Teaching Mathematical Reasoning: From Textbooks to Software

Ljiljana Brankovic, Alexander Muir and Helen Giggins
School of Electrical Engineering and Computer Science, The University of Newcastle
Corresponding Author's Email: Ljiljana Brankovic@newcastle.edu.au

Structured Abstract

BACKGROUND

This project addresses critical challenges associated with learning and teaching mathematical reasoning. Mathematics is not only an enabling science that provides foundations and enhances performance in many disciplines, but also "a critical skill" for "every Australian citizen", and especially for computer science and engineering students. We explore how the effectiveness of learning and teaching mathematics can be enhanced by supplementing traditional teaching material, such as textbooks and exercise collections, with a software package. We use "GrInvIn", open source software developed at Ghent University by Adriaan Peeters, Gunnar Brinkmann, Kris Coolsaet and others, which offers a graphical, interactive, and multi-level approach to teaching and learning. GrInvIn or similar software has been used for teaching at universities in USA, Belgium, Germany, Slovenia, Croatia and Serbia, and in high schools in Belgium (Flanders).

DESIGN/METHOD

In this project we explore how GrInvIn can be used to enhance mathematical reasoning in students. In a typical teaching scenario, a student chooses (or is given) a graph invariant, and GrInvIn generates a conjecture about that invariant that the student has to prove or disprove. While this works well at postgraduate and senior undergraduate level, at junior undergraduate and high school level a randomly generated conjecture may be too difficult for the student, which may result in a loss of motivation. Consequently, the authors of GrInvIn suggested that in introductory courses it may be necessary to use pre-generated teaching scenarios, to control the level of challenge, so as to not discourage the students. We have developed a basic GrInvIn Instructor's Manual that includes teaching scenarios, graph lists, conjectures and minimum counterexamples.

RESULTS

We ran a GrInvIn instruction session for student volunteers studying towards BCompSc and BEng and the University of Newcastle, and collected their feedback which was then used to evaluate the effectiveness of teaching scenarios and effectiveness of GrInvIn for teaching junior levels. After instruction sessions, students were asked to fill in a questionnaire.
The results of the survey and the focus group discussion indicate that GrInvIn is an excellent tool for senior students and that the teaching scenarios are an effective addition to GrInvIn to accommodate junior students. The discussions emphasised the importance of the introductory lecture and identified a need for support throughout the GrInvIn session.

CONCLUSIONS

The main strength of GrInvIn is engaging and motivating students to learn mathematics, providing that sufficient support is given to novice learners. GrInvIn, together with teaching scenarios for introductory levels and possibly some additional written material that students can refer to during the GrInvIn sessions, has a potential to foster positive behaviours pertinent to competitive learning style and intrinsic motivation, at least in senior and some junior students.

KEYWORDS

GrInvIn, teaching software, intrinsic learning

Introduction

This paper addresses critical challenges associated with learning and teaching mathematical reasoning. Mathematics is not only an enabling science that provides foundations and enhances performance in many disciplines, but also “a critical skill” for “every Australian citizen” (Rubinstein, 2009). We aim to enhance the effectiveness of learning and teaching mathematics by supplementing traditional teaching material, such as textbooks and exercise collections, with a software package. We use “GrInvIn”, Ghent University software, which offers a graphical, interactive, and multi-level approach to teaching and learning and was developed specifically to enhance mathematical reasoning in students. GrInvIn can be used at different levels to support a variety of student cohorts: first year undergraduate students, undergraduate education students, high school students, as well as honours, Master and RHD students who specialise in theoretical computer science or a related discipline.

In this paper we explore how the benefits of GrInvIn at the junior undergraduate level can be enhanced by using pre-generated teaching scenarios in order to control the level of challenge to which the students are exposed. To this end we developed a basic Instructor’s Manual that includes teaching scenarios together with conjectures and minimum counterexamples. To evaluate the effectiveness of the teaching scenarios we ran a GrInvIn instruction session followed by a a questionnaire and a focus group discussion.

The organisation of the paper is as follows. In the next section we present in some detail the GrInvIn software and we discuss the need for the teaching scenarios. In the following section we give a brief overview of other relebant software in order to put GrInvIn into perspective. In the subsequent sections we present the teaching method and scenarios and evaluate them through a focus group discussion and a questionnaire. In the conclusion we discuss the results and the limitations of the study, and we offer some future directions.

Learning Styles

A learning style can be defined as a way we concentrate on, process, internalize, and remember new information (Dun and Dun, 1978; Ma and Ma, 2014). Dun and Dun (1978) identified 18 distinct learning style aspects, including environmental, emotional, sociological and physical elements. The most relevant aspects for this study are motivation as an emotional element and competitive vs cooperative styles as sociological elements.

Academic motivation has been extensively studied (Bandura, 2010). There are two basic types of academic motivation: intrinsic, which is a desire to learn for learning sake, as the process of learning is enjoyable, and extrinsic, where a student learns in order to obtain rewards or avoid punishment (Middleton and Spanias, 1999). This project focuses on intrinsic motivation. When students are motivated intrinsically, they choose more difficult tasks, spend increased time on them, persist in the face of failure, and exhibit greater creativity (Middleton and Spanias, 1999; Lepper, 1998).

Ma and Ma (2014) studied the effects of competitive vs cooperative learning styles on mathematical performance across USA and 3 top performing East Asian countries (Japan, Korea and Hong Kong) and found that competitive learning increases mathematics performance across all countries. In the section “Teaching Method and Teaching Scenarios” we argue that GrInvIn software encourages the intrinsic motivation as it introduces a competition into the learning and students take the ownership of the subject allocated to them.

GrInvIn

The effectiveness of the interactive learning approach is well recognised, as it engages students, gives them time to think, and provides opportunities to ask questions and learn from mistakes (Rodger, 1995). GrInvIn (**Gr**aph **Inv**ariant **In**vestigator) uses graph theory to promote not only interactive learning but also problem-based learning, where the knowledge acquisition process is vastly engaging, and students learn to think for themselves (Savery and Duffy, 2001). GrInvIn is an open source software developed at Ghent University, Belgium, by Adriaan Peeters, Gunnar Brinkmann, Kris Coolsaet and others. Academics in Belgium, Germany, USA, Slovenia, Croatia and Serbia, have used GrInvIn or similar software (Graffiti, Graffiti.pc) to complement traditional teaching methods and create blended learning opportunities that motivate students (Brinkmann et al., 2008). Additionally, GrInvIn has been used at the high school level in Belgium (Flanders). It is worth noting that the Flemish part of Belgium has a very good reputation in mathematical education and has been ranked first in the 2003 OECD Programme for International Student Assessment (PISA) for mathematics, which included 41 countries (OECD). In fact, the use of GrInvIn in Belgium (Flanders) was facilitated by the class structure in Flanders, which includes "free hours" where the curriculum is left to the discretion of the teachers. Based on this previous teaching experience and our own experience with GrInvIn, we trust that this vibrant and interactive software has a potential to modernise Australian mathematics education. Indeed, it has been noted that "undergraduate computing education today often looks much as it did several decades ago" (National Science Foundation), which is even more true for mathematics where some lecturers still use blackboards. Then it comes as a no surprise that students who belong to a "Nintendo generation" often vote with their feet (Lister, 2008).

GrInvIn is potentially suitable for many and varied student cohorts including:

1) first year undergraduate students in programs that require mathematics as an enabling subject; these include science, engineering, economics, finance, business and health;
2) undergraduate education students and teachers who need professional development;
3) high school students, especially for the purpose of engaging them with mathematics and providing extra curriculum activities for year 11 and 12 students;
4) postgraduate students in the areas of mathematics and computer science; these students will use GrInvIn in a less structured and more research oriented fashion than high school and junior undergraduate students.

At the high school and junior undergraduate levels GrInvIn can be used to invoke interest in discrete mathematics and teach basic concepts and critical mathematical thinking in a problem based learning environment. Following the model already used at Ghent University, the students would be provided with a short introductory lecture outlining the main concepts in graph theory, and will be expected to learn the rest on a need-to-know basis. At the senior undergraduate and postgraduate levels, the software can be used to deepen and sharpen the students' problem solving skills and facilitate understanding and checking of more challenging mathematical theorems.

About GrInvIn and why it is a good choice

The current version of GrInvIn is based on graph theory, which is a particularly good choice for the following reasons:

(1) unlike some other branches of mathematics, graph theory is a very approachable discipline as its problems can be formulated in very simple terms so that even young children can understand them; they do not require sophisticated terminology and notation;
(2) graph theory is relevant to many areas of science, engineering and humanities, and also to problems encountered in everyday life situations, such as social networks, road maps, to mention but a few. Not surprisingly, graph theory has often been used for professional development of high school teachers in the US and other counties (e.g., by DIMASC).

GrInvIn has been designed having both research and teaching in mind and is inspired by Graffiti and Graffiti.pc developed by S. Fajtlovicz and E. DeLaVina, respectively. However, unlike GrInvIn, Graffiti.pc was never intended to be distributed.

GrInvIn provides a user friendly interface (see Figure 1) that is easy-to-use and very intuitive, thus the students can easily guess how to use the menu and most of the time they will not need to read the documentation, which will shorten the time required to become familiar with the software (Peeters et al., 2009). It operates based on drag and drop principles.

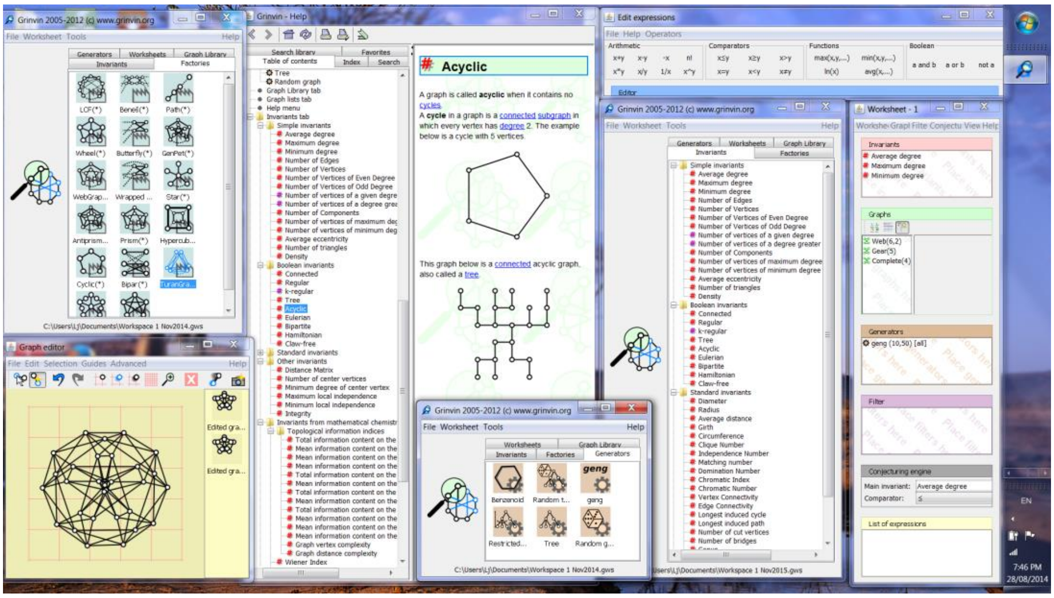

Figure 1. GrInvIn interface

Overview of Other Graph Theory Software Systems

Apart from GrInvIn, there exist a number of other graph theory software packages. Some of them were designed for both teaching and research, such as Combinatorica (Pemmaraju and Skiena, 2003), Graph Magics (Ciubatii, 2004), LINK (Berry, 2009), AutoGraphiX (Caporossi and Hansen, 2000), as well as Four Colour Theorem and Alpha Labelling software still under development (Reynolds and Brankovic, 2013). Other software packages were designed predominantly for research, including Graffiti and Graffiti.pc ((Fajtlowicz et al., 2005), GRAPH and newGRAPH (Stevanovic et al., 2003), GraPHedron (Mélot, 2008), CABRI-Graph (Carbonneaux et al., 1996), as well as some more specialised software such as plantri and fullgen for generating certain classes of planar graphs (Brinkmann and McKay, 2011) and nauty for finding isomorphisms and automorphisms of graphs (McKay and Piperno, 2013). Prior to GrInvIn, the conjecture generator was incorporated in some other packages, including GRAPH, newGRAPH, Graffiti, Graffiti.pc, GraPHedron, LINK and AutoGraphiX. There are hundreds of conjectures generated by Graffiti that mathematicians worked on (Fajtlowicz et al., 2005) and there is a large number of papers whose authors were assisted by GRAPH (Cvetkovic and Simic, 2005). Graffiti.pc (DeLaVina, 2005) is the software that inspired GrInvIn, and it is closest to it. We chose GrInvIn over Graffiti for its portability (unlike Graffiti.pc, it is written in Java) and its availability (unlike Graffiti.pc, it is an open source software).

Teaching Method and Teaching Scenarios

A highly attractive feature of GrInvIn is its learning method, originally used for research purposes in Graffiti and Graffiti.pc (Fajtlowicz et al., 2005). In this method, students use software to generate a conjecture about their chosen graph invariant (a parameter describing a certain mathematical structure), and then independently prove or disprove this conjecture. Throughout this process, students learn mathematical facts and acquire reasoning skills. They typically find this learning method very motivating and often identify with their chosen invariant and strive to learn as much as possible about it (Peeters et al., 2009). This is a fine example of a learner becoming competitive, “developing ownership for the overall problem or task“ (Savery, 2001) and moving from extrinsic motivation where they learn for marks to intrinsic motivation where learning becomes fun. Peeters et al. (2009) also report that many students who used this learning method in Graffiti and Graffiti.pc in beginners graph theory courses were motivated to continue their education in graph theory and chose to attend regular graph courses afterwards.

GrInvIn’s teaching method has elements of problem based learning (Du et al., 2009; Brankovic et al., 2013), as it relies on students to acquire knowledge as they need it in order to complete the task in hand. The learning process proceeds as follows (Peeters et al., 2009).

1. A student chooses (or is given) a graph invariant for the left hand side of the inequality that GrInvIn generates as a conjecture; this is “their” invariant.

2. The student chooses (is given) a pool of invariants from which the invariants on the right hand side will be selected; he/she is not necessarily familiar with all the pool invariants.

3. Finally, the student selects a list of graphs on which the GrInvIn conjecture is based.

4. GrInvIn generates a conjecture in such a way that it is always true for all the graphs in the list. Initially this list typically consists of a single graph, or a very small number of basic graphs. The rights hand side of the “conjecture inequality” contain constants and some of the invariants from the pool, which students need to familiarise themselves with in order to attack the conjecture.

5. The student either proves or disproves the conjecture。
    a. To disprove it, the student constructs a minimum counterexample and prove its minimality. Then the student adds the counterexample to the list of graphs, and GrInvIn generates a new, typically harder conjecture. Thus, as the list of graphs grows, so does the difficulty of the conjectures, as well as the student’s knowledge and mathematical reasoning ability.
    b. In the case that the conjecture is true, the student proves it; additionally, their task is to investigate how tight the bound is, that is, how big the difference between the left and the right hand side of the inequality is. In order to drive GrInvIn to generate a different conjecture, the students can add another graph to the list and, if that is not successful, delete one of the right hand side invariants.

At the introductory level, students may choose invariants in an unfortunate way so that the conjecture posed by GrInvIn may be too difficult to prove or disprove. For that reason the authors of GrInvIn suggest using teaching scenarios where all the invariants to be used on either left or right hand side of the inequality were preselected by the teacher. Then each student is very likely to have the same experience with GrInvIn, since all the counterexamples found by students are minimum. We have developed 8 teaching scenarios to support learning with GrInvIn at the introductory level and used them in the instruction session with student participants.

Evaluation

We conducted an instruction session, focus group and a questionnaire with five student volunteers from our Faculty. Two of them were computer science students and have had done more than three maths courses, including graph theory. The remaining three students were studying engineering, have had done three or fewer maths courses, and none of them have had done graph theory.

We started by presenting a brief whiteboard introduction to explain the main concepts and terminology for the benefit of those participants who did not have any previous exposure to graph theory. We then ran a GrInvIn session, after which we asked the participants answer the questionnaire. Subsequently, we conducted a focus group with the participants where the teaching with GrInvIn software was discussed.

For each question in the questionnaire, the participants had to choose one of the following answers: 'strongly agree, 'agree', 'neutral', 'disagree' and 'strongly disagree'. Their responses were coded with integers 5 through 1, where 5 corresponded to "strongly agree".

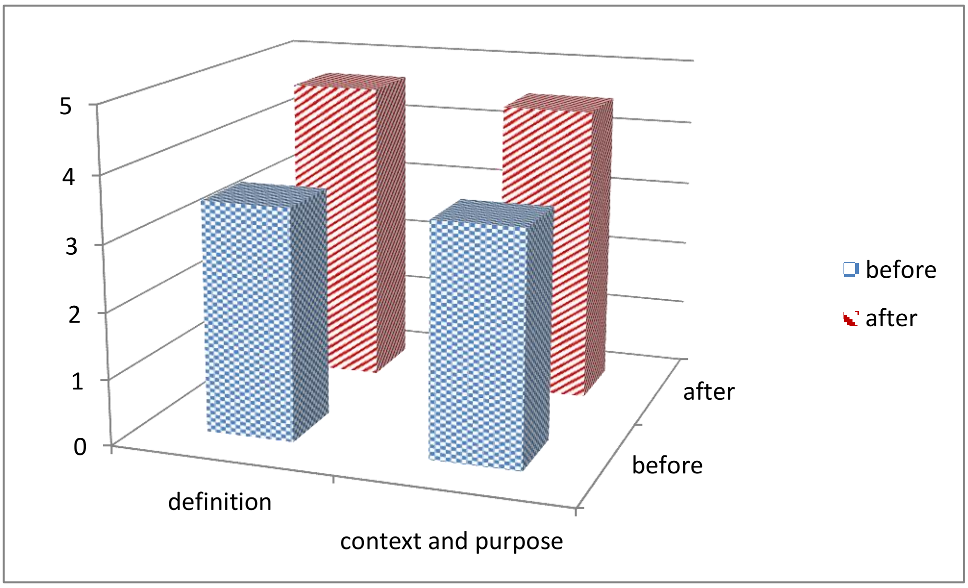

Figure 2: Understanding the concept definition, and the concept context and purpose, averaged over all concepts and all respondents.

The questionnaires identified four concepts from graph theory: graph invariant, handshake theorem, number of edges (bounds on) and circumference (bounds on). These four concepts were initially explained by a lecturer using whiteboard, and were subsequently used in the session with GrInvIn. In the questionnaire the students had to rate their understanding of the 1) concept definition, and 2) context and purpose; both of these were to be rated 1) before the session with GrInvIn and 2) after it. The results are presented in Figure 2, while the responses to the other questions are summarised in Table 1.

Table 1: Questionnaire Responses – averaged over all respondents on the scale 1 to 5

Question	Response
My overall understanding of the concept has been enhanced by the GrInvIn software.	3.4
The presented GrInvIn teaching scenarios are useful.	4
The GrInvIn software preforms a useful service.	4.2
Software diagrams and illustrations are very helpful.	4.2
GrInvIn software interface is intuitive.	3.4
The software has a sufficient level of interactivity.	3.2
This session with GrInvIn software inspired my interest in graph theory.	3.4

During the focus group discussion, it was apparent that the participants without previous experience with graph theory appreciated the opportunities the GrInvIn offers for experimentation ("I thought it was a good way to experiment", "I can see how it is set up, you first just play with things, see there is a mistake, and then you go bring another thing, like more things together and eventually you can come to a conclusion", "I see how it would be a good way to learn, pick it up").

In sharp contrast, the students who had previously studied graph theory seemed to favour the way they learned it over experimentation ("When I learned it first the key thing was to understand why", "I thought if you did not know a lot about graph theory, and maybe I am wrong because everyone else said differently, I did not find it very useful", "If you don't get it in the first instance you are not going to sit there and bash your head against the wall trying to get it to work in your favour.")

The three students who had not studied graph theory previously also emphasised the importance of a direct instruction session to grasp the main concepts and a need for sufficient support through the GrInvIn session ("I think it is an all right way to teach if you can if you have somebody who is pretty knowledgeable with subject explain it to you all along", "if you are not familiar with a subject I wouldn't expect you to take off with it if you had a little help to begin with, but if you have somebody to kind of explain the concepts to you whilst you use them you are going to grasp them pretty good").

Some of the discussion revolved around possibilities for students to accidentally deviate from a teaching scenario and how that can be prevented ("I think the teaching scenarios did work very well when you actually could follow them"). It was reiterated that the only opportunity for an error is to select wrong invariants and the conclusion was reached that the other invariants should be disabled to minimise the risks of deviating from the scenarios.

Regarding the usefulness of GrInvIn for learners at higher levels, all participants were very positive, including those who had prior experience with graph theory ("I actually think for the higher level it's particularly useful", "If you are a little bit more confident then the ability to be able to play around with it and see what conjectures it comes up with, I think that really good", "That's where I'd spent a lot of time, playing around with it").

A suggestion was given to avoid too much terminology (e.g. invariant), have more interactivity ("highlight a vertex and get it's degree").

When prompted to think about the difference between experimentation/exploration and direct instruction where students are presented with facts and explanations, the participants thought both are important ("I don't think it is either-or proposition. What I'd suggest is that you need some information to be able to have 'guided' exploration – you understand some of the concepts and then you try to infer your knowledge by extending it so it's a bit of mixing them both - you explore and when you come across something that looks unusual you can go back, you learn a bit more so it's sort of going from one to the other", "In my case I tend to learn more by hands-on approach to thing, I learn by experimenting if I have the basic gist of what I have to do, what I have to achieve")

The discussion was finally summarised as the following:

1) GrInvIn is very useful at the higher level where students already know elementary graph theory and can now experiment
2) At the lower level GrInvIn needs to be employed with a bit of care:
    a. There is a great need for an introductory lecture to present the basic concepts
    b. Complex terminology should initially be kept to the minimum
    c. In teaching scenarios only invariants that are used should be enabled

At the end of the focus group discussion 4 out of 5 participants reached a consensus that GrInvIn is very useful at both higher level and an introductory level, as long as care was taken to explain basic concepts, avoid complex terminology, make sure students can't deviate from the teaching scenarios and support them throughout the experience ("The good thing about software is it really gives people who want to do that free rains to go wild, as long as you remember that some kids need that hand help at least at the start otherwise you are going to lose them"). The remaining student felt that the way he learned was superior to experimentation offered by GrInvIn ("Just the way I learnt graph theory I just don't feel that this software would be great for the introductory level").

Discussion and Conclusion

Limitations of this study include a small number of participants in the focus group and questionnaire (five). Additionally, the focus group was conducted for GrInvIn together with two other types of educational software so there was only a limited time available for students to experience GrInvIn. A larger dedicated study is needed to better understand the suitability of GrInvIn for students at junior levels.

There was a consensus among the participants that GrInvIn is an excellent tool for senior students. Regarding suitability of GrInvIn at the junior level, there was a clear distinction in opinions between the two participants who studied graph theory, and the three participants who did not, where the letter group expressed appreciation for GrInvIn and the teaching scenarios for junior level from the very beginning of the discussion. In the former group, by the end of discussion one participants formed an opinion the GrInvIn is useful not only at the higher level, but also at an introductory level as long as the care was taken to introduce the students to the basic concepts and support them throughout the experience, while the other participant remained convinced that the direct instruction approach works better at the introductory level. There are at least three possible explanations for this difference in opinion. The first one is that the exploration and experimentation offered by GrInvIn works very well for some students but is not suitable for others. The second one is “the original and the best” syndrome, where we appreciate the most what we experience first. Finally, the third explanation is that experimentation is most valuable when we find ourselves in a totally new terrain, which explains why the students without previous experience with graph theory appreciated experimentation at the introductory level, while others originally only appreciated it at the higher level.

In conclusion, the main strength of GrInvIn is engaging and motivating students to learn mathematics, providing that sufficient support is given to novice learners. GrInvIn, together with teaching scenarios for introductory levels and possibly some additional written material that students can refer to during the GrInvIn sessions, has a potential to foster positive behaviours pertinent to competitive learning style and intrinsic motivation, at least in senior and some junior students.

References

Berry, J. (2009). Improving discrete mathematics and algorithms cirricula with LINK. ACM's 1997 SIGCSE/SIGCUE Conference on Integrating Technology into Computer Science Education , 2009.

Bandura, A. (2001). Social cognitive theory: An agentic perspective, Annual Review of Psychology, 52, 1-26.

Brankovic, L., Ye, H. and Reynolds, M. (2013). Reflection and Guided Problem Based Learning in Software Engineering, *Proceedings of the 2013 AAEE Conference*, Gold Coast, Queensland, Australia.

Brinkmann, G., Coolsaet, K., Peeters, A. and Van Cleemput, N. (2008). Teaching Reasoning with GrInvIn, talk delivered at the *4th International Conference on Computers in Scientific Discovery (CSD4)*.

Caporossi, G. and Hansen, P. (2000). Variable neighborhood search for extremal graphs: I. The AutoGraphiX system, *Discrete Math*. 212, 29-44.

Carbonneaux, Y., Laborde, J-M. and Madani, R. M. (1996). CABRI-Graph: A tool for research and teaching in graph theory. *Graph Drawing, Lecture Notes in Computer Science*, 1027/1996.

Cvetkovic, D. and Simic, S. (2005). Graph theoretical results obtained by the support of the expert system "Graph" - An extended survey. In *Graphs and Discovery, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 69.

Du, X., de Graaff, E. and Kolmos, A. (2009). *Research on PBL Practice in Engineering Education*, Sense Publishers, Rotterdam.

Dunn, R. and Dunn, K. (1978). Teaching Students through their Individual Learning Styles: A Practical *Approach*. Reston, Virginia: Reston Publishing Company, division of Prentice-Hall Inc., 15-17.

Dunn, R. and Dunn, K. (1979). Learning styles/teaching styles: Should they... can they... be matched, Educational Leadership, 238-244.

Fajtlowicz, S., Fowler, P.W., Hansen, P., Janowitz, M. F. and Roberts, F. S. Eds. (2005). Graphs and Discovery. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 69.

Ciubatii, D. (2004). *Graph Magics.* Retrieved August, 2014, from http://www.graph-magics.com/

DeLaVina, E. (2005). In: *Graphs and Discovery*, ed. by Fajtlowicz, S., Fowler, P.W., Hansen, P. Janowitz, M.F. and Roberts, F.S., American Mathematical Society, Rhode Island, 71–79.

Lepper, M.R. (1998). Motivational considerations in the study of instruction. *Cognition and Instruction*, 5, 289–30.

Lister, R. (2008). After the gold rush: toward sustainable scholarship in computing, In Proc. Tenth Australasian Computing Education Conference (ACE 2008), Wollongong, NSW, Australia. CRPIT, 78. Simon and Hamilton, M., Eds. ACS. 3-18.

Ma, V. J. and Ma, X. (2014). A comparative analysis of the relationship between learning styles and mathematics performance, *International Journal of STEM Education*, 1:3, 1-13.

McKay, B. D. and Piperno, A. (2013). Practical Graph Isomorphism, II, *J. Symbolic Comp.,* 60, 94-112.

Mélot, H. (2008). Facet defining inequalities among graph invariants: The system GraPHedron, *Discrete Applied Mathematics,* 156(10), 1875–1891.

Middleton, J.A. and Spanias, P.A. (1999). Motivation for Achievement in Mathematics: Findings, Generalizations, and Criticisms of the Research. *J. for Research in Math. Education*, 30(1), 65–88.

National Science Foundation. CISE Pathways to Revitalized Undergraduate Computing Education (CPATH), Retrieved May 2010 from http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=500025.

OECD Programme for International Student Assessment (PISA), Retrieved August 2014, from www.pisa.oecd.org.

Peeters, A., Coolsaet, K., Brinkmann, G. and Van Cleemput, N. and Fack, V. (2009). GrInvIn in a nutshell. *J Math Chem*, 45:471–477.

Pemmaraju, S. and Skiena, S. (2003). *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Cambridge University Press.

Brinkmann, G. and McKay, B. (2011). *plantri and fullgen*, Retrieved October 2, 2013, from http://cs.anu.edu.au/people/bdm/plantri/.

Reynolds, M and Brankovic, L. (2013). Hard Theorems Made Easier: Visualisation and Interactivity, *Proceedings of the 2013 AAEE Conference*, Gold Coast, Queensland, Australia.

Rodger, S. H. (1995). "An Interactive Lecture Approach to Teaching Computer Science", *ACM SIGCSE Bulletin*, 27(1), 278 – 282.

Rubinstein, H. (2009). *A National Strategy for Mathematical Sciences in Australia*, National Committee for the Mathematical Sciences and Australian Council of Heads of Mathematical Sciences.

Savery, J. R and Duffy, T.M. (2001). *Problem Based Learning: An instructional model and its constructivist framework*, CRLT Technical Report No. 16-01.

Stevanović, D., Brankov, V., Cvetković, D. and Simić, S. (2003). *NewGRAPH*, Retrieved August, 2014, from http://www.mi.sanu.ac.rs/newgraph/

Acknowledgements

The authors thank anonymous referees for they helpful comments.

Copyright statement