# Lab 3: Group Role Assignment with Conflicting Agents on Roles

**Project 3**: Python-Pulp: Task Assignment using GRACAR.

1. **Preliminary Knowledge**
   1. **Group Role Assignment with Conflict Agents on Roles (GRACAR)**

Group Role Assignment with Confict Agents on Roles (GRACAR) is derived from the Group Role Assignment (GRA) submodel. Compared to the GRA model, the GRACAR model extends the applicable scenarios compared with the GRA model. The GRA model assumes that **there is no conflict between agents when performing the same role**. On the other hand, the GRACAR model **takes into account the conflicted relationship between agents** and **introduces the agent conflict matrix** *Ac* to formalize the conflicted relationship between agents. Fig. 1 illustrates the relationship between the E-CARGO-related models mentioned so far. The mathematical expression of the GRACAR model is shown in Fig. 2. It is an efficient tool to solve 1-M (one-to-many) assignment problems with existing conflicted relationships between task executors when performing tasks.

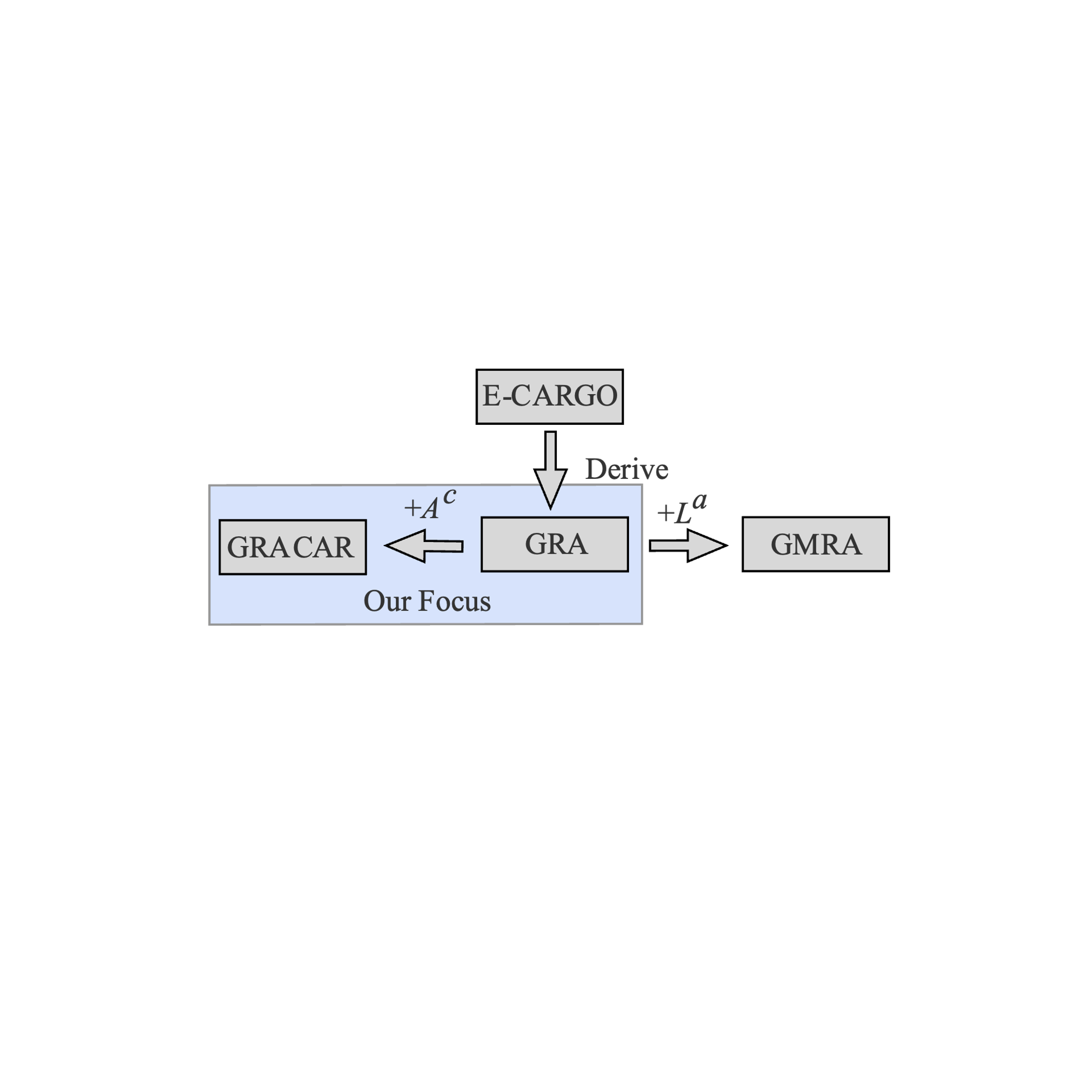


Fig. 1. The relationship between the E-CARGO-related submodels currently mentioned.

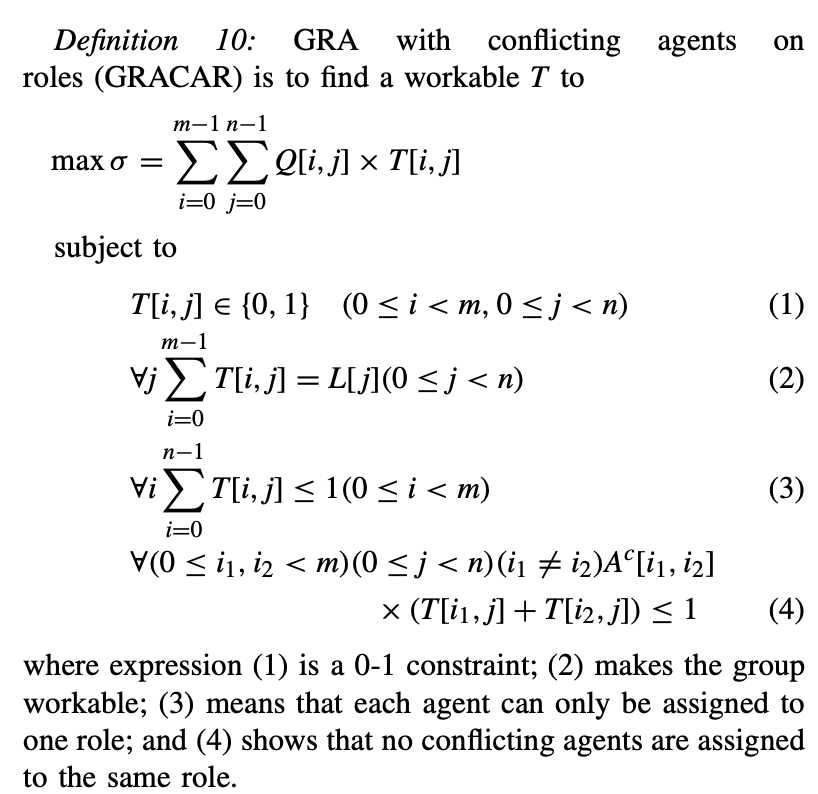


Fig. 2. Mathematical expression of the GRACAR model.

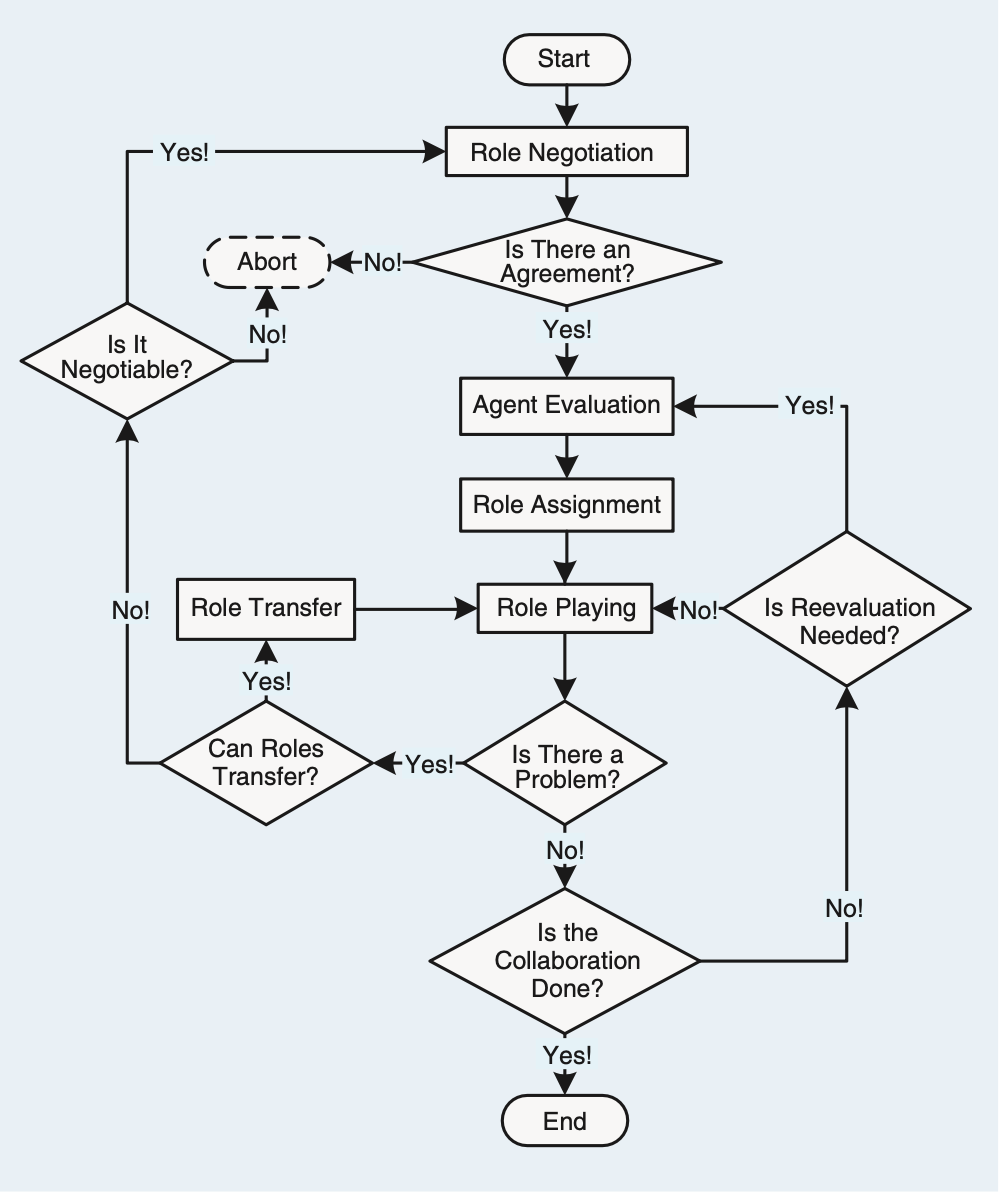
Learn for detail: <https://ieeexplore.ieee.org/abstract/document/7122926>

### Ojbectives

The objectives of this course are two-fold:

1. To understand the process of RBC (see Fig 3) and learn about how to formalize problems with the GRACAR submodel;

2. To practice using Python to program the GRACAR model.



**Main Focus**

Fig. 3. 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 agents may have conflicting relationships while performing the same role. Please utilize the *AC* matrix to represent the conflicted relationship when agents perform roles.**

2. You need to divide the complex into smaller subtasks (Roles), i.e., role negotiation in Fig 3. **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. 3) 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 GRA program (Group Role Assignment) 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

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.