TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
CIVIL ENGINEERING | BUDOWNICTWO
2-B/2016

DOI: 10.4467/2353737XCT.16.155.5766

JACEK BOROŃ*, PAWEŁ BARTYLA**

APPLICATION OF THE MATHEMATICAL PROGRAMMING IN SOLVING THE MANAGERIAL PROBLEMS ENCOUNTERED IN THE PRODUCTION, TRADE OF CONSTRUCTION MATERIALS AND PURCHASES ORGANIZATION

ZASTOSOWANIE METOD PROGRAMOWANIA MATEMATYCZNEGO DO ROZWIĄZYWANIA PROBLEMÓW MENEDŻERSKICH NAPOTYKANYCH W PRODUKCJI MATERIAŁÓW BUDOWLANYCH, HANDLU I ORGANIZACJI DOSTAW

Abstract

The article presents, in a manager friendly way, three practical applications of methods of mathematical programming as tools for computer-aided purchase organisation and sawmill order realisation. The problems are illustrated by examples of optimising: the head saw use in planning the timber production realisation, the purchases for small warehouse and the supply of cement to the network of concrete plants or precast plants.

Keywords: managerial problems, mathematical programming, organisation of supply, production

Streszczenie

W niniejszej pracy zaprezentowano w sposób przyjazny dla menedżerów budownictwa trzy skrajnie różne przykłady praktycznego zastosowania programowania matematycznego w planowaniu produkcji tartacznej, zaopatrzenia małej hurtowni materiałów budowlanych i organizacji dostaw do sieci zakładów produkujących prefabrykaty betonowe.

Słowa kluczowe: organizacja dostaw, problemy menedżerskie, produkcja, programowanie matematyczne

* Ph.D. Eng. Jacek Boroń, Faculty of Civil Engineering, Wroclaw University of Science and Technology.
** M.Sc. Eng. Paweł Bartyla, Project Manager in Pekabex Pref sp. z o.o.

1. Introduction

Timber production plants, building material warehouses, construction sites and precast plants are places where contemporary managers have to face various decision-making problems related to production, management policy, or the right course of action. It is often the case that, due to many variables, complex selection criteria, and the constraints on the solution space, a seemingly uncomplicated problem becomes difficult to solve, and finding even a satisfactory solution may take a lot of time or turn out almost impossible with just 'pen and paper'. Mathematic programming techniques have been formulated, developed and used in the last 50 years to such extent that now they represent a considerably mature area of applied mathematics [1–3]. Optimisation tools became widely available thanks to spreadsheets [4] (i.e. MS Excel), which allow any user of such a software to try optimisation techniques in practice. Optimisation techniques, which, until recently, were associated mainly with mathematic programming techniques, have become a vital element of work of constructors and managers alike, also in the construction industry [2]. It is a result of easy access to personal computers and rapid increase of their performance, along with the development of simple software tools which are user-friendly and do not scare potential users with a seemingly complicated mathematic interface. Using spreadsheet-type software, you can build a transparent mathematical model of a particular decision-making problem, which reflects the problem's true nature in a faithful or sufficiently accurate manner. It is usually possible to describe the real relationships in indispensable measure for finding a solution or a course of action within the set of feasible solutions. Solving an optimisation problem, i.e. 'combing' the set of feasible decisions and choosing the best one, is possible thanks to the Excel Solver add-in [5, 6]. A properly conducted optimisation process allows us to make the right decision which was 'not visible' before, but is better than the previous one (the one based on assumptions or intuition). It has the potential to bring substantial cost savings in the area of production or resource management, and the bigger the scale of the problem, the bigger the savings [1, 2].

2. Case no 1: optimisation of head saw use in planning the sawmill order realisation

The following production issues were analysed in the course of the deliberation (see Table 1 below):

Table 1

General specification of the sawmill order

The wi the width of the plank [mm]	**150**	**150**	**150**	**150**
The thickness of the plank [mm]	19	25	32	45
The total length of the planks needed [m]	150	140	200	30

It was established that the sawmill has 8 hours including one technical break (approximately 1 hour), to realise the order, which leaves 7 hours to fulfil the task. The sawmill has three head saw lines at its disposal with the following exemplary technical specification (see Table 2):

Table 2

Basic characteristics of owned equipment (exemplary data)

Parameters →	**The cutting speed** [m/h]	**The cost of cutting** [PLN/h]	**The maximum thickness of the planks** [mm]
Head saw 1	40	150	63
Head saw 2	42	190	38
Head saw 3	50	225	28

Minimal cost of timber production was set as the optimisation criterion. In order to simplify the process, the criterion was limited to the constituent originating from owned head saws. Construction of the mathematic model started with defining decisive (design) variables. In this example, the quantities of certain types of timber measured in linear metres assigned to specific head saws will serve as decisive (design) variables. Head saw no. 1 is able to produce planks in any ordered thickness (see Table 1 and compare with Fig. 1), hence the quantitative variables for it can be specified in any column. Head saw no. 3, on the other hand, produces planks with a maximum thickness of 25 mm, hence the variables can be inserted only in '19 mm' and '25 mm' columns. Alternatively, for this exercise, linear metres can be changed to head saw work time needed to produce a specific type of timber. Due to the multitude of possible situations, a seemingly easy issue becomes much more complicated.

⌂	A	B	C	D	E	F
9	TIMBER TO PRODUCE [m]					
10	available		the desired thickness of the planks			
11	sawmill machinery		19 mm	25 mm	32 mm	45 mm
12	head saw 1		t1g19	t1g25	t1g32	t1g45
13	head saw 2		t2g19	t2g25	t2g32	-
14	head saw 3		t3g19	t3g25	-	-
15	for example t2g32 = an unknown amount of planks with a thickness of 32 mm ,					
16	to be produced by the head saw 2					

Fig. 1. An example of timber ordered for production (an excerpt of a spreadsheet)

2.1. Objective function and optimality criterion

In the contemplated example, the minimal average cost of timber production was set as the decisive criterion. The data of head saws engaged in the process was inserted in the chart (see Fig. 2). It was used to apply appropriate function dependencies based on time

which set the basic construction of the utmost simplified cost. Having the equipment speed specifications, the total work time of each head saws was calculated.

⌂	A	B	C	D	E
1	THE CHARACTERISTICS OF TIMBER CUTTING MACHINES				
2		machine	machine	maximum	impassable
3	machine type	velocity	cost	thickness of the plank	net working time
4		[m/h]	[PLN/h]	[mm]	[h]
5	head saw 1	40	150	63	7
6	head saw 2	42	190	38	
7	head saw 3	50	225	28	

Fig. 2. Extended characteristics of the sawmill machines (an excerpt of a spreadsheet)

Next, a general (total) timber production time was determined; since the head saws can operate simultaneously, the total work time will be the longest cutting time calculated for one of them (G17=MAX(G12:G14); see Fig. 3). This way, fundamental part of order production cost minimisation, head saws work plan-dependent mathematic model was built.

⌂	A	B	C	D	E	F	G	H
9	TIMBER TO PRODUCE [m]							
10	available		the desired thickness of the planks				the time	the cost
11	sawmill machinery		19 mm	25 mm	32 mm	45 mm	[h]	[PLN]
12	head saw 1		0	0	0	0	0.00	0.00
13	head saw 2		0	0	0	-	0.00	0.00
14	head saw 3		0	0	-	-	0.00	0.00
15			start values				TET (total execution time)	Σ
16	total timber production		0	0	0	0		
17	size of the order [m]		150	140	200	30	0.00	0.00

Fig. 3. Timber ordered for production (construction of a spreadsheet with time and cost calculation)

2.2. Optimisation constraints

Limitations that form the area or space of applicable solutions can be related to the quantity of available materials, money, time, production or order specifications etc. These variables are very important in the process of optimisation because without them, due to iteration discrepancies, finding a solution may be impossible. The following assumptions were applied in the contemplated issue:

– allowable production time (see Fig. 2, E5÷7 = allowable_time <= 7), meaning non-extendible net time (after subtracting technical breaks)
– order size demand (see Fig. 3, C17÷F17). The following formula was inserted into C16 cell: =SUM(C12:C14). Next, it was copied to cells from the D16:F16 range. Quantities of ordered timber of specific thickness were inserted into C17:F16 range cells. Next, two new cell range name definitions were introduced: C16:F16 = overall_produced, and C17:F17 = order.

Data from the abovementioned mathematic optimisation model was inserted into Excel Solver software add-on window. Since we weren't interested in all possible real solutions

(i.e. length of planks cut on subsequent machines shown in cm), discrete limitations for changeable cells were introduced. Also, the following option was selected in Excel Solver software: 'Set non-negative values for variables without limitations'. It is obvious that negative values of planks are absurd, yet the software 'doesn't know that' and wastes time calculating unacceptable areas. It is also worth noting that all function dependencies applied to the model are linear, and so the LP Simplex [8, 10] method can be used to solve the issue. Parameters available in *Options* were set to 'default'.

2.3. Interpretation of the results and discussion

Optimal values appeared in changeable (decisive) cells – best possible option to achieve lowest costs under specified limitations. Such conclusion can be derived since the model is linear and the abovementioned LP Simplex method was used to solve the problem. It should be noted that, in non-linear problems this method is absolutely useless. Results obtained in the contemplated example (see Fig. 4) appear to be 'rational' (meaning reasonable enough to try in practice).

	A	B	C	D	E	F	G	H
9	TIMBER TO PRODUCE [m]							
10	available		the desired thickness of the planks				the time	the cost
11	sawmill machinery		19 mm	25 mm	32 mm	45 mm	[h]	[PLN]
12	head saw 1		50	0	200	30	7.00	1050.00
13	head saw 2		0	0	0	-	0.00	0.00
14	head saw 3		100	140	-	-	4.80	1080.00
15	results						TET (total execution time)	Σ
16	timber produced [m]		150	140	200	30		
17	size of the order [m]		150	140	200	30	7.00	2130.00

Fig. 4. Result of the optimisation problem solution (an excerpt of a spreadsheet)

And what if calculations done in a little bit different limiting conditions gave a similar solution (see Fig. 5)? The following question comes into mind: 'Is it worth turning on machine no. 2 only to cut 2 metres worth of plank?' It all depends on the real-life situation and what we consider as rational actions. Apart from that, the planks are cut from long, usually 4–8 m logs of wood, and therefore, the total length of the produced planks should equal a multiple of the length of available wood.

	A	B	C	D	E	F	G	H
9	TIMBER TO PRODUCE [m]							
10	available		the desired thickness of the planks				the time	the cost
11	sawmill machinery		19 mm	25 mm	32 mm	45 mm	[h]	[PLN]
12	head saw 1		50	0	200	30	7.00	1050.00
13	head saw 2		0	2	0	-	0.05	9.05
14	head saw 3		100	138	-	-	4.76	1071.00
15	results						TET (total execution time)	Σ
16	timber produced [m]		150	140	200	30		
17	size of the order [m]		150	140	200	30	7.00	2130.05

Fig. 5. Alternative solution of the optimisation problem (an excerpt of a spreadsheet)

Therefore, it is recommended to analyse the obtained results after solving the problem in order to determine their real-life introduction sense (the results come from a mathematic

model, which does not always describe or determine all characteristics of the contemplated case and therefore cannot be trusted without further consideration).

2.4. A hypothetical compromise between cost and time

A production plant, such as a sawmill, needs to realise customers' orders nonstop, and therefore, every time an order is placed, the manager 'squeezes' it into an already tight production schedule. Therefore, a simulated, seemingly 'double criteria' optimisation resulting in a compromise between time of production and cost might be useful in this case. In order to make such analysis, the abovementioned calculations were repeated for the selected time period (in the contemplated example for the 13.0–4.5 hours' time range, counted every 30 minutes). The results of these calculations were presented below (see Fig. 6). It should be noted that the time production extension will result in cost reduction since Excel Solver assigned more work to head saw no. 1, which is the most cost-effective. In order to fulfil the task in a shorter period of time, head saw no. 3 (the fastest and least cost-effective) joins in the production process along with head saw no. 2 if the production time is supposed to be lower than 6 hours. The presented graph can be used to choose the most appropriate compromise between order completion cost and time. Compromise solutions are the essential part of multi criteria optimisation. Unfortunately, the spreadsheet does not allow vector optimisation, however, it allows a simulation of such. The choice is always made by the manager, or decision-maker, which was described in [11].

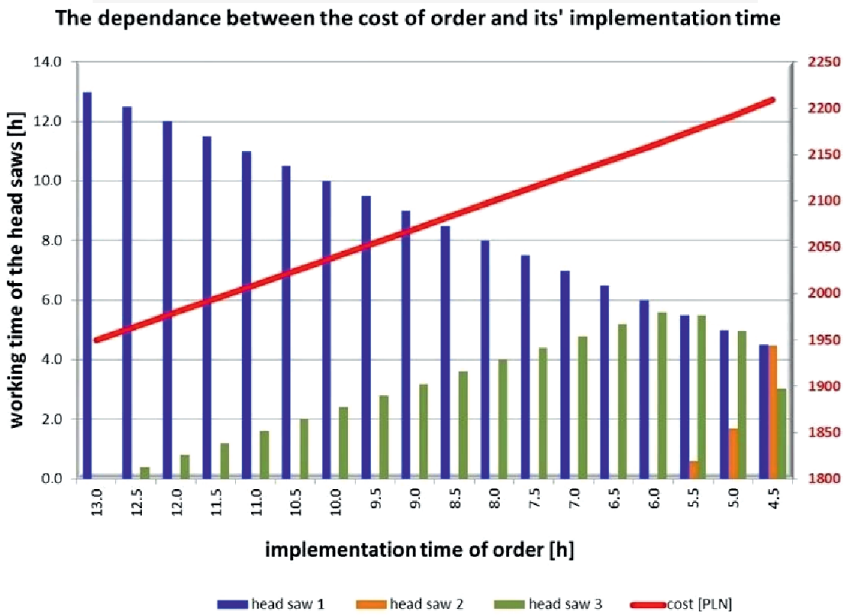


Fig. 6. Simulated double criteria optimisation: the relation between order, cost and its completion time

3. Case no 2: optimisation of the monthly supply of warehouse with building materials [5]

Let's assume that a certain building material warehouse operates in the N city in Lower Silesia (Poland). The warehouse consists of a hall for storing materials that require weather protection and of a partially covered storage yard for storing weatherproof materials of more sizeable overall dimensions. Every month, the warehouse manager has to face the same dilemma: 'What will be the demand for the building materials next month and what quantities of these materials should I order to keep the warehouse upkeep costs to a minimum?' In literature, a problem of this type is referred to as the Economic Order Quantity (EOQ). The easiest way to illustrate the problem is through the chart below:

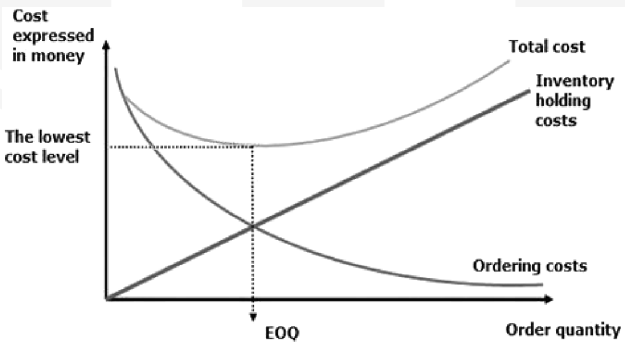


Fig. 7. The chart illustrating the concept of Economic Order Quantity (EOQ)

In a nutshell, the costs of maintaining a building material warehouse primarily include the following:

- ordering costs, e.g. fuel costs (theoretically, the more you order at once, the smaller the cost per one unit of material);
- inventory holding costs in the warehouse, e.g. tax per 1 square meter of the storage area or wages for employees handling the materials (i.e. employees who unload, move, organise the inventory and make sure it is kept clean). There is a formula, which can be used to calculate the optimal quantity of the product to order, knowing the one-time cost per order, the demand, the unit purchase price, and the holding cost of the product (measured as a percentage of the unit purchase price). However, the above formula does not include limited storage space and, therefore, does not take into account the mutual relations between other materials. The table below (see Table 3) shows a sample inventory of products stored on a storage yard and their descriptions. For this small warehouse, we will attempt to build a mathematical model, which will serve to determine the quantities of materials to order each month, so that the ordering and holding costs (and, therefore, the warehouse upkeep costs) are at their minimum. One limitation which makes the use of the EOQ formula impossible, is the limited storage area of only 300 sq. m net. This is the yard's storage area less traffic routes and any other type of non-storage area (i.e. where products cannot be stored).

In addition, the following data for our model warehouse has been assumed:
- tax per 1 sq. m of surface area = PLN 0.74;
- one month's salary for a warehouse employee = PLN 2.500;
- average monthly cost of fuel for a forklift = PLN 700.00;
- transportation cost for ordered materials = PLN 1.00 per 1 kilometre.

3.1. Developing the mathematical model

Based on the above parameters, we can now proceed to build the mathematical model of EOQ optimisation. The decision criterion is the minimum possible monthly cost of warehouse operation; therefore, what needs to be calculated are the expenses related to each product (see Fig. 8). The demand for each product is provided based on the statistics from previous years and the warehouse manager's predictions and assumptions.

The cost of the order:
- transportation cost x distance from manufacturer x 2 x 1.3 vehicle depreciation

The cost of storage:
- surface storage unit x [tax + (fuel for forklift + salary for a warehouse employee)/storage area]

The demand for each product is provided based on the statistics from previous years and the warehouse manager's predictions and assumptions. The next step is to determine the range of cells to be changed by Solver (marked in grey) and linking them with costs and the surface area used by each product (see Fig. 9). Let's define the range of cells L19:L29 as 'ilosci" (quantity) and enter the initial formula =L19 x J19 into cell M19. However, as the warehouse products are sold gradually over time, the formula =L19 x J19/2 is much more correct. The same course of action needs to be pursued with the remaining cells in column M. In cell N19, you enter the formula =(H19/L19) x I19 for ordering cost, and copy it into the remaining cells of this column. The 'total cost' column is the sum of costs from columns M and N. Cell N33/34 should sum up the range of cells O19:O29 and be defined as 'koszt_laczny'. The next step is developing an algorithm, which calculates the surface area used by each product stored in the warehouse. In cell P19, you enter the formula =(L19 x F19)/2.

The same goes for other cells of column P. Like in the case of holding costs, it is assumed that if there is 100% of the inventory on the storage yard at the beginning of the period (i.e. between one order and another), there is 0% at the end. Therefore, on average, there are 50% of the products on the yard throughout the entire period.

Naturally, it is possible to make a different assumption which will be more suitable for a particular warehouse. After summing up the surface areas taken up by different materials, the formula =SUM(P19:P29) should be entered into cell P33 and the cell should be defined as 'zuzyta_powierzchnia' (surface used).

Table 3

The inventory of warehouse products stored on a partially covered storage yard

The range of construction materials stored in the storage yard									
Construction material	**Distance from manufacturer [km]**	**Dimensional characteristics w x d x h [mm]**	**Type of unit**	**Unit sizes [m]**	**Manner of storage**	**Surface occupied [m2]**	**Surface area per unit [m2]**	**Price (per piece) [PLN]**	**Quantity per pallet (or in the stack)**
Airbricks MAX	120	188 x 288 x 220	pallet	1.2 x 0.8	up to 3 pallets vertically	1.76	0.587	4.38	96
Airbricks SILKA E24	230	240 x 333 x 199	pallet	1.2 x 0.8	up to 3 pallets vertically	1.76	0.587	6.07	45
Airbricks YTONG	230	Energo 300 x 599 x 199	pallet	1.2 x 0.8	up to 2 pallets vertically	1.76	0.880	15.04	40
Full bricks ordinary	50	Class 25 250 x 120 x 65	pallet	1.2 x 1.2	up to 2 pallets vertically	2.56	1.200	1.48	408
Ceiling beams POROTHERM	120	16 x 6000 x 17,5	piece	6.0 x 0.16	only one stock 5x5	4.80	0.192	247.48	[-]
Flooring blocks POROTHERM	120	250 x 525 x 230	pallet	1.2 x 1.2	only 1 pallet vertically	2.56	2.560	9.83	50
Cement (in sacks)	300	Portland cem. 25 kg R 32.5	sack	[-]	only 1 pallet vertically	2.56	0.073	10.25	35
Ceramic roofing tiles Röben	400	[-]	pallet	1.2 x 0.8	up to 2 pallets vertically	1.76	0.88	4.00	240
Ceramic roofing tiles Braas	60	[-]	pallet	1.2 x 0.8	up to 2 pallets vertically	1.76	0.88	3.60	300
Formwork pine boards (20 mm)	40	200x6000x20	piece	[-]	up to 2 stocks vertically (5x12)	6.00	0.05	11.60	60
Roof battens	40	60 x 40 x 6000	piece	[-]	up to 2 stocks vertically (20x10)	7.20	0.018	9.60	200

Characteristics of building materials stored in the storage yard						
No	Construction material	Surface area per unit	Unit	Demand	Ordering cost	Inventory holding unit cost
[-]		[m^2]	[-]	piece/month	[PLN]	[PLN/piece]
1	Airbricks MAX	0.587	pallet	150	312	6.69
2	Airbricks SILKA E24	0.587	pallet	30	598	6.69
3	Airbricks Ytong	0.880	pallet	60	598	10.04
4	Full Bricks ordinary	1.200	paiiet	20	130	13.69
5	Celling beams POROTHERM	0.192	piece	60	312	2.19
6	Flooring blocks POROTHERM	2.560	pallet	28	312	29.20
7	Cement (in sacks)	0.073	sack	400	780	0.83
8	Ceramic roofing tiles Röben	0.880	pallet	140	1 040	10.04
9	Ceramic roofing tiles Braas	0.880	pallet	80	156	10.04
10	Formwork pine boards (20 mm)	0.050	piece	450	104	0.57
11	Roof battens	0.018	piece	4 500	104	0.21

Fig. 8. The list of stored products with their key characteristics

	L	M	N	O	P	Q
12						
13	SOLVER - nonlinear GRG model (starting point)					
14						
15	Quantity	Inventory holding cost	Ordering cost	Total cost	Surface occupied	Ordering submitted every ...
16						
17	[piece]	[PLN]	[PLN]	[PLN]	[m^2]	[month]
18						
19	150	501.89	312.00	813.89	44.000	1.00
20	30	100.38	598.00	698.38	8.800	1.00
21	60	301.14	598.00	899.14	26.400	1.00
22	20	136.88	130.00	266.88	12.000	1.00
23	60	65.70	312.00	377.70	5.760	1.00
24	28	408.81	312.00	720.81	35.840	1.00
25	400	166.86	780.00	946.86	14.629	1.00
26	140	702.65	1 040.00	1742.65	61.600	1.00
27	80	401.51	156.00	557.51	35.200	1.00
28	450	128.33	104.00	232.33	11.250	1.00
29	4500	461.97	104.00	565.97	40.50	1.00
30						
31						
32						
33		Monthly cost of storage [PLN]		7 822.13	296.0	Warehouse surface [m^2]
34						

Fig. 9. The mathematical model of optimisation (hidden) for the non-convex nonlinear problem

3.2. Solving the ordering policy problem

Another step is entering the data into the dialogue box of the optimisation software. For the warehouse inventory model, the *GRG Nonlinear* model should be chosen. Since it is a non-convex model, the *Use Multistart* check box should be selected in *GRG* options (which

automatically limits the population size to 1 000 and the (iteration) convergence to 0.00001). Other Solver options should be left as default (earlier, such parameters were tested through trial and error). The starting values in 'grey cells' (Fig. 9, range L19:L29) can be any non-negative integers (in practice, they are usually greater than or equal to 1). To avoid the risk of re-scaling, it is recommended not to use starting values that are bigger than the possible future solution. In this specific model, the starting values are based on the monthly demand (Fig. 8, the pink column). The final step is running the optimisation software by clicking *Solve*. The results obtained are presented in Fig. 10.

	L	M	N	O	P	Q
12						
13	SOLVER - nonlinear GRG model (solution)					
14						
15	Quantity	Inventory holding cost	Ordering cost	Total cost	Surface occupied	Ordering submitted every ...
16						
17	[piece]	[PLN]	[PLN]	[PLN]	[m^2]	[month]
18						
19	116	386.97	404.66	791.63	33.924	0.77
20	72	239.58	250.54	490.13	21.004	2.39
21	83	414.97	433.95	848.93	36.380	1.38
22	19	130.45	136.41	266.86	11.436	0.95
23	128	140.01	146.41	286.42	12.274	2.13
24	24	349.24	365.22	714.46	30.617	0.85
25	846	352.79	368.93	721.72	30.928	2.11
26	167	835.94	874.18	1710.11	73.285	1.19
27	49	244.74	255.93	500.67	21.456	0.61
28	396	112.97	118.14	231.11	9.904	0.88
29	2088	214.34	224.15	438.49	18.79	0.46
30						
31						
32						
33		Monthly cost of storage [PLN]		7 000.52	300.0	Warehouse surface [m^2]
34						

Fig. 10. Solution of the problem related to the ordering policy in a building material warehouse

3.3. Conclusions on the problem of the ordering policy

To conclude the analysis of this problem, it is worth mentioning that it is possible to obtain a different end result after changing the *GRG parameters*. This is due to the specific nature of non-convex nonlinear models and the applied *Multistart* algorithm. The problem has been solved dozens of times, with different starting points and different parameters. The lowest value of the costs obtained was PLN 7.000 and 52/100. Therefore, it can be assumed with high probability that the above solution is the unquestionable globally optimal solution for the above-defined problem conditions. Having compared the obtained solution with the

Economic Order Quantity shown in Figure 11, it is clear that the *EOQ* result is slightly better in terms of reducing the total cost, but it does not satisfy the most important and the only constraint – the constraint related to the available storage area.

	L	M	N	O	P	Q
12						
13		Economic Order Quantity (EOQ)				
14						
15	Economic quantities of building materials	Inventory holding cost	Ordering cost	Total cost	Surface occupied	Ordering submitted every ...
16						
17	[piece]	[PLN]	[PLN]	[PLN]	[m^2]	[month]
18						
19	118	395.72	395.72	791.43	34.692	0.79
20	73	245.00	245.00	490.01	21.479	2.44
21	85	424.36	424.36	848.72	37.203	1.41
22	19	133.40	133.40	266.79	11.695	0.97
23	131	143.18	143.18	286.35	12.552	2.18
24	24	357.14	357.14	714.28	31.310	0.87
25	865	360.77	360.77	721.54	31.628	2.16
26	170	854.84	854.84	1709.69	74.942	1.22
27	50	250.27	250.27	500.54	21.941	0.62
28	405	115.52	115.52	231.05	10.128	0.90
29	2135	219.19	219.19	438.38	19.216	0.47
30						
31						
32						
33		Monthly cost of storage [PLN]		6 998.77	306.8	Warehouse surface [m^2]
34						

Fig. 11. Solution of the problem related to Economic Order Quantity (EOQ)

4. Case no 3: the ordering policy in a concrete plant's chain (a network of ready-mixed concrete manufacturers)

A certain company, which produces ready-mixed concrete, has 7 plants located in the south-western part of Poland (see Fig. 12 – green pins). The cement plants, which supply cement to the said company, are marked with red pins in Fig. 12. Each of the 6 cement plants offers a distinct cement selling price, not to mention the additional cost of delivery, which is dependent on the distance between the selling cement plant and the buying concrete plant. The problem is to organise the ordering policy in such a way so that the cost of cement purchase and delivery in each concrete plant is kept at its minimum, and the monthly demand in each concrete plant is satisfied. This problem does not seem complicated until you realise that each cement plant can sell only a limited amount of cement (after all, there are other buyers too), which does not necessarily satisfy the entire demand of the concrete plants.

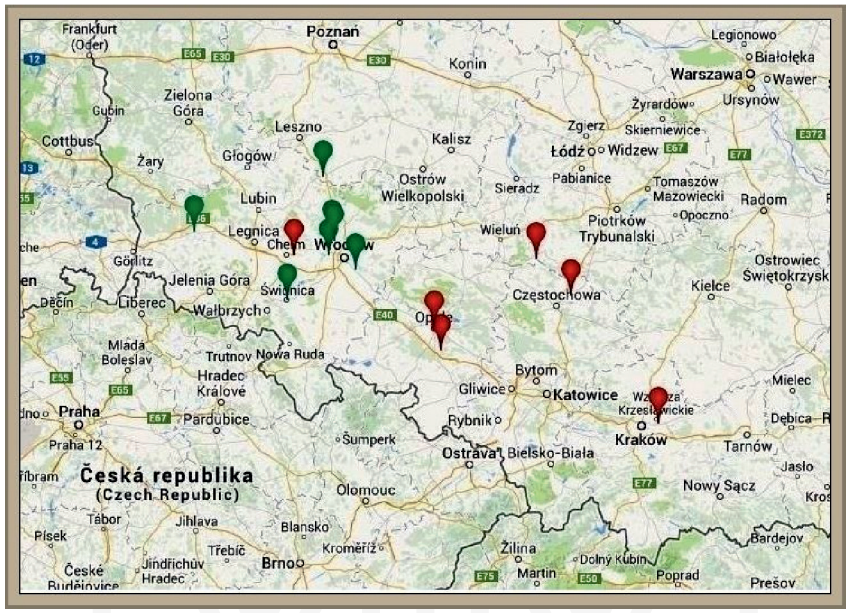


Fig. 12. Concrete plants (green pins) and cement plants (red pins) – location in Poland

4.1. The mathematical model of the ordering policy in the concrete plants

Before developing the model, it is necessary to determine and enter data, which describes the problem. Based on the Google Maps on-line application, it was possible to determine the length of each possible route from the cement plant to the concrete plant (see Fig. 13). The lengths are presented for a one-way route only (i.e. no return). Next, the table (see Fig. 14) should be populated with business parameters for each cement plant. The values of production constraints in each cement plant are provided for the purposes of this model only and do not reflect the real-life production capacity of the above cement plants.

The next step is defining the names for specific ranges of cells:

F20:K20 'cena_za_tone' (price per 1 t = 1000 kg)

F21:K21 'koszt_dostawy' (delivery cost)

F23:K23 'maksymalna_mozliwa_produkcja' (maximum possible production)

The values to be changed by Excel Solver include cement quantities expressed in tones (t is the so-called historical, non-SI unit of measurement) and purchased in particular cement plants by the concrete plants. Since the cement quantities can vary in the range from zero to several hundred tons, it is possible that the solution will be to buy a very small quantity (e.g. 1 or 2 t) in one cement plant and the rest of the cement in other plants. However, such a solution is extremely uneconomical despite having satisfied the conditions of the local extreme. In order to avoid this, the cells changed by Excel Solver include the number of cement batches expressed in tones 't'. The batch size has been determined at 50 t. The cement quantities to purchase are presented in the table of orders (see Fig. 15). The formula '=T29 x porcja'

has been entered into cell F29. The same procedure has been followed for the whole range (F29:K35). As presented in Fig. 15 below, the cement quantities for each row (29÷35) are summed up in red column L.

	A	B	C	D	E	F	G	H	I	J	K
5											
6		**Distances**		**[km]**		CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole
7											
8		Wrocław, Jerzmanowska		**concrete plant**	1	35.9	156	122	162	303	105
9		Wrocław, Betonowa		**concrete plant**	2	62.2	143	120	149	302	103
10		Siechnice		**concrete plant**	3	70.2	133	106	161	288	89
11		Bolesławiec		**concrete plant**	4	72.8	265	214	271	395	197
12		Świdnica		**concrete plant**	5	38.2	202	151	208	332	134
13		Masłowo k/Rawicza		**concrete plant**	6	81.0	184	186	190	368	169
14		Paniowice		**concrete plant**	7	76.1	157	140	163	258	123

Fig. 13. The list of all the distances on all the routes between the concrete and the cement plants

	A	B	C	D	E	F	G	H	I	J	K
17											
18		**Offers**				CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole
19											
20		**price per 1 tone [PLN]**				360.00	360.00	352.00	330.00	315.00	340.00
21		**delivery cost per every km [PLN]**				4.00	4.00	4.00	4.00	4.00	4.00
22		**capacity of cement tanker [t]**				26	26	28	27	26	28
23		**maximum possible production [t]**				**550**	**700**	**600**	**600**	**400**	**500**

Fig. 14. Cement manufacturers' offers including delivery costs

	A	B	C	D	E	F	G	H	I	J	K	L	M
26													
27		**List of orders [t]**				CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole	**Ordered together** [t]	**Monthly demand** [t]
28													
29		Wrocław, Jerzmanowska		**concrete plant**	1	0	0	0	0	0	0	0	100
30		Wrocław, Betonowa		**concrete plant**	2	0	0	0	0	0	0	0	500
31		Siechnice		**concrete plant**	3	0	0	0	0	0	0	0	150
32		Bolesławiec		**concrete plant**	4	0	0	0	0	0	0	0	300
33		Świdnica		**concrete plant**	5	0	0	0	0	0	0	0	400
34		Masłowo k/Rawicza		**concrete plant**	6	0	0	0	0	0	0	0	200
35		Paniowice		**concrete plant**	7	0	0	0	0	0	0	0	350
36													
37		**Production [t]**				**0**	**0**	**0**	**0**	**0**	**0**		

Fig. 15. The list of orders between the concrete plants and the cement plants (starting values)

Column M contains the information on each concrete plant's monthly demand for cement, and the cement quantities produced by each cement plant to satisfy the demand of the concrete plant chain are summed up in row 37. The formula '=SUM(F29:F35)' has been entered into cell F37 and copied into the remaining cells of this range. The next step is defining the names for specific ranges of cells:
L29:L35 'zamowiono_lacznie' (ordered together)
M29:M35 'miesieczne_zapotrzebowanie' (monthly demand)
F37:K37 'produkcja' (production)

To calculate the cost of the purchased cement, the formula '= F29 x cena_za_tone' has been used in cell F42 and copied into the remaining cells of this range (F42:K48). In order to calculate the cost of cement transportation, the cost of a single one-way route covered by a cement tanker has to be determined first (see Fig. 16). In cell F56, the formula '= F10 x koszt_dostawy' has been entered and then copied into other cells. 'Delivery cost per km' (see Fig. 3) includes the cost of the return journey that an empty cement tanker has to cover from the concrete plant to the cement plant. Additionally, it is necessary to determine how many tankers are required to transport a specific quantity of cement (e.g. 200 t), and you need to calculate the total delivery cost. The model assumes that the cement plants have a fleet of cement tankers available, and their load capacities are included in the business parameters (see Fig. 14). The transportation cost formula should include the fact that if the tanker load capacity is e.g. 28 t, the delivery cost of 40 t of cement will amount to:

$$2\times \begin{matrix}\text{the cost of the single travel}\\ \text{for a single cement tanker}\end{matrix} \quad \text{and not} \quad \times \begin{matrix}\text{the cost}\\ \text{of the single travel}\end{matrix} \quad (40\text{ t}/28\text{ t})$$

Due to the reason above, the following formula has been entered into cell T56: '= ROUNDUP((F31/F$24);0)·F56'. Next, the formula has been copied into the rest of the cells from that range (T56:Y62). The final step is summing up the costs and formulating the optimisation criterion. Cell F68 sums up purchase and transportation costs with '=F44+T56', and this has been copied into the rest of the cells in this range. Cell L76 contains the formula for the total ordering cost, which is: '=SUM(F68:K74)' and constitutes the optimisation criterion for this problem.

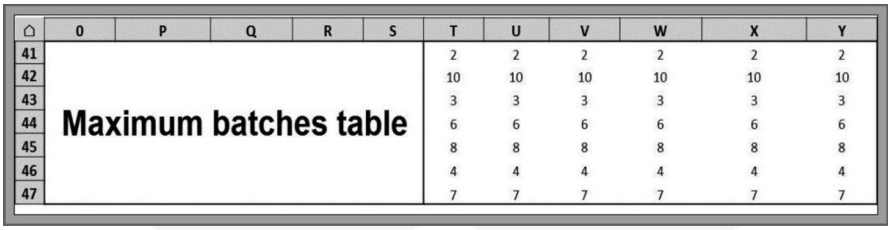
	A	B	C	D	E	F	G	H	I	J	K
53											
54		**Transportation costs table**				CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole
55		**(for a single cement tanker)**									
56		Wrocław, Jerzmanowska			1	144	624	488	648	1 212	420
57		Wrocław, Betonowa			2	249	572	480	596	1 208	412
58		Siechnice			3	281	532	424	644	1 152	356
59		Bolesławiec		**concrete plant**	4	291	1 060	856	**[PLN]** 1 084	1 580	788
60		Świdnica			5	153	808	604	832	1 328	536
61		Masłowo k/Rawicza			6	324	736	744	760	1 472	676
62		Paniowice			7	304	628	560	652	1 032	492

Fig. 16. Transportation costs for a single cement tanker (the list of values)

4.2. Optimisation constraints

This model is nonlinear and most likely non-convex; therefore, global optimisation requires the use of the *Multistart method*. This method is much more effective if lower and upper bounds on the variables changed by Excel Solver are defined in an accurate manner. This is due to the fact that, occasionally, this method finds solutions for the cells which are changed by the software. Unless there are direct constraints defined on the variables, a large

part of such ‘shots’ is misguided and generates false results. By defining direct constraints, it is possible to significantly increase the effectiveness of the optimisation algorithm. In cell T41, the formula: ‘=$M31/porcja’ has been entered and then copied into other cells in the table. It is important to remember that the batch value has to be the divisor for the demand of each concrete plant, otherwise the values in the Table of maximum batches (see Fig. 17) will not be integers. The range of cells (T41:Y47) should be defined as ‘tabela_max_porcji’.



	O	P	Q	R	S	T	U	V	W	X	Y
41						2	2	2	2	2	2
42						10	10	10	10	10	10
43						3	3	3	3	3	3
44						6	6	6	6	6	6
45						8	8	8	8	8	8
46						4	4	4	4	4	4
47						7	7	7	7	7	7

Fig. 17. The list of maximum batches (an important constraint on decision variables)

4.3. Entering the data into Solver and the course of action during optimisation

The Solver’s settings window is shown in the figure below (see Fig. 18). The ‘Use Multistart’ check box should be selected in GRG options. In the ‘Population Size’ box, type at least 100 (but not more than 7 x 6 x 200 = 8400), and in the ‘Random Seed’ box, type the lowest value from the table of maximum batches equal to 2. The iteration convergence should be set at 0.001. In the ‘Derivatives group’ box, select ‘Forward’.

The following course of action has been adopted during optimisation: in order to have a point of reference for the optimisation result, one solution was first determined ‘manually’ based on comparison of the best prices (see Fig. 19). Then, the function *ROUNDUP* was excluded from the transportation costs table (for all tankers), which simplified the optimisation model. After several attempts with the *Multistart method*, a range of results was obtained, for which the ordering costs were calculated while maintaining the real costs of transportation (after the function *ROUNDUP* was re-entered for all tankers into the transportation costs table; see Table 4).

Table 4

The problem of cement ordering policy in a concrete plant – optimisation results

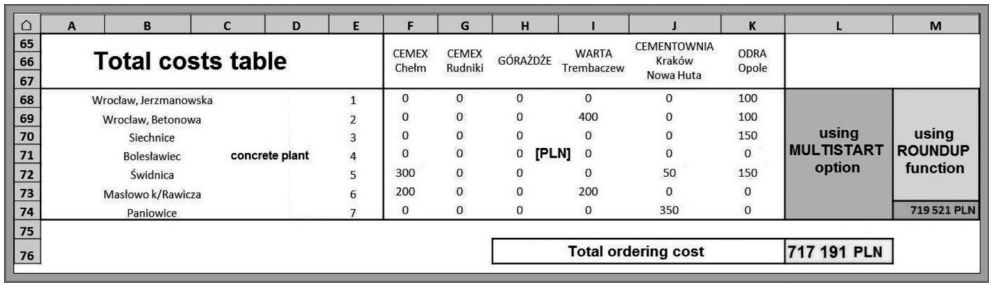
Without ROUNDUP function	After entering ROUNDUP function
717 191 PLN	**719 521 PLN**
717 375 PLN	719 748 PLN
717 933 PLN	720 377 PLN
717 174 PLN	719 529 PLN

Fig. 18. The Excel Solver add-in dialogue box – the concrete plant's chain ordering policy

	A	B	C	D	E	F	G	H	I	J	K	L
65												
66		**Total costs table**				CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole	
67												
68		Wrocław, Jerzmanowska			1	0	0	0	100	0	0	**Solution determined "manually" (a point of reference for optimal result)**
69		Wrocław, Betonowa			2	0	0	0	500	0	0	
70		Siechnice			3	0	0	0	0	0	150	
71		Bolesławiec	**concrete plant**		4	0	0	0	**[PLN]** 0	0	300	
72		Świdnica			5	400	0	0	0	0	0	
73		Masłowo k/Rawicza			6	150	0	0	0	0	50	
74		Paniowice			7	0	0	0	0	350	0	
75												
76								**Total ordering cost**				**721 159 PLN**

Fig. 19. The problem of cement ordering policy in the concrete plants – the 'manual' solution

As expected, it is not necessarily the case that the minimum value on the left corresponds to the minimum value on the right (see Table 4 above). Therefore, the best solution that was obtained is shown in Fig. 20.



	A	B	C	D	E	F	G	H	I	J	K	L	M
65	**Total costs table**					CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole		
66													
67													
68		Wrocław, Jerzmanowska			1	0	0	0	0	0	100		
69		Wrocław, Betonowa			2	0	0	0	400	0	100		
70		Siechnice			3	0	0	0	0	0	150	using MULTISTART option	using ROUNDUP function
71		Bolesławiec	concrete plant		4	0	0	0 [PLN]	0	0	0		
72		Świdnica			5	300	0	0	0	50	150		
73		Masłowo k/Rawicza			6	200	0	0	200	0	0		
74		Paniowice			7	0	0	0	0	350	0		719 521 PLN
75													
76								Total ordering cost				717 191 PLN	

Fig. 20. The problem of cement ordering policy in the concrete plants – the optimal solution

4.4. Conclusions on the problem of the ordering policy

To sum up the case no 3 problem, one question needs to be answered: was this complication necessary? After all, there were earlier optimisation attempts made directly on the model with the *ROUNDUP* function, which is presented in the transportation costs table (for all tankers). However, it should be noted that these results are much worse than the 'manual' solution. This is due to the fact that the above-mentioned function introduces discontinuities resulting from discretisation into the mathematical model. Therefore, optimisation attempts were also made on the discontinuous discrete model, using the evolutionary algorithm. Again, the results, which they yielded were not better than those obtained 'manually' either, even after the attempts that lasted several dozen hours. As shown in previous examples, the *GRG method* is very effective if the optimisation problem is smooth and continuous. In our ordering policy problem, the continuous version is very similar to the discontinuous version (i.e. the one where the *ROUNDUP* function was applied), which suggests that optimal solutions will be similar to one another too. The presented optimisation model has made it possible to solve the problem in a fairly easy way, using the real-life costs. Moreover, the commercial version of the software provided by Frontline Solvers for demonstration purposes was only used in this example. This commercial 'engine' of the *GRG method* is a much more powerful tool than the one used in standard MS Excel worksheet, which, in turn, made it possible to conduct the optimisation directly in the discontinuous model and solve the problem in a much shorter period of time (see Fig. 21).

It is clear that the solution obtained in this way is better by over PLN 320 than the one obtained using the *standard GRG engine* (see Fig. 20, solution using *ROUNDUP* function).

Based on the above-described example, it can be concluded that the optimisation used to solve managerial issues can bring tangible benefits such as increased productivity and savings in terms of costs, time, and material. Problems that are difficult or impossible to solve with the proverbial pen and paper can be solved in a short period of time using the MS Excel spreadsheet and the Excel Solver add-in. This is possible thanks to efficient optimisation algorithms [12] and the high computing power of today's computers, which is more than sufficient in the case of most problems. Although the Excel Solver add-in only

offers the possibility of single-criterion optimisation, it is possible to find a compromise between the two criteria by developing your model and conducting the iterative calculations in an appropriate way (vector optimisation).

	A	B	C	D	E	F	G	H	I	J	K	L
65												
66		**Total costs table**				CEMEX Chełm	CEMEX Rudniki	GÓRAŻDŻE	WARTA Trembaczew	CEMENTOWNIA Kraków Nowa Huta	ODRA Opole	
67												
68		Wrocław, Jerzmanowska			1	50	0	0	0	50	0	**optimum obtained with commercial GRG "engine"**
69		Wrocław, Betonowa			2	0	0	0	400	0	100	
70		Siechnice			3	0	0	0	0	0	150	
71		Bolesławiec		**concrete plant**	4	300	0	0	0	0	0	
72		Świdnica			5	150	0	0	0	0	250	
73		Masłowo k/Rawicza			6	0	0	0	200	0	0	
74		Paniowice			7	0	0	0	0	350	0	
75												
76								**Total ordering cost**				**719 198 PLN**

Fig. 21. The problem of cement ordering policy in the concrete plants – the optimal solution using commercial GRG

5. Final conclusions

Optimisation models constructed using a spreadsheet and Excel Solver software operating within it, which use a user-friendly mathematic programming method, are easy to use and very useful in many areas, starting with construction design and ending with production management. User interface of this optimiser, which has developed greatly in the last decade (evolution techniques were added and 'optimisation engine' was improved to name just a few changes), is so easy to use that it only takes learning to use the spreadsheet on intermediate level to make use of it [12]. Benefits of using Excel Solver are measurable and important, especially in solving logistic, production or even investment related problems. Like with painting or singing, one cannot simply learn modelling from the books or scientific articles. However, even an article and surely a book can provide basic rules and examples, and offer additional practical exercises. With these foundations, an active reader should be able to perform gradual internalisation of the learned problem solutions and finally acquire skills allowing him to use them in 'real-life situations' of a different type or in his work [8–10]. Experience in modelling (the opposite of understanding and respecting its value) can only be gained by practice.

Many problems and relationships can be easily described with numbers and mathematical functions, and intuitive work in a spreadsheet makes it easier to design a model. The possibility to describe a problem mathematically is a prerequisite for the application of mathematical programming (and this is the only trouble spot from managers' perspective). Problems with a large number of variables but with a low degree of complexity (without nested 'if' conditions), are solved much more efficiently than discontinuous problems with a small number of decision variables. If only possible, it is recommended to build linear or smooth nonlinear models. If this condition is met, it will be possible to obtain a solution that is optimal or close to optimal. When building a mathematical model that reflects a particular

problem, it is necessary to have at least basic knowledge of Excel Solver options and the features of various optimisation methods. Default option values do not always suffice to solve a problem, and choosing the wrong method might make it impossible to conduct correct optimization.

Therefore, it is advisable to become familiar with Excel Solver parameters and its optimization methods, which can be done just by reading the information available on the manufacturer's website. It is sometimes the case that the obtained solution, although theoretically optimal model-wise, does not seem to be rational and fit for real life. A solution that is worse (but closer to the optimum) may turn out to be much better because it can be implemented in reality.

Optimisation by way of mathematical programming still requires some rational thinking, and although the real-life relationships can be described mathematically, it is necessary to distinguish those occurring in a spreadsheet from those occurring in reality [4–6].

References

[1] Ragsdale C., *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics*, 7th ed., Cengage Learning, Stamford 2015.

[2] Camm J.D., Cochran J.J., Fry M.J., Ohlmann J.W., Anderson D.R., Sweeney D.J., Williams T.A., *Essentials of Business Analytics*, Cengage Learning, Stamford 2015.

[3] Powell S.G., Baker K.R., *Management Science: The Art of Modeling with Spreadsheets*, 4th ed., Wiley 2013.

[4] Baker K.R., *Optimization Modeling with Spreadsheets*, 2nd ed., Wiley 2011.

[5] Boroń J., Bartyla P., *Optimization of warehouse supply by the methods of mathematical programming*, [in:] *Information systems architecture and technology: systems analysis approach to the design, control and decision support*, edited by J. Świątek [et al.], Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2014, p. 117–126.

[6] Boroń J., *Optimizers in solving managerial and structural problems*, [in:] *Information systems architecture and technology: model based decisions*, edited by J. Świątek [et al.], Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2008, p. 15–25.

[7] Winston W.L., *Operations Research: Applications and Algorithms*, 4th ed., Cengage Learning, Stamford 2003.

[8] Winston W.L., Venkataramanan M., *Introduction to Mathematical Programming: Applications and Algorithms*, Vol 1, 4th ed., Thomson Learning, 28.10.2002).

[9] Gould F.J., Schmidt C.P., Moore J.H., Weatherford L.R., *Introductory Management Science: Decision Modeling With Spreadsheets*, 5th ed. (by Eppen, G.D.), Prentice Hall, New Jersey1998.

[10] Vanderbei R.J., *Linear Programming: Foundations and Extensions*, 4th ed., International Series in Operations Research & Management Science, Vol. 114, Springer Science + Business Media, New York 2014.

[11] Köksalan M., Wallenius J., Zionts S., *An Early History of Multiple Criteria Decision Making*, J. Multi-Crit. Decis. Anal., 20, p. 87–94.

[12] Frontline Solvers, http://www.solver.com (access: 08.09.2016).