Two Letters on Ways to Solve Problems
Letter 2

Ray Solomonoff

Oxbridge Research
Mailing Address: P.O.B. 400404, Cambridge, Ma. 02140, U.S.A.
prettyvivo@gmail.com http://raysolomonoff.com

General Problemsolving (Human) (For machines: see (16)).

1. If difficulty defining problem: Go to higher level: Why do I want to solve this problem? Maybe higher level problem is more easily defined. What is the problem? Define it clearly. If it is an optimization problem, *exactly* what is the GORC? If this can't be done see "study problems": (7).

2. If the problem can't be defined clearly, make *that* the new "top problem"... i.e. how to define it clearly.

3. What are *similar* or *analogous* problems that I can solve or understand better? Is the problem in a *class of solvable problems*, or can I transform it into such a class?

4. Can I break the problem into sub-problems (divide and conquer)? (See 14 - also the AND/OR Net is a generalization of this approach) Can I express problem as *Equivalent* problem?

5. Guess at a solution, then prove it works. (Several previous analyses of *why* they don't work can give good understanding of the problem.)

6. Try a numerical solution (— either approximate or *exact* for certain cases).

7. Devise "study problems": These are similar a/o *analogous* problems. Solving them teaches one concepts likely to be useful in solving the main problem.

8. Generalize the original problem. This may simplify it and make it easier to see what the important questions are.

9. Specialize the original problem; solving one or more special cases can give concepts useful for solving the original problem.

10. Does the problem have to be solved? Is the problem meaningless as stated (see 1, 17)? It may be avoidable: It may be the *wrong problem*! It may be part of the elements of a problem that can be broken down in a different way. If so, look at the *top* problem.

11. Is the problem solvable? (Either no solution exists or the computation cost is too large or may be infinite). One might spend some fraction of time showing problem is unsolvable.

12. Can the problem be configured as a *Hill Climbing* problem? If so, is the topology smooth enough to make simple localized "steepest assent" possible? Consider "non-greedy" methods.

13. Can the problem be configured as a *GPS* (General Problem Solver) problem (*vector* hill height). If so, try to find good difference functions and associated operators.

14. Study the problem. If it has parts, (see (4)) study the parts and try to understand them. If the problem is *very difficult*, *devise a new kind of mathematics* dealing with how parts of these kinds interact. (Also see (6): devise study problems).

15. If the problem is a well known, unsolved problem: Try to reformulate it in *novel ways*; ways that would not be thought of by most researchers. Or reformulate it into an area that *I* am very familiar with ... more so than most researchers. Some tricks:

.....Less Common.....	.....More Common.....
Discrete parameters $->$ continuous	Continuous parameters $->$ discrete

R wins $->$ x wins (inversion generalized terms). Use of functional equations, difference equations. Probability instead of exact values of parameters.

16. If you are a *machine*, the problem is *already* clearly defined. Try to define it *less clearly*. Make the concepts in the problem *vague*. One way is to *remove/weaken* *constraints/parameters* that define the problem. This is *one* way to *generalize* the problem. (Standard, easy way, to generalize.) It can suggest "related problems" with (perhaps) related solution techniques. Try other ways to generalize the problem so (8, 3) can be done. Try to restore/retrieve the contextual information that a human would need for the problem.

17. (related to 1, 10) At meta level: Do I unconsciously *not want* to solve the problem? If so, why? Is it legitimate? Maybe I should, indeed, not work on it!

18. What is the *best* solution? The best method of solving the problem? [What is the *criterion* for *goodness of solution to the problem?] This

last is a *major heuristic* for many areas of inquiry. For computer cost = infinity, for time = infinity, for maximum /= infinity.