International Joint Conference on Environmental and Light Industry Technologies

18-19 November 2010, Budapest, Hungary

Óbuda University

EDUCATION OF DECISION SUPPORTING SYSTEMS

Kornélia Dr. AMBRUS-SOMOGYI

Óbuda University, Budapest

Abstract:

In the higher education these courses are very important, that the students can immediately use in their own work. We can say, the course of Decision Supporting Systems is one of these courses.

We have been teaching Operation Research for technical managers and for environmental protection engineers for years in Hungarian. The name of the subject is Mathematical Programming. Unfortunately, this course is not part of the new curriculum at the managers any longer.

We announced the course of Decision Supporting Systems in English language twice, in the last semester and in this semester. Our educational aims are to give mathematical and informatics definition for economical and organization type of problems, to find the exact and approach solving of these problems. The students get to know the (software) programme packages that help to solve exercices.

Categories, which we teach:

- *Elements of graph theory, shortest path algorithms*
- *Timing problems, critical path method*
- *Linear programming, the graphical and the simplex method*
- *Integer programming, the assignment, the rucksack and the marriage problem*
- *Transportation problem*
- *Introduction to game theory*

In our presentation we show the algorithms of these problems, and give some example, where the students can use these algorithms.

Keywords:

Operation research, shortest path algorithms, timing, linear programming, transportation problem.

1 INTRODUCTION

We have been teaching Operation Research at the university and earlier at the technical college more than fifteen years. At first we taught the managers – for students who specialized in quality management and in bussiness guiding. The name of this subject was Mathematical programming. After the integration of three colleges, on the Keleti Károly Faculty of Business and Management the students in the BA programme Economics and Management and in the BSC programme Engineering Manager have learnt this themes. The name of this courses were Operation research or rather Mathematics III. Unforturnately this course is not part of the current curriculum at the managers.

At present we have been teaching Mathematical programming for the enviromental protection engineers, if the students choose the informatics specialist. There are two courses for students, in the 6th and the 7th semester in Hungarian.

In the last semester and in this semester we announced the course of operation research in English language. The name of the course is Decision Suppporting Systems.

2 TOPICS OF THE COURSE

Our educational aims are to give mathematical and informatics definition for economical and organization type of problems, to find the exact and approach solving of these problems.

We teach the following themes: graph-theory, shortest path algorithms, timing problems, linear and integer programming, transportation problem, game theory.

2.1 Graph theory, shortest path algorithms

The students can solve the practical exercises only if they know the foundations. So we begin with the elements of graph theory: definitions, graph, vertex, edge, path, cycle, degree of points, tree, conectivity, directed graph, weighted graph, etc.The first algorithm is the so called Kruskal greedy algorithm – determinnig the minimum weight spanning tree.

There is a very important – in the traffic – how can we decide the optimal route, how can we minimize the distance or the time of our path. This problem is the so called shortest path problem. We have the following problems connected with shortest path:

- We must decide the length of the shortest path and the shortest path between two points, or from the starting point to the all points of the graph
- We must decide the shortest paths and the lengths of the shortest paths between all points This is the **all-pairs shortest path problem**, in which we have to find the shortest paths between every pair of vertices v, v' in the graph.
- We must decide from a fixed starting point to a fixed endpoint the length of the first k shortest path.
- Look at such a network, where are deficits.We must decide from one starting point to all endpoint the path with minimal deficites. In the multiterminal case: we must decide in the case of all node pairs the minimal deficit paths.
- It is given a fixed point. We must decide the length of the longest path and the longest path to an other fixed point.

There are more algorithms to decide the shortest path. Two multiterminal algorithms are the Warshall(Floyd) and the Bellman(Kalaba) algorithm.

Warshall algorithm

The substance of the algorithm: We look the edges of the graphs, and examine, if we can shorten in the step k=1, 2, …, n with the P_k point.

Definitions:

$\boldsymbol{l_{ij}}$ – the weight of edge from i-th point the j-th point

$\mathbf{Q}^{(0)}$ – starter matrix, it contains the edges (potential matrix), $\mathbf{Q}^{(k)}$ – it contains the length of those shortest paths, they contain the intermediate points at most the 1, 2, …, k points.

$\mathbf{Q}^{(n)}$ – it contains the shortest distances between every two points.

The algorithm is the determination of the $\mathbf{Q}^{(0)} \rightarrow \mathbf{Q}^{(n)}$ potential matrices and the determination of the $\mathbf{R}^{(0)} \rightarrow \mathbf{R}^{(n)}$ label matrices – these given the order of entrance:

<u>0. step</u>: setting of $\mathbf{Q^{(0)}}$ and $\mathbf{R^{(0)}}$

$$q_{ij}^{(0)} = \begin{cases} 0, \text{ if } \mathrm{i} = \mathrm{j} \\ l_{ij}, \text{ if exists (i, j) edge} \\ \infty, \text{ if not exists (i, j) edge} \end{cases} \qquad (1)$$

$$r_{ij}^{(0)} = j$$

<u>1. step</u>: application of 1. point, determination of $\mathbf{Q^{(1)}}$ és $\mathbf{R^{(1)}}$

$$q_{ij}^{(1)} = \min(q_{ij}^{(0)}, q_{i1}^{(0)} + q_{1j}^{(0)})$$

$$r_{ij}^{(1)} = \begin{cases} r_{ij}^{(0)}, \text{ if } q_{ij}^{(0)} \le q_{i1}^{(0)} + q_{1j}^{(0)} \\ r_{i1}^{(0)}, \text{ if } q_{ij}^{(0)} > q_{i1}^{(0)} + q_{1j}^{(0)} \end{cases} \qquad (2)$$

<u>k. step</u>: determination of $\mathbf{Q^{(k)}}$ and $\mathbf{R^{(k)}}$

$$q_{ij}^{(k)} = \min(q_{ij}^{(k-1)}, q_{i1}^{(k-1)} + q_{1j}^{(k-1)})$$

$$r_{ij}^{(k)} = \begin{cases} r_{ij}^{(k-1)}, \text{ if we not improved} \\ r_{ik}^{(k-1)}, \text{ if we call k. point in the path} \end{cases} \qquad (3)$$

<u>n. step</u>: determination of $\mathbf{Q^{(n)}}$ és $\mathbf{R^{(n)}}$

We made a computer program in the spreadsheet Excel with application the Visual Basic programming language[1][2].

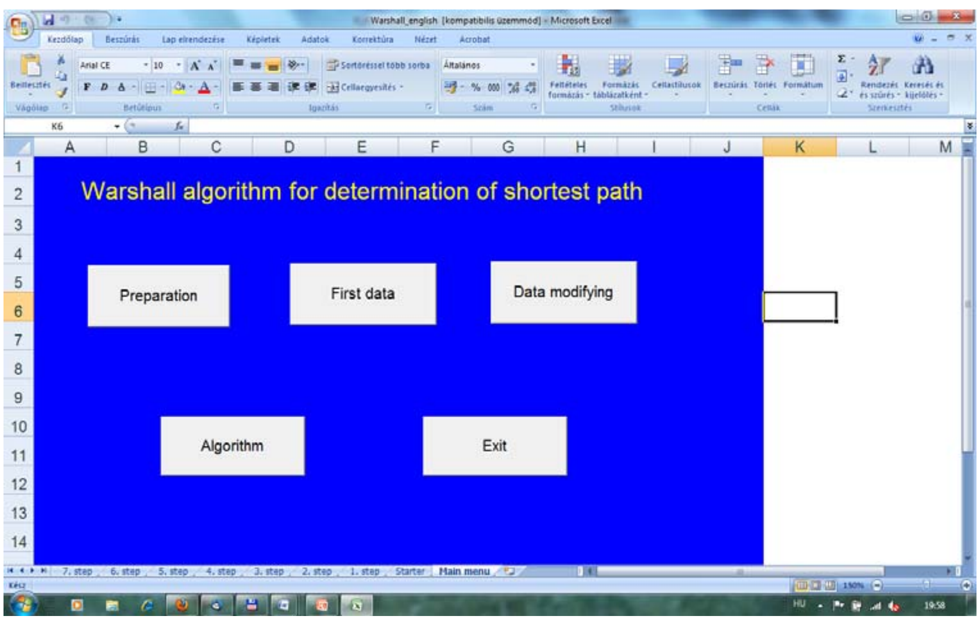
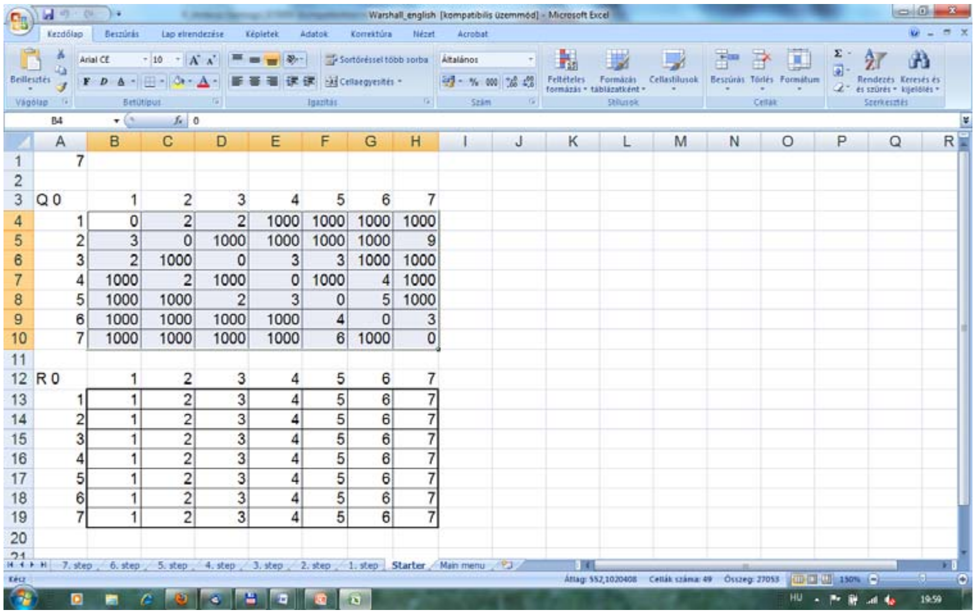
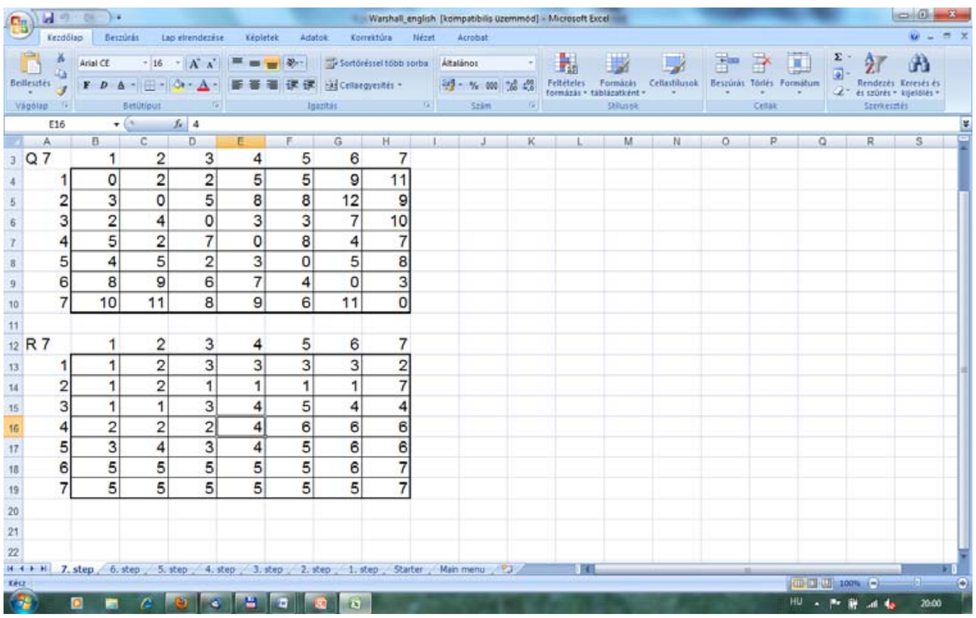


Figure 1: Starting screen – Main menu



7								
Q 0		1	2	3	4	5	6	7
	1	0	2	2	1000	1000	1000	1000
	2	3	0	1000	1000	1000	1000	9
	3	2	1000	0	3	3	1000	1000
	4	1000	2	1000	0	1000	4	1000
	5	1000	1000	2	3	0	5	1000
	6	1000	1000	1000	1000	4	0	3
	7	1000	1000	1000	1000	6	1000	0
R 0		1	2	3	4	5	6	7
	1	1	2	3	4	5	6	7
	2	1	2	3	4	5	6	7
	3	1	2	3	4	5	6	7
	4	1	2	3	4	5	6	7
	5	1	2	3	4	5	6	7
	6	1	2	3	4	5	6	7
	7	1	2	3	4	5	6	7

Figure 2: Starting matrices



Q 7		1	2	3	4	5	6	7
	1	0	2	2	5	5	9	11
	2	3	0	5	8	8	12	9
	3	2	4	0	3	3	7	10
	4	5	2	7	0	8	4	7
	5	4	5	2	3	0	5	8
	6	8	9	6	7	4	0	3
	7	10	11	8	9	6	11	0
R 7		1	2	3	4	5	6	7
	1	1	2	3	3	3	3	2
	2	1	2	1	1	1	1	7
	3	1	1	3	4	5	4	4
	4	2	2	2	4	6	6	6
	5	3	4	3	4	5	6	6
	6	5	5	5	5	5	6	7
	7	5	5	5	5	5	5	7

Figure 3: The shortest paths

Bellman algorithm

The substance of algorithm is founded of the principle of the dynamic programming. It uses the fact: a path is the shortest, if its each sub path is the shortest too.

Here we decide the shortest paths, which have at most 1, 2, …, n-1 edges.

The exercise is the determination of the $\mathbf{Q}^{(1)}$, …, $\mathbf{Q}^{(k)}$, …, $\mathbf{Q}^{(n-1)}$ matrices, where $\mathbf{Q}^{(k)}$ contains the length of the shortest paths which include maximum k edges.

0. step: Setting $\mathbf{Q}^{(1)}$ and $\mathbf{R}^{(1)}$ matrices – here the $\mathbf{Q}^{(1)}$ matrix contains the edges.

$$q_{ij}^{(1)} = \begin{cases} 0, \text{ if } i = j \\ l_{ij}, \text{ if exists (i, j) edge} \\ \infty, \text{ if not exists (i, j) edge} \end{cases} \qquad (4)$$

$$r_{ij}^{(1)} = j$$

1. step: Determination of $\mathbf{Q}^{(2)}$, it contains the path with at most 2 edges.

k. step: Determination of $\mathbf{Q}^{(k)}$ and $\mathbf{R}^{(k)}$

$$q_{ij}^{(k)} = \min_{1 \le f \le n} (q_{if}^{(k-1)} + q_{fj}^{(1)})$$

$$r_{ij}^{(k)} = \begin{cases} r_{ij}^{(k-1)}, \text{if there is no change} \\ r_{if}^{(k-1)}, \text{ if we call f. point into the path} \end{cases} \qquad (5)$$

$$k = 1, 2, \ldots, n-1$$

We made the program of this algorithm too. If we give a graph, the student must write the starting matrices and with the computer program decide the solution and can read the result.

The practical benefit of the shortest path algorithms is by the traffic networks, node resistance.

2.2 Timing problems

The use of the so called network-diagram methods – which are based on the graph-theory – help the optimal decision support, they support the important functions of the management of any company: the job of the planning, the organization, the leading and the controlling.

The network-diagram methods:

- The fundamental models: CPM, PERT (1955-59)
- The developed models: MPM, PEP, MOST, CPM-COST, PERT-COST

The substance of the network method:

- The labour process is phased to details, to actions (works).
- The major states are fixed, they are the events.
- They discover the serial and the parallel connections between actions (events), than they illustrate them with a graph or a matrix.
- The actions(works) receive time-span, resources, and they do the calculation for the concrete model.
- On the ground of the plan - the labour process is organized. During the execution the network would be refreshed so, that the proposed execution time preferably not increases.

The students learn the CPM, PERT and the CPM-COST methods.

CPM is an event-orientated network diagram, the vertices of the graph are the events, the edges of the graph are the actions. The students must not only count the critical path, decide the earlier starting time and the latest finishing time of the actions, but they must edit the CPM network diagram by the action-list too.

2.3 Linear programming

If all inequalities and the objective function are linear, all variables are on the first power, we speak about linear programming exercises. The fundamental linear programming exercise is the so called product mix problem:

- The company produces n different products.
- The number of resource is m.
- a_{ij} : it is the wanted resource of j-th product from i-th resource.
- The capacity of resource is given: b_i : the capacity of i-th resource.
- The revenue (profit) of each products is given: c_j : the revenue of j-th product.
- The number(amount) of each product denotes x_j .
- The question is the maximal profit.

The mathematical description of the exercise:

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n \le b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n \le b_2 \\
&\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n \le b_m \\
&x_1 \ge 0 \\
&\ldots \\
&x_n \ge 0 \\
&c_1x_1 + c_2x_2 + \ldots c_nx_n \to \max
\end{aligned} \tag{6}$$

The methods of solving are:

- in case of two products graphical method,
- simplex-method,
- computer programme packages,
- the Solver of Excel.

The students learn the graphical method and the simplex method too. We made a program to help the acquirement of simplex method [3]. This program shows the steps of simplex method, the steps of basis transformation. The exercises we can solve with the help of Excel, with Solver in both case too.

The example: One company makes 5 sort of products using 3 resources. The wanted resources of the products are following:

$\begin{bmatrix}1\\0\\1\end{bmatrix} \begin{bmatrix}2\\1\\0\end{bmatrix} \begin{bmatrix}1\\1\\1\end{bmatrix} \begin{bmatrix}0\\1\\1\end{bmatrix} \begin{bmatrix}1\\1\\1\end{bmatrix}$. The available capacity: $\begin{bmatrix}100\\80\\50\end{bmatrix}$. The profit of the products: $\begin{bmatrix}2 & 1 & 3 & 1 & 2\end{bmatrix}$.

Determine with the Simplex method or Solver, how many products needed to be produced to reach the maximal profit, and how much the maximal profit is.

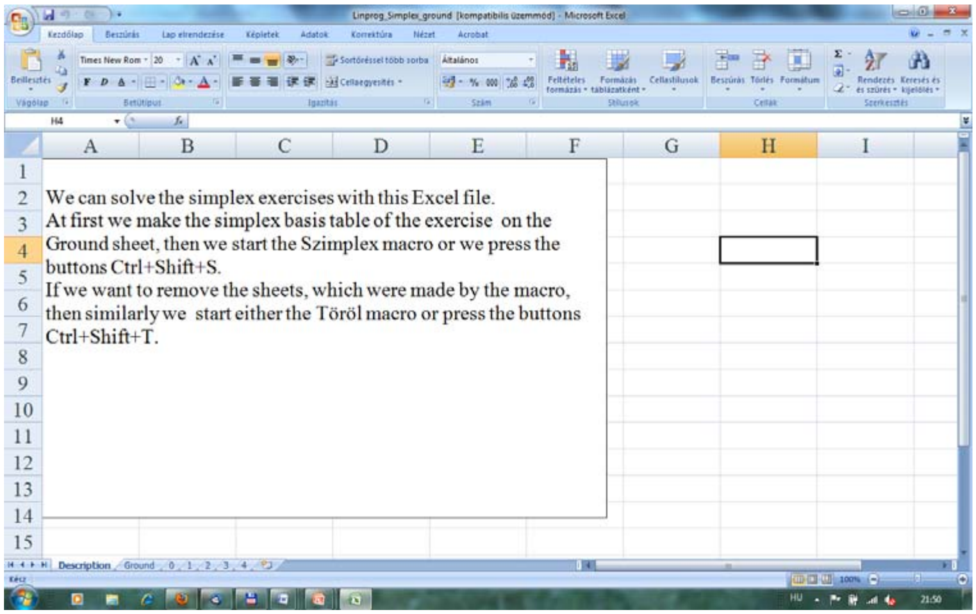
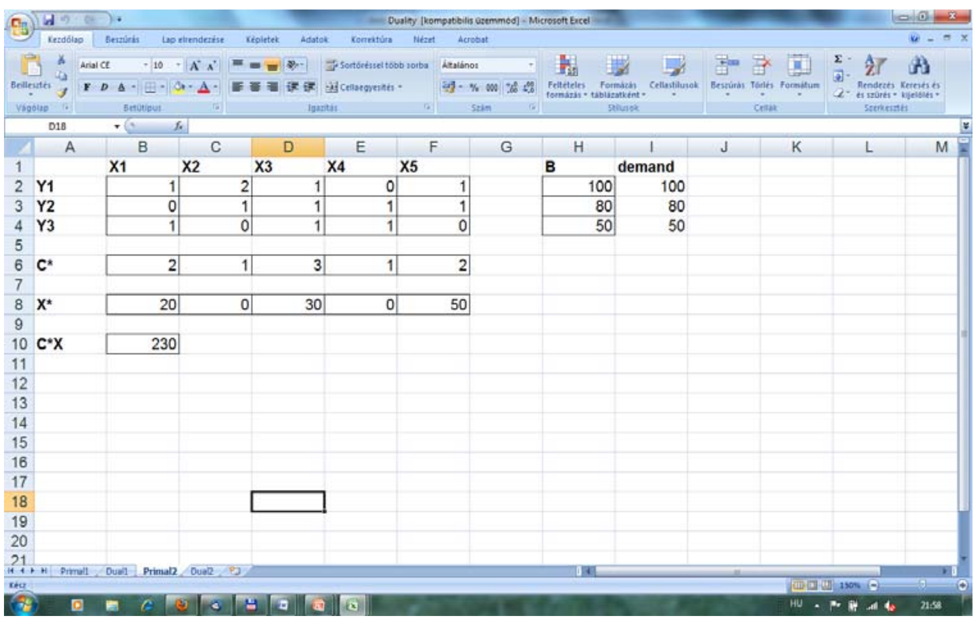


Figure 4: The Help screen of program



	X1	X2	X3	X4	X5		B	demand
Y1	1	2	1	0	1		100	100
Y2	0	1	1	1	1		80	80
Y3	1	0	1	1	0		50	50
C*	2	1	3	1	2			
X*	20	0	30	0	50			
C*X	230							

Figure 5: Solution with Solver

2.4 Integer programming

Assigment problem

Fundamental exercise:

– In a workshop *n* workers are working, and they must do *n* different tasks.
– Any workers can do any tasks, but with different efficiency.

– The aim is to divide the tasks between workers, that the cost of all jobs would be as minimal as possible.

The mathematical model:

x_{ij} signes, that the *j*-th worker does the *i*-th task, let be $x_{ij} = 1$, if the j-th worker does the *i*-th task, x_{ij}=0 otherwise.

Someone does the *i*-th task (but only one): $\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,...,n$.

The *j*-th worker does either tasks: $\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,...,n$.

c_{ij} denotes the cost of the *i*-th task, if the *j*-th worker does it. These numbers are known.

So the exercise is:

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \qquad (7)$$

$$z \to \min$$

This is a special linear programming exercise, because the variables have integer values.

We can incase the variables of the exercises and the elements of cost in matrices. We search an **X** matrix with elements 1 and 0, and the matrix has in each row and in each column only one 1 element.

We trace back the exercise with transformation such a matrix, wich is equivalent with the original from aspect of the optimal solution, but its cost-matrix has quite a lot 0-s.

The method is so called „Hungarian method", Kőnig Dénes and Egerváry Jenő Hungarian mathematicians wrote the process.

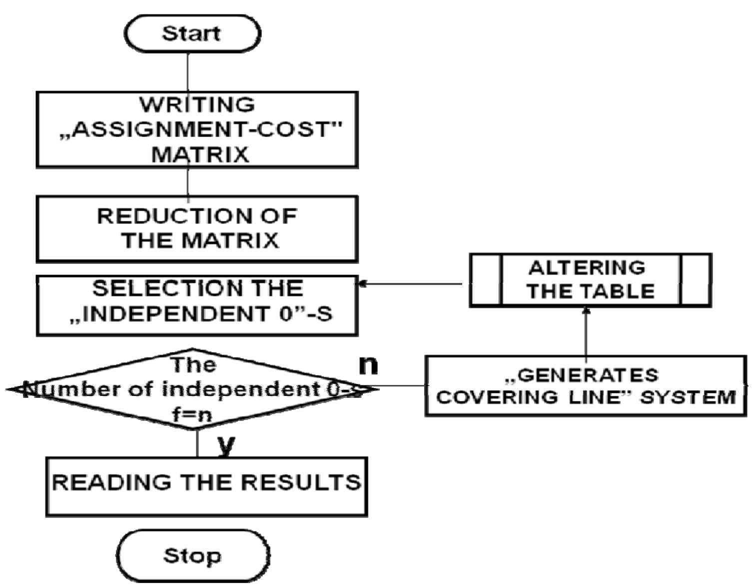


Figure 6: The algorithm of the solution

The rucksack problem

The exercise:

- n things are given,
- the weight of the things are: $a_1, a_2, \ldots a_n$, $j=1,\ldots,n$,
- the use of the things are: $c_1, c_2, \ldots, c_n$,
- the capacity of the rucksack is: K,
- x_i=1, if the tourist takes the j-th thing,   0, if does not take it,
- the exercise is to determine the rucksack with maximal using value

The condition: $\sum_{j=1}^{n} x_j a_j \le K$

$x_j = 0 \text{ or } 1$

The objective function: $z = \sum_{j=1}^{n} x_j c_j \to \max$.

The solution:

- We order the data by descending relative use: $\frac{c_1}{a_1} \ge \frac{c_2}{a_2} \ge \ldots \ge \frac{c_n}{a_n}$
- If r exists , $1 \le r \le n$, that $\sum_{j=1}^{r} a_j = K$, then $x_1 = \ldots = x_r = 1$, $x_{r+1} = \ldots x_n = 0$.
- If such r does not exist, then it isn't sure, that the first r values give the good solution.
- Then we can use the so called **branch and bound** method.

Marriage problem

Exercise:

- n boys and n girls are given,
- A so called sympathy matrix **A** is given, its size is $n*n$. a_{ij}=1, if the i-th boy and the j-th girl sympathize with each other,
- We must decide, if so called **total coupling** exists, such **X** matrix, which elements are: x_{ij}=1, if the i-th boy obtains the j-th girl, 0 otherwise

Conditions:

- Every boy obtains one girl, $\sum_{j=1}^{n} x_{ij} a_{ij} = 1$, $i = 1,\ldots,n$,
- every girl obtains one boy, $\sum_{i=1}^{n} x_{ij} a_{ij} = 1$, $j = 1,\ldots,n$.

The exercise does not always have a solution. The condition of solvability is, that the matrix has n independent 1 elements.

All problems of integer programming can be solved with Excel Solver, the students learn these methods.

2.5 Transportation problem

Fundamental exercise:

- m depots are given, from where we must transport the products, the capacity of the i-th depot is: a_i ,
- n places of destinations (warehouses) are given, where we must transport the products, the capacity (demand) of the j-th warehouse is: b_j .
- The transportation costs from depots to warehouses are given, from the i-th depot to the j-th warehouse the cost is: c_{ij} ,
- we can suppose: $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, in the opposite case we can achieve it by picking-up a nominal depot or a nominal warehouse,
- the transportable amount are denoted with: x_{ij} ,
- we must allocate the transportable amounts with the minimal cost of the transportation .

Conditions:

- we must transport all items, $\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,...,m$,
- we must satisfy all demands, $\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,...,n$,
- we must minimize the cost of transportation. $z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min$.

The methods of solution:

- Simplex method – it has mn variables, it is a special product mix problem.
- Distribution method: After the determination of the starting solution comes the optimization. Finding the starting solution: northwest corner method, selection of the place with the least cost.
- With the Solver of Excel.

We use the distribution method and the Solver for the solution this problem.

2.6 Introduction to game theory

This is the formalization of the real human behavior, where the conflict is the game, and the participants of the conflict are the players Problems:

- Who are the participants of the conflict?
- How do they take part in the conflict?
- What are the possible issues of the game?
- Who and how is interested in the possible issues?

The game can be:

- Game of hazard – the issue of the game depends on coincidence,
- Game of strategy – the human decisions influence the issue of the game.

Strategy: this is a behavior plan of the player, what he does in every situation of the game. The game is finite, if the number of the players and the number of possible strategies is finite. Acquittance or gain function – at the application of single strategies who gains or loses and how much.

We call the twosome(two players), finite, with gain function and zero-sum games **matrix-games**.

The representation of balance – the principle of the warranted gain: the optimal strategy is for any players, which the player should not change, supposing that, the other player does not change the application of his own optimal strategy either.

The warranted gain of the player A:

We determine the gain of the worst case in every row: $f(i) = \min_j a_{ij}$.

We consider the strategy as optimal, where the *f(i)* is the biggest. $\alpha = \max_i f(i) = \max_i \min_j a_{ij}$.

Similarly, the player B look at the loses in the worst case relevantly of each strategy: $g(j) = \max_i a_{ij}$,

After this he considers that strategy optimal, which guarantees him the least loss in the worst case too: $\beta = \min_i g(j) = \min_j \max_i a_{ij}$.

We search the $\alpha = \beta$ balance. If such a point exists, we call the point the saddle point of the matrix – this element is the least in its row and it is the biggest in its column. If such a point exists, than we say the game has a so called **clear strategy.**

If the matrix hasn't got such a point, then the optimal strategy does not exist for the players. In this case we can decide a so called mixed strategy, which is a probability distribution defined on the set of clear strategies. We suppose, that the players play the game several times one after the other. The players apply the clear strategies in turns.

3 CONCLUSIONS

The students like these courses. The results good are enough. My opinion, that in the higher education these courses are very important, that the students can immediately use in their own work. We can say, the course of Decision Supporting Systems is one of these courses. Earlier, in the Hungarian education, the students of the Mathematical programming learned that course sometimes parallel with the practice. They said, it was very good, they understood the discussions at the company.

References

[1] Ambrusné Somogyi, K., Dr. Horváth, A.: GRÁFELMÉLETI ÉS HÁLÓTERVEZÉSI ALGORITMUSOK AZ OPERÁCIÓKUTATÁS OKTATÁSÁBAN, *MATEMATIKA, FIZIKA, SZÁMÍTÁSTECHNIKA FŐISKOLAI OKTATÓK XXVIII. KONFERENCIÁJA*, NYÍREGYHÁZA 2004. AUGUSZTUS 25-27

[2] Ambrusné Somogyi, K.: MATEMATIKAI PROGRAMOZÁS – OPERÁCIÓKUTATÁS. MIT ÉS HOGYAN? ,*MATEMATIKA, FIZIKA ÉS SZÁMÍTÁSTECHNIKA OKTATÓK XXX. KONFERENCIÁJA*, PÉCS 2006. AUGUSZTUS 23-25. CD PROCEEDINGS P. 8.

[3] Ambrusné Somogyi, K., Kiss, L.: MATEMATIKAI PROGRAMOZÁSI PROGRAMCSOMAG, FELSŐFOKÚ *ALAPKÉPZÉSBEN MATEMATIKÁT, FIZIKÁT ÉS INFORMATIKÁT OKTATÓK XXXII. KONFERENCIÁJA*, KECSKEMÉT, 2008. AUGUSZTUS 25-27.CD PROCEEDINGS PP. 7-14.

Corresponding author:

Kornélia Dr. AMBRUS-SOMOGYI
Institute of Media Technology and Light Industry
Faculty of Light Industry and Environmental Protection Engineering
Óbuda University
Doberdó út 6.
1034, Budapest
Hungary
Phone: +36 20 566 8726 fax: +36 1 666 5922
e-mail: a_somogyi.kornelia@rkk.uni-obuda.hu