Linear Programming with Computer
Lineární programování s využitím počítače

Jiří Neubauer

Abstract: Linear programming is a mathematical discipline solving optimization problems in different areas. It uses especially the knowledge of linear algebra. The contribution presents selected useful software (mostly freeware) that can solve linear programming problems efficiently.

Abstrakt: Lineární programování je matematická disciplína, sloužící k řešení optimalizačních úloh z různých oblastí, využívá k tomu poznatky především z oblasti lineární algebry. V článku je představeno několik užitečných programů, většinou volně šiřitelných, pomocí nichž lze úlohy lineárního programování efektivně řešit.

1. Introduction

Linear programming is a mathematical discipline solving optimization problems in different areas. It uses especially the knowledge of linear algebra. During and just after World War II, linear programming was developed for mathematically solving certain kinds of resource allocation problems. Since then, this technique has been applied to various kinds of problems and has become an important tool for operational research (see Dantzig (1963), Ecker a Kupferschmid (2004) or Jablonský (2002)).

Linear programming is taught at many universities (technical, economic, managerial, etc.). Simple models can usually be solved by manual calculation. Principles of the simplex algorithm can be shown and explained step by step. In the case of complex models, manual calculation is not practical and is not effective from a didactic point of view. The use of appropriate software is obvious. Professional software is not usually necessary for teaching purposes. These products are often rather expensive. In addition, it is helpful if students can try and test the software on their own computers and not be limited to the school computer laboratories. The contribution presents selected useful software (mostly freeware) that can solve linear programming problems efficiently. Software applications are presented on a simple example (a material cutting problem).

2. Simplex algorithm – example

We present the calculation of a simple linear programming problem using the simplex algorithm (the two-phase method) on the example of material cutting problem (see Jablonský (2002)).

The paper mill produces paper rolls with a length of 500 meters in a standard width of 2 meters. Some customers prefer a different width rolls – specifically 0.5 m, 0.8 m and 1.2 m. Paper rolls of these widths are obtained from the standard size rolls by suitable cutting. In accordance with the certain contract they must produce at least 500 pieces of rolls with the width of 0.5 m, 1000 rolls of the width of 0.8 m and 400 pieces of rolls with the width of 1.2 m. We can find five different ways of basic role cutting (see table 1). The last row of the table contains the potential waste corresponding to each variant of the basic paper role division.

Variant	**I**	**II**	**III**	**IV**	**V**
0,5 m [pcs]	4	2	1	0	0
0,8 m [pcs]	0	1	0	2	1
1,2 m [pcs]	0	0	1	0	1
Waste [m]	0	0,2	0,3	0,4	0

Table 1: The variants of basic role (2 meters) cutting

The mathematical model of this simple example contains a non-negative variables $x_1, \ldots, x_5$ ($x_i \geq 0, i = 1 \ldots, 5$), where the variable x_1 is the number of the basic 2 meter rolls that are cut by the variant I, analogically the variables $x_2, \ldots, x_5$ determine the number of rolls cut by variants II, …, V. Thus, for each width requirement we get following constraints.

$$\begin{array}{lr}
4x_1 + 2x_2 + x_3 \qquad\qquad \geq \;\; 500 & \text{(for 0.5 m)} \\
x_2 \quad + 2x_4 + x_5 \geq 1000 & \text{(for 0.8 m)} \\
x_3 \qquad + x_5 \geq \;\; 400 & \text{(for 1.2 m)}
\end{array}$$

The objective function reflects the number of the basic rolls, ie. the sum of all variables. We want to find the minimum of this function.

$$z = x_1 + x_2 + x_3 + x_4 + x_5 \rightarrow \min$$

This model can be solved using the simplex method (the method of artificial variables – the two-phase method or the dual simplex method). We present a solution using the artificial variables (the two-phase method). At first, we convert the system of linear inequalities to the system of linear equations by *surplus variables* $x'_1 \geq 0, x'_2 \geq 0$ and $x'_3 \geq 0$. Then we employ *artificial variables* $x''_1 \geq 0, x''_2 \geq 0$ and $x''_3 \geq 0$ to get the system in the *canonical form.* We add the second objective function $z' = x''_1 + x''_2 + x''_3$. We will try to find the minimum of this function in the first phase of the method, the aim is to achieve zero value of the function z' (see for example Ecker a Kupferschmid (2004) or Jablonský (2002)). We get the following model.

$$\begin{array}{rcl}
4x_1 + 2x_2 + x_3 \qquad - x'_1 \qquad + x''_2 & = & 500 \\
x_2 + 2x_4 + x_5 \qquad - x'_2 \qquad + x''_2 & = & 1000 \\
x_3 + x_5 \qquad - x'_3 \qquad + x''_3 & = & 400 \\
z - x_1 - x_2 - x_3 - x_4 - x_5 & = & 0 \\
z' \qquad\qquad - x''_1 - x''_2 - x''_3 & = & 0
\end{array}$$

The last modification of the model consists in adding the last line (the objective function z') with equations containing artificial variables (in our case with the first three equations). This adjustment ensures that the model is in the canonical form.

$$\begin{array}{rcl}
4x_1 + 2x_2 + x_3 \qquad - x'_1 \qquad + x''_2 & = & 500 \\
x_2 + 2x_4 + x_5 \qquad - x'_2 \qquad + x''_2 & = & 1000 \\
x_3 + x_5 \qquad - x'_3 \qquad + x''_3 & = & 400 \\
z - x_1 - x_2 - x_3 - x_4 - x_5 & = & 0 \\
z' + 4x_1 + 3x_2 + 2x_3 + 2x_4 + 2x_5 - x'_1 - x'_2 - x'_3 & = & 0
\end{array}$$

The first phase of the simplex algorithm (minimization of the objective function z') is shown in table 2. The pivot positions are marked by the frame. The optimal value of the objective function

z' (in our case the minimum) is achieved when all the coefficients in the objective function row are not positive. If not, we find the largest positive number in this row (we identify the pivot column – the entering variable). Furthermore, we determines the proportion of right-hand side coefficients β_i (the last column of the simplex tableau) and positive coefficients in the pivot column and we identify the smallest one among them (see Dantzig (1963) or Jablonský (2002)). This number determines the pivot line – the leaving variable. The intersection of the pivot column and row determine the pivot position. In our example, the pivot is 4, it means that in the next step of the simplex algorithm calculation (iteration) we obtain the basic solution where the variable x_1 is basic (the entering variable) and the originally basic variable x_1'' becomes non-basic (the leaving variable). After three iterations we achieve the optimal solution (in terms of the objective function z'), the value of z' is zero. We obtain the feasible basic solution (the initial solution was not feasible).

The first phase of the calculation ends at this point. We omit the objective function z' together with the artificial variables x_1'', x_2'' and x_3'' from the model – see table 3. All coefficients in the objective function row are not positive, so in terms of minimization we have come to the optimal solution $x_1 = 120$, $x_4 = 300$ and $x_5 = 400$ with the objective function value $z = 825$. The remaining variables are zero. The minimum number of basic two-meter parer rolls is 825. This can be achieved if we cut 125 basic rolls by the variant I, 300 rolls by the variant IV and 400 basic rolls by the variant V. We can see that there is zero coefficient in the column of the non-basic variable x_2 in the row of the objective function z (table 3). If we choose this variable as entering and identify the leaving variable as usual, we get the different basic solution that has the same objective function value $z = 825$. This solutions is also optimal (the alternative solution). The same minimum number of basic paper rolls cut can be achieved when using 250 times the variant II ($x_2 = 250$), 175 times the variant IV ($x_4 = 175$) and 400 times the variant V ($x_5 = 400$).

basis	x_1	x_2	x_3	x_4	x_5	x_1'	x_2'	x_3'	x_1''	x_2''	x_3''	β_i	β_i/α_{ik}
x_1''	$\boxed{4}$	2	1	0	0	−1	0	0	1	0	0	500	500
x_2''	0	1	0	2	1	0	−1	0	0	1	0	1000	—
x_3''	0	0	1	0	1	0	0	−1	0	0	1	400	—
z	−1	−1	−1	−1	−1	0	0	0	0	0	0	0	min
z'	4	3	2	2	2	−1	−1	−1	0	0	0	1900	min
x_1	1	1/2	1/4	0	0	−1/4	0	0	1/4	0	0	125	—
x_2''	0	1	0	2	1	0	−1	0	0	1	0	1000	1000
x_3''	0	0	1	0	$\boxed{1}$	0	0	−1	0	0	1	400	400
z	0	−1/2	−3/4	−1	−1	−1/4	0	0	1/4	0	0	125	min
z'	0	1	1	2	2	0	−1	−1	−1	0	0	1400	min
x_1	1	1/2	1 4	0	0	−1/4	0	0	1/4	0	0	125	—
x_4	0	1	−1	$\boxed{2}$	0	0	−1	1	0	1	−1	600	300
x_3''	0	0	1	0	1	0	0	−1	0	0	1	400	—
z	0	−1/2	1/4	−1	0	−1/4	0	−1	1/4	0	1	525	min
z'	0	1	−1	2	0	0	−1	1	−1	0	−2	600	min
x_1	1	1/2	1/4	0	0	−1/4	0	0	1/4	0	0	125	
x_4	0	1/2	−1/2	1	0	0	−1/2	1/2	0	1/2	−1/2	300	
x_5	0	0	1	0	1	0	0	−1	0	0	1	400	
z	0	0	−1/4	0	0	−1/4	−1/2	−1/2	1/4	1/2	1/2	825	min
z'	0	0	0	0	0	0	0	0	−1	−1	−1	0	min

Table 2: The first phase of the algorithm

basis	x_1	x_2	x_3	x_4	x_5	x'_1	x'_2	x'_3	β_i	β_i/α_{ik}
x_1	1	$\boxed{1/2}$	$1/4$	0	0	$-1/4$	0	0	125	250
x_4	0	$1/2$	$-1/2$	1	0	0	$-1/2$	$1/2$	300	600
x_5	0	0	1	0	1	0	0	-1	400	—
z	0	0	$-1/4$	0	0	$-1/4$	$-1/2$	$-1/2$	825	min
x_2	2	1	$1/2$	0	0	$-1/2$	0	0	250	
x_4	-1	0	$-3/4$	1	0	$1/\ 4$	$-1/2$	$1/\ 2$	175	
x_5	0	0	1	0	1	0	0	-1	400	
z	0	0	$-1/4$	0	0	$-1/4$	$-1/2$	$-1/2$	825	min

Table 3: The second phase of the algorithm

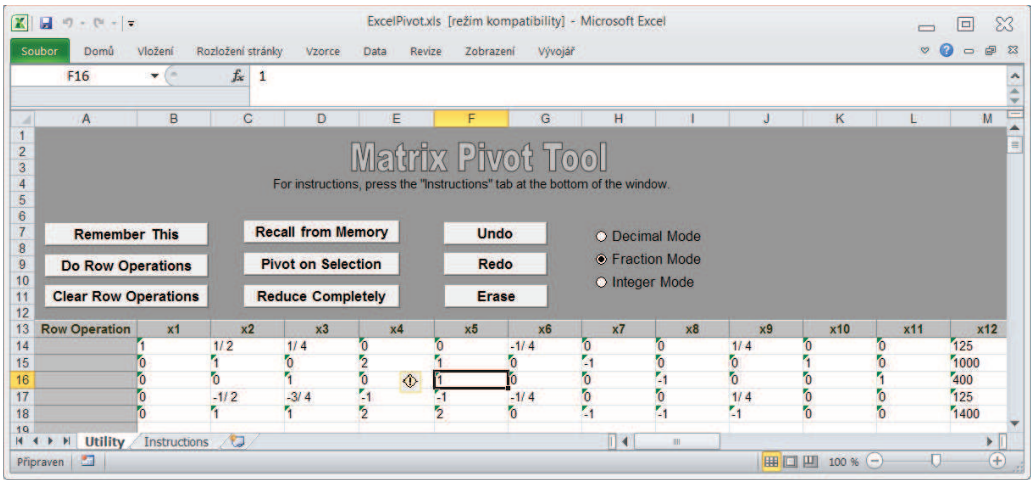
3. Selected Software for Linear Programming

ExcelPivot Utility

Several simplex tableau recalculations are needed during the described computation. Manual processing (using a calculator) can be rather lengthy. The calculation can be accelerated by using an Excel utility ExcelPivot.xls. This application is available along with other useful information on the website `http://www.zweigmedia.com/RealWorld/Summary4.html`.

Firstly, we write the simplex tableau into the excel file, then we select the pivot position and click on the button "Pivot on Selection"which causes simplex table recalculation (see figure 1). You can select whether the numbers in the table are decimal (Decimal Mode), fractions (Fraction Mode) or an integer (Integer Mode). We do not use the last option in our case. This application is not suitable only for the recalculation of the simplex tableau, you can use it to perform different elementary row operations in the systems of equations (addition, subtraction, multiplication of equations, etc.). A brief guide is available in this Excel file too.

It should be noted that the this utility does not always works correctly (a problem with the step back (the button"Undo"in fractional mode), etc.). These problems may be caused by the incompatibility of the version with higher versions of Microsoft Office. There is also an online version (`http://www.zweigmedia.com/RealWorld/tutorialsf1/scriptpivot2.html`) without these bugs. It is a useful tool for teaching purposes that allows teachers or students to speed up the calculations and focus mainly on principles of the simplex algorithm.



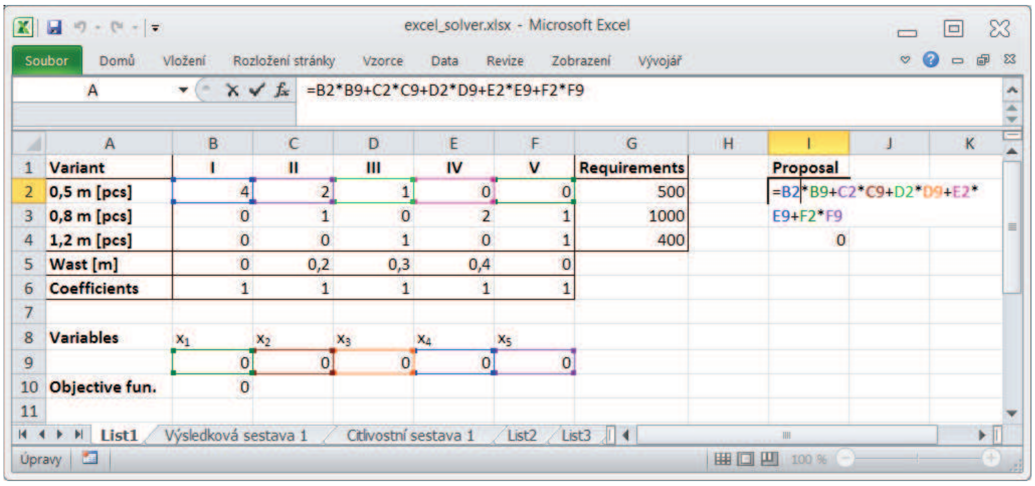
Matrix Pivot Tool

For instructions, press the "Instructions" tab at the bottom of the window.

Decimal Mode [✓ Fraction Mode] Integer Mode

Row Operation	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
	1	1/ 2	1/ 4	0	0	-1/ 4	0	0	1/ 4	0	0	125
	0	1	0	2	1	0	-1	0	0	1	0	1000
	0	0	1	0	1	0	0	-1	0	0	1	400
	0	-1/ 2	-3/ 4	-1	-1	-1/ 4	0	0	1/ 4	0	0	125
	0	1	1	2	2	0	-1	-1	-1	0	0	1400

Figure 1: The utility ExcelPivot.xls

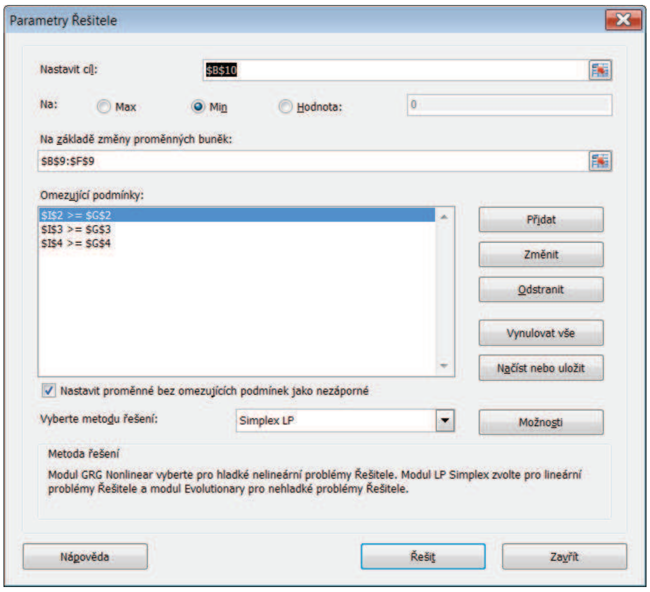

=B2*B9+C2*C9+D2*D9+E2*E9+F2*F9

	A	B	C	D	E	F	G	H	I
1	Variant	I	II	III	IV	V	Requirements		Proposal
2	0,5 m [pcs]	4	2	1	0	0	500		=B2*B9+C2*C9+D2*D9+E2*E9+F2*F9
3	0,8 m [pcs]	0	1	0	2	1	1000		
4	1,2 m [pcs]	0	0	1	0	1	400		0
5	Wast [m]	0	0,2	0,3	0,4	0			
6	Coefficients	1	1	1	1	1			
7									
8	Variables	x_1	x_2	x_3	x_4	x_5			
9		0	0	0	0	0			
10	Objective fun.	0							

Figure 2: Solution in Excel using Add-in "Solver"

Excel – Solver

Excel, as a part of Microsoft Office, is not necessary to introduce. Excel is not freely available but the vast majority of users is familiar with it and can use it. This spreadsheet includes a tool called "Solver" that can be used to solve linear programming problems. We briefly describe the way how to use this tool on the previously solved example. The basic specification of the problem is shown in figure 2. The table contains possible variants of paper roll cutting supplemented with the row containing the objective function coefficients (cells from B6 to F6). The values of the variables $x_1, \ldots, x_5$ (set to zero now) are in the cells B9 – F9.


Parametry Řešitele

Nastavit cíl: $B$10

Na: Max [✓ Min] Hodnota: 0

Na základě změny proměnných buněk: $B$9:$F$9

Omezující podmínky:
$I$2 >= $G$2
$I$3 >= $G$3
$I$4 >= $G$4

[✓] Nastavit proměnné bez omezujících podmínek jako nezáporné

Vyberte metodu řešení: Simplex LP

Metoda řešení
Modul GRG Nonlinear vyberte pro hladké nelineární problémy Řešitele. Modul LP Simplex zvolte pro lineární problémy Řešitele a modul Evolutionary pro nehladké problémy Řešitele.

Figure 3: "Solver settings"

We insert the formula =B2*B9+C2*C9+D2*D9+E2*E9+F2*F9 into the cell I2. In Czech version of the program we can also use the command SOUČIN.SKALÁRNÍ(B2:F2;B9:F9)[1]. It expresses the number of 0.5 m roles corresponding to the values of variables in the cells B9

[1]In English version of Excel use the command SUMPRODUCT(B2:F2;B9:F9)

– F9. Analogically, we insert the command SOUČIN.SKALÁRNÍ(B3:F3;B9:F9), respectively SOUČIN.SKALÁRNÍ(B4:F4;B9:F9) into the cell I3, respectively I4. These values correspond to the number of rolls with width 0.8 m, respectively 1.2 m. The objective function value can be calculated by B6*B9+C6*C9+D6*D9+E6*E9+F6*F9 or SOUČIN.SKALÁRNÍ(B6:F6;B9:F9).

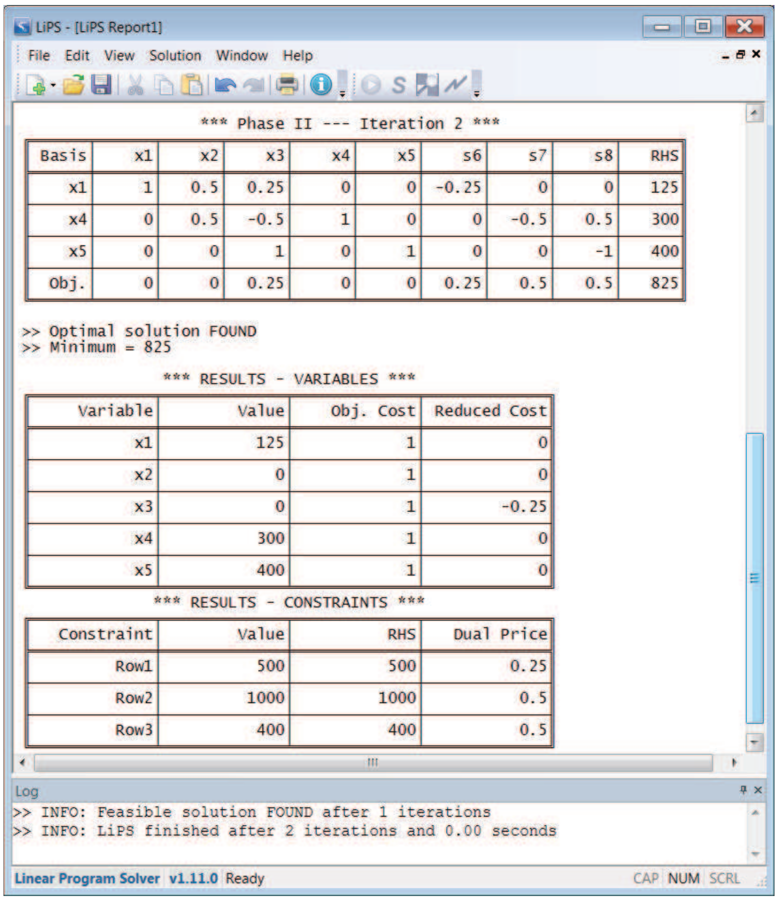
A dialog box appears after running Add-in "Solver"(in Excel 2010 the Solver command is available in the Analysis group on the Data tab). The setup is shown in figure 3. We obtain the solution $x_1 = 0$, $x_2 = 250$, $x_3 = 0$, $x_4 = 175$ and $x_5 = 400$ with the value of the objective function 825. We can get more detailed results from the answer report. One of the possible outputs is the sensitivity analysis (sensitivity report).

Linear Program Solver

Linear Program Solver (LiPS) is an optimization package intended for solving linear, integer and goal programming problems. This freely available software can be downloaded from the web pages `http://sourceforge.net/projects/lipside/`. We can specify a new model for example as follows. At first, we create a new model (File → New → Text Model), then we write the model in the form described bellow.

```
Min: x1+x2+x3+x4+x5;

4*x1+2*x2+x3>=500;
x2+2*x4+x5>=1000;
x3+x5>=400;
```



*** Phase II --- Iteration 2 ***

Basis	x1	x2	x3	x4	x5	s6	s7	s8	RHS
x1	1	0.5	0.25	0	0	-0.25	0	0	125
x4	0	0.5	-0.5	1	0	0	-0.5	0.5	300
x5	0	0	1	0	1	0	0	-1	400
Obj.	0	0	0.25	0	0	0.25	0.5	0.5	825

```
>> Optimal solution FOUND
>> Minimum = 825
```

*** RESULTS - VARIABLES ***

Variable	Value	Obj. Cost	Reduced Cost
x1	125	1	0
x2	0	1	0
x3	0	1	-0.25
x4	300	1	0
x5	400	1	0

*** RESULTS - CONSTRAINTS ***

Constraint	Value	RHS	Dual Price
Row1	500	500	0.25
Row2	1000	1000	0.5
Row3	400	400	0.5

```
Log
>> INFO: Feasible solution FOUND after 1 iterations
>> INFO: LiPS finished after 2 iterations and 0.00 seconds
```

Figure 4: Linear Program Solver

We can run optimization by pressing F5 key (or click on the green arrow button or via the menu LiPS → Solve Model). The output is the window containing the simplex tables corresponding to the successive iterations. It should be noted that the input table of the two-phase method is different from the table that was used for manual calculation. The program, prior to the optimization, attempts to find such structural variables that could be basic (fulfilling non-negativity constraints). The final table (see figure 4) is identical to table 3 – upper part (except for signs in the objective function). Using manual calculation we have found two solutions. This software offers only one of them. Sensitivity analysis is one of the possible outputs too.

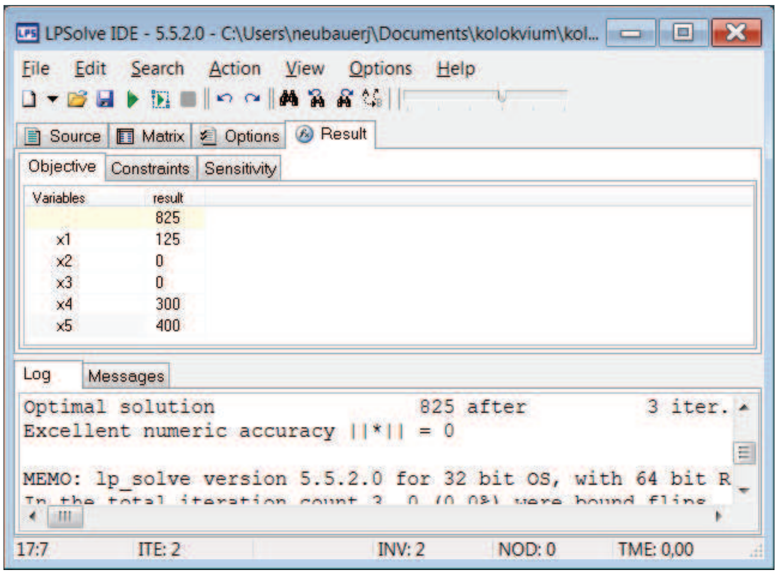
LPSolve IDE

LPSolve solves pure linear, (mixed) integer/binary, semi-continuous and special ordered sets models. This is also a freely available software, it can be obtained from the website `http://sourceforge.net/projects/lpsolve/`. Model definition is similar to LiPS.

```
/* Objective function */
min:x1+x2+x3+x4+x5;

/* Variable bounds */
4 x1+2 x2+x3>=500;
x2+2 x4+x5>=1000;
x3+x5>=400;
```

Optimization can be started by pressing F9 or by clicking on the green triangle button in the menu bar or via the menu (Action → Solve). The solution contains the values of the variables (Result tab – Objective), the values of individual constraints (Result tab – Constraints) and sensitivity analysis (Result tab – Sensitivity). Each steps of iterations, as can be obtained from the software LiPS, are not available. The program offers a number of possible options and settings of optimization procedures. Using LPSolve we get the same (one) optimal solutions as we have obtained from LiPS.



Variables	result
	825
x1	125
x2	0
x3	0
x4	300
x5	400

```
Optimal solution                    825 after          3 iter.
Excellent numeric accuracy ||*|| = 0

MEMO: lp_solve version 5.5.2.0 for 32 bit OS, with 64 bit R
```

```
17:7          ITE: 2                    INV: 2        NOD: 0        TME: 0,00
```

Figure 5: LPSolve IDE

Linear programming in R

Computational environment R (http://www.r-project.org/) offers variety of procedures and method for statistical data analysis. It is not primarily designed for solving optimization problems. However, with packages lpSolve or lpSolveAPI we can solve linear programming problems. The following script describes possible way of the calculation (the paper rolls cutting example).

```
> install.packages("lpSolve")
> library(lpSolve)
> f.obj<-c(1,1,1,1,1)
> f.con<-matrix(c(4,2,1,0,0,0,1,0,2,1,0,0,1,0,1),nrow=3,byrow=TRUE)
> f.dir <- c(">=", ">=",">=")
> f.rhs <- c(500,1000,400)
> lp ("min", f.obj, f.con, f.dir, f.rhs)
Success: the objective function is 825
> lp ("min", f.obj, f.con, f.dir, f.rhs)$solution
[1] 125   0   0 300 400
```

We have come to the same result as in the previously mentioned software. Sensitivity analysis is also available. A possible alternative is to use package lpSolveAPI (see the following script).

```
> library("lpSolveAPI")
> my.lp <- make.lp(3, 5)
> set.column(my.lp, 1, c(4, 0, 0))
> set.column(my.lp, 2, c(2, 1, 0))
> set.column(my.lp, 3, c(1, 0, 1))
> set.column(my.lp, 4, c(0, 2, 0))
> set.column(my.lp, 5, c(0, 1, 1))
> set.objfn(my.lp, c(1,1,1,1,1))
> set.constr.type(my.lp, rep(">=", 3))
> set.rhs(my.lp, c(500, 1000, 400))
> my.lp
Model name:
            C1    C2    C3    C4    C5
Minimize     1     1     1     1     1
R1           4     2     1     0     0  >=   500
R2           0     1     0     2     1  >=  1000
R3           0     0     1     0     1  >=   400
Kind       Std   Std   Std   Std   Std
Type      Real  Real  Real  Real  Real
Upper      Inf   Inf   Inf   Inf   Inf
Lower        0     0     0     0     0
> solve(my.lp)
[1] 0
> get.objective(my.lp)
[1] 825
> get.variables(my.lp)
[1] 125   0   0 300 400
> get.primal.solution(my.lp)
[1]  825  500 1000  400  125    0    0  300  400
```

```
> get.dual.solution(my.lp)
[1] 1.00 0.25 0.50 0.50 0.00 0.00 0.25 0.00 0.00
```

4. Conclusion

The contribution deals with possible application of freely available software (utility) that can speed up the learning process of the simplex method. We will let the reader assess their advantages or disadvantages. The goal was not to give a complete list of available software, but rather to point out what these programs can offer. Unfortunately, numerical calculations have become significant obstacles for many students. It is not advisable to spend time just with simplex tableau recalculations, the emphasis should be focused on the student's ability to build mathematical models of selected problems. On the other hand, it is essential that students can understand the software outputs, they must be adequately familiar with the basic calculation principles.

References

BERKELAAR, Michel et al. lpSolve: Interface to Lp_solve v. 5.5 to solve linear/integer programs. R package version 5.6.6, 2011. http://CRAN.R-project.org/package=lpSolve.

DANTZING, George B. *Linear Programming and Extensions*. Princeton: Princeton University Press, 1963.

ECKER, Joseph G., KUPFERSCHMID, Michael. *Introduction to Operations Research.* Malabar: Krieger Publishing Company, 2004 (reprint edition). ISBN 1-57524-198-6.

GOURVEST Henri, PATTON William and Peter NOTEBAERT. *LPSolve*, version 5.5.2.0. http://sourceforge.net/projects/lpsolve/.

JABLONSKÝ, Josef. *Operační výzkum: kvantitativní modely pro ekonomické rozhodování.* 1. vyd. Praha: Professional Publishing, 2002, 323 s. ISBN 80-864-1942-8.

MELNICK Michael. *Linear Program Solver (LiPS)*, version 1.11.0 http://sourceforge.net/projects/lipside/.

WANER Stefan. *Excel Pivot Utility.* http://www.zweigmedia.com/RealWorld/Summary4.html.

Mgr. Jiří Neubauer, Ph.D.
Department of Econometrics, Faculty of Economics
and Management, University of Defence Brno
Kounicova 65, Brno, 612 00, Czech Republic
E-mail: Jiri.Neubauer@unob.cz
Telefon: + 420 973 442 029