Rebars Steel Company Project

1 Introduction

The course project, which arises from a company case, focuses on managing the real-life complexity of production and distribution planning. We have preprocessed the data for this course. You can find the data and the problem description in this document.

2 Case Description

The company produces steel bars in three factories located at IJmuiden, Segal, and South Wales, and have to be transported to the customers in the Germany Ruhr area on a given planning horizon consisting of different periods. In particular, the company produces reinforcing bars, which we refer to as *rebars*, with a diameter equal to 57 millimetres (mm) and different lengths: 2.40 meters (rebar type A), 3.60 meters (rebar type B), and 4.20 meters (rebar type C).

The rebars are obtained by cutting longer bars, which we refer to as *long bars*, at the factories. Long bars are produced at the factories in two different lengths: 9 meters (long bar type 1), and 12 meters (long bar type 2). All the long bars have a diameter equal to 57 millimetres (mm).

The long bars are produced using a type of steel with a density equal to 7.85 tonnes per cubic meter (t/m^3). Each factory has a limited long bars production capacity (per period) expressed in tonnes of steel and reported in Table 1.

Table 1: Production capacity of each factory per time period

Factory	IJmuiden	Segal	South Wales
Production Capacity (in tonnes)	12	10	28

The cost to produce long bars at the facilities consists of a fixed and a variable cost. The fixed cost is paid in every period in which the factory is used to produce at least one long bar. The unit cost is paid for each long bar produced and it is assumed that it does not depend on the long bar type (i.e., the unit cost to produce long bar type 1 and 2 are identical). The fixed and variable costs for each factory are reported in Table 2.

Table 2: Production cost of each factory

Factory	IJmuiden	Segal	South Wales
Fixed Cost	400	500	250
Unit Cost (€ per long bar)	5	3	2

Long bars that are produced in a certain period, can be cut to obtain rebars in the *same* period. Long bars produced and not used in a certain period, cannot be stored and cut in the next periods (i.e., it is not possible to store inventory at the factories). Moreover, long bars cannot be transported from one factory to another: long bars produced in a certain factory can be cut only in the same factory. The same applies to rebars: rebars produced in a certain factory, cannot be shipped to other factories. They can only be shipped directly to customers to satisfy their demands.

The customer demands are aggregated based on geographical positions into eight different areas: Bochum, Boenen, Dortmund, Gelsenkirchen, Hagen, Iserlohn, Neuss, and Schwerte. Tables 3, 4 and 5 report the demand, of each region and in each period, for rebars A, B, and C, respectively.

Table 3: Customers demand for rebar A in each period (in number of rebars)

Period	Bochum	Boenen	Dortmund	Gelsenkirchen	Hagen	Iserlohn	Neuss	Schwerte
1	2	4	2	5	19	13	20	4
2	6	8	7	5	23	19	16	5
3	5	5	6	5	25	17	14	3
4	3	10	5	5	16	14	26	4

Table 4: Customers demand for rebar B in each period (in number of rebars)

Period	Bochum	Boenen	Dortmund	Gelsenkirchen	Hagen	Iserlohn	Neuss	Schwerte
1	4	5	4	9	15	22	12	2
2	5	8	5	10	33	26	23	8
3	7	12	8	6	31	20	30	2
4	8	13	10	6	33	27	30	6

Table 5: Customers demand for rebar C in each period (in number of rebars)

Period	Bochum	Boenen	Dortmund	Gelsenkirchen	Hagen	Iserlohn	Neuss	Schwerte
1	6	6	7	10	12	14	22	5
2	7	10	6	9	35	25	32	6
3	7	15	4	9	33	23	31	7
4	7	12	12	10	38	24	31	2

Deliveries to customers are outsourced to an external logistic company. The logistics company operates by shipping directly from factories to customers. In other words, a shipment is a direct delivery from a factory to a customer. Multiple shipments can depart from each factory, but each customer can be served by only one factory in each period, i.e., it can receive a single shipment per period.

The logistic company applies a pricing policy consisting of a fixed cost per shipment and variable transportation costs.

The fixed shipping cost must be paid for each delivery from a factory to a customer (e.g., if a factory serves two customers in one period, the fixed shipping cost must be paid twice in that period). The fixed costs per shipment from each factory are reported in Table 6.

Table 6: Fixed shipping costs

Factory	IJmuiden	Segal	South Wales
Fixed Cost per Shipment (€)	130	150	100

The variable transportation costs amount to € 0.50 per kilometre per tonne, i.e., they depend on the distance and the quantity transported. The distances (in kilometres) between factories and customers are reported in Table 7.

Table 7: Distances between customers and factories (in kilometres)

	IJmuiden	Segal	South Wales
Bochum	250	203	866
Boenen	282	242	914
Dortmund	266	222	885
Gelsenkirchen	234	198	859
Hagen	289	206	903
Iserlohn	299	226	913
Neuss	259	140	843
Schwerte	279	216	901

3 Report

Address each of the following questions in a separate section of the report:

1. Formulate an optimization model to determine an optimal production and delivery plan for meeting the customer demands every period at the minimum total cost. In particular:

   a. Provide the network representation of the problem and some examples of cutting patterns.

   b. Describe the mathematical programming model precisely and in detail. Clearly define the sets, parameters, decision variables, objective functions, and constraints.

   c. Implement the mathematical model developed to answer question 1b in Python using Pyomo and solve it using Gurobi.

   d. Determine and report an optimal plan for each period.

2. Suppose that each customer and each factory has a dedicated warehouse that can be used to store bars. Factories can store long bars and rebars. Stored long bars can be cut in the next periods to produce rebars and stored rebars can be shipped to customers in the next periods. Warehouses at customers can store rebars that can be used to satisfy the demand for the next periods. As in Part 1, there is no possibility to transfer inventory from one customer to another and from one factory to another.

   The total inventory capacity of each warehouse (expressed in tonnes) is reported in Table 8. Extend the model developed in Part 1 to use the warehouses for improving the production and distribution plan. Assume that the starting inventory level at the beginning of the planning horizon is equal to zero for each customer and factory.

   As in Part 1, each customer can be supplied by at most one factory in each period.

   a. Provide the revised mathematical programming model. Clearly define the sets, parameters, decision variables, objective functions, and constraints. Describe the model in detail.

   b. Implement the mathematical model developed to answer question 1a in Python using Pyomo and solve it using Gurobi.

   c. Report the new distribution plan and compare the solutions obtained with the first mathematical model developed in Part 1 and the solution obtained with this revised model. Is there any benefit in using the warehouses?

3. The rebars company is now evaluating the possibility of investing in its fleet of vehicles and using them to deliver rebars to customers, instead of relying on the external logistic company. Assume now that at each factory one vehicle with a maximum transportation capacity of 25 tonnes is available for

Table 8: Inventory capacity for each warehouse (in tonnes)

Warehouse	Max Inventory
Ijmuiden	30
Segal	15
South Wales	60
Bochum	10
Boenen	7
Dortmund	12
Gelsenkirchen	10
Hagen	12
Iserlohn	9
Neuss	8
Schwerte	5

deliveries. In each period, each vehicle can perform at most one tour, which must start and end at the same factory, to deliver rebars to customers. A vehicle can serve multiple customers in the same tour, but, as in Parts 1 and 2, each customer can be served by at most one factory per period (i.e., it can be served by a single vehicle per period).

The transportation cost for using the fleet is equal to 3 € per kilometre (i.e., it depends on the travelled distance). The distances between locations are reported in Table 9.

Table 9: Distances between locations (in kilometres)

	IJmuiden	Segal	South Wales	Bochum	Boenen	Dortmund	Gelsenkirchen	Hagen	Iserlohn	Neuss	Schwerte
IJmuiden	0	284	826	250	282	266	234	289	299	259	279
Segal	284	0	750	203	242	222	198	206	226	140	216
South Wales	826	750	0	866	914	885	859	903	913	843	901
Bochum	250	203	866	0	55	21	19	41	51	56	39
Boenen	282	242	914	55	0	34	56	44	54	102	32
Dortmund	266	222	885	21	34	0	34	21	36	75	15
Gelsenkirchen	234	198	859	19	56	34	0	56	66	62	54
Hagen	289	206	903	41	44	21	56	0	20	66	14
Iserlohn	299	226	913	51	54	36	66	20	0	85	15
Neuss	259	140	843	56	102	75	62	66	85	0	75
Schwerte	279	216	901	39	32	15	54	14	15	75	0

Define a version of the model developed in Part 2 in which, instead of using the external logistics provider, the company performs shipments using its fleet. In particular:

a. Provide the new mathematical programming model. Clearly define the sets, parameters, decision variables, objective functions, and constraints.

b. Implement and solve the model (developed in question a) for the optimal distribution plan over the planning horizon using Python, Pyomo, and Gurobi.

c. Report the new distribution plan and compare the solutions obtained with the mathematical models developed in Parts 1 and 2 and the solution obtained with this revised model. How much can the company benefit from performing deliveries using its fleet?

4. Perform a sensitivity analysis on a (single) input parameter of your choice and discuss the impact on the solution and the objective function. Ideally, you should use the model developed in Part 3. If that is not possible (e.g., because you were not able to define it, because you have a bug in the code or because it takes too long to be solved), you can use the model developed in Part 2. Remember to discuss the findings of your sensitivity analysis and highlight the difference you notice (if any) in terms of the optimal solution, objective function, and computational complexity (e.g., is the problem solved faster with certain parameter settings?).

4 Notes and Regulations

- Given the complexity of the project, you have to work in teams of four students.
- Each team has to hand in a 6-page report (excluding the cover page and including any appendices) by Sunday, March 17 (end of the day). Late submissions will not be accepted.
- Each team has to hand in an **additional pdf document**, therefore it should be a separate file from the report, documenting the usage of AI (e.g., ChatGPT) in the assignment. In this document, you have to describe for which part of the assignment you used AI tools and how you used them. For example, if you use ChatGPT, you should include the links to the chats regarding the assignment.
- If the report submitted exceeds the 6-page limit, only the first 6 pages will be graded.
- The report has to be a `pdf` file. If the report is not a `pdf` file, it will not be graded.
- The models have to be **integer linear** or **mixed-integer linear** programming models if possible. Providing a non-linear model whereas a linear model is possible is counted as a mistake.
- All the Jupyter Notebooks files you refer to in the report have to be submitted via Canvas. Make sure that the same answers documented in the report can be easily reproduced using the submitted files.
- Each Jupyter Notebook has to be properly explained with comments.
- In the report, the answer to each question has to be in a separate section.
- If one of your models cannot find the optimal solution quickly, you can set a maximum running time (in seconds) for the solver: in this way, the solver will provide the best solution found within the time limit. Remember to reflect on the optimality gap of the solution. The following lines of code will set 3600 seconds as the maximum running time (you can of course give more time to the solver, based on the optimality gap you achieve):
```
solver = po.SolverFactory('gurobi')
result = solver.solve(model, tee = True, options={'TimeLimit':3600})
```
  Another option is to set a higher *optimality gap*. The following lines of code will set the relative *optimality gap* to 5%.
```
result = solver.solve(model, tee = True, options='MIPGap':0.05)
```
  This means that the solver will stop as soon it can certify that the current best solution is, in the worst case, 5% away from the optimal solution. The smaller the optimality gap the longer it can take for the solver to terminate. However, only solutions with an optimality gap equal to 0 can be claimed as optimal.
- The grade will depend on the correctness and consistency of your mathematical models and implementation, the correctness and the reflections on the solutions you provide, and the presentation of the models and the results in the report.