Operations Research in Data Management

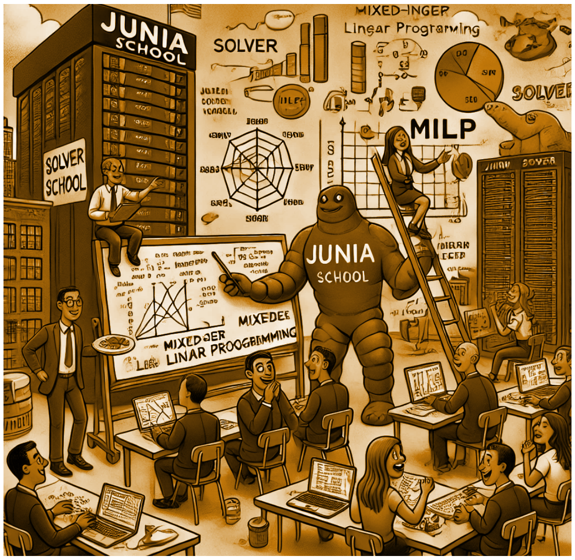
(M1)

Practical lab

(a group of 6 to 8 students)

Data Storage

Company Management



pic: chatGPT 4o

Abstract. Data management and storage are critical components of any enterprise's IT infrastructure. This project focuses on optimizing operations related to **disk packing**, **data center location**, and **data transport** problems for a hypothetical large-scale cloud service provider. The aim is to minimize operational costs while maintaining high efficiency in data handling and storage. By solving the continuous relaxation and then the MILP for several instances, you will explore the complexities and interdependencies of real-world optimization problems.

JUNIA ISEN

amina.el-yaagoubi / samuel.deleplanque at junia.com

OVERVIEW

The company JUNIATA is fortunate to have been created by …you! In the long term, it will offer storage spaces to businesses and individuals. To ensure the company is profitable, it is essential to minimize costs while meeting certain constraints and maintaining a defined quality of service. You are responsible for optimizing your company, particularly in these three areas:

- P1: Minimization of the cost of the storage units
- P2: Minimization of the total cost of opening data centers
- P3: Maximization of the speed of system updates in data centers

These three problems are actually operations research problems that can be modeled using mathematical programs. Some were covered in the course in a different form, or are provided in this document.

The problems P1, P2, and P3 are interconnected in the sense that the solution found for P1 becomes part of the inputs for P2, and the solution to P2 then becomes an input for P3 (see Fig. **1**). Nevertheless, to ensure you are not blocked in case you cannot find the solution to the previous problem, sufficient data will be provided to you.

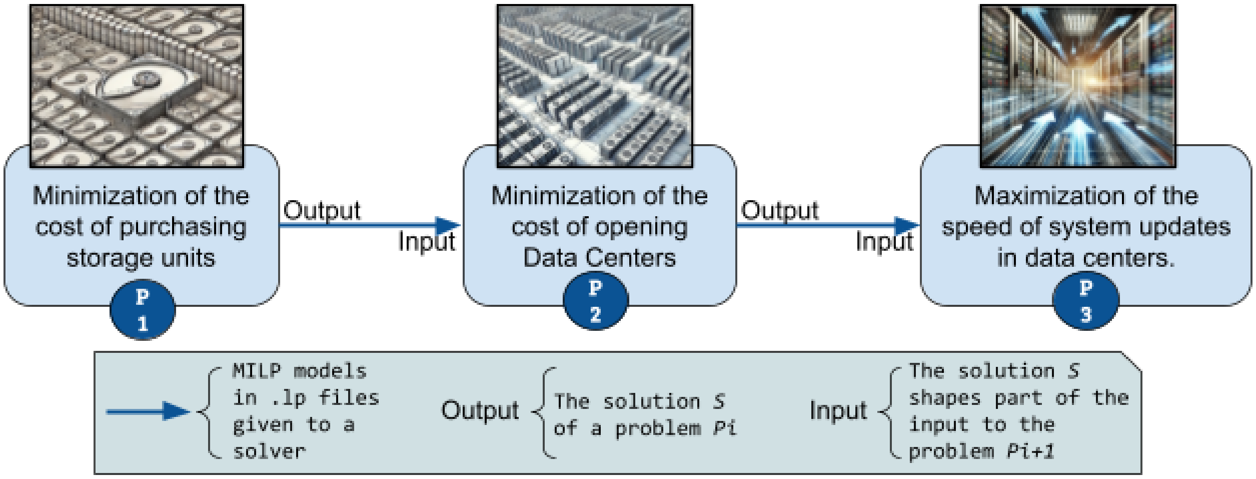

Minimization of the cost of purchasing storage units
P 1
Output
Input
Minimization of the cost of opening Data Centers
P 2
Output
Input
Maximization of the speed of system updates in data centers.
P 3
MILP models in .lp files given to a solver
Output
The solution S of a problem Pi
Input
The solution S shapes part of the input to the problem Pi+1

Figure 1. Diagram summarizing the 3 operations research problems to be addressed in this lab. (3 pics: chatGPT 4o)

For each situation, we ask you to write the mathematical model in an .lp file, and depending on the case, the continuous relaxation (real variables) and the program with integer variables will be solved.

TOOLS

You do not necessarily need to install a solver for .lp files; you can directly upload your .lp files to online solvers, such as:

- https://cocoto.github.io/glpk-online/ : A simple interface for solving linear programs using the GNU Linear Programming Kit (GLPK).
- https://neos-server.org/neos/solvers/index.html : A powerful platform that provides access to several solvers, including open-source and commercial solvers like CPLEX, Gurobi, and GLPK.
- https://neos-server.org/neos/solvers/lp:CPLEX/LP.html : Direct access to IBM's CPLEX solver for solving linear programming problems.

There are also many installable solvers, some of which are open-source, including:

- https://www.gnu.org/software/glpk/ : The GNU Linear Programming Kit is a free, open-source solver for linear programming and mixed-integer programming problems. It can be installed locally and used via command line or integrated into other systems (e.g., Python).
- https://github.com/coin-or/Cbc : The COIN-OR Branch and Cut solver is another open-source MILP solver that is highly customizable.
- https://lpsolve.sourceforge.net/5.5/ : An open-source linear programming solver that can handle both linear and integer programming problems.
- https://www.gurobi.com/ : A commercial solver that offers free academic licenses. It is known for its speed and performance on large-scale problems.
- https://www.ibm.com/products/ilog-cplex-optimization-studio : A commercial optimization solver by IBM, which also provides free licenses for academic purposes.

DATA & TASKS (P1)

Context

Optimize the packing of data onto physical storage disks to minimize the number of disks used. Efficient disk packing is essential for maximizing storage utilization and minimizing costs by determining how to pack data efficiently.

Case to be addressed

Four different levels of quality are offered to the client:

- Basic, with a capacity of **2TB**
- Medium 1, with a capacity of **5TB**, but without data redundancy
- Medium 2, with a capacity of 5TB, with redundancy (effectively **10TB**)
- Premium, with a capacity of 10TB, with redundancy (effectively **20TB**)

10 clients[1] require a certain amount of space:

Services	Basic	Medium 1	Medium 2	Premium
Number of clients	4	3	2	1

Table 1 . Clients per service

We assume that a sufficient number of disks is available, specifically **10 disks**. This is derived from the worst-case scenario where, theoretically, each client's storage requirements would be allocated to its own disk without any optimization. Given that each disk can store up to **25TB**, the goal is to satisfy all clients' needs while minimizing the number of disks used.

Tasks

1. Provide an example of a feasible solution for the disk packing problem based on the case described above. You may also include an illustration of the solution if you find it helpful.
2. Formulate the problem as a Mixed Integer Linear Program (MILP). Clearly explain the role of each part: data, decision variables, the objective function, and the constraints. Some help: we have seen this…in the previous class…with bins instead of disks…just pick-up and adapt !

[1] We have deliberately reduced the number of clients so that you can complete this lab exercise!

3. Determine the lower bound of the problem by solving its continuous relaxation.
4. Solve the MILP and compare results with the continuous relaxation.
5. **New Scenario**: Imagine a situation where a single client's data can be split across multiple disks. Modify the MILP to reflect this change, generate the corresponding .lp file, and solve it.
6. Report all the .lp files and .sol files in the appendix, and for each solution, express the reality of the variables values.

To continue optimizing the company's services, we will take into account the result obtained in point 2, where the data is indivisible.

DATA & TASKS (P2)

Context

Perhaps you didn't know yet, but you're no longer so young because it's now 2050 🥵, in Lille, one year after solving your first problem. From now on, any new heat-emitting infrastructure must be buried. The goal of this second problem is to determine the locations where the data centers should be placed to house the storage disks, the number of which has already been optimized.

For each option, the costs often differ. For example, the cost of inserting a disk into a data center usually has its own specific price. This corresponds to the total cost of connecting the data center containing the disk to all of its users. Unfortunately, it's no longer possible to go back, which means that clients can no longer be grouped based on storage costs. Here, all clients are associated according to the disks in which they were integrated in a solution of P1. Additionally, a fixed cost must be paid for each data center that receives one or more disks.

Therefore, the task is to locate the data centers in such a way that all disks are placed exactly once while minimizing the total cost.

Case to be addressed

Since P1 was just a warm-up, **we assume that you found the optimal solution**, which we now provide for P2.

Only three disks are necessary, and we will try to determine the best locations to install them in order to minimize the total cost. We group all the clients into three groups (related to the 3 disks), which we denote as ***d_1*, *d_2*, *d_3***. These disks will be installed in Data Centers, and each Data Center does NOT have a maximum limit on the number of disks that can be installed[2].

There are two types of costs:

- We have 6 possible data centers, each with its own **fixed cost**[3] (***f_j*** where ***j*** represents the index of the data center ***dc_j***).
- For each possible pair (***d***, possible data center ***dc_j***) there is a **dedicated cost**[4] (***c_{ij}*** where ***i*** and ***j*** represent the disk ***d_i*** and the data center ***dc_j***, respectively).

Figure 2 shows an example of dedicated cost and fixed cost for a case with only three data centers and two disks[5]. Figure 3 presents two solutions: the first with the selection of two data centers to host one disk each, and another solution with only one data center selected to host both disks. You can see the corresponding total cost at the bottom of each representation.

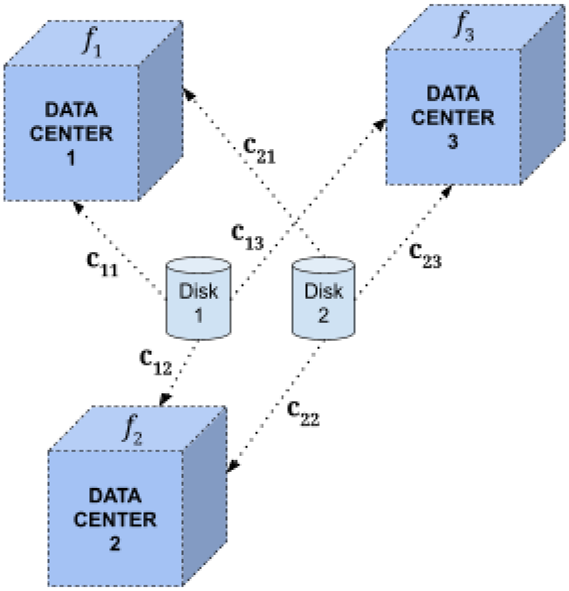


Figure 2. Representation of fixed and dedicated costs for 3 Data Centers and 2 Disks.

[2] For instance an optimal solution could place all the disks in the same data center.
[3] This cost is only paid once if at least one disk is installed in it.
[4] A dedicated cost is paid for each selected couple Disk<->data center
[5] This is not the case you have to treat.

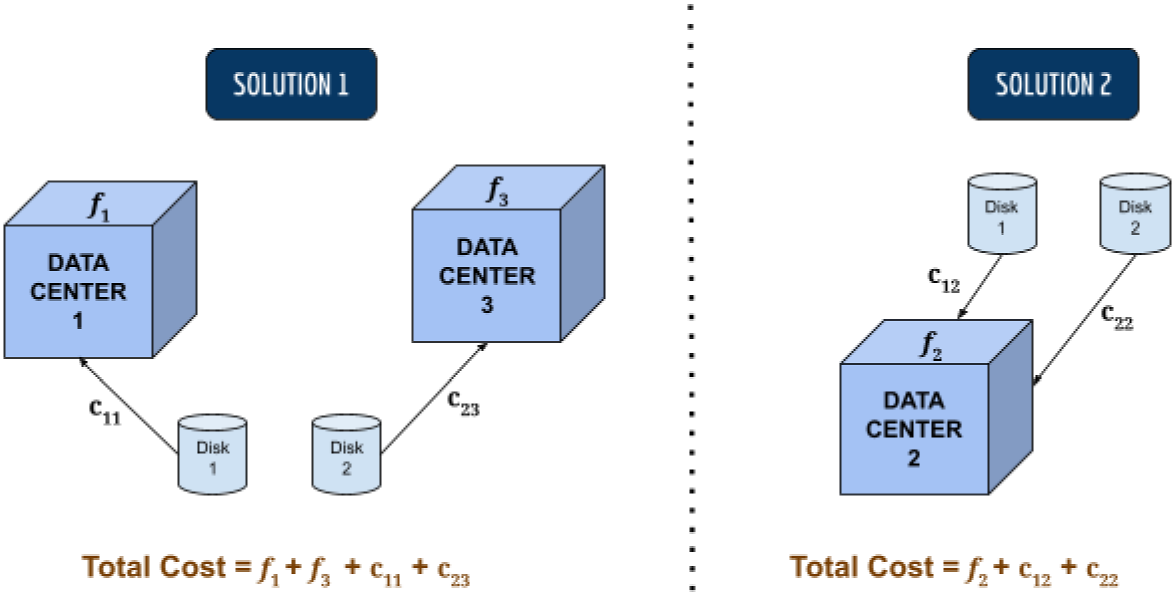


Figure 3. Representation of 2 different solutions of the case of Figure 2.

The following two tables provide the fixed and the dedicated costs, respectively.

$dc_j \rightarrow$	dc_1	dc_2	dc_3	dc_4	dc_5	dc_6
$f_j \rightarrow$	25	13	15	20	22	17

Table 2 - Fixed costs f_j for each data center dc_j to open (i.e. if at least one disk is in it)

c_{ij} ↘	dc_1	dc_2	dc_3	dc_4	dc_5	dc_6
d_1	14	28	13	20	30	31
d_2	19	20	36	25	35	12
d_3	18	16	16	22	18	20

Table 3 - Dedicated costs c_{ij} form each group of disk d_i and each data center dc_j

The objective function is the following:

$$\min_{x,y} \quad \sum_i \sum_j c_{ij} x_{ij} + \sum_j f_j y_j$$

with two vectors of binary variables (with the index *i* related to the disk and *j* the data center):

$$x_{ij}, y_j \in \{0,1\}$$

If x_{ij} takes the value 1, disk *i* is located in data center *j*, 0 otherwise. If y_j takes the value 1, the data center *j* is opened, 0 otherwise.

The two series of constraints are as follows: the first set must be satisfied **for any *j***, and the second set must be satisfied **for any pair** *(i,j)*:

$$\sum_j x_{ij} = 1$$

$$y_j \geq x_{ij}$$

Tasks

1. Explain in detail how each series of constraints works, then rewrite the complete generic model.
2. Formulate the continuous relaxation of the case to be addressed (Tables 2 and 3) in an .lp file, solve it, and express the reality of the solution you obtained.
3. Solve the original model and compare results with the continuous relaxation.
4. Illustrate the optimal solution.
5. Report all the .lp files and .sol files in the appendix.

DATA & TASKS (P3)

Context

Since we know where each client will save their data, and which data center is taking the corresponding disk, we will finish by a last problem related to the optimization of the critical updates of the data center infrastructure. In short, since a security problem happens, we want to be able to update the firmware of the disks (or of the systems managing them) as soon as possible. This security update is launched from one source S and has to be broadcast to all the opened (i.e., from the solution of P2) data centers.

Here, we consider 'cost' as the time it takes to travel from one point to another, and an optimal solution involves knowing the entire path from the source of the security update **to each** open data center.

2 cases to be addressed

Case 1. Let's start with a small exercise based on Figure 4, where the three types of equipment are represented: the computer from which the update is initiated, the data centers, and the routers.

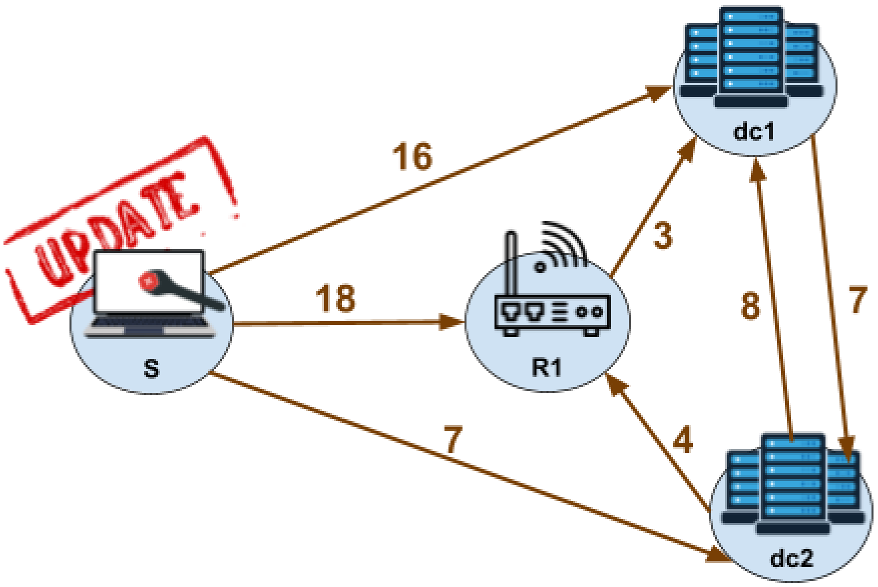


Figure 4. Example of a data center update deployment. The update originates from a node labeled S and must be deployed as quickly as possible to Data Center 1 (dc1) and Data Center 2 (dc2). The only available router (R1) can be used if it helps speed up the update to one or both data centers. The values on the arcs between the nodes of the graph represent the transition **duration** of data for each update.

This case has to be addressed according to the first question in the task section below.

Case 2. Let's now return to the year 2050, after P2 has been resolved, where P3 involves determining the most efficient paths for updating the data centers. The graph in Figure 5 represents the scenario with all the data centers that were available before their selection. We can see the source S, from which the updates are initiated, the routers R, and the data centers dc.

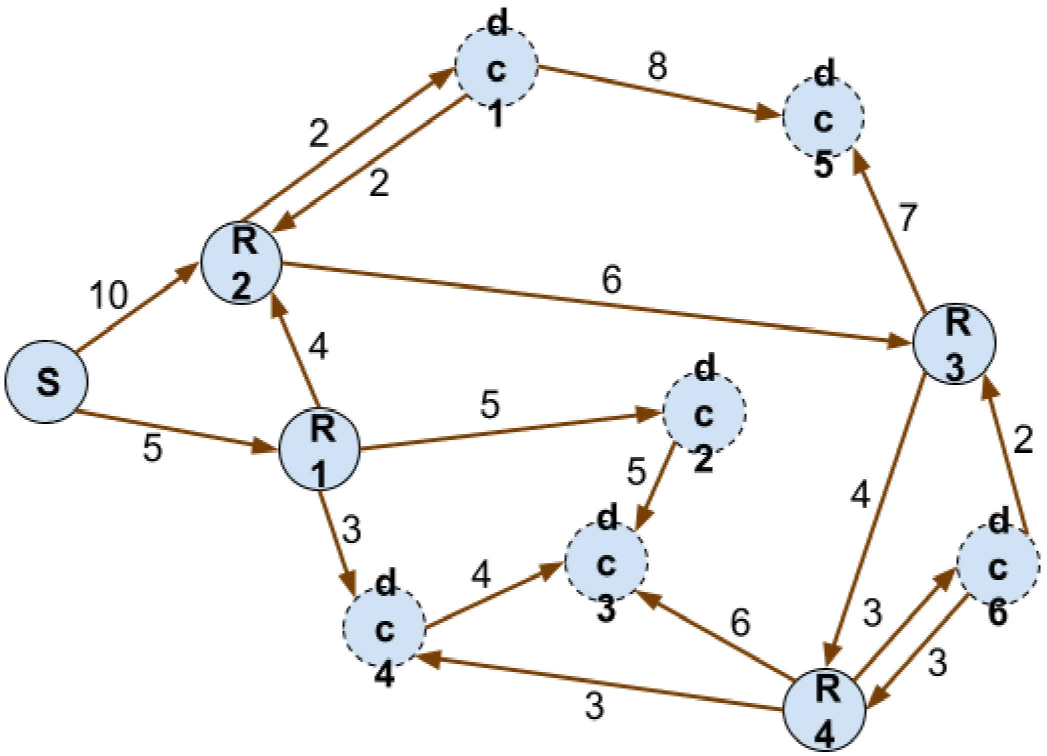


Figure 5.. Network for dispatching an update from the source S to the possible data centers. 4 routers R1, R2, R3 and R4 can take part of the shortest paths from S to all the selected data centers (P2). Based on the data centers (dc) selected from the solution of the previous problem, you will obtain a reduced graph by removing the nodes corresponding to the data centers that were not selected.

Tasks

1. Using *Case 1* of the Figure 4, provide by hand (without modeling and solver calls) the optimal solution, that is, the path**s** leading **to each** data center, starting from node S, which maximize the update speed.
2. Formulate the *Case 2* problem into a continuous relaxation through .lp file and solve it.
3. Convert the problem of *Case 2* to a MILP into a second .lp file and solve it.
4. Report all the .lp files in the appendix and for each solution, express the reality of the variables values.

DISCUSSION

It's clear that you hardly needed a computer with a solver to tackle these three problems, given the time available during the course. The scale of these problems does not lead to a combinatorial explosion of possibilities, nor does it involve a significant number of constraints that would filter them. That said, all the work you've done can easily be scaled up, where your mathematical models treated by solvers can make such a system efficient.

We aimed to have you work on reference problems, most of which were covered in class, but not directly presented in their original form in this lab. In the future, always try to identify such known problems in situations where you need to optimize a system, so you don't reinvent the wheel every time.

Finally, the reference problems discussed in class often have multiple ways to be modeled, even though only one is typically presented. To determine which is best for a given situation, it is usually necessary to either test them or consult the benchmarks found in scientific literature.

A large number of problems can be modeled as a bin packing problem. pic: chatGPT 4o

BONUS

B1a - You may have already done this, but if you have been using an online .lp file solver up to this point, now try one of the solvers available (either open-source or with an academic license) that can be installed on your machine.
B1b - Learn how to use APIs that, unlike .lp files, allow you to directly declare variables, objective functions, and constraints in code, as well as execute the resolution process.

B2a - Back in the 2000s, problems P1, P2, and P3 were too interdependent to be solved separately. Try to model and solve them with a solver by merging the problems according to one of these pairs or trio:

- P1 ∪ P2 then P3
- P1 then P2 ∪ P3
- P1 ∪ P2 ∪ P3

B2b - If merging the problems causes too many difficulties for the solver (e.g., with P1 ∪ P2 ∪ P3), try using only continuous relaxation, and then consider an algorithmic approach that seeks a solution with the correct variable type based on the continuous relaxation.

B3 - Imagine a situation related to P1 where you are trying to maximize your company's revenue by considering the four services offered. Here, the variables correspond to the prices set for each service. Obviously, you need to find a way to express constraints that reflect the economic model of supply and demand. This bonus, which is quite challenging, should lead you to explore what are called Pricing Problems in the scientific literature, which are aptly named. Try to write such a model as a mathematical program using the same notations as in this work, and then attempt to solve it.

B4 - P3 is a shortest path problem between a source and a set of destination nodes. You may have seen some fast algorithms in the past that can solve this problem. For example, Dijkstra's algorithm finds the shortest paths from a source to all other nodes in the graph in polynomial time (make sure to research, understand, and explain what this means in your report). For this bonus challenge, you will make things more complicated, which will justify the use of a mathematical program. You will imagine capacities for the different arcs of the graph, assuming that the graph already provided in this document represents the time it takes for data to pass through when the amount of data equals the arc capacities. You will then need to set the amount of data to be transferred for the update. This amount must reach all data centers while respecting the arc capacities. The goal, once again, is to minimize the total time for all updates.
This bonus is difficult (too difficult?), but it becomes interesting when you work on the generic model (objective: how to integrate capacities?). If you have time left, you can try to find capacity values and data amounts that represent a realistic and balanced situation, and then solve it with a solver. If you manage to do all this, you will automatically receive the maximum grade for this lab.

WORK TO BE DONE

(Tasks Recall)

1. Compile the results from all three problems into a comprehensive **report**. It includes:
   a. the .lps and .sols files in appendix
   b. screenshot of the results given by the solvers
   c. expression of these results in the “real” context given in this lab
   d. basically what is already asked in the *Tasks* sections of the three problems
2. Imagine another application using one of the three models you used in this lab which is related to your speciality (AI, big data, or cybersecurity).
3. What recommendations would you make for future research or practical applications based on your speciality? You can propose a problem that can be interesting to treat in the OR Class by the student of the next year (thanks for them 🙂 and for you 🙂).

Bonuses live up to their name; they only improve the grade if the rest isn't perfect and therefore doesn't deserve the maximum score. A satisfactory work involves correctly solving the first two problems, while a good work also includes significant progress on solving problem P3.

This work must be submitted via JuniaLearning by October 24, 2024.