Optimization of Cutting Large Amounts of Dense Material

Ivan Brezina [1], Jaroslav Kultan [2], Juraj Pekár [1]

[1] *University of Economics in Bratislava, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemská cesta 1/b, 852 35 Bratislava*
[2] *University of Economics in Bratislava, Faculty of Economic Informatics, Department of Applied Informatics, Dolnozemská cesta 1/b, 852 35 Bratislava*

Abstract **– The effective operation of any manufacturing enterprise cannot be achieved without optimal allocation of available resources. Efficient distribution of input materials, entering production, and procurement costs are significant parts of production costs. One of the primary requirements to ensure a given company's production efficiency and competitive capability is its optimization. An area within this context is the distribution of, for instance, dense material required for constructing concrete structures such as bridges, roads, buildings, and more. When cutting dense material, it is necessary to create a plan for optimal cutting to minimize waste. This paper aims to create a methodology for optimizing the cutting of metal rods with the goal of waste reduction. The software MS Excel and Python solutions have been developed and used based on the methods employed. The computed solutions can serve as a foundation for creating a data repository, which holds information about purchased materials and realized production and material remnants that can be utilized in subsequent cutting. Elements of the data repository will encompass cutting plans and remnants resulting from the implementation of calculated cutting plans. The data repository enables a multidimensional view for managing the cutting process.**

DOI: 10.18421/TEM131-03
https://doi.org/10.18421/TEM131-03

Corresponding author: *Ivan Brezina,*
University of Economics in Bratislava, Faculty of Economic Informatics, Department of Operational Research and Econometrics, Dolnozemská cesta 1/b, 852 35 Bratislava,
Email: ivan.brezina@euba.sk

Received: 05 October 2023.
Revised: 09 January 2024.
Accepted: 16 January 2024.
Published: 27 February 2024.

Keywords **– Optimal material allocation, Python, cutting plan, mathematical programming.**

1. Introduction - Motivation

In response to challenges faced by companies like those using metal rods for construction reinforcement or wood sheets for furniture production, the authors of this paper propose a novel system for accurate and efficient material cutting based on customer demands. In practice, approaches to solving cutting problems may not only provide cut plans but also address other aspects and answer additional questions. If a new cutting plan that is different from the previous one is implemented, it can save time and resource consumption. Therefore, approaches to solving cutting problems have been developed to minimize material waste. The problems of creating cutting plans are quite often used when cutting from large paper discs, when cutting grinding discs to the dimensions ordered by customers, when cutting corrugated cardboard, or in the steel industry for cutting steel strips or plates to the dimensions required by customers [4]. A large number of published articles are primarily focused on the organization of cutting in such a way as to minimize downtime for setup machines or for optimization of already used cutting plans. However, the submitted contribution is more focused on describing the possibilities of creating cutting plans and their subsequent optimization.

Due to incorrect cutting plans, significant material losses can occur during cutting operations, resulting in inefficient material utilization, substantial waste, energy losses, and more. Presently, optimal decision-making in utilizing existing material and energy resources plays an increasingly significant role. This addresses the problem of scarcity in many metallurgical products, especially iron rods for bridge repairs, construction of new buildings, or renovation of existing structures. Properly solving cutting-related issues can translate into substantial savings of valuable materials.

For a company that utilizes a considerable volume of metal rods with varying diameters and lengths, requiring these to be cut into a specific number of shorter pieces for further processing, any inaccuracies can lead to substantial material and economic losses. Hence, optimally dividing cuts for individually available metal rods into the desired quantity of specified lengths receives significant attention. This is because with each delivery, there can be significant unnecessary financial costs.

Many Slovak companies address this issue by relying on experienced workers who create cutting plans based on empirical knowledge. The effectiveness of such practices is highly debatable, and companies often incorporate material losses into the prices of their products, rendering them less competitive.

The paper presents a potential solution to this problem for a company that receives significant orders for preparing iron reinforcement structures for large-scale constructions. The optimal distribution of cuts can lead to substantial financial savings for the company.

Numerous companies producing reinforcement for various constructions or concrete structures deal with the challenges of efficiently utilizing available materials. These materials can be entirely new products or remnants from prior activities. One such type of material is long rods of construction material with varying diameters, used to manufacture profiles of diverse shapes.

The mentioned material cutting issues can be formulated as a mathematical model that represents a simplified description of the real system of cutting dense material (metal rods). The application of mathematical programming methods seems to be effective, as they allow a relatively straightforward transformation of analyzed processes into a mathematical model in the form of functions, inequalities, or equations. The formulated model can be solved using mathematical programming tools supported by corresponding software tools, specifically the Python program. This approach enables a relatively accurate quantitative realization of selecting a specific solution from a vast array of possible solutions, with the best choice being determined according to the mathematical formula.

Based on the solution to the relevant mathematical programming problem, it is possible to create a data structure – a Data Warehouse (DWH) – that stores data about materials in stock, final products, and individual calculated solutions. The DWH also holds information about material remnants. A data warehouse can provide a multidimensional view of the acquired data, which could be a prerequisite for more efficient utilization of working time and materials.

In the paper, the authors generalized the experiences gained while resolving this problem using mathematical programming tools and the Python program.

The most intriguing challenge was not necessarily solving the constructed mathematical programming problem, known as the cutting problem, as found in the literature. Instead, the construction of custom cutting plans served as the basis for assembling the corresponding mathematical programming problem. Cutting plans for extensive assignments involves numerous variations that can be eliminated using appropriately formulated rules while leveraging the Python program.

By analyzing solutions to similar tasks, it is possible not only to advance the theory of solving problems of this type and their implementation in today's critical areas but also to develop new solutions for similar tasks.

Therefore, the contribution is oriented in such a way that the motivation for solving the problem of cutting a large amount of material is presented in the introduction. In the second part, the problem of describing the real problem of a company that deals with the production and distribution of construction steel to the end consumer is formulated. In the third part, the cutting problem is formulated as a task of mathematical programming; in the fourth part, the solution to the cutting problem is presented as a mathematical programming task with an emphasis on creating cutting plans using the Python program, and the results are summarized in the conclusion.

2. Problem Description

The considered company is engaged in the production and distribution of construction steel to end consumers, as well as the distribution of other dense materials. Its activities primarily involve cutting metals, including cutting and shearing metals, computer-controlled metal cutting (CNC), precision metal cutting, bending metals, press bending of metals, curving, bending and stretching of sheets, and bending of metal rods and profiles. Using a specialized process, the machinery is designed to process construction steel with diameters ranging from Ø6 mm to Ø32 mm. The production preparation utilizes the LP System program developed by the German company LENNERTS&PARTNER GmbH.

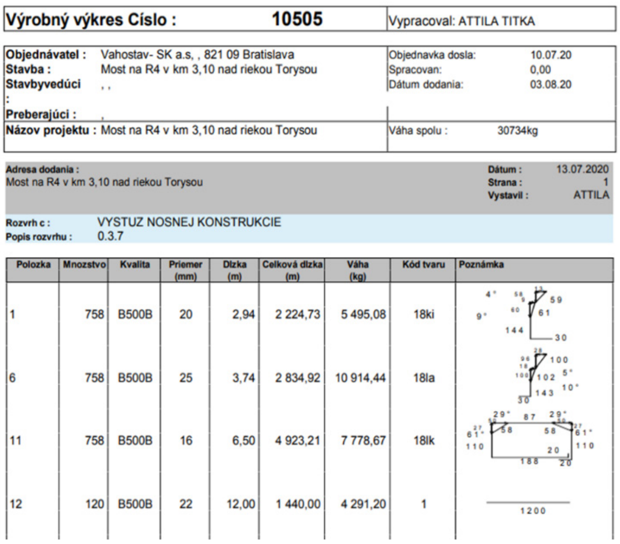
Raw materials are imported to the facility in 15-meter bundles with 8 mm to 32 mm diameters. Waste generated from processing construction steel is minimized during the processing phase to maximize the company's yield from the input construction steel due to economic interests. Waste management is in accordance with the Waste Act No. 79/2015 Coll. and relevant implementing regulations.

The production process is implemented in such a way that the production planners deliver to the shift leader the production labels for the individual positions of the manufactured product in the form of production labels; the shift leader sorts the individual production labels according to the diameter and type of machine and ensures that the products produced for the given project will be made of concrete steel on this project classified. The regulations established by the MSZ EN 13670:210 standard are observed in the production of concrete steel.

Based on the labels, machinists set the values on the machine tool that indicate the dimensions to which the raw material should be machined. After entering the data and setting the machine, the required amount of reinforcing steel is taken from the loaded bundle and manually transferred from the preparation table to the cutter's cutter.

Based on orders and production drawings, labels intended for bending machines are created at the beginning of the production process.

For each product, a ticket is created (Fig. 1), where the primary data on the desired shape of the product, the diameter of the material, the required length from which the product is made, and other economic parameters are listed.



Výrobný výkres Císlo : 10505 — Vypracoval: ATTILA TITKA

Objednávatel : Vahostav- SK a.s, , 821 09 Bratislava
Stavba : Most na R4 v km 3,10 nad riekou Torysou
Stavbyvedúci : , ,
Preberajúci : ,
Názov projektu : Most na R4 v km 3,10 nad riekou Torysou

Objednavka dosla: 10.07.20
Spracovan: 0,00
Dátum dodania: 03.08.20
Váha spolu : 30734kg

Adresa dodania :
Most na R4 v km 3,10 nad riekou Torysou

Dátum : 13.07.2020
Strana : 1
Vystavil : ATTILA

Rozvrh c : VYSTUZ NOSNEJ KONSTRUKCIE
Popis rozvrhu : 0.3.7

Polozka	Mnozstvo	Kvalita	Priemer (mm)	Dlzka (m)	Celková dlzka (m)	Váha (kg)	Kód tvaru	Poznámka
1	758	B500B	20	2,94	2 224,73	5 495,08	18ki	
6	758	B500B	25	3,74	2 834,92	10 914,44	18la	
11	758	B500B	16	6,50	4 923,21	7 778,67	18lk	
12	120	B500B	22	12,00	1 440,00	4 291,20	1	

Figure 1. Basic product information sheet for Most R4.
Source: Company documents

The data from the given product ticket are transformed into a table or stored in the product database. Figure 2 shows the form of an entity of the database system.

Host: *185.64.219.10*
Database: *database*
Generated: *Wed 26. Apr 2023, 10:03*
Generated by: *phpMyAdmin 3.5.8.1 / MySQL 5.6.51-91.0-log*
SQL query: *SELECT * FROM `01_product` LIMIT 0, 30;*
Rows: *2*

production drawing	item	quantity	diameter	length	truck number
10505	1	758	20	2.94	103
10505	6	758	25	3.74	103

Figure 2. Output of product data from the database system.
Source: Own processing

Based on the provided data, it is necessary to create a cutting plan, and the resulting data is again recorded in tables or directly in the database system. The key arguments for this data are the production drawing and the item, or a separate attribute can be created – the request number. Additionally, information about available materials is stored in the database – the warehouse entity. The main parameters are the diameter and length of the rod, as well as whether it is a new or already used rod (Fig. 3).

The result of the SQL query
Host: *185.64.219.10*
Database: *database*
Generated: *Wed 26. Apr 2023, 12:38*
Generated by: *phpMyAdmin 3.5.8.1 / MySQL 5.6.51-91.0-log*
SQL query: *SELECT * FROM `03_sklad` LIMIT 0, 30;*
Rows: *2*

diameter	quantity	length	attribute	unit of measure	number	note
20	500	25000	N	mm	1	
24	200	15000	P	mm	2	

Figure 3. Data on available materials.
Source: Own processing

Based on the provided basic data, cutting plans are created for each item and order. To solve this problem, methods related to material cutting can be employed. Among the most well-known are methods for cutting long or two-dimensional materials.

The optimization of cutting the corresponding rods is based on the assumption that a cutting plan needs to be constructed for each rod, within which the fundamental data of individual cuts will be specified. According to these cutting plans for each diameter, multiple orders must be executed, each containing several rods of the given diameter destined for cutting.

The process and interconnection of individual activities can be algorithmically described as follows:

1. ***Create an Orders Table***:

1.1. Extract requirements from each order and populate a table with order details, diameters, and quantities.

1.2. Group individual items by diameter of iron rods.

2. Select Available Stock Table:

2.1. Retrieve a table with available rod diameters and lengths in stock.

3. Generate Cutting Plans for Each Diameter:

3.1. Construct a table of cutting plans for the first considered diameter.

3.2. Formulate the corresponding mathematical programming problem (cutting problem).

3.3. Solve the cutting problem for the first diameter.

3.4. Record data in the table of a cut with the following information:

- Order number/loading number
- Item number
- Diameter
- Number of rods
- Number of cuts
- Waste per rod
- Total waste

3.5. Conclude the preparation of cuts for the specific diameter.

4. Select Next Diameter and Repeat Step 3:

• Repeat steps 3.1 to 3.5 for each subsequent diameter.

5. Create a File with Ordered Data:

• Compile a file containing the various diameters sorted by order, items, used rod counts, and cut counts for individual lengths.

3. Rod Cutting Problem as a Mathematical Programming Task

The problem of cutting rods, also known as the cutting-stock problem (CSP), e.g. [5], [6], belongs to the class of mathematical programming problems [2] designed to find the best solution under given constraints.

The cutting-stock problem represents an integer mathematical programming model in which a solution is sought for cutting (dividing) larger pieces into smaller ones while minimizing waste. This task can involve one-dimensional problems, where cutting is performed in only one direction, such as rods or bars, cables, strips of the same length [1], [8], [9], or two-dimensional problems [3], [7]. Cutting problems can generally be characterized based on their dimensions and shapes. The dimensionality in Cartesian space can be one, two, or three-dimensional. Cutting problems can be described based on the shapes of objects (2D – rectangles, 3D – cubes, etc.), but they can also be characterized based on spatial shapes like rectangles, ellipses, and circles.

The practical determination of a high-quality cutting plan (all possible cutting variants) presents a mathematical problem that appears to be time and memory-intensive. When considering a cutting plan as a mathematical programming task, it is necessary to create individual cutting plans, which means generating all possible approaches to cutting larger pieces – how many alternatives can be used for this purpose.

A cutting plan can then be represented as shown in Figure 4 (assuming a total length of the standard rod is 100, with 25 rods of length 50l required, 50 rods of length 20l, and 35 rods of length 30l, with no waste allowed), or in the general Table 1 (where the value $m1$ represents the total number of different types of required smaller pieces).

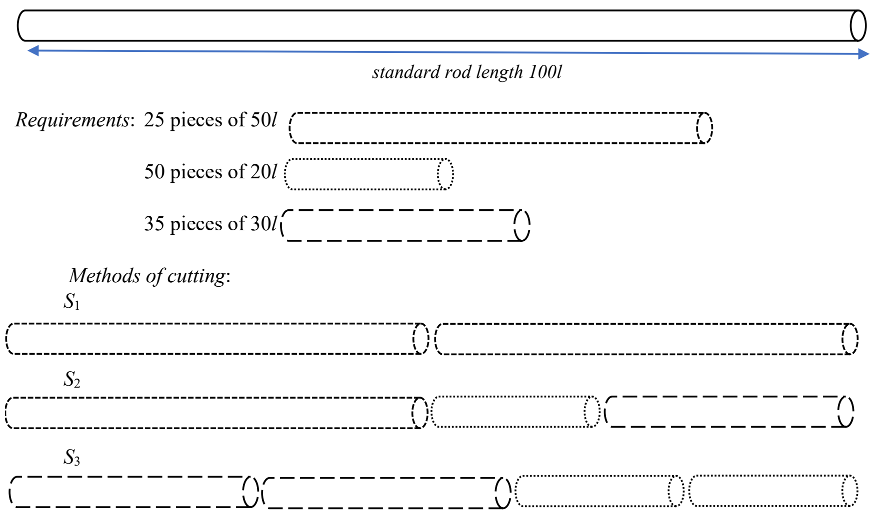


Figure 4. Possibilities of cutting smaller units.
Source: Own processing

Table 1. Creating a cutting plan

The length of the whole	Methods of cutting						Required number of pieces b_i
	S_1	S_2	S_3	...	...	S_n	
d_1 cm	a_{11}	a_{12}	a_{13}			a_{1n}	b_1
d_2 cm	a_{21}	a_{22}	a_{23}			a_{2n}	b_2
...							...
d_{m1} cm	$a_{m1,1}$	$a_{m1,2}$	$a_{m1,3}$			$a_{m1,n}$	b_{m1}
Garbage	o_1	o_2	o_3			o_n	
The decision variable	x_1	x_2	x_3	...	...	x_n	

Source: Own processing

The practical finding of a high-quality cutting plan (all possible cutting methods) is a mathematically NP-complete problem, the solution of which is still time and memory-intensive. Based on the generated cutting plan, a mathematical programming problem can be formulated and solved using appropriate methods (branch and bound method, Gomory's cutting-plane method, etc.).

As mentioned earlier, when formulating the cutting problem as a mathematical programming task, possible cutting plans must be created first. The number of cutting methods represents the number of decision variables (n) incorporated into the mathematical programming problem. Structural constraints (m) represent all the constraints that need to be included in the problem, with the basic constraints for the quantities of smaller pieces usually in the form of "=" if an exact amount of smaller pieces needs to be cut or in the form of "≥" if at least a certain quantity needs to be achieved. The task of finding the optimal cutting plan can then be formulated as follows:

$$\begin{aligned} &\min f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j \\ &\sum_{j=1}^{n} a_{ij} x_j \left\{ \begin{matrix} = \\ \geq \end{matrix} \right\} b_i ,\ i = 1,\ 2,\ \dots m \\ &x_j \geq 0,\ j = 1,2,\dots n \\ &x_j \in D_j,\ j = 1,2,\dots n \\ &x_j \in Z,\ j = 1,2,\dots n \end{aligned} \tag{1}$$

where

n – the number of decision variables representing the number of cutting methods,

m – the number of constraints, representing the number of different types of required rods,

d_j – total length of each type of input rod

x_j – decision variables representing the number of cuts in the *j-th* cutting method, where $j = 1, 2, ...n$,

c_j – coefficients of the objective function, representing the material waste in the *j-th* cutting method, where $j = 1, 2,...n$,

a_{ij} – technological coefficients of the constraint system, representing the number of rods of the *i-th* type obtained in the *j-th* cutting method, where $i = 1, 2,...m$, $j = 1, 2,...n$,

b_i – right-hand side coefficients, representing the required number of rods of the *i-th* type, where $i = 1, 2,...m$,

D_j – domains of values that x_j can take, where $j = 1, 2, ...n$,

Z – the set of integers.

The conditions $x_i \in D_j$, j=1,2,...n represent additional constraints, such as the number of available input rods used in the j-th cutting, requirements for the maximum allowed waste in the j-th cutting way, etc.

A one-dimensional cutting problem can be formulated as a classical integer linear programming problem. However, the nature of a two-dimensional cutting problem (as well as other variants of cutting problems) is significantly more complex.

The formulated problem of cutting metal rods essentially represents a one-dimensional cutting problem. In this type of linear programming problem, the challenge lies in determining what the decision variables mean. Each larger piece can be divided (cut) into smaller pieces in numerous ways, and the process refers to applying one of these potential division methods. Therefore, the individual decision variables correspond to the techniques (cutting methods), and the values of these decision variables indicate how many times the respective division method was used. The constraints in the mathematical programming problem must encompass all limiting conditions derived from the problem statement, including minimum or exact quantities of smaller pieces, limitations on the availability of standard larger pieces (rods), requirements for maximum allowable waste, and so on.

3.1. Formulation of a Problem with One Input Material

This sentence sets the scenario: Metal rods are available, each with a length of 750 cm. According to customer requirements, there is a need to produce three types of metal bars (m = 3). Specifically, 200 pieces of length l_1 = 250 cm, 250 pieces of length l_2 = 200 cm, and 50 pieces of length l_3 = 150 cm:

• to create an optimal cutting plan based on the criterion of minimizing waste, with waste not exceeding 100 cm.
• minimize the number of standard 750 cm long rods required to obtain the desired quantity of each metal bar type.

To solve the problem formulated in this way, it is necessary first to create all possible cutting methods captured in Table 2. In this table, the decision variables x_j represent the number of cuts of the standard rod using the j-th cutting method.

Table 2. Creation of a cutting plan when cutting bars 750 cm

Method of cutting	Product dimensions			Garbage	Number of cuts
	l_1 = 250 cm	l_2 = 200 cm	l_3 = 150 cm		
1	3	0	0	0	x_1
2	2	1	0	50	x_2
3	2	0	1	100	x_3
4	1	2	0	100	x_4
5	1	1	2	0	x_5
6	1	0	3	50	x_6
7	0	3	1	0	x_7
8	0	2	2	50	x_8
9	0	1	3	100	x_9
10	0	0	5	0	x_{10}
Number of required pieces	200	250	50		

Source: Own processing

Creating a Mathematical Model of the Problem

The generated cutting plan shows that ten cutting methods are possible with three types of metal bars (m = 3); thus, n = 10. The mathematical model of this problem consists of a system of 3 (m) equations with 10 (n) unknown parameters and constraints for non-negativity and integrality. The objective of the problem, expressed through the objective function, is based on the waste parameter, and it aims to minimize material losses from each cut. The corresponding mathematical programming problem, based on (1), is as follows:

$$\min f(\mathbf{x}) = 0x_1 + 50x_2 + 100x_3 + 100x_4 + 0x_5 + 50x_6 + 0x_7 + 50x_8 + 100x_9 + 0x_{10}$$

$$3x_1 + 2x_2 + 2x_3 + x_4 + x_5 + x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} = 200$$
$$0x_1 + x_2 + 0x_3 + 2x_4 + x_5 + 0x_6 + 3x_7 + 2x_8 + x_9 + 0x_{10} = 250$$
$$0x_1 + 0x_2 + x_3 + 0x_4 + 2x_5 + 3x_6 + x_7 + 2x_8 + 3x_9 + 5x_{10} = 50$$
$$x_{1,2,\dots10} \geq 0, x_{1,2,\dots10} \in Z \qquad (2)$$

3.2. Formulation of the Problem with Multiple Input Materials

If we assume the existence of input materials with multiple lengths, it is necessary to create an appropriate cutting plan for each input length.

Let is consider that metal rods are available in 3 different sizes ($l1$ = 250 cm, $l2$ = 230 cm, $l3$ = 200 cm). The task is to cut the following quantities of metal bars of different lengths: 140 pieces of 150 cm length, 100 pieces of 120 cm length, 200 pieces of 100 cm length, 130 pieces of 90 cm length, 200 pieces of 50 cm length, 120 pieces of 45 cm length, and 320 pieces of 30 cm length (Table 3).

Table 3. Requirements for production

Length	150	120	100	90	50	45	30
Count	140	100	200	130	200	120	320

Source: Own processing

To solve the problem formulated in this way, it is necessary to create all possible cutting methods, captured in Table 4, where the decision variables x_{ij} represent the number of cuts of the standard rod of the i-th type by the j-th method. Table 4 shows part of the cutting plan for several lengths of input material and various wastes. It is assumed that the waste must be smaller than the petite required length of the final product.

Table 4. Sample of part of the cutting plan

Method of cutting	Entry length	Product dimensions							Garbage	Number of cuts
		l_1=150	l_2=120	l_3=100	l_4=90	l_5=50	l_6=45	l_7=30		
1	250	1		1					0	x_{11}
2	250	1			1				10	x_{12}
3	250	1				2			0	x_{13}
	....	...	....	...	...	...	...	...	...	...
4	230	1					1	1	5	x_{21}
5	230		1			2			10	x_{22}
	....		....	...	...	...	...	...	...	...
6	200	1				1				x_{31}
7	200		1			1		1		x_{32}
8	200	1					1		5	x_{33}
...	....	....	....	...	...	...	...	...	...	...
Number of required pieces		140	100	200	130	200	120	320		

Source: Own processing

We create a mathematical model in accordance with (1) and (2) for each input rod length.

However, the mentioned cutting plan does not consider the available amount of each type of input material, so it is problematic to introduce as inputs also the offcuts from other cutting plans. Such an option would allow for better utilization of available materials.

3.3. Cutting Plan with Resource Constraints

The problem arises in finding a cutting plan also in cases where input resources are limited. This is a practical extension of the analyzed problem, within which the remnants of material in shorter lengths arise from previous deliveries, which should be utilized. A similar problem arises when changing material suppliers, especially in critical situations when the main supplier does not have sufficient required inputs. The substitute supplier will only provide a limited, necessary amount; if possible, input resources are again taken from the main supplier. In this case, the available number of rods is limited in accordance with the Table 5.

Table 5. Restrictive conditions

Rods	Count
200	500
230	200
250	100

Source: Own processing

We will create the cutting plan similarly to the previous case (Tab.6) and a mathematical model similar to the previous case in accordance with (1) and (2). When creating a mathematical model in accordance with the previous case. The mathematical model of the given task differs from the previous variant mainly by introducing limiting conditions resulting from the available amount of material. Moreover, when solving the given task, the two-dimensional matrix **X** of size n x m_i for i=1, 2, ...n, where n is the number of different inputs, was replaced by a vector $\boldsymbol{x}$ of size nm, where nm is the sum of all m_i values.

The mathematical programming model based on Table 4 will have the form:

$$\min f(\mathbf{x}) = 0x_1 + 10x_2 + 0x_3 + 5x_4 + 10x_5 + 0x_6 + 0x_7 + 5x_8 + \dots$$

$$\begin{aligned}
&x_1 + x_2 + x_3 + x_4 + 0x_5 + x_6 + 0x_7 + x_8 + \dots = 140\\
&0x_1 + 0x_2 + 0x_3 + 0x_4 + x_5 + 0x_6 + x_7 + 0x_8 + \dots = 100\\
&x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + \dots = 200\\
&0x_1 + x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + \dots = 130\\
&0x_1 + 0x_2 + 2x_3 + 0x_4 + 2x_5 + x_6 + x_7 + 0x_8 + \dots = 200\\
&0x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 0x_6 + 0x_7 + x_8 + \dots = 120\\
&0x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 + 0x_6 + x_7 + 0x_8 + \dots = 320\\
&x_1 + x_2 + x_3 + \dots \le 100\\
&x_4 + x_5 + \dots \le 200\\
&x_6 + x_7 + x_8 + \dots \le 500\\
&x_i \ge 0, x_i \in Z, x_i \in D_i, i = 1,2,\dots n \qquad (3)
\end{aligned}$$

Where the conditions $x_i \in D_i$, i=1,2,...n represent, as already mentioned, additional restrictions (e.g., the number of available input rods used in the i-th cutting, requirements for the maximum allowed waste in the i-th cutting way, etc.).

4. Solving the Cutting Problem as a Mathematical Programming Task

Based on the created cutting plan, it is possible to formulate and solve the task of integer linear programming. For real solutions to mathematical programming tasks, and thus also tasks of integer linear programming, professional software products such as GAMS, LINDO, LINGO, Python, Gurobi, etc. are available. However, in practice and academia, many other software and software languages are used to solve mathematical programming tasks (e.g. MATLAB, R language, Python, etc.). Next, we present a way to solve the task of the cutting problem as a task of integer linear programming in Python.

The solution to the problem can be divided into two relatively independent parts. The more computationally demanding one is finding a set of feasible cutting plans, which is very interesting from the point of view of the development of the theory of mathematical programming task formulation. The actual solution of the formulated task of mathematical (integer) programming consists of finding an optimal solution using standard methods.

When solving mathematical programming tasks in Python, it is primarily necessary to determine the programming environment. In addition to the basic IDLE (Integrated Development and Learning Environment) environment, tools such as Jupyter Notebook, and Spyder, which can be found under the integrated system Anaconda Navigator, can be used. Next, the solution to the task of integer linear programming (as a subset of mathematical programming tasks) in the IDLE environment will be introduced.

The next step is to choose a suitable module for solving general mathematical programming tasks. Two basic modules, SCIPY and MIP, are available for practical solutions. The SCIPY module is intended for solving mathematical programming tasks (linear and nonlinear tasks). Unlike the SCIPY module, the MIP module focuses on solving tasks of linear, integer, and bivalent programming. The variability of the MIP module can be characterized by the possibility of choosing an optimization solver Coin-or branch and cut (CBC), which is automatically installed with the MIP module, Gurobi (GRB), SCIP, GLPK, which can be used in case they are installed from sources other than Python.

The presented Python code for solving the analyzed problem is divided into two fundamental blocks: one for generating feasible cutting plans and another for the actual optimization of the linear programming problem.

The first block, the block for generating feasible cutting plans, can be divided into two subparts. In the first part, a set of all potentially usable combinations of cutting methods is created - a cutting plan (Fig. 5). This part of the code is focused on determining the maximum possible number of pieces obtained from the longest rod, i.e., setting the value of the variable "mm." In the next part, a set of all m-element combinations is constructed, with values ranging from 0 to mm+1, ensuring that no feasible combination of the cutting plan is omitted. To create the combinations mentioned, we use the "product" command, which includes parameters: a set of values from which the combinations will be created and the number of elements in each combination.

```
#determination of the maximum number of pieces of products of the same length from one rod
mm=0
for k in T:
 for i in V:
  if mm<=tyce[k][0]//vstup[i][0]:
      mm=tyce[k][0]//vstup[i][0]
# creating all potentially admissible combinations
res = [ele for ele in product(range(0, mm + 1), repeat = m)]
```

Figure 6. Creating a cutting plan when determining the maximum number of products of the same length from one rod.
Source: Own processing

In the second block, the block of actual mathematical programming optimization, you need to select only those cutting methods from the set of all methods that satisfy the conditions of the maximum possible waste when cutting (Fig. 6). In this block, the check is performed to determine whether the cutting plan is feasible for each type of input rods.

The feasibility check involves determining whether the cutting plan can be implemented at all, meaning whether the sum of cut rods is not greater than the length of the standard rod, and the waste does not exceed the maximum specified size. After selecting the feasible cutting methods, you can obtain the desired parameters of the mathematical model, where in the programmed code, the variable "rp" represents the quantities of obtained required rods from one cutting, "rod" is the length of the cut rod, and "o" is the waste after using the respective cutting plan (Fig. 6).

Then, user can proceed to solve the actual optimization problem, which is generally formulated as a mathematical programming problem (1). In the Figure 7, the problem is written in the Python programming language.

```
#selection of permissible cutting plans with a specified maximum waste of 200 mm
modpad=200
for k in T:
 for i in list(res):
  if sum(np.multiply(dlzky,np.array(i)))<=tyce[k][0]:
     if tyce[k][0]-sum(np.multiply(dlzky,np.array(i)))
          <=modpad:
      j+=1
      rp[j-1]=np.array(i)
      tyc[j-1]=tyce[k][0]
      o[j-1]=tyce[k][0]-sum(np.multiply(dlzky,np.array(i)))
```

Figure 7. Selection of permissible cutting plans with a specified maximum waste.
Source: Own processing

```
#declaration of mathematical model variables
x = [m.add_var(name='x', var_type = "I", lb=0) for i in PR]
# defining the objective function
m.objective = minimize(xsum(o[i]*x[i] for i in PR)
# defining structural boundaries
for j in V:
    m += PS[j]-xsum(rp[i][j]*x[i] for i in PR) == 0
for j in T:
    m+=tyce[j][1]-xsum(x[i] for i in PR
        if tyce[j][0]==tyc[i]) >= 0
m.optimize()
```

Figure 8. Declaration of variables of the mathematical model
Source: Own processing

Two Excel files are used as input data to solve the problem of cutting iron reinforcing bars as a mathematical programming task in the Python language. The data entry in the first file is represented by Table 6, in which the user enters the lengths and numbers of standard bars that will be used to cut the desired resulting bars. Values in the Table 7 represent the number of requests represented with the required length and their number.

Table 6. Number of available bars

Standard rods (cm)	Count
200	500
230	200
250	100

Source: Own processing

Table 7. Cutting requirements

Length of required bars (cm)	Count
30	320
45	120
50	200
90	130
100	200
120	100
150	140

Source: Own processing

After starting the Python program, data from MS Excel tables will be loaded, while the program will subsequently determine all permissible cutting plans and solve the formulated optimization task.

The results of the mentioned task are exported to an MS Excel file; the output format is shown in the table. Table 8 represents the results of solving the task with a sufficient amount of resources, and Table 9 represents the solution with a limited number of resources from some input set in accordance with the data presented in the Table 8.

The given software solved both cases, i.e., if the manufacturer has enough bars of the given dimensions, i.e., the manufacturer always cuts from a new supply, or if he uses waste from previous cuts and then uses cuttings, the number of which is limited.

Table 8. Cutting plan with a sufficient number of bars of the basic size

Method of cutting	Rod length	Product dimensions							Garbage	Number of cuts
		l_1=150	l_2=120	l_3=100	l_4=90	l_5=50	l_6=45	l_7=30		
1	250	1		1					0	100
4	250		1			2		1	0	50
6	250			1	1			2	0	60
13	230	1				1		1	0	40
14	230		1			1		2	0	15
15	230			2				1	0	20
17	230				2	1			0	35
35	230		1				1	2	5	25
39	230						5		5	19
23	200		1			1		1	0	10

Source: Own processing

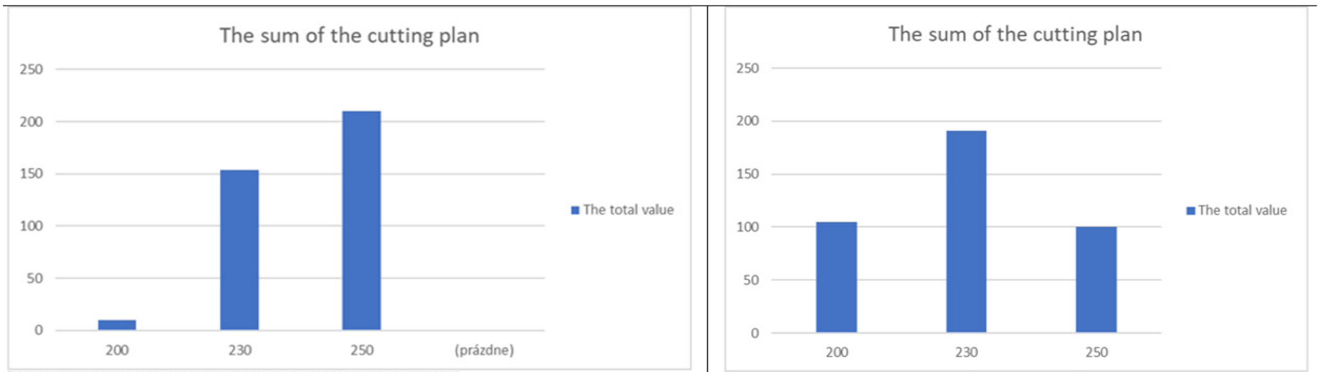
Table 9. Cutting plan with a limited number of inputs

Method of cutting	Rod length	Product dimensions							Garbage	Number of cuts
		l_1=150	l_2=120	l_3=100	l_4=90	l_5=50	l_6=45	l_7=30		
1.	250	0	1	1	0	0	0	1	0	5
2.	250	0	0	1	1	0	0	2	0	17
3.	250	1	0	1	0	0	0	0	0	78
4.	230	0	1	0	0	0	1	2	5	18
5.	230	0	0	0	2	1	0	0	0	24
6.	230	0	0	0	1	1	2	0	0	37
7.	230	1	0	0	0	1	0	1	0	62
8.	230	0	0	2	0	0	0	1	0	50
9.	200	0	1	0	0	1	0	1	0	77
10.	200	0	0	0	1	0	1	2	5	28

Source: Own processing

Based on the calculated optimal solution, it can be concluded that, in total, it is necessary to implement ten cutting methods of individual lengths to obtain the required number of bars, utilizing all types of bars (200 cm, 230 cm, and 250 cm). The total waste is 230 cm, with the largest from one bar being 5 cm, and will be realized in the 4th and 10th cutting plans.

Based on the results of individual cutting plans, we can compare the total amount of material used and the number of bars consumed with limited and unlimited resources.



line marking	sum from the cutting plan	the sum of the lengths
200	10	2000
230	154	35420
250	210	52500
Total sum	**374**	**89920**

line marking	sum from the cutting plan	the sum of the lengths
200	105	21000
230	191	43930
250	100	25000
Total sum	**396**	**89930**

Figure 9. Comparison of solving a multi-input problem with unlimited and limited resources.
Source: Own processing

In the Figure 9 on the left side, the number of individual bars and the total length of the consumed material are shown when there is a sufficient number of inputs of the required length. The required materials lengths are 89700 cm, and the total consumption is 89920 cm. It is evident from this that, with the given cutting plan, 220 more units are consumed. It also shows the total number of bars consumed, 374. When using a limited amount of input material, i.e., when using remnants from previous cuttings, more bars, 396, needed to be cut, and for the required amount of 89700 units, 89930 were consumed, which is 230 more compared to the required amount and is also more compared to the variant without limitations (Fig.9) - on the right.

5. Conclusion

The paper has grown in response to a specific request from a company specializing in the segmentation (cutting) of long metal bars or bars with various dimensions into lengths required by customers (cutting steel reinforcements or bars needed for large construction sites, especially highway construction). It is a response to the generalization of the authors experience in preparing a solution to the abovementioned problem using mathematical programming tools and the Python application program.

The proposed and implemented mathematical model for cutting materials is an effective tool for solving the problem mentioned above. Solving this problem as a formulated mathematical programming problem is supported by adequate software tools in the Python language and allows for finding an adequate solution from a vast number of possible solutions and finding the one that seems to be the most suitable from a mathematical point of view.

In addition to solving the formulated mathematical programming problem (which can be applied using standard tools), the design and implementation of cutting plans that create the basis for designing an appropriate mathematical programming problem represent the most interesting theoretical and practical problem. The creation of cutting plans involves many variants in large layouts, which can be eliminated by using suitable rules and the Python application program.

The problem-solving mentioned above allows for creating a suitable data structure and data warehousing (DWH) for storing data related to the amount of material in stock and manufactured products and for storing computational solutions. The DWH also contains data related to the rest of the material and generates a multidimensional data view, which could be very useful for more efficient material and working time consumption.

In further developing the calculation of the given methods, some possibilities of using parallel computations in solving the given tasks are being developed. One of the suggestions is the creation of possible cutting plans, with parallelization being divided according to the type of individual inputs. Another challenge is to reduce the number of possible solutions by introducing a maximum waste height of the minimum required length.

Solving the given task has opened a new problem in solving the given problem. On the one hand, utilizing waste allows for the efficient use of the available amount of material; on the other hand, it increases the number of cuts and the amount of material used when there is a limited amount of some inputs. This leads to the use of a larger number of input size bars, which may increase the labor intensity of the given activity. Therefore, a significant element in expanding the given task is the optimization of the number of cuts, which can reduce energy consumption and working time.

Acknowledgments

The Grant Agency of Slovak Republic supported this work – VEGA grant no. 1/0120/23 „Environmental models as a tool for ecological and economic decisions making.“

References:

[1]. Ágoston, K. C. (2019). The effect of welding on the one-dimensional cutting-stock problem: The case of fixed firefighting systems in the construction industry. *Advances in Operations Research*, *2019*.

[2]. Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., & Cochran, J. J. (2018). *An introduction to management science: quantitative approach*. Cengage learning.

[3]. Furini, F., Malaguti, E., & Thomopulos, D. (2016). Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, *28*(4), 736-751.

[4]. Henn, S., & Wäscher, G. (2013). Extensions of cutting problems: setups. *Pesquisa Operacional*, *33*, 133-162.

[5]. Jahromi, M. H., Tavakkoli-Moghaddam, R., Makui, A., & Shamsi, A. (2012). Solving an one-dimensional cutting stock problem by simulated annealing and tabu search. *Journal of Industrial Engineering International*, *8*, 1-8.

[6]. Kellerer, H., Pferschy, U. & Pisinger, D. (2004). *Knapsack Problems*. Berlin, Heidelberg: Springer-Verlag.

[7]. Malaguti, E., Durán, R. M., & Toth, P. (2014). Approaches to real world two-dimensional cutting problems. *Omega*, *47*, 99-115.

[8]. Tanir, D., Ugurlu, O., Guler, A., & Nuriyev, U. (2016). One-dimensional cutting stock problem with divisible items. *arXiv preprint arXiv:1606.01419*.

[9]. Tanir, D., Ugurlu, O., Guler, A., & Nuriyev, U. (2019). One-dimensional cutting stock problem with divisible items: A case study in steel industry. *TWMS Journal of Applied and Engineering Mathematics*, *9*(3), 473-484.