Problem Solving and Professional Skills

Dr Eric Hall • ehall001@dundee.ac.uk

2025-08-19

University of Dundee

Table of contents

Introduction

Welcome to PH11002 Problem Solving and Professional Skills at the University of Dundee.

These notes are available at dundeemath.github.io/PH11002/ and also as a PDF (visit the page and click on the PDF icon to download).

Licence

Part I

Problem Solving

1 Purpose

What are we here for?

"[T]he mathematician's main reason for existence is to solve problems [...] therefore, what mathematics *really* consists of is problems and solutions" (Halmos 1980).

Throughout your mathematics and scientific journey, you have likely faced a variety of problems that provided essential context to guide you toward the "appropriate" solution methods. Typically, you encounter a specific topic, technique, or method, followed by a series of related exercises that reinforce that concept. While this approach can be beneficial for solidifying your understanding, it can also be somewhat disconnected from real-world challenges. In reality, problems often integrate multiple concepts and span various areas of mathematics, requiring a more holistic application of your knowledge.

In this module, we want to develop skills that help you to solve problems more generally. This will be done by engaging with tasks for which the solution method is not known in advance. To get good at attacking such open-ended tasks, you will also learn to reflect on your experiences and problem solving processes. This is because problem-solving is not simply a product of your mathematical resources (i.e., *what you know*), it is also a function of your perceptions of that knowledge that you derive from your *experiences* with mathematics (Schoenfeld 1985).

In this module, our focus is on developing skills that enhance your ability to solve problems in a broader context. We will achieve this by working on tasks for which the solution methods are not known in advance. To excel in tackling these open-ended challenges, you will also engage in reflecting on your experiences and the processes you use for problem-solving. This is essential because effective problem-solving relies not only on your mathematical resources, i.e., *what you know*, but also on how you perceive that knowledge (Schoenfeld 1985). Your perception of mathematics is influenced by your experiences, and a key aspect of this module will involve engaging in problem-solving tasks, both individually and collaboratively, and reflecting on these experiences to enhance your learning.

The purpose of this problem solving module is to:

- build your confidence in solving unseen problems,
- provide prompts to support reflection that will deepen your perception of mathematical knowledge (and its interconnections),
- give generic support scaffolding problem solving (vs scaffolding the problem) that will provide a foundation for your development as a mathematician and scientist.

2 Problems

> "Problem. A doubtful or difficult question; a matter of inquiry, discussion, or thought; a question that exercises the mind." (*Oxford English Dictionary* 1989)

Problems often involve more complexity than straightforward exercises in that the method of solution is not proscribed. We will differentiate between two types of problems and present a general framework, introduced by Pólya in a series of monographs (Pólya 1945, 1954a, 1954b), for understanding problem solving. This framework is intended to serve as a foundation for analysing your approach to problem solving.

2.1 Two Types of Problems

We distinguish between different types of problems based on the desired goal.

Problems to find. The task is to produce or create an object that satisfies specified conditions, which may include a number, function, construction, example, counterexample, or algorithm. Common prompts for these tasks include terms such as determine, compute, construct, and classify. The success of these endeavors is evaluated based on criteria such as correctness, completeness, and in some cases, optimality or uniqueness.

> **Example of a problem to find**
>
> Find all integers n such that $n(n+1)$ is a perfect square.

Problems to prove. The task is to justify a claim beyond reasonable doubt. Typical prompts include: show, prove, disprove, establish, and deduce. To be successful, one must ensure validity, clarity, and appropriate use of definitions and prior results.

> **Example of a problem to prove**
>
> Prove there are infinitely many primes congruent to $3 \mod 4$.

> **Many problems mix both tasks!**
>
> Find all objects with property P and prove your list is complete.

In this module, we will focus on *problems to find*.

2.2 Problem solving framework

The problem solving framework (see Pólya 1945) is a four-phase cycle for tackling open-ended problems. The phases of the problem solving framework are depicted in Figure 2.1; the name of each phase is listed in bold text, with key terms in normal text. Knowing which phase of the framework you are in may help you choose the best prompt to move forward (see the table below in Section 2.3).

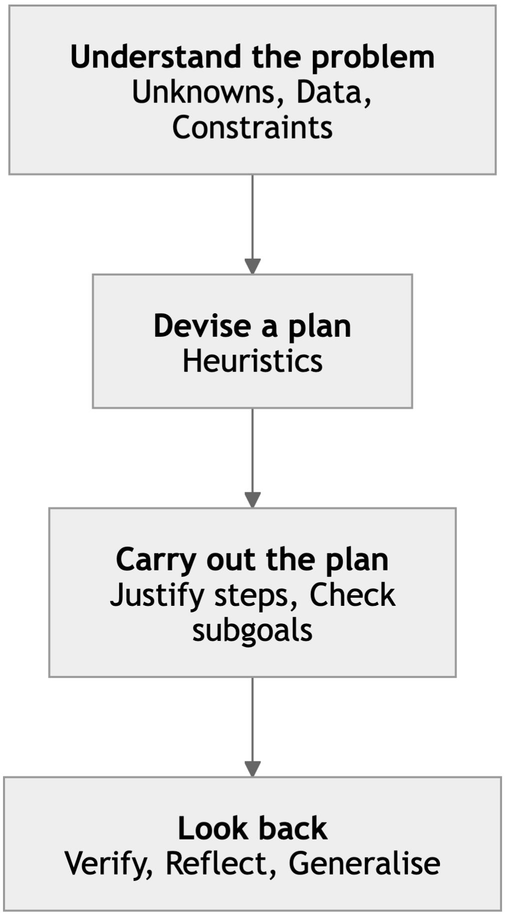

Figure 2.1: Four phases of problem solving.

The phases are roughly as follows. First, **understand the problem**: identify givens, unknowns, and conditions; restate it in your own words; sketch, tabulate, or probe small cases. Next, **devise a plan** by choosing a route. Possible routes are to work backward, look for patterns, simplify or specialise, use symmetry or invariants, introduce an auxiliary object, estimate or bound, change representation, or reduce to a known problem (we will investigate these problem solving *heuristics* later in Chapter 3). Then **carry out *your* plan**: execute cleanly, justify each step, check subgoals, and pivot if a step stalls. Finally, **look back**: verify the result, test edge cases, assess efficiency and clarity, and capture the key idea. The final phase is *essential*. Use the framework to reflect on what you learned so your future problem solving gets faster and more reliable.

2.3 Phases of problem solving

Use Table Table 2.1 as a working checklist and a reflection guide, not a rigid recipe. As you tackle an open-ended problem, you might find that you are “stuck”. First, don’t panic! Decide which phase of the problem solving framework you are in and scan the prompts in that row. Answering the prompt may lead to a concrete next action. Reflect on this new action for a few minutes; if it stalls, return to the table, pick a different prompt, or take a break. Over time, the prompts will become more familiar and the process of solving an open-ended problme will be less daunting.

Table 2.1: Phases of problem solving, adapted from (Pólya 1945, xvi–xvii).

Phase	Purpose	Prompts to ask yourself	Useful tactics
Understanding the problem	For a problem to find, understand the problem by making the unknown, data, and conditions precise.	What is unknown? What is given or what are the data? What conditions/constraints apply? Can I restate the task in my own words? What do small or extreme cases look like? What diagram/notation will help?	Define symbols. Draw a figure. List constraints. Test tiny cases. Identify edge cases. Rephrase the question. Separate various parts of the condition.
Devising a plan	Find the connection between the data and the unknown that provides a path towards a solution.	Have I seen the problem before? Have I seen the same problem in a slightly different form? Do I know a related problem? Do I know a theorem that might be useful? Can I work backward from the goal? Can I simplify/specialise first? What pattern, invariant, or symmetry might apply? Can I introduce an auxiliary element or change representation? Did I use all the data in devising my plan? Did I account for all the conditions?	Heuristics such as analogy; special/edge cases; generalisation; using invariants and symmetry; bounding/estimating; pigeonhole; substitution; set up equations or a new diagram.
Carry out your plan	Carry out plan, checking each step!	Would I be able to clearly explain that the step is correct? Can I prove that it is correct? Does each step follow from assumptions or known results? What subgoal can I verify now? If a step fails, which alternative route will I try next?	Justify steps. Prove lemmas. Compute carefully. Frequently take stock (checkpoint). Pivot quickly if a line of attack stalls. Don't panic.
Look back	Validate the result and reflect on learning.	Can I check the result? Can you explain your arguments? Does the result meet all conditions? Any counterexamples? Is it complete/optimal? Can I shorten or generalise the solution? Can I derive the result differently? What key idea made it work, and where else could it apply?	Record the central insight. Phases of the framework. Prompts answered. Heuristics used and how.

3 Heuristics

3.1 What is a hueristic?

A *heuristic* is a problem-solving device or strategy that provides a way to seeing or approaching a problem. Choosing a suitable heuristic often leads us closer to a solution. These strategies are versatile, applying across a wide range of domains and topics. However, heuristics alone are not enough to solve a problem; they must be combined with relevant knowledge and a refined ability to select and deploy mathematical resources effectively. By explicitly discussing and reflecting on problem-solving strategies, we aim to bring the use of heuristics into your conscious awareness. This focus will help you create connections between different areas of mathematical knowledge and enhance your reasoning skills. By honing these abilities, you will be equipped with the tools necessary to become a proficient and literate problem solver.

3.2 Dictionary of heuristics

Below we include a "dictionary" of common heuristics. Each of these heuristics should be viewed as a label for a closely related family of devices. That is, each heuristic in the dictionary is not precise enough to allow for unambiguous interpretation and subsequent application to a particular problem! Key challenges that arise when trying to apply any of these heuristics is firstly to select appropriately and second to decompose the heuristic into a targeted strategy that you can actually execute. Use the prompts to trigger action. For each heuristic we have indicated a source: (P) = after Pólya (Pólya 1945); (S) = after Schoenfeld (Schoenfeld 1985); (M) = modern/computational after (Michalewicz and Fogel 2004).

> ⚠ Heuristics will not replace shaky mastery of a subject!
>
> "Despite the fact that their application cuts across various mathematical domains, the successful implementation of heuristic strategies in any particular domain often depends heavily on the possession of specific subject matter knowledge." (Schoenfeld 1985)

> Variation of the problem (P)
>
> Using callouts is an effective way to highlight content that your reader give special consideration or attention. **Idea.** Decompose; special cases; generalise; analogy.
> **Prompts.** Subgoal to unlock? Small/extreme cases?
> **Targets.** - Decomposing & recombining - Specialisation / Generalisation - Analogy

Variation of the problem (P)

- Decomposing and recombining
- Establishing and using subgoals
- Generalisation
- Specialisation

“To better understand an unfamiliar problem, you may wish to exemplify the problem by considering various special cases. This may suggest the direction of, or perhaps the plausibility of, a solution.| (S)

- Analogy

3.2.1 Auxiliaries (P)

- Auxiliary elements
- Auxiliary problem

3.2.2 Notation (P)

3.2.3 Figures (P)

3.2.4 Examine your guess (P)

3.2.5 Working backwards (P)

3.2.6 Setting up equations [as translation] (P)

3.2.7 Check the result (P)

3.2.8 Reductio ad absurdum and indirect proof (P)

3.2.9 Type checking and dimensional analysis (P)

3.2.10 Approximation (modelling)

3.2.11 Estimation

3.2.12 Exhaustive search (M)

3.2.13 Greedy algorithms (M)

makes the best local choice at each step, hoping to find a global optimum solution ### Neural Networks (M)

4 Control

Attitudes to problem solving. - Selecting and pursuing the right approach and recovering from inapproriate choices

> **!** Important
>
> If you find yourself 'stuck' try answering these questions [Schoenfeld:1985ps]: - What exactly are you doing? Can you describe it precisely? - Why are you doing it? How does it fit into the solution? - How does it help you? What will you do with the outcome when you obtain it?

References

Halmos, P. R. 1980. "The Heart of Mathematics." *The American Mathematical Monthly* 87 (7): 519–24. https://doi.org/10.2307/2321415.

Michalewicz, Zbigniew, and David B. Fogel. 2004. *How to Solve It: Modern Heuristics*. 2nd ed. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-662-07807-5.

Oxford English Dictionary. 1989. 2nd ed. Oxford: Oxford University Press.

Pólya, George. 1945. *How to Solve It*. Princeton University Press.

———. 1954a. *Mathematics and Plausible Reasoning, Volume 1: Induction and Analogy in Mathematics*. Princeton, NJ: Princeton University Press.

———. 1954b. *Mathematics and Plausible Reasoning, Volume 2: Logic, Symbolic and Mathematical*. Princeton, NJ: Princeton University Press.

Schoenfeld, Alan H. 1985. *Mathematical Problem Solving*. Academic Press. https://doi.org/10.1016/C2013-0-05012-8.

Part II

Professional Skills