# Lab 2: Group Multi-Role Assignment

**Project 2**: Python-Pulp: Task Assignment using GMRA.

1. **Preliminary Knowledge**
   1. **Group Multi-Role Assignment (GMRA)**

Group Multi-Role Assignment (GMRA) is a typical submodel of the RBC methodology, and it is derived from the Group Role Assignment(GRA) submodel. Compared to the GRA model, the GMRA model has broader applicability. In the **GRA model**, we assume that **an agent can only perform one role in the assignment process**. The GMRA model, on the other hand, assumes that **an agent can perform one or more roles at a time**. Here, we use the vector *La* to represent the number of roles an agent can perform (see Constraint (3) in Fig. 1). The mathematical expression of the GMRA model is shown below. It is an efficient tool to solve M-M (many-to-many) assignment problems.

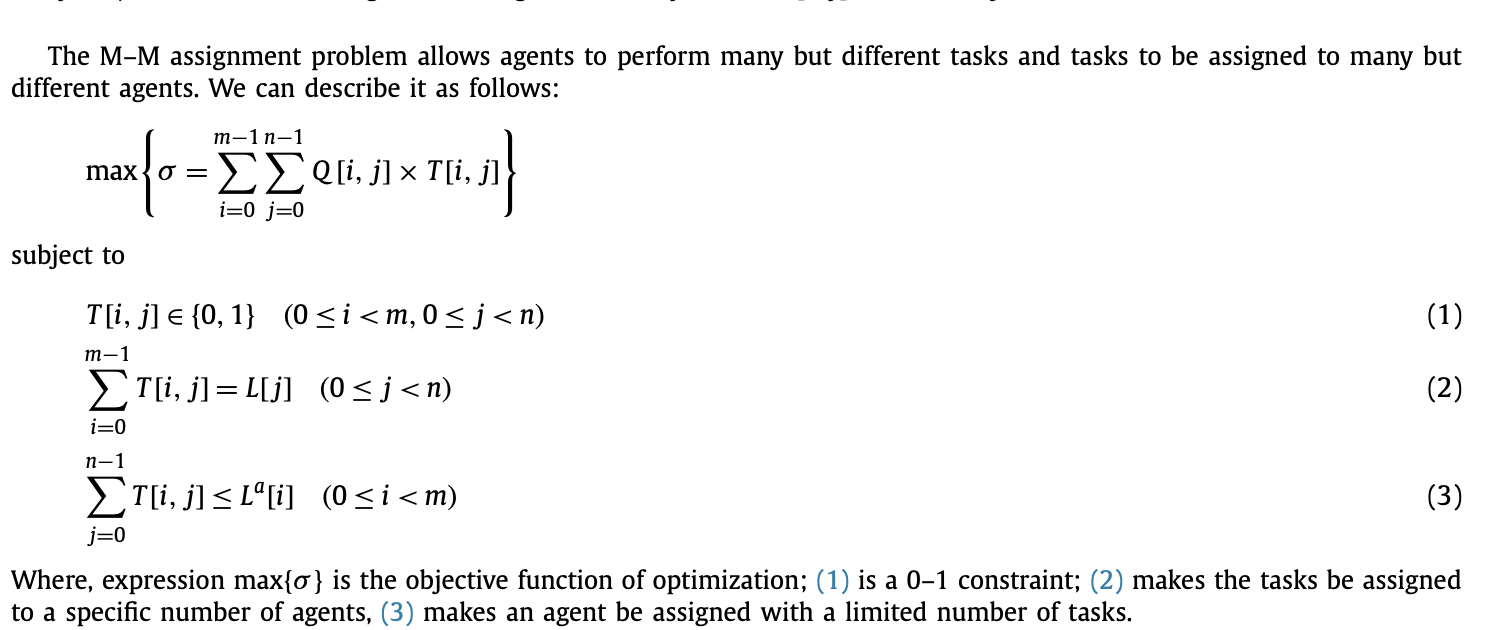


Fig. 1. Mathematical expression of the GMRA model.

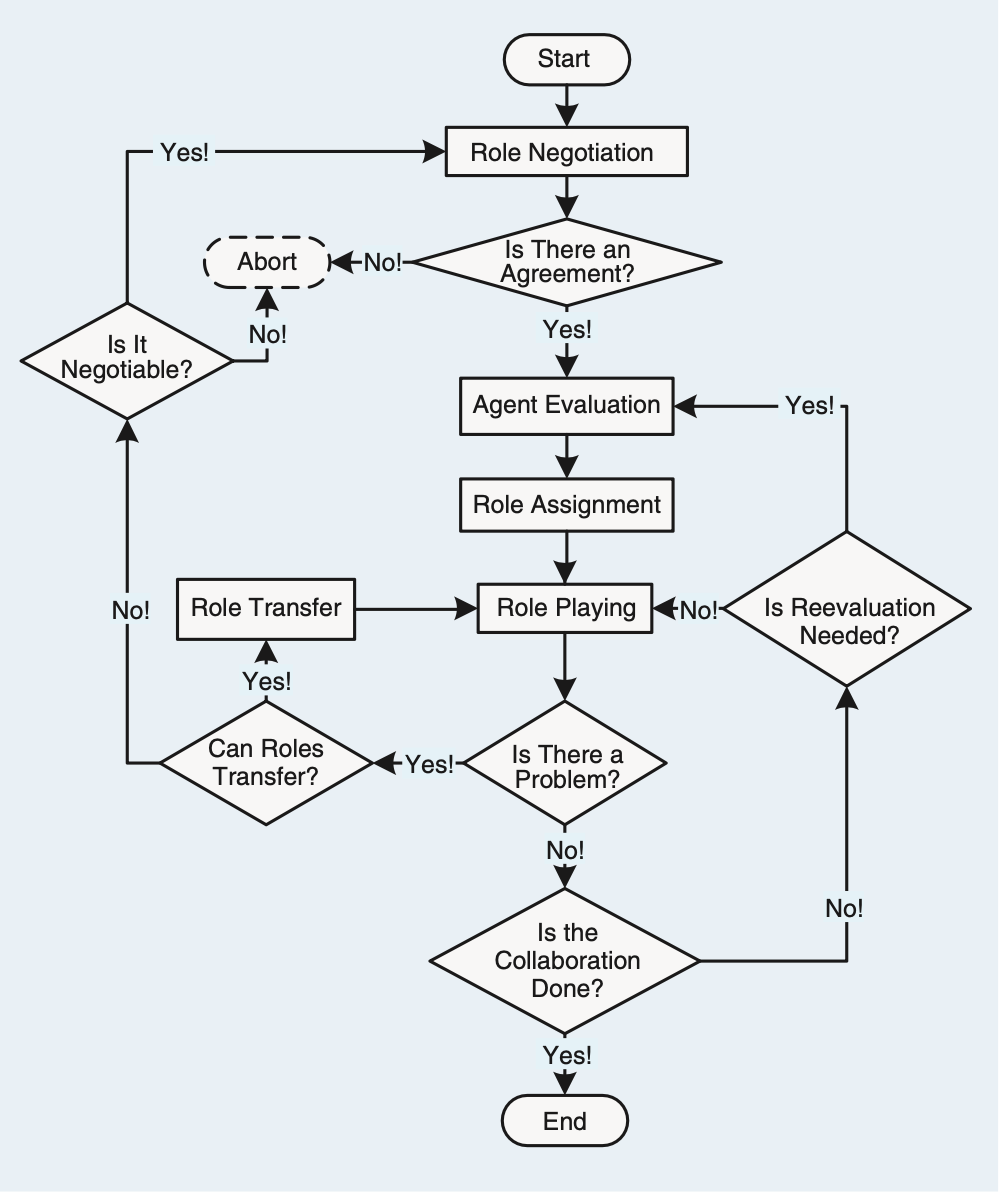
Learn for detail: <https://www.sciencedirect.com/science/article/pii/S0304397516000037>

### Ojbectives

The objectives of this course are two-fold:

1. To understand the process of RBC (see Fig. 2) and learn about how to formalize problems with the GMRA submodel;

2. To practice using Python to program the GMRA model in order to solve M-M assignment problems.



**Main Focus**

Fig. 2. The process of Role-Based Collaboration (RBC).

### Assignment for this lesson

### The assignment requirements for this lesson are as follows:

1. Suppose you are a manager of a company. Imagine and describe a scenario, i.e., to accomplish a complex task (RBC) by managing 30 agents (people, equipment, robots, groups of people, etc.). In addition to the scenario required for the first lesson assignment, **add a constraint that each agent can perform one or more roles (tasks)**.

2. You need to divide the complex into smaller subtasks (Roles), i.e., role negotiation in Fig 2. **Be creative, and any simple method can be used.**

3. You can choose from a list of candidates (Agents) to join the team to accomplish the task. **Be creative, and any simple method can be used.**

4. You need to determine a list of requirements for each task (role), i.e., role specification including *L* and other required properties. **Be creative and reasonable. Any simple method can be used.**

5. Suppose that every agent should have a list of qualifications corresponding to the roles’ requirements. You create the evaluation (i.e., the agent evaluation part in Fig. 2) of each agent for each subtask (role), i.e., the *Q* matrix. **Be creative, and any simple method can be used.**

6. After you obtain the *Q* matrix, use the GMRA program to get the optimal assignment result, i.e., *T*.

7. Analyze whether the assignment is good or not from your own personal perspective. Argue why an optimized assignment result may not be the best choice.

8. Consider whether there are **scalable aspects** in this scenario (i.e., future works), as this is relevant to future lessons.

9. Encode and calculate assignment results using Python’s PuLP, and present the mathematical model and corresponding assignment results in the format of an IEEE paper.

10. If possible, please choose to **expand the self-defined scenario from the first lesson**, to make the background of the problem you are researching more generic.

### Turn in Requirement

1. A project report including the descriptions of your process details.

2. The report should be in the IEEE conference paper format, page limit = 4 pages. Refer to: <https://www.ieee.org/conferences/publishing/templates.html>, Choose Microsoft Word and US letter.

3. (**\*Vital)** **The submitted paper should provide the source code and simulation data. The paper should also depict plausible scenarios and provide a rational explanation for the data.**

**Download the Relevant Materials:**

**https://github.com/jiangqian1997/E-CARGO-Codes/tree/main/Summber\_School\_Laboratory/Lab\_2**