# Lab 1: Group Role Assignment

**Project 1**: Python-Pulp: Task Assignment using GRA.

1. **Preliminary Knowledge**
   1. **PuLP Package**

PuLP is an open-source linear programming (LP) package in Python. It provides tools for describing and solving linear and integer programs. PuLP can be used to model optimization problems as mathematical models, which can then be solved using various algorithms, or solvers. PuLP’s main benefits are that it’s easy to install, easy to use, and its syntax closely resembles mathematical expressions. This makes the process of formulating your problem, implementing it in PuLP, and interpreting the results straightforward. For example, you can solve a simple optimization problem such as: maximize x + y under the conditions x + y <= 2, x - y >= 0, and x, y >= 0. With PuLP, you can easily model this problem in Python and solve it using any of the available solvers, such as CBC or GLPK. Remember that PuLP is just an interface, a way to describe your problem in Python, it doesn’t solve the problem itself. It relies on external libraries (solvers) to find the solutions.

Learn for detail: https://www.coin-or.org/PuLP/pulp.html

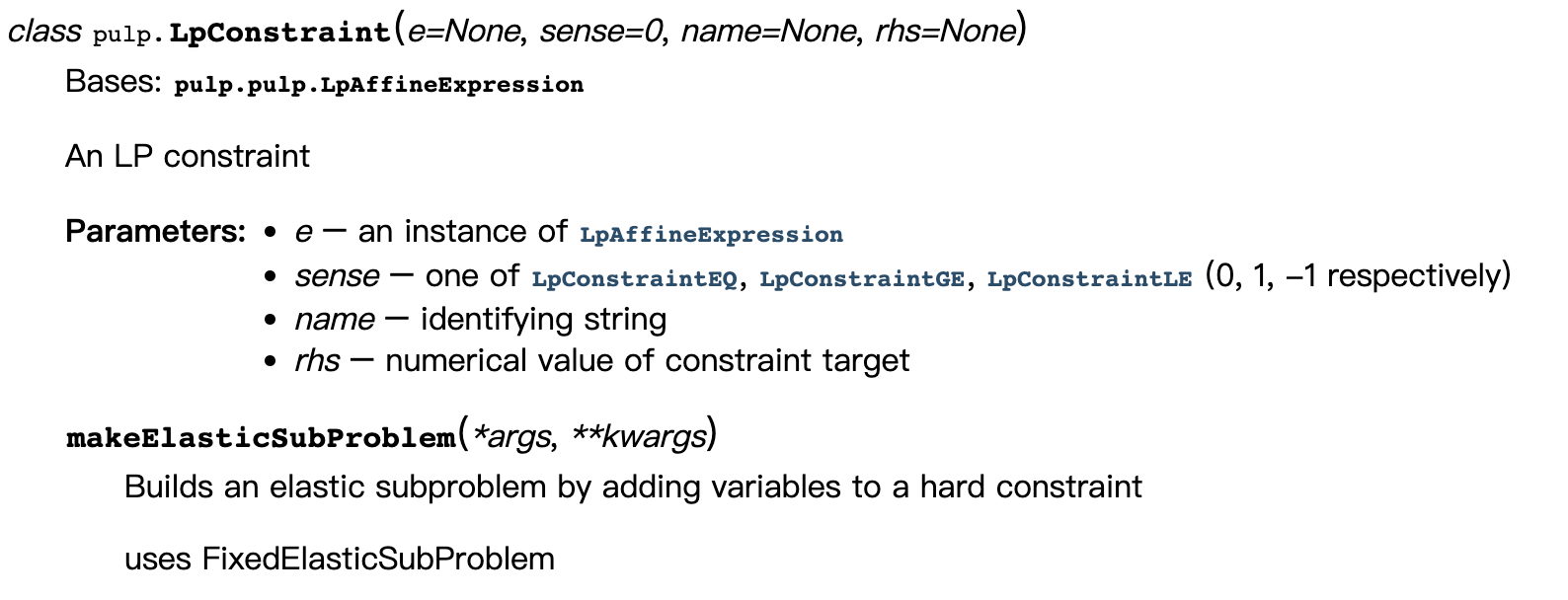


Fig. 1. Explaining the parameters of a certain function and their corresponding meanings on the website.

* 1. **Group Role Assignment (GRA)**

Group Role Assignment (GRA) is a submodel of the RBC methodology, and it is derived from the E-CARGO fundamental model. The mathematical expression of the GRA model is shown below. It is an efficient tool to solve the 1-M (one-to-many) related problems.

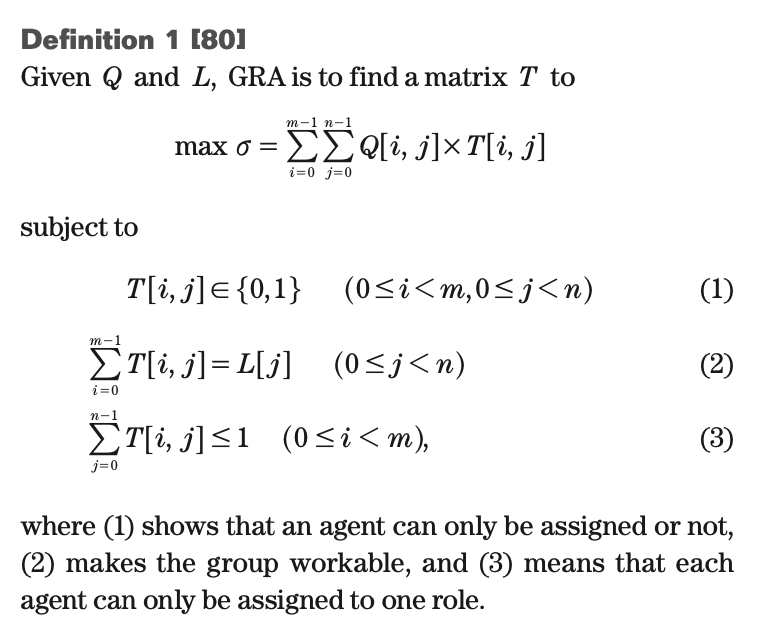


Fig. 2. Mathematical expression of the GRA model.

Learn for detail: <https://ieeexplore.ieee.org/abstract/document/7426538>

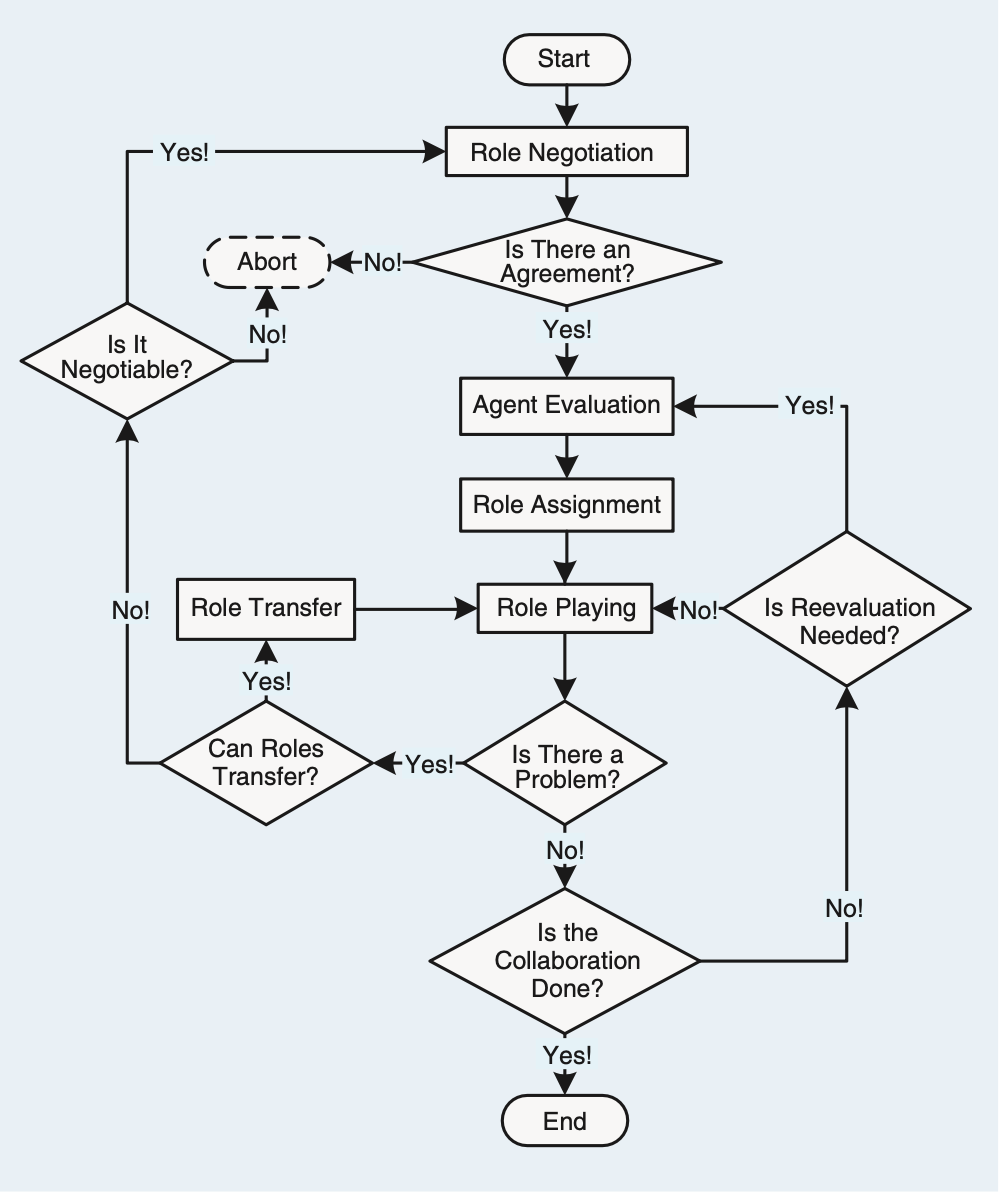
Source code from github: <https://github.com/haibinnipissing/E-CARGO-Codes/blob/main/Python_Pulp_GRA.txt>

### Objectives

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 GRA submodel;

2. To practice using Python to program the GRA model in order to solve the 1-M assignment problem.



**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.).

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.

### 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\_1**