Balance Preferences with Performance in Group Role Assignment

Dongning Liu, *Member, IEEE*, Yunyi Yuan, Haibin Zhu, *Senior Member, IEEE*, Shaohua Teng, and Changqin Huang, *Member, IEEE*

***Abstract*—Role assignment is a critical element in the role-based collaboration process. There are many factors to consider when decision makers undertake this task. Such factors include a decision maker's preferences and the team's performance. This paper proposes a series of methods, relative to these factors, to solve the group role assignment with balance problem through an association with the one clause at a time approach that is a well-accepted and logic-based association rule mining method. The proposed methods are verified by simulation experiments. The experimental results present the practicability of the proposed solutions. Using the proposed methods, decision makers need only to establish coarse-grain preferences. The fine-grain preferences can be mined. Furthermore, a balance is obtained between the fine-grain preferences and the team's performance.**

***Index Terms*—Assignment, group role assignment (GRA), one clause at a time approach (OCAT), preference, role-based collaboration (RBC), team performance.**

I. Introduction

ROLE assignment has been revealed as a complex process throughout the life cycle of role-based collaboration (RBC) [1], 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 [2] and greatly affects collaboration efficiency and the degree of satisfaction of members involved in RBC.

RBC is a very different topic compared with role-based access control (RBAC) [3]–[5] that is a well-known approach to conduct system protections. The specifications of roles in RBC are also different from those in RBAC. Previous work in RBC has clarified the differences between RBAC and RBC [1], [6]. In short words, RBC utilizes roles as fundamental mechanisms to facilitate collaboration but RBAC access control. Roles in RBC specify both rights and responsibilities but those in RBAC emphasize rights [1], [6].

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 [7], [8]) [9]. That means GRA becomes a straightforward process, which is devoted to finding the maximum team performance.

Note that in addition to individual agent role performance, there are other factors such as preferences that determine whether roles are performed effectively by agents. Psychologically, preferences could be considered as an individual's attitude toward a set of objects. Such an attitude is typically reflected in an explicit decision-making process [5], [10]. Alternatively, one could interpret the term preferences to mean the judgment of a decision maker in the sense of liking or disliking an agent [11], [12]. Preferences may also express those factors that are highly intelligent and difficult to state. For example, in a battlefield, the preferences of the commander is more important than the objective evaluations of soldiers and groups of soldiers. Therefore, in role assignment, we need to consider not only the team performance but also the preferences of decision makers.

However, in the real-world, preferences and performances are often in conflict. For example, in a company, manager A may prefer staff member B, but if A considers objective evaluations, staff member C may be more competitive. How should manager A choose one from B and C? If there are teams involved and many tasks to undertake, the manager's decision making is far more difficult. Moreover, how does a supervisor present his/her preferences? If these preferences are coarse-grained, how does an analyst refine and balance them with team performance?

Although there are many investigations of assignment problems in different applications [13]–[17], we still require an assignment solution that effectively balances preferences with team performance. Therefore, the GRA with balance (GRAB) problem is formed. The primary objective of this paper is to provide a series of practical solutions to the GRAB problem. Although such problems arise in the research of RBC, they are also important and challenging in the domains of administration, production, and engineering.

Manuscript received December 15, 2016; revised April 8, 2017; accepted June 6, 2017. Date of publication August 17, 2017; date of current version May 15, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61402118 and Grant 61370229, in part by the Natural Sciences and Engineering Research Council, Canada, under Grant RGPIN262075-2013, in part by the S&T Project of Guangdong Province under Grant 2016B010108007 and Grant 2016B010109008, in part by the S&T Project of Guangzhou under Grant 201604020145, and in part by GDUPS (2015). This paper was recommended by Associate Editor Q. Shen. *(Corresponding author: Haibin Zhu.)*

D. Liu, Y. Yuan, 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; 970208091@qq.com; shteng@gdut.edu.cn).

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

C. Huang is with the School of Information Technology in Education, South China Normal University, Guangzhou 510631, China (e-mail: cqhuang@zju.edu.cn).

Digital Object Identifier 10.1109/TCYB.2017.2715560

The contributions of this paper include: 1) the concise formalization of the GRAB problem; 2) a fundamental approach to solving the GRAB problem by providing a feasible solution; and 3) a series of improvements to the fundamental approach by combining with the one clause at a time approach (OCAT) [43], which is an iterative, and logic-based association rule mining method.

The proposed methods require a decision maker to establish only coarse-grain preferences. The fine-grain preferences will be mined, and then the balance between the performance and the fine-grain preferences is established by using a GRAB solution. For example, if a manager expresses his/her coarse-grain preferences as like and dislike, our proposed method can conduct GRA with fine-grain preferences, or Likert-scale preferences, i.e., strongly like, like, neutral, dislike, and strongly dislike.

This paper is organized as follows. Related work is discussed in Section II. It describes a real-world scenario related to the proposed problem in Section III and formally specifies the GRAB problem with the revised environments—classes, agents, roles, groups, and objects (E-CARGO) model in Section IV. Section V presents the fundamental solution and Section VI explains how preferences can be mined and refined by OCAT. After that, Section VII proposes two methods to improve the fundamental solution by combining with OCAT to solve the GRAB problem. Section VIII is used to supplement Section VII and deals with the negative association preference set while the former processes the positive set. Section IX illustrates experiments and analyzes their results. This paper concludes and points out future works in Section X.

II. Related Work

Preference research is an important and challenging topic in decision theory, operation research, computer science, economics, etc. [19]–[23]. Currently, such research is more popular than ever. As for human's behavior or performance, preference research can be divided into two aspects.

1) Preference and personal behavior, such as recommendations, retrieval, social network, and customer classification [24], [25].
2) Preference and team performance. In this paper, we mainly focus on the latter.

In the research of preferences and team performance, Salterio [26] showed that managers respond to their own incentives and preferences when subjectively evaluating performance. He indicated that performance evaluation biases affect not only current performance ratings, but also future employee incentives. Mohammadi *et al.* [27] proposed a metric for evaluating the performance of user preferences based on evolutionary multiobjective algorithms by defining a preferred region. Olaverri-Monreal *et al.* [28] investigated whether drivers would like to have additional in-vehicle information that can be also found in other mobile environments. They also studied the driving performance with the preferred locations for in-vehicle information. Zeydan *et al.* [29] considered both qualitative and quantitative variables in evaluating performances for selection of suppliers based on efficiency and effectiveness in one of the biggest car manufacturing factory in Turkey. Melville *et al.* [30] revealed that studies examining the association between information technology and organizational performance are divergent in how they conceptualize key constructs and their interrelationships.

Because preferences and performances have different measurements, many researches always deal with task assignments as a multiobjective problem [18], [31]–[33]. However, in this paper, based on the introduction of roles, RBC and GRA, via normalization, we use methods of dynamic preference weights of agents, such that, we can consider both preferences and performance intuitively. This is in favor of finding the balance points between preferences and performance. Simply, existing methods for solving role assignment problems cannot be used in dealing with the proposed GRAB problem.

As regards to role assignment, some related research work focuses on role assignment for agents in multiagent systems [34], [35] or nodes in networked systems [36]. Bhardwaj and Chandrakasan [36] presented a real-world application that requires role assignment. Dastani *et al.* [34] presented research on the determination of conditions under which an agent can perform a role. Durfee *et al.* [37] proposed a new formulation of the team formation by modeling the assignment and scheduling of expert teams as a hybrid scheduling problem. Shen *et al.* [38] 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. We formally identified a group of problems in role assignment. We propose a formal provision of taxonomy for collective GRA and an indication of the complexities arising from this type of problem through simulations and experiments. Based on these, we successively deal with role transfer, adaptive, conflicts problem [1], [2], [9], [39]–[41], etc.

The aforementioned research indicates a strong need to fundamentally investigate role assignments to balance between preferences and performance.

III. Real-World Scenario

Company X hopes to release a new product. Ann, the Chief Executive Officer, asks Bob, the Human Resources (HR) officer, to organize a team of employees for the project. Bob drafts a team position list (Table I) and a candidate staff shortlist shown as column 1 in Table II. Then, Bob initiates an evaluation process by asking branch officers to evaluate employees for each possible position (Table II). As a routine policy, the new team leader's preferences are considered in the evaluation of team members. Therefore, Bob requests Chris, the Product Manager and Team Leader, to offer her preferences in the recruiting process. To avoid personal conflict with team members, Chris avoids providing a list of names. Instead, she offers a list of preferences for staff properties (Table III). Chris' demand is reasonable because the nonobjective factors, such as gender, experiences and personalities of team members affect the quality of the work. Based on Chris' preferences, Bob creates Table IV based on staff's properties and Table III. Now he obtains the preferred staff list (the rightmost column

TABLE I
REQUIRED POSITIONS

Position	Sales	E-Business	Channel	Ad	Client Service	Logistics
Required Number	3	2	2	3	2	3

TABLE II
CANDIDATES AND POSITION EVALUATIONS

Candidates \ Positions	Sales	E-Business	Channel	Ad.	Client Service	Logistics
Ashley	0.55	0.21	**0.99**	0.63	0.87	0.84
Ava	**0.96**√	0.78	0.45	0.56	0.80	0.23
Azarias	0.66	0.56	0.27	0.46	0.40	0.55
Barry	0.50	0.38	0.74	0.27	0.76	0.61
Beverley	0.71√	0.54	0.24	0.16	0.23	0.34
Bill	0.44	0.65	**0.96**√	0.36	0.32	0.84
Carol	0.35	0.13	0.38	0.23	0.37	0.72√
Dominic	0.56	0.18	0.43	0.69	0.75√	0.27
Eley	0.49	**0.97**	0.83	0.63	0.55	0.76
Fide	0.78	0.80	0.35	**0.98**√	0.78	0.61
Geeley	0.13	0.89	0.24	0.52	0.62	0.36
Hubert	0.28	0.21	0.94√	0.30	0.23	0.84
Ivy	0.11	0.74	0.89	0.38	0.88√	0.87
Jojo	0.23	0.89√	0.62	**0.96**	0.30	0.67
Kyne	0.84	0.79	0.52	0.25	0.59	0.51
Lily	0.55	0.32	0.54	0.60	0.24	**0.92**√
Mark	**0.98**	0.53	0.60	0.78	0.95	0.19
Nancy	0.61	0.18	0.65	0.19	0.84	0.62
Oliver	0.24	**1.0**√	0.43	0.94	0.87	0.75
Peter	0.60	0.28	0.52	0.42	**0.95**	0.92
Qutar	0.90	0.10	0.37	0.92√	0.10	0.35
Richard	0.61	0.10	0.46	0.71	0.19	0.63
Simon	0.67	0.34	0.73	0.28	0.23	0.50
Tom	0.87	0.92	0.24	0.21	0.40	0.41
Tutu	0.68	0.88	0.17	**0.97**	0.53	0.36
Winfred	0.29	0.72	0.62	0.22	0.76	**0.97**√
Xanthus	0.34	0.64	0.40	0.96√	0.67	**0.99**
Yeal	**0.93**√	0.79	0.31	0.95	0.19	0.68
Zero	0.11	0.24	0.85	0.39	**0.95**	0.36
Zoe	0.11	0.87	0.73	0.30	0.83	0.34

of Table IV). He must assign the most qualified candidates to jobs to maximize the team performance while satisfying Chris' preferences.

The challenge to Bob is that there are not enough positively preferred staff members to fulfill the required positions. On the other hand, as a qualified HR officer, Bob knows that he must consider a balance between overall team performance and Chris' preferences.

Bob thinks that he can lower the performance scores of Chris' negatively preferred staff appropriately. If so, there will be an impact on the opportunities for staff to be chosen. Therefore, Bob sets the weight of Chris' negative preferences to 0.5 and the weight for positive preferences to 1. Performance scores in Table II would therefore be adjusted, i.e., if one's score for a position is 0.9, then the score is $0.9 \times 0.5 = 0.45$ if she/he is in the negative preference list. Note if she/he is in the positive preference list, the score remains 0.9. This can lower the chances of negatively preferred staff to be chosen based on Chris' preferences.

If Bob wants to choose staff according to the newly updated performance score list, he still needs to consider those who are not in Chris' preference list, such as Ashley, Jojo, Richard, etc.

TABLE III
CHRIS' PREFERENCES OF STAFF'S PROPERTIES. (a) POSITIVE PREFERENCE. (b) NEGATIVE PREFERENCE

(a)

Number	Gender	Department	Experience
1	F	Marketing	>5 years' experience
2	F	Sales	<1 year's experience
3	M	Marketing	>5 years' experience
4	M	Sales	3-5years' experience

(b)

Number	Gender	Department	Experience
1	F	Sales	3-5 years' experience
2	F	Sales	>5 years' experience
3	M	Marketing	<1 year's experience
4	M	Marketing	1-3 years' experience
5	M	Sales	1-3 years' experience
6	M	Sales	>5 years' experience

TABLE IV
STAFF'S PROPERTIES AND PREFERENCES AFFILIATION

Name	Gender	Department	Experience	Affiliation
Ashley	F	Marketing	<1 year's experience	*M*
Ava	F	Marketing	1-3 years' experience	*M*
Azarias	M	Marketing	>5 years' experience	*P*
Barry	F	Sales	3-5 years' experience	*N*
Beverley	F	Marketing	>5 years' experience	*P*
Bill	M	Marketing	>5 years' experience	*P*
Carol	F	Sales	<1 year's experience	*P*
Dominic	M	Sales	3-5 years' experience	*P*
Eley	F	Sales	3-5 years' experience	*N*
Fide	M	Marketing	3-5 years' experience	*M*
Geeley	F	Sales	>5 years' experience	*N*
Hubert	M	Sales	<1 year's experience	*M*
Ivy	F	Marketing	<1 year's experience	*M*
Jojo	F	Marketing	1-3 years' experience	*M*
Kyne	F	Marketing	3-5 years' experience	*M*
Lily	F	Marketing	<1 year's experience	*M*
Mark	M	Sales	>5 years' experience	*N*
Nancy	F	Marketing	1-3 years' experience	*M*
Oliver	M	Marketing	3-5 years' experience	*M*
Peter	M	Sales	>5 years' experience	*N*
Qutar	M	Sales	<1 year's experience	*M*
Richard	M	Sales	<1 year's experience	*M*
Simon	M	Marketing	3-5 years' experience	*M*
Tom	M	Marketing	<1 year's experience	*N*
Tutu	F	Sales	>5 years' experience	*N*
Winfred	M	Marketing	3-5 years' experience	*M*
Xanthus	M	Marketing	3-5 years' experience	*M*
Yeal	F	Marketing	<1 year's experience	*M*
Zero	M	Marketing	<1 year's experience	*N*
Zoe	M	Marketing	1-3 years' experience	*N*

Note: *P*: positive preference; *N*: negative preference; *M*: no preferences, i.e., middle preference (Please see the nomenclature in the Appendix, i.e., Table A-I).

What weight should be assigned in this situation? If this weight is too big, it will reduce chances of positively preferred staff. On the other hand, if the weight is too small, positively preferred staff with low-performance scores may be chosen. As a result, the overall team performance is lowered.

In consideration of the above situations, 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.

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 preferences and performance [1], [2], [9], [39]–[41]. However, according to GRA the algorithm, the final optimized solution, is shown as the bold numbers in Table II, i.e., a tuple set as: {[Ashley, {Channel}], [Ava, {Sales}], [Bill, {Channel}], [Eley, {E-Business}], [Fide, {Ad}], [Jojo, {Ad}], [Lily, {Logistics}], [Mark, {Sales}], [Oliver, {E-Business}], [Peter, {Client Service}], [Tutu, {Ad}], [Winfred, {Logistics}], [Xanthus, {Logistics}], [Yeal, {Sales}], and [Zero, {Client Service}]. The total sum of the assigned evaluation values is 14.48. Note that it only considers the team performance. Members from Chris' negative preference list such as Eley, Mark, Peter, Tutu, and Zero are assigned jobs. It is obvious that the result cannot satisfy Chris' demands. What should Bob do? This scenario clearly demonstrates the significance of the proposed problem. It indicates that the challenge of how to balance the team leader's preferences with overall team performance is not trivial.

IV. Problem Formalizations With the Extended E-CARGO Model

With the E-CARGO model [1], [2], [9], [39]–[41], 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. An environment regulates a group.

In discussing role assignment problems [1], [2], [9], [39]–[41], environments and groups are simplified into vectors and matrices, respectively. Furthermore, we use non-negative 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 $\mathcal{R}$, $i_0, i_1, i_2, \ldots$ the indices of agents, and $j_0, j_1, j_2, \ldots$ the indices of roles. Note: we use $\mathcal{N}$ to denote the set of non-negative integers.

Definition 1 [Property Set (Ⓢ)]: A property set Ⓢ is a set of properties defined as an object that is identified by $p_0, p_1, \ldots, p_{l-1}$, where $l = |$Ⓢ$|$.

Note that in the above scenario, Ⓢ = {"Gender," "Department," "Experience"}.

Definition 2 (Role): A role [1], [2], [9], [39]–[41] is defined as $r ::= < id$, Ⓡ$>$, where id is the identification of r and Ⓡ is the set of requirements of properties for agents to play r.

Definition 3 (Agent): An agent [1], [2], [9], [39]–[41] is defined as $a ::= < id$, Ⓠ$>$, where id is the identification of a; Ⓠ is the set of a's values corresponding to the properties required in the group.

For example, Agent Ashley = <"Ashley," {F, Marketing, <1 year's experience}>.

Definition 4: A *role range vector* L [1], [2], [9], [39]–[41] is a vector of the lower bound of the ranges of roles in environment e of group g.

Definition 5: A *qualification matrix* Q [9], [39], [40] 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 $j \in \mathcal{N}(0 \le j < n)$. $Q[i, j] = 0$ 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 [9].

Definition 6: A *role assignment matrix* T [1], [2], [9], [39]–[41] is defined as an $m \times n$ matrix, where $T[i, j] \in \{0,1\}$ $(0 \le i < m, 0 \le j < n)$ indicates whether agent i is assigned to role j or not. $T[i, j] = 1$ means yes and 0 no.

Definition 7: The group performance σ [9], [39], [40] 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 8: Role j is *workable* [9], [39], [40] in group g if it has been assigned enough agents, i.e., $\sum_{i=0}^{m-1} T[i, j] \ge L[j]$.

Definition 9: T is *workable* [9], [39], [40] if each role j is workable, i.e., $\sum_{i=0}^{m-1} T[i, j] \ge L[j]$ $(0 \le j < n)$. Group g is *workable* if T is workable.

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

Definition 10: GRA [9], [39], [40] is to find a workable T to

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

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

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

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

where expression (1) is a 0-1 constraint; (2) makes the group workable; and (3) means that each agent can only be assigned to one role.

For example, the case in Table II works with vector L = [3, 2, 2, 3, 2, 3] (Table I). The sum of the assigned values is 14.48. The qualification matrix and the assignment matrix for Table II can be found in Fig. 14 of the Appendix.

Note GRA does not consider preferences. From the scenario discussed in Section III, we need to consider preference weights in revising the qualification matrix then making agent assignments accordingly. This new requirement imposes the introduction of new definitions.

Definition 11: A *property preference list* Ⓘ is defined as a list of instances of the property set and is identified by e_0, e_1, ..., e_{q-1}, where $q = |$Ⓘ$|$.

We use Ⓘ$^+$ and Ⓘ$^-$ to express positive and negative preference lists, respectively. For the scenario in Section III, Chris' positive preference list in Table III is a property preference list, i.e., Ⓘ$^+$ = {[F, Marketing, >5 years' experience], [F, Sales,

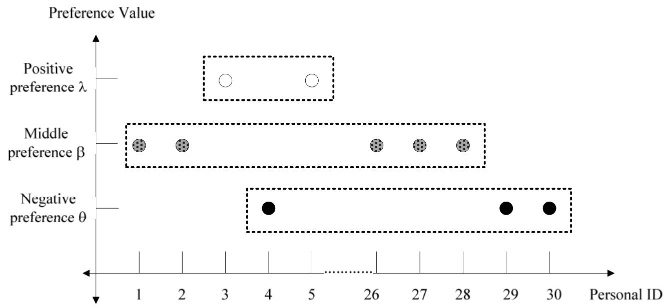


Fig. 1. Classification of preferences of agents.

<1 year's experience], [M, Marketing, >5 years' experience], [M, Sales, 1–3 years' experience]}.

Definition 12: A *preference vector* P_a is an m vector, where $P_a[i] \in (0, 1]$, $i \in \mathcal{N}$ $(0 \leq i < m)$, which indicates the preference weights of agents according to the agents' property preference list Ⓛ.

Note that we use λ to denote a positive preference weight, θ a negative one, and β a weight that is neither positive nor negative, called middle preference weight (such as Ashley, Ava, etc. in Table IV), which is shown in Fig. 1. Therefore, $P_a[i] \in \{\lambda, \beta, \theta\}(\lambda > \beta > \theta)$.

Fig. 1 is corresponding to Table IV. Note that, P_a can be obtained by comparing Ⓛ with each agent's property value set Ⓠ. For the scenario in Section III, $P_a = [\beta, \beta, \lambda, \theta, \lambda, \lambda, \lambda, \lambda, \theta, \beta, \theta, \beta, \beta, \beta, \beta, \beta, \theta, \beta, \beta, \theta, \beta, \beta, \beta, \theta, \theta, \beta, \beta, \beta, \theta, \theta]$, i.e., if agent i's Ⓠ matches the positive Ⓛ, then $P_a[i] = \lambda$; if negative $P_a[i] = \theta$; and otherwise $P_a[i] = \beta$.

Definition 13: An *preferred qualification matrix* Q_p is an $m \times n$ matrix, where $Q_p[i, j] = Q[i, j] \times P_a[i] \in [0, 1]$ expresses the preferred qualification value of agent $i \in \mathcal{N}$ $(0 \leq i < m)$ for role $j \in \mathcal{N}$ $(0 \leq j < n)$. $Q_p[i, j] = 0$ indicates the lowest value and 1 the highest.

Definition 14: The preferred group performance σ_p of group $\mathcal{g}$ is defined as the sum of the assigned agents' preferred qualifications, that is

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

With the brief introduction, if vector P_a is given, we can formalize the proposed problem with a very concise way as presented in Definition 15.

Definition 15: GRAB is to find a workable T to

$$\max \sigma_p = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q_p[i, j] \times T[i, j]$$
$$\text{subject to} \quad (1)-(3).$$

With such a T, the group performance is represented as σ_p. Note that σ_p is not the maximum for Q, because T is determined through consideration of preferences.

The GRAB problem is in fact a special GRA problem that considers the balance between preferences and performance, i.e., the Q matrix in GRA is changed into a new matrix Q_p.

V. Solution 1: The Fundamental Solution

Given a tuple $<\lambda, \beta, \theta>$ $(\lambda > \beta > \theta)$, via RBC and the E-CARGO model, one can find the maximum σ_p via the GRA algorithm [9]. However, as shown in Section III, the difficulty is how to find a method that can compute the appropriate P_a and then balance the preferences with performance. Therefore, we put forward the fundamental solution to find an available set of $<\lambda, \beta, \theta>$ to solve the problem.

To compute preferences, it is common to use the average value of the chosen agents, such as

$$P_{\text{avg}} = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P_a[i] \times T[i, j]}{\sum_{j=0}^{n-1} L[j]} \in (0, 1)$$

where (0, 1) means all the real numbers between 0 and 1 exclusively.

Note that we use $\lambda > \beta > \theta$ as the prerequisite for the values of P_a. If $\lambda = \beta = \theta = 1$, the P_{avg} and σ are maximum, too. This is transformed to the GRA problem, which is a special case of the GRAB problem. Because the GRA solution cannot reflect conflicts between preferences and performance, there are no difference among positive, negative, and middle preferences. Therefore, we define a gradient value to distinguish them at first, and then display the variation of σ accompanied with the variations of λ, β, and θ.

Definition 16: A *preference gradient* P_g is a positive decimal, where if $P_a[i] \in \{\lambda, \beta, \theta\}(\lambda > \beta > \theta,\ 0 \leq i < m)$, then

$$P_g = \frac{\sum_\lambda \text{Agent} + 2^{-1} \sum_\beta \text{Agent} + 2^{-2} \sum_\theta \text{Agent}}{\sum_{j=0}^{n-1} L[j]} \in (0, 1).$$

$\sum_\lambda$ Agent indicates the total number of agents chosen in T with positive preferences, $\sum_\beta$ Agent that with middle ones, and $\sum_\theta$ Agent that with negative ones, respectively.

With respect to the gradient equation, we use the split-half method to deal with preferences. The split-half method is intuitive and common in computational algorithms [15].

Without loss of generality, there might be another variable preference value for a set of agents, i.e., α, where $\theta < \beta < \alpha < \lambda$ (in Sections VI and VIII, we use an example to explain why and when α is introduced), then the *preference gradient* P_g is changed to be

$$P_g = \frac{\sum_\lambda \text{Agent} + 2^{-1} \sum_\alpha \text{Agent} + 2^{-2} \sum_\beta \text{Agent} + 2^{-3} \sum_\theta \text{Agent}}{\sum_{j=0}^{n-1} L[j]} \in (0, 1).$$

It is obvious that the bigger λ is and the smaller β and θ are, the bigger P_g is. In the meantime, σ is becoming smaller, because agents with higher preferences but lower qualifications have better chances of being chosen while agents with lower preferences but higher qualifications have fewer.

In most cases, positive and negative preference values can be manually acquired. Without loss of generality, we might set $\lambda = 1$ and $\theta = 0.5$, i.e., positive preference values are not changed but negative ones are reduced by half. Now, the kernel of the solution is finding the value of β and how to balance preferences with performance via β.

In this situation, for every given β, we can compute every P_g, T, σ_p, and σ. Because $\lambda > \beta > \theta$, we can set the value of β via sampling. For example, let β and θ be initialized as 0.5

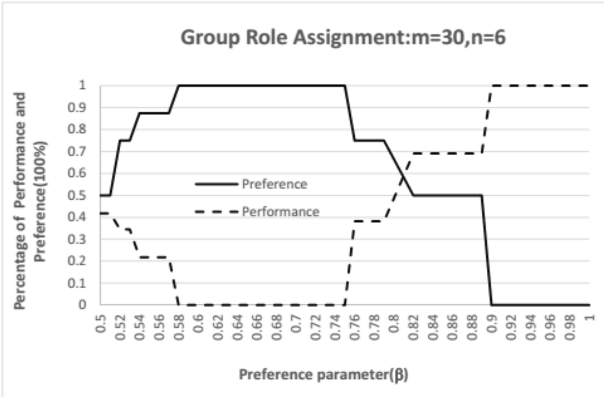


Fig. 2. Result of solution 1.

and increase β with a step of 0.01, such that we can get a list of βs and other values as functions of β, such as $P_g(\beta)$, $T(\beta)$, $\sigma_p(\beta)$, and $\sigma(\beta)$.

On the other hand, P_g and σ have different measurements. In order to balance them, we must normalize them to the same scale, such as the percentage of the maximum P_g and σ.

Definition 17: The normalized preference gradient P_g^β of group g is a positive decimal, where β is a preference variable for a set of agents, and

$$P_g^\beta = \frac{P_g(\beta) - \min\{P_g\}}{\max\{P_g\} - \min\{P_g\}} \in (0, 1)$$

where $\min\{P_g\}$ indicates the minimum value of $P_g(\beta)$ for all possible βs while $\max\{P_g\}$ the maximum.

Definition 18: The normalized group performance σ^β of group g is a positive decimal, where β is a preference variable for a set of agents, and

$$\sigma^\beta = \frac{\sigma(\beta) - \min\{\sigma\}}{\max\{\sigma\} - \min\{\sigma\}} \in (0, 1)$$

where $\min\{\sigma\}$ indicates the minimum value of $\sigma(\beta)$ for all possible βs while $\max\{\sigma\}$ the maximum.

Note that if there is another variable preference value for a set of agents, i.e., α, then P_g^β and σ^β become $P_g^{\alpha,\beta}$ and $\sigma^{\alpha,\beta}$.

Now, we can compare preferences with performance as in Algorithm 1.

Fig. 2 illustrates the basic solution (solution 1) with respect to the example mentioned in Section III. We can now compare preferences and performance via normalization. As a result, the intersection in Fig. 2 is the balance point, where $\beta = 0.81$ and $P_g^\beta = \sigma^\beta = 0.594$.

The left portion indicates that preferences are in favor relative to the performance, while the right portion indicates that the performance is favored relative to preferences. In this case, $\beta = 0.81$ is the balance point between preferences and performance, and $P_g^\beta = \sigma^\beta = 0.594$.

According to this balance β, the assignments are shown as underlined numbers in Table II. Now, the team performance σ_p is 13.54, which is not the maximum for Q but the most balanced one. The preference gradient P_g is 0.618.

Algorithm 1 Solution 1

Input:
- ➢ Agent list $\mathcal{A}$;
- ➢ Positive preference list $Ⓓ^+$;
- ➢ Negative preference list $Ⓓ^-$;
- ➢ A role range vector L; and
- ➢ A qualification matrix Q.

Output:
- ➢ A normalized preference gradient P_g^β; and
- ➢ A normalized group performance σ^β.

CPP$\sigma^\beta(\mathcal{A}, Ⓓ^+, Ⓓ^-, L, Q)$ //Calculation of P_g^β and σ^β
{ Initialization: $\lambda = 1$ and $\beta = \theta = 0.5$;
  Get P_a based on $\mathcal{A}$, $Ⓓ^+$, $Ⓓ^-$, λ, θ, β;
  while $(!(\lambda = \beta))$ {
    $\beta+ = 0.01$;
    Updated βs' value in P_a;
    for $(i = 0;\ i < m;\ i++)$
      for $(j = 0;\ j < n;\ j++)$ $Q_p[i, j] = Q[i, j] \times P_a[i]$;
    Calculate $T(\beta)$ and $\sigma_p(\beta)$ with Q_p via the Kuhn-Munkres Algorithm;

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

    Save $\max\{\sigma\}$ and $\min\{\sigma\}$;

$$P_g(\beta) = \frac{\sum_\lambda Agent + 2^{-1}\sum_\beta Agent + 2^{-2}\sum_\theta Agent}{m};$$

    Save $\max\{P_g\}$ and $\min\{P_g\}$;
  }
  $\beta = 0.5$;
  while $(!(\lambda = \beta))$ {
    $\beta+ = 0.01$;

$$P_g^\beta = \frac{P_g(\beta) - \min\{P_g\}}{\max\{P_g\} - \min\{P_g\}};\ \sigma^\beta = \frac{\sigma(\beta) - \min\{\sigma\}}{\max\{\sigma\} - \min\{\sigma\}};$$

  }
}

VI. Positive Association Preference Mining Via OCAT

Based on the method mentioned in Section V, we can basically balance preferences with performance. However, this "basic" method is "coarse-grain," because the middle preference is coarse-grain. We can mine and refine it.

In the view of data mining, positive and negative preferences can be treated as positive and negative training sets, respectively. After mining, there are some association preferences that can be discovered in the middle preference set. For example, because β is 0.81 and bigger than 0.75 in solution 1, there are some positive association preferences (assume that the value is α) in the middle preferences, which are similar to what is shown as Fig. 3 that is a modification of Fig. 1. To balance preferences with performance in the fine-grain scale, we ought to discover the middle preferences and then compute a revised weight value.

According to the characteristics of preferences, we use a data mining method called OCAT, which is a logic-based association rule mining method [43]. The input data is the two collections of disjoint positive and negative binary examples. It determines a set of conjunction normal formula (CNF)

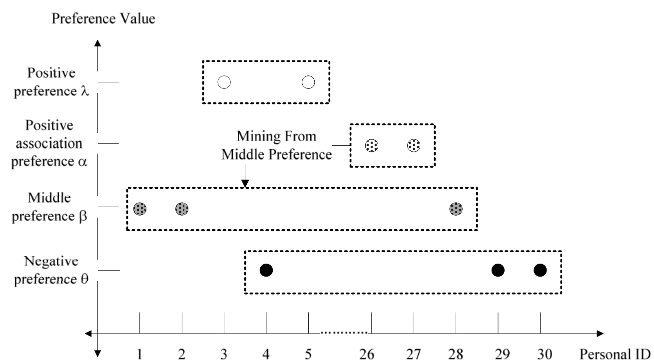


Fig. 3. Classification of preferences of agents after mining.

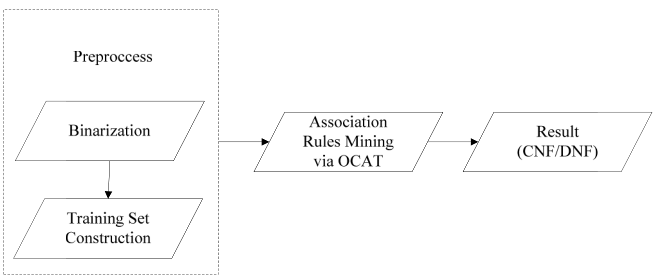


Fig. 4. Process of OCAT in preference mining.

TABLE V
CHRIS'S BINARY CODE OF PREFERENCES OF STAFF

Gender	Department	Experience
A_1	A_2	A_3A_4

clauses, where each clause accepts all the positive examples but the conjunction of all the clauses rejects the negative ones.

Preferences shown in Section III are easily expressed as binary numbers, i.e., binarized. Then, they can be changed to positive or negative training sets, such that we can construct CNF clauses via OCAT. These are reasons why we choose OCAT. The process is shown in Fig. 4.

According to our case and OCAT, we set binary fields corresponding to the attributes as shown in Table V first. Note that, gender is an important factor for a specific product in marketing and sales.

The corresponding relations are as follows:

$$A_1 = \begin{cases} 0, \text{female} \\ 1, \ \text{male} \end{cases}$$

$$A_2 = \begin{cases} 0, \ \text{sales} \\ 1, \ \text{marketing} \end{cases}$$

$$A_3A_4 = \begin{cases} 00, \ > 5 \text{ years' experience} \\ 01, \ 3-5 \text{ years' experience} \\ 10, \ 1-3\text{years' experience} \\ 11, \ < 1 \text{ year's experience.} \end{cases}$$

Based on these, we code positive and negative preferences as binary numbers.

Following the classical OCAT approach [43] and the example illustrated in Section III, we can get an association rule as the following:

$$(A_2 \lor A_4) \land (\neg A_2 \lor \neg A_3) \land (A_1 \lor A_3 \lor \neg A_4).$$

TABLE VI
ASSOCIATION POSITIVE PREFERENCE SET AND MIDDLE PREFERENCE SET. (a) POSITIVE ASSOCIATION PREFERENCE IN BINARY. (b) MIDDLE PREFERENCE IN BINARY

(a)

Binary	Gender	Department	Experience
1101	M	Marketing	3-5 years' experience
1011	M	Sales	<1 year's experience

(b)

Binary	Gender	Department	Experience
0010	F	Sales	1-3 years' experience
0101	F	Marketing	3-5 years' experience
0110	F	Marketing	1-3 years' experience
0111	F	Marketing	<1 year's experience

It indicates that we can split the middle preference set into two sets as shown in Table VI.

Now, in addition to the positive, negative, and middle preference sets, we have one more preference set, which can be named "positive association preference" set, and we set its weight as α.

Note that, the OCAT algorithm becomes slower when dealing with an increasing number of attributes. Therefore, to accelerate it, one can combine it with the branch and bound algorithm [43], which is a heuristic method. In fact, this is the third solution in mining association preferences in this paper.

On the other hand, preferences can be divided into many segments by using OCAT repeatedly. We can use the bisection method to change the bounds of positive or negative preference training sets. Every time, we get one more preference set that makes preferences finer. This gives us an option to improve the solution of GRAB.

VII. IMPROVEMENT OF THE GRAB SOLUTION

To illustrate the validity of the improved method, we still use the scenario from Section III. Now, we have four preference sets, the positive preference set whose weight is λ, the negative one whose weight is θ, the middle one whose weight is β, and the positive association one whose weight is α, which are shown in Fig. 3. Assume that λ and θ are still 1 and 0.5, we have two methods to deal with α and β.

A. *Solution 2: Variable α and Constant β*

Now, the values of α and β both affect the normalized preference gradient and the normalized group performance. Intuitively, we can set either α or β to be a constant and change the other value to find the balance. As the value of the middle preference set, we may set $\beta = (\lambda + \theta)/2 = 0.75$ or other approximate middle values. It is obviously that $\theta < \beta < \alpha < \lambda$, therefore α varies in [0.75, 1]. We can calculate the new normalized preference gradient and the normalized group performance. Considering variable α and constant β, the calculation procedure of P_g^{β} and σ^{β} is changed to the procedure of calculating P_g^{α} and σ^{α} as in Algorithm 2.

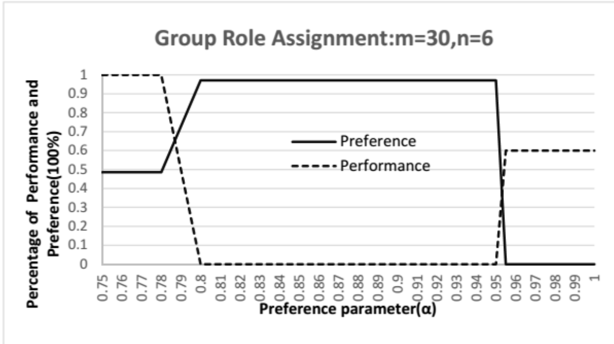


Fig. 5. Result of solution 2.

Algorithm 2 Solution 2

Input:
➢ Agent list $\mathcal{A}$;
➢ Positive preference list $①^+$;
➢ Negative preference list $①^-$;
➢ A role range vector L; and
➢ A qualification matrix Q.
Output:
➢ The normalized preference gradient P_g^α; and
➢ The normalized group performance σ^α.
CPP $\sigma^\alpha(\mathcal{A}, ①^+, ①^-, L, Q)$ //Calculating of P_g^α *and* σ^α
{ Initialization: $\lambda = 1$, $\alpha = \beta = 0.75$ and $\theta = 0.5$;
Get elements' affiliations for P_a via OCATP;
Get P_a based on $\mathcal{A}$, $①^+$, $①^-$, λ, θ, α, β;
while $(!(\lambda = \alpha))$ {
$\alpha+ = 0.01$; Updated αs' value in P_a;
for $(i = 0;\ i < m;\ i++)$
for $(j = 0;\ j < n;\ j++)$ $Q_p[i, j] = Q[i, j] \times P_a[i]$;
Calculate $T(\alpha)$ and $\sigma_p(\alpha)$ *with* Q_p via the Kuhn-Munkres Algorithm;

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

Save $\max\{\sigma\}$ and $\min\{\sigma\}$;

$$P_g(\alpha) = \frac{\sum_\lambda Agent + 2^{-1}\sum_\alpha Agent + 2^{-2}\sum_\beta Agent + 2^{-3}\sum_\theta Agent}{m};$$

Save $\max\{P_g\}$ and $\min\{P_g\}$;
}
$\alpha = 0.75$;
while $(!(\lambda = \alpha))$ {
$\alpha+ = 0.01$;

$$P_g^\alpha = \frac{P_g(\alpha) - \min\{P_g\}}{\max\{P_g\} - \min\{P_g\}};\ \sigma^\alpha = \frac{\sigma(\alpha) - \min\{\sigma\}}{\max\{\sigma\} - \min\{\sigma\}};$$

}
}

Fig. 5 shows the result of solution 2. Note that, when α increases, there are three cases.

Case 1: Agents referred by β are chosen originally. With the increasing of α, they may not be picked up now. Then P_g^α may be increasing and σ^α decreasing.

Case 2: Agents referred by λ are favored originally. With the increasing of α, they are not picked up now. Then P_g^α may be decreasing and σ^α increasing.

Case 3: Although α is increasing, no agent changes the preference status, then P_g^α and σ^α keep the same.

Because θ is only 0.5 so small that the corresponding agents will not be chosen, it will not affect the variations of the normalized preference gradients and the normalized group performance.

On the other hand, in solution 2, there are often many intersections among the curves of P_g^α and σ^α. As a result, we choose the biggest value of $P_g^\alpha = \sigma^\alpha$ among intersections as the balance point. For samplings of α and in a 2-D figure, this point is unique.

Therefore, in Fig. 5, the balance point of P_g^α and σ^α is 0.651, and α is 0.79. According to this balance α, the binary code matrix, the assignments are shown as the highlighted numbers in Table II. Now, the team performance σ_p is 13.33, which is not the maximum for Q but the most balanced one with the assumption of solution 2. The preference gradient P_g is 0.588.

B. Solution 3: Both α and β Are Variables

In solution 2, β is a constant. In fact, we can set β as a variable, too. In this case, the unary functions P_g^α or P_g^β is transformed to a binary one $P_g^{\alpha,\beta}$, so is $\sigma^{\alpha,\beta}$. If they have two unknowns, the intuitions are lost, and it is too complex for us to analyze.

It is worth noting that, if we assume that x is the total number of chosen agents (that will be assigned eventually) referred by α, y is the total number of selected agents (that will be assigned eventually) referred by β, and ξ is the result of solution 1, then we ought to get an identical equation, that is

$$\xi = \frac{x \times \alpha + y \times \beta}{x + y}.$$

According to this, one can get

$$\beta = \frac{\xi \times (x + y) - x \times \alpha}{y}.$$

Then, $P_g^{\alpha,\beta}$ and $\sigma^{\alpha,\beta}$ are transformed to two unary functions, i.e., $P_g^{\alpha,\beta}(\alpha)$ and $\sigma^{\alpha,\beta}(\alpha)$, which are intuitive and easy to analyze.

However, the difficulty is that x and y are unknown now. If we do not know what are α and β, then we could not get x and y, even though we could get ξ from solution 1.

To solve this problem, we use the iteration method [15].

1) We set $X(0)$ as the total number of the chosen agents (that are assigned eventually) referred by α under ξ in solution 1.
2) We set $Y(0)$ as the total number of chosen agents (that are assigned eventually) referred by β.
3) We set $\alpha(0) = \beta(0) = \xi$.

Then

$$\beta(0) = \frac{\xi \times (X(0) + Y(0)) - X(0) \times \alpha(0)}{Y(0)}.$$

To obtain the new α (and β), such as $\alpha(1)$ [and $\beta(1)$], we can use ξ, $X(0)$ and $Y(0)$ as the prerequisite knowledge and assume

$$\beta(1) = \frac{\xi \times (X(0) + Y(0)) - X(0) \times \alpha(1)}{Y(0)}$$

where $\alpha(1) = \alpha(0) + 0.01$.

Algorithm 3 Solution 3

Input:
- ➢ Agent list $\mathcal{A}$;
- ➢ Positive preference list $\textcircled{1}^{+}$;
- ➢ Negative preference list $\textcircled{1}^{-}$;
- ➢ A role range vector L;
- ➢ A qualification matrix Q; and
- ➢ ξ, X, and Y.

// get from Solution 1, X as $X(0)$, Y as $Y(0)$

Output:
- ➢ The normalized preference gradient $P_g^{\alpha,\beta}$; and
- ➢ The normalized group performance $\sigma^{\alpha,\beta}$.

CPP$\sigma^{\alpha,\beta}(\mathcal{A}, \textcircled{1}^{+}, \textcircled{1}^{-}, L, Q, \xi, X \text{and} Y)$

//Calculating of $P_g^{\alpha,\beta}$ *and* $\sigma^{\alpha,\beta}$

{ Initialization: $\lambda = 1$, $\gamma = 0.1$ and $\alpha = \beta = \xi$; // $\xi < \alpha$ (*and* β) <1.

Get elements' affiliations for P_a via OCATP;

Get P_a based on $\mathcal{A}$, $\textcircled{1}^{+}$, $\textcircled{1}^{-}$, λ, θ, α, β;

while $(!((\lambda = \alpha) \parallel (\theta \geq \beta)))$ {

$\alpha+ = 0.01;$

$$\beta = \frac{\xi \times (X + Y) - X \times \alpha}{Y};$$

Updated αs' and βs' value in P_a;

for $(i = 0;\ i < m;\ i++)$

for $(j = 0;\ j < n;\ j++)$ $Q_p[i,j] = Q[i,j] \times P_a[i]$;

Calculate $T(\alpha)$, and $\sigma_p(\alpha)$ with Q_p via Kuhn-Munkres Algorithm;

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

Save $\max\{\sigma\}$ and $\min\{\sigma\}$;

$$P_g(\alpha) = \frac{\sum_{\lambda} Agent + 2^{-1}\sum_{\alpha} Agent + 2^{-2}\sum_{\beta} Agent + 2^{-3}\sum_{\theta} Agent}{m};$$

Save $\max\{P_g\}$ and $\min\{P_g\}$;

Update X and Y;

}

$\alpha = \xi$;

while $(!((\lambda = \alpha) \parallel (\gamma \geq \beta)))$ {

$\alpha+ = 0.01;$

$$P_g^{\alpha} = \frac{P_g(\alpha) - \min\{P_g\}}{\max\{P_g\} - \min\{P_g\}};\ \sigma^{\alpha} = \frac{\sigma(\alpha) - \min\{\sigma\}}{\max\{\sigma\} - \min\{\sigma\}};$$

}

}

Now, we can get $\beta(1)$ by $\alpha(1)$. Via $\beta(1)$, we can get $X(1)$ and $Y(1)$ immediately. Then, we can iteratively calculate $\alpha(1)$ [and $\beta(1)$], $\alpha(2)$ [and $\beta(2)$], $\alpha(3)$ [and $\beta(3)$], ... until $\alpha \geq \lambda$ or $\beta = \theta$.

That is to say, we use

$$\beta(k+1) = \frac{\xi \times (X(k) + Y(k)) - X(k) \times \alpha(k+1)}{Y(k)}$$

where $\alpha(k+1) = \alpha(k) + 0.01$ $(k \geq 0)$. We take $\alpha(k) \geq \lambda$ or $\beta(k) = \theta$ as an exit condition of the iterative equation to calculate an appropriate α, and then get the balance point of $P_g^{\alpha,\beta}$ and $\sigma^{\alpha,\beta}$.

The process of solution 3 is Algorithm 3

Fig. 6 displays the result of solution 3. It is similar to that of solution 2, where there are intersections among the curves of $P_g^{\alpha,\beta}$ and $\sigma^{\alpha,\beta}$. As a result, we choose the biggest intersection of $P_g^{\alpha,\beta} = \sigma^{\alpha,\beta}$ as the balance point (in this case, there is only one intersection. In some other cases, there may be more than one).

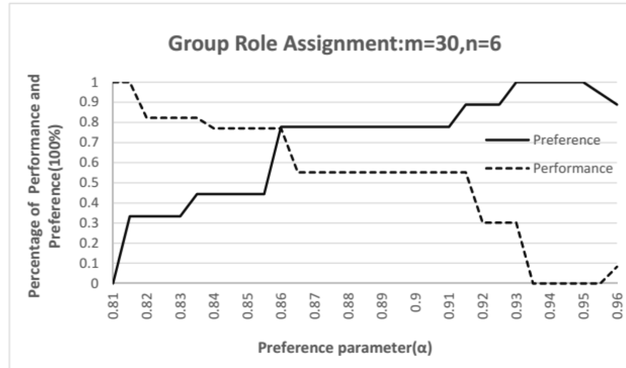


Fig. 6. Result of solution 3.

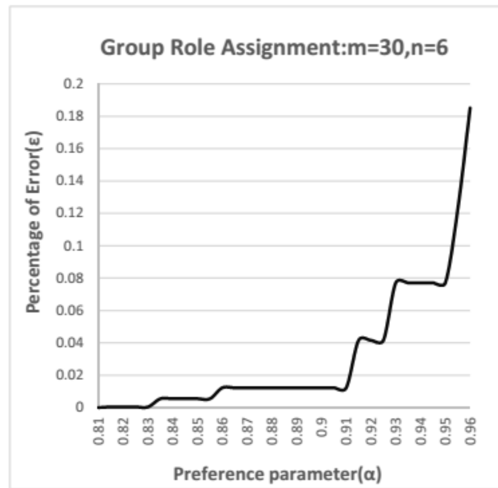


Fig. 7. Variation of ϵ in solution 3.

Therefore, in Fig. 6, P_g^{α} and σ^{α} at the balance point are both 0.782, and α is 0.86 while β is 0.76. According to balance α, the binary code of agent matrix, the assignments are shown as the $\surd$ checked numbers in Table II. The team performance σ_p is 13.49, which is not the maximum for Q but the most balanced one with the assumption of solution 3. The preference gradient P_g is 0.60.

Here, if $\alpha > 0.96$, then $\beta < 0.5 = \theta$ is invalid. That is, we need to pay attentions to computing errors due to their importance for an iterative method. In order to check the validity of the solution, we set

$$\varepsilon = \frac{\frac{X \times \alpha + Y \times \beta}{X + Y} - \xi}{\xi}$$

as an error. In the process, the variation of ε is shown as in Fig. 7. In this case, the final ε is 0.01, which can be accepted, because the acceptable error in engineering is ± 0.1 [42].

VIII. Negative Association Preference and Its Solution

In above cases, we debate how to balance preferences and performance via positive association preference mining

$$\begin{bmatrix} 0&1&1&1\\ 0&1&1&0\\ 1&1&0&0\\ 0&0&0&1\\ 0&1&0&0\\ 1&1&0&0\\ 0&0&1&1\\ 1&0&0&1\\ 0&0&0&1\\ 1&1&0&1\\ 0&0&0&0\\ 1&0&1&1\\ 0&1&1&1\\ 0&1&1&0\\ 0&1&0&1\\ 0&1&1&1\\ 1&0&0&0\\ 0&1&1&0\\ 1&1&0&1\\ 1&0&0&0\\ 1&0&1&1\\ 1&0&1&1\\ 1&1&0&1\\ 1&1&1&1\\ 0&0&0&0\\ 1&1&0&1\\ 1&1&0&1\\ 0&1&1&1\\ 1&1&1&1\\ 1&1&1&0 \end{bmatrix} \quad \begin{bmatrix} 0.9&0.9&0.45&0.33&0.48&0.93\\ 0.63&0.48&0.2&0.46&0.57&0.78\\ 0.41&0.46&0.51&0.67&0.62&0.4\\ 0.95&0.54&0.64&0.39&0.75&0.89\\ 0.27&1.0&0.27&0.94&0.17&0.59\\ 0.64&0.81&0.15&0.74&0.69&0.27\\ 0.67&0.47&0.18&0.65&0.82&0.32\\ 0.54&0.55&0.15&0.18&0.93&0.63\\ 0.72&0.67&0.97&0.72&0.53&0.61\\ 0.99&0.75&0.63&0.33&0.7&0.17\\ 0.81&0.96&0.93&0.13&0.58&0.28\\ 0.12&0.41&0.48&0.24&0.36&0.62\\ 0.32&0.4&1.0&0.65&0.75&0.37\\ 0.52&0.82&0.1&0.79&0.87&0.61\\ 0.58&0.69&0.3&0.23&0.65&0.55\\ 0.52&0.56&0.79&0.15&0.57&0.8\\ 0.53&0.5&0.58&0.46&0.33&0.2\\ 0.28&0.97&0.94&0.25&0.63&0.65\\ 1.0&0.7&0.36&0.54&0.43&1.0\\ 0.85&0.85&0.58&0.32&0.44&0.16\\ 0.85&0.43&0.13&0.26&0.36&0.63\\ 0.28&0.74&0.21&0.6&0.99&0.7\\ 0.95&0.47&0.35&0.52&0.55&0.2\\ 0.57&0.63&0.29&0.97&0.42&0.97\\ 0.69&0.75&0.83&0.29&0.58&0.21\\ 0.13&0.76&0.53&0.37&0.5&0.29\\ 0.79&0.13&0.75&0.3&0.76&0.52\\ 0.89&0.5&0.63&0.56&0.83&0.99\\ 0.78&0.82&0.76&1.0&0.48&0.82\\ 0.85&0.48&0.82&0.5&0.38&0.99 \end{bmatrix}$$

(a) (b)

Fig. 8. Matrices of the new case. (a) Binary code matrix. (b) Qualification matrix Q.

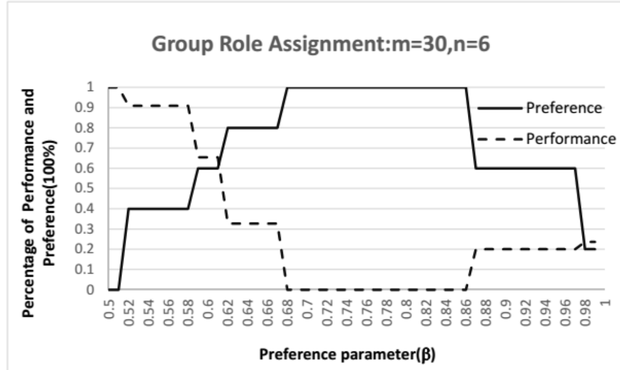


Fig. 9. Result of the fundamental solution of new case.

TABLE VII
ASSOCIATION NEGATIVE PREFERENCE SET AND MIDDLE PREFERENCE SET. (a) NEGATIVE ASSOCIATION PREFERENCE IN BINARY. (b) NEW MIDDLE PREFERENCE IN BINARY

(a)

Binary	Gender	Department	Experience
1101	M	Marketing	3-5 years' experience
0101	F	Marketing	3-5 years' experience

(b)

Binary	Gender	Department	Experience
0010	F	Sales	1-3 years' experience
0110	F	Marketing	1-3 years' experience
0111	F	Marketing	<1 year's experience
1011	M	Sales	<1 year's experience

at a fine-grain scale. To change values of α and β by our method, one can obtain a solution.

Note that, the precondition for us to mine the positive association preference set is that β is bigger than 0.75 in solution 1. If β is smaller than 0.75, the settings of solutions 2 and 3 are invalid. Therefore, we still need to process cases while β is smaller than 0.75. To solve them, we need to mine the negative association preference set rather than the positive one.

With the OCAT method, we can exchange the positive training set with the negative one. The rest remains the same. Then, we can acquire the negative association preference set. To illustrate these, we give another example, which is shown in Fig. 8.

This case is different from the above example in the qualification matrix Q. The binary scheme is not changed.

Based on solution 1, when β is 0.61 (smaller than 0.75), and P_g^{α} and σ^{α} are 0.620, the solution is balanced (P_g is 0.635 and σ is 13.85), which is shown in Fig. 9.

Because β is smaller than 0.75, if we interchange positive and negative preferences, then we get the rule of the negative association preference via OCAT as the following:

$$(\neg A_1 \vee A_2 \vee \neg A_4) \wedge (\neg A_2 \vee A_3 \vee A_4) \wedge (A_1 \vee \neg A_3).$$

Now, we get the negative association preference set and the new middle preference set shown as Table VII.

Note that 1101 is always in the positive association preference set, because the intersection set of positive and negative association preference is not empty. This is determined by the data set itself. One can deal with 1101 as the middle preference, or still deal with it as the negative (positive) association preference. Because the emphasis of this paper is not on data mining, we ignore the detailed process.

We set the weight of the negative association preference set as γ, the weight of the middle preference set still as β, and P_g as the following:

$$P_g = \frac{\sum_{\lambda} \text{Agent} + 2^{-1}\sum_{\beta} \text{Agent} + 2^{-2}\sum_{\gamma} \text{Agent} + 2^{-3}\sum_{\theta} \text{Agent}}{\sum_{j=0}^{n-1} L[j]} \in (0, 1).$$

With these settings, solution 2 is transformed to the solution with the variable γ and constant β. We still set β as 0.75, but the initial γ is 0.5. The process adds 0.01 to γ after every iteration until γ is 0.75. Functions become unary functions on γ, and other calculations remain the same. Then, we get the solution, i.e., when γ is 0.57, P_g^{γ} and σ^{γ} are 0.545, the preferences and the performance are balanced (P_g is 0.635 and σ is 13.34), as shown in Fig. 10.

Solution 3 is transformed to a solution in which both γ and β are variables. We use ξ (0.61, result of solution 1), $X(0)$ and $Y(0)$ as the prerequisite knowledge, γ changes from 0.61 to 0.5 by a step of 0.01, and then the result is: when γ is 0.59, β is 0.70, and P_g^{γ} and σ^{γ} are 0.500, the preferences and performances are balanced (P_g is 0.633 and σ is 13.13) shown in Fig. 11. The error variation is shown as in Fig. 12.

In this case, ε of the final result is 0.06, which can be accepted in applications [42].

To draw a conclusion, if β is bigger than 0.75 in solution 1, then a better result can be obtained by mining the positive preference set in solutions 2 and 3. Otherwise, if β is smaller than

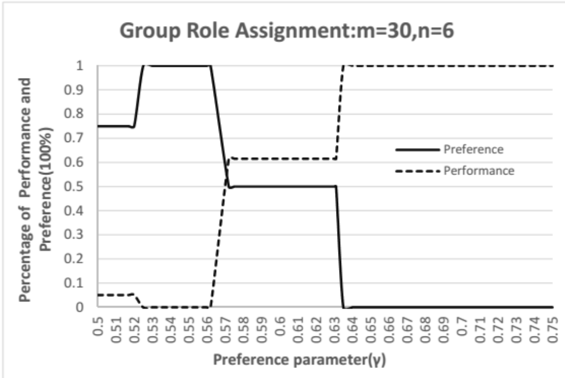


Fig. 10. Result of solution 2 with γ instead of α for the new case.

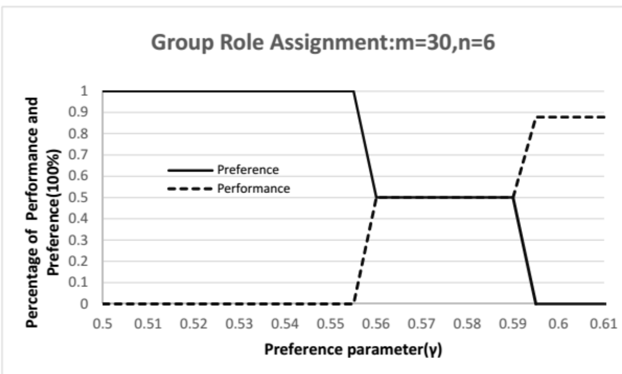


Fig. 11. Result of variable γ and β solution in the new case.

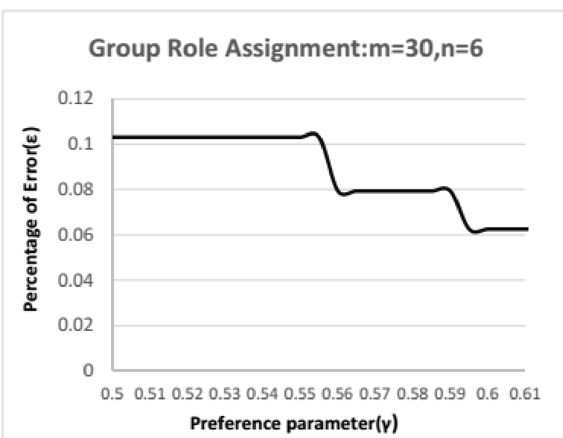


Fig. 12. Variation of ϵ *of variable* α *and* β *solution*.

0.75, then the negative preference set is appropriate. Actually, preferences can be divided into many segments, such as the case of this section. If we use the middle preference set as the negative training set and the positive one as the positive training set, then we can get the new positive association preference set.

IX. Simulation Experiments

To check the applicability of the three solutions, we conduct experiments with random groups by different scales, where m is from 30 to 120 with an increment of 30, $n = m/5$, $\sum_{j=0}^{n-1} L[j]$(abbr. $\Sigma L) = m/2, 0 < L[j] < 5$ $(0 \leq j < n)$, and the binary bits of attributes are from 4 to 16 with an increment of 4. By "a random group," we mean a group with a Q matrix whose values are produced randomly, after m, and n are chosen.

For each scale, we produce 100 random groups and collect performances, the error statistics of solution 3 and times used by three solutions (Fig. 13). The experiments are based on the platform shown in Table VIII.

The experiments indicate that although team performances from solutions 1–3 have no significant changes, but the balances become more and more accurate, which are more appropriate for decision makers to apply.

Table IX indicates the difference between the performances of the three solutions and that of the original GRA, where "maximum D-value" means the maximum difference between the performances of a solution and GRA among 100 random groups, "minimum D-value" the minimum one, and "average D-value" the average one. "Ratio of average D-value" means the ratio of the average difference compared with the performance of GRA.

TABLE VIII
Test Platform Configuration

Hardware	
CPU	Intel Pentium Dual-Core E5300 CPU@2.6GHz
MM	4GB
Software	
OS	Windows 7 Enterprise
Eclipse	Luna Service Release 2 (4.4.2)
JDK	Java 7 Update (45)

TABLE IX
D-Values Among the Three Solutions With Original Solution. (a) Solution 1. (b) Solution 2. (c) Solution 3

(a)

Scale of *m* (*n*)	Maximum D-Value	Minimum D-Value	Average D-value	Ratio of Average D-value
30 (6)	-1.77	-0.33	-0.828	-0.0582
60 (12)	-3.83	-0.19	-1.578	-0.0559
90 (18)	-4.45	-0.06	-2.186	-0.0515
120 (24)	-7.15	-0.55	-3.351	-0.0577

(b)

Scale of *m* (*n*)	Maximum D-Value	Minimum D-Value	Average D-Value.	Ratio of Average D-value
30 (6)	-1.58	-0.01	-0.539	-0.0379
60 (12)	-3.24	-0.07	-0.905	-0.0321
90 (18)	-5.72	-0.04	-1.395	-0.0329
120 (24)	-5.78	-0.10	-2.079	-0.0358

(c)

Scale of *m* (*n*)	Maximum D-Value	Minimum D-Value	Average D-Value	Ratio of Average D-value
30 (6)	-1.77	-0.11	-0.842	-0.0592
60 (12)	-3.89	-0.12	-1.873	-0.0664
90 (18)	-5.63	-0.24	-2.759	-0.065
120 (24)	-6.89	-0.92	-3.363	-0.0579

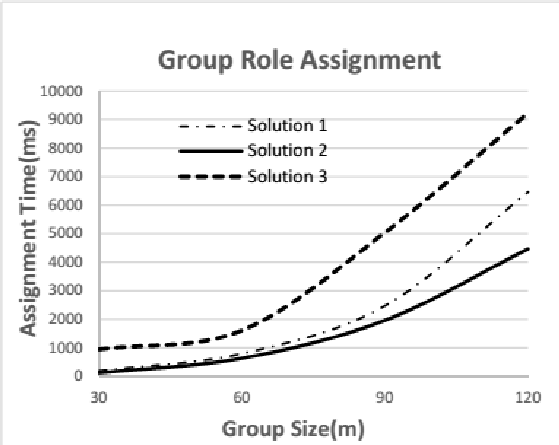


Fig. 13. Time cost comparison of three solutions.

On the other hand, in order to prove the validity of the iteration of solution 3, we do error statistics. The experiment result indicates that errors are in the controllable range [42].

At last, we do the time cost experiment, where Fig. 13 shows the average times used by the three solutions. The results show that our solutions are practical.

Note that, preference and performance may not be always at the opposite ends. If preference and performance are consistent, our proposed method does not have much influence. If good performers obtain nicer preferences, then they have better opportunities to be chosen via the Kuhn–Munkres algorithm, because their values in the preferred qualification matrix Q_p (see Definition 13) are less influenced. For example, Fide in Table II, who is a good performer having a nice preference. He is chosen to do Ad. in all of the three solutions.

X. Conclusion

GRAB is an important, interesting, and complex problem. We implement a fundamental approach to solve it. Based on this solution, we give a series of improved methods that combine with the OCAT logical association rule mining algorithm and an iteration method. Experiments indicate that the improved approaches are effective and practical.

1) The preferences and performance are influenced by each other.
2) If middle preference β becomes bigger, then it tends to be positive preference, vice versa.
3) If β is big enough (i.e., $\beta > 0.75$), then we can mine and separate positive association preferences from it; else if β is small enough (i.e., $\beta < 0.75$), then we can mine and separate negative association ones.
4) This is a bisection and iteration process, decision makers are only requested to provide "coarse" preferences. Our methods can mine that "fine" preferences from the coarse ones and find a more accurate balance in the fine-grain scale via the GRA algorithm.

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

1) Although the proposed algorithm is fine-grain, it is still possible to make the solutions finer.

TABLE X
Nomenclature

Symbol	Meaning
P	Positive preference
N	Negative preference
M	No preferences (middle preference)
C	A set of classes
O	A set of objects
$\mathcal{A}$	A set of agents
$\mathcal{M}$	A set of messages
$\mathcal{R}$	A set of roles
$\mathcal{E}$	A set of environments
$\mathcal{G}$	A set of groups
s_0	The initial state of a system
$\mathcal{H}$	A set of users
m	$=\lvert\mathcal{A}\rvert$, the size of the agent set $\mathcal{A}$
n	$= \lvert\mathcal{R}\rvert$, the size of the role set $\mathcal{R}$
$i_0, i_1, i_2, \ldots$	The indices of agents
$j_0, j_1, j_2, \ldots$	The indices of roles
k	A free variable
$\mathcal{N}$	The set of nonnegative integers
Ⓢ	A property Set.
r	A role
a	An agent
L	The role range vector
Q	The qualification matrix
T	The role assignment matrix
σ	The group performance
Ⓘ	The property preference list
P_a	The preference vector
Q_p	The preferred qualification matrix
σ_p	The preferred group performance
λ	The positive preference weight
θ	The negative preference weight
α	The positive association preference weight
β	The middle preference weight
γ	The negative association preference weight
P_{avg}	The average preference value of the chosen agents
P_g	The preference gradient
P_g^α (P_g^β and $P_g^{\alpha,\beta}$)	The normalized preference gradient corresponding to different preference weight
σ^α (σ^β and $\sigma^{\alpha,\beta}$)	The normalized group performance corresponding to different preference weight
ξ	The result of Solution 1
$X(0)$	The total number of the chosen agents (that are assigned eventually) referred by α under ξ in Solution 1
$Y(0)$	The total number of chosen agents (that are assigned eventually) referred by β in Solution 1
ε	Error
A_k	The binary code of preferences of staffs' properties
P^+	The positive preference set
P^-	The negative preference set
C	A CNF expression
c_k	A disjunction clause
$\sum L$	The abbreviation of $\sum_{j=0}^{n-1} L[j]$

2) The preferences in this paper are common, there are still some more special and complicated preferences we need to investigate and then balance with performance.
3) Because preferences and performance are influenced by each other, it is not the best appropriate method to measure the two factors separately. We may need

$$
\underset{\text{(a)}}{\begin{bmatrix}
0.55 & 0.21 & 0.99 & 0.63 & 0.87 & 0.84 \\
0.96 & 0.78 & 0.45 & 0.56 & 0.81 & 0.23 \\
0.66 & 0.56 & 0.27 & 0.46 & 0.4 & 0.55 \\
0.50 & 0.38 & 0.74 & 0.27 & 0.76 & 0.61 \\
0.71 & 0.54 & 0.24 & 0.16 & 0.23 & 0.34 \\
0.44 & 0.65 & 0.96 & 0.36 & 0.32 & 0.84 \\
0.35 & 0.13 & 0.38 & 0.23 & 0.37 & 0.72 \\
0.56 & 0.18 & 0.43 & 0.69 & 0.75 & 0.27 \\
0.49 & 0.97 & 0.83 & 0.63 & 0.55 & 0.76 \\
0.78 & 0.80 & 0.35 & 0.98 & 0.78 & 0.61 \\
0.13 & 0.89 & 0.24 & 0.52 & 0.62 & 0.36 \\
0.28 & 0.21 & 0.94 & 0.30 & 0.23 & 0.84 \\
0.11 & 0.74 & 0.89 & 0.38 & 0.88 & 0.87 \\
0.23 & 0.89 & 0.62 & 0.96 & 0.30 & 0.67 \\
0.84 & 0.79 & 0.52 & 0.25 & 0.59 & 0.51 \\
0.55 & 0.32 & 0.54 & 0.60 & 0.24 & 0.92 \\
0.98 & 0.53 & 0.60 & 0.78 & 0.95 & 0.19 \\
0.61 & 0.18 & 0.65 & 0.19 & 0.84 & 0.62 \\
0.24 & 1.0 & 0.43 & 0.94 & 0.87 & 0.75 \\
0.60 & 0.28 & 0.52 & 0.42 & 0.95 & 0.92 \\
0.90 & 0.10 & 0.37 & 0.92 & 0.10 & 0.35 \\
0.61 & 0.10 & 0.46 & 0.71 & 0.19 & 0.63 \\
0.67 & 0.34 & 0.73 & 0.28 & 0.23 & 0.5 \\
0.87 & 0.92 & 0.24 & 0.21 & 0.40 & 0.41 \\
0.68 & 0.88 & 0.17 & 0.97 & 0.53 & 0.36 \\
0.29 & 0.72 & 0.62 & 0.22 & 0.76 & 0.97 \\
0.34 & 0.64 & 0.40 & 0.96 & 0.67 & 0.99 \\
0.93 & 0.9 & 0.31 & 0.95 & 0.19 & 0.68 \\
0.11 & 0.24 & 0.85 & 0.39 & 0.95 & 0.36 \\
0.11 & 0.87 & 0.73 & 0.30 & 0.83 & 0.34
\end{bmatrix}}
\quad
\underset{\text{(b)}}{\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}
$$

Fig. 14. Sample matrices. (a) Qualification matrix Q. (b) Assignment matrix T.

to study more strategies to synthesize preferences and performance into an integrated parameter and try to optimize it with relevant constraints.

4) There may be additional factors that require further investigations around the problem of GRA.
5) It is a meaningful task to investigate a more efficient way to find the best parameters, i.e., α, β, and γ, for GRAB to save time, because many parts in Figs. 5, 6, and 9–11 show that P_g is evidently far away from σ.

Appendix

See Fig. 14 and Table X.

Acknowledgment

The authors would like to thank M. Brewes of Nipissing University for his assistance in proofreading this paper.

References

[1] 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.

[2] H. Zhu and M. C. 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.

[3] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Trans. Inf. Syst. Security*, vol. 5, no. 4, pp. 492–540, Nov. 2002.

[4] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Trans. Inf. Syst. Security*, vol. 4, no. 3, pp. 224–274, Aug. 2001.

[5] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Trans. Inf. Syst. Security*, vol. 2, no. 1, pp. 105–135, Feb. 1999.

[6] H. Zhu and M. C. Zhou, "Roles in information systems: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 377–396, May 2008.

[7] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistic Quart.*, vol. 2, pp. 83–97, 1955, (Reprinted in vol. 52, no. 1, pp. 7–21, 2005).

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

[9] 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.

[10] S. Lichtenstein and P. Slovic, *The Construction of Preference*. New York, NY, USA: Cambridge Univ. Press, 2006.

[11] K. R. Scherer, "What are emotions? And how can they be measured?" *Soc. Sci. Inf.*, vol. 44, no. 4, pp. 695–729, 2005.

[12] S. Son and K. M. Sim, "A price- and-time-slot-negotiation mechanism for cloud service reservations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 713–728, Jun. 2012.

[13] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems.* Philadelphia, PA, USA: SIAM, 2012.

[14] 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.

[15] D. E. Knuth, *The Art of Computer Programming*, vol. 1, 3rd ed. Upper Saddle River, NJ, USA: Addison-Wesley, 1997.

[16] 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.

[17] H. Zhu *et al.*, "Solving the many to many assignment problem by improving the Kuhn–Munkres algorithm with backtracing," *Theor. Comput. Sci.*, vol. 618, pp. 30–41, Mar. 2016.

[18] J. Wang, Y. Deng, and C. Jin, "Compositional modeling and performance evaluation of traffic control systems via stochastic timed Petri nets," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 10, no. 6, pp. 735–757, 2000.

[19] M. A. Bashar, A. Obeidi, D. M. Kilgour, and K. W. Hipel, "Modeling fuzzy and interval fuzzy preferences within a graph model framework," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 765–778, Aug. 2016, doi 10.1109/TFUZZ.2015.2446536. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7126962

[20] L. Fang, K. W. Hipel, D. M. Kilgour, and X. Peng, "Scenario generation and reduction in the decision support system GMCR II," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Orlando, FL, USA, vol. 1. Oct. 1997, pp. 341–345.

[21] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[22] Z. Xu, "Deviation square priority method for distinct preference structures based on generalized multiplicative consistency," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 1164–1180, Aug. 2015.

[23] Y. Zhang, "Simulating human social behaviors," in *Concepts and Methodologies for Modeling and Simulation* (Simulation Foundations, Methods and Applications), L. Yilmaz, Ed. Cham, Switzerland: Springer, 2015, pp. 289–319.

[24] J. Liu, Y. Xiong, C. Wu, Z. Yao, and W. Liu, "Learning conditional preference networks from inconsistent examples," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 376–390, Feb. 2014.

[25] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[26] S. E. Salterio, "Barriers to knowledge creation in management accounting research," *J. Manag. Account. Res.*, vol. 27, no. 1, pp. 151–170, 2015.

[27] A. Mohammadi, M. N. Omidvar, and X. Li, "A new performance metric for user-preference based multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, Jun. 2013, pp. 2825–2832.

[28] C. Olaverri-Monreal, A. E. Hasan, J. Bulut, M. Körber, and K. Bengler, "Impact of in-vehicle displays location preferences on drivers' performance and gaze," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1770–1780, Aug. 2014.

[29] M. Zeydan, C. Çolpan, and C. Çobanoğlu, "A combined methodology for supplier selection and performance evaluation," *Exp. Syst. Appl.*, vol. 38, no. 3, pp. 2741–2751, Mar. 2011.

[30] N. Melville, K. Kraemer, and V. Gurbaxani, "Review: Information technology and organizational performance: An integrative model of it business value," *J. MIS Quart.*, vol. 28, no. 2, pp. 283–322, Jun. 2004.

[31] J. M. Bradshaw *et al.*, "Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications," in *Agent Autonomy*, H. Hexmoor, R. Falcone, and C. Castelfranchi, Eds. Dordrecht, The Netherlands: Kluwer, pp. 243–280, 2003.

[32] G. Fortino, F. Rango, and W. Russo, "Engineering multi-agent systems through statecharts-based JADE agents and tools," in *Transactions on Computational Collective Intelligence* (LNCS 7270), N. T. Nguyen Ed. Heidelberg, Germany: Springer-Verlag, 2012, pp. 61–81.

[33] W. Shen, L. Wang, and Q. Hao, "Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 563–577, Jul. 2006.

[34] 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.

[35] 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. Heidelberg, Germany: Springer, 2004, pp. 214–230.

[36] 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.

[37] 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.

[38] 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.

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

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

[41] 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.

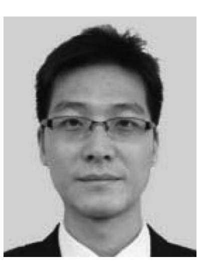
[42] R. G. Hills and T. G. Trucano, "Statistical validation of engineering and scientific models: Background," Sandia Nat. Lab., Livermore, CA, USA, Tech. Rep. SAND99-1256, 1999. [Online]. Available: http://www4.ncsu.edu/~rsmith/Hills-1.pdf

[43] E. Triantaphyllou, *Data Mining and Knowledge Discovery via Logic-Based Methods*. New York, NY, USA: Springer, 2010.

Dongning Liu (M'17) was born in 1979. He received the Ph.D. degree in logic from Sun Yat-sen University, Guangzhou, China, in 2007.

He is an Associate Professor with the Guangdong University of Technology, Guangzhou, where he is responsible for teaching discrete mathematics with the School of Computer Science and Technology. His current research interests include education and technology transfer on artificial intelligent logic and collaborative computing. He has published over 30 papers on computer magazines and international conferences.

Dr. Liu is a Reviewer for the *IEEE Systems, Man, and Cybernetics Magazine* and the *Journal of Computer*. He is a Committee Member of the Technical Committee on Collaborative Computing of Computer Academy of Guangdong and Guangdong Logic Association, and a member of IEEE.



Yunyi Yuan was born in 1989. He received the B.S. degree in computer science and technology from the School of Software, Zhengzhou University, Zhengzhou, China, in 2013. He is currently pursuing the master's degree in collaborative computing with the School of Computer, Guangdong University of Technology, Guangzhou, China.

He has co-authored a research paper on collaboration technologies.

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, 1988 and 1997, respectively.

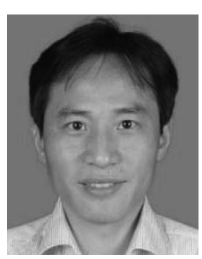
He is a Full Professor with the Department of Computer Science and Mathematics, and the Founder and the Director of Collaborative Systems Laboratory, Nipissing University, North Bay, ON, Canada, and a Visiting Professor with the Department of Control Science and Engineering, Nanjing University, Nanjing, 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 160+ research papers, four books, and four book chapters on object-oriented programming, distributed systems, collaborative systems, and computer architecture.

He was a recipient of the 2011 Chancellors' Award for excellence in research and the 2006 and 2011 Research Achievement Awards from Nipissing University, the 2004 and 2005 IBM Eclipse Innovation Grant Awards, the Best Paper Award from the 2004 ISPE International Conference on Concurrent Engineering, the Educator's Fellowship of OOPSLA'03, 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, 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, and the Second Class Excellent Textbook Award of the Ministry of Electronics Industry of China in 1996. 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*, the 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.

Shaohua Teng was born in 1962. He received the Ph.D. degree in industry engineering from the Guangdong University of Technology, Guangzhou, China.

He is a Professor with the Guangdong University of Technology, where he has been a Teacher, since 1982, responsible for teaching data mining with the School of Computer Science and Technology. He is engaged in education and technology transfer on knowledge discovery issues, and research in big data, role assignment, network security, cooperative computing, and machine learning. He has applied for seven patents on his invention. He has published 200 papers on computer magazines and international conferences. He has two books published.

Dr. Teng was a recipient of the Provincial Science and Technology Award, and the Guangdong Outstanding Teacher.



Changqin Huang (M'16) was born in 1972. He received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2005.

He is currently a Professor with the School of Information Technology in Education, South China Normal University, Guangzhou, China. He was a Visiting Scientist with Zhejiang University. His current research interests include service computing, big data, semantic Web, and enabling technologies for Web-based education.

Dr. Huang is a Senior Member of CCF, and a member of IEEE and ACM.