SOLVING PROBLEMS

HOW TO SOLVE A PROBLEM

Life is full of difficulties (problems) we must face to keep alive, or we want to face to ameliorate. Solving problems is finding satisfactory answers to challenging objectives bounded by imposed (or assumed) restrictions. Examples: Find food; answer: gather what is within reach, keep a food stowage, synthesise food? (maybe in a future). Find the circle quadrature; answer: not possible. Find a mechanism for transportation of people and goods; answer: pulling or pushing over land with some smoothing agent (leaves, stems, wheels, rails, levitation), floating on water (streams, rowing, sailing, blowing), moving across the air (floating in an aerostat, sailing in a plane, hovering in an helicopter), shooting through space (ballistic, rocket).

To solve a problem, at least in theory, a solver is needed (human or machine), i.e. a trained person or a programmed machine that takes the statement and yields the solution. Many times there is need of some interaction between the solver and the proposer (to ask details to clarify the problem) and/or the real world (to make experiments to clarify the problem).

How are problems solved? Three possibilities may be considered:

- By themselves, i.e., either by chance or necessity, there are problems that, after some time, they are no longer problems. Living beings are born with some pre-programmed problem-solving knowledge (e.g. how to reproduce). Waiting for the problem to solve by itself is seldom a good approach; it is said that there is no (individual) problem lasting more than a person's lifetime.
- By expertise; i.e., by knowing the answer from previous experience (i.e. a problem is solved by reducing it to a previous one, already solved). But there is such a variety of problems, that relaying on previous experience on that particular problem is seldom the best approach; expertise nonetheless, when sharing amongst a large team (e.g. large data bases on the internet, with friendly querying interfaces), may help a lot to not-reinventing the wheel every time.
- By method, i.e., by knowing tools and skills to simplify problems and find answers, based on previous education. This approach has been found to be the best. The heuristic approach of finding solutions by 'a clever trial' or 'happy idea', instead of following well-known algorithms, rarely works (only wise persons abound with clever ideas).

Most engineering problems are solved without proper knowledge (e.g. fire was mastered well before any rudimentary idea of combustion existed). But education, i.e. the teaching-learning of reliable predictions, is

what has allowed humankind to master powerful problem-solving skills other than the basic trial-and error procedure.

To help knowing how to solve problems, i.e. how to train persons to do it or how to program machines to automatically do it, the student needs different tools, i.e. more formal training on abstractions (mathematics) than computing skills, for instance. However, even during the preparatory stage of learning how to solve problems, it is important to make use of solvers that make life easier by solving already-known parts of the problem in order to not dispersing too much the attention. I believe that students must learn how to add 12 and 21 (and how to solve $x\hat{}x=2$, at university level), but they should also be exposed to practical use of calculators and powerful computer applications (realising that every real calculator and every real software package has a trade name behind, and its own peculiarities, many of which may take out of focus the objective of the problem).

But, neither for training nor for practice, there is such a tool as "the solver" that works for every problem; not even a "thermal solver". There is a range of solvers for partial tasks, with their own peculiarities. On one side, there stands the general-purpose mathematical solver (e.g. to solve $x\hat{}x=a$ for x) with numerical, graphical and symbolic capabilities. On the other side one may place the single-purpose packages (that may be a finite analysis package or a routine to find the physical phase of water at a given temperature and pressure. Using a general-purpose tool, i.e. a programming language, the user may solve any problem, but at a high cost in required skill and time, whereas the single-purpose tool (e.g. a partial differential equation solver) cannot solve but what has been programmed for.

Glossary

- Finding. Finding by selective look-and-see and trying (making decisions), not by chance (only learned people seems to face good chance).
- Satisfactory. Only a few ideal problems have a unique perfect solution; most real problems have uncertainties in the data and the goals, and one just looks for a satisfactory answer.
- Answer: Answer to a question; the first step in problem solving is knowing the problem (objectives and restrictions), that may be imposed or assumed.
- Challenging: If a problem is not challenging, then it is a mere exercise, e.g. a sequence of known explicit (forward) steps. The art of posing a challenging problem may be harder than solving it (some teachers say).
- Assumed: Some problems are imaginary (e.g. the circle quadrature), not immediately practical, but deserving attention anyway (for training and, maybe, for future practical use).
- Restrictions: If there are no restrictions, any answer is good and there is no problem. Typical problem constrains (restrictions) are: available time, money, labour, regulations and natural laws. Time management, economics, human resources and legislation must be learnt and accounted for, but they can change; natural laws cannot be changed (the physics of the problem is always imperative: the fact of the matter). Try to get rid of imaginary restrictions (false preconceptions).

Tasks in solving a problem

Four sequential stages may be identified, related to the purpose at hand, the method to give it a structure (modelling), the actual process to follow, and the fruits worth to gather.

1. Statement	=purpose 1. Identify the problem (proper channelling of the problem to solve is the best route to success). Most problems are imposed: carefully read and listen for instructions on what to do. Self-confidence is a must in approaching the unknown (if you believe you can, you will). 2. Quantify the problem. Find out given data (from the identified statement) and guess background data (from context). Guess if it is feasible and affordable to you. 3. Take on the problem (if it is relevant). Assume the problem is solvable, perhaps by relaxing some stringent condition (imposed or assume); keep an open mind, no constraint is infinitely rigid (all data have uncertainty). 4. Figure out the main subject at hand (mathematics, physics, mechanics, structural, electrical, thermal...) and forecast the expected solution (by deduction, induction, contradiction, recasting, or heuristics). Is it similar to another problem already known? If this quick-look analysis is not sufficient, proceed to a detailed formulation.
2. Formulation	=modelling 1. Model. A model is a mathematical application that provides an output (function) to an input (variables), usually based on some well-established laws (constraints). All real-world models are only approximate because of the intrinsic uncertainty in our perception. No model can ever be exact or true (or false): old models (old approximations) are substituted by new models (new approximations) when we want to incorporate some additional findings 2. Reduce. Make it easier; a model is just a tool: the simpler, the better. Solving a less difficult problem might be acceptable (many times restrictions can be relaxed), or even the only practical way-out (sometimes results are needed immediately, or not at all worth); in any case, a simpler problem teaches and encourages to more challenging tasks. Generic simplifying tactics are: consider only the most relevant effect, consider symmetries and constancies, consider bounding cases, find invariants, etc. 3. Name. Simple notation simplifies the development, debugging and documentation. Use standard symbols. Make an annotated sketch (the simpler the better); it is worth a thousand words (Confucius). Nature-behaviour is written in mathematical language (Newton). Substituting variables by numbers (constants) is no good: simple algebraic symbols are far easier to follow that strings of decimal digits.
3. Analysis	=processing 1. Is the problem another of a kind? If the modelling coincides with previous problems, apply the same tool. Societies have accumulated empirical knowledge (codes of best practice) worth following (sometimes compulsory). 2. Apply first physical principles you are sure about (e.g. the mass conservation

	principle, and not 'a volumetric conservation principle' of your own). 3. Apply standard mathematical tools trying to resolve (i.e. simplify) the model. 4. Pinpoint model characteristic behaviour (tendencies). 5. Check orders of magnitude. 6. Estimate uncertainties (significant figures).
4. Conclusion	=achievements 1. Document the paved way: what you understood was the problem, what you undertook to solve it, what happened to be right from your efforts (for profitable repetition), what happened to be wrong (if significant, to avoid repetition), and what could be further done (if more resources were allocated). 2. Pinpoint main results that give solution to the stated problem, and possible applications (where these results may be extrapolated), with expected uncertainties (your confidence). 3. Feedback on assumed context (e.g. the technical problem solved, irrespective of economic or environmental constraints), expected feasibility (difficulty) to carry it along, and guidelines to follow on (e.g. by adding or releasing some constraints).

Tools

The first steps in problem solving require vague tools to synthesise data and realise the size of the forest without being masked by the trees. This is an art called strategy (from the Greek 'commandment'), exercised by a director or general commander, and although general guidelines and specific techniques may be given, becoming an artist cannot be directly taught but by means of prolonged exposure to the works of great strategists: Pythagoras, Alexander the Great, Newton, Einstein. This implies a long-term learning known as 'background' or 'education'.

A second step in problem solving requires putting in order the items conceived in the strategies, and is more an art than a craft, called tactics (from the Greek 'dispose'), where the different actions thought are balanced in timeline and deepness for the best trade-off. This is a near-term activity heavily dependent on the foreseen and unexpected interactions amongst the parts.

A third step in problem solving requires putting in action the appropriate techniques (from the Greek 'skill'), known to solve specific problems: mechanics and materials for structural problems, thermodynamics and heat transfer for thermal problems, etc. Mastering a technique is more a craft than an art, and usually it only requires practice (like for driving a car).

The smallest step in problem solving requires performing a well-defined task, and only demands for an appropriate tool (an aid): a hard mass like a fist or a hammer, a sharp cutter like a tooth or a knife, a meter like a thermometer, a routine to invert matrices, to solve partial differential equations, to read values from a table or data base, etc.

In summary, from outside in (or top down), the approach to solve problems has the following layers:

-Strategies : long-term global plans
-Tactics : short-term local plans
-Techniques: multi-step general procedure
-Tools : single-step particular procedure.

REFERENCES

1. Zeitz, P., "The Art and Craft of Problem Solving", Wiley, 1999.

Back to Problems