# Lab 4: Group Role Assignment with Cooperation and Conflict Factors

**Project 4**: Python-Pulp: Task Assignment using GRACCF.

1. **Preliminary Knowledge**
   1. **Group Role Assignment with Cooperation and Conflict Factors (GRACCF)**

Group Role Assignment with Cooperation and Conflict Factors (GRACCF) is derived from the Group Role Assignment (GRA) and Group Role Assignment, the GRACCF model also extends the applicable scenarios compared with the GRA model. **Unlike the GRACAR model**, which **only considers conflicts between agents when performing the same role**, the GRACCF model is applicable when **there are differences between roles**, and it **refines the granularity of conflict and collaboration relationships**. For example, suppose there are employees *a* and *b* from two companies (i.e., agents), and there is no inherent conflict between employees *a* and *b*. At the same time, the company needs to temporarily dispatch employees a and b to develop a software project. Suppose this project requires them to choose tasks to execute, such as: algorithm development, UI design, front-end development, and back-end development. At this point, based on the past experience, there may be a **conflict (i.e., conflict relationship)** when employee *a* performs front-end development and employee *b* acts as a back-end developer. But when employee *a* performs front-end development and employee *b* acts as an algorithm developer, **they may have a cooperative relationship.** **The conflict degree between them can be represented from -1 to 0, and the degree of their cooperation can be represented from 0 to 1.**

Fig. 1 illustrates the relationship between the E-CARGO-related models mentioned so far. The mathematical expression of the GRACCF model is shown in Fig. 2. The GRACCF model is applicable to assignment problems where **there are significant differences between roles** and **a need to refine the cooperation and conflict relationships that exist when agents perform roles.**

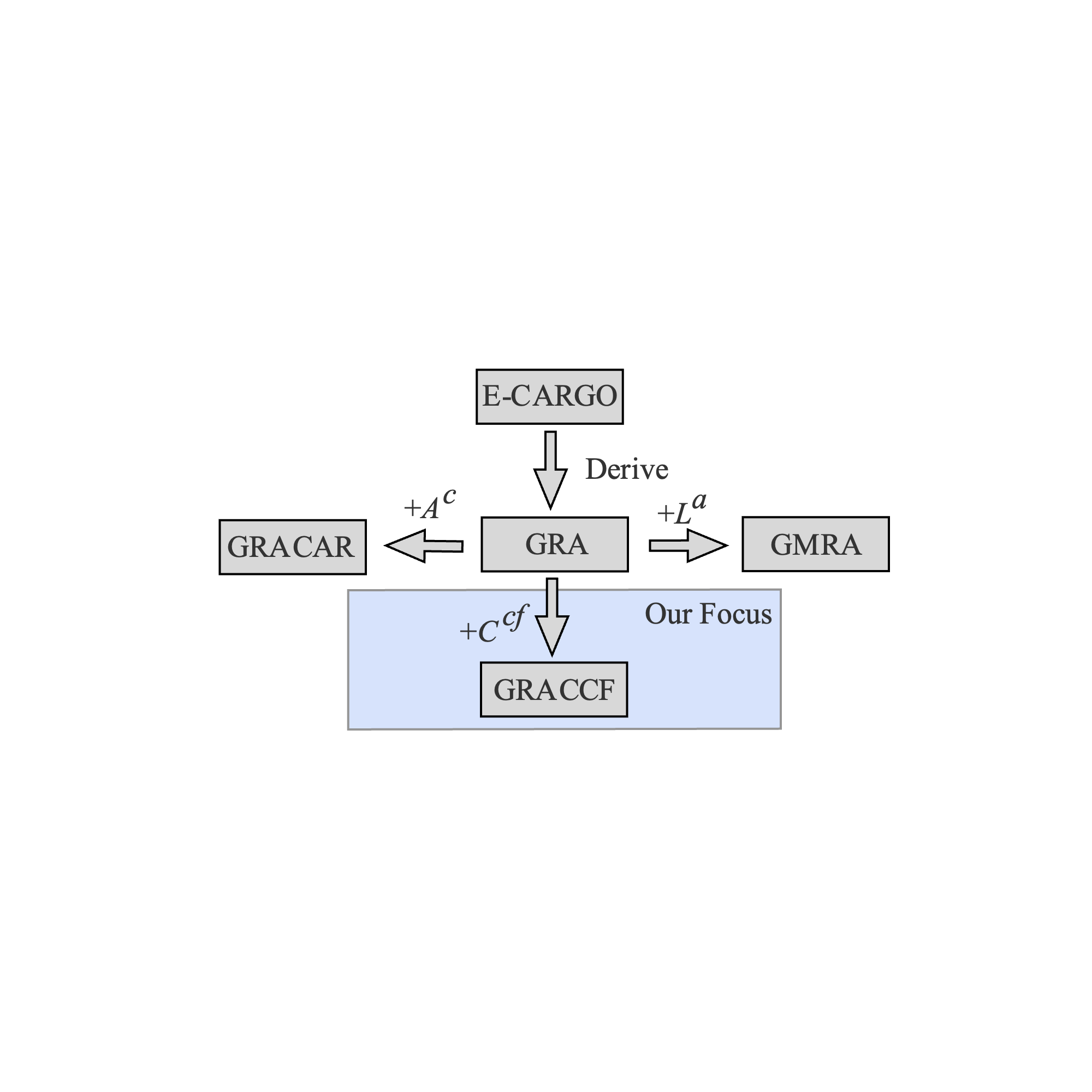


Fig. 1. The relationship between the E-CARGO-related submodels currently mentioned.

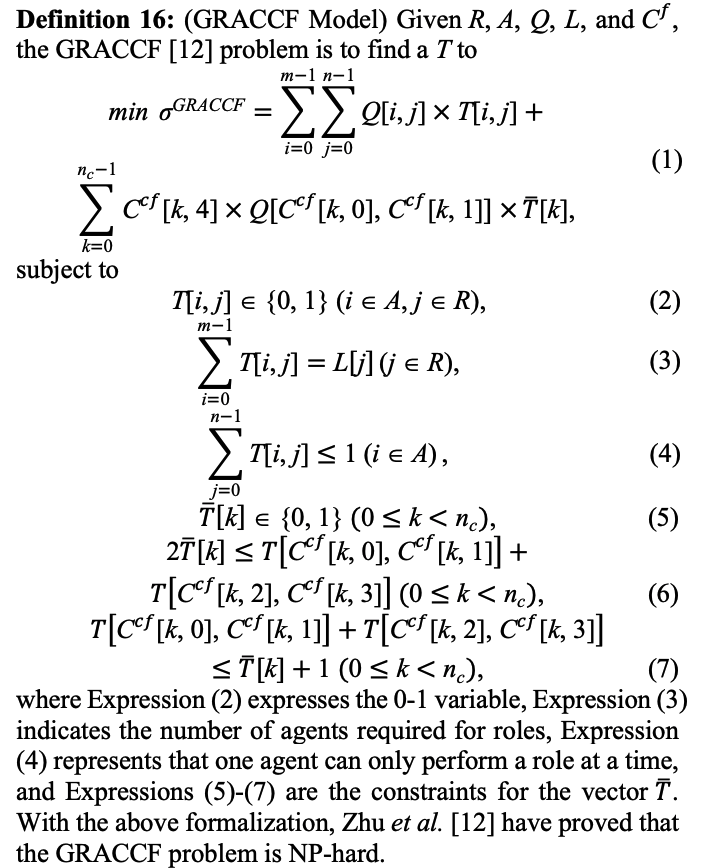


Fig. 2. Mathematical expression of the GRACCF model.

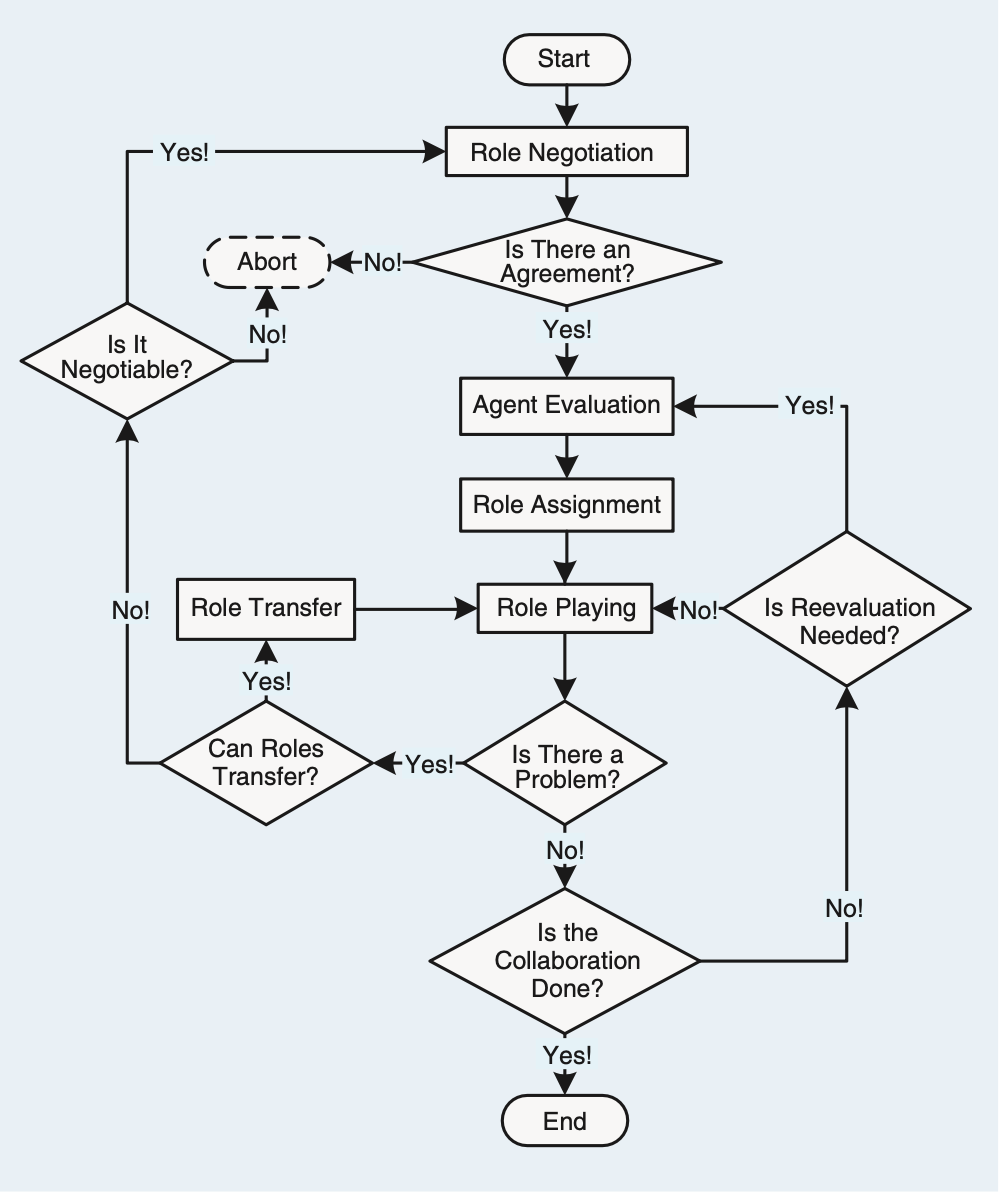
Learn for detail: <https://ieeexplore.ieee.org/abstract/document/7782466>

### 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 GRACCF submodel;

2. To practice using Python to program the GRACCF 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 6-8 agents (people, equipment, robots, groups of people, etc.). In addition to the scenario required for the first lesson assignment, **Add a constraint that when different agents perform different roles, they may have cooperation and conflict relationships. Use the *Ccf* matrix to represent this cooperation and conflict relationship. The degree of cooperation is represented by a floating-point number from 0 to 1, with larger values indicating more pleasant cooperation. The conflicting relationship is represented by a floating-point value from -1 to 0, with smaller values indicating less conflict.**

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 GRACCF 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.

\*11. This is an optional question. Read Ref [1], and based on your own understanding, explain the evolution process of the GRACCF model in Ref [1], that is, how does the GRACCF model transform from a nonlinear model to a linear model?

[1] H. Zhu, Y. Sheng, X. Zhou, and Y. Zhu, “Group Role Assignment With Cooperation and Conflict Factors,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 6, pp. 851–863, 2018.

### 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.

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\_4**