AFFORDANCES OF THE 'BRANCH AND BOUND' PARADIGM FOR DEVELOPING COMPUTATIONAL THINKING

Joris van der Meulen and Mark Timmer

ELAN, Department of Teacher Development, University of Twente

As technological advances in engineering and computer science happen more and more quickly, we must shift focus from teaching specific techniques or programming languages to teaching something more transcending: computational thinking (Wing, 2006). Wing explained this concept later as "*the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out*". It includes abstraction, heuristics, algorithm design, efficiency and complexity. While programming classes add to students' competence in some of these topics, mathematics too may foster computational thinking (Weintrop et al., 2016). However, few resources are currently available to support teachers in meeting computational thinking learning goals.

This design-based qualitative study explores which aspects of computational thinking can be addressed well through a mathematics project for secondary school students aged 16-17 in the Netherlands. As puzzle-like problems help students to think more algorithmically (Levitin & Papalaskari, 2002), we designed a three-hour project using such problems to familiarize students with the algorithm design paradigm 'branch and bound', which efficiently enumerates candidate solutions in discrete optimization. Allowing students to come up with heuristics, analyse the complexity of approaches, and do some calculations by hand, we aim to improve their computational thinking.

We will present our design and an evaluation of the project carried out by 50 students, discussing findings from in-class observations and interviews with 5 case students. Data collection is currently underway, and the results will be available at the time of presentation. We expect to report on our experiences in teaching the branch and bound paradigm and its affordances and limitations for helping students learn to think computationally. We focus on skills helping them contribute to tomorrow's society: algorithmic thinking, while still being able to reflect on efficiency and correctness.

References

Levitin, A., & Papalaskari, M. A. (2002). Using puzzles in teaching algorithms. *ACM SIGCSE Bulletin, 34*(1), 292-296.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.