fred g) George h) Harry i) Ice j) Jeo

Fig. 6. Empty questionnaire.

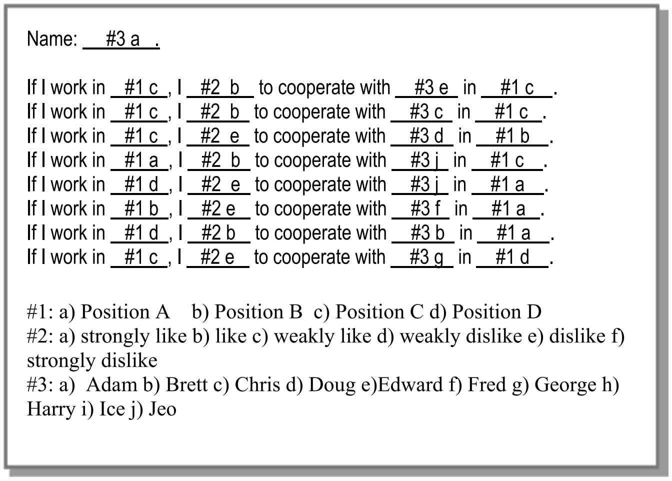
VII. COOPERATION AND CONFLICT FACTOR COLLECTION

GRACCF pursues the optimized group performance in consideration of CC among agents. The solution of GRACCF problems has been verified to possess benefits from the above simulations. However, the solution to a GRACCF problem is based on the establishment of matrix C^{cf} and we must provide a feasible way to collect CCFs in order to apply the above-proposed GRACCF solution.

We propose a questionnaire to collect the opinions from all the related members in a group and form matrix C^{cf}. In an enterprise, we recommend a questionnaire for employees to fill in (Fig. 6). Then, we collect the form and transform them into the compact matrix C^{cf}. In designing the questionnaire, we recommend n_c between 5 and 10 times of m, i.e., employees are asked to fill in the form 5–10 <employee, position> tuples that significantly affect their work in the team. Although the larger the n_c is, the more the benefits from GRACCF is, the time complexity and the effort of processing the survey must be considered.

This proposed method of CCF collection is verified by a case study. The survey is sent to all the graduate students in the Department of Automatic Control and System Engineering, Nanjing University, Nanjing, China. Ten replies are received. Intuitively, this is a reasonable and practical number of replies for a team with up to 50, or even 100 people, because most people only care what positions they take but not the positions others may take. To investigate the required rate of replies is valuable but is beyond the scope of this paper. It is believed that qualifications (performances) of those people who do not reply are not affected by others' role assignments.

In the questionnaire, each employee should select an option from the numbered list and fill in the corresponding blank. In most cases, it is difficult for employees to express their CCFs mathematically. As a result, we use some fuzzy descriptions, e.g., strongly like, like, weakly like, etc. Actually, the questionnaire borrows the ideas from the Likert-type scale. However, in our questionnaire, we hope to create values to reflect the factors of cooperation or conflict. In our proposed method, we


Name: #3 a .

If I work in #1 c , I #2 b to cooperate with #3 e in #1 c .
If I work in #1 c , I #2 b to cooperate with #3 c in #1 c .
If I work in #1 c , I #2 e to cooperate with #3 d in #1 b .
If I work in #1 a , I #2 b to cooperate with #3 j in #1 c .
If I work in #1 d , I #2 e to cooperate with #3 j in #1 a .
If I work in #1 b , I #2 e to cooperate with #3 f in #1 a .
If I work in #1 d , I #2 b to cooperate with #3 b in #1 a .
If I work in #1 c , I #2 e to cooperate with #3 g in #1 d .

#1: a) Position A b) Position B c) Position C d) Position D
#2: a) strongly like b) like c) weakly like d) weakly dislike e) dislike f) strongly dislike
#3: a) Adam b) Brett c) Chris d) Doug e)Edward f) Fred g) George h) Harry i) Ice j) Jeo

Fig. 7. Returned questionnaire.

TABLE XIII
RULES OF TRANSFERRING FUZZY DESCRIPTIONS TO NUMBERS

Intentions	Number
Strongly like	0.9
Like	0.5
Weakly like	0.1
Weakly dislike	-0.1
Dislike	-0.5
Strongly dislike	-0.9

TABLE XIV
PART OF THE COLLECTED C^{cf}

$C^{cf}[k,0]$	$C^{fe}[k,1]$	$C^{cf}[k,2]$	$C^{cf}[k,3]$	$C^{cf}[k,4]$
0	2	4	2	0.5
0	2	2	2	0.5
0	2	3	1	−0.5
0	0	2	2	0.5
0	3	9	0	−0.5
0	1	5	0	−0.5
0	3	1	0	0.5
0	3	6	3	−0.5

assign *"weakly like"* with a value of 0.1 to express cooperation but "*weakly dislike"* with a value of −0.1 to express conflict. The neutral Likert-type scale can be reflected with "no answer" in our questionnaire.

For example, Fig. 6 is a questionnaire to collect the CCFs. In the questionnaire, we use $n_c = 8$ m. Fig. 7 is an example of a returned survey. Note that, we translated the questionnaire from Chinese into English in Figs. 6 and 7. In this paper, we replace the Chinese names in our survey to assumed English names for understanding and anonymity.

In the survey, positions mean roles, and employees mean agents. In order to obtain the numerical CCFs, the fuzzy descriptions must be transferred to numbers. Transferring rules can be various, and we use the following ones as shown in Table XIII without loss of generality.

Table XIV shows a part of the final CCFs collected and transferred from the surveys in our case study. Based on the CCFs, we are able to conduct the GRACCF for our team.

VIII. RELATED WORK

Although GRACCF is evidently an important and challenging problem in organizational performance [1], system construction [21], system management [24], [37], [38] and robot task allocations [8], [9], there is little fundamental and complete research on such issues to the knowledge of the authors. Some related research works concern role assignment for agents in multiagent systems [10], [11], [15], [16], [31]. Some others concern conflict management in multiagent systems [19], [26], [30], [38], [39].

Assignment problems have been investigated extensively in the past decades [2], [3], [5], [6], [22], [27], [33]. However, the proposed assignment problem in this paper has been ignored, because other researchers used different modeling and discovery methodologies from those used in this paper. This situation demonstrates that RBC and E-CARGO [44]–[50] are a new promising methodology and model to discover different assignments problems and can extend the research of assignment problems.

Our previous work on avoiding conflicts with GRA, i.e., GRACAR/G [44] proposes a way to avoid future conflicts by carefully dealing with GRA. It seems that we only change the $m \times m$ conflict matrix A^c into an $(m \times n) \times (m \times n)$ CCF matrix C^f. However, the specification of GRACCF changes significantly compared with that of GRACAR/G. That is, the approach to solving GRACAR/G problems cannot be used to solve GRACCF ones. Also, the creation of the matrix C^f is not trivial. This paper extends our previous work on GRACAR/G in both formalization and practicability.

Choi *et al.* [8] addressed task allocation to coordinate a fleet of autonomous vehicles by presenting two auction algorithms. These algorithms are utilized for decentralized task selection, and use a consensus routine based on local communications to resolve conflicts and achieve agreements on the winning bid values. These two algorithms can produce conflict-free feasible solutions. However, the conflict in [8] means that each task is assigned to no more than one agent, and this meaning is different from that in [40]. Moreover, auction algorithms can only give suboptimal solutions.

Choi *et al.* [9] extended one of their proposed approaches in [8] and applied it to allocating heterogeneous tasks to a team of networked agents with different capabilities. This paper considers two types of agents and three types of tasks. However, this way of categorizing is not comprehensive because there are a variety of agents and tasks in the real world. In fact, the differences between agents/tasks can be reflected by evaluation values.

Kumar *et al.* [24] discussed task assignment considering cooperation between a pair of tasks in workflows. They developed a model to capture the compatibility between actors, and propose an approach to maximize overall compatibility across an end-to-end workflow instance. Note that each task only needs one actor in [24], and the cooperation only exists between tasks. Actually, one task may require more than one agent, and there can be cooperation or conflicts in a task or a group.

Durfee *et al.* [28] proposed a new formulation of the team formation by modeling the assignment and scheduling of expert teams as a hybrid scheduling problem. Their work demonstrates the significance and complexity of the problem of assignment and scheduling in expert teams. It also demonstrates that the assignment and scheduling problem in expert teams can be transferred into an ILP or a mixed integer LP problem and that it is possible for a team formation problem to be solved by a standard mathematical programming package such as ILOG used in this paper.

Nyanchama and Osborn [30] described a role-graph model for role-based access control. In discussing the model, they clarified the roles in conflict. They use the graph model to provide taxonomy for kinds of conflicts. They simplified the complex problem of role assignment in consideration of role conflict by partitioning the role graph into nonconflicting collections that can together be assigned to an agent.

Odell *et al.* [31] pointed out that the roles played by an agent may change over time. They described 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.* [37] proposed 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 administrators in selecting appropriate workers to perform the assigned tasks. Their work can be applied into making the Q matrix for GRA.

Tessier *et al.* [39] presented the problem of conflicting agents in the context of multiagent systems. They defined conflicts among agents based on propositional attributes. They also classified the human conflict handling methods and propose a conflict handling action model.

Zhang [43] 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 primary contribution is a typical way for task assignment with decision trees.

The aforementioned research indicates a strong need to fundamentally investigate GRAs including the problems discussed in this paper.

IX. Conclusion

GRACCF is a complex problem. We first formalize the GRACCF problem. Then, we confirm the complexity of the problem and propose an ILOG-based solution by revising the formalization into an LP problem. Next, we verify the benefit of considering the CCFs in GRA through simulations. The experiments indicate that this proposed solution performs well enough for a relatively large set of instances of the problem ($m = 200$). After that, we propose a practical approach to collect the CCFs to make it easier for practitioners to apply the proposed solution.

From this paper, further investigations may be required along the following lines.

1) We specify the GRACCF problem with Definition 14 that inherently possesses the linear properties of the CCFs. It is valuable to investigate a situation that the CCFs affect the group performance nonlinearly, e.g., the individual agent qualification on a role can be changed by multiplying a CCF. Such a problem may be highly complex both in theory and practice. Therefore, it is valuable to investigate due to the fact that many real-world problems can be classified as nonlinear optimization problems.
2) We may consider additional constraints for GRA, e.g., agents' preferences on roles, and administrators' preferences of role assignment.
3) It is valuable to develop dynamic GRAs that consider the changing of CCFs, i.e., adaptive collaboration [38] with consideration of CCFs.
4) Last but not the least, we need to develop more pertinent methods to transfer the subjective opinions from team members to numerical values in order to apply the proposed approach to conduct role assignment.

Acknowledgment

The authors would like to thank Prof. Q. Zhang from Hong Kong City University and Dr. D. Liu of the Guangdong University of Technology, China, for their discussions in the problem formalizations. They would also like to thank M. Brewes of Nipissing University, Canada, for his assistance in proofreading this paper.

References

[1] B. E. Ashforth, *Role Transitions in Organizational Life: An Identity-Based Perspective*. Mahwah, NJ, USA: Lawrence Erlbaum Assoc., 2001.
[2] E. Balas and J. B. Mazzola, "Nonlinear 0–1 programming: I. linearization techniques," *Math. Program.*, vol. 30, no. 1, pp. 1–21, 1984.
[3] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied Mathematical Programming*. Boston, MA, USA: Addison-Wesley, 1977.
[4] M. Bristow, L. Fang, and K. W. Hipel, "Agent-based modeling of competitive and cooperative behavior under conflict," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 834–850, Jul. 2014.
[5] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems.* Philadelphia, PA, USA: SIAM, 2009.
[6] R. E. Burkard, "Quadratic assignment problems," in *Handbook of Combinatorial Optimizations*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. New York, NY, USA: Springer, 2013, pp. 2741–2814.
[7] B. Canbaz, B. Yannou, and P.-A. Yvars, "Resolving design conflicts and evaluating solidarity in distributed design," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1044–1055, Aug. 2014.
[8] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, Aug. 2009.
[9] H.-L. Choi, A. K. Whitten, and J. P. How, "Decentralized task allocation for heterogeneous teams with cooperation constraints," in *Proc. Amer. Control Conf. (ACC)*, Baltimore, MD, USA, Jun./Jul. 2010, pp. 3057–3062.
[10] 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.
[11] J. Ferber, O. Gutknecht, and F. Michel, "From agents to organizations: An organizational view of multi-agent systems," in *Agent-Oriented Software Engineering IV* (LNCS 2935), P. Giorgini, J. Müller, and J. Odell, Eds. Heidelberg, Germany: Springer, 2004, pp. 214–230.
[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman, 1979.
[13] C. C. Gonzaga, "On the complexity of linear programming," *Rensenhas IME-USP J.*, vol. 2, no. 2, pp. 197–207, 1995.
[14] P. Hansen, B. Jaumard, and V. Mathon, "State-of-the-art survey—Constrained nonlinear 0–1 programming," *ORSA J. Comput.*, vol. 5, no. 2, pp. 97–119, May 1993.

[15] M. Hou, H. Zhu, M. Zhou, and G. R. Arrabito, "Optimizing operator–agent interaction in intelligent adaptive interface design: A conceptual framework," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 161–178, Mar. 2011.

[16] M. Hou, S. Banbury, and C. M. Burns, *Intelligent Adaptive Systems: An Interaction-Centered Design Perspective*. Boca Raton, FL, USA: Taylor & Francis, Dec. 2014.

[17] *ILOG CPLEX Optimization Studio*, IBM, New York, NY, USA, 2016. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[18] J. Jumadinova, P. Dasgupta, and L.-K. Soh, "Strategic capability-learning for improved multiagent collaboration in ad hoc environments," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1003–1014, Aug. 2014.

[19] D. M. Kilgour and K. W. Hipel, "The graph model for conflict resolution: Past, present, and future," *Group Decis. Neg.*, vol. 14, no. 6, pp. 441–460, Nov. 2005.

[20] R. A. Kinsara, D. M. Kilgour, and K. W. Hipel, "Inverse approach to the graph model for conflict resolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 5, pp. 734–742, May 2015.

[21] M. Klein, "Supporting conflict resolution in cooperative design systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 21, no. 6, pp. 1379–1390, Nov./Dec. 1991.

[22] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Q.*, vol. 2, pp. 83–97, Mar. 1955.

[23] H. W. Kuhn, "Nonlinear programming: A historical view," in *Traces and Emergence of Nonlinear Programming*, G. Giorgi and T. H. Kjeldsen, Eds. Basle, Switzerland: Springer, 2014, pp. 393–414.

[24] A. Kumar, R. Dijkman, and M. Song, *Optimal Resource Assignment in Workflows for Maximizing Cooperation* (LNCS 8094), Heidelberg, Germany: Springer, 2013, pp. 235–250.

[25] A. V. Lyamin and S. W. Sloan, "Lower bound limit analysis using non-linear programming," *Int. J. Numer. Methods Eng.*, vol. 55, no. 5, pp. 573–611, Oct. 2002.

[26] T. Malsch and G. Weiss, "Conflicts in social theory and multi-agent systems," in *Conflicting Agents*, C. Tessier, L. Chaudron, and H.-J. Müller, Eds. New York, NY, USA: Springer, 2002, pp. 111–149.

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

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

[29] J. A. Marvel, J. Falco, and I. Marstio, "Characterizing task-based human–robot collaboration safety in manufacturing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 260–275, Feb. 2015.

[30] M. Nyanchama and S. Osborn, "The role graph model and conflict of interest," *ACM Trans. Inf. Syst. Security*, vol. 2, no. 1, pp. 3–33, Feb. 1999.

[31] 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.

[32] C. H. Papadimitriou, "On the complexity of integer programming," *J. ACM*, vol. 28, no. 4, pp. 765–768, Oct. 1981.

[33] L. Pitsoulis and P. M. Pardalos, "Quadratic assignment problem," in *Encyclopedia of Optimization*, L. Pitsoulis and P. M. Pardalos, Eds. New York, NY, USA: Springer-Verlag, 2009, pp. 3119–3149.

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

[35] A. Rapoport and A. M. Chammah, *Prisoner's Dilemma: A Study in Conflict and Cooperation*, vol. 165. Ann Arbor, MI, USA: Univ. Michigan Press, 1965.

[36] J. Geldermann and A. Schöbel, "On the similarities of some multi-criteria decision analysis methods," *J. Multi Criteria Decis. Anal.*, vol. 18, nos. 3–4, pp. 219–230, 2011.

[37] 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.

[38] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 1, pp. 76–92, Jan. 2016.

[39] C. Tessier, H.-J. Müller, H. Fiorino, and L. Chaudron, "Agents' conflicts: New issues," in *Conflicting Agents*, C. Tessier, L. Chaudron, and H.-J. Müller, Eds. New York, NY, USA: Springer, 2002, pp. 1–30.

[40] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998.

[41] S. B. Walker, K. W. Hipel, and H. Xu, "A matrix representation of attitudes in conflicts," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 6, pp. 1328–1342, Jun. 2013.

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

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

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

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

[46] H. Zhu, M. C. Zhou, and M. Hou, "Adaptive collaboration based on the E-CARGO model," *Int. J. Agent Technol. Syst.*, vol. 4, no. 1, pp. 59–76, 2012.

[47] H. Zhu and M. C. Zhou, "Efficient role transfer based on Kuhn–Munkres algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 2, pp. 491–496, Mar. 2012.

[48] 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.

[49] 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.

[50] 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.

Haibin Zhu (M'02–SM'04) received the B.S. degree in computer engineering from the Institute of Engineering and Technology, Zhengzhou, China, in 1983, and the M.S. and Ph.D. degrees in computer science from the National University of Defense Technology (NUDT), Changsha, China, in 1988 and 1997, respectively.

He is a Full Professor with the Department of Computer Science and Mathematics and the Founder and the Director of Collaborative Systems Laboratory, Nipissing University, North Bay, ON, Canada, and a Visiting Professor with the Department of Control Science and Engineering, Nanjing University, Nanjing, China. He was a Visiting Professor and a Special Lecturer with the College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ, USA, from 1999 to 2002, and a Lecturer, an Associate Professor, and a Full Professor with NUDT, from 1988 to 2000. He is the founding researcher of role-based collaboration and adaptive collaboration. He has published 150+ research papers, four books, and four book chapters on object-oriented programming, distributed systems, collaborative systems, and computer architecture.

Dr. Zhu was a recipient of three First Class Ministerial Research Achievement Awards from the Commission of Science Technology and Industry for National Defense of China in 1991, 1994, and 1997, the Second Class Excellent Textbook Award of the Ministry of Electronics Industry of China in 1996, the Second Class National Award of Education Achievement from the Ministry of Education of China in 1997, the Second Class National Award of Excellent Textbook from the Ministry of Education of China in 2002, the Educator's Fellowship of OOPSLA'03, the 2004 and 2005 IBM Eclipse Innovation Grant Awards, the Best Paper Award from the 2004 ISPE International Conference on Concurrent Engineering, and the 2011 Chancellors' Award for Excellence in Research and the 2006–2007/2011–2012 Research Achievement Awards from Nipissing University. He is serving and served as the Co-Chair of the Technical Committee of Distributed Intelligent Systems of the IEEE SMC Society, an Associate Editor for the *IEEE Systems, Man, and Cybernetics Magazine* and the *International Journal of Agent Technologies and Systems*, an Associate Editor-in-Chief for the *International Journal of Advances in Information and Service Sciences*, an Editorial Board Member for the *International Journal of Software Science and Computational Intelligence*, a Guest Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, an organizer of the workshops and special sessions for 10+ international conferences, and a Program Committee Member for over 50 international conferences. He is a member of ACM, and a Life Member of the Chinese Association for Science and Technology, USA.

Yin Sheng received the B.S. degree in automation and the Ph.D. degree in management science and engineering from Nanjing University, Nanjing, China, in 2009 and 2015, respectively.

He is currently a Software Engineer with Huawei Technologies Company Ltd., Shenzhen, China. From 2013 to 2014, he was a Research Assistant with the Collaborative Systems Laboratory, Nipissing University, North Bay, ON, Canada. His current research interests include role-based collaboration, service computing, and intelligent systems.

Dr. Sheng was a recipient of the Best Student Paper Award from the 2013 IEEE 11th International Conference on Networking, Sensing and Control held in Miami, FL, USA.

Xianzhong Zhou (M'10) received the B.S. and M.S. degrees in system engineering and the Ph.D. degree in control theory and application from the Nanjing University of Science and Technology, Nanjing, China, in 1982, 1985, and 1996, respectively.

He is currently a Professor with the Department of Control and Systems Engineering, and the Founder and the Director of the Research Center for Novel Technology of Intelligent Equipment, Nanjing University, Nanjing. His current research interests include intelligent information processing and intelligent systems (novel decision-making system based on humanware technology, rough set theory, and intelligent human–computer interaction), collaboration and mission planning of networked multiagent systems, and theory and technology of advanced integrated automation systems (command and control systems). He has published 120+ research papers, five books, and four book chapters in the above areas.

Dr. Zhou was a recipient of the Excellent Science and Technology Worker of Jiangsu Province in 2000. He was among the people selected for "333 Engineering" of Jiangsu Province in 2002 and 2007. He is currently the Executive Director of the Systems Engineering Society of China, the Director of the Chinese Institute of Command and Control, the Vice-President of the Systems Engineering Society of Jiangsu Province, and a Senior Member of the International Rough Set Society.

Yu (Davis) Zhu is currently pursuing the bachelor's degree in mathematics with the University of Waterloo, Waterloo, ON, Canada, with a focus on business.

He was a co-op student for different companies, such as Blueprint Software Systems, Toronto, ON, Canada, Indigo, Toronto, ON, Canada, Proofpoint, Toronto, ON, Canada, Sunnyvale, Toronto, ON, Canada, FundServ Inc., Toronto, ON, Canada, and CBC Radio, Toronto, ON, Canada. He has published as a co-author research papers on assignment problems.