# Developing a Task Allocator Using Network Flow and Linear Programming

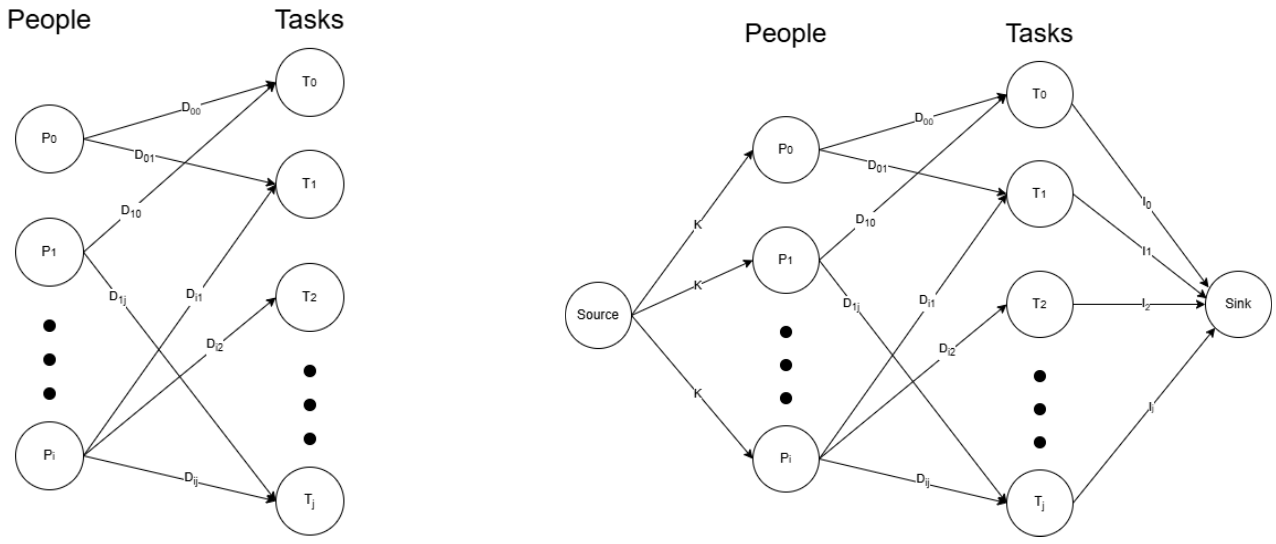
## Project Goal

The goal of this project is to develop a program that helps a group of people allocate tasks based on individuals' willingness to perform certain tasks and the importance of the task. I aim to apply this to two situations. The first is a house cleaning scenario, where roommates are able to divide up their weekly chores based on preference, ensuring that everything is completed at least every two weeks. The second is inspired by my robotics team and sets up a more complex scenario involving more conditionals. This new task allocator must divide tasks based on preference, skill, number of people per certain task, and time available. The first scenario, should solely be able to use a maximum flow algorithms. However, the extra constraints in the second scenario make it more easily solved with linear programming approach.

## Scenario One - House Chores

### Laying out the Problem

This problem can be represented as a network graph, shown in *Figure 1a*. This will allow us to use a maximum-flow algorithm to divide up the tasks. To ensure that I can use a network flow algorithm, I've added in total sources and sinks, shown in *Figure 1b*.



A: Original Graph

B: With Source and Sink

Figure 1: Graph Network of Hosue Chore Scenario

In these diagrams, the notation is as follows:

People are represented by $P_0, P_1, ..., P_i$

Tasks are represented by $T_0, T_1, ..., T_j$

The desires of a specific person to do a certain task are represented by $D_{00}, D_{01}, ..., D_{ij}$

The importance of a specific tasks is represented by $I_0, I_1, ..., I_j$

$K$ represents the maximum number of tasks each person is allowed to be assigned.

## Scenario Two - Robotics Tasks