# Solving the Many to Many Grouped Task Allocation Problem via E-CARGO

Mozhdeh Noroozi Rasoolabadi, Haibin Zhu and Chun Wang



October 23, 2023

# Solving the Many to Many Grouped Task Allocation Problem via E-CARGO

Mozhdeh Noroozi Rasoolabadi
*Institute for Information Systems Engineering*
*Concordia University*
Montreal, Canada
mozhdeh.noroozirasoolabadi@mail.concordia.ca

Haibin Zhu, *Senior Member, IEEE*
*Department of Computer Science and Mathematics*
*Nipissing University*
North Bay, Ontario, Canada
haibinz@nipissingu.ca

Chun Wang
*Institute for Information Systems Engineering*
*Concordia University*
Montreal, Canada
chun.wang@concordia.ca

***Abstract*— The task allocation between team members of each organization's departments is a crucial project management activity that needs adequate attention to improve organizational efficiency. If a related decision is not made properly, it may immediately cause problems with coordination and communication, task delays, and increased costs. Tasks are usually grouped and then these groups are assigned to the agents. Moreover, it is important to consider the difficulty level of the tasks and the member's ability. The task allocation process becomes increasingly complex when administrators strive to maintain the balance between the workload of allocated task groups and the agents' capabilities. The E-CARGO (Environments - Classes, Agents, Roles, Groups, and Objects) model, created for the purpose of Role-Based Collaboration (RBC), has been effectively utilized to address the given problem. In this paper, a novel approach is presented to formalize the Grouped Task Allocation Problem (GTAP) employing an extended Group Multirole Assignment with Conflicting Roles problem (GMRACR), which is a sub-model of E-CARGO. In this paper, a role negotiation method is introduced, which is based on GMRACR. It employs a partitioning clustering algorithm along with the Analytic Hierarchy Process (AHP) to evaluate the difficulty level of tasks and the abilities of agents, respectively. This formalization of GTAP facilitates the identification of a solution using the IBM ILOG CPLEX optimization package (CPLEX). The proposed method helps decision-makers with a framework to successfully assign balanced grouped tasks. Through simulation experiments on a real-world problem, the efficacy of the suggested approach is substantiated. Experimental results reveal the practicality of the solutions recommended in this paper.**

***Keywords*— *Role-Based Collaboration (RBC), Group Multirole Assignment with conflicting roles (GMRACR), Grouped task allocation, Partitioning clustering method, IBM ILOG CPLEX optimization (CPLEX) package.***

## I. INTRODUCTION

The purpose of task allocation is to assign agents (often workers) to tasks (typically jobs) to maximize overall effectiveness or reduce overall cost/time. A general assignment problem is a specific instance of the transportation problem and is sometimes referred to as maximum-weight matching in the literature [1]. The problem of grouped task assignment (GTA) in project management emphasizes assigning grouped tasks in such a way that there is a balance between the ability of the workers and the difficulty of the tasks assigned to them. In project management in the real world, GTA is frequently used. For example, an organization with a matrix organizational structure [2] combines structures of the projects and functions. In this kind of organization, each project manager (PM) may handle some projects (tasks) of a specific type and each project has to be controlled and monitored by a project management officer (PMO). Hence, the senior manager of the PMO department has to assign PMOs to the PMs' projects to control and monitor them. For allocation, the senior manager should consider different constraints, such as assigning the same type of projects to each PMO. Also, considering that the workload of each project is different, the senior manager should not assign more projects to PMOs than their ability to control, and in other words, maintain a balance so that there is no dissatisfaction among PMOs. Because the projects of each PM are of the same type and PMOs should be assigned projects of the same type, all the projects of each PM can be considered as a group task and instead of assigning PMOs to projects, PMOs are assigned to each PM to control and monitor his/her projects. (Fig. 1)

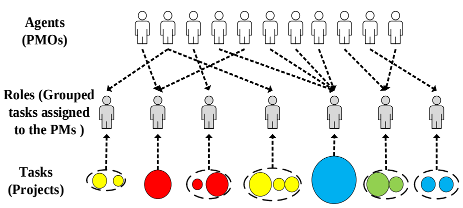


Fig. 1. A GTAP graph. A circle represents a project as a task set. The color of a circle shows the type of a project and the size demonstrates the difficulty level, which is obtained based on the partitioning clustering method.

The above assignment problem cannot be specified by traditional assignment problems Because the difficulty level of projects and the agents' ability should be considered in the allocation. Since in the project's life cycle, two or more tasks may be performed by one staff member within the same scope, we have a many-to-many (M2M) assignment problem. M2M problems are typically addressed by employing heuristic approaches or auction algorithms to attain a globally near-optimal solution, as assignment solutions incorporating multiagent systems tend to be time-consuming[3]–[5].

Fortunately, the Role-Based Collaboration (RBC)[7] along with its E-CARGO (Environments - Classes, Agents, Roles, Groups, and Objects) model offers a more promising approach to address such problems. RBC utilizes roles to facilitate collaboration and coordination within systems. Through RBC and E-CARGO, complex problems can be efficiently formalized and solved. The RBC life cycle involves processes such as role negotiation, agent evaluation, role assignment, role-playing, and role transfer, with role negotiation being a highly domain-oriented and complex task[7]–[10]. In this paper, a novel approach is presented to incorporate GTAP as an M2M assignment within the E-CARGO framework. This framework suggests that an M2M assignment can be managed using group multirole assignment with conflicting roles (GMRACR)[6], which is a significant component of group role assignment (GRA). GRA is a sub-model of E-CARGO, which serves as a general model for RBC[7]. GRA[10] aims to achieve an optimal assignment of roles to agents based on individual assessments. Contrary to GRA, which only allows one agent to be assigned to one single role, GMRACR allows one role to be given to many, but distinct, agents and one agent to perform many, but different and nonconflicting roles. In this paper, considering that only roles that have the same scope can be assigned to an agent, and we aim to achieve a balance between workload and an agent's capabilities, the GMRACR cannot directly solve it, hence, we present the GTAP model, which is extended to the GMRACR. After applying RBC, we use a partitioning clustering method to create sets of difficulty levels for the tasks in each group. Then, we apply the Analytic Hierarchy Process method (AHP) to determine each agent's ability. After fairly evaluating each agent's roles, we can then apply an efficient assignment algorithm, which can strike a balance between the workload of the grouped tasks allocated and the agents' abilities. Employees can be thought of as agents and roles are simply groups of tasks (jobs). The following are some of the contributions made by this paper. (1) The novel formalization of the grouped task allocation considering the balance between workload and agents' ability via GTAP which is extended of GMRACR; (2) The introduction of difficulty levels of tasks in GTA and the determination of them using partitioning clustering; (3) Useful solutions to the GTAP, such as the IBM ILOG CPLEX[11] optimization package (CPLEX) solution. (4) Show that using GTAP improves the real performance and centrality of the system compared to using GMRACR. This paper is organized as follows: Section II discusses a real-world scenario of the proposed problem, and Section III presents the GTAP along with the revised GMRACR and ECARGO models. Section IV introduces the CPLEX solution and simulation experiments, the outcomes of which substantiate the effectiveness of the proposed solution. Section V reviews the relevant literature, while Section VI summarizes the work and highlights potential future work.

## II. A REAL-WORD SCENARIO

Company X provides telecommunication and IT management services in Tehran, Iran. In this company, various design, implementation, and maintenance services are implemented in the form of projects under the supervision of PMs. The Chief Executive Officer of the company, Maryam, asked Afshin, the senior manager of the PMO department, to assign his existing PMOs to the projects to manage and control them. A PMO can handle some tasks (projects), and a task may require many PMOs. Afshin evaluates PMOs for each possible project (Table I) and drafts a PM list including the project information as shown in Table II. Afshin wants to follow the scenario that each PMO can only work on a certain type of project. He knows a GMRACR model may help solve this problem. Afshin figures out that the amount of work that PMOs have to do for each project is different, for example, the NI projects take more time and effort than the NPO projects from PMOs. To indicate the complexity level, he gathered additional information about each project's characteristics and put it in a table (Table III). Afshin knows that applying this difficulty level can establish a balance between PMOs' workloads. To establish roles and agents, perform agent evaluations, and conduct simulation tests, he must put up a lot of effort. Now, Afshin must figure out 1) How to cluster projects according to their degree of difficulty and then quantify that degree; 2) Evaluate the candidates' ability 3) How to assess a candidate's suitability for a project; 4) How to centralize and optimize the performance of the entire PMO.

TABLE I. PMO CANDIDATES AND PROJECT TYPE EVALUATIONS

| | **NI** | **NPO** | **NO** | **SAM** | **DC** | **IT** |
|---|---|---|---|---|---|---|
| Ali | 1.6 | 7.4 | 8.4 | 5.2 | 7.1 | 4.8 |
| Sam | 6.5 | 0.7 | 9.4 | 8.9 | 9.6 | 6.2 |
| Rose | 7.4 | 1.4 | 8.3 | 1.4 | 9.7 | 4.4 |
| Mia | 3.0 | 6.7 | 8.0 | 0.9 | 2.0 | 6.4 |
| Zoe | 8.1 | 6.1 | 2.5 | 7.2 | 3.6 | 7.9 |
| Eli | 0.3 | 8.1 | 1.6 | 2.2 | 5.4 | 7.4 |
| Eva | 6.4 | 5.2 | 1.2 | 2.0 | 1.6 | 7.8 |
| Ian | 5.0 | 5.7 | 9.0 | 0.6 | 3.8 | 1.8 |
| Ivy | 9.0 | 8.9 | 9.0 | 8.9 | 5.3 | 3.0 |
| Kai | 1.6 | 7.4 | 8.4 | 5.2 | 7.1 | 4.8 |
| Tim | 10 | 5.1 | 1.7 | 6.5 | 7 | 8.5 |
| Nora | 7.4 | 3.5 | 8.3 | 5 | 9.7 | 4.4 |
| Lili | 1.5 | 9.2 | 7.2 | 6.3 | 1.6 | 7.8 |
| Li | 4.5 | 6.5 | 8.5 | 7.2 | 3.6 | 7.9 |
| Mona | 6.4 | 5.2 | 1.2 | 5.0 | 1.6 | 7.8 |
| Nina | 7.4 | 2.4 | 8.5 | 1.4 | 9.7 | 4.4 |
| Hadi | 10 | 5.1 | 1.7 | 6.5 | 6 | 8.5 |
| Mari | 2 | 8 | 8.4 | 5.2 | 7.1 | 5 |

Note: This Number is between 0 and 10.

TABLE II. PROJECT INFORMATION

| **PM** | **Nader** | **Mahsa** | **Kia** | **Sara** | **Babak** | **Mina** | **Reza** | **Zahra** | **Dana** | **Arash** | **Donya** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N.P | 3 | 5 | 3 | 4 | 1 | 2 | 1 | 4 | 1 | 4 | 3 |
| T.P | NI | NO | NPO | SAM | NI | IT | NI | NPO | NO | NI | DC |

Note: PM means Project Manager// N.P means Number of Projects assigned to each PM// T.P means Type of Project // NI means Network Implementation // NPO means Network Planning and Operation // NO means Network Optimization // SAM means Site Acquisition // DC means Data Center // IT means Information and Technology.

TABLE III. A SAMPLE OF ATTRIBUTES OF PROJECT

| PM | T. P | Project | #DR | #DM | #CE | AT | CS | CM | AH |
|---|---|---|---|---|---|---|---|---|---|
| Mahsa | NO | P1 | 0.2 | 0.2 | 0.5 | 0.25 | 1 | 1 | 0.4 |
| | | P2 | 0.4 | 0.4 | 0.5 | 0.25 | 0 | 0 | 0.2 |
| | | P3 | 0 | 0.2 | 0 | 0 | 0 | 0 | 0.2 |
| | | P4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | P5 | 0.2 | 0.2 | 0 | 0 | 0 | 0 | 0.2 |

Note: #DR means the average number of daily Report// #DM means the average number of daily meetings// #CE means the average number of daily cost evaluation should be done// AT means Average time required to prepare a cost estimate // CS means control of project service//CM means control of project material// AH shows ad-hoc activity, 1 means the least ad hoc activity and 5 means the highest level of ad hoc activity.

To address the initial challenge of accurately determining task difficulty levels, a potential solution is to utilize key project attributes and apply the K-means Partitioning clustering algorithm to form the difficulty sets effectively[12]. Now, all the projects are clustered to form new attributes, and the results of this clustering are divided into four new project categories: hard, medium, low, and easy. Then, according to the expert judgment, for each of these qualitative values, the equivalent of 3, 2, 1.5, and 1 person-hour per day is considered, respectively. Now, we can add the difficulty level of each project to the PM's projects, i.e., the second row of Table IV. To deal with the second problem, we set the ability numbers between 1 and 6, which are assigned to a PMO by using the AHP method based on criteria like work experience. Then, the resulting ability numbers are converted into person-hours per day for equalization. For example, the most capable person is available for a total of 6 hours a day. Table V shows a candidate PMO shortlist and a limited number list with information on each candidate's capacity to prevent overload. The two obstacles he still has to overcome can be overcome with the aid of E-CRAGO and GRA.

## III. PROBLEM FORMALIZATIONS WITH E-CARGO MODEL

Afshin's problem is a GTAP, which is a variant of the GMRACR. Hence, we use the E-CARGO to specify the GTAP at first. The E-CARGO uses a 9-tuple $\sum$::=< $\mathcal{E}$, $\mathcal{C}$, $\mathcal{O}$, $\mathcal{A}$, $\mathcal{M}$, $\mathcal{R}$, $\mathcal{G}$, $\mathcal{H}$, $s_0$>, as finite sets of environments, classes, objects, agents, messages, roles, groups, human users and the initial state of the system. Environment $e$ and group $g$ are defined with matrices and vectors in discussions of group role assignment [10], [13]. This study employs non-negative integers $m$ (=$|\mathcal{A}|$) and $n$ (= $|\mathcal{R}|$) to represent the size of set $\mathcal{A}$ and $\mathcal{R}$, respectively, $i, i_1, \ldots \in \{0, 1, \ldots, m-1\}$ the indices of agents, and $j, j_1, \ldots \in \{0, 1, \ldots, n-1\}$ the indices of roles. In particular, the grouped projects (grouped tasks) of each PM can be viewed as roles, and the PMOs (staff members) as agents, regarding the allocation of the PMOs (GTAP).

*Definition 1*[10]**:** A *role range vector L* is the lower range vector of role $i$ in environment $e$ of group $g$ to indicate the least number of agents required for a single role, $L[j] \in \mathcal{N}$. In our scenario, the PM's difficulty level is determined by collecting the difficulty levels of all their projects and converting it into a role range vector $L$ which denotes the number of person-hours per day required by each role (grouped projects of each PM) at least. That is, $L$= [7,6,3,9.5,3,2,3,4,1.5,9,7.5].

*Definition 2:* A weight parameter $\alpha$ is a non-negative decimal, where $\alpha \in [0,1]$. In the same way, a weight parameter $\beta$ is a non-negative decimal, where $\beta + \alpha = 1$. We consider $\alpha = 0.4$ in the scenario presented in section II. This setting for a multi-objective parameter is typical [14]. The following section will explain how changing the values of $\alpha$ and $\beta$ can produce an array of equilibrium solutions that satisfy both goals.

*Definition 3***:** $T$ is an expression of person-hours-per-day assignment in a non-negative integer matrix. $T\,[i,j] \geq 0$ indicates how many person-hours per day of agent $i$ are assigned to role $j$ whereas $T\,[i,j] = 0$ means that no person-hour of agent $i$ is assigned to role $j$.

*Definition 4*[10]**:** A *qualification matrix* $Q$ is an $m \times n$ matrix to indicate the competent quality that an agent performs on a role, i.e., $Q\,[i,j] \in [0,1]$ express agent $i$'s qualification value for role $j$, and 0 means the lowest value whereas 1 the highest value. It should be noted that a $Q$ matrix can be generated by comparing all agent qualifications to all role specifications. The $Q_{n,initial}$ matrix in the scenario above discussed in Section II reflects the scoring matrix obtained following the normalization process in [0,1] of Table I ($Q_{initial}[i,j]$).

$$Q_{n,initial}[i,j] = \frac{Q_{initial}[i,j] - min\{Q_{initial}\}}{max\{Q_{initial}\} - min\{Q_{initial}\}} \quad (1)$$

*Definition 5*[7]**:** a qualification matrix $Q_T$ is an $m \times n$ matrix. Since maximizing the matching degrees and centralizing the agents are two objective functions with different dimensions, they need to be normalized before weighting. Hence, we define $Q_T$ as follows.

$$Q_T[i,j] = \frac{Q_{n,initial}[i,j]}{L^a[i]} \quad (2)$$

*Definition 6*[16]**:** A binary matrix $Y\,[i, j]$ represents the role assignment, $Y\,[i, j] = 1$ denotes that agent $i$ is assigned to role $j$ whereas $Y\,[i, j] = 0$ means not. In definition 3, if $T\,[i,j] > 0, Y\,[i,j] = 1$; otherwise $Y\,[i,j] = 0$

*Definition 7*[7]**:** The *group performance* $\sigma$ is the sum of all the designated agents' qualifications, i.e., $\sigma = \sum_{j=0}^{n-1}\sum_{i=0}^{m-1}(\alpha \times Q_T[i,j] \times T[i,j] - \beta \times Y[i,j])$.

*Definition 8*[10]**:** Role $j$ *workable* if assigned to sufficient person-hours per day of agents, i.e., $\sum_{i=0}^{m-1} T[i,j] = L[j]$.

*Definition 9*[10]**:** $T$ is *workable* if each role $j$ is workable, i.e., $\sum_{i=0}^{m-1} T[i,j] = L[j] (0 < j < n)$, so does Group $g$.

*Definition 10*[15]**:** An $n \times n$ matrix known as $R^c$ is used to describe potential conflicts between two roles, $j_1$ and $j_2$ ($j_1, j_2 \in \mathcal{R}$, $j_1 \neq j_2$). When $R^c\,[j_1, j_2] = 1$ $j_1$ and $j_2$ are in conflict, however when $R^c\,[j_1, j_2] = 0$ they are not.

*Definition 11*[8]**:** An *ability limit vector* $L^a$ is an $m$-vector, where $L^a[i]$ ($0 \leq i < m$) represents the maximum person-hour per day required for roles that can be allocated to an agent. For the scenario in Section II, $L^a$ = [5,6,4,6,6,5,4,5,6,6,5,6,6,4,6,6,5,4].

TABLE IV. PROJECTS DIFFICULTY LEVEL

| PM | Nader | Mahsa | Kia | Sara | Babak | Mina | Reza | Zahra | Dana | Arash | Donya |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Difficulty Level | 7 | 6 | 3 | 9.5 | 3 | 2 | 3 | 4 | 1.5 | 9 | 7.5 |

TABLE V. PMOS' ABILITY NUMBER

| PMO | Ali | Sam | Rose | Mia | Zoe | Eli | Eva | Ian | Ivy | Kai | Tim | Nora | Lili | Li | Mona | Nina | Hadi | Mari |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ability Number | 5 | 6 | 4 | 6 | 6 | 5 | 4 | 5 | 6 | 6 | 5 | 6 | 6 | 4 | 6 | 6 | 5 | 4 |

*Definition 12:* $M$ is a big number to indicate the maximum number of $T[i,j]$ $(0 \le i < m, 0 \le j < n)$.

*Definition 13:* Suppose that we have $\mathcal{A}$, $\mathcal{R}$, $Q$, $L$, $L^a$, and $R^c$, GTAP intends to search for a $T$ and $Y$ to

$$\sigma = \text{Max} \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (\alpha \times Q_T[i,j] \times T[i,j] - \beta \times Y[i,j]) \quad (3)$$

subject to:

$$T[i,j] \ge 0 \quad (0 \le i < m, 0 \le j < n), \quad (4)$$

$$\sum_{i=0}^{m-1} T[i,j] = L[j] \quad (0 \le j < n), \quad (5)$$

$$\sum_{j=0}^{n-1} T[i,j] \le L^a[i] \quad (0 \le i < m), \quad (6)$$

$$R^c[j_1, j_2] \times T[i, j_1] \times T[i, j_2] = 0 \quad \left(0 \le i < m, 0 \le j_1, j_2 < n, j_1 \ne j_2\right), \quad (7)$$

$$T[i,j] > -M(1 - Y[i,j]) \quad (0 \le i < m, 0 \le j < n), \quad (8)$$

$$T[i,j] \le M \times Y[i,j] \quad (0 \le i < m, 0 \le j < n), \quad (9)$$

Where (3) is the group performance (4) being a sign constraint; (5) guarantees any role will be workable; (6) means that agents are not assigned roles more than their abilities; (7) expresses that the roles assigned to the same agent do not conflict; (8) and (9) indicates that that variable $T$ and $Y$ are related. Because of (7), the GTAP is a nonlinear programming problem. But we can rewrite (7)-(9) as follows.

$$R^c[j_1, j_2] \times (Y[i, j_1] + Y[i, j_2]) \le 1 \quad \left(0 \le i < m, 0 \le j_1, j_2 < n, j_1 \ne j_2\right), \quad (10)$$

$$T[i, j_1] > -M(1 - Y[i, j_1]) \quad (0 \le i < m, 0 \le j_1 < n), \quad (11)$$

$$T[i, j_1] \le M \times Y[i, j_1] \quad \left(0 \le i < m, 0 \le j_1 < n\right), \quad (12)$$

$$T[i, j_2] > -M(1 - Y[i, j_2]) \quad \left(0 \le i < m, 0 \le j_2 < n\right), \quad (13)$$

$$T[i, j_2] \le M \times Y[i, j_2] \quad (0 \le i < m, 0 \le j_2 < n) \quad (14)$$

$$Y[i,j] \in \{0,1\} \quad (0 \le i < m, 0 \le j < n), \quad (15)$$

Therefore, the new expression of GTAP is to:

$$\text{Max} \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (\alpha \times Q_T[i,j] \times T[i,j] - \beta \times Y[i,j])$$

Concerning (4)–(6) and (10)–(15), which the IBM ILOG CPLEX optimization package can handle. We use $T^*$ and $Y^*$ to express the $T$ and $Y$ that makes $\sigma$.

*Theorem*: A necessary condition for GTAP is that $\sum_{i=0}^{m-1} L^a[i] \ge \sum_{j=0}^{n-1} L[j]$.

*Proof*: See the proof of Theorem 1 of[8].

## IV. SOLUTIONS AND EXPERIMENTS

We carry out a randomized simulation experiment to evaluate the efficiency and dependability of this method. In traditional solutions, allocation is done without considering the ability of people and also the difficulty of the work, which has caused a decrease in the potential performance of the system as well as an increase in people's dissatisfaction due to the imbalance in the assigned tasks. Therefore, we use the partitioning clustering algorithm to find the difficulty level and the AHP method to assign ability numbers to the PMOs. Hence, each group of tasks and roles has its own value to have a balanced assignment. One of the most important things that should be addressed is determining the appropriate value of $\alpha$ and $\beta$. Hence, we can have the highest value in performance and centrality. We simulate 4 experiments with various scale levels in order to get the best value for them. Projects, PMs, PMOs, and specific project types are present in varying numbers at each scale level. The range of projects will increase from 31 to 130, the number of PMs allocated to the projects will increase from 11 to 35, and the number of PMOs will increase from 18 to 45 when the scale of the experiment is increased from 1 to 4. In Table VI, the specific details are displayed. Next, the solution to the GTAP is found using the IBM ILOG CPLEX Optimization Package built on the platform shown in Table VII. We consider 10 different values with a step of 0.1 for $\alpha$. As you can see in Fig. 2 and 3, although the rate of change of both parts is very slow between the values of 0.1 and 0.9, but for $\alpha = 0.4$, the rate of unit performance increases without increasing the degree of decentralization. The increase continues until $\alpha = 0.7$, and we see an increase in decentralization, i.e., the values between 0.4 and 0.7 can be the best values for $\alpha$. But in Fig. 4, we see that the solution time from $\alpha = 0.4$ to 0.7 is rapidly increasing, thus it is better to choose $\alpha$ near 0.4, given the time required for a larger scale.

TABLE VI. DATA INFORMATION UNDER DIFFERENT SCALES

| | Scale 1 | Scale 2 | Scale 3 | Scale 4 |
|---|---|---|---|---|
| Number of Projects | 31 | 70 | 100 | 130 |
| Number of PMs | 11 | 15 | 25 | 35 |
| Number of PMOs | 18 | 30 | 40 | 45 |
| Number of Types of Projects | 6 | 10 | 10 | 12 |

TABLE VII. THE CONFIGURATION OF THE EXPERIMENTAL PLATFORM

| | |
|---|---|
| CPU | Intel(R) Core (TM) i7-3537U CPU @ 2.00GHz 2.50 GHz |
| Memory | 10.0 GB |
| OS | Windows 10 Pro |
| Platform | GAMS 24.1.2 |

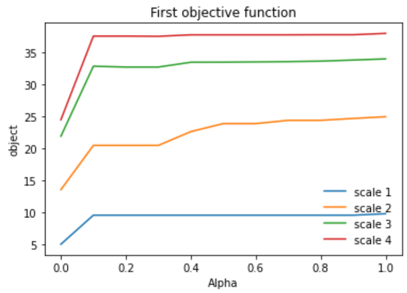


Fig. 2. Performance curve in different scales.

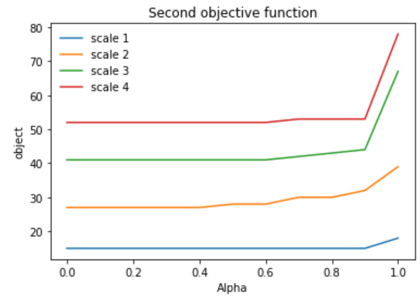


Fig. 3. Centrality curve in different scales.

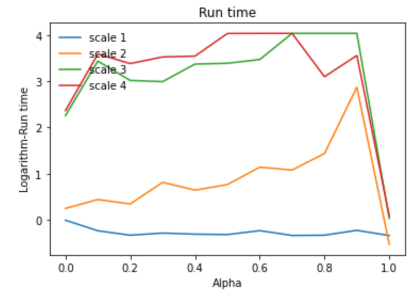


Fig. 4. Time cost of each scale based on Alpha.

To verify the benefits of GTAP compared with GMRACR, as a traditional way, if we compare the group performance of the GTAP with the basic GMRACR, then we should see an increase in performance.

*Definition 14*[15]**:** Suppose we have $Q_{n,initial}$, $L$=1, $L^a$ = $\left\lfloor \frac{n}{m} \right\rfloor + 1$, and $R^c$, GMRACR intends to search for a $T$ to get:

$$\sigma' = \text{Max} \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} ( Q_{n,initial}[i,j] \times T[i,j] \tag{16}$$

subject to:

$$T[i,j] \epsilon \{0,1\} \quad (0 \le i < m, 0 \le j < n), \tag{17}$$

$$\sum_{i=0}^{m-1} T[i,j] = L[j] \quad (0 \le j < n), \tag{18}$$

$$\sum_{j=0}^{n-1} T[i,j] \le L^a[i] \quad (0 \le i < m), \tag{19}$$

$$R^c[j_1, j_2] \times (T[i,j_1] + T[i,j_2]) = 0 \\ (0 \le i < m, 0 \le j_1, j_2 < n, j_1 \ne j_2) \tag{20}$$

Where (17) tells that an agent can be assigned or not; (18) makes the group workable; (19) indicates the allowed number of roles taken by each agent; and (20) means that the roles assigned to the same agent do not conflict. We use $T'$ to express the $T$ that makes $\sigma'$. Further simulations are run to verify this assumption, and data from each experiment is gathered as follows:

1. $\sigma_1$ is the actual group performance with GTAP. $\sigma_1 = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} Q_{n,initial}[i,j] \times Y^*[i,j]$.
2. $\sigma_2$ is the maximum group performance with GMRACR. $\sigma_2 = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} Q_{n,initial}[i,j] \times T'[i,j]$.
3. The benefit of GTAP over GMRACR, $\lambda$ is calculated as $\frac{\sigma_1 - \sigma_2}{\sigma_2}$.

TABLE VIII. SIMULATION AVERAGE PERFORMANCE

| | $m$ | $n$ | $\sigma_1$ | $\sigma_2$ | $\lambda$ |
|---|---|---|---|---|---|
| Scale 1 | 18 | 11 | 13.660 | 9.936 | 37% |
| Scale 2 | 30 | 15 | 23.463 | 13.910 | 69% |
| Scale 3 | 40 | 25 | 34.625 | 22.560 | 53% |
| Scale 4 | 45 | 35 | 47.941 | 31.580 | 45% |

TABLE IX. SIMULATION RUN TIME (SEC.)

| | **Run Time $\sigma_1$** | | | **Run Time $\sigma_2$** | | |
|---|---|---|---|---|---|---|
| | ***Min*** | ***Avg*** | ***Max*** | ***Min*** | ***Avg*** | ***Max*** |
| Scale 1 | 0.029 | 0.458 | 0.967 | 0.140 | 0.158 | 0.181 |
| Scale 2 | 4.276 | 143.747 | 470.768 | 0.195 | 0.225 | 0.467 |
| Scale 3 | 602.29 | 725.01 | 1305.342 | 0.301 | 0.417 | 1.846 |
| Scale 4 | 1249.88 | 2326.58 | 3919.019 | 2.472 | 4.704 | 10.624 |

With a small, average, and large group of agents and roles, we examine the advantage of the GTAP in Tables VIII and IX. The simulations were then carried out 100 times for each scale, and the average of the results are shown in Tables VIII and IX. Although benefits differ depending on group size, they are favorable in all cases and ranged from 37% to 69%. As you can see in Table IX, the run time for large scale in the GTAP increases exponentially, which indicates the NP-hardness of the model. Although the run time of larger-scale problems has increased significantly, it is evident that both small and large groups can perform much better when using the GTAP. Also, since in the traditional method, the difficulty of work and the ability of people are not considered, many of the answers obtained by the traditional model are infeasible in reality. This means that although the traditional model reaches the answer in less time, but that answer will not be effective in real conditions because the assignment does without considering the difficulty of the tasks and the ability of the people.

## V. RELATED WORK

The difficulty in assigning tasks within a group lies in the need to allocate roles to team members while recognizing that each task may require a different person to perform it, and that one person may be responsible for multiple roles. Therefore, many researchers are conducting relevant studies in dealing with M2M assignment problems and have made good progress in more effectively solving task assignments between team members. Dadvar et al. [17] presented a hunter-and-gatherer system of task allocation based on coupling and collaboration between complementing teams. Xiao et al. [18] investigate a novel form of spatial crowdsourcing called competitive detour tasking, in which workers have the ability to undertake multiple tasks and can strategically compete for their preferred tasks by indicating their diversion costs. The goal of their study is to identify an optimal assignment solution that maximizes overall social welfare while safeguarding the privacy of the workers' personal information. Most previous research on task assignment ignores the issue of dependency across jobs, which leads to some bad matching pairs and wastes workers' time. Zhao et al.[19] proposed a new SC problem that aims to maximize overall profit by assigning employees to complex jobs across multiple stages. Ni et al. [20] presented a model that simplifies the assignment of workers to difficult tasks within skill limits by breaking down the complex assignment into manageable subtask assignment problems. Wu et al. [21] investigated fair task distribution in supply chains. They introduced a well-designed constraint based on fair budget allocation, aiming to maximize task quality while ensuring group fairness and considering additional limitations. Zhu et al. [22] formalized staff and task assignments on the basis of the E-CARGO model and then proposed an algorithm to solve it. First time, the E-CARGO model presented by Zhu et al. [23], [24] is used to define the relationship between roles and agents. They showed that task allocation is one of the most challenging elements of collaboration. They developed the engineering methodology for RBC to focus on accurate task assignments [7], [10]. Liu et al.'s [25] formalization of the tree-structured task allocation problem (TSTAP) with GMRA with the goal of solving it provides practical and efficient decision support for dealing with the challenging TSTA issues. By adapting the process to a GRA and providing a novel method based on the Kuhn-Munkres algorithm. GMRA was developed by Zhu et al. [8] to formalize an important engineering problem. GRA is one of RBC's fundamental components. The task assignment procedure in GRA seeks to provide a task with the best team performance based on agent ratings. Assignment issues with diverse limitations that are adapted to specific applications are the subject of many studies. Assignment difficulties are complicated by the several constraints that roles and agents face, such as mutual exclusion, quantity, time, conflicts, and others. Based on them, the main technical techniques in this article are E-CARGO and its sub-model GRA, with RBC serving as the underlying theory and methodology.

## VI. Conclusion

This study reviews and extends GMRACR, a new category of assignment problem, and formalizes the grouped task allocation problem GTAP as an M2M assignment. We first describe the GTAP as a nonlinear problem, and then map it into a linear programming problem. After that, we verify the benefits of considering the ability of employees and also the difficulty level of the tasks in GMRACR through simulations. We apply the partitioning clustering algorithm to find the difficulty level and the AHP method to assign ability numbers to employees. Furthermore, the determination of weight for objectives is discussed. From simulations, we obtain: 1) Unlike GMRACR, GTAP generates a new assignment that improves the actual group performance. 2) For larger groups, the benefit of GTAP is typically close to 45%. 3) Considering the run time and the balance between maximizing performance and people centrality, the weighting factors of 0.4 and 0.6 are the best values for the objectives. In summary, the proposed formalizations are valuable. Further investigations may be conducted along the following directions: 1) Considering the dynamic GTAP according to the addition or subtraction of projects. 2) After the first role assignment, techniques for role reassignment and conflict resolution should be created and incorporated into the E-CARGO. 3) Composing better methods to form an $L$ with difficulty level instead of expert judgment. 4) Proposing solution algorithms to reduce the run time in large-scale problems.

## Reference


[1] J. Munkres, “Algorithms for the Assignment and Transportation Problems,” J. Soc. Ind. Appl. Math., vol. 5, no. 1, pp. 32–38, Mar. 1957.

[2] S. Rowlinson, “Matrix organizational structure, culture and commitment: a Hong Kong public sector case study of change,” Constr. Manag. Econ., vol. 19, no. 7, pp. 669–673, Nov. 2001.

[3] G. A. Korsah, A. Stentz, and M. B. Dias, “A comprehensive taxonomy for multi-robot task allocation,” Int. J. Robot. Res., vol. 32, no. 12, pp. 1495–1512, Oct. 2013.

[4] D.-H. Lee, “Resource-based task allocation for multi-robot systems,” Robot. Auton. Syst., vol. 103, pp. 151–161, May 2018.

[5] J. Schwarzrock, I. Zacarias, A. L. C. Bazzan, R. Q. de Araujo Fernandes, L. H. Moreira, and E. P. de Freitas, “Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence,” Eng. Appl. Artif. Intell., vol. 72, pp. 10–20, Jun. 2018.

[6] H. Zhu, “Group Role Assignment With Constraints (GRA+): A New Category of Assignment Problems,” IEEE Trans. Syst. Man Cybern. Syst., vol. 53, no. 3, pp. 1670–1683, Mar. 2023.

[7] H. Zhu and M. Zhou, “Role-based collaboration and its kernel mechanisms,” IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 36, no. 4, pp. 578–589, Jul. 2006, doi: 10.1109/TSMCC.2006.875726.

[8] H. Zhu, D. Liu, S. Zhang, S. Teng, and Y. Zhu, “Solving the Group Multirole Assignment Problem by Improving the ILOG Approach,” IEEE Trans. Syst. Man Cybern. Syst., vol. 47, no. 12, pp. 3418–3424, Dec. 2017.

[9] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng (2016). Solving the many to many assignment problem by improving the Kuhn–Munkres algorithm with backtracking. Theoretical Computer Science, 618, 30-41.

[10] H. Zhu, M. Zhou, and R. Alkins, “Group Role Assignment via a Kuhn–Munkres Algorithm-Based Solution,” IEEE Trans. Syst. Man Cybern. - Part Syst. Hum., vol. 42, no. 3, pp. 739–750, May 2012.

[11] “ILOG CPLEX Optimization Studio - Overview,” Oct. 17, 2022. https://www.ibm.com/products/ilog-cplex-optimization-studio (accessed Mar. 14, 2023).

[12] J. Yadav and M. Sharma, “A Review of K-mean Algorithm,” Int. J. Eng. Trends Technol., vol. 4, no. 7, 2013.

[13] H. Zhu and R. Alkins, “Group role assignment,” in 2009 International Symposium on Collaborative Technologies and Systems, May 2009, pp. 431–439.

[14] A. Mohammadi, M. N. Omidvar, and X. Li, “A new performance metric for user-preference based multi-objective evolutionary algorithms,” in 2013 IEEE Congress on Evolutionary Computation, Jun. 2013, pp. 2825–2832.

[15] H. Zhu, “Group multi-role assignment with conflicting roles and agents,” IEEECAA J. Autom. Sin., vol. 7, no. 6, pp. 1498–1510, Nov. 2020.

[16] J. Qi, H. Zhu, J. Xiang, and D. Liu, “Revising Portfolio with Preferences via Higher-Order Group Multi-Role Assignment,” in 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), May 2021, pp. 540–545.

[17] M. Dadvar, S. Moazami, H. R. Myler, and H. Zargarzadeh, “Multiagent Task Allocation in Complementary Teams: A Hunter-and-Gatherer Approach,” Complexity, vol. 2020, pp. 1–15, Mar. 2020.

[18] M. Xiao et al., “SRA: Secure Reverse Auction for Task Assignment in Spatial Crowdsourcing,” IEEE Trans. Knowl. Data Eng., vol. 32, no. 4, pp. 782–796, Apr. 2020.

[19] Z. Liu, K. Li, X. Zhou, N. Zhu, Y. Gao, and K. Li, “Multi-stage complex task assignment in spatial crowdsourcing,” Inf. Sci., vol. 586, pp. 119–139, Mar. 2022.

[20] W. Ni, P. Cheng, L. Chen, and X. Lin, “Task Allocation in Dependency-aware Spatial Crowdsourcing,” in 2020 IEEE 36th International Conference on Data Engineering (ICDE), Apr. 2020.

[21] B. Wu, K. Han, and E. Zhang, “On the task assignment with group fairness for spatial crowdsourcing,” Inf. Process. Manag., vol. 60, no. 2, p. 103175, Mar. 2023.

[22] X. Zhu, Y. Wang, W. Hu, X. Zhou, J. Yang, and H. Zhu, “Multi-task Assignment in Information Technology Companies Based on the E-CARGO Model,” in 2015 IEEE International Conference on Systems, Man, and Cybernetics, Oct. 2015, pp. 310–315.

[23] H. Zhu, “Fundamental issues in the design of a role engine,” in 2008 International Symposium on Collaborative Technologies and Systems, pp. 399–407, May 2008.

[24] H. Zhu, E-CARGO and Role-Based Collaboration: Modeling and Solving Problems in the Complex World. John Wiley & Sons, 2021.

[25] D. Liu, B. Huang, and H. Zhu, “Solving the Tree-Structured Task Allocation Problem via Group Multirole Assignment,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 1, pp. 41–55, Jan. 2020.