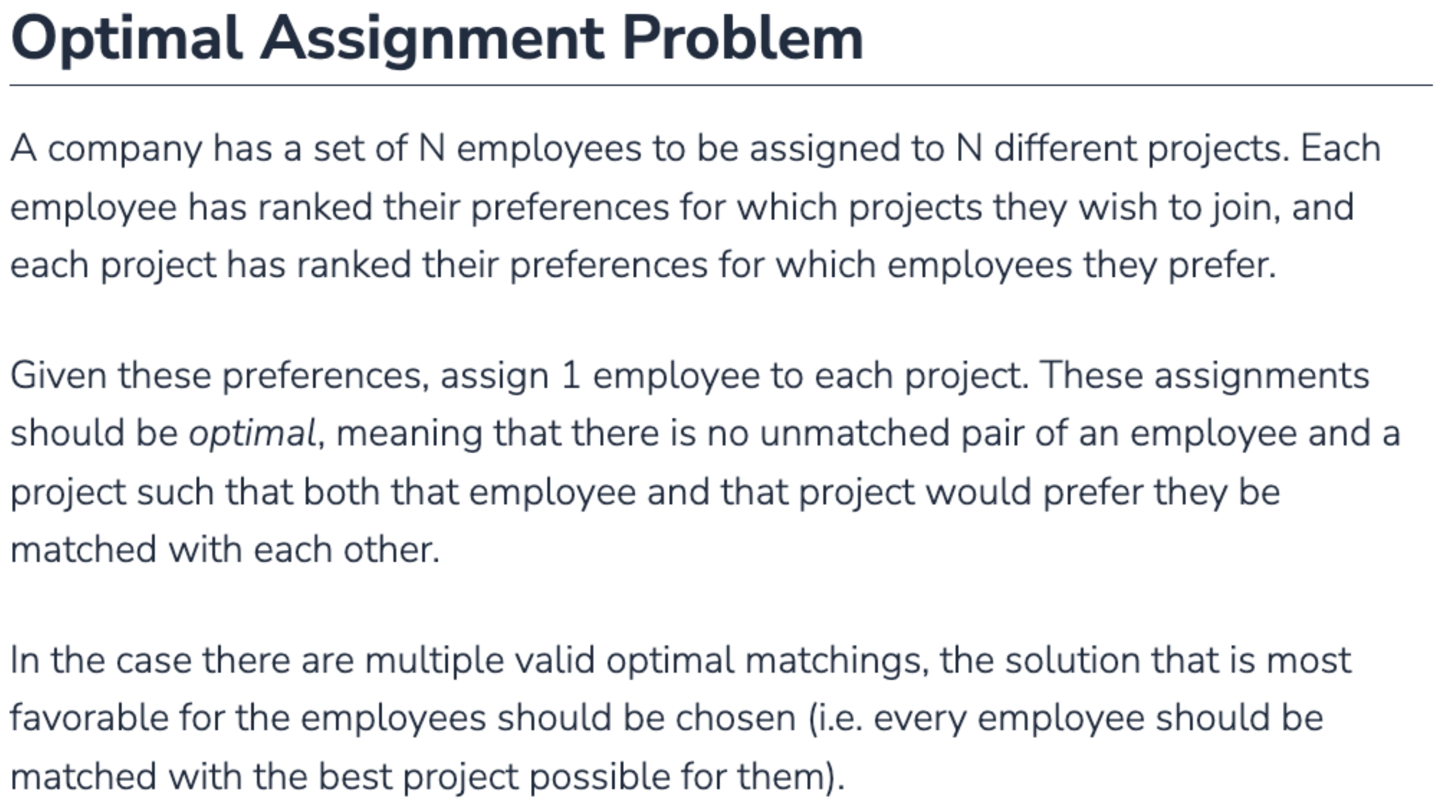
OPTIMAL ASSIGNMENT PROBLEM

DEFINITION



Optimal Assignment Problem

A company has a set of N employees to be assigned to N different projects. Each employee has ranked their preferences for which projects they wish to join, and each project has ranked their preferences for which employees they prefer.

Given these preferences, assign 1 employee to each project. These assignments should be *optimal*, meaning that there is no unmatched pair of an employee and a project such that both that employee and that project would prefer they be matched with each other.

In the case there are multiple valid optimal matchings, the solution that is most favorable for the employees should be chosen (i.e. every employee should be matched with the best project possible for them).

Step 1: Initial Setup

Employee Preferences:

[

Ranked choices

Employee0:[0,1,2],

Employee1:[0,2,1],

Employee2:[1,2,0]]

Project Preferences:

[

Ranked choices

Project 0:[2,1,0],

Project 1:[0,1,2],

Project 2:[0,1,2]]

Step 1: Initial Setup

`assigned_projects` :{ }

`free_employees` : [0, 1, 2] #these are our employees, start from the beginning or the ends does not matter (stack or queue up to you)

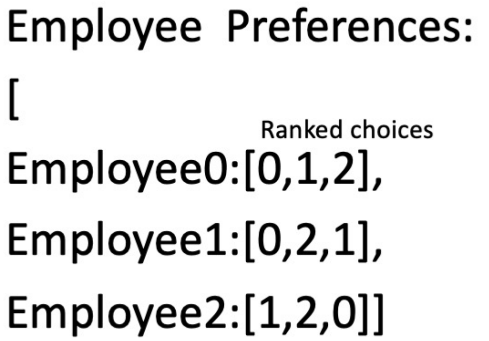
`current_employee_choices` : [0 ,0, 0] #this starts with each employee's highest-ranking choice which is the 0th index.

matches:[
{(2:0), (1: 0), (0:2)}

{(0:0), (1:1), (2:2)}

{(0:0), (1:1), (2:2) }
]

Step 2 from free employee let’s start from the end go with **Employee 2**



Employee Preferences:

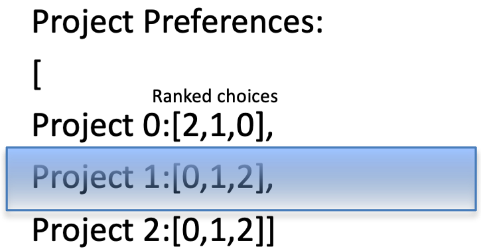
[

Ranked choices

Employee0:[0,1,2],

Employee1:[0,2,1],

Employee2:[1,2,0]]



Project Preferences:

[

Ranked choices

Project 0:[2,1,0],

Project 1:[0,1,2],

Project 2:[0,1,2]]

Employee 2 **prefers** Project 1 as their first choice

Chosen Employees: {Project 1: Employee 2}

Free Employee : [0,1]

Employee 2 is assigned to Project 1.

current_Employee _choices : [0 ,0, 1]

Step 3: Employee 1 Chooses

Employee Preferences:

[

Ranked choices

Employee0:[0,1,2],

Employee1:[0,2,1],

Employee2:[1,2,0]]

Project Preferences:

[

Ranked choices

Project 0:[2,1,0],

Project 1:[0,1,2],

Project 2:[0,1,2]]

Employee 1 prefers Project 0

Chosen Employee : {Project 1: Employee 2, Project 0: Employee 1}

Free Employee : [0]

Employee 1 is assigned to Team 0.

current_Employee _choices : [0 ,1, 1]

matches:[
{(2:0), (1: 0), (0:2)}

{(0:0), (1:1), (2:2)}

{(0:0), (1:1), (2:2) }
]

Step 4: Employee 0 Chooses

- Employee 0 prefers Project 0, but Project 0 is taken.
- Project 0 compares ranks between Employee 1 and Employee 0.Employee 1 is better ranked, **so Employee 0 remains free** and tries the next team. current_Employee _choices : [1 ,1, 1]
- Chosen Employee : {Project 1: Employee 2, Project 0: Employee 1}
- Free Employee : [0]

Project Preferences:

[

Ranked choices

Project 0:[2,1,0],

Project 1:[0,1,2],

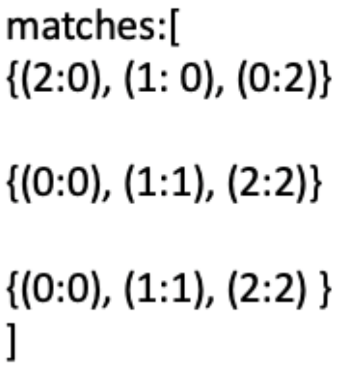
Project 2:[0,1,2]]

{Project 1: Employee 2, Project 0: Employee 1}

Step 5: Employee 0 Chooses Again

{Team 1: Candidate 2, Team 0: Candidate 1}

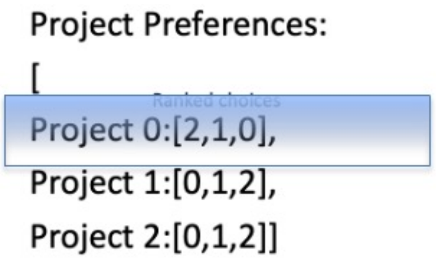
- Employee 0 now chooses Project 1.
- Project 1 compares Employee 2 with Employee 0.
- Employee 0 is better ranked than Employee 2, so Employee 2 is freed.
- Chosen Employee : {**Project 1: Employee 0**, Project 0: Employee 1}
  Free Employee : [2]



```
matches:[
{(2:0), (1: 0), (0:2)}

{(0:0), (1:1), (2:2)}

{(0:0), (1:1), (2:2) }
]
```



```
Project Preferences:
[
Project 0:[2,1,0],
Project 1:[0,1,2],
Project 2:[0,1,2]]
```

Step 6: Employee 2 Chooses Again

Employee 2 now chooses Project 2.

Team 2 is free, so Employee 2 is assigned to Project 2.

Chosen Employee :

{Project 1: Employee 0, Project 0: Employee 1, Project 2: Employee 2}

Free Employee : []

Algorithm completes.