Parsons Problems to Scaffold Code Writing: Impact on Performance and Problem-Solving Efficiency

Xinying Hou
xyhou@umich.edu
University of Michigan
Ann Arbor, Michigan, United States

Barbara Jane Ericson
barbarer@umich.edu
University of Michigan
Ann Arbor, Michigan, United States

Xu Wang
xwanghci@umich.edu
University of Michigan
Ann Arbor, Michigan, United States

ABSTRACT

Novice programmers struggle with writing code from scratch. One possible way to help them is by using an equivalent Parsons problem on demand, where learners place mixed-up code blocks in the correct order. In a classroom study with 89 undergraduate students, we examined how using a Parsons problem as scaffolding impacts performance and problem-solving efficiency. Results showed that students in the Parsons as Help group achieved significantly higher practice performance and problem-solving efficiency than students who wrote code without help, while achieving the same level of posttest scores. These results improve the understanding of Parsons problems and contribute to the design of future coding practices.

CCS CONCEPTS

• **Applied computing → Interactive learning environments**; **Computer-assisted instruction**; **Education**.

KEYWORDS

Parsons problems, Introductory Programming, Code Writing

ACM Reference Format:

Xinying Hou, Barbara Jane Ericson, and Xu Wang. 2023. Parsons Problems to Scaffold Code Writing: Impact on Performance and Problem-Solving Efficiency. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2023), July 8–12, 2023, Turku, Finland.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3587103.3594182

Novice programmers find writing code challenging, particularly when learning new topics in introductory programming courses. Parsons problems are popular programming exercises where students place mixed-up code blocks in the correct order. They can help novices who struggle while writing code from scratch. In an earlier study, there was a ceiling effect on the pretest when investigating the benefits of using a Parsons problem to scaffold code writing on Python 3 Basics, suggesting that students had already mastered this topic [3]. The current study examined the effects of Parsons problem as scaffolding when learning to write Python classes before this topic was covered in the course.

An in-class experiment was conducted in a Python programming course at a large public research university in the northern US. The experiment had two conditions: Parsons-Help (PH) - a text-entry write-code interface allowing students to open an adaptive Parsons problem when needed, and No-Help (NH) - only the text-entry interface. Students first completed three parts in order: introduction to classes and objects, system introduction, and a survey of prior knowledge. Then participants were assigned to a condition, practiced four write-code problems, and completed a posttest. The final sample contained 89 students who completed all of the materials in order (41 in PH and 48 in NH). For each student, the *problem-solving efficiency* was calculated following the likelihood model from Hoffman and Schraw [2]: Write-code practice score (*Max* = 40) / Practice time (mins). Practice time was calculated as the time used for practice, excluding any periods of inactivity over 5 minutes. The highest efficiency is 4.07, achieved by a student who finished all four write-code practice problems (40 points) in 9.83 minutes.

We built three linear regression models, using the practice score, problem-solving efficiency, and posttest score as the dependent variables, the assigned condition as the independent variable, and self-rated prior knowledge as a covariate. When controlling for students' prior knowledge, PH students got significantly higher practice scores ($M = 20.98$) on average compared to NH ($M = 10.83$), $p = 0.003$, Cohen's $f^2 = 0.11$ (small [1]). Similarly, when controlling for prior knowledge, PH students ($M = 1.34$) had significantly higher problem-solving efficiency on average during practice compared to NH ($M = 0.81$), $p = 0.041$, Cohen's $f^2 = 0.05$ (small [1]). However, when controlling for prior knowledge, there were no significant differences between students' posttest scores, even though PH students ($M = 15.15$ of 40) achieved higher than NH students on average ($M = 11.51$ of 40), $p = 0.349$.

Our study shows that using scaffolding, such as Parsons problems, can significantly enhance practice performance and efficiency when practicing code writing, while maintaining a similar level of learning performance. This finding also leads to future implications on appropriate Parsons problems as scaffolding. For example, personalizing Parsons problems and providing faded scaffolding for those closer to a correct solution can be effective strategies.

Acknowledgements The funding for this research came from the National Science Foundation award DUE-2143028. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences.* Routledge.
[2] Bobby Hoffman and Gregory Schraw. 2010. Conceptions of efficiency: Applications in learning and problem solving. *Educational Psychologist* 45, 1 (2010), 1–14.
[3] Xinying Hou, Barbara Jane Ericson, and Xu Wang. 2022. Using Adaptive Parsons Problems to Scaffold Write-Code Problems. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1.* 15–26.

ITiCSE 2023, July 8–12, 2023, Turku, Finland

ACM ISBN 979-8-4007-0139-9/23/07.
https://doi.org/10.1145/3587103.3594182