ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LXI 259 Number 7, 2013

http://dx.doi.org/10.11118/actaun201361072311

THE DEVELOPMENT OF E-LEARNING APPLICATIONS SOLVING PROBLEMS FROM GRAPH THEORY

Pavel Kolman, Petr Zach, Josef Holoubek

Received: April 11, 2013

Abstract

KOLMAN PAVEL, ZACH PETR, HOLOUBEK JOSEF: *The development of e-learning applications solving problems from graph theory.* Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 2013, LXI, No. 7, pp. 2311–2316

Authors are in long term developing e-learning supports for some parts of Operation Research course. The original purpose was to prepare e-learning supports for students of FBE Mendelu, mainly for part-time form students, who have subscribed Economic mathematical methods course (EMM). Considering decreasing number of lessons on part-time form of study (16 hours in a semester, in comparison to 56 hours in full-time form of study), was for part-time form students even more difficult to fulfill exam requirements. As a help for students there was stage by stage prepared several of programs, they allow self-contained practicing of some linear programming methods. Programs did allow to users step-by-step verify their solution, i.e. whether their calculation are in accordance with algorithm described in lectures. Advantage for the students consists in fact, that each mistake (numerical or algorithmic) they were able to uncover, what contributes to increase of self-study effectiveness and from that resulting higher study motivation. Resulting from existing experience, authors decided to request for a new FRVŠ grant for academic year 2012, focused on e-learning support of selected graph theory problems. Within this project there was developed a tool allowing to make and according to the user needs interactively modify created graphs. On this graph it is possible individually, step by step (in compliance with on lectures presented algorithm) to practice solving of selected graph theory and network analysis problems (e.g. minimal spanning tree, shortest path in a graph, testing for cycles in a graph, critical path method etc.). Project is realized as modular and was realized in Delphi developing tool. Described algorithms are saved in dynamic linked libraries. There for it is very easy to add here new (newly programmed) algorithms. Project results (i.e. project experience obtained from e-learning supports) will be available for all FBE Mendelu members interested in this problematic.

graph theory, optimization, operations research, Delphi

Growing content of EMM course and decreasing time allowance, mainly in part-time form of study, led to request to offer students supporting study materials. This materials should students allow and ease their individual home preparation and raise their study motivation. Increased usage of information technologies offers new ways in accessing study matherials to students. Several e-learning courses and educational software (e.g. Pokorný, Stávková, 2007) are implemented to improve study efficiency. Foltýnek (2006) and Tomanová (2010) deal with methodology and efficiency of e-learning matherials.

Authors of this paper have in last few years prepared and released for students of FBE lot of programs, which allowed students to acquire linear programming methods (e.g. simplex and dual simplex method, transportation problems, etc.). Programs allow contrary to the algorithms, verify step by step correctness of algorithms procedure. Response of students to these pedagogical programs is without exception only positive. They appreciate particularly option to verify, how they did acquire this methods and alternatively find out, where they make mistakes. These options verifiably increase their study interest and evaluation of this course practical application. As it results from obtained experience and in agreement with announced student's requirements they were prepared animated programs for practicing of methods and solution of selected graph theory problems. The response on their use in education process was relatively high, mainly because these types of problems are discussed on the end of course and there is relatively short time for their practicing in courses. The only disadvantage of those animated programs is fact, that each of described methods was demonstrated only on one sample example. Requirements of students for practicing of these methods on "any" example led authors for requesting of this FRVŠ grant.

METHODS AND RESOURCES

Most of authors, e.g. Dudorkin (1997), Fábry (2003), Gros (2003), Jablonský (2002), Plevný and Žižka (2007), Walter and Vejmola (1978), Stevenson (1992), Stevenson and Ozgur (2007), use for explanation of graph theory problems always the same system. They verbally describe individual steps of the algorithm on an example. If the student does not understand the algorithm, it is for him difficult to practice given problem, and has no opportunity to identify, where is he making the mistake. Some of the authors, like Filípek (2008), Filípek and Černý (2008), Garcia *et al.* (2007), Lee and Lee (2006), improved this process and prepared some flash animations. The main disadvantage of this improved process is that it cannot be applied to any graph, but only to the described one. This disadvantage was the main reason for creating a Graphs.exe application that eliminates these shortcomings.

DISCUSSION AND RESULTS

Application description

Graphs.exe application is created as a self-installing package that is available for download at http://akela.mendelu.cz/~holoubek. After downloading and installing the program in the Start menu, folder is created with reference to the start of the program and its eventual removal. Since this is a program intended for Czech students, is created in Czech language. From the same reason, the pictures are in the Czech language, but it is always available English translation.

The authors created an application program Graphs.exe in accordance with the processes they are used in the teaching of selected graph theory algorithms in lectures and practices. This fact allows students to practice the use of these algorithms, just as described in the lessons, respectively as they were presented in the teaching. Therefore, students can concentrate on understanding the algorithm without the risk they would be misunderstood.

Application is created as a MDI (Multiple Document Interface). This means that the program can have multiple open windows with calculations simultaneously. The main advantage is that if the application is open with an unfinished graph and we need to open the new one, it is not necessary to close the original one (which would be necessary if the application has not been so programmed), but may be opened multiple graphs simultaneously. Between individual graphs can then be easily switched and calculations can be performed simultaneously.

Any calculation on any graph can be stepped (forward and backward), which is a requirement for programs designed for teaching. Students have the opportunity during their own calculation to check whether their steps are in accordance with the algorithm used to solve a given task type. If the students do numerical or logical error during the calculation, the program is enable immediately detect this error to them. (The students which already know the actual algorithm and are interested only verify the correctness of their solution; the program offers the option to skip directly to the final solution).

The program is designed so that each graph it could simply be exported to TeX. When developing algorithms were programmed to create an identical graph in a PSTricks format, which is a TeX package intended to create images. The advantage is that it is the source code of the image (graph), so it is no problem to edit, respectively to add there additional information. This graph can then be converted to PDF, in vector format. The advantage is a small file size and the ability to resize the created image without losing quality. It should be emphasized that the PDF conversion is not realized in the described program, but it is necessary to have installed on the computer TeX. This fact, in view of the fact that TeX is protected by the GNU-GPL license, should not be a problem.

Saving graph is realized through its own format. This can be in addition to the main application edited in any text editor. Description of the graph is divided into several parts. Each part begins with the text in curly brackets, which states what is in this section just described. This text is followed by the parameters of each part.

In the first section there are described the dimensions of the canvas, the units are given in pixels. The second section describes graph type and consists of three binary variables. The first and second variable indicates whether a graph is directed or undirected and rated or unrated. The third variable indicates whether it is a "common" or CPM graph type. The third part consists of a nodes names list. The fourth part shows list of edges. These are determined by the number of nodes they connect. On each row there is a list of edges leading from that node to other nodes. For each node is given edge length in brackets. In case of unrated

```
{PLATNO}(osax; osay)
(800;600)
{GRAF}orientace ohodnocenost CPM
1 0 1
{UZLY}pocet (nazev1; nazev2;.. nazevpocet)
(1;2;3;4;5;6;7;8)
{HRANY}pocatecni_uzel:          koncovy_uzel(delka_hrany)
[koncovy_uzel(delka_hrany)])
3: 8(1)
4: 8(1)
...
{GRAFIKA}--zcela  volitelna  cast--pocatek  souradnic  v
levem hornim rohu, uzel: (souradnice_x,souradnice_y)
1: (118;306;15)
2: (212;175;15)
...
```

1: *Source code of the given graph*

graph, the edge length is determined by the value one. In this moment we know all the information, which is necessary for creation of a graph.

The fifth part is optional and allows to specify the layout of nodes and edges of the canvas. In case that this part is not available, the nodes on the canvas are placed randomly. In the opposite case, for each node are given its x-and y-coordinates on the canvas, and the node size. Sample format for storing graph is shown in Fig. 1. Because the program is written in the Czech language, the source code is not translated into English.

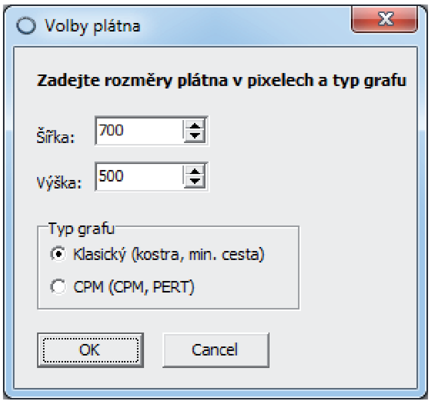
Any graph can also be downloaded from the incidence matrix. Then the nodes are numbered automatically according to the column index and the length of the edges according to the values in the appropriate matrix field. Delimiter when loading a file can be whitespace (i.e. space or tab). When the graph is loading from the matrix it is not possible to specify the position of nodes, so the nodes are distributed randomly on the canvas.

Graphs.exe program is designed to operate always in one of two modes. In the first mode, you can create, edit and save any graph. In the second mode we select for an arbitrary graph the algorithm which we want to use, and we can step through our calculation. In case we want this graph additionally modify, we go back to the first mode.

The use of program for solving selected graph problems.

After starting the program, a window appears with setting the option of canvas size and type of the graph. The graph can be either a “typical”, respectively CPM type. These options are fixed from the beginning and during the work with a given graph can no longer be modified.

After starting the program is in the mode of graph creation. This mode is visible in Fig. 3 in the box labeled “Nástroje” (Tools). The left part of the window, intended for creating and editing a graph is active. Right part of the window intended for stepping of selected algorithms is inactive. In the left part of the window there are 4 buttons with these functions:



Zadejte rozměry plátna v pixelech a typ grafu

Šířka: 700

Výška: 500

Typ grafu

● Klasický (kostra, min. cesta)

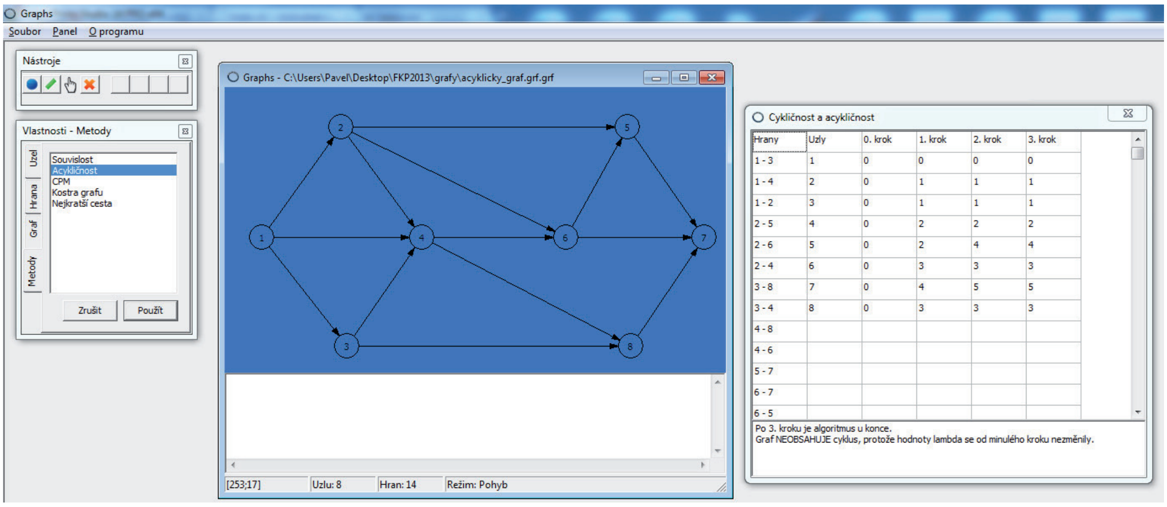
○ CPM (CPM, PERT)

2: *Selecting a graph type and dimensions of the canvas*

- Blue dot (first button from the left) – to add a node. Node can be added anywhere on the canvas except where there would be overlap with other existing node.
- Green line (second button from the left) – to add an edge. It is necessary to click the mouse button on the node where the edge begins, hold down the left mouse button and move over the node where the edge ends. If it is a directed graph, an arrow appears at the node, in which the edge ends.
- Hand (third button from the left) – to change the position of nodes. Changing the position of the node evoke changes the location of edges, which come from the node, respectively entering the node.
- Red cross (the fourth button from the left) – is intended for erasing. By clicking on an edge it will be deleted. Clicking on a node will delete both the node and all edges leading to / from the erased node.

After drawing the graph it is necessary to switch the focus to the “Vlastnosti” (Properties) window. Here are 4 available bookmarks (see Fig. 3):

- Bookmark “Uzel” (Node) changes node position (it is possible to exactly specify its coordinates on the canvas), its description, and size.

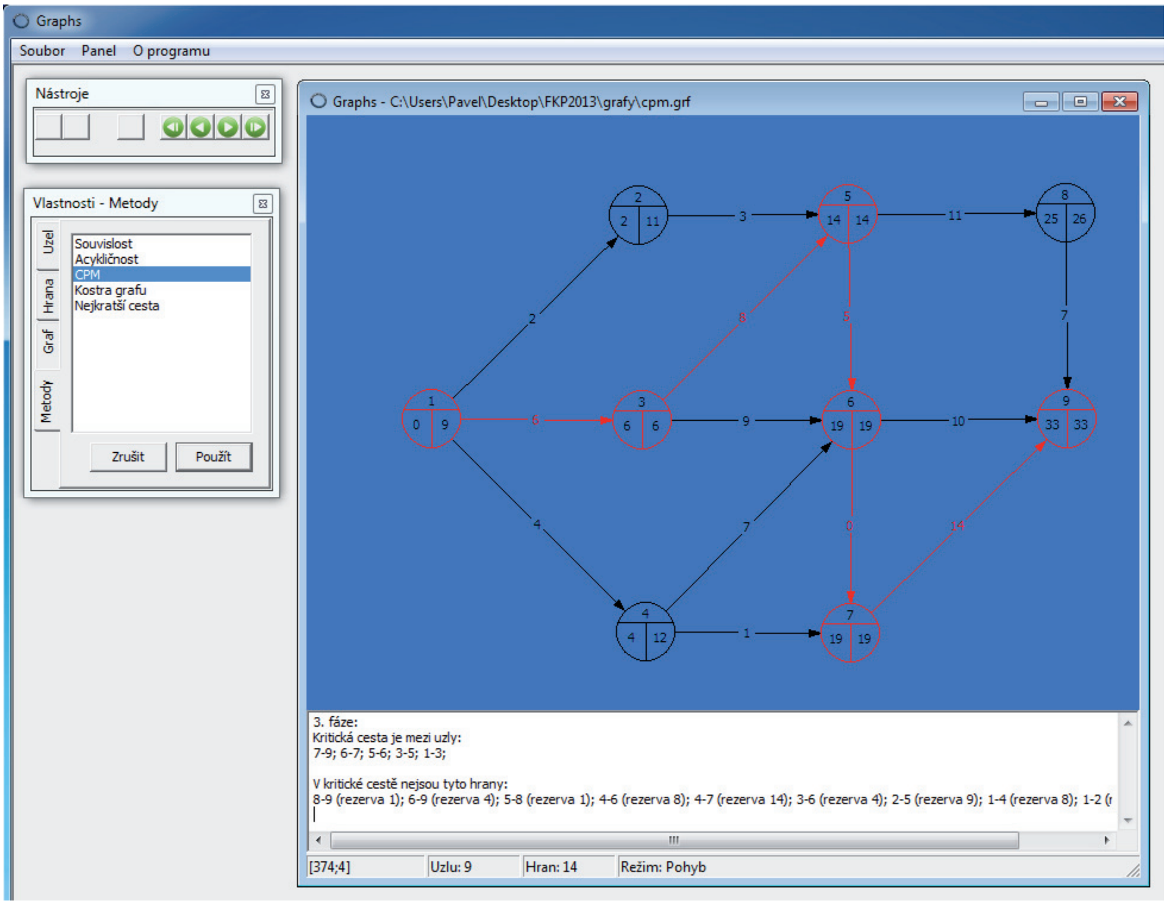


Hrany	Uzly	0. krok	1. krok	2. krok	3. krok
1 - 3	1	0	0	0	0
1 - 4	2	0	1	1	1
1 - 2	3	0	1	1	1
2 - 5	4	0	2	2	2
2 - 6	5	0	2	4	4
2 - 4	6	0	3	3	3
3 - 8	7	0	4	5	5
3 - 4	8	0	3	3	3
4 - 8					
4 - 6					
5 - 7					
6 - 7					
6 - 5					

Po 3. kroku je algoritmus u konce.
Graf NEOBSAHUJE cyklus, protože hodnoty lambda se od minulého kroku nezměnily.

[253;17] Uzlu: 8 Hran: 14 Režim: Pohyb

3: *Testing of cycles in the graph. "Nástroje" (Tools) panel is set to create the graph*



3. fáze:
Kritická cesta je mezi uzly:
7-9; 6-7; 5-6; 3-5; 1-3;

V kritické cestě nejsou tyto hrany:
8-9 (rezerva 1); 6-9 (rezerva 4); 5-8 (rezerva 1); 4-6 (rezerva 8); 4-7 (rezerva 14); 3-6 (rezerva 4); 2-5 (rezerva 9); 1-4 (rezerva 8); 1-2 (r

[374;4] Uzlu: 9 Hran: 14 Režim: Pohyb

4: *Finding of Critical Path in a graph. "Nástroje" (Tools) panel is set to step algorithm*

- The "Hrana" (Edge) bookmark allows change edge value and the nodes where the edge begins, respectively where the edge ends.
- In the "Graf" (Graph) bookmark it is possible to set some properties of the graph – specifically if the graph is rated and directed.
- The "Metody" (Methods) bookmark lists the methods applicable to the given type of graph. These are the methods that are available in a dynamically Linked Libraries (DLLs).

In order to initiate stepping algorithm, it is necessary in the "Metody" (Methods) bookmark choose any of them, and then confirm the selected one by pressing the relevant button. Then in the

I:

	Number of students	Students who have completed their evaluation	Average preparing for the exam (hours)	The average preparation during the semester (hours)	Average grade (counts all attempts)
after the adding of program (school year 2012/13)	291	186	44.35565	40.00081	3.23
before adding of the program (school years 2010/11 and 2011/12)	821	470	44.58404	41.49638	3.22
P-value of the test	x	x	0.842	0.112	0.867

“Nástroje” (Tools) window will be activated buttons, allowing stepping on the graph according to the selected algorithm. At this moment, the program goes from graph creation to the stepping mode. Buttons for graph creating become inactive. Stepping buttons corresponds to already known rules, in order from left to right correspond to the buttons: jump to the beginning of the calculation, step back, step forward, jump to the end of the calculation (see Fig. 4).

The program was created as easily extensible using dynamically Linked libraries. They must be programmed, but the main program does not need to be modified. Available algorithms are currently stored in them, i.e. in files with the *.DLL extensions that are located in a Libraries subfolder. Adding a new algorithm simply consists of copying it to the appropriate folder. This allows anyone who is able to program, add algorithms he needs to the application, without having to have the source code of the main program.

Currently these are available algorithms: testing connection graph, testing cycles in the graph, determination of the minimum spanning tree, shortest paths in the graph and determination of the critical path in the graph using the CPM method.

Changing of the efficiency of studies

After adding the program to the e-learning it was subsequently tested whether and how much the preparation time during the semester, time of preparation for the exam and average mark were reduced. The testing was performed using data from evaluations of students in each semester. In calculating of the average mark, all examination attempts of all students were enrolled. The data from categories “time of preparation for the exam” and “preparation during the semester” were used only by students who completed the evaluation. Here it should be mentioned that this is the classified data and the students estimate their preparation time, so the results may be distorted. Summarized data are available in the table above.

It is interesting how closely the data before and after adding of the program varies, in all monitored categories. For accurate quantification of changes in the data before and after adding the program, the two mean values two-sided t-test was used. For testing, the Gretl program was used. In all three cases, the zero hypothesis was not rejected. It can be said that the addition of programs for working with graphs did not reduce the preparation time or did not improve test results. However, the opinion of students on this program in evaluations is generally positive.

SUMMARY

The paper introduces readers with further results, which the authors come when preparing support materials designed for self-study of selected graph theory problems. The need to create such materials is mainly due to a substantial decrease in the number of teaching hours for part-time form of study, the increasing content of EMM course and changes in the level of skills and knowledge of students. The aim of the authors was under given conditions to offer students the opportunity to acquire knowledge of methods applicable to the solution of selected graph theory problems. To achieve this, a computer program was prepared. This program allows to perform the necessary operations with graphs (i.e. create graph and edit various graph parameters, to save and load graphs from/to files, etc.) and then using these graphs to solve selected types of problems. The program was created so that the process of problem solving would exactly correspond with the procedure used in the practices. Anyone interested in the problem has the opportunity to verify step by step if the solution process is correct, find out where he eventually made a mistake, and if it was a numeric or logical error. The option of quickly and easily identify the error in the solution helps accelerate the study and also to increase student motivation. Now, the program offers to check knowledge of selected methods, i.e. to test the connection and cyclicality of graph, to determine the minimum spanning tree, shortest path and critical path in the graph. However, it is designed to be easily extended by methods used to solve other types of graph theory problems. The program was created as a support, which assumes a certain level of knowledge the reader gains either participation in education or self-study. It is primarily intended for pedagogical purposes. As a bonus, we can consider the fact that the program

allows to export created graphs into TeX. These graphs then may be used by teachers for example in the preparation of textbooks or materials for testing.

Acknowledgement
The program was created with the support of a FRVŠ 2646/2012 grant.

REFERENCES

DUDORKIN, J., 1997: *Operační výzkum*, Praha: ČVUT, 296 s. ISBN 80-01-01571-8.

FÁBRY, J., 2003: *Management science*, Prague: Oeconomica, 129 s. ISBN 80-245-0586-X.

FILÍPEK, J., 2008: 'Animated didactic tests for specialised courses', *Proceedings of the conference Pedagogický software 2008*, České Budějovice: Scientific Pedagogical Publishing, pp. 211–213. ISBN 80-85645-59-9.

FILÍPEK, J. and ČERNÝ, M., 2008: 'Testing the mechanical properties of materials', *Proceedings of the conference Pedagogický software 2008*, České Budějovice Scientific Pedagogical Publishing, pp. 75–78. ISBN 80-85645-59-9.

FOLTÝNEK, T., 2006: *Metodika využití eLearningových technologií ve vzdělávacím procesu*. Dissertation thesis. MENDELU Brno.

FOLTÝNEK, T., HALUZA, P., MOTYČKA, A., 2011: Předmět Teorie grafů: Zkušenosti s výukou a tvorbou e-learningové opory. *Informatika XXIV/2011.* Brno: ES Mendelovy univerzity v Brně, 1, pp. 25–26, ISBN 978-80-7375-520-1.

GARCIA, P. R. et al., 2007: Interactive multimedia animation with macromedia flash in descriptive geometry teaching, *Computer & Education,* 49, 3: 615–639. ISSN-0360-1315.

GROS, I., 2003: *Kvantitativní metody v manažerském rozhodování*, Praha: Grada, 432 s. Expert. ISBN 80-247-0421-8.

JABLONSKÝ, J., 2002: *Operační výzkum*, Praha: Professional Publishing, 323 s. ISBN 978-80-86946-44-3.

LEE, K-J. and LEE, J., 2006: Programming physics softwares in Flash, *Computer Physics Communications, Proceedings of the Conference on Computational Physics 2006 – CPP 2006*, 177, 1–2, pp. 195–198. Available from: http://www.ccp2006.postech.edu/.

PLEVNÝ, M., and ŽIŽKA, M., 2007: *Modelování a optimalizace v manažerském rozhodování*, Plzeň: ZČU, 296 s. ISBN 978-80-7043-435-2.

POKORNÝ, M., STÁVKOVÁ, J., 2007: Výukový software podporující finanční analýzu – FinAnalyzátor v0.1. *Obchod a spotřebitel '06.* Brno: Konvoj, pp. 111–114. ISBN 978-80-7302-124-5.

TOMANOVÁ, V., 2010: Efektivita využití distančních opor při výuce. Diploma thesis. Brno.

WALTER, J. and VEJMOLA, S., 1978: *Síťová analýza*, Praha: VŠE. 280 s.

STEVENSON, W. J., 1992: *Introduction to Management science*, New York: Irwin, 909 s. ISBN 0-256-08809-8.

STEVENSON, W. J. and OZGUR, C., 2007: *Introduction to Management science with Spreadsheet*, New York: Mc Graw-Hill, 812 s. ISBN 978-0-07-325290-2.

Address
Ing. Pavel Kolman, Ph.D, Department of Statistics and Operation Analysis, Ing. Petr Zach, Department of Informatics, doc. Ing. Josef Holoubek, CSc. Department of Statistics and Operation Analysis, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic, e-mail: xkolman@mendelu.cz, petr.zach@mendelu.cz, holoubek@mendelu.cz