IE 202 - INTRODUCTION TO MODELING AND OPTIMIZATION
PROJECT STAGE 2

SPRING 2023

April 7, 2023

Contents

Problem Definition

Optimization is simply defined as the methodology of doing something as good as possible with respect to some criteria. In mathematical optimization, the goal is finding the maximum or minimum of a function with given limitations. This process consists of mathematically modeling the problem while keeping the given limitations as “constraints” and the given function as an “objective function”, and solving the model using an optimization software. Such applications can be found in many different industries such as fast-moving consumer goods, the car industry, or logistics. Consulting is another industry in which optimization is heavily utilized.

A&N is a consulting firm that operates in more than 30 countries. In this consulting firm, there are 6 different types of positions: associate, senior associate, manager, senior manager, director, and partner. In the Ankara office, there are K employees in total with different position levels. In addition, the A&N consulting firm gives services in M different departments in this office. We assume that all employee has enough knowledge of all departments.

Currently, there are N available projects. Complexity of the projects differ in three levels. Projects in each level require a predetermined amount of workload (in terms of hours spent on the project) based on their difficulty levels. More complex projects are considered as more profitable.

A project can be assigned to a department if it is suitable for the department’s qualifications. While every department needs to conduct at least one project, a project can be assigned to a single department or left unassigned. There is a penalty cost for each project that is not assigned and done. Each employee must be assigned to at least one project but at most five. Also for each project, there is an upper and lower limit on the number of employees assigned from each position.

Workload requirement of project must be met by the employees. Every employee usually works 8 hours per day. Employees’ usual salaries per project depends on their position.

In addition to the normal working hours, employees may overwork only if they are assigned to a project with difficulty level 3. They get 50 dollars per each extra hour they work. Furthermore, regardless of their position, each employee can work up to H hours in a month, including overtime. (assume that there are 30 working days in a month).

Also, there are some specific requirements: Since the Department 1 and Department 2 have fierce competition, number of project assigned to these departments should be similar. If Department 1 will conduct Project 1 and 2, then Project 3 cannot be conducted by the Department 3.
Projects are prioritized based on their complexity. Due to this, if a project with difficulty level 3 is left unassigned then no project with difficulty level 2 can be completed.

Some reminders

- There are different ways to formulate a given problem. Below solution is just one of them. It is okey (and sufficient) if the formulation you proposed is consistent and somehow covers all elements given in the problem description.
- Remember, parameters are the *given* pieces of information in the problem description while decision variables are the things we are asked to determine. When you are not sure if something is a parameter of a decision variable ask the questions: "Will my model know its value prior to solving the problem or will it determine its value after solving the problem?" or "will it be an input or output while solving this problem?" Before writing your model, write your decision variables and parameters as separate lists. Otherwise, it becomes impossible to track your model (both for you and for us, graders).
- Any objective function or any constraint must include a decision variable: an expression without a decision variable is just another parameter (a number). It does not make any sense to maximize/minimize 5, right? Similarly maximizing/minimizing a is meaningless if we already know the value of a.
- If the constraint is valid for every element of a set (say I) then at the end of the constraint we write "for all $i \in \{1, \ldots, I\}$" indicating that we actually write this constraint for each element of the set I.
- You need to be very careful about what is "non-linear" and what is "linear". For example, multiplying parameters or a parameter with a decision variable is totally safe. Yet, multiplying two decision variables will result in non-linearity.
- If you use a binary variable to linearize some "if ($Condition1$) then ($Condition2$)" statement, it might be good to check what happens when this variable takes the value of 1 and 0. Does your constraint become redundant (as it should) when $Condition1$ does not hold?
- Do not attempt to formulate everything at once. First think about possible decision variables and parameters, then try to write the constraints and the objective function. If you cannot express something using the notation, try to update your variables (or their indices).
- Your model does not know the relation between the decision variables unless you clearly state them as constraints. If you miss such logical constraints, the model may yield unacceptable/inapplicable solutions (for example assigning customers to a facility which is not opened).
- Do not forget to include domain constraints for variables.

Part A

Write the mathematical model which makes the proper assignments between the elements of the company while maximizing monthly profit.

Sets & Indices:

Employee Types {1:Associate, 2:Senior Associate, 3:Manager, 4:Senior Manager, 5:Director, 6:Partner}: $t \in \{1, 2, 3, 4, 5, 6\}$

Department Types: $m \in \{1...M\}$

Employees: $k \in \{1...K\}$

Projects: $i \in \{1...N\}$

Difficulty level for projects: $j \in \{1, 2, 3\}$

Parameters:

$$et_{tk} = \begin{cases} 1, & \text{If employee } k \text{ is type } t, \\ 0, & \text{otherwise} \end{cases} \qquad k \in \{1...K\}, t \in \{1, \ldots, 6\}$$

R_i = Number of working hours required to complete project i, $i \in \{1...N\}$

L = Threshold value for the difference between the number of projects of the first and second departments have

p = Penalty cost for any undone project

H = Maximum working hours for each employee

G_j = Money earned from a project with difficulty type j, $j \in \{1, 2, 3\}$

S_t = Salary of employee type t, $t \in \{1...6\}$

ub_{ti} = Upper bound for employee type t, t for project i, $t \in \{1...6\} i \in \{1...N\}$

lb_{ti} = Lower bound for employee type t, t for project i, $t \in \{1...6\} i \in \{1...N\}$

$$d_{ij} = \begin{cases} 1, & \text{If project i is of difficulty level j,} \\ & \qquad i \in \{1...N\}, j \in \{1, 2, 3\} \\ 0, & \text{otherwise} \end{cases}$$

$$a_{im} = \begin{cases} 1, & \text{If project i is suitable to department m,} \\ & \qquad i \in \{1...I\}\ m \in \{1...M\} \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables:

$$x_{im} = \begin{cases} 1, & \text{If project i is assigned to department m,} \\ & \qquad i \in \{1...N\}\ m \in \{1...M\} \\ 0, & \text{otherwise} \end{cases}$$

$$e_{ki} = \begin{cases} 1, & \text{If employee k is assigned to project i} \\ & \qquad k \in \{1...K\}\ i \in \{1...N\} \\ 0, & \text{otherwise} \end{cases}$$

$$b_{i} = \begin{cases} 1, & \text{If project i is assigned to some department} \\ & \qquad i \in \{1...N\} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{k} = \begin{cases} 1, & \text{If employee k is assigned to some project} \\ & \qquad k \in \{1...K\} \\ 0, & \text{otherwise} \end{cases}$$

o_{ik} = Total number of overworking hours of employee k in a month for project $i \in \{1...N\}\ k \in \{1...K\}$

Model:

$$\max \quad \sum_{i=1}^{N}\sum_{j=1}^{3} d_{ij}\, b_i G_j \;-\; \sum_{i=1}^{N} p*(1-b_i) \;-\; \sum_{t=1}^{6}\sum_{i=1}^{N}\sum_{k=1}^{K} e_{ki}\, S_t\, et_{tk} \;-\; \sum_{k=1}^{K}\sum_{i=1}^{N} o_{ik}*50 \tag{1}$$

$$\sum_{i=1}^{N} e_{ki} \geq 1 \qquad k \in \{1...K\} \tag{2}$$

$$\sum_{i=1}^{N} e_{ki} \leq 5 \qquad k \in \{1...K\} \tag{3}$$

$$\sum_{k=1}^{K} e_{ki}\; et_{tk} \leq ub_{ti} \qquad i \in \{1...N\}\; t \in \{1...6\} \tag{4}$$

$$\sum_{k=1}^{K} e_{ki}\; et_{tk} \geq lb_{ti} \qquad i \in \{1...N\}\; t \in \{1...6\} \tag{5}$$

$$\sum_{i=1}^{N} x_{im} \geq 1 \qquad m \in \{1...M\} \tag{6}$$

$$\sum_{m=1}^{M} x_{im} \leq 1 \qquad i \in \{1...N\} \tag{7}$$

$$\sum_{i=1}^{N} x_{i1} - \sum_{i=1}^{N} x_{i2} \leq L \tag{8}$$

$$\sum_{i=1}^{N} x_{i2} - \sum_{i=1}^{N} x_{i1} \leq L \tag{9}$$

$$y_k * 8 * 30 + \sum_{i=1}^{N} o_{ik} \leq H \qquad k \in \{1..K\} \tag{10}$$

$$y_k \leq \sum_{i=1}^{N} e_{ki} \qquad k \in \{1..K\} \tag{11}$$

$$(|N|)y_k \geq \sum_{i=1}^{N} e_{ki} \qquad k \in \{1..K\} \quad (12)$$

$$o_{ik} \leq H * e_{ki} \qquad k \in \{1..K\}, i \in \{1..N\} \quad (13)$$

$$\sum_{m=1}^{M} x_{im} \geq b_i \qquad i \in \{1...N\} \quad (14)$$

$$x_{im} \leq a_{im} \qquad i \in \{1...N\}, m \in \{1...M\} \quad (15)$$

$$\sum_{k=1}^{K} e_{ki} * 8 * 30 + o_{ik} d_{i3} \geq R_i b_i \qquad i \in \{1, \dots, N\} \quad (16)$$

$$x_{11} \geq w_1 \quad (17)$$

$$x_{21} \geq w_1 \quad (18)$$

$$x_{11} + x_{21} - 1 \leq w_1 \quad (19)$$

$$1 - w_1 \leq x_{33} \quad (20)$$

$$(1 - b_i) d_{i2} \leq w_2 \qquad i \in \{1, \dots, N\} \quad (21)$$

$$1 - w_2 \geq (1 - b_i) d_{i3} \qquad i \in \{1, \dots, N\} \quad (22)$$

$$x_{im} \in \{0, 1\} \qquad i \in \{1, \dots, N\}\ m \in \{1, \dots, M\} \quad (23)$$

$$e_{ki} \in \{0, 1\} \qquad i \in \{1, \dots, N\}\ k \in \{1, \dots, K\} \quad (24)$$

$$o_{ik} \geq 0 \qquad i \in \{1, \dots, N\}\ k \in \{1, \dots, K\} \quad (25)$$

$$b_i \in \{0, 1\} \qquad i \in \{1, \dots, N\} \quad (26)$$

$$y_k \in \{0, 1\} \qquad k \in \{1, \dots, K\} \quad (27)$$

Explanations of the Constraints and the Objective Function

1. Objective function (1) maximizes the total monthly profit of the company. This consists of four parts. First part is the total money earned from all projects. The other three parts are cost elements thus, subtracted from the revenue. Second part is related to the penalty cost of undone projects. Third and fourth parts are associated with the salary given to employees for usual working hours and overworking hours.

2. Constraints (2) and (3): Every employee should be assigned to at least one, at most five projects.

3. Constraints (4) and (5): For each project, there is an upper and lower limit on the number of employees assigned from each position.

4. Constraint (6): Every department should conduct at least one project.

5. Constraint (7): Every project should be assigned to at most one department.

6. Constraints (8) and (9): Number of project assigned to Department 1 and Department 2 should be similar. These constraints are used to linearize the following:

$$\left|\sum_{i=1}^{N} x_{i2} \quad - \quad \sum_{i=1}^{N} x_{i1}\right| \quad \leq \quad L$$

7. Constraint (10): Total working hours of an employee cannot exceed H hours. This consists of usual working hours (if assigned to a project) and overworking hours.

8. Constraints (11) and (12): These constraints are used to relate the variables y_k and e_{ki}. If $\sum_{i=1}^{N} e_{ki} = 0$ for employee k, this means they are not assigned to any project. In this case: i) Constraint (11) forces y_k to be 0, ii) Constraint (12) becomes redundant. Please note that $|N|$ acts as a big M. If $\sum_{i=1}^{N} e_{ki} > 0$, then employee k is assigned to at least one project. In this case i) Constraint (11) becomes redundant, ii) Constraint (12) forces y_k to be 1.

9. Constraint (13) relates the variables o_{ik} and e_{ki} in the sense that an employee cannot overwork for a project if they are not assigned to that project.

10. Constraint (14) relates the variables x_{im} and b_i. If department m does not conduct any project i.e., if $\sum_{i=1}^{N} x_{im} = 0$, then b_i must be 0. This inequality can be also be converted to an equality due to Constraint (7).

11. Constraint (15) states that a project can be assigned to a department only if it is suitable for the department's qualifications. This "suitability" is expressed as a binary parameter (a_{im}).

12. Constraint (16) ensures that the workload requirements of projects are met by the employees. Note that this is valid when the project is assigned to some department (when b_i is 1). This is why the right hand side of the constraint is multiplied with the variable b_i. If the project

is not assigned to any department, then this constraint becomes redundant. On the left hand side, total usual working hours of assigned employees and overworking hours are summed up. Please note that overworking hours are counted only if the project is of difficulty level 3. This is ensured by multiplying the overworking variable (o_{ik}) with the binary parameter d_{i3}.

13. Constraints (17), (18), (19) and (20): If Department 1 will conduct Project 1 and 2, then Project 3 cannot be conducted by the Department 3. Constraints (17), (18) and (19) are used to linearize multiplication of two binary variables ($x_{11}x_{21}$) by introducing a new binary variable w_1:

$$x_{11}x_{21} = w_1$$

Following this, Constraint (20) ensures that if $w_1 = 1$ then $x_{33} = 0$.

14. Constraints (21) and (22): If a project with difficulty level 3 is left unassigned then no project with difficulty level 2 can be completed. If a project with difficulty level 3 is left unassigned then for some $i \in \{1, \ldots, N\}$, $(1 - b_i)d_{i3}$ should be 1. This forces binary variable w_2 to be 0. Then Constraint (21) ensures that $(1 - b_i)d_{i2} = 0$ for all $i \in \{1, \ldots, N\}$ (in other words, no projects with difficulty level 2 can be assigned to a department). Please note that if all projects with difficulty level 3 are assigned, then these constraints become redundant.

15. Constraints (23-27) are the domain constraints for the decision variables.

Part B

Revise your model such that the objective of the firm is to maximize the number of completed projects. How would the monthly profit change in this case?

New objective function:

$$max \quad \sum_{i}^{N} b_i \tag{28}$$

Total profit would decrease or stay the same since it is not maximized in the updated model.

How to determine modelling features (variables, parameters, constraints and objective function) in a problem definition?

I wanted to share with you how I would formulate such a problem as a mathematical model step by step. Note that this is not *the only* way to approach such a problem, this is just how my mind works. Nonetheless, I believe that some of you may benefit from this recipe :)

Step1 Carefully read the problem description and detect your sets. For example: set of projects, set of employees...

Step2 Give distinct indices to your sets. For example: i is the index for the project set.

Step3 Write parameters, decision variables, constraints **using words**. You can use this as an intermediate step between reading the description and writing the mathematical model to ease your task. For example if you see a statement "The company earns a pre-determined amount of money for each project" while reading the project description, take a note "profit" on your parameter list. After completely reading the problem description, determine the index (or indices) you would use for this parameter. For constraints: lets say you read a statement "each project should be assigned to at least 1 department", take a note "number of departments project i is assigned ≥ 1 for all i".

Step4 Make assumptions if there are vague statements or unspecified points.

Step5 Now, use **mathematical notation** for your parameters and decision variables.

Step6 Write your objective function using this notation. Check if all of the indices are covered in the sum functions.

Step7 Try to see the natural relation between your decision variables. Write the constraints relating the variables to each other using the mathematical notation.

Step8 Write the limitations given by the problem description as constraints using the mathematical notation. First determine which decision variables and parameters to include. Then construct the right and left hand sides of the constraint. Do not forget to check "for all..." parts. Tackle the non-linear parts using the methods you have learned during the lectures (define new variables if necessary).

Step9 Find proper values for Big M's you used in the formulation.

Step10 Write the domain constraints for your variables.

Step11 Read the problem description one last time to check if your model covers every requirement you are asked to include.

Lets go over the problem description given in this project and try to see the model elements (parameters,decision variables,constraints,objective function):

1. *In the Ankara office, there are K employees in total with different position levels.*
This sentence indicates that you should have a parameter to express which employee is of which type. You can do this by using sets (employee set for each type) or binary parameters. Lets go with the second option. This parameter (et) determines the binary relationship between *each type* and *each employee* therefore should be indexed over both.

2. *Complexity of the projects differ in three levels.*
Very similar to the first one. Now we need to know which project is of which difficulty level. You can do this by using sets (project set for each difficulty level) or binary parameters. Lets use a binary parameter. This parameter (d) determines the binary relationship between *each difficulty level* and *each project* therefore should be indexed over both.

3. *Projects in each level require a predetermined amount of workload (in terms of hours spent on the project) based on their difficulty levels.*
So, we need some parameter, lets say R, to express this pre-determined amount of workload for each project. The sentence indicates that this information depends on the "difficulty level" of the project, therefore we can either use difficulty level index (j) or project index (i) for this parameter.

4. *More complex projects are considered as more profitable.*
This tells that each project has a known profit value. Lets denote it as G. Similar to the previous one, this parameter can be indexed over projects or difficulty levels (since the sentence indicates that this value depends on difficulty type).
Since we are asked to write a mathematical model to maximize the monthly profit, the parameter we defined above (G) will be used in the objective function. Do not forget that this value should be counted in the objective function if the corresponding project is assigned to some department. If not, the company does not earn any profit from that project.

5. *A project can be assigned to a department if it is suitable for the department's qualifications.*
Departments have qualifications and might be suitable or not for each project. Suppose we denote this parameter as a. We can clearly see that this is a binary situation (suitable/not suitable) and changes for each project and for each department. Therefore a should be indexed over projects and departments.
The first part of the sentence indicates a decision: each project can be assigned to a department. Note that this is something our model will tell us based on the given information. Hence, it is a decision variable (x). Our model will decide if a project is assigned to a department or not, which tells us this variable is binary: it will be either 0 or 1.
This sentence also expresses a limitation on the previously defined binary variable (x) based on the previously defined "suitability" parameter between departments and projects (a). We should write a constraint stating that if a department is not suitable for a project ($a = 0$) then the project should not be assigned to that department ($x = 0$) (see Constraint 15).

6. *While every department needs to conduct at least one project, a project can be assigned to a single department or left unassigned.*
Fortunately for us, the first part is a clear constraint! We already defined a decision variable for the assignment of project to departments (x). Now, we have to ensure that each department is assigned at least one project. Lets start with the easy part: "At least one" will look something like "≥ 1", right? Now, we will decide what to place on the left hand side of this inequality. It should be "the number of project assigned to a department". Lets take Department 1. Remember we have the assignment variable x_{1m} that takes the value of 1 is project number m is assigned to Department 1. If we sum these variables over all projects, assigned projects will count as 1 while the unassigned projects will count as 0 (as we are counting binary variables). So, this sum will give total number of projects that are assigned to Department 1:
$$\sum_{i=1}^{N} x_{1m} \geq 1$$
We need to write this constraint for every department. We can explicitly write them one by one if we have a small sized department set, but what if we have 100 departments? There is a simpler way:
$$\sum_{i=1}^{N} x_{\mathbf{i}m} \geq 1 \quad \forall m \in \{1, \ldots, M\}$$
Using the expression of "for all", we can write the same constraint regardless of the value of M (see Constraint 6).
Similarly, we also should ensure that each project is assigned at most one department, as the second part of the sentence suggests. Note that this time the constraint will be written for all projects (see Constraint 7).

7. *There is a penalty cost for each project that is not assigned and done.*
The value of this cost is (or will be) given before we attempt to solve the problem. Therefore, we should define it as a parameter (p). This can differ for each project, or not. The statement does not indicate which one is true. This is one of the things that are written vaguely on purpose. At this point, you need to make an assumption on which path you want to take: you can use either p without any indices (penalty cost is the same for all projects) or p_i with project index (penalty cost depends on the project). Even p_j is possible (penalty cost depends on the difficulty level).
This sentence gives us something that affects the monthly profit. Therefore it should be (somehow) placed in the objective function.
If something (penalty cost in this case) occurs based on a binary event (assigned to some department/left unassigned), then a binary variable should be written to control when will it happen. That is exactly why we need b_i for each project i. Our model will decide which projects will be done and which will be left unassigned.
We have introduced a new variable (b) above. Now we need to check if there is any logical relationship between this variable and the old ones. If there is, we need to convey this

relationship to our model through constraint(s). In this case, b is closely related to x. For example if project i is assigned to a department (if $x_{im} = 1$ for some $m \in \{1, \ldots, M\}$), then the value of b_i should be decided accordingly (see Constraint 14).

8. *Each employee must be assigned to at least one project but at most five.*
   Any task that is given to us (the model) indicates a decision variable. Here, similar to x, we need to perform another assignment: assignment of projects to employees. Since this is a binary situation (assigned/not assigned) our variable must be a binary variable. Lets denote it as e.
   We need to write some constraint(s) to control how many projects are given to an employee. As this suggests, we need to write this constraint for all employees (see Constraints 2 and 3).

9. *Also for each project, there is an upper and lower limit on the number of employees assigned from each position.*
   We have upper (ub) and lower (lb) bounds for each project and for each position. Therefore these parameters should be indexed over both positions and projects.
   "For each project..." and "...From each position" parts tell us we need to write a constraint and place "$\forall i \in \{1, \ldots, N\}$ and $\forall t \in \{1, \ldots, 6\}$" at the end. On the left hand side we need to mathematically write "number of type t employees assigned project i". Can I write $\sum_{k=1}^{K} et_{tk}$? No, this would disregard the assignment variable (it would count all employees without considering if they are actually assigned to this project). I need to include the variable e into this mathematical expression. Can I write $\sum_{k=1}^{K} e_{ki}$? No, this would disregard the position of the employee. What about $\sum_{k=1}^{K} e_{ki} et_{tk}$? Yes! This would count employees if and only if i) they are assigned to project i ($e_{ki} = 1$) and ii) they are of type k ($et_{tk} = 1$). On the right hand side, we use the parameters we defined above (ub, lb) (see Constraints 4 and 5).

10. *Workload requirement of project must be met by the employees. Every employee usually works 8 hours per day.*
    We need to use one of the parameters we defined earlier: R_i (workload requirement of project i). Intuitively, we think of something like ".... $\geq R_i$" for any project i. Lets try to find the left hand side step by step. We need to mathematically write "total working hours spent on project i". We can define a variable to indicate the hours spent by employee k on project i and sum this over all employees. We can also assume that 240 hours (8 hours/day * 30 days) will be spent on a project by employee k, if the employee k is assigned to this project (if $e_{ki} = 1$). So, total working hours spent on project i will be expressed as $\sum_{k=1}^{K}(240e_{ki})$. There is also a hidden logical condition on the workload requirement. Workload requirement of project must be met by the employees **only if the project is assigned to somewhere**. If the project will be left undone, then the constraint we write should be redundant. Which variable is controlling whether a project is left unassigned or not? b. So we can somehow incorporate this variable such that the constraint we write will be active when b_i is 1 and redundant when b_i is 0. Since we are using "$\geq$" sign and want this constraint to be redundant

when b_i is 0, it might be useful to use b_i on the right hand side. Since R_i is a parameter, we can multiply it with b_i. Now, the constraint does what we want when b_i is 1, and becomes redundant by taking the form "... ≥ 0" when b_i is 0. So the current form of the constraint is as follows:

$$\sum_{k=1}^{K} e_{ki} * 8 * 30 \geq R_i b_i \quad i \in \{1, \ldots, N\}$$

11. *Employees' usual salaries per project depends on their position.*
Here, we detect another parameter. Each employee has a salary (granted for each assigned project): S. Statement indicates this value depends on the position of the employee, so it is safe to use "employee type" index.
This statement is closely related to our objective function as it will contribute to the total monthly expense of the company. Note that this amount will be given to the employee for every project they are assigned to. In other words, when $e_{ki} = 1$ and $et_{tk} = 1$ then amount S_t will be subtracted from the monthly profit in the objective function. We can tackle this by multiplying parameters S_t, et_{tk} with variable e_{ki}. Since we want to find the total salary given to all employees for all projects, we sum this expression over set of projects, set of employee types, and set of employees.

12. *In addition to the normal working hours, employees may overwork only if they are assigned to a project with difficulty level 3. They get* 50 *dollars per each extra hour they work.*
The statement says that employees "may" overwork. Therefore, this is something that we (our model) will decide. We need to introduce a new variable, say o_k for an employee k. Should it be binary (1 if employee k overworks) or not (number of hours employee k overworks)? At the end of the statement, we are told that a variable cost will incur **per each overworking hour**. If we use a binary variable, then we cannot capture the hourly increase in this cost value. However if we define the variable as "number of hours employee k overworks", then cost per hour can be simply multiplied with this variable in the objective function. The statement also puts a condition on overworking practice of the company. Difficulty level of the project must be 3 if they want someone to overwork for it. But we defined our variable in an aggregated fashion: it indicates the **total** overworking hours of an employee over all projects. This means that we need to update how we define variable o_k by integrating project index i in addition to the employee index k. Now, we can differentiate overworking hours by considering which project they are spent on.
Similar to the penalty cost and usual salaries, this will affect our objective function. Using the decision variable we defined, o_{ik}, we need to subtract the total cost spent for overworking hours from our monthly profit.
Since we introduced a new variable o_{ik}, we need to check the logical relations between this and other parameters. For example, an employee cannot overwork for a project if they are not assigned to that project. This is the relation between e_{ki} and o_{ik}. Since one is binary and other is not, we should be careful while constructing the logical constraints (Big M's might be necessary) (see Constraint 13).

We should ensure that no one overworks for projects with difficulty levels 1 and 2 (using parameter d_{ij} and variable o_{ik}).
Now, we need to update the constraint we wrote above for workload requirement of the projects. The current form is given as:

$$\sum_{k=1}^{K} e_{ki} * 8 * 30 \geq R_i b_i \quad i \in \{1, \ldots, N\}$$

We know that overworking hours will contribute to the left hand side of this constraint. And overworking hours should be summed over all employees. The constraint is updated as:

$$\sum_{k=1}^{K} (e_{ki} * 8 * 30 + o_{ik}) \geq R_i b_i \quad i \in \{1, \ldots, N\}$$

We can also go ahead and count overworking hours only if the project is of difficulty level 3. This would yield the following constraint (see Constraint 16):

$$\sum_{k=1}^{K} (e_{ki} * 8 * 30 + o_{ik} d_{i3}) \geq R_i b_i \quad i \in \{1, \ldots, N\}$$

As you can see, the variable o_{ik} will be count only if $d_{i3} = 1$. Since d is a parameter, this multiplication does not harm linearity. If we use this form of the above constraint, we do not have to explicitly say "no one overworks for projects with difficulty levels 1 and 2". Because even then, our model will not break this rule due to two things: i) overworking will worsen the objective function value, ii) overworking hours for less difficult projects will not contribute to workload requirement. Without contributing to the workload requirement, overworking is just wasting money & the model will not select this option.

13. *Regardless of their position, each employee can work up to H hours in a month, including overtime. (assume that there are* 30 *working days in a month).*
This describes a new parameter, H. Since this is a constant value for all employee, we do not need any indices.
We are dealing with a constraint in the form of "... $\leq H$ " for all employee $k \in \{1, \ldots, K\}$. "For all" part is due to "each employee" statement given in the sentence. Lets determine the left hand side of this inequality. We need to mathematically write "total monthly working hours for employee k". This consists of two things: usual working hours and overworking hours (see Constraint 10). Now, you can make an assumption and say "each employee usually works for 240 hours even if they are not assigned to any project". Or, you can assume that "each employee works for 240 hours only if they are assigned to some project". Suppose we select the second option. Then, we need something to indicate if employee i is assigned to some project or not. This calls for a binary variable: y_k (1 if employee k is assigned to a project, 0 otherwise). Since we add a new variable, we need to check its relation to the previous ones. Clearly, y_k is related to e_{ki} (see Constraints 11 and 12).

14. *Since the Department 1 and Department 2 have fierce competition, number of project assigned to these departments should be similar.*
This is a vague statement and we need to make an assumption on how to ensure number of projects assigned to these departments are "similar". Lets say that number of projects assigned to these departments are similar when their difference is under a threshold. The moment we define a threshold, it becomes a problem parameter (lets denote this as L) and we need to write a constraint in the form of "... $\leq L$" (given that this parameter is an upper limit for something). The left hand size of this inequality should indicate "difference between the number of projects assigned to Dept.1 and 2.". Lets further decompose this: what is the "number of projects assigned to Dept. 1"? It should be related to the variable x_{i1}. If you sum these variables over the project index i, then it will be equal to the number of projects assigned to Dept 1. Similarly, number of projects assigned to Dept. 2 is $\sum_{i=1}^{N} x_{i2}$. We want to limit the difference between these two sums, which is equal to $\left|\sum_{i=1}^{N} x_{i2} - \sum_{i=1}^{N} x_{i1}\right|$. We found the left hand side of our constraint! The next step is to linearize this absolute value (see Constraints 8 and 9).

15. *If Department 1 will conduct Project 1 and 2, then Project 3 cannot be conducted by the Department 3.*
The constraint will be related to the assignment variables between projects and departments. More specifically, x_{11}, x_{21} and x_{33}. Lets convert the statement into a mathematical form using these variables: if x_{11} and x_{21} are 1 then $x_{33} = 0$. Note that "x_{11} and x_{21} are 1" is equivalent to "$x_{11}x_{21} = 1$". This is not linear of course, but we will deal with it later. The constraint we want should force x_{33} to be 0 (or ≤ 0) under some conditions. So, let us start with an uncompleted form: "$x_{33} \leq$ ...". Right hand side should take the value of 0 when "some condition" holds (i.e. when $x_{11}x_{21} = 1$) and it should be 1 when the condition does not hold (i.e. when $x_{11}x_{21} = 0$). The latter case will yield a redundant constraint (it becomes $x_{33} \leq 1$). What can be this right hand side? $1 - x_{11}x_{21}$ perfectly fits our needs (see Constraint 20). This non-linear expression can be linearized by introducing a new binary variable (see Constraints 17,18 and 19).

16. *If a project with difficulty level 3 is left unassigned then no project with difficulty level 2 can be completed.*
This should be related to the variables indicating projects are assigned or not (b). Since we need to consider the difficulty level of projects, we need to integrate the parameters d in these constraints. Now that our ingredients are ready, we can construct the constraint. How can we mathematically say "a project with difficulty level 3 is left unassigned"? For this statement to be true, two things should hold for a project i: i) $(1 - b_i)$ should be 1 and ii) d_{i3} should be 1. This is equivalent to say that $d_{i3}(1 - b_i) = 1$. If $d_{i3}(1 - b_i) = 1$ for some project i, then for all projects the following should hold: $d_{i2}(1 - b_i) \leq 0$ (no project with difficulty level 2 is completed). The next step is to linearize this "if-then" statement using a new binary variable (see Constraints 21 and 22).

Data

Sets & Indices:

Employee Types {1:Associate, 2:Senior Associate, 3:Manager, 4:Senior Manager, 5:Director, 6:Partner} t = {1,2,3,4,5,6}

Department Types = {1...8}

Number of employees = {1...20}

Number of projects = {1...10}

Difficulty level for projects = {1,2,3}

Parameters:

$$R_i \quad = \quad \begin{bmatrix} 250 & 1800 & 1050 & 3550 & 550 & 3600 & 700 & 3650 & 1350 & 2800 \end{bmatrix}$$

$$et_{tk} \quad = \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L \quad = \quad 2$$

$$p \quad = \quad 1500$$

$$H \quad = \quad 280$$

$$G_j \quad = \quad \begin{bmatrix} 10000 & 50000 & 800000 \end{bmatrix}$$

$$S_t \quad = \quad \begin{bmatrix} 8000 & 16000 & 24000 & 32000 & 40000 & 48000 \end{bmatrix}$$

$$ub_{ti} = \begin{bmatrix} 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 3 \\ 3 & 2 & 3 & 2 & 3 & 3 & 3 & 2 & 2 & 2 \\ 2 & 2 & 2 & 3 & 2 & 2 & 2 & 2 & 3 & 2 \\ 3 & 2 & 3 & 2 & 2 & 2 & 3 & 2 & 3 & 3 \\ 2 & 3 & 2 & 2 & 2 & 3 & 3 & 2 & 3 & 3 \\ 2 & 3 & 3 & 2 & 3 & 2 & 2 & 2 & 3 & 3 \end{bmatrix}$$

$$lb_{ti} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$d_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$a_{im} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Part C

Yigit&Bora Partners is a consulting firm that wants to assign its four senior members; Yigit, Bora, Gozde, and Michelle to four companies, namely; Amazor, ProctorGambler, Pegasos, Sielens. Each company has 3 available projects. It means there are 12 available projects in total. The hourly works must be done to complete each project is given below.

	Amazor	Proctor&Gambler	Pegasos	Sielens
Project 1	13	70	45	60
Project 2	45	66	33	80
Project 3	60	55	98	40

Table 1: Hour Table

Each senior will paid hourly for the work they done.

Seniors	Costs
Yiğit	100
Bora	85
Gözde	90
Michelle	80

Table 2: Senior Costs

Bora and Michelle are busy these days, so they cannot work more than 80 hours. Also, company has a policy which restricts an employee work less than 60 hours and more than 1000 hours. Gözde is new to the company so she needs to work more than 150 hours to gain experience in project 1, however she cannot work in project 3 since they are difficult. Also Michelle has a request to not work in any companies project 1. Yiğit also has request to work with Proctor&Gambler at least 30 hours. (Assume that an employee can do all the projects, and more than 1 employee could assign to a single project of any companies) Find the minimum cost to complete all projects. Solve this **Linear Programming** problem on Excel Solver and conduct a sensitivity analysis.