Towards a problem-solving methodology for coping with increasing complexity: an engineering approach

J. T. BOARDMAN* AND G. C. A. ROBSON
Department of Electrical and Electronic Engineering, Brighton Polytechnic, Moulsecoomb, Brighton, BN2 4GJ

A methodology is sought that provides some formalism to the phenomenon of increasing problem complexity. The paper describes an engineering approach to the formulation of problem-solving strategies for a sequence of increasingly complicated variants of the well-known Towers of Hanoi problem. This approach defines the preferred nature of tools required for the discovery of fresh and promising insights into handling increasing complexity. As a consequence of the research which the paper describes, an optimal recursive algorithm for solving the Towers of Hanoi problem with invariant initial conditions has been revealed, and this is presented. Conclusions are made as to the relevance of a possible principle of evolutionary elegance for generalised problem-solving strategies.

Received November 1984

1. PROBLEM STATEMENT

In the management of complex engineering systems and projects the unexpected must always be expected. Complications are the norm rather than the exception, and the richness of unwritten heuristics among the body of practising engineers bears testimony to the manifold subtleties and intricacies of engineering problems. It is therefore not surprising that engineers will constantly be asking the question: 'What if...?'

A common requirement that the engineer faces is that of increasing the efficiency of plant and equipment. Thus, when he is not faced with designing the next generation of a system, he is often confronted with the task of significantly improving its existing performance. In any event he must use his design skills to solve a problem, similar to that with which he already has a high degree of familiarity but which possesses nevertheless a degree of complexity substantially beyond that which he has formerly known.

In such a situation, the spreadsheet is merely a tool with a value of its own, but wholly inadequate in and of itself. The engineer is well aware of how essential it is to be provided with a well-endowed toolbox, not merely literally but also figuratively speaking. What would be of great assistance then is a formalised methodology that would guide the problem-solving process in a direction which exploited the particular phenomenon of increasing problem complexity.

This paper describes an engineering approach to the formulation of a problem-solving methodolgy which ought to be capable of exploiting the increasing complexity in any given situation. To illustrate this approach a succession of variants of the well-known Towers of Hanoi problem is studied.

Now, it is not of particular importance to most people to learn how anyone first discovers the Towers of Hanoi problem. In a sense this is a pity because no one can possibly realise at the time the extent to which the approach taken to wresting a solution is influenced by the details of that initial encounter. For the engineer, however, who utilises this traditional skills of combining practical experience with imaginative modelling techniques, it is a matter of considerable importance that he pursues all possible means of uncovering the relationships that exist between model building and its impact on practical problem-solving. On the basis of that argument this elementary problem affords a suitable opening for an engineering approach. The fact that there are ramifications to this problem in the generation of Gray-codes, production scheduling and material handling, all of which fall within the engineering purview, is merely an added bonus.

For some their introduction to the Hanoi problem is by way of a simple wooden toy. This modern age has produced what is perhaps the equivalent to this: the interactive colour graphics game. For many that first encounter was a verbal description which, unless it became visualised with alacrity, did not gain in meaning and hence produced little further interest. Let us attempt now to state this prosaically with reference to Fig. 1. 'A number of discs of differing diameter may be placed upon any of three pegs in any way so long as no disc rests upon another of smaller diameter. This condition is strictly observed throughout. The standard arrangement, initially, is to place all the discs on a given peg – the source peg. Whatever the initial conditions, the problem is always to collect all the discs on to another peg, nominated as the destination peg, by observing the simple rule that only the top disc (or the only one) on any peg may be moved to another peg at any one time.' At first sight this problem may to some appear impossible. Then one has to inquire 'At what point does it become possible, for it is evidently trivial for one disc?'

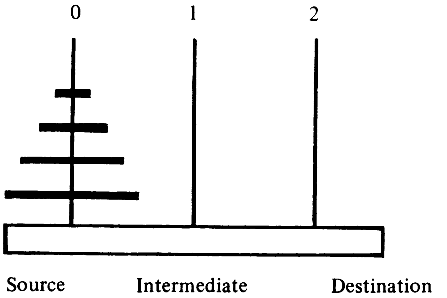


Figure 1. Classical Hanoi model

Moreover, a solution for two discs is deduced with little difficulty. Certainly, if the problem does not become impossible as the number of discs increase it does look to become increasingly more complex. Interestingly, the truth is that the solution never changes in complexity, a point which is heavily underscored by stating that solution recursively.

Recent interest in this problem has been keen, as

* To whom correspondence should be addressed.

evidenced by the volume of literature. One is particularly impressed by the geographical spread among authors.[1–10] Certainly with interest rising in Artificial Intelligence through the intitiatives in the UK and elsewhere in Information Technology, and with the emergence of Expert Systems, opportunities to get into this field, at least through a cursory examination of simple games, quite obviously will abound. But the questions are asked: 'Can it be that this particular problem does indeed have something peculiar to offer as to the way in which we can derive elegant solutions through careful attention to the representation of the problem as a model?', and, 'Does the problem allow us to discover principles for the development of problem-solving strategies by the successive generation of algorithms, similar in form, to cope with increasing complexity in the problem's conditions?'

2. STATE-SPACE MODEL

The following question arises: 'If it is not the complexity of the solution that changes as the number of discs increase, what is it that does change about the problem?' The answer is – the state-space. This represents the number of possibilities that may exist, and the state-space diagram may be portrayed in such a way as to reveal the complexity of the problem itself (Fig. 2).

It should come as no surprise to discover that these

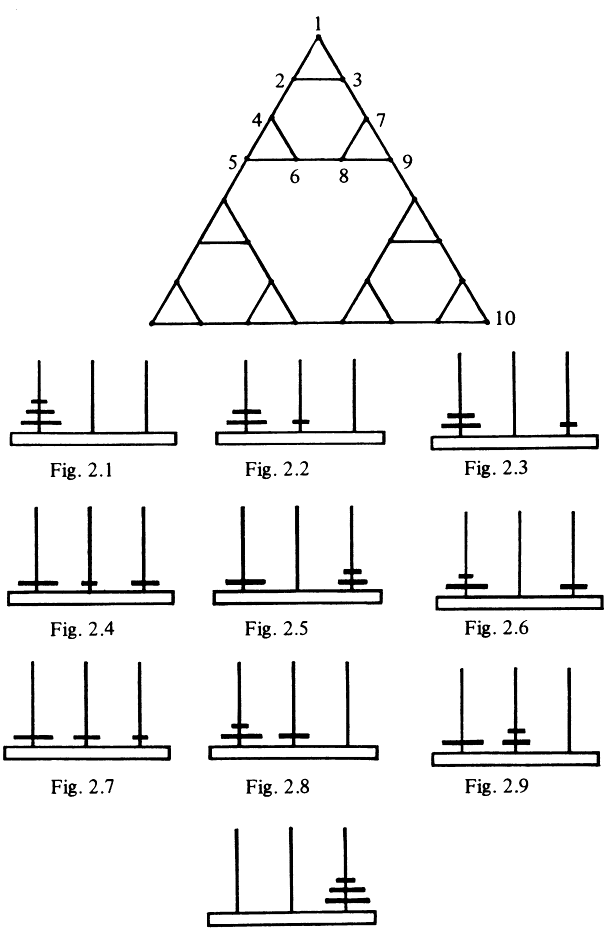


Figure 2. State-space diagram

figures were drawn by means of a recursive algorithm. This problem simply abounds in recursive riches. Suppose the number of states in a problem with N discs is x_N. For $N = 1$ we know that $X_1 = 3$. Then for $N+1$ discs the states for the N-disc situations combine with any of three states for the new (and largest) disc.

Therefore $x_{N+1} = 3x_N$; it follows therefore that $x_N = d^N$, by induction. This is exactly the number of vertices in the state-space diagram for N discs. But notice now how difficult it becomes to find a way from any given point in this space to a specific point, for example extreme bottom-right vertex. Knowing that $2^N - 1$ is the optimal number of moves, the η of the optimal algorithm is $2^N - 1$ or approximately $(2/3)^N$ for large N. How do we go about finding an algorithm that will achieve this level of efficiency for any and every possible set of initial conditions? The answer is – recursively!

The basic recursive algorithm, which we will call HANOI, is extremely well known, but it is most expedient to restate it here for it becomes the parent to successive generations as we add further complications to the problem. The solution is stated as follows:

procedure *HANOI* (*Disc number*, *Source peg*, *Destination peg*)
  if *Disc is present on some peg*
    HANOI (*Next smaller disc*, *Source peg*, *Intermediate peg*)
    Transfer disc from Source to Destination
    HANOI (*Next smaller disc*, *Intermediate peg*, *destination peg*)
  endif
end procedure

In simple English the transference to N discs neatly piled on the source peg is achieved by:

(*a*) the transference of all but the largest disc to an intermediate peg (the same problem reduced in scale by one disc);

(*b*) the removal of the largest disc to the destination;

(*c*) the transference of the pile on the intermediate peg to the destination.

The effect of this algorithm is 'seen' in the state-space by a direct traversal of a wall of the triangle from one vertex to the other, each representing the complete sequential pile of discs on the source and destination pegs respectively. The simplicity of the recursive HANOI algorithm relies on the regular nature of the initial state of the problem: all discs are piled sequentially in decreasing size on one 'source' peg. The algorithm can proceed to solve the problem optimally using only three items of data, which are manipulated in combinatorial operations to generate the successive moves. Due to the lack of interaction with a data structure representing the state of the discs on the pegs the algorithm is unable to cope with an initial condition anywhere in the state-space other than a major vertex. The question arises: 'What algorithm would achieve the minimum number of moves if the initial conditions related to any point in the state-space diagram other than a vertex?'

3. DISTRIBUTED RECURSIVE ALGORITHM-OPTIM

The problem now is to devise an algorithm that employs a minimum number of moves in moving from an initial set of conditions where discs are distributed randomly but in a legal fashion, i.e. with no disc resting on a smaller one, to a specific goal, e.g. a major vertex in the state-space. It was finally arrived at by reason of two tenets which become important to the methodology. First, the firm belief that a recursive algorithm of similar type but with more elaboration actually existed, and could be arrived at by a trial-and-error process keeping the general form of the basic algorithm in view. Secondly, by experimenting with various attempts and watching their effects using an interactive colour graphics simulation.

In order to solve the problem, the algorith OPTIM relies on a suitable data structure which models the state of the discs at each move; no assumption is made about the initial positions of the discs – in fact, some of the discs may be missing.

It was not known if it would be possible to obtain a recursive solution to this problem: familiarity with the HANOI algorithm seemed to limit any development. The OPTIM algorithm was suggested by observing the process of obtaining the solutions by hand using the interactive graphics package. It was seen that in generating the moves the same basic steps as in HANOI were required. An exception existed when a disc was already situated on the required destination and therefore steps (*a*) and (*b*) given in Section 2 were not required. The algorithm was thus noted to be a refinement of HANOI:

procedure *OPTIM* (*Disc, Destination peg*)
  if *Disc is present on some peg*
    if *Disc is not on Destination peg*
      OPTIM (*next smaller disc, Intermediate peg*)
      Transfer Disc to Destination peg
    endif
    OPTIM (*Next smaller disc, Destination peg*)
  endif
end procedure

The marked similarity with the HANOI formulation should be noted. This algorithm is very similar to that presented by Scarioni and Speranza (Ref. 1) although arrived at quite independently. In fact their algorithm is insufficient inasmuch as it transfers all the discs to that peg which currently holds the largest disc. This constraint implies one of two deficiencies. Either their algorithm will only solve problems which are more than half-way to the solution or, alternatively, the user cannot dictate which is to be the destination peg. Being more than halfway to the solution would eliminate two thirds of the state-space diagram directly. It would put the initial conditions immediately into one of the largest subtriangles, given that the goal was at a vertex of one such. In addition, their algorithm is more complicated than necessary: having transferred the largest disc not yet on their 'destination' to that peg, after piling the smaller discs on the intermediate peg, they invoke the HANOI algorithm to finish the task, where a recursive call to their own procedure would have been sufficient. This correction to their algorithm has been effected, thereby making it as effective as OPTIM, though not as efficient because of superfluous coding.

There is also similarity between OPTIM and a means–end algorithm, which Simon[11] defines as follows.

(1) If the problem is to move N discs from peg X to peg Z, find the largest disc on peg Z, and set up the goal of moving it;

(2) if this disc can be moved to its target peg, move it, delete the goal, and go to (1);

(3) otherwise, delete the goal of moving it and find the largest of the discs that either lie above it or lie on its target peg;

(4) set up the goal of moving that disc to the remaining peg, and go to (2).

4. INITIAL-CONDITIONS INVARIANCE RECURSIVE ALGORITHM – RECOVER

The question now arises, as it will fairly naturally in the mind of an engineer, 'What if the initial distribution of discs is illegal?' In other words what would be the appearance of an algorithm, assuming that such were to exist, if it did not care how it first found the disc situation, some possibly being illegally placed, but nevertheless always obeyed the rule of never placing a disc on top of a smaller one? The belief in the existence of a slightly more elaborate recursive algorithm, coupled with the extensive use of experimentation with an interactive simulation tool, again provided the answer.

The additional level of complexity being introduced into the problem now allows an initial condition where discs had been placed 'illegally', i.e. directly above a smaller disc on the same peg; the solution must still be achieved using legal moves. Any of the pegs may contain one or more illegal discs.

Observation of the close relationship between OPTIM and HANOI prompted the belief that a recursive solution to this problem could be developed using OPTIM as a framework.

It was noted, when using the package to solve the problem by hand, that an 'illegal' disc prevented smaller discs below it on a peg from performing steps (*a*) and (*b*) of the basic HANOI operations (Section 4) until the illegal disc was moved to a legal position. Discs which were not affected in this way could be moved using all three steps.

A special case existed where an illegally placed disc, d_I, was situated on the overall destination peg, the disc being larger than all other discs on the overall source or intermediate pegs, and larger than any other illegally placed discs on the same peg.

In this case the steps are:

(*d*) all discs smaller than d_I must be piled on either the overall source or intermediate peg (alternative peg A);

(*e*) the disc, d_I, is transferred from destination to the remaining peg (alternative peg B);

(*f*) discs smaller than d_I on the destination must be merged with pile formed in step (*d*);

(*g*) the disc, d_I, is transferred to the destination peg. The process then continues with step (*c*).

The following recursive algorithm therefore obtains.

procedure *RECOVER* (*Disc, Destination peg*)
  if *Disc is present on some peg*
    if *Disc not on Destination peg*
      if *No larger disc anywhere above Disc on this peg*
        RECOVER (*Next smaller disc, Intermediate peg*)
        Transfer Disc to Destination peg
      endif
    else
      if (*Disc is largest illegal disc on peg*)

```
        and (Destination peg is Overall Destination peg)
      RECOVER (Next smaller disc, Alternative peg A)
      Transfer Disc to Alternative peg B
      RECOVER (Next smaller disc, Alternative peg A)
      Transfer Disc to Destination peg
      endif
    endif
    RECOVER (Next smaller disc, Destination peg)
  endif
end procedure
```

5. INTERACTIVE COMPUTER SIMULATION

The successive development of OPTIM and RECOVER from the basic HANOI algorithm was greatly facilitated by a computer package that was designed specifically to experiment with various approaches to the Towers of Hanoi problem. In retrospect, neither the inspiration for algorithm design nor the necessary conditions for tolerating the perspiration content of this investigation would have been possible by any other means. From the outset the package was designed to have a wide range of 'user friendly' features including colour graphics, user intervention with algorithms at critical points in their progress through solution, and the facility to include a wide range in solution methods for comparative purposes. The package was written in FORTH for a Microkey 4500 Personal Colour Computer and was given the acronym HARPO (Hanoi Algorithm Recursively Programmed for Optimality). The microcomputer has dual micro floppy disc (512 kbytes), 64K RAM, 32K ROM and a wide range of suitable interfaces for printers, and interactive devices, e.g. mouse.

The overall design of HARPO consists of four subsystems covering data structure design, user interactions, the algorithm suite, and visual display of algorithm performances. It is a comprehensive system with facilities for a complete examination of the Hanoi problem with added complications and pertinent applications. Some of these go beyond the scope of this paper, but it is hoped that they may be reported in future publications.

Each of these subsystems is now briefly described.

(*a*) Data structure design

The state of the discs is represented by an $(N+1)$ by 3 matrix, where N is the total number of discs present. One element of each peg's data contains the number of discs on that peg. The remaining elements for each peg contain the appropriate disc sizes in their respective positions on that peg.

Various functions have been defined to return values such as the peg on which disc n is situated, the disc size which is next smallest from disc n and the top disc size on peg p.

(*b*) User interaction

In its simplest sense this means the following:

(i) The selection of an appropriate problem in terms of the number of discs, their initial distribution and the specification of an overall destination;

(ii) the selection of a particular algorithm and the choice of an appropriate means of display;

(iii) the choice as to whether to save the result of studies and/or retrieve them, which may be done at will for further detailed examination; the cross referencing of these results is made simple and checks makes possible the exploration of ideas generated on account of the interlinking of patterns thus observed.

However, there is a more elaborate means of user intervention afforded by the very nature of FORTH. This includes dynamic data setting and testing, and vectored addressing for the selection and dissemination of test algorithms. The computing environment is thereby made extremely user friendly for the purpose of uncovering algorithmic structures.

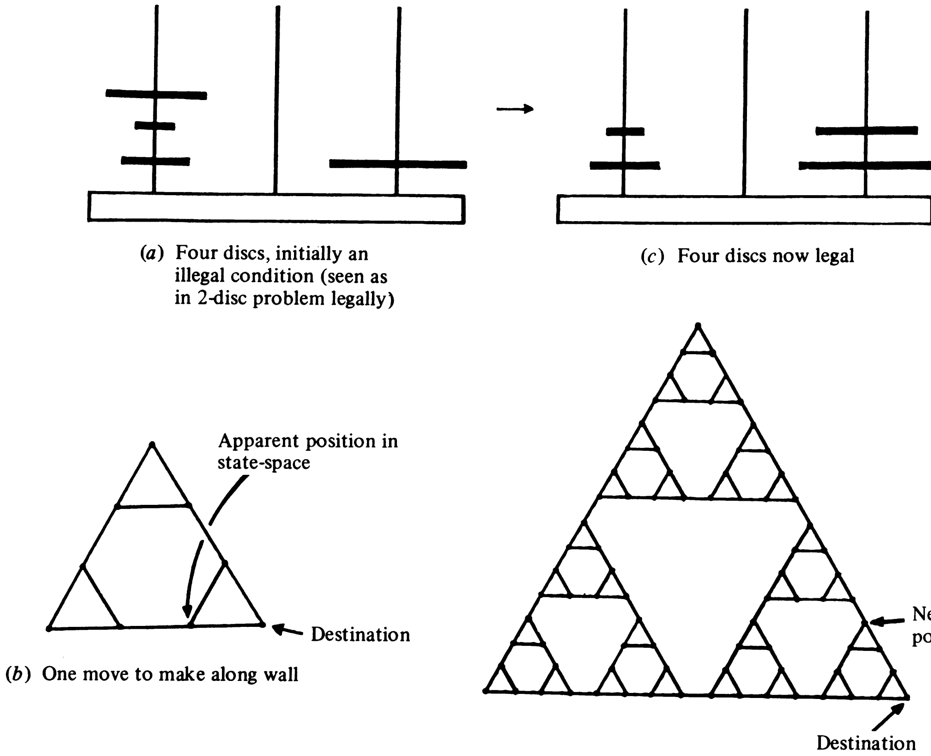
(*c*) Algorithm suite

Any number of algorithms may be included in the package up to the limit of the disc store. The three described in the paper are now available as standard, having been proved over a long period of exhaustive testing. Several often described in the literature such as in reference 1 are fairly easily added. There is also the facility to solve the problem manually, which formerly has proved an interesting excursion for many willing school children, who are usually keen to volunteer their efforts whenever the system is on public display. This incidentally often proves a rich source of heuristics. Additionally, a random-move generator has been included to permit observation as to how the computer behaves quite aimlessly, knowing nothing other than the legality of a move and the point at which it may stop. This was done not simply for amusement but as a basis for the development of an intelligent rule-based solution. This has been achieved and will hopefully be fully reported in a future paper.

(*d*) Visual display

It was stated at the beginning of this paper that a very significant relationship exists between the way the Hanoi problem is viewed and the form of solution developed therefrom. That this is so is an essential tenet for the authors' work. For those who meet the problem via a wooden toy it is extremely difficult to conceptualise the problem and thereby to hypothesise some structure which will solve it. The first attempt at visual display was to animate the movement of discs around the pegs viewed from a side elevation. This perspective in one sense achieves a move forward purely due to the animated effects. But this leaves much to be desired. For example, observation of clockwise and anticlockwise motion of alternating discs is not readily apparent. Knowing that this is a pattern leads one to consider an aerial perspective of the problem, and this has been implemented. Tests have shown that this plan view does enable first timers to the problem to discover the cyclic pattern which the basic Hanoi problem exhibits.

Movement through the state-space diagram as directed by an algorithm will also display its efficiency. Up to the point of reaching that design which gives optimal results, any deficiencies in an algorithm are more easily spotted



(a) Four discs, initially an illegal condition (seen as in 2-disc problem legally)

(c) Four discs now legal

(b) One move to make along wall

(d) Three moves to make along wall

Figure 3. Picturing illegalities

by visual inspection of its erratic behaviour in attempting to proceed through the state-space to a final destination.

Complications with this mode of display arise when one is faced with the question: 'What does the state-space diagram look like for illegal conditions?' The clue to answering this is given by the nature of the RECOVER algorithm. At first sight it would appear that something much more complicated than the triangle diagram is required – possibly a multi-dimensional object. The very thought of this is usually quite abhorrent to an engineer. A simpler model is suggested by retaining the 2-dimensional triangle diagram and reinforcing the aerial perspective. Therefore, in the illegal case, larger discs obscure any smaller ones which may lie beneath them. In some cases a sandwich effect is possible, though this is revealed in elevation but not plan view. Obscured discs when seen from above are not revealed until a move is made to resolve that particular illegality. At this point there is a new 2-dimensional triangle diagram defined in terms of the new number of discs increased from the former situation by the number of discs revealed in that particular move. This is a simpler model to deal with, to visualise and to simulate. Altogether it provides a much stronger and more versatile tool to aid in the discovery of the algorithm being defined. An example of how this works is depicted in Fig. 3.

6. CONCLUSIONS

A reasonable question to ask at this stage is 'Has the particular line of experimentation with the Hanoi problem produced a formalism that can be offered as a basic methodology to cope with increasing complexity?' The dangers of elucidating principles which must of necessity be general, from a very specific and possibly unique context must of course be avoided.[12] The authors believe that they have succeeded here for the following reasons. First there is clear evidence that there has been increasing complexity in the sequence of Hanoi variants, which can easily be continued. Although at each stage the next variant seems always to prove intractable, a solution can always be found by maintaining belief in its existence and trusting that its character is essentially that of its predecessor. It is probably true that incremental complexity leads to exponential effort in problem-solving capability, although one is always deflated to discover in retrospect the inherent elegance of a solution and its characteristic conformity with predecessors. The exponential effort is greatly expedited by developing suitable tools to give better insight into the essence of the problem. For the engineer this is confirmation of the indispensable value of modelling and simulation tools in problem analysis. To paraphrase Gerald Jay Sussman 'If you don't like the tools that are available, then build your own before you start work'.[13] The authors wish to emphasise that they could not have carried out their investigations had they not designed HARPO (Section 5) to suit the purposes which they foresaw at the very beginning.

Part of the engineering approach then has been belief in the existence of extended solutions to enhanced problems with a thematic characteristic. Another facet has been this emphasis on tool building. A third aspect has been the way to view the problem. The state-space model is very revealing in many respects as to the nature of the Hanoi problem, but this must also be true for any problem for which a coherent definition is relevant. As the authors look to future speculations of complexity the virtue of the state-space model becomes more apparent.

Suppose we consider more pegs. First we note that for N discs and p pegs then if $p > N$ the minimum number of moves is always $2N$-1 but otherwise there could be a multiplicity of optimal solutions and the question is: 'How might these be found?' Further, 'What does the state-space model look like?', and, 'How can its form be exploited to yield the problem-solving strategy?' In so far as this is taking the form of a recursive algorithm, the authors wish to put forward the following postulate.

The basic character of a recursive algorithm which optimally solves a generic problem is preserved through future generations of algorithms which will need to be designed in order to cater for successive problems created as a result of introducing increasing degrees of complexity.

It should prove interesting to see how the postulate would be of support in resolving such problematic complications as: (i) the appearance/disappearance of disc/pegs in the course of a solution; (ii) concurrent operation (i.e. multiple simultaneous disc transfers); (iii) dynamic prioritising of disc transfers.

These may all seem fanciful at this stage, but perhaps they would seem less so if we could solve them either by recursive algorithms or by intelligent rules (heuristically) especially if those strategies emerged from a methodology. Were this proved to be the case, we should be much wiser about the nature of this problem, its potential as an analogue and about the way in which problem-solvers at large need to think about, look at and persevere with problems.

Finally, engineers are always interested in analogies where the solutions to more abstract situations so often find applications to what many regard as the real world. It has already been noted that solutions to this Hanoi problem bear relevance to Gray codes and binary systems which are fundamental to the development of digital systems. If nothing else the authors have produced a powerful sorting technique embodied in RECOVER where constraints on sorting and merging various entries might be particularly severe. What is of much greater satisfaction is the knowledge that this whole line of research is directed towards asking such questions as: 'What defines a problem and more particularly its complexity?'; 'What is the best form of conceptualising a problem in order to expedite the solution to it?'; 'What principles exist for deriving more elaborate solutions to more complex problems of the same type from primitive beginnings?' By the investigation of these questions we proceed towards a methodology.

REFERENCES

1. F. Scarioni and H. G. Speranza, A probabilistic analysis of an error-correcting algorithm for the Towers of Hanoi puzzle. *Information Processing Letters* **18**, 99–103 (1984).
2. P. J. Hayes, A note on the Towers of Hanoi problem. *The Computer Journal* **20** (3), 282–285 (1977).
3. P. Buneman and L. Levy, The Towers of Hanoi problem. *Information Processing Letters* **10** (4, 5), 243–244 (1980).
4. M. D. Atkinson, The cyclic Towers of Hanoi. *Information Processing Letters* **13** (3) 118, 119 (1981).
5. M. C. Er, A representation approach to the Tower of Hanoi problem. *The Computer Journal* **25** (4), 442–447 (1982).
6. T. R. Walsh, The Towers of Hanoi revisited: moving the rings by counting the moves. *Information Processing Letters* **15** (2), 64–67 (1982).
7. T. R. Walsh, Iteration strikes back – at the cyclic Towers of Hanoi. *Information Processing Letters* **16**, 91–93 (1983).
8. M. C. Er, The colour towers of Hanoi: a generalization. *The Computer Journal* **27** (1), 80–82 (1984).
9. M. C. Er, The cyclic towers of Hanoi: a representation approach. *The Computer Journal*. **27** (2), 171–175 (1984).
10. M. C. Er, The generalized colour towers of Hanoi: an iterative algorithm. *The Computer Journal* **27** (3), 278–282 (1984).
11. H. A. Simon, Artificial Intelligence systems that understand. *IJCAI, 1977*, invited paper no. 3, pp. 1059–1073.
12. I. Bratko, Streamlining problem-solving processes. In *Introductory Readings in Expert Systems*, edited D. Michie, pp. 177–191. London: Gordon and Breach (1982).
13. T. Durham, Over the horizon. *Computing: the magazine*, p. 27 (29 Nov. 1984).