TEACHERS' DESIGN OF HEURISTIC TREES

Rogier Bos[1] and Theo van den Bogaart[2]
[1]Utrecht University, Freudenthal Institute
[2]HU University of Applied Sciences, Archimedes Institute, Utrecht

Abstract

In scaling up the use of heuristic trees to facilitate students' mathematical problem solving, we developed a design course and design protocol for heuristic trees. However, designing heuristic trees is a challenging task. The study reported in this paper aims to collate an inventory of teachers' difficulties in designing heuristic trees. We analyzed a sample of heuristic trees teachers designed after participating in the course. Through open coding we arrived at a list of mistakes in the designs, and showed how these relate to the design principles for heuristic trees. We see support for the conclusion that most of the design principles are not straightforward to implement. In particular, teachers need to address the general techniques and concepts in the problem, and provide support for students in a way that needs no further intervention by the teachers.

Keywords: mathematics education, teacher education, problem solving, heuristic trees.

INTRODUCTION

The importance of both teaching problem solving and teaching through problem solving is generally acknowledged by researchers (e.g., Lester & Cai, 2016). Problem solving is considered an important 21st century skill that should be central in the modern curriculum. Additionally, teaching *through* problem solving invites students to raise their ability to apply knowledge to more challenging situations. The goal is to promote a reorganization of the students' knowledge of concepts, theory, and techniques in such a way that not only reproduction in familiar situations is achievable, but also a creative approach in new problem situations.

However, it is a challenge for teachers to implement problem solving (Lester, 2013). Students get stuck in a problem and need help, but a teacher cannot always be there in time to support the students, either because of the large amount of students that need individual help, or because the teacher is not around (e.g., in homework situations). This causes frustration for students, who have the tendency to consult the "model answers" instead of thinking about the problem themselves. To support problem solving in the absence of a teacher, the authors proposed a digital tool called heuristic tree (Bos, 2017; Bos & van den Bogaart, 2021, for an example of a heuristic tree, see https://edspace.nl/htree/heuristiekboom.php?boom_id=146). The tool provides students with just in time support through hint cards ordered in a tree structure. These hints are

Corresponding author: Rogier Bos

specific for the problem at hand, so each problem needs its own, tailor-made heuristic tree. For designers of heuristic trees we provide detailed design principles on how to formulate and structure the help within the tree. The design principles of heuristic trees, reflect the underlying theoretical ideas on the phases of problem solving (Pólya, 1945), and on compression of mathematical knowledge (Tall, 2013; Thurston, 1990). Studies have shown that heuristic trees allow students to work in absence of a teacher and to engage strongly with problems, maintaining ownership of the solution methods (Bos, 2017; Bos & van den Bogaart, 2021; Lemmink, 2019).

In order to scale up the use of heuristic trees, teachers need to become designers, creating heuristic trees that match the requirements of their own students. To facilitate this, we designed an author environment on our website https://edspace.nl/htree/. All tools on this website are open source. However, designing heuristic trees following the design principles is not a straightforward task. Hence, our aim in this study is to investigate how teachers design heuristic trees, and what support teachers need in this process. To this purpose, we developed a heuristic tree design protocol and a short course. We analyzed the heuristic trees designed by the in-service teachers that attended the course. We expected that following the design protocol and implementing the design principles would be a challenge for the participating teachers. The way teachers did this informed us to what extent they were able to identify and bring to the fore in their help the compressed, general concepts, techniques, and theory that underly the problems they set their students. The designed heuristic trees also reflected their personal ideas on how to support students' problem solving. In addition, this study also contributes to our understanding of how teachers prepare and structure help for students' problem solving tasks.

THEORETICAL FRAMEWORK

Compression and decompression. As one learns, knowledge is organized and reorganized. A central way of reorganizing one's mathematical knowledge is compression. Compression can be applied to concepts, techniques, and statements (propositions) and it can concern cases or steps. Our notion of "technique" ranges from general heuristics, like "drawing a helpline" (Pólya, 1945), to more specific algorithms, such as long division. *Compression on cases* is characterized by a shift of attention from a multitude of phenomena (cases) to the common properties of those phenomena (Bos & van den Bogaart, 2021; Tall, 2013; Thurston, 1990). *Compression on cases for a concept* refers to the conception of a new category for a multitude of cases, see Table 1 for an illustration. *Compression on cases for techniques or statements* means recognizing that the technique or statement does not only apply to a single case but to an overarching category of cases. *Compression on details of a concept* means that the definition or important properties of the concept are stored in long-term memory or reference books. *Compression on details of a technique or statement* means that relevant steps and conditions are stored in long-term memory or reference books. Both forms of compression entail a form of hierarchical reorganization in which a multitude is represented by a singular, allowing thought with less working memory involved. Such thought in bigger chunks is essential in overcoming the multiple steps that might constitute a mathematical problem.

Table 1. Examples of six types of compression

	Cases	Details
Concepts	Cases of similar triangles are considered as a new category or concept, e.g., right triangles.	The defining property of a right triangle is that it has one right angle.
Techniques	Considering the Pythagorean Theorem as a technique for cases where a length needs to be computed	For the Pythagorean Theorem as a technique storing steps like identifying a right triangle, labeling the sides, filling in the unknowns, substituting in $a^2 + b^2 = c^2$, etc., and replacing them by a more general description, like "computing one side, knowing the others".
Statements	Recognizing the Pythagorean Theorem (as a statement) applying to all cases of right triangles	For the Pythagorean Theorem (as a statement) storing details of a line of reasoning that supports the statement (a proof).

Compression is key to problem solving. It is essential, for example, to learn to recognize and apply techniques to a wider class of situations than where they are taught. Moreover, solutions to problems are discovered by thinking in terms of general techniques and reasoning with general statements, rather than in terms of the multiple steps and the details that compose them. This relates precisely to some central challenges that students face while problem solving: Recognizing which general techniques and reasonings to apply in concrete cases, and supplying and applying the adequate details and steps for the concrete problem situation. Decompression refers to the inverse cognitive process – applying concepts, techniques and statements to concrete cases and supplying adequate steps and details. Decompression is a central feature of heuristic trees.

Heuristic trees: design principles. A heuristic tree consists of help cards ordered in the shape of a tree, see Figure 1. The content of a card can only be accessed by clicking on it, and the content can only be revealed if the card closer to the trunk has already been accessed. There are several branches for each phase of problem solving: orientation, making and executing plans, completion. The design of a heuristic tree is shaped by a list of six design principles (Bos, 2017).

P1. *Compression-decompression ordering:* The order along a branch should be from general to more concrete hints, thereby decompressing the initial concepts, techniques and statements.

P2. *Logical ordering:* The structure of the tree (both along and across branches) should represent a logical order of reasoning within a solution model. It should highlight the main structure of the argument, and separate main and side issues.

P3. *Problem solving phases ordering:* The branches should be ordered following the phases of problem solving: orientation, making and executing plans, completion.

P4. *Independence:* The help offered in different branches should be independent stepping stones, in the sense that for the help offered in one branch no information in any of the other branches should be needed.

P5. *Rationing:* Each click should not give more help than needed.

P6. *Revelation:* The questions that are shown on the unopened cards should give an indication of what help can be obtained along that branch, but not give away any content. In the same way, a hint on a card

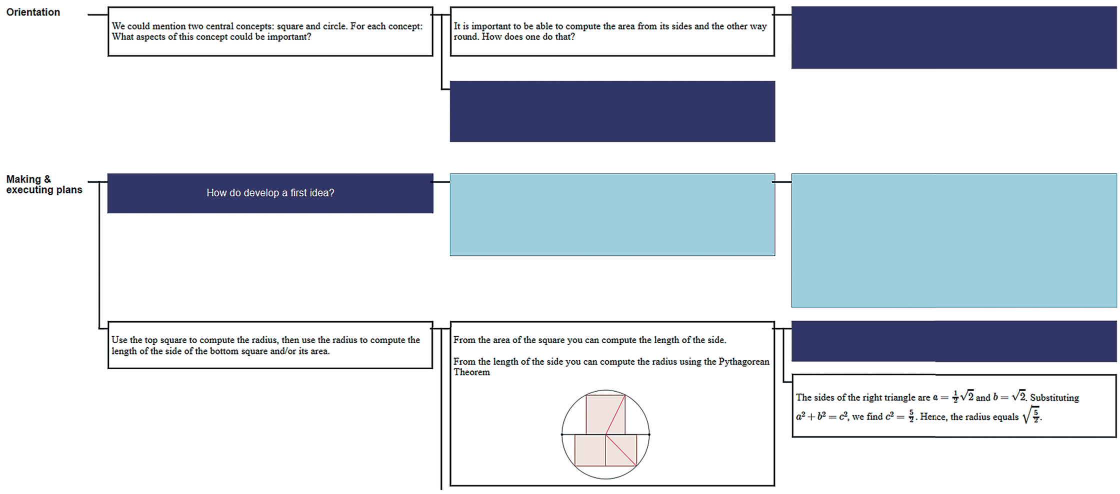

Figure 1. Top half of a heuristic tree(see https://edspace.nl/htree/heuristiekboom.php?boom_id=146# for the full heuristic tress. Note also that blue cards are closed, where dark blue cards are in line to be opened next).

should indicate what can be expected in the next hint, but not give away the content.

Principle 1 (P1) emphasizes the importance of inviting students to compress their mathematical knowledge on concepts, techniques and theory with respect to the problem. Each time students need help, they will first be exposed to a compressed version. Only next, students will find what the details are and how they apply to the problem at hand. Principles P2, P3, and P6 when implemented, help to navigate the tree and find the help suitable for the phase at which they are stuck. Implementing principles P4, P5, and P6 supports the students to maintain maximal ownership of the solution by providing no more help than is needed. Even though opening all cards would reduce the problem to following a set of steps, the goal of the tree is for students to first encounter help on the general concepts, theory and techniques, thereby inviting students to develop conceptual and heuristic thoughts. The goal is for students to develop so that they no longer need to proceed to the end of the branch before they know how to continue their problem-solving approach.

Consider the problem: You have a string of 1 meter in length. Place this string against a wall in such a way that the string forms three sides of a rectangle and the wall the fourth side. How should you lay the string so that the area of the rectangle is maximal?

Figure 2 shows the orientation phase branches for the supporting heuristic tree. The questions for the closed cards are placed above the card content for the left-most cards. Note how these questions and the following information is carefully chosen to suit the orientation phase, following P3. Note that the cards address activities that prepare for embarking on a plan, but do not suggest one. The branches in the orientation phase might lead to an idea, but do not suggest any concrete plans of attack. Also note that the questions always indicate but not reveal the content of the next card following P6. The figure shows precisely which type of decompression is used in each step to the left, making distinctions between decompression on cases and on details.

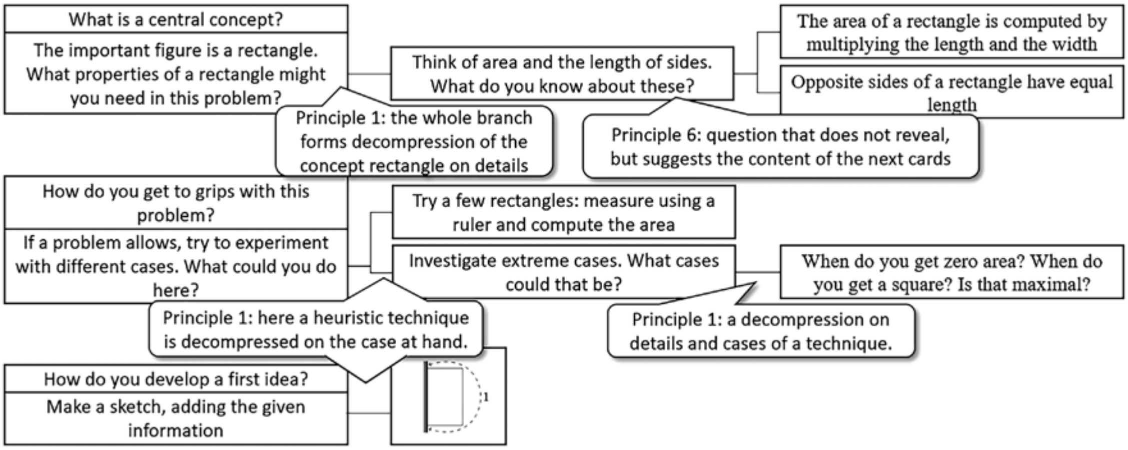

Figure 2. Annotated branches of the orientation phase

Figure 3 shows the branches of the making and executing plans phase. Note how the first branch supports the development of an overarching plan. The second and third branch represent the logical ordering of the problem approach (P2). Note how the first card of the third branch deals with independence (P4), since inevitably the formula for A is pre-knowledge for this branch. Branch three demonstrates the rationing (P5) by each time trying to leave as much of the work and discovery for the student to do. Figure 4 shows a single branch of the completion phase.

Note that decompression on statements does not play a role here. Compression on statement plays a more important role if the solution of the problem needs more theoretical development, such as proof-situations. Bos and van den Bogaart (2022) provides more information on how heuristic tree design and implementation relates to other developments in the research field of problem-solving in mathematics education.

Research questions. As stated before, designing a heuristic tree for students involves several challenges for the teacher: Foremostly adhering to the design principles, but also finding the right level for a diverse set of students, choosing what solutions to include, etcetera. With an interest to scaling up the use of heuristic trees and educating teachers in the design process, we want to investigate what difficulties teachers have with designing heuristic trees. In particular, in this paper we aim to answer the following research questions: (1) To what extent do teachers apply the design principles, in particular the principle of compression and decompression? (2) What difficulties do teachers encounter while designing heuristic trees?

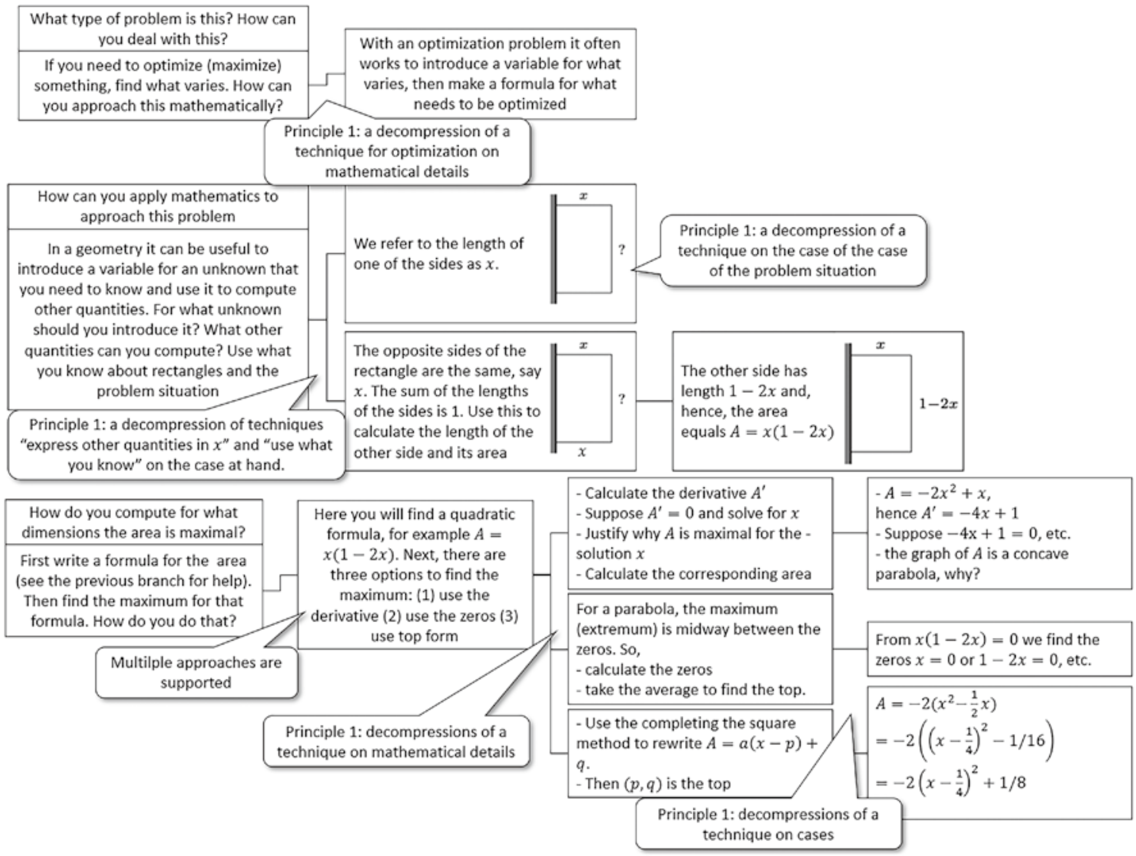

Figure 3. Annotated branches of the making and executing a plan phase

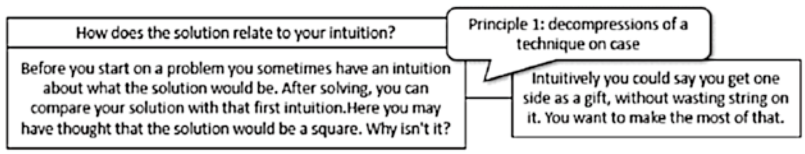

Figure 4. Branch of the completion phase

METHOD

The intervention consisted of a workshop on heuristic tree design, which was made up of two meetings of approximately 120 minutes. This workshop was pretested in a workshop at the ICTMT15 conference in Copenhagen before it was used for this research. The main components of this workshop were for the first meeting (1) a problem solving session supported by existing heuristic trees (online), (2) some theoretical background on heuristic trees, (3) a paper-and-pen heuristic tree design session; and for the second meeting (1) an online implementation session, (2) a try-out session of the participant-developed materials, (3) a critical discussion and feedback session.

The workshop participants received the heuristic tree design protocol developed by us, as a guideline for designing heuristic trees. The protocol consists of the eight steps listed in Table 1.

Table 1. Steps of a design protocol for heuristic trees

Design stage	Guiding steps
Preparation	1. List relevant concepts 2. List the useful general techniques and theory 3. List key insights 4. List where you expect difficulties for the students
Implementation	5. Make horizontal branches exhibiting decompression of a concept, technique or statement, on cases and techniques. 6. For each branch, formulate a closed card question. Use general phrasing, e.g., – What are central concepts for this problem? – How do you start? – What is the key step? – How can you generalize this problem? 7. Make the tree, putting the branches in a logical sequence rooted in three phases: – Orientation – Making & Executing Plans – Completion
Validation	8. Check your design against the design principles

The workshop was held in October and December at HU University for Applied Sciences in the Netherlands. The workshop was attended by participants of a master course for in-service mathematics teachers. The participants of the workshop were invited to share the heuristic trees they had designed during the workshop on the website www.edspace.nl/htree. These heuristic trees were analyzed using an open coding approach (Robson, 2011), with some predefined categories of codes based on the design principles. The final set of codes is presented in Table 2. The two authors coded the heuristic trees independently. The differences in coding between the two authors were then discussed, after which a final coding was settled. Each author also selected representative examples of each registered code, from which the authors jointly decided on a selection of the most informative for the paper.

RESULTS

We analyzed seven heuristic trees sent in by participants of the course. These trees featured a total of 153 cards in 48 branches. Table 2 shows the codes that were the result of the open coding process.

Table 2. The set of codes and the counted frequencies of occurrences in the trees in the data set

Design princple	Codes	Frequency
1. (De)compression order	i. An opportunity for decompression is **missed**	
	a. Concept on cases	1
	b. Concept on details	2
	c. Technique/statement on cases	14
	d. Technique/statement on details	7
	ii. An opportunity for decompression is **taken**	
	a. Concept on cases	1
	b. Concept on details	5
	c. Technique/statement on cases	17
	d. Technique/statement on details	15
	iii. The order along the branch is sequential	13
	iv. A concept, technique or statement could be formulated in a more general way as an advice	12
2. Logical order	Jumps in the reasoning going to the right or down along the tree	3
3. Phases order	Placing support in the wrong phase	9
4. Independence	A hint is dependent on information in a different branch.	11
5. Rationing	A hint is suddenly given, without opportunity for students to come up with this idea themselves.	13
6. Revelation	i. The content of the hint is neither announced on the closed card nor on the previous card	9
	ii. A question on a closed card reveals part of the solution	7
7. General	i. Questions and hints are not well formulated	21
	ii. Irrelevant information is presented	5
	iii. Help on important parts is missing	2
	iv. Information is unnecessarily repeated	1
	v. Low quality support in the completion phase	3

In some cases, decompression on cases and on details was done in one step, as in our example in Figure 2. In this case both codes 1.ii.c and d were scored.

In order to get an idea of how much a tree branches out, we computed the average number of cards that directly follow a card that is not at the end of a branch. For sample trees 1 to 7 this equals respectively: 1, 1, 1.2, 1.3, 1.6, 1, 1. Note that an average of 1 means that each card has at most one following card.

Example 1 (codes 1.i.d, 1.ii.c, 1.iii, 6.i, 6.ii). The problem concerns an isosceles triangle ABC, with AB = 10 and AC = 15. In the situation of Figure 5 on the right, compute DE. In Figure 5 on the left the two last branches of the orientation phase of the first sample tree are displayed.

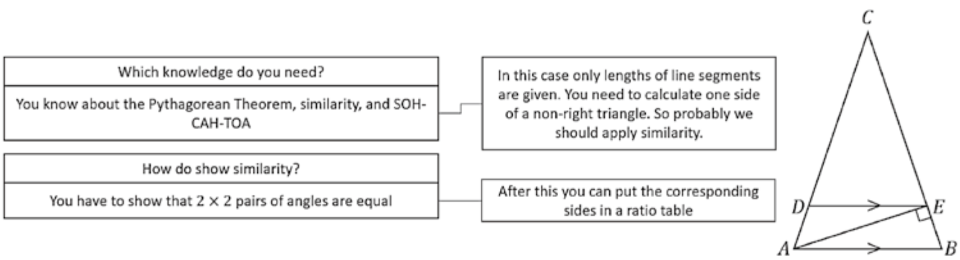

Figure 5. Two branches of the orientation phase of sample tree number 1 (left) and a picture to support the problem.

The second card in the first branch explains why the technique of applying similarity should apply to this case. We interpret this as a decompression of a technique on cases (code 1.ii.c). However, the content of the second card is not announced in the first (code 6.i). In the next branch, the visible question reveals a central technique, thus not allowing students to come up with that themselves (code 6.ii). Again, the content of the second card is not announced in the first card; the student would probably expect more information on what pairs of angles are equal (code 6.i). Instead, the cards in this branch are ordered as two consecutive steps (code 1.iii). Since this branch is part of the orientation phase the idea is to give more information about the technique of using similarity in general, for example providing more details without applying it to the specific case. This is what the question suggests that might be the content, hence we code this as a missed opportunity to decompress a technique on details (code 1.i.d).

Example 2 (codes 1.i.d,1.ii,1.iii,1.iv,2,3,4,6.i,7.i,7.iii). The problem concerns the question what the width is of a rectangular raft with length 56 that would fit in a circular moat with inner radius 37 that is twice as wide as the raft (see the pictures in Figure 6). Figure 6 shows the last three branches from the making and executing plans phase of the second sample tree.

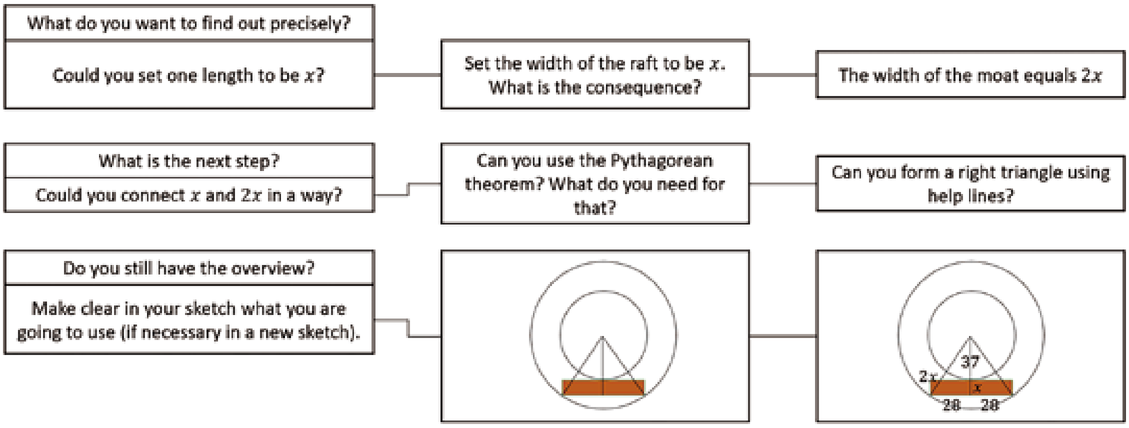

Figure 6. Three branches of the making and executing plans phase of sample tree number 2.

The top branch suggests a technique, introducing x for a length, which is then applied to the case of the raft in the moat. This is coded as decompression of a technique on a case (code 1.ii.c). This technique and also the technique of expressing the other lengths in x could be phrased more generally though (code 1.iv): "For a geometry problem try introducing a variable for a central unknown length, express the other relevant lengths in this variable, and express a relation as an equation in this variable". There is a mismatch between the question on the first card and the content. The question belongs to the orientation phase (code 3) and does not announce the content of the card (code 6.i). The question on the second card "What is the consequence?" does not have a single clear interpretation (code 7.i). The middle branch begins with the question "what is a next step". For a student that has not opened cards of the first branch this does not help navigating to the right help. That is: it is not independent (code 4). The more generally phrased technique above could have also been the beginning of this branch. Moreover, the order of suggestions is now sequential, in the sense that each next card gives a next step in a line of reasoning, instead of a decompression (code 1.iii). Moreover, both suggested and related techniques, using the Pythagorean Theorem and drawing suitable help lines, should be the starting point of a decompression on details (code 1.i.d). Generally, in this fragment the tree does not follow the logical line of reasoning, which would begin with making a sketch and possibly drawing some helplines, then introducing the algebra, and finally forming the equation (code 2). The question on the card of the third branch does not make clear that the support is about drawing a sketch and suitable helplines (code 6.i). The technique of drawing a sketch is clearly applied to the case at hand: a case of decompression on cases (code 1.ii.c). The last card suddenly shows extra information that was not announced in the previous card (code 6.i). Finally, this was the last branch of the "making and executing plans"-phase and a branch of help on the algebraic part of the solution is missing (code 7.iii).

Example 3 (codes 1.i.d, 1.iv ,1.ii.c, 1.iii, 4, 5, 6.ii, 7.i, 7.v). In the problem students are asked to give equations for all lines in the plane that have a given distance to two given points. Figure 7 shows the two branches from the making and executing plans phase, followed by the two branches from the completion phase.

The first two branches exhibit clear examples of decompression of a technique on cases (code 1.ii.c): in the top branch the distance formula is applied to the specific problem, while the second provides help on solving an equation that is specific for the problem case. An opportunity is missed to give decompression on details (code 1.i.d), especially in the second branch which fails to emphasize a general technique for solving a system of two equations. The second branch also contains a sequential progression through the algorithm to solve a system of equations (code 1.iii). In the first branch, the question on the closed card is not clear (code 7i) and the suggestion to use the distance formula comes too soon (code 5); it would be better if the card started with a text such as "How can you make an equation out of the given requirements?". Lastly, the

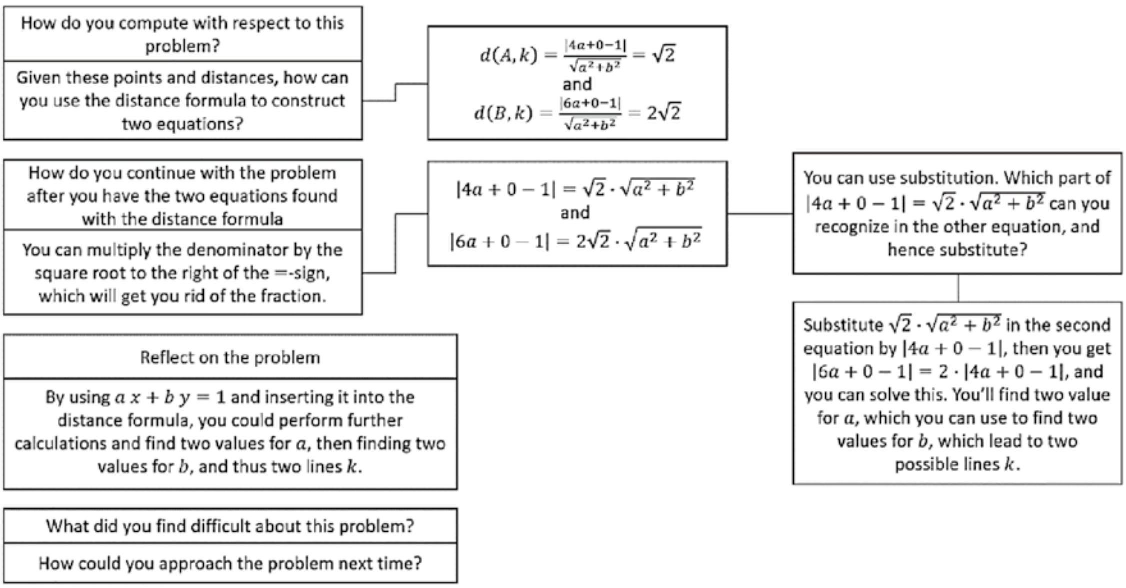

Figure 7. Two branches from the phase Making and executing plans followed by two branches from the completion phase, both from sample tree number 6.

first branch involves information about the distance formula that was on a card in the orientation phase, so the branch is not independent (code 4). Also notice that the question on the closed card in the second branch refers to the distance formula, so the closed card already gives away part of the solution (code 6.ii). The third and fourth branch are short ones; this turns out to be a typical characteristic of branches in the completion phase. But the third one is actually too short: an opportunity is missed to emphasize the general heuristic (code 1.iv) "Summarize the techniques you have used in the problem's solution". Instead, a summary of the solution is given away in the first card (code 5). The last hint is an example of low quality support in de completion phase (code 7.v). It is unclear how to make sense of the question "How could you approach the problem next time", if the problem has just been solved a certain way, and moreover, it is no support in trying to answer the question of what was difficult.

DISCUSSION AND CONCLUSION

In our discussion we follow the order of the design principles P1 to P6, and address to what extent teachers managed to follow them, and implement the advice given in the design protocol.

Table 2 shows that in total 38 opportunities for decompression were taken in the seven sample trees. Most of those opportunities concerned decompression on techniques. This might be a reflection of the approach of teachers and school book being more procedural than conceptual. In many cases, decompression on cases and details was performed at once, as in our example in Figure 2. Instead, it would be better to first present details of the techniques, allowing the students to apply the details to the case. Support on this could in turn be the content of the next card. In this way, maximal ownership of the solution process is allowed, which is a central aim underlying the heuristic trees' design principles.

We observed 24 instances where an opportunity to apply decompression along the branches was missed, and in 12 cases an opportunity to phrase a technique more generally was missed (code 1.iv). This casts doubt on whether the protocol steps 1 and 2 were performed in those cases, and whether the idea of decompression as a central structuring element was fully understood. This leads us to reconsider our approach of the design course. The notion of compression may be to technical; the participants may be better supported by a more practical approach. For example, a more concrete advice to promote a compression structure – without using the term – could be: "start a branch by formulating advice in such a way that it could apply to a similar problem. Try to introduce concepts, technique, and statements in general, before providing details or applying them to the context of the problem".

Teachers seem to have little difficulty following the logical order of a solution method (see Table 2, code 2). However, support might end up in the wrong phase. In particular, we observed too much detail on solution plans already being stated in the orientation phase, or the other way around, suggestions that would suit in the orientation phase – like making a sketch – are presented in the second phase. The support in the orientation phase might at best lead to an idea of an approach, but should not address any concrete plan.

Teachers have a tendency to order information sequentially along a branch (13 occurrences, see Table 2, code 1.iii). The disadvantage of this is that students need to view information on steps that they have already solved to arrive at suitable help, and that opportunities for decompression are missed. The course needs to draw attention to this issue. If, for example, the solution plan consists of three major heuristic techniques, then these should be announced in a first card, followed by a branching into three outgoing lines to cards that decompress on details of those techniques. We observed that four out of seven trees have an average of one outgoing card, that is, no use of branching is made. Even in the other three trees, that have higher averages, we do hardly observe use of this method. We conclude that this should be addressed explicitly in the design protocol.

Let us address design principles P4, P5, and P6. Maintaining independence between information in different branches is a challenge for teachers. There are 11 cases in our sample where this goes wrong (Table 2, code 4). This is difficult indeed, and the only way out is either to strategically repeat information when necessary, or to refer explicitly to other branches for more help. See for example how this is solved in the bottom branch of our example in Figure 3. There are 13 cases where teachers present a hint that gives away information of the solution, without announcing it properly so that students have a chance to find this part of the solution themselves. This happens in particular when revealing a useful technique in the question on a closed card, as in example 3 and Table 2, code 6.ii. It is also related to a lack of decompression, when a technique is immediately presented in detail and applied to the case in the first card of a branch.

Finally, we observed 21 cases where there was an issue with the formulation of the support on the cards. The text is not clear enough in suggesting what to do or think of next. This is a consequence of teachers applying a conversational tone on the cards, as they might use in classroom while supporting a student. The problem is that, whereas in classroom teachers can always clarify themselves, in a heuristic tree the text should be completely self-explanatory. The text should be precise and reveal a well-considered amount of new information. Even though it is good practice to support students by posing questions, revealing a general technique can take a different shape. A good card often contains some new information in the form of a statement, then followed by some questions announcing the following cards.

For the validity of this study, it is important that the participants were given a fair chance to acquaint themselves with the protocol and the principles, and we believe they did through the course. We conclude that, even though the teachers designed heuristic trees that satisfied many of the principles we set out and communicated through the protocol, many issues need to be addressed more effectively in the course, most notably decompression and formulation. It is not clear whether missed opportunities to use decompression come from an inability of the teachers to formulate concepts and techniques in more general terms, or by a lack of attention for this aspect. This could be an issue for further research in the form of interviews with the involved teachers. As discussed above, we believe that many issues could be addressed by improving the course and the protocol. This could also give rise to a follow-up study.

The sample size in this study is small. We stopped including trees after saturation occurred: new trees no longer gave reasons for new codes. Our aim was to give an inventory of difficulties in heuristic tree design. We do not claim the frequencies in Table 2 to be telling more than a rough indication of what are common and what are less common mistakes. The (relative) frequencies will in the end strongly depend on the prior training of the designers. However, there are no reasons why a different group of teachers would make different sorts of mistakes after a similar course.

Our findings also shed light on the way teachers support students generally, i.e., in situations where heuristic trees are not used. Compared to a classroom situation, the teachers have a lot of time to think and discuss about their choices in the way they shape the support for students. In that sense, their heuristic trees present a version of how they support students that should be at least as good as what they would do in classroom content wise—we do not refer to affect here.

Designing heuristic trees turns out to be a demanding and time-consuming enterprise. Nevertheless, we believe this worth the teacher's investment. Firstly, heuristic trees, once designed, can be used over and over again; and, if necessary, be improved. Secondly, the process of design allows teachers to develop their problem-solving guiding skills. Thirdly, joint design invites teachers to discuss their ideas on both the problem content and on their ideas on how to guide problem-solving. And finally, the main goal, a well-designed tree allows successful problem-solving in absence of the teacher.

References

Bos, R. (2017). Structuring hints and heuristics in intelligent tutoring systems. In G. Aldon & J. Trgalová (Eds.), *Proceedings of the 13th International Conference on Technology in Mathematics Teaching* (pp. 436–439). https://hal.archives-ouvertes.fr/hal-01632970

Bos, R., van den Bogaart, T. (2022). Heuristic Trees as a Digital Tool to Foster Compression and Decompression in Problem-Solving. Digit Exp Math Educ 8, 157-182. https://doi.org/10.1007/s40751-022-00101-6

Lemmink, R. (2019). *Improving Help-Seeking Behavior for Online Mathematical Problem-Solving Lessons* [Utrecht University]. https://dspace.library.uu.nl/handle/1874/382857

Lester, F. K. (2013). Thoughts about research on mathematical problem- solving instruction. *Mathematics Enthusiast*, *10*(1–2), 245–278. https://scholarworks.umt.edu/tme/vol10/iss1/12

Lester, F. K., & Cai, J. (2016). Can Mathematical Problem Solving Be Taught? Preliminary Answers from 30 Years of Research. In *Posing and Solving Mathematical Problems* (pp. 117–135). Springer International Publishing. https://doi.org/10.1007/978-3-319-28023-3_8

Pólya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press. https://doi.org/doi:10.2307/j.ctvc773pk

Robson, C. (2011). *Real World Research*. Wiley.

Tall, D. (2013). How Humans Learn to Think Mathematically. In *How Humans Learn to Think Mathematically: Exploring the Three Worlds of Mathematics*. Cambridge University Press. https://doi.org/10.1017/CBO9781139565202

Thurston, W. (1990). Mathematical Education. *Notices of the AMS*, *37*, 844–850.

Rogier BOS
Utrecht University, Freudenthal Institute, Utrecht, Netherlands
E-mail: r.d.bos@uu.nl
https://orcid.org/0000-0003-2017-9792

Theo van den BOGAART
HU University of Applied Sciences, Archimedes Institute, Utrecht, Netherlands
E-mail: theo.vandenbogaart@hu.nl