Formulation of the Truck Packing Problem (Bin Packing Problem)

It is required to divide the proposed array of roll weights into trucks of two types (22.2 and 27.6, these are maximum allowable loads of trucks that cannot be exceeded), so that the specific load is maximum and all rolls are distributed among the trucks. It is considered that there are unlimited number of trucks of both types.

The **specific load** is the ratio of the total weight of all rolls to the maximum allowable load of all used trucks.

We introduce the following notations:

n – number of rolls;
m – number of available trucks;
$W = \{w_1, \dots, w_n\}$ – weights of trucks;
$C = \{c_1, \dots, c_m\}$ – maximum allowable load of trucks, $c_i \in \{22.2, 27.6\}$, $i = \overline{1, m}$.

We define the variables:

$x_{i,j}$ – binary variable that is set to 1 if roll i is loaded into track j and 0 otherwise;
y_j – binary variable that is set to 1 if machine j is loaded with at least one roll, and 0 otherwise.

The binary linear programming model for this problem is presented below:

Minimize

$$\sum_{j=1}^{m} c_j \cdot y_j \tag{1}$$

Subject to:

$$\sum_{j=1}^{m} x_{i,j} = 1, \quad i = \overline{1, n}; \tag{2}$$

$$\sum_{i=1}^{n} w_i \cdot x_{i,j} \le c_j \cdot y_j, \quad j = \overline{1, m}; \tag{3}$$

$$x_{i,j}, y_j \in \{0,1\}, \quad j = \overline{1, n}, j = \overline{1, m}; \tag{4}$$

The objective function (1) maximizes the specific load. In the formulation of the problem, it is required to maximize the objective function $\frac{\sum_{i=1}^{n} w_i}{\sum_{j=1}^{m} c_j \cdot y_j}$, but since the numerator of this function is a constant, it is enough to minimize the sum $\sum_{j=1}^{m} c_j \cdot y_j$.

Constraints (2) ensure that each roll is placed in exactly one track. Restrictions (3) ensure that the load of each used truck does not exceed its maximum allowable load. Constraints (4) establish the binary nature of all variables.

An exact approach for solving the Truck Packing Problem

The decision-making process consists of three steps:

1. The minimum number of trucks of the first type (minNumTrucksT1) is determined in the case when rolls are loaded only into first type trucks (22.2). To find the solution, the above model is used, in which the number of available trucks of the first type is equal to $m = \lceil n/\lfloor 22.2/max(W) \rfloor \rceil$, and the second type trucks are not used (27.6).
2. The minimum number of trucks of the second type (minNumTrucksT2) is determined in the case when rolls are loaded only into second type trucks (27.6). To find the solution, the above model is used, in which the number of available trucks of the second type is equal to $m = \lceil n/\lfloor 27.6/max(W) \rfloor \rceil$, and the first type trucks are not used (22.2).
3. The optimal solution of the initial problem is determined. In this case, the rolls can be loaded into trucks of the first and second types. To find a solution, the above model is used, in which the number of available trucks of the first and second types is equal to minNumTrucksT1 and minNumTrucksT2, respectively.

Input data

The input data is stored in "weights.xlsx" file. The weights of rolls are specified in the first column of the file (see Figure 1).

	A	B
1	**weight**	
2	8,78	
3	8,77	
4	8,77	
5	8,76	
6	8,74	
7	8,73	
8	8,72	
9	8,7	
10	8,63	
11	8,63	
12	8,62	
13	8,62	
14	8,62	
15	8,61	
16	8,6	
17	8,57	
18	8,57	
19	8,53	
20	8,53	
21	8,5	
22	8,49	
23	8,49	
24	8,47	
25	8,44	
26	8,42	
27	8,41	
28	8,39	
29	8,38	
30	8,37	
31	8,33	
32	8,31	
33	8,31	
34	8,28	
35	8,27	
36	8,27	
37	8,26	
38	8,2	
39	8,15	
40	8,12	
41	8,12	
42	7,85	
43	7,14	
44	7,13	
45	7,06	
46	6,31	
47	6,3	
48	6,03	
49	6,02	
50	5,81	
51	5,78	
52	5,76	
53	5,76	
54	5,75	
55	5,74	
56	5,64	
57	5,31	
58	5,31	
59		

Figure 1.

Implementation and result

Python 3.9 and Gurobi 10.0 were used to implement the program. The total computation time for input data from Figure 1 does not exceed 55 seconds on an AMD Ryzen 5 5600U processor with 10 threads.

The result of running a Python program using the Gurobi solver is presented below:

```
Set parameter MIPGap to value 0.001
Set parameter TimeLimit to value 150
Set parameter Threads to value 10
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: AMD Ryzen 5 5600U with Radeon Graphics, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 10 threads

Optimize a model with 86 rows, 1682 columns and 3335 nonzeros
Model fingerprint: 0x14d48695
Variable types: 0 continuous, 1682 integer (1682 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+01]
  Objective range  [2e+01, 2e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 621.6000000
Presolve time: 0.00s
Presolved: 86 rows, 1682 columns, 3335 nonzeros
Variable types: 0 continuous, 1682 integer (1682 binary)

Root relaxation: objective 4.441800e+02, 167 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  444.18000    0   39  621.60000  444.18000  28.5%     -    0s
H    0     0                     577.2000000  444.18000  23.0%     -    0s
H    0     0                     555.0000000  444.18000  20.0%     -    0s
H    0     0                     532.8000000  444.18000  16.6%     -    0s
     0     0  444.18000    0   49  532.80000  444.18000  16.6%     -    0s
H    0     0                     510.6000000  444.18000  13.0%     -    0s
     0     0  444.18000    0   59  510.60000  444.18000  13.0%     -    0s
     0     0  444.18000    0   39  510.60000  444.18000  13.0%     -    0s
     0     0  444.18000    0   38  510.60000  444.18000  13.0%     -    0s
     0     2  444.18000    0   38  510.60000  444.18000  13.0%     -    0s

Cutting planes:
  Gomory: 6
  Cover: 9
  Clique: 1
  MIR: 83
  StrongCG: 169
  Flow cover: 37
  Inf proof: 3

Explored 3207 nodes (110299 simplex iterations) in 4.60 seconds (4.14 work units)
Thread count was 10 (of 12 available processors)

Solution count 5: 510.6 532.8 555 ... 621.6

Optimal solution found (tolerance 1.00e-03)
Best objective 5.106000000000e+02, best bound 5.106000000000e+02, gap 0.0000%
```

```
Set parameter MIPGap to value 0.001
Set parameter TimeLimit to value 150
Set parameter Threads to value 10
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: AMD Ryzen 5 5600U with Radeon Graphics, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 10 threads

Optimize a model with 76 rows, 1102 columns and 2185 nonzeros
Model fingerprint: 0x9dcd435d
Variable types: 0 continuous, 1102 integer (1102 binary)
Coefficient statistics:
  Matrix range     [1e+00, 3e+01]
  Objective range  [3e+01, 3e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 524.4000000
Presolve time: 0.00s
Presolved: 76 rows, 1102 columns, 2185 nonzeros
Variable types: 0 continuous, 1102 integer (1102 binary)

Root relaxation: objective 4.441800e+02, 154 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  444.18000    0   23  524.40000  444.18000  15.3%     -    0s
H    0     0                     496.8000000  444.18000  10.6%     -    0s
     0     0  444.18000    0   50  496.80000  444.18000  10.6%     -    0s
     0     0  444.18000    0   52  496.80000  444.18000  10.6%     -    0s
     0     0  444.18000    0   41  496.80000  444.18000  10.6%     -    0s
     0     0  444.18000    0   31  496.80000  444.18000  10.6%     -    0s
     0     2  444.18000    0   28  496.80000  444.18000  10.6%     -    0s
H  117    69                     469.2000000  469.20000  0.00%  18.8    0s

Cutting planes:
  Cover: 44
  Clique: 3
  MIR: 28
  StrongCG: 9

Explored 128 nodes (4482 simplex iterations) in 0.45 seconds (0.36 work units)
Thread count was 10 (of 12 available processors)

Solution count 3: 469.2 496.8 524.4

Optimal solution found (tolerance 1.00e-03)
Best objective 4.692000000000e+02, best bound 4.692000000000e+02, gap 0.0000%
Set parameter MIPGap to value 0.001
Set parameter TimeLimit to value 150
Set parameter Threads to value 10
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: AMD Ryzen 5 5600U with Radeon Graphics, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 10 threads

Optimize a model with 97 rows, 2320 columns and 4600 nonzeros
Model fingerprint: 0x48635b5f
Variable types: 0 continuous, 2320 integer (2320 binary)
Coefficient statistics:
  Matrix range     [1e+00, 3e+01]
  Objective range  [2e+01, 3e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
```

```
Found heuristic solution: objective 775.2000000
Presolve time: 0.01s
Presolved: 97 rows, 2320 columns, 4600 nonzeros
Variable types: 0 continuous, 2320 integer (2320 binary)

Root relaxation: objective 4.441800e+02, 233 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  444.18000    0   29  775.20000  444.18000  42.7%     -    0s
H    0     0                     570.0000000  444.18000  22.1%     -    0s
H    0     0                     542.4000000  444.18000  18.1%     -    0s
     0     0  444.18000    0   48  542.40000  444.18000  18.1%     -    0s
H    0     0                     498.0000000  444.18000  10.8%     -    0s
H    0     0                     486.6000000  444.18000  8.72%     -    0s
H    0     0                     469.8000000  444.18000  5.45%     -    0s
     0     0  444.18000    0   44  469.80000  444.18000  5.45%     -    0s
     0     0  444.18000    0   38  469.80000  444.18000  5.45%     -    0s
     0     0  444.18000    0   57  469.80000  444.18000  5.45%     -    0s
     0     0  444.18000    0   30  469.80000  444.18000  5.45%     -    0s
     0     0  444.18000    0   28  469.80000  444.18000  5.45%     -    0s
     0     2  444.18000    0   25  469.80000  444.18000  5.45%     -    0s
H   39    14                     464.4000000  444.18000  4.35%  43.0    0s
  1677  1064  453.39449   89   62  464.40000  453.39449  2.37%  25.3    5s
  3447  1968  453.57132   67   66  464.40000  453.56786  2.33%  40.7   10s
 12767  9087  454.20000   81   60  464.40000  453.56786  2.33%  44.9   15s
 19223 13751  454.20000  155   49  464.40000  453.56786  2.33%  48.7   20s
 30349 22212  454.20000   99   63  464.40000  453.56786  2.33%  50.6   25s
 32457 24508     cutoff  208       464.40000  453.56786  2.33%  52.0   30s
 34326 24517  454.57192   90   44  464.40000  453.57876  2.33%  52.2   35s
 34528 24600  460.13955   68   48  464.40000  453.57876  2.33%  53.3   41s
H34536 23370                     459.0000000  453.57876  1.18%  53.3   41s
 35764 22957  457.03698   84   58  459.00000  455.39391  0.79%  54.4   45s

Cutting planes:
  Gomory: 16
  Cover: 21
  Clique: 3
  MIR: 158
  StrongCG: 112
  Flow cover: 115
  Inf proof: 4
  Zero half: 5

Explored 36534 nodes (1987854 simplex iterations) in 46.25 seconds (46.25 work units)
Thread count was 10 (of 12 available processors)

Solution count 8: 459 464.4 469.8 ... 775.2

Optimal solution found (tolerance 1.00e-03)
Best objective 4.590000000000e+02, best bound 4.590000000000e+02, gap 0.0000%
```

The optimal specific load (максимальная удельная загрузка): 0.9677124183006532
Load of trucks (распределение рулонов по машинам):
Truck 1 (22.2) is loaded with rolls: [36, 37, 54] with weights: [8.26, 8.2, 5.74].
Truck 2 (22.2) is loaded with rolls: [32, 41, 47] with weights: [8.31, 7.85, 6.03].
Truck 3 (22.2) is loaded with rolls: [31, 39, 52] with weights: [8.31, 8.12, 5.76].
Truck 4 (22.2) is loaded with rolls: [33, 35, 55] with weights: [8.28, 8.27, 5.64].
Truck 5 (22.2) is loaded with rolls: [22, 30, 56] with weights: [8.49, 8.33, 5.31].
Truck 6 (22.2) is loaded with rolls: [34, 38, 50] with weights: [8.27, 8.15, 5.78].
Truck 7 (22.2) is loaded with rolls: [17, 44, 46] with weights: [8.57, 7.06, 6.3].
Truck 8 (27.6) is loaded with rolls: [3, 11, 19] with weights: [8.77, 8.62, 8.53].
Truck 9 (27.6) is loaded with rolls: [1, 4, 10] with weights: [8.78, 8.76, 8.63].

Truck 10 (27.6) is loaded with rolls: [14, 15, 16] with weights: [8.61, 8.6, 8.57].
Truck 11 (27.6) is loaded with rolls: [24, 26, 29] with weights: [8.44, 8.41, 8.37].
Truck 12 (27.6) is loaded with rolls: [28, 43, 48, 49] with weights: [8.38, 7.13, 6.02, 5.81].
Truck 13 (27.6) is loaded with rolls: [25, 40, 53, 57] with weights: [8.42, 8.12, 5.75, 5.31].
Truck 14 (27.6) is loaded with rolls: [2, 6, 13] with weights: [8.77, 8.73, 8.62].
Truck 15 (27.6) is loaded with rolls: [5, 7, 8] with weights: [8.74, 8.72, 8.7].
Truck 16 (27.6) is loaded with rolls: [9, 12, 20] with weights: [8.63, 8.62, 8.5].
Truck 17 (27.6) is loaded with rolls: [18, 21, 23] with weights: [8.53, 8.49, 8.47].
Truck 18 (27.6) is loaded with rolls: [27, 42, 45, 51] with weights: [8.39, 7.14, 6.31, 5.76].