Published in J. ALGORITHMS 4, 397-411 (1983)

The NP-Completeness Column: An Ongoing Guide

DAVID S. JOHNSON

AT&T Bell Laboratories, Murray Hill, New Jersey 07974

This is the ninth edition of a quarterly column which provides continuing coverage of new developments in the theory of NP-completeness. The presentation is modeled on that used by M. R. Garey and myself in our book ''Computers and Intractability: A Guide to the Theory of NP-Completeness,'' W. H. Freeman & Co., New York, 1979 (hereinafter referred to as ''[G&J]''; previous columns will be referred to by their dates). A background equivalent to that provided by [G&J] is assumed, and, when appropriate, cross-references will be given to that book and the list of problems (NP-complete and harder) presented there. Readers who have results they would like mentioned (NP-hardness, PSPACE-hardness, polynomial-time-solvability, etc.), or open problems they would like publicized, should send them to David S. Johnson, Room 2C-355, AT&T Bell Laboratories, Murray Hill, NJ 07974, including details, or at least sketches, of any new proofs (full papers are preferred). If the results are unpublished, please state explicitly that you would like them to be mentioned in the column. Comments and corrections are also welcome. For more details on the nature of the column and the form of desired submissions, see the December 1981 issue of this Journal.

1. INTRODUCTION

This month, in celebration of The NP-Completeness Column's second anniversary, we take time off for some fun and games. To be more precise, this column will summarize recent complexity results about games and puzzles, one of the twelve categories of problems covered in [G&J].

In a sense, the study of games is a natural continuation of our recent focus on parallel and concurrent computing. One of the standard models for parallel computation is the *alternating Turing machine (ATM)* [6,7], whose definition can be stated in terms of a (two-person) game. An ATM is like our familiar *nondeterministic* Turing machine (NDTM) in that its computations can be viewed as trees rather than as paths. The two machines differ only in their definitions of what it means for a computation to yield a ''yes'' answer. In an NDTM, the answer is ''yes'' if and only if there is a solution to the ''puzzle'' (or *one*-person game) whose goal is to find a path in the tree from its root (the initial machine configuration) to a leaf representing a final ''yes'' state. In an ATM, the answer is ''yes'' if and only if Player 1 has a forced win in the following two-person game: *Positions* are the vertices of the computation tree, with the initial position being the root of the tree (the TM's initial configuration). Players alternate choosing the next position from among the immediate successors (in the computation tree) of the current one. The final positions are the final states, and such a position is a win for Player 1 if and only if it is an accepting (''yes'') state. (For a more detailed and formal definition of an ATM, which makes explicit the ''alternation of quantifiers'' implicit in the above, see [6,7].)

A fundamental result about alternating Turing machines is that the set of decision problems solvable by ATM's in polynomial time (i.e., using polynomial depth computation trees) is precisely PSPACE, the set of problems solvable by *deterministic* Turing machines in polynomial space [6,7]. This is another instance of the ''parallel computation thesis'' [7] mentioned in Column 6 [June 1983], and suggests why so many complexity results about games involve PSPACE-completeness, rather than NP-completeness. In addition to PSPACE- and NP-completeness results, this column contains polynomial time solvability results, as well as more than one exponential time lower bound. Such provable intractability results also are artifacts of alternation. The decision problems solvable in polynomial *space* on an ATM are precisely those that can be solved deterministically in time $O(2^{n^c})$ for some c [6,7], i.e., those in ''EXPTIME.'' Roughly speaking, a game can be solved in ''alternating'' polynomial space (and hence EXPTIME) if its ''positions'' can be described using polynomial space (and its legal moves can be recognized in polynomial time); if in addition there is a polynomial bound on the number of moves in any ''play'' of the game, the game can be solved in alternating polynomial time (and hence PSPACE).

We begin in Section 2 with puzzles, and although alternation is not normally an issue here, we shall see that it can appear in disguised form. Section 3 then turns to two person games, where alternation is rampant. In the concluding Section 4, we move beyond simple alternation to consider the potential for additional complexity in games where there is incomplete information (e.g., ignorance of the cards in your opponent's hand), a random element (e.g., a roll of the dice), or more than two players. Many of the results I will present were motivated by little more than the entertainment value (and mathematical challenge) involved in showing that common puzzles and games are difficult (or would be if suitably generalized). The reader should be forewarned, however, that games also have their serious side, both as models for practical problems and as tools for gaining a better understanding of the nature of computation, and this aspect of their character will not go totally unrepresented.

2. PUZZLING DEVELOPMENTS

Given the many books on the subject of recreational mathematics (e.g., see [2,3,4,12,18,19,20]) and the wide variety of puzzles they describe, researchers have shown admirable restraint in applying the techniques of computational complexity to this area. Although such popular variable-sized recreations as the jigsaw puzzle [GP13] and the crossword puzzle [GP14] were early victims, relatively few *fixed-size* puzzles have been forced to endure the rigors of generalization that are required if a finite problem is to be turned into one for which asymptotic analysis is appropriate. The only such problem added to the casualty list in [G&J] was Instant Insanity® (a Parker Brothers puzzle) [GP15].

Perhaps one reason why so few NP-hardness results have been proved is that many of the popular puzzles are not all that difficult. For instance, Driscoll and Furst [15] have shown that many puzzles based on the unraveling of permutations, for example the recent ‘‘Alexander’s Star’’ and ‘‘Hungarian Rings’’ puzzles, have group-theoretic structures that are easy to deal with, even when generalized. In particular, they show that if a group is generated by cycles of a fixed order, or if the group is primitive and contains a 3-cycle that is a polynomial-length product of generators, then the group has polynomially bounded diameter, and there are polynomial time algorithms for expressing arbitrary group elements as polynomial length products of generators. However, their results do not yet extend to Rubik’s Cube®, whose group generators (the slice-wise rotations of the cube) are *products* of cycles involving the faces of the subcubes [17]. Thus, although the $3 \times 3 \times 3$, $4 \times 4 \times 4$, and $5 \times 5 \times 5$ versions are all well-understood by the cognoscenti, it is not yet known whether the techniques generalize to efficient algorithms. Fortunately, there are still engineering difficulties to be resolved before the $n \times n \times n$ version starts rolling off the Ideal Toy Corporation assembly lines.

Turning from puzzles of the 80’s to puzzles of the 70’s (1870’s, that is − see [18])*, we come to the famous ‘‘15’’ puzzle of Sam Loyd [4,18], and a generalization that does lead to NP-hardness.

[1] SLIDING BLOCK REARRANGEABILITY

INSTANCE: Polygonal region R, collection $C = \{b_1, \ldots, b_n\}$ of rectangular *blocks*, each block b_i having an associated integer height h_i and width w_i, along with an initial position I_i and final position F_i within the given polygon R, where a position is specified by a set of integer coordinates for the corners of the corresponding block, and all initial positions (as well as all final positions) are

---

*Actually, the abovementioned Hungarian Rings puzzle, which consists of two intersecting circular tracks by means of which colored balls can be scrambled and unscrambled, isn’t so recent itself (nor is it Hungarian). It was patented in 1893 by William Churchill of Detroit, Michigan [8].

disjoint.

QUESTION: Supposing that all blocks start in their initial positions, is there a way of continuously moving the blocks so as to reach a configuration in which each block is in its final position, and such that at no time do any two blocks overlap (i.e., intersect in a region of non-zero area) nor does any block extend beyond the confines of the bounding polygon R?

Reference. Spirakis and Yap [38]. Transformation from 3-PARTITION.

Comment. NP-hard and conjectured to be PSPACE-hard. Remains NP-hard even if R is a rectilinear polygon, and no matter whether rotations are allowed or prohibited. The 15-puzzle corresponds to the case where there are 15 blocks, all unit squares, and R is a 4×4 rectangle, which has room for a 16th block, and hence for rearranging the 15 blocks already present. Other sliding block puzzles that can be viewed as special cases are described in [4,20]. The general problem can also be viewed as a model for questions about planning the coordinated motion of ensembles of robots, and the current results are extended from rectangular robots to circular ones in [39]. If non-convex robots are allowed, the problem becomes PSPACE-hard [22]. (The problem for one robot is solvable in polynomial time, as was mentioned in the Column 6 [March 1983] discussion of ROBOT ARM REACHABILITY.)

A more tractable generalization of the 15-puzzle is the following puzzle on graphs, where the vertices correpond to the possible positions for blocks (now called *pebbles*), and after each step only one position remains empty: We are given a connected graph $G = (V,E)$ and $|V| - 1$ distinct pebbles, each with an initial position and a desired final position. No two pebbles have the same initial position or the same final position. A *step* is the movement of a pebble from one of the neighbors of the currently empty vertex to that vertex. Is there a sequence of steps by which all the pebbles can be moved from their initial to their final positions? In the 15-puzzle, G is the 4×4 grid graph. By a result of Wilson [44], this general graph puzzle can be solved in polynomial time; in fact, with certain trivial exceptions, if G is not bipartite, the answer is always ''yes.'' If G is bipartite, the answer depends on the parity of a composite permutation based on the initial and final positions. Given that the answer is ''yes,'' the desired sequence of interchanges can be found in polynomial time by a clever application of the techniques of Driscoll and Furst mentioned above [38,40]. If there are fewer than $|V| - 1$ pebbles the problem is also polynomial except possibly in the following two open cases: (1) graphs that are not biconnected and (2) biconnected bipartite graphs with precisely $|V| - 2$ pebbles [40].

The 15 puzzle can be viewed as a partially filled matrix that we wish to rearrange. In our next problem, we are given a partially filled matrix and wish to fill it up.

[2] PARTIAL LATIN SQUARE EXTENDABILITY

INSTANCE: An $n \times n$ matrix M, with entries from the set $\{0,1,...,n\}$, such that no column or row contains any repeated entry other than 0. (M corresponds to a ''partial Latin square,'' with the 0's corresponding to missing entries. For discussions of Latin Squares and their history, see [2,4,19].)

QUESTION: Can M be extended to a (full) *Latin square*, i.e., can we replace each 0 entry in M by a member of $\{1,2,...,n\}$ in such a way that no column or row contains a repeated entry?

Reference. Colbourn [10]. Transformation from EDGE PARTITION INTO TRIANGLES (see Column 3 [June 1982]).

Comment. NP-complete, as is the question of whether M has more than one completion to a Latin square (even when given the first as part of the input) [11]. The corresponding question for symmetric Latin squares, i.e., assuming the given matrix M is symmetric, can it be extended to a Latin square that is also symmetric, is also NP-complete, even if M is known to be extendable to a *non-symmetric* Latin square [9].

The final puzzle in this section will probably not be found in the recreational mathematics literature; its real source is a problem in computer register allocation.

[3] THE PEBBLE GAME

INSTANCE: Acyclic directed graph $G = (V,A)$ with a distinguished vertex v_0 that is the only vertex in V with out-degree 0 (i.e., is the *root* of G), positive integer k.

QUESTION: Can one, while obeying the following rules, place a pebble on v_0?

(1) At no time can there be more than k pebbles on the graph.

(2) A pebble can be removed from the graph at any time.

(3) A pebble can be placed on a vertex v if and only if all vertices u such that $(u,v) \in A$ currently contain pebbles. (Note that this implies that vertices with in-degree 0 can be pebbled at any time.)

Reference. Gilbert, Lengauer, and Tarjan [21]. Transformation from QUANTIFIED BOOLEAN FORMULAS [LO11].

Comment. PSPACE-complete, which is perhaps surprising given that this is not a two-person game and hence there is no obvious form of alternation. The complexity comes from the possibility that vertices may be repeatedly pebbled and unpebbled, and indeed this may be necessary if k pebbles are to suffice. If no re-pebbling is allowed, the problem becomes the REGISTER SUFFICIENCY [PO1] problem of [G&J] and is only NP-complete. This was shown in [37], which also proved the current problem to be at least NP-hard. In the register allocation interpretation, the graph represents the dependency structure of intermediate results in a computation, and the pebbles correspond to the registers in which intermediate results are stored. Re-pebbling a vertex corresponds to recomputing an intermediate result. Since computers usually have instructions of the form ‘‘Modify the contents of register K by incrementing the current contents by *x*,’’ one might also wish to consider the variant in which rule (3) is modified so that the pebble placed on *v* can be one of those currently on *v*’s immediate predecessors. This variant is also PSPACE-complete [21].

An interesting variant that rates mention as an ‘‘Open Problem of the Month’’ is THE BLACK-WHITE PEBBLE GAME, introduced by Cook and Sethi [13]. In addition to ‘‘black’’ pebbles that follow the rules (2) and (3), this game also includes *white* pebbles that follow dual rules: (2′) a white pebble can be placed on any vertex at any time, and (3′) a white pebble cannot be *removed* unless all its immediate predecessors currently contain pebbles (the colors of the pebbles on the predecessors is not important). Starting with an empty graph, the goal is now to pebble the root and then return the graph to an unpebbled state. As before, only *k* pebbles can be on the graph at any time, but there is no constraint on how their colors are distributed (and this distribution can change as pebbles are removed and replaced). This problem models a nondeterministic version of computation in which one is allowed to ‘‘guess’’ the value of an intermediate result, so long as one eventually verifies it by obtaining (either deterministically or nondeterministically) all the values upon which it depends. Such nondeterministic computations can sometimes get by with fewer registers than are required by ordinary machines. However, the complexity of determining just how many registers (pebbles) suffice remains open, except in the case where re-pebbling is forbidden. Here NP-completeness reigns [28], just as it did in the monochromatic problem. With re-pebbling allowed, the problem, although conjectured to be PSPACE-complete, is not even known to be NP-hard.

3. GAMES PEOPLE DON’T PLAY

Two person games can also serve as models for practical problems about computers. For instance, in [25] such a game is used to model a problem in distributed concurrency control and prove that the problem is PSPACE-complete. (The problem is the DISTRIBUTED SERIALIZABILITY ASSURANCE problem mentioned in my last column [Sep. 1983]; the game involves the progressive conversion of an undirected graph to a directed one, with the goal of Player 1 being to postpone the creation of a directed cycle for as long as possible.) In [27] another game is used to model the ‘‘lockout’’ problem for systems of communicating processes. Such games are usually of little interest outside their immediate application, however, so I shall spare the reader the details, and instead go on to a game that, at first glance, would seem to be a rather arbitrary variant on the pebble games mentioned in the last section.

[4] YET ANOTHER PEBBLE GAME

INSTANCE: Acyclic directed graph $G = (V,A)$ with a distinguished *goal vertex* v_0, set $S \subseteq V$ of *start vertices*, and a collection R of *rules*, each rule being a triple (x,y,z) of distinct vertices, where (x,z) and (y,z) are both arcs in A.

QUESTION: Does the first player have a forced win in the following game: Initially there is a pebble on each vertex in S. Given a configuration of pebbles on G, a rule $(x,y,z) \in R$ is *applicable* if there are pebbles on x and y but not on z. If (x,y,z) is applicable, the corresponding *application* consists of moving the pebble from x to z. The players alternate choosing applicable rules and applying them. A player loses if there are no applicable moves on his turn, or if his opponent ever places a pebble on v_0.

Reference. Kasai, Adachi, and Iwata [26]. Transformation from LINEAR BOUNDED AUTOMATON ACCEPTANCE [AL3].

Comment. PSPACE-complete. The corresponding one-person game (can a single player pebble v_0 by starting from the intial position and repeatedly applying applicable rules?) is NP-complete. If the graph is not required to be acyclic, the two-person game becomes EXPTIME-complete, and the one-person game is promoted to PSPACE-completeness. At second glance this game still seems like an arbitrary variant on our original pebble game. It does, however, have an interesting property. Suppose we restrict attention to just those (not-necessarily-acyclic) instances where $|S| = 2k+1$, for some fixed k. This restricted problem can be shown to *require* $\Omega(n^k)$ time on a one-tape deterministic Turing machine, where n is the number of vertices [1], even though it can be solved in polynomial time (by exhaustive search of the game tree − there are only $O(n^{2k+1})$ possible positions). Although problems in P with such nontrivial lower bounds are known to exist by diagonalization arguments (e.g., see [23]), this is to my knowledge the first example of a ‘‘natural’’ problem with this property. Other examples, also from [1], include a variant on the ‘‘cat and mouse’’ game to be discussed in the next section, and a multi-peg variant on the famous [4,18] Tower of Hanoi puzzle.

Having now paid our dues by surviving a series of results that have real computer science applications, let us once again return to the good stuff: games that, at least in their ungeneralized form, have actually been played. I shall begin with board games (readers unfamiliar with the rules of the following games should consult a general reference such as [14]). As evidenced by the results presented in [G&J], there are three progressively more general (and progressively less natural) ways of taking a standard board game to asymptopia: (1) Allow the dimensions of the board (and the number of pieces) to grow arbitrarily large, and ask whether a given ‘‘endgame position’’ is a forced win (as in N × N CHECKERS [GP10] and N × N GO [GP11]), (2) Generalize the board to an arbitrary planar graph (PLANAR GEOGRAPHY [GP2] would be an example if only Geography were a board game), and (3) Generalize the board to an arbitrary (not-necessarily-planar) graph (GENERALIZED HEX [GP1]). A sizable collection of new results has been proved along these lines.

(1). The result for HEX has been strengthened by Reisch [35], who shows that N × N HEX is PSPACE-complete. Actually, he shows that it is ‘‘PSPACE-vollständig,’’ but since there is a log-space transformation from (technical) German to English, the result follows.

(2). N × N CHESS is PSPACE-complete if the rule ‘‘50 moves without a pawn move or a capture yields a draw’’ is generalized to one in which the ‘‘50’’ is replaced by a fixed polynomial in N [42]; if the rule is omitted the problem becomes complete for EXPTIME [16], as does N × N CHECKERS [36]. (Without such a drawing rule, N × N GO is also a candidate for EXPTIME-completeness.)

(3). N × N GOBANG (GO-MOKU) is PSPACE-complete, even if the goal of getting 5 stones in a row is retained without generalization [34].

(4). GENERALIZED CHINESE CHECKERS (played on arbitrary graphs) is EXPTIME-complete, via a transformation from YET ANOTHER PEBBLE GAME [26], although the planar and N × N versions (if a star-shaped board can be considered N × N) remain open.

(5). In a result that wins the award for the generalization of the most obscure game, N × N PRESS-UPS® is PSPACE-complete [30]. (‘‘Press-Ups’’ is a trademark of Invicta, Inc.)

When we go beyond the realm of board games, we find that complexity theorists were not the first to analyze generalized version of games. Many games, such as Nim [3,4,18], have already been completely ‘‘solved’’ in their general form (see [18] for references going back to 1901). All that complexity theorists can contribute is the observation that the ‘‘solutions’’ are actually polynomial time algorithms. For an extensive survey of (and elaboration on) what is known about strategies for playing such combinatorial games, see the delightful books by Conway [12] and Berlekamp, Conway, and Guy [3,4]. I conclude this section with a result from the latter about a ‘‘board game’’ whose standard version is already defined asymptotically: ‘‘Dots and Boxes.’’ The basic starting position of this game is an N × N grid, and, although small children usually set N equal to 3 or 4, I can recall long battles on fields with much larger N in the back seats of my high school history class. The only real limitation on N is the fact that the N^2 grid points normally have to be hand-drawn before play can begin.

Despite the asymptotic nature of the standard game, a slight generalization is still required for the current complexity result to hold, and I shall describe this generalization in detail. (The reference, although always entertaining, is sometimes a bit obscure, using such technical terms as ‘‘loony endgame’’ and ‘‘double-dealing move,’’ and in general proceeding as one might expect from authors capable of inventing ‘‘Left-Right Hackenbush for Redwood Furniture’’ [GP12].)

[5] GENERALIZED DOTS AND BOXES

INSTANCE: Planar graph $G = (V,E)$, positive integer *handicap* K.

QUESTION: Can Player 1 guarantee himself a score no worse than K points behind that of his opponent in the following game: Initially, all edges in E are *undrawn*. On a player’s turn, he chooses an undrawn edge and ‘‘draws’’ it. If, in doing so, he completes the perimeter of a (non-infinite) region of G (draws the last undrawn edge on the region’s boundary), he scores one point for each such region completed (there may be a second one), and gets another turn. (In the standard game, these regions are the $(N-1)^2$ grid squares in the $N \times N$ grid graph.) Otherwise, his turn is over and the other player proceeds. The game ends when all edges of G have been drawn.

Reference. Berlekamp, Conway, and Guy [4]. Transformation from planar PARTITION INTO TRIANGLES [GT11].

Comment. NP-hard. Actually, Berlekamp et al. did not show the planar problem NP-hard, but rather the generalization to arbitrary graphs of the dual version of the game (which they call *Strings and Coins*). In this dual game (which is equivalent to the given one for planar graphs), edges are not ‘‘drawn’’ but ‘‘cut,’’ and a point is scored for each vertex the current cut succeeds in isolating from all its neighbors. Based on a theory of optimal endgame play, Berlekamp et al. show that the problem reduces to one of Player 1 identifying a maximum number of disjoint cycles in G, and hence to PARTITION INTO TRIANGLES. What they did not know is that the latter problem has recently been proven NP-complete for planar graphs [5], which means that they needn’t have generalized quite so far. If the NP-completeness of PARTITION INTO TRIANGLES also holds for planar graphs of maximum degree 4 (which unfortunately is not implied by the proof in [5]), then we would be able to obtain NP-hardness for optimal endgame play in standard DOTS AND BOXES (the handicap K would then simply be the current lead of Player 1 over Player 2 in the given endgame position).

4. GAMES OF CHANCE AND OTHER UNCERTAINTIES

There is often a major discrepancy between the way problems are described for complexity-theoretic purposes and the way they actually occur in practice. When you see a problem described in this column for example, a typical ‘‘instance’’ is described completely and precisely, as if all graphs, distances, weights, etc. were known in advance and available to the problem-solver. In practice, topologies may be subject to change, distances may be inaccurately measured, and weights may just be educated guesses based on some unknown probability distribution. Presumably, all this uncertainty just makes problems more difficult, but quantifying this increase in difficulty has until recently been beyond the reach of complexity theory. This section discusses two approaches to extending that reach, both of which use games as their model.

In [32], Reif considers the complexity of games of ‘‘incomplete information,’’ i.e., games in which the players may not know all the details of the current position, as in Poker, where players may not know all the cards their opponents are holding. As an illustration of the types of results he obtains, let us consider the following game as an example. In its ‘‘perfect information’’ form, as I shall initially describe it, the game is due to Chandra and Stockmeyer [7,41] and was one of the first known examples of an EXPTIME-complete game.

[6] PEEK

INSTANCE: Integer $K > 0$, set A of *plates*, where each plate is a $K \times K$ square with holes punched through it at certain integer-coordinate interior points and with one of its four sides specified as a *handle*, and a partition of A into sets A_0, A_1 and A_2, where A_0 consists of a single ‘‘fixed’’ plate a_0 each hole of which is labelled ‘‘1’’ or ‘‘2’’, and A_1 and A_2 are the plates belonging to Players 1 and 2 respectively.

QUESTION: Does Player 1 have a forced win in the following game: Initially all plates are stacked in a pile with their edges aligned, their handles all pointing in the same direction, and with plate a_0 on top. Physically, we may imagine the plates as shoved all the way into a frame with grooves along its interior so that individual plates may subsequently be partially withdrawn. On a player’s turn, he has three options: he may do nothing (‘‘pass’’), pull one of his plates halfway out of the box (in the direction of the plate’s handle), or push back one of his previously withdrawn plates; each plate has just two positions: ‘‘in’’ and (halfway) ‘‘out.’’ Player i, $i \in \{1,2\}$, wins if, at the end of his move, there is some hole labelled i in the fixed plate a_0 that is aligned with holes in all the other plates, i.e., if he can ‘‘peek’’ all the way through the stack of plates via one of the holes in a_0 belonging to him.

Reference. Chandra and Stockmeyer [7,41].

Comment. EXPTIME-complete. As with $N \times N$ Chess, the complexity of this game arises from the potential for exponentially long games, since if we specify a polynomial p and declare that all plays of the game that last more than $p(K)$ moves are draws, the problem becomes ''only'' PSPACE-complete.

Suppose now that we modify this game by hiding some of the information from the players. In particular, suppose that, although both players know the structure and ownership of all the plates, each is somehow blocked from viewing any of the positions of his opponent's plates. Reif [32] calls this new game BLIND PEEK, and shows that the lack of information indeed makes the problem of telling whether Player 1 has a forced win harder (assuming standard conjectures): it is now complete for exponential *space*. If we next turn to the version in which each player is only blocked from viewing *some* of his opponents plates, we gain startling confirmation of the dictum that a little knowledge is a dangerous thing: for this game (PRIVATE PEEK), the problem is complete for *double exponential time* (the union for all $c > 0$ of time $O(2^{2^{cn}})$), and hence conceivably more difficult than BLIND PEEK. (This is not as paradoxical as it sounds, given that a forced win is much less likely in BLIND PEEK, and hence the possibility that one exists may be easier to rule out.) A further complexity can be added to the situation if we replace the players by 2-person teams, with the two team members playing alternately and having their own personal sets of plates, only some of which are visible to their teammates. This new problem (TEAM PEEK) is undecidable [31]. So much for teamwork.

For more on the precise nature of the complexity jumps due to incomplete information, including methods of modelling the phenomena using Turing machines with ''private'' tapes, see [31,32]. Also see [24] for an earlier, more restricted result along these lines. For entertainment, here is one more problem from [32].

[7] BLIND PURSUIT

INSTANCE: Directed graph $G = (V,A)$, specified vertices c, m, and h, where c represents the starting position of the *cat*, m represents the starting position of the *mouse*, and h represents the *mouse hole*.

QUESTION: Does a blind cat have a strategy for capturing a blind mouse in the following game: Starting from their initial positions at c and m, the cat and the mouse alternate turns, on each turn traversing some directed arc from their current position (vertex) to a neighboring one. The goal of the mouse is to reach the mouse hole h. The goal of the cat (who is too big to occupy the mouse hole) is to catch (i.e., occupy the same vertex as) the mouse before it reaches its hole. The cat and the mouse both know the structure of G and the location of h, but at any given time they know the position of their adversary only if he is close enough to be detected by sense of smell (to be precise, for each of the vertices to which they can move, they know if their adversary either occupies it or can reach it in one step).

Reference. Reif [32]. Transformation from FINITE STATE AUTOMATON INEQUIVALENCE [AL1].

Comment. PSPACE-complete, but solvable in polynomial time if someone turns on the lights (i.e., if the cat and mouse both have complete information). If there is actually a team of cats looking for the mouse, with every cat required to make a move between each two successive mouse moves, then the problem is EXPTIME-complete when the lights are on, and EXPSPACE-complete with the lights off, assuming that the cats all have two-way radios and hence know where their teammates are [32].

Another way in which a game player can have imperfect information about his situation is when the game involves a random element, such as the spin of a wheel or a roll of the dice. One-person games with a random element are especially interesting, as they can be used to model many practical problems involving attempts to optimize performance from a probabilistic point of view. In his paper ‘‘Games against nature’’ [29], Papadimitriou provides techniques for analyzing how the random element affects the complexity of such problems.

His key observation is that, although technically these are one-person games, they can more profitably be viewed as two-person games against a ‘‘random opponent,’’ and that such games are basically no harder than ordinary two-person games (although often no easier either). Here is one example.

[8] DYNAMIC NETWORK RELIABILITY

INSTANCE: Graph $G = (V,E)$, specified start and finish vertices s and t, and for each pair (v,e), $v \in V$ and $e \in E$, a non-negative rational *conditional failure probability* $p(v,e)$.

QUESTION: Is there a way of getting from s to t with probability greater than 1/2 under the following ground rules: You can only move along ‘‘functional’’ edges, one step at a time. Initially all edges are functional, but if you are at vertex v, the probability that edge e will remain functional until you reach your next stop is just $1 - p(v,e)$ for each functional edge $e \in E$ (individual edge failures are assumed to be independent during that interval, however). Once an edge becomes non-functional, it never revives.

Reference. Papadimitriou [29]. Transformation from QUANTIFIED BOOLEAN FORMULAS.

Comment. PSPACE-complete. Here the ‘‘random opponent’’ is the process causing the edges to fail. This result can be contrasted with the ‘‘static’’ problem, in which failure probabilites $p(e)$ depend only on the individual edges, and we ask for the probability that there remains a path from s to t, given that each edge e has failed (independently) with probability $p(e)$ (see the comments to NETWORK RELIABILITY [ND20]). Here we make the more realistic assumption that paths cannot be traversed instantaneously, and that your presence in the network can affect its behavior. The static problem is ''only'' #P-complete [43], and hence possibly easier than the dynamic one.

I conclude by returning once more to questions about familiar games. The above results raise the possibility that, with similar techniques, we might at last be able to address the complexity of such long open problems as generalized Bridge and Backgammon. My colleagues assure me that determining the complexity of these games would fill a ''much-needed gap'' in the literature, and so I propose them as additional ''Open Problems of the Month.''

REFERENCES

1. A. ADACHI, S. IWATA, AND T. KASAI, Low level complexity for combinatorial games, *in* ''Proceedings 13th Ann. ACM Symp. on Theory of Computing,'' pp. 228-237, Association for Computing Machinery, New York, 1981.
2. W. W. R. BALL AND H. S. M. COXETER, ''Mathmatical Recreations and Essays,'' University of Toronto Press, Toronto, 1974.
3. E. R. BERLEKAMP, J. H. CONWAY, AND R. K. GUY, ''WINNING WAYS for your mathematical plays, Volume 1: Games in General,'' Academic Press, New York, 1982.
4. E. R. BERLEKAMP, J. H. CONWAY, AND R. K. GUY, ''WINNING WAYS for your mathematical plays, Volume 2: Games in Particular,'' Academic Press, New York, 1982.
5. F. BERMAN, F. T. LEIGHTON, P. SHOR, AND L. SNYDER, private communication (1983).
6. A. K. CHANDRA, D. C. KOZEN, AND L. J. STOCKMEYER, Alternation, *J. Assoc. Comput. Mach.* **28** (1981), 114-133.
7. A. K. CHANDRA AND L. J. STOCKMEYER, Alternation, *in* ''Proceedings 17th Ann. Symp. on Foundations of Computer Science,'' pp. 98-108, IEEE Computer Society, Los Angeles, 1976.
8. W. CHURCHILL, ''Puzzle,'' U. S. Patent 507,215, Oct. 24, 1893.
9. C. J. COLBOURN, Embedding partial Steiner triple systems is NP-complete, *J. Combinatorial Theory Ser. A*, to appear.
10. C. J. COLBOURN, The complexity of completing partial Latin squares, *Disc. Applied Math.*, to appear.
11. C. J. COLBOURN, M. J. COLBOURN, AND D. R. STINSON, The computational complexity of recognizing critical sets, *in* ''First Southeast Asian Conf. on Graph Theory,'' Department of Mathematics, National University of Singapore, 1983 (to appear).
12. J. H. CONWAY, On numbers and games, Academic Press, New York, 1976.
13. S. COOK AND R. SETHI, Storage requirements for deterministic polynomial time recognizable languages, *J. Comput. System Sci.* **13** (1976), 25-37.
14. THE DIAGRAM GROUP, ''The Way to Play,'' Bantam Books, New York, 1977.
15. J. R. DRISCOLL AND M. L. FURST, On the diameter of permutation groups, *in* ''Proceedings 15th Ann. ACM Symp. on Theory of Computing,'' pp. 152-160, Association for Computing Machinery, New York, 1983.

16. A. S. Fraenkel and D. Lichtenstein, Computing a perfect strategy for $n \times n$ chess requires time exponential in n, *J. Combinatorial Theory Ser. A* **31** (1981), 199-213.
17. M. Furst, private communication (1983).
18. M. Gardner, ''The Scientific American Book of Mathematical Puzzles and Diversions,'' Simon and Schuster, New York, 1959.
19. M. Gardner, ''New Mathematical Diversions from Scientific American,'' Simon and Schuster, New York, 1966.
20. M. Gardner, ''The Sixth Book of Mathematical Games from Scientific American,'' Charles Scribner's Sons, New York, 1971.
21. J. R. Gilbert, T. Lengauer, and R. E. Tarjan, The pebbling problem is complete for polynomial space, *SIAM J. Comput.* **9** (1980), 513-524.
22. J. E. Hopcroft, J. T. Schwartz, and M. Sharir, PSPACE-hardness of the motion planning problem for rectangular objects, manuscript (1983).
23. J. E. Hopcroft and J. D. Ullman, ''Formal Languages and their Relation to Automata,'' Addison-Wesley, Reading, Mass., 1969.
24. N. D. Jones, Blindfold games are harder than games with perfect information, *Bulletin of the European Assoc. for Theoretical Comput. Sci.* **6** (1978), 4-7.
25. P. C. Kanellakis and C. H. Papadimitriou, The complexity of distributed concurrency control, *in* ''Proceedings 22nd Ann. Symp. on Foundations of Computer Science,'' pp. 185-197, IEEE Computer Society, Los Angeles, 1981.
26. T. Kasai, A. Adachi, and S. Iwata, Classes of pebble games and complete problems, *SIAM J. Comput.* **8** (1979), 574-586.
27. R. E. Ladner, The complexity of problems in systems of communicating sequential processes, *J. Comput. System Sci.* **21** (1980), 179-194.
28. T. Lengauer, Black-white pebbles and graph separation, manuscript (1980).
29. C. H. Papadimitriou, Games against nature, *in* ''Proceedings 24th Ann. Symp. on Foundations of Computer Science,'' IEEE Computer Society, Los Angeles, 1983.
30. G. L. Peterson, Press-Ups is PSPACE complete, manuscript (1979).
31. G. L. Peterson and J. H. Reif, Multiple-person alternation, *in* ''Proceedings 20th Ann. Symp. on Foundations of Computer Science,'' pp. 348-363, IEEE Computer Society, Los Angeles, 1979.
32. J. H. Reif, Universal games of incomplete information, *in* ''Proceedings 11th Ann. ACM Symp. on Theory of Computing,'' pp. 288-308, Association for Computing Machinery, New York, 1979.
33. J. H. Reif, The complexity of two player games of incomplete information, *J. Comput. System Sci.*, to appear.
34. S. Reisch, Gobang ist PSPACE-vollständig, *Acta Inform.* **13** (1980), 59-66.
35. S. Reisch, Hex ist PSPACE-vollständig, *Acta Inform.* **15** (1981), 167-191.
36. J. M. Robson, N by N Checkers is Exptime-complete, *SIAM J. Comput.*, to appear.
37. R. Sethi, Complete register allocation problems, *SIAM J. Comput.* **4** (1975), 226-248.
38. P. Spirakis and C. Yap, ''On the combinatorial complexity of motion coordination,'' Report No. 76, Computer Science Department, New York University, 1983.
39. P. Spirakis and C. Yap, Strong NP-hardness of moving many discs, manuscript (1983).
40. P. Spirakis and C. Yap, Moving many pebbles in a graph is polynomial time, manuscript (1983).
41. L. J. Stockmeyer and A. K. Chandra, Provably difficult combinatorial games, *SIAM J. Comput.* **8** (1979), 151-174.
42. J. A. Storer, On the complexity of chess, manuscript (1979).
43. L. G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979), 410-421.

44. R. W. WILSON, Graphs, puzzles, homotopy, and the alternating group, *J. Combinatorial Theory Ser. B* **16** (1974), 86-96.