Adaptive Support versus Alternating Worked Examples and Tutored Problems: Which Leads to Better Learning?

Amir Shareghi Najar[1,*], Antonija Mitrovic[1], and Bruce M. McLaren[2]

[1] Intelligent Computer Tutoring Group, University of Canterbury, New Zealand
amir.shareghinajar@pg.canterbury.ac.nz,
tanja.mitrovic@canterbury.ac.nz
[2] Human-Computer Interaction Institute, Carnegie Mellon University, USA
bmclaren@cs.cmu.edu

Abstract. Learning from worked examples has been shown to be superior to unsupported problem solving when first learning in a new domain. Several studies have found that learning from examples results in faster learning in comparison to tutored problem solving in Intelligent Tutoring Systems. We present a study that compares a fixed sequence of alternating worked examples and tutored problem solving with a strategy that adaptively decides how much assistance the student needs. The adaptive strategy determines the type of task (a worked example, a faded example or a problem to be solved) based on how much assistance the student received in the previous problem. The results show that students in the adaptive condition learnt significantly more than their peers who were presented with a fixed sequence of worked examples and problems.

Keywords: Intelligent Tutoring System, adaptive worked examples, assistance, self-explanation.

1 Introduction

Learning from worked examples has been shown to be an effective learning strategy. Sweller and Cooper [1] suggested presenting worked examples to students in the initial stages of learning, followed by problem solving once students have acquired enough knowledge [2]. Examples are a suitable approach for novices, since examples reduce the cognitive load and increase initial learning. Sweller [3] explained the worked-example effect based on the Cognitive Load Theory. Novices often have incomplete knowledge which makes problem solving difficult due to the high cognitive load, but worked examples present step-by-step explanations of how problems are solved with associated knowledge.

Many studies have compared learning from examples to unsupported problem solving, and showed that learning from examples is more effective [4][5]. Intelligent Tutoring Systems (ITS) are different from unsupported problem solving as ITSs support problem solving by providing adaptive scaffolding in terms of feedback, guidance, problem selection and other types of help. Only recently several studies have compared learning from examples to learning with ITSs (e.g. [6][7][8]). However, little attention has been devoted so far to the difference between novices and

V. Dimitrova et al. (Eds.): UMAP 2014, LNCS 8538, pp. 171–182, 2014.
© Springer International Publishing Switzerland 2014

advanced students in learning from examples and learning from supported problem solving. Research shows that students need different levels of assistance [9] and therefore ITSs should provide it adaptively.

Salden et al. [10] compared fixed faded worked-out examples with adaptive ones. Fixed faded examples are the same for all students, but the solution steps in adaptive faded examples are removed in accordance to the student's prior knowledge. They conducted two studies, one in a lab (Germany), the other in a classroom (Pittsburgh). In the lab study, adaptive examples led to better learning and higher transfer compared to the other condition. In the classroom study, however, there was no significant difference in the immediate post-test, but in the delayed post-test students who used adaptive examples learned more.

Kalyuga and Sweller [11] proposed an adaptive model for using examples based on the Cognitive Efficiency (CE)[1], which is calculated form students' performance and self-reported cognitive load. They used a different formula from what was previously proposed [12][13] as it was necessary to calculate CE in real time during the experiment. Performance was based on the number of steps the student required to solve a problem. The method was tested using the Algebra cognitive tutor enriched with worked examples and faded examples. Students in the adaptive condition were allocated to one of the four stages of faded worked examples (stage 1 fully worked-out examples, stage 4 fully problem-solving tasks) based on their cognitive efficiency scores in the pre-test. All students had to proceed to the final stage of fading (stage 4) from the stage they started. In each stage, a diagnostic task decides if the student needs more information (in the forms of 2 worked examples or 4 shortened worked examples). The adaptive condition scored marginally significantly higher than the non-adaptive condition, and also showed significantly higher efficiency gains.

In our previous study, we compared learning from examples only (EO), alternating examples and tutored problems (AEP), and tutored problems only (PO) in the area of specifying database queries in SQL [8][14]. We scaffolded examples and problems with Self-Explanation (SE) prompts [15][16][17], requiring students to explain the worked examples provided or how they solved problems. The results showed that students benefitted the most from alternating examples and problems. In that study, we used a fixed sequence of examples and problems; therefore, it is possible that some students have received less or more information than they needed. This encouraged us to propose a new adaptive learning strategy that decides what type of task to present to the learner. The learning tasks are problem solving, 2-step faded examples, 1-step faded examples, and worked examples, with faded steps chosen based on the student's performance.

2 Study

The study was conducted in the context of SQL-Tutor, a constraint-based tutor [18][19] that teaches the Structured Query Language (SQL). Fig. 1 illustrates the problem-solving page in SQL-Tutor, which presents the problem text and the database schema. Students write queries by filling in the necessary boxes for the SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY clauses.

[1] Cognitive efficiency = Performance / Cognitive Load.

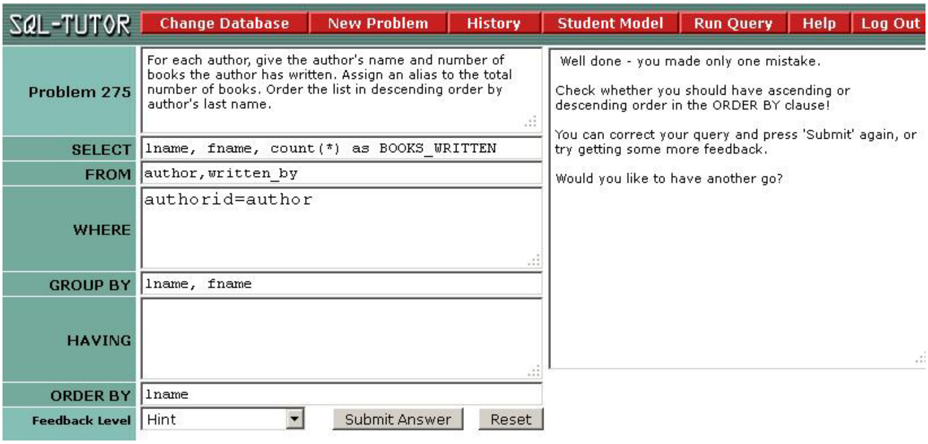

Fig. 1. Problem-solving environment in SQL-Tutor

Students can choose the level of feedback they want to receive in case their answer is incorrect. The level of feedback defines how much assistance is provided to the student. SQL-Tutor offers six levels of feedback: positive/negative feedback, error flag, hint, all errors, partial solution and complete solution. Positive/negative feedback has the lowest level of assistance, and it informs students whether their answer is correct or not. The message also shows how many errors students have in their solution. An error flag message identifies the clause in which the error happened. More information about the type of error will be provided when a hint-type feedback is requested (illustrated in Figure 1). The partial solution shows the correct content of the clause which the student got wrong. Feedback of type *all errors* displays hint-type messages for all errors the student has made. At the maximum level, the complete solution simply reveals the pre-specified ideal solution of the problem. When a student starts solving a new problem, the default feedback level is positive/negative. The student can attempt the same problem as many times as needed [19].

The version of SQL-Tutor used in this study had four modes: problem solving, 2-step or 1-step faded example, and worked example. The problem-solving mode is similar to the original SQL-Tutor. The 2-step / 1-step faded example modes differ in that the student needs to complete two or just one clause. The worked example mode presents the completed solution and an explanation.

The study was conducted in a single, 100-minute long session in which the participants (46 undergraduate students from the University of Canterbury) studied ten pairs of isomorphic tasks of increasing complexity. Fig. 2 shows the design of the study. The students took a pre-test for 10 minutes, consisting of eight multiple-choice and two problem-solving questions. The multiple-choice questions measured conceptual knowledge (one mark each). For the problem-solving questions, students had to write SQL queries (four marks each). Participants were randomly allocated to either the control (22 students) or experimental group (24).

	Control	**Experimental**
	n = 22	n = 24
	Pre-test	
Pair 1	1st task: problem 2nd task: example	1st task: problem 2nd task: rehearsal task (problem, 2/1 step faded example, worked example, or skip)
Pair 2 to 10	1st task in each pair: example 2nd task in each pair: problem	1st task in each pair: preparation task (problem, 2/1 step faded example, worked example, or skip) 2nd task in each pair: problem
	Each problem followed by a C-SE prompt and each example followed by a P-SE prompt	Each problem or faded example followed by a C-SE prompt, and each example followed by a P-SE prompt
	Post-test	

Fig. 2. Design of the study

The control condition worked with example-problem pairs: each pair consisted of an example followed by an isomorphic problem to solve. The only exception is the first pair, in which the control group received a problem followed by an example. Therefore, the control condition in this study is identical to the best condition (AEP - alternating examples/problems) from [8] with the exception of the first pair. The experimental group had pairs consisting of a preparation task followed by a problem, except for the first pair. The first pair consisted of a problem followed by a rehearsal task; this was necessary so that this problem can provide the necessary information for the adaptive strategy. Rehearsal tasks are the same as preparation tasks, but because they were provided after the isomorphic problem we called them rehearsal tasks. The adaptive strategy decided what type of preparation task to present.

Similar to [8], we presented participants with SE prompts after worked examples and problems. Conceptual-focused Self-Explanation prompts (C-SE) and Procedural-focused Self-Explanation prompts (P-SE) are meta-cognitive prompts requiring students to reflect on concepts required to solve problems or on procedural steps of worked examples. Students were given C-SE prompts after problems or faded examples, and P-SE prompts after examples. At the end of the session, students were given 10 minutes to complete the post-test. However, students could start the post-test during the learning session and finish the study earlier. The post-test was of similar complexity to the pre-test.

The fading strategy is based on the student's performance on the current task. Domain knowledge is represented in SQL-Tutor as constraints. Every time the student submits an attempt, the system analyses it and records information about the constraints that were satisfied or violated. It is therefore possible to find out how much the student learnt about a particular domain concept by comparing his/her knowledge before and after the current problem. Our fading strategy sorts the concepts that the student learnt in the current problem and selects the concept the student learnt the most (or the top two concepts, if two steps are to be faded). Then the system fades one or two steps of the next problem. If the next problem does not include the selected concept(s), the strategy fades the next concept (or two) from the sorted list. The idea is to help students rehearse what they have just learnt.

Our adaptive strategy is based on a measure of assistance the student received while solving a problem. Table 1 shows the score H_i we assigned to each level i of feedback in SQL-Tutor. Level 0 (H_0) presents minimum assistance (score = 1) and level 5 (H_5) shows the maximum assistance (score = 6).

Table 1. Assistance scores for different levels of help

Name	**i**	**H_i**
Positive/Negative	0	1
Error flag	1	2
Hint	2	3
Partial solution	3	4
All errors	4	5
Complete solution	5	6

The easiest way to calculate the assistance score is to sum up the assistance scores of all requested help, as in Equation 1. In SQL-Tutor, students can ask for the same level of feedback several times; therefore, the assistance scores of feedback messages are multiplied by the number of times they have been requested (n_i).

$$\text{Assistance score: T} = \sum_{i=0}^{5} H_i\, n_i \tag{1}$$

When a student has seen a particular feedback message, and then requests it again, the message does not contain the same amount of new information; therefore, the assistance score should be less than Equation 1. For instance, when a student requests a complete solution, the next time s/he asks for the complete solution, the same solution will be shown. Therefore, we multiplied the assistance score by the power two series of n, with n showing the number of requests for the level of feedback (Equation 2). Power two series converges to two, as shown in Equation 3.

$$Power\ two\ series\ (n){:}\ Po(n) = \sum_{j=1}^{n} \frac{1}{2^{(j-1)}} \tag{2}$$

$$\lim_{n\to\infty} Po(n) \approx 2 \tag{3}$$

In Equation 4, we rewrite Equation 1 using Equation 2:

$$\text{T} = \sum_{i=0}^{5} H_i Po(n_i) \tag{4}$$

While Equation 4 appears mathematically sound, it does not take into account the student's behaviour after receiving feedback. For instance, the current formula shows that Student A who solved a problem by receiving H_0 H_1 H_2 (without getting a partial or complete solution), received the same information as Student B who saw a complete solution (H_5) once. It is important to distinguish between students who complete problems with minimum assistance and students who request the complete solution in the first attempt. One way is to change the scoring system we presented in Table 1. However, changing the scoring system does not help to distinguish between students who saw a complete solution in the first attempt and students who saw a complete solution after several attempts to solve the problem. For instance, students who get a complete solution after several incorrect attempts may search for their mistakes when they see the complete solution. Moreover, seeing a complete solution in the first attempt encourages students to copy the solution, which leads to shallow learning [20].

In order to include the student behaviour in the assistance score formula, we introduced parameter B, which represents the average score of requested feedback levels (Equation 5). As an example, when a student requests H_1 three times followed by H_4, the value of B is 3.5. Parameter B indicates whether the student tends to use high or low levels of assistance; for instance, if B is 2.5, the student mostly uses low feedback levels, but when B is 4.5, the student uses high levels of feedback more than low-level feedback to solve the problem.

$$\text{Student Behaviour: B } = AVERAGE\ (H_m),\quad m\ is\ the\ list\ of\ requested\ feedback\ levels \tag{5}$$

This information was not available in Equation 4. Having such information, we can design an appropriate coefficient, but would a linear coefficient be a suitable approach (Equation 6)? Equation 6 does not discriminate well between different levels of feedback. For instance, there is a small difference between B = 1, B = 2, B = 3 or B = 4. In fact, B = 4 shows that students used a partial or a complete solution to accomplish the task, while B = 3 shows that students definitely did not see a complete solution, but might use partial solution in conjunction with some other low assistance hints. Therefore, we should use different slopes for each behaviour. An appropriate function that accounts for this is shown in Equation 7.

$$\mathrm{T} = B\sum_{i=0}^{5} H_i Po(n_i) \tag{6}$$

$$\mathrm{f(x)} = \sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1 \tag{7}$$

In order to make a bigger difference between low-level and high level assistance scores, in Equation 8 we use a power two of Equation 7. Since g(x) starts from zero, we incremented the formula to avoid a zero coefficient, and obtain Equation 9. We also changed the name of the function to Skewness slope.

$$\mathrm{g(x)} = \left(\sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1\right)^2 \tag{8}$$

$$\text{Skewness slope: K(x)} = \left(\sin\left(\frac{\pi}{2}\left(\frac{x}{3} - 1\right)\right) + 1\right)^2 + 1 \tag{9}$$

Overall, from Equation 6 and Equation 9, we rewrite the assistance score formula, and Equation 10 shows the final result.

$$\mathrm{T} = K(B) \sum_{i=0}^{5} H_i Po(n_i) \tag{10}$$

We tested Equations 4 and 10 using the data from our previous study, in which 12 students solved problems in SQL-Tutor. The results show that Equation 10 leads to higher accuracy than Equations 1 and 4. Therefore, in this study we used Equation 10 to calculate the assistance score after each problem is solved.

Paas and Van Merrienboer [13] calculated cognitive efficiency as the difference between the z-scores of performance (P) and mental effort rating (R), CE = z_P - z_R. This way, CE can only be calculated after the experiment is completed. In order to determine CE in real time, Kalyuga and Sweller [11] used mental effort (R) and performance (P) to calculate Cognitive Efficiency as CE = P ÷ R. Mental effort was indicated by students, and performance was calculated from the number of steps the student required to solve a problem. Our adaptive strategy is also based on a measure of cognitive efficiency. The participants were asked to rate the mental effort (R) after solving each problem (*How much effort did you invest to complete this task?*) on a 9-point rating scale. We calculated the student's performance P from the assistance score T:

$$\mathrm{P} = T_{High} - T \tag{11}$$

When a student asks for a partial solution several times, effectively the student modifies the problem into a worked example. Examples provide maximum assistance; the assistance score for the situation when the student has seen partial solution several times corresponds to a high level of assistance which we refer to as T_{High}. Thus, using Equation 10 we calculate T_{High} to be 26 (H_3 = 4; K(4) = 3.25). Therefore, performance P can be calculated as:

$$\mathrm{P} = 26 - T \tag{12}$$

Please note that T can have a value greater than T_{High}. Because T_{High} represents turning problems into examples, we set all the assistance scores greater than T_{High} to 26. Therefore, P never becomes negative.

Performances are then scaled to the range [0, 9]. Like Kalyuga and Sweller (2005), we define the critical level of cognitive efficiency as $CE_{cr} = P_{max} \div R_{max}$, where $P_{max} = R_{max} = 9$. We consider CE > CE_{cr} to be high cognitive efficiency; thus, students who solved a problem with CE > 1 were expected to be able to solve the next problem without needing a preparation task.

The first pair of tasks is different from the other pairs. In this pair, the participants worked with problem 1 followed by a rehearsal task. A rehearsal task is the same as a preparation task, but because this preparation task is provided after problem 1, we refer to it as a rehearsal task. If the student's CE is greater than 1 in problem 1, the system skipped the rehearsal task from the first pair and the preparation task of pair 2. As CE scores were updated after solving problems only, in the preparation task of the second pair the students received the same type of task as the rehearsal task from the first pair. The system behaviour for the second pair is the same as for all later pairs, as depicted in Fig. 3.

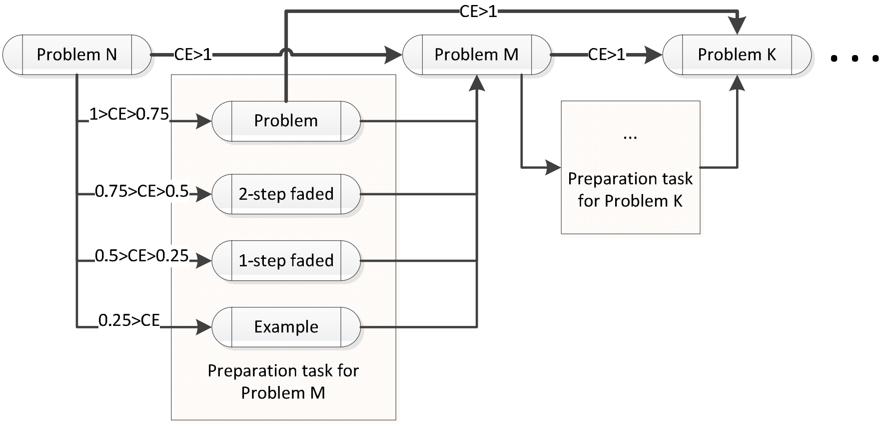

Fig. 3. Study flow

Our adaptive strategy uses cognitive efficiency CE to decide whether the student needs preparation before the next problem as shown in Table 2. A CE of below 1 and above 0.75 (6.75/9) shows relatively good performance on the current problem, but indicates the need to prepare for the next problem by solving an isomorphic problem first. Students with CE between 0.75 (6.75/9) and 0.25 (2.25/9) receive 2-step or 1-step faded examples as the preparation task. As we mentioned before, the steps are faded based on how much the student has learnt from the current task for each concept. Students who scored below 0.25 (2.25/9) get an isomorphic worked example before solving the next problem. When the student asked for a partial solution more than twice, or saw the complete solution, the strategy presents a worked example as a preparation task regardless of the student's CE. The system calculates the CE score only after problems. If a student performed well (CE>1) on a problem which is shown as a preparation task, the system skips the next problem and the preparation task for the subsequent problem.

Table 2. Decision table

Condition	CE>1	1>CE>0.75	0.75<CE<0.5	0.5<CE<0.25	CE<0.25
Preparation type	Skip preparation	Problem	2-step faded example	1-step faded example	Worked example

3 Results

The basic statistics about the two groups are presented in Table 3. There was no significant difference between the pre-test performances of the two groups. The t-test revealed a significant difference between the post-test results. The post-test performance of the control group was significantly lower than the experimental group. The students in both conditions improved significantly between the pre- and the post-test, as shown by the paired t-tests reported in the *Improvement* row of Table 3. Correlations between the pre- and post-test scores are also reported in Table 3, but only the control condition had a significant correlation (p < .01, r = .55). There was also a significant difference between the mean learning times of the two groups. The experimental group spent significantly less time in the intervention than the control group.

Table 3. Basic statistics for the two conditions (* denotes significance at the 0.05 level)

	Control (22)	**Exper. (24)**	**p**
Pre-test (%)	50.3 (13.7)	45.3 (18.9)	.31
Post-test (%)	77.8 (13.9)	85.7 (12.6)	*.05
Improvement	*p<.01, t=-9.9	*p<.01 , t=-10.5	
Pre/post-test correlation	p<.01, r=.55	p=.10, r=.34	
Learning time (min)	73.6 (16.3)	58.9 (19.0)	*<.01
Normalised learning gain	.56 (.25)	.73 (.20)	*.01
Conceptual knowledge gain	.76 (.30)	.88 (.18)	.13
Procedural knowledge gain	.30 (.38)	.62 (.37)	*<.01
Number of problems solved (incl. faded)	7.0 (2.5)	8.6 (3.0)	.06
Problems solved (excl. faded examples)	7.0 (2.5)	6.9 (2.4)	.95
2-step faded		.8 (1.2)	
1-step faded		.9 (1.2)	
Number of examples	7.9 (3.0)	1.8 (1.9)	*<.01
Number of attempts per problem	4.5 (2.0)	4.3 (1.7)	.72
Maximum complexity level	13.4 (5.2)	14.0 (5.3)	.71

The normalised learning gain[2] of the experimental group was significantly higher than the gain of the control group. When we analysed normalised learning gains on the conceptual knowledge questions (questions 1 to 8), we found no significant difference between the groups. On the other hand, the normalised learning gain on procedural knowledge (questions 9 and 10) of the experimental group was significantly higher than that of the control group (p < .1).

The experimental group participants solved marginally significantly more problems than the control group (p = .06), when faded examples are included. In order to solve faded examples, students had to fill in the faded steps. Therefore, we analysed the number of problems solved, excluding faded examples, and there was no significant difference between the two groups. The average number of 2-step faded examples solved by the experimental group is 0.8, and the average for 1-step faded examples is 0.9. The experimental group received significantly fewer examples than the control group (p < .01). There was no significant difference in the number of attempts per problem between the two conditions. The problem complexity gradually increased from pair 1 to pair 10. There was no significant difference between the average maximum complexity levels of problems the students in the two groups solved.

Students rated their mental effort after they solved problems (not after examples and faded examples, as we could not calculate performance in those cases), which the

[2] Normalised learning gain = (Post test - Pre test) / (Max score - Pre test).

adaptive strategy used to calculate CE. As mental effort rate is specified on a 9-point scale, we used non-parametric tests for this analysis. We used Spearman's rho test to analyse correlations, reported in Table 4. We found significant negative correlations between the pre-test scores and mental effort ratings, as well as between mental effort and CE, for both groups. There were significant positive correlations between the pre-test and CE for both groups. Next, we used the Mann-Whitney U test to compare the groups on CE and mental effort. There is no significant difference between the experimental and control groups ($p = .24$) on reported mental effort, but the experimental group had a marginally significantly higher CE scores than the control group ($p = .09$).

Table 4. Cognitive efficiency and mental effort analysis

	Control	**Experimental**	**p**
Correlation: pre-test and mental effort	p=.03, r = -.48	p=.02, r = -.48	
Correlation: pre-test and CE	p<.001, r = .69	p=.03, r = .44	
Correlation: mental effort and CE	p=.001, r = -.67	p<.001, r = -.73	
Cognitive Efficiency (CE)	2.28 (2.29)	2.70 (1.85)	.09
Mental effort	4.77 (1.71)	4.38 (1.20)	.24

As mentioned earlier, the participants received C-SE prompts after problems and P-SE after examples. We analysed the SE success rates for the two groups, which are reported in Table 5. There was neither significant difference between the groups in the overall SE success rate nor in the P-SE success rates, but there was a marginally significant difference in the C-SE success rate ($p = .08$). Students in the experimental condition performed better on C-SE than the control group.

Table 5. Analyses of SE prompts

	Control	**Experimental**	**p**
Overall SE success rate	82.6 (12.2)	88.0 (12.5)	.14
Procedural SE success rate	90.3 (12.9)	90. (11.5)	.97
Conceptual SE success rate	73.6 (15.9)	84.0 (20.1)	.08

Overall, the results show that the experimental group participants, who worked with the adaptive strategy, learnt more and faster than the control group. The results clearly show the effectiveness of our adaptive strategy in comparison with the non-adaptive sequence.

4 Conclusions

In this study, we compared a fixed sequence of alternating examples and problems with a strategy that adaptively decides how much assistance the student needs. The adaptive strategy determines the type of task (a worked example, a faded example or a problem to be solved) based on how much assistance the student received on the previous problem. We proposed a novel approach to measure the performance score.

Using performance and mental effort scores enable us to calculate the cognitive efficiency, which is then used to choose appropriate learning tasks for students. The fading strategy is also adaptive: the system fades the solution steps about the concepts that the student learnt the most in the previous task. The results show that the experimental group learnt more and faster than the control group.

Prior research has shown that adaptive faded examples are superior to non-adaptive faded examples [10], but their fading strategy was based on students' performance in answering self-explanation prompts. In our study, we used the student model to see how much students learnt about each concept, and then faded the steps about the concepts students learnt the most in the previous problem. Prior research also used cognitive efficiency to provide appropriate learning tasks [11], but they used students' performance which was based on how many steps students required to solve testing tasks. In our study, we measured cognitive efficiency based on how much assistance students received when solving problems.

Using our approach, an ITS can use assistance scores to identify novices and advanced students. If the system knows that a student is novice or advanced, then it is possible to provide proactive messages.

We have evaluated the adaptive strategy in the area of specifying SQL queries. One of the limitations of our study is the relatively small sample size. We plan to perform additional studies with a larger set of participants. It is also important to evaluate the adaptive strategy in other types of instructional tasks in order to test its generality. In future work, we plan to combine self-explanation scores and assistance scores to measure performance more accurately, which will result in improved cognitive efficiency scores. We also plan to evaluate such an improved performance measure and the adaptive strategy in other domains, including those with well-defined tasks.

References

1. Sweller, J., Cooper, G.A.: The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. Cognition and Instruction 2, 59–89 (1985)
2. Kalyuga, S., Chandler, P., Tuovinen, J., Sweller, J.: When problem solving is superior to studying worked examples. Educational Psychology 93, 579–588 (2001)
3. Sweller, J.: The worked example effect and human cognition. Learning and Instruction 16, 165–169 (2006)
4. Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from Examples: Instructional Principles from the Worked Examples Research. Review of Educational Research 70, 181–214 (2000)
5. van Gog, T., Rummel, N.: Example-Based Learning: Integrating Cognitive and Social-Cognitive Research Perspectives. Educational Psychology Review 22, 155–174 (2010)
6. Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., Salden, R.: The worked-example effect: Not an artefact of lousy control conditions. Computers in Human Behavior 25, 258–266 (2009)
7. McLaren, B.M., Isotani, S.: When Is It Best to Learn with All Worked Examples? In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS, vol. 6738, pp. 222–229. Springer, Heidelberg (2011)

8. Shareghi Najar, A., Mitrovic, A.: Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning? In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P., et al. (eds.) AIED 2013. LNCS, vol. 7926, pp. 339–348. Springer, Heidelberg (2013)
9. Koedinger, K., Aleven, V.: Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. Educational Psychologist Review 19, 239–264 (2007)
10. Salden, R., Aleven, V., Renkl, A., Schwonke, R.: Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? Topics in Cognitive Science. Sci. 1, 203–213 (2009)
11. Kalyuga, S., Sweller, J.: Rapid dynamic assessment of expertise to improve the efficiency of adaptive e-learning. Educational Technology Research and Development 53, 83–93 (2005)
12. van Gog, T., Paas, F.: Instructional Efficiency: Revisiting the Original Construct in Educational Research. Educational Psychologist 43, 16–26 (2008)
13. Paas, F., Van Merrienboer, J.: The Efficiency of Instructional Conditions: An Approach to Combine Mental Effort and Performance Measures. Human Factors 35, 737–743 (1993)
14. Shareghi Najar, A., Mitrovic, A.: Do novices and advanced students benefit differently from worked examples and ITS? In: Wong, L.H., Liu, C.-C., Hirashima, T., Sumedi, P., Lukman, M. (eds.) Int. Conf. Computers in Education, Indonesia, Bali, pp. 20–29 (2013)
15. Chi, M.T.H., De Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive Science 18, 439–477 (1994)
16. Brown, A.L., Kane, M.J.: Preschool children can learn to transfer: Learning to learn and learning from example. Cognitive Psychology 20, 493–523 (1988)
17. Hattie, J.: Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement. Routledge, New York (2009)
18. Mitrović, A.: Experiences in implementing constraint-based modeling in SQL-Tutor. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) ITS 1998. LNCS, vol. 1452, pp. 414–423. Springer, Heidelberg (1998)
19. Mitrovic, A.: An Intelligent SQL Tutor on the Web. Artificial Intelligence in Education 13, 173–197 (2003)
20. Deeks, A.: Web Based Assignments in Structural Analysis. Centre for Educational Development, Nanyang Technological University (2000)