Klaudiusz Kozłowski

Krakow, 01/2025

Final project from the subject:

Modeling and Optimization
with General Algebraic Modeling System

Topic:

Factory order selection system

created with the use of:

Table of Contents:

General description

This program is a simple ERP system that supports the management of production processes in the enterprise. Its main task is to optimize production planning by allocating orders to production lines, taking into account available resources, time and costs. The system calculates the required quantities of materials based on the composition of products, schedules production processes and maximizes profit, taking into account such factors as employee costs, machine configuration and material consumption. It enables effective order management, analyzing their profitability and lead time, which allows for making informed production decisions.

Code overview

This section discusses the subsequent lines of code presented in the "Code" section.

Sets

Four sets were defined, representing 8 products, 3 production lines in the factory, 5 orders with a defined number of required products and 10 materials from which the products are manufactured.

Parameters

The next step is to load tabular data from excel files. The Orders.xlsx file contains a table where the required quantities of products for each order are defined.

Order/Quantity	1	2	3	4	5	6	7	8
1	10000	0	800	2100	0	0	0	0
2	0	12500	0	0	300	900	0	0
3	500	0	0	8000	0	0	600	0
4	0	400	15000	0	0	0	0	2500
5	0	0	0	100	0	9000	1500	0

The second imported table (Ingredients.xlsx) contains the quantities of materials that are used to produce the products.

Material/Product	1	2	3	4	5	6	7	8
1	0,4	0,5	0	0	0	0,5	0	0
2	0,3	0	0,3	0	0	0	0,4	0
3	0,3	0	0,4	0,1	0	0	0	0
4	0	0,5	0	0,4	0,4	0	0	0,3
5	0	0	0	0	0,6	0	0	0
6	0	0	0,3	0	0	0,3	0	0,5
7	0	0	0	0	0	0	0,3	0
8	0	0	0	0	0	0	0	0,2
9	0	0	0	0,3	0	0	0	0
10	0	0	0	0,2	0	0,2	0	0,2

The next lines of code define the remaining parameters. SetupTime(i) defines the changeover time in hours needed to prepare the production line for manufacturing product i, which affects production scheduling. ProductionRate(i) is the production efficiency of product i expressed in kilograms per hour, determining the pace of order fulfillment. EmployeeCostPerHour represents the cost of employee labor per hour, and FixedCostPerHour the fixed operating costs per hour, both of which affect the total production costs. MaterialCost(m) provides the unit cost of material m in euros per kilogram, which is important when calculating raw material costs. SellingPrice(i) is the selling price of one kilogram of product i, which is crucial for calculating revenues and production profitability.

The TimeLimit is responsible for limiting the available production time.

Variables

The model contains four variables. The most important one, which is maximized in the model, is the total cost. Variable y is responsible for assigning the product to the line and variable z informs whether the order has been produced. The last variable x describes how many and which products from which order were produced on each line.

Equations

Objective: This equation defines an objective function, the goal of which is to maximize total profit. It calculates the sales revenue from fulfilled orders, then subtracts the material costs, the labor costs associated with the changeover of production lines, and the overhead and employee costs resulting from production time.

TimeLimitPerLine(l): This constraint ensures that the total time spent on production on each line l does not exceed the specified time limit TimeLimit. The sum of the production time (the quantity of the product produced divided by its capacity) and the changeover time for all products on a given line must be less than or equal to the available time.

OrderCompletion(o,i): This equation guarantees that if an order o for product i is fulfilled (i.e. z(o) = 1), then the sum of the quantities produced of this product on all production lines will be equal to the ordered quantity orderQuantity(o,i). If the order is not fulfilled (z(o) = 0), the production of this product is zero.

LineUsage(o,i,l): This constraint ensures that the production of product i for order o on line l is only possible if line l is assigned to the production of this product (y(i,l) = 1). In other words, if a line does not produce a given product, then the production of this product assigned to it must be zero.

Consistency(o): This equation ensures consistency in the selection of orders to be filled. If order o is filled (z(o) = 1), then at least one production line must be assigned to the production of at least one product contained in this order. If no product from the order is assigned to any line, the order cannot be filled (z(o) = 0).

Examples

Results for data from the "Code" section

```
----     95 VARIABLE x.L  Quantity of product i for order o on line l

              1           2           3

2.2                              12500.000
2.5      300.000
2.6      900.000
3.1                  500.000
3.4                                8000.000
3.7                  600.000
4.2                                 400.000
4.3                15000.000
4.8     2500.000
5.4                                 100.000
5.6     9000.000
5.7                 1500.000


----     95 VARIABLE y.L  Is product produced on line

            1           2           3

1                    1.000
2                                1.000
3                    1.000
4                                1.000
5        1.000
6        1.000
7                    1.000
8        1.000


----     95 VARIABLE z.L  Order selected

2 1.000,    3 1.000,    4 1.000,    5 1.000


----     95 VARIABLE TotalProfit.L         =   173626.757  Total profit
```

The production distribution on the lines was checked to verify the correct operation of the program.

Line 1				
Prod	Quantity	Rate	Changeover	Total Time
5	300	160	6	7,88
6	9900	205	3	51,29
8	2500	135	5	23,52
Sum:				82,69

Line 2				
Prod	Quantity	Rate	Changeover	Total Time
1	500	280	4	5,79
7	2100	215	4	13,77
3	15000	200	5	80,00
Sum:				99,55

Line 3				
Prod	Quantity	Rate	Changeover	Total Time
2	12900	250	3	54,60
4	8100	240	2	35,75
Sum:				90,35

The lines are fairly evenly loaded. Each product is manufactured on one line, which allows you to avoid additional changeover, which saves time and increases income. Order 1 cannot be produced during the planned time.

Employee Cost:	1600,00
Fixed Cost:	96235,74

Product	Cost [EUR/kg]	All orders
1	4,82	2410
2	5,71	73659
3	4,28	64200
4	5,53	44793
5	4,63	1389
6	4,88	48312
7	4,69	9849
8	4,88	12200
Production Cost:		256812

Product	Cost [EUR/kg]	All orders
1	8	4000
2	9,5	122550
3	11,2	168000
4	11,3	91530
5	11,15	3345
6	9	89100
7	9,5	19950
8	12	30000
Sell:		528475

Sell	528475,00
Total Cost	354647,74
Income	173827,26

Revenues and costs were calculated for the number of orders given in the solution. The result is very similar to that obtained in GAMS.

Results with unprofitable order

```
----     96 VARIABLE x.L  Quantity of product i for order o on line l

             2

1.1   10000.000
1.3     800.000
1.4    2100.000
3.1     500.000
3.4    8000.000
3.7     600.000
4.2     400.000
4.3   15000.000
4.8    2500.000
5.4     100.000
5.6    9000.000
5.7    1500.000


----     96 VARIABLE y.L  Is product produced on line

            2

1       1.000
2       1.000
3       1.000
4       1.000
6       1.000
7       1.000
8       1.000


----     96 VARIABLE z.L  Order selected

1 1.000,    3 1.000,    4 1.000,    5 1.000


----     96 VARIABLE TotalProfit.L         =   165532.140  Total profit
```

TimeLimit was set to 1000 to allow time to complete all orders. Then the selling price of product 2 was reduced to 1.50 EUR, which means that it is unprofitable to produce. The simulation result shows that order 2, which is mainly based on the production of this item, was not taken. Due to the considerable time, the solution did not require dividing the production into multiple stations. This confirms that the model works correctly.

Summary

The program was successfully executed and no errors were detected in its operation. Calculations, including material costs and product compositions, were performed accurately and adapted to the data provided. The current implementation is modular and allows for further development and integration of additional functionalities.

Code

```
Sets
    i 'Products' /1*8/
    l 'Production lines' /1*3/
    o 'Orders' /1*5/
    m 'Materials' /1*10/;

$ontext
OrderQuantity(o,i) 'Order quantities in kg'
        1       2       3       4       5       6       7       8
    1   10000   0       800     0       0       0       0       0
    2   0       12500   0       0       300     900     0       0
    3   500     0       0       8000    0       0       600     0
    4   0       400     15000   0       0       0       0       2500
    5   0       0       0       100     0       9000    1500    0;

ProductComposition(i,m) 'Material composition of each product'
    /1.1 0.4, 1.2 0.3, 1.3 0.3,
     2.1 0.5, 2.4 0.5,
     3.6 0.3, 3.7 0.4, 3.2 0.3,
     4.3 0.1, 4.8 0.4, 4.9 0.3, 4.10 0.2,
     5.4 0.4, 5.6 0.4, 5.10 0.2,
     6.1 0.5, 6.7 0.5,
     7.9 0.2, 7.2 0.4, 7.5 0.25, 7.3 0.15,
     8.10 0.2, 8.4 0.3, 8.6 0.5/
$offText

Parameters
OrderQuantity(o, i) 'Orders table'
$call gdxxrw input=Orders.xlsx output=orders.gdx par=orderQuantity rng=Orders!A1:I6 rdim=1
cdim=1
$gdxin orders.gdx
$load orderQuantity

ProductComposition(i,m) 'Material composition of each product';
$call gdxxrw input=Ingredients.xlsx output=ingredients.gdx par=productComposition
rng=Composition!A1:I11 rdim=1 cdim=1
$gdxin ingredients.gdx
$load productComposition

Parameters
SetupTime(i) 'Changeover time for products in hours'
        /1 4, 2 3, 3 5, 4 2, 5 6, 6 3, 7 4, 8 5/
ProductionRate(i) 'Production rate in kg/hour'
        /1 280, 2 250, 3 200, 4 240, 5 160, 6 205, 7 215, 8 135/
EmployeeCostPerHour
        /50/
FixedCostPerHour
        /350/
MaterialCost(m) 'Material cost per kg'
        /1 4.15, 2 3.25, 3 7.20, 4 5.50, 5 4.30, 6 3.20, 7 5.60, 8 4.00, 9 6.20, 10 5.75/

SellingPrice(i) 'Selling price per kilogram of product i'
        /1 8.00, 2 9.50, 3 11.20, 4 11.30, 5 11.15, 6 9.00, 7 9.50, 8 12.00/;

Scalar
    TimeLimit /100/;
```

```
Variables
    TotalProfit 'Total profit';

Binary Variables
    y(i,l) 'Is product produced on line'
    z(o) 'Order selected';

Positive Variables
    x(o,i,l) 'Quantity of product i for order o on line l';

Equations
    Objective 'Maximize total profit'
    TimeLimitPerLine(l) 'Ensure limit of production speed'
    OrderCompletion(o,i) 'Ensure order completion'
    LineUsage(o,i,l) 'Link production to line usage'
    Consistency(o) 'Consistency in order selection';

Objective ..
    TotalProfit =e=
        sum(o, z(o) * sum(i, OrderQuantity(o,i) * SellingPrice(i))) -
        sum((o,i,m,l), x(o,i,l) * ProductComposition(i,m) * MaterialCost(m)) -
        sum((i,l), y(i,l) * SetupTime(i) * EmployeeCostPerHour) -
        sum((o,i,l), (x(o,i,l) / ProductionRate(i)) * (FixedCostPerHour +
EmployeeCostPerHour));

TimeLimitPerLine(l) ..
    sum((o,i), x(o,i,l) / ProductionRate(i)) + sum(i, y(i,l) * SetupTime(i)) =l= TimeLimit;

OrderCompletion(o,i) ..
    sum(l, x(o,i,l)) =e= OrderQuantity(o,i) * z(o);

LineUsage(o,i,l) ..
    x(o,i,l) =l= OrderQuantity(o,i) * y(i,l);

Consistency(o) ..
    z(o) =l= sum((i,l)$(OrderQuantity(o,i) > 0), y(i,l));

Model MaximizeProfit /all/;
Solve MaximizeProfit maximizing TotalProfit using MIP;

Display x.l, y.l, z.l, TotalProfit.l;
```