BATTLESHIPS AS DECISION PROBLEM

Merlijn Sevenster[1]

Amsterdam, The Netherlands

ABSTRACT

We define the well-known puzzle of Battleships as a decision problem and prove it to be NP-complete, by means of a parsimonious reduction. By applying Valiant and Vazirani's (1986) result we immediately obtain that the variant of this problem, viz. promising that there is a unique solution, is NP-complete as well, under randomized reductions. As this finding is in sheer contrast with the general experience of Battleship puzzles being well playable (i.e., effectively solvable), we arrive at a hypothetical explanation for this state of affairs.

1. INTRODUCTION

Nowadays many newspapers and magazines contain a variety of puzzles. A great deal of them can be called *logical puzzles*, as every step in the process of solving is supposed to be logically necessitated. Popular examples of such games are Battleships and Nonograms (or Japanese puzzles). (Note that certain Minesweeper puzzles can be considered logical puzzles as well, ignoring the ones in which one at some stage inevitably has to guess to get any further.)

In this article we study the logical puzzle of Battleships (also known as 'Fathom it!') from a complexity point of view. Battleships puzzles are well playable and are regularly found in magazines, newspapers, and on the Internet, (e.g., Puzzelsport, 2003). This particular formal angle learns us to appreciate the relationship between the logical (and effective) solvability on the one hand and the consequences concerning computational classification on the other hand. Although we will only discuss Battleships, we are convinced that the formal results established in this contribution can be transferred to many other logical puzzles.

A Battleships puzzle consists of a puzzle-grid, a column and row tally, and a fleet containing a certain number of ships of varying length (the width of a ship is 1). The initial grid is partially filled with ship segments or water (denoting the absence of a ship segment). The goal of a puzzle is to show that the provided ships can be positioned on the puzzle-grid, meeting the following four conditions:

C1: all ships in the fleet are put in the grid;

C2: the indications in the initial grid are respected;

C3: no two ships occupy adjacent (orthogonally or diagonally) squares;

C4: the number of ship segments in column (row) i is equal to the ith value of the column (row) tally.

As a case in point, we reproduce a small puzzle in Figure 1 that is used in a commercial magazine to explain the essence of Battleships. The puzzle on the left-hand side shows the initial puzzle; the puzzle in the middle shows it at some intermediary stage of the solving process; the puzzle on the right-hand side depicts the solution. As the reader can check the last puzzle meets conditions C1 to C4.

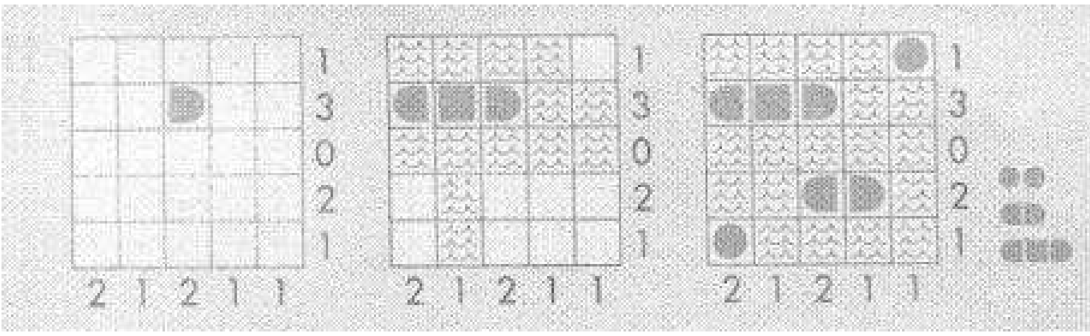

Figure 1: The almost-empty grid on the left plus the fleet (given on the right) make up a Battleships puzzle, that has the puzzle on the right as its solution. The puzzle in the middle depicts an intermediary stage one might encounter while solving the puzzle. The puzzle on the right satisfies conditions C1 to C4. (Puzzle copied, with permission, from July/August 2003 *Puzzelsport* issue. Copyrights remain with *Puzzelsport*.)

1.1 The Course of the Article

We will define the Battleships puzzle as a decision problem in Section 2 and argue that it is obviously NP-complete. In Section 3 we give a parsimonious reduction, that again establishes NP-completeness of the Battleships decision problem.

A reduction is parsimonious, if it preserves the number of solutions. Valiant and Vazirani (**Undefined reference**) showed that if the classical satisfiability problem is parsimoniously reducible to some problem A, then its promise variant, PROMISE A is also NP-hard, under randomized reductions. Valiant and Vazirani define PROMISE A as follows:

Input:	an instance a of A
Output:	a solution to a
Promise:	a has exactly one solution.

We will use the property of parsimonicity in Section 4, when we will show that the promise variant of the Battleships problem is NP-complete as well, under randomized reductions. This result is interesting, since the promise variant of the problem is the puzzle we actually solve at our leisure, assuming that every Battleships puzzle has exactly one solution. As to Battleships, printed solutions in magazines or an expert function on Internet are tacit hints in this direction. As to other logical puzzles such as Nonograms, the solution is a picture, and as such must be unique. NP-hardness of the promise variant of Battleships is surprising in the sense that while solving a puzzle that is guaranteed to have a unique solution, one does not feel forced to guess non-deterministically the positions of ship segments or water. Instead, every step in the process of solving a 'commercial' puzzle seems necessitated by the current state of the puzzle and the conditions C1 to C4. In fact, this is what makes Battleships puzzles in magazines logical puzzles. In Section 5, we give hints as to how one might generate logical Battleships puzzles that are solvable by humans. Section 6 contains a summary of results.

2. THE DECISION PROBLEM OF BATTLESHIPS

Formally, we treat an $m \times n$-sized Battleships puzzle (m rows and n columns) as a tuple $\langle I, C, R, F \rangle$, where

$$I : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \{\textit{ship}, \textit{water}, ?\}$$

is a function that represents the initial filling of the starting grid. For instance, $I(i, j) = \textit{ship}$ denotes that cell (i, j) contains a ship segment. If $I(i, j) = ?$, cell (i, j)'s filling is not given: it is up to the puzzle solver

[1]ILLC, Universiteit van Amsterdam, Amsterdam, The Netherlands. Email: sevenstr@science.uva.nl.

to fill it with either a water or a ship segment. C is a function representing the column tally by adding a number[2] to every column $i \in \{1, \ldots, n\}$; idem for the function R representing the row tally, with respect to $j \in \{1, \ldots, m\}$. The function F represents the fleet: there are $F(k)$ ships of ship-length k.

If J is a function such that

$$J : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \{\mathit{ship}, \mathit{water}\};$$

it is the case that $I(i, j) = J(i, j)$, if $I(i, j) \neq ?$; and J meets condition C1 to C4 with respect to C, R and F, then J is a *solution* to the Battleships puzzle $\langle I, C, R, F\rangle$. This formalizes the notion of solution that we used intuitively, when we considered the small puzzle in Figure 1.

In some Battleships puzzles one does not only learn that there is a ship segment in a certain field, but also what *kind* of ship segment is on that field, e.g., the ship segment on cell $(2, 3)$ in Figure 1. This limitation, however, does by no means affect any of the results in this article, as every problem as defined above is also a decision problem in the broader sense.

Throughout the article we will take it for granted that we can harmlessly switch between (parts of) the formalization of a puzzle and (parts of) its visual representation. For instance, it is an easy exercise to formalize the puzzle from Figure 1.

We base the decision problem BATTLESHIPS on the Battleships puzzle as follows: Given a Battleships puzzle $\langle I, C, R, F\rangle$; does this puzzle have a solution?

It is easily obtained that BATTLESHIPS is NP-complete. Below we give a reduction, proposed by Kaye (**Undefined reference**), from BIN PACKING. In line with Papadimitriou (**Undefined reference**), we define (a slight variant of) the BIN PACKING problem as follows:

Input:	n positive integers $a_1, \ldots, a_n$ (the items), two integers C (the capacity) and B (the number of bins) such that $a_1 + \ldots + a_n = CB$
Question:	can $a_1, \ldots, a_n$ be partitioned into B subsets, each of which has total sum exactly C?

We transform an instance of BIN PACKING into a BATTLESHIPS puzzle by reserving a vertical strip of length a_i per pair of item a_i and bin $b \in \{1, \ldots, B\}$. All other cells are initially filled with water. Note that this reduction assumes that the numbers $a_1, \ldots, a_n$ are represented in *unary*, since the strips themselves are unary representations of these numbers. This assumption does not effect the result, since the problem BIN PACKING is still NP-complete if its numbers are represented in unary. This property makes the problem *strongly* NP-complete. For definitions we refer to **Undefined reference**.

The idea is that every open strip represents the possibility of item a_i being put in bin b (see Figure 2a).The item itself is represented by a ship of length a_i. To make sure that every item a_i is placed in only one bin, we have 1s on the relevant row tally. This also enforces that every ship of length a_i is put on a strip of *exactly* length a_i. The reason is as follows: suppose one would put a ship of length less than a_i on a strip of length a_i, then at least one ship of greater length a_j would have to be placed on a strip which is shorter than a_j; this is evidently impossible.

This reduction proves that BATTLESHIPS is NP-hard. It is easy to see that a non-deterministic guess is checked to be a solution in polynomial time. Hence BATTLESHIPS is NP-complete. However, although the provided reduction is parsimonious, no such reduction exists from 3-SAT to BIN PACKING. If an instance of BIN PACKING has one solution, it must have $B!$ solutions, due to permuting the contents of the B bins. This prevents us from applying Valiant and Vazirani's (1986) result. In the next section we will therefore prove the existence of a parsimonious reduction from 3-SAT to BATTLESHIPS.

[2] Formally, C is of type $\{1, \ldots, n\} \to \{0, 1, \ldots\}$. Puzzles with *incomplete* tallies are quite playable as well, that is, every column is assigned a natural number or no number at all; which formally boils down to $C : \{1, \ldots, n\} \to \{0, 1, \ldots\} \cup \{?\}$. In this article we will restrict our attention to the latter, as any result proved can be transferred to the former.

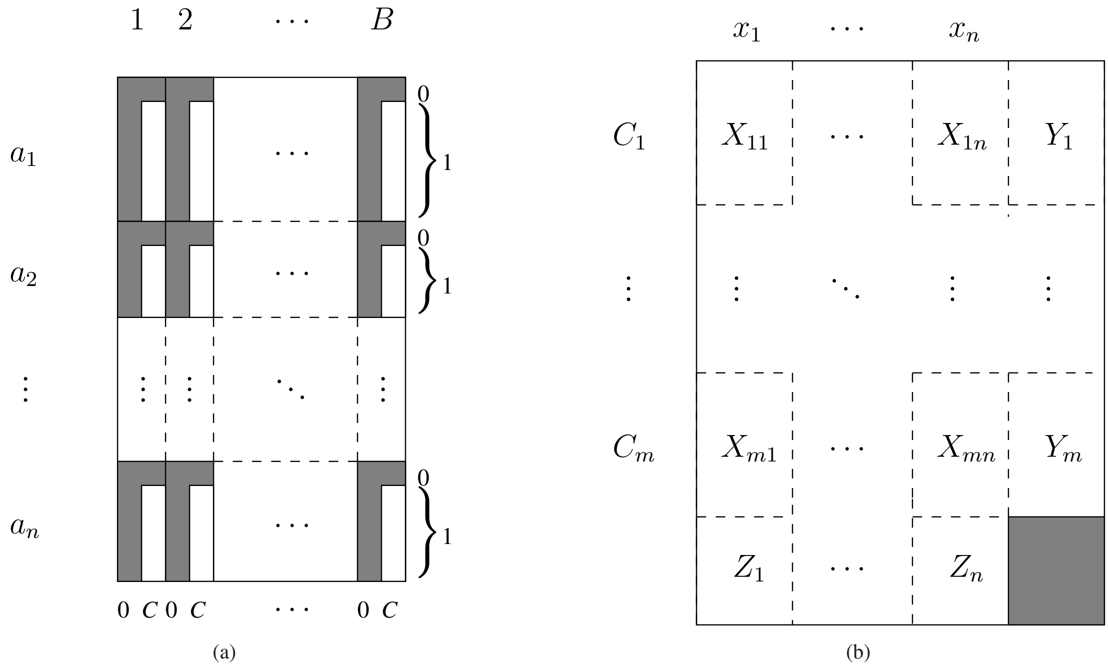

Figure 2: (a) schematically shows a reduced BIN PACKING instance. Every column $b \in \{1, \ldots, B\}$ represents a bin, that possibly contains item a_i. Every item a_i is represented by one ship of length a_i in the resulting puzzle; putting item a_i in bin b amounts to putting a ship of length a_i in the bth column. (b) depicts the grid that forms the basis of the reduction of φ to an instance of BATTLESHIPS. The lower right-corner is always filled with water. Depending on the structure of φ, gadgets will be put on the regions marked with X, Y, and Z.

3. THE PARSIMONIOUS REDUCTION

In this section, we will show the existence of a parsimonious reduction from 3-SAT to our Battleships decision problem.

Theorem There exists a parsimonious reduction from 3-SAT to BATTLESHIPS.

Proof We will prove this by providing two parsimonious reductions: one from 3-SAT to an intermediary problem $3, \{3, 4\}$-SAT, and the other from $3, \{3, 4\}$-SAT to BATTLESHIPS. 3-SAT is the set of satisfiable boolean formulae that are in conjunctive normal form and have three literals per clause. As usual, we assume that no variable occurs twice in a single clause. $3, \{3, 4\}$-SAT is the subset of 3-SAT that contains all satisfiable formulae with exactly three variables per clause and every variable occurring three or four times. Deciding whether an instance of $3, \{3, 4\}$-SAT is satisfiable was proven NP-complete by Tovey (**Undefined reference**); the result was established by a parsimonious reduction from 3-SAT.

We will devote the remainder of this proof to a parsimonious reduction from $3, \{3, 4\}$-SAT to BATTLESHIPS. To this end, let $\varphi \equiv C_1 \wedge \ldots \wedge C_m$ be an instance of $3, \{3, 4\}$-SAT over the variables $x_1, \ldots, x_n$.

Figure 2b outlines the Battleships puzzle obtained by the reduction of φ. In Figure 2b, we associate with every clause C_i and variable x_j a region X_{ij}. Furthermore, every clause C_i (variable x_j, respectively) is associated with a region marked Y_i (Z_j, respectively). We construct the puzzle by putting gadgets in the regions marked with X, Y, and Z alongside with providing the fleet and the row and column tallies, using φ. We end the construction with fixing the remaining open tally-values. We proceed as follows.

- For X_{ij}: if x_j occurs positively in C_i put in the gadget from Figure 3a; if x_j occurs negatively in C_i put in the gadget from Figure 3b; otherwise — if x_j does not occur in C_i — put in the all-water gadget from Figure 3c. All gadgets are of height $8i + 3$. If x_j occurs in C_i, we add one ship of length

1, the X_{ij}-ship, to the fleet.

- For Y_i: put in the gadget from Figure 3d of height $8i + 3$; add one ship of length $4i + 1$, one ship of length $4i$ and one ship of length 1. The added ships we call Y_i-ships. This gadget also determines the row tally for the rows intersecting Y_i.
- For Z_j: if x_j occurs four times in φ put in the gadget from Figure 3e and add a ship of length 4; otherwise — if x_j occurs three times — put in the gadget from Figure 3f and add a ship of length 3. The added ship we call the Z_i-ship. This gadget also determines the row tally for the rows intersecting Z_j.
- We conclude the construction by assigning tally-values to the bottom-most rows and right-most columns, that until now remained open:

row	*tally-value*	*column*	*tally-value*
$9m + 1$	0	$3n + 1$	$\sum_{1 \leq i \leq m}(4i + 1)$
$9m + 2$	n_4	$3n + 2$	0
$9m + 3$	n	$3n + 3$	$\sum_{1 \leq i \leq m}(4i + 1)$
$9m + 4$	n		
$9m + 5$	n,		

where n_4 equals the number of variables that occur four times in φ.

It suffices to show that the number of truth assignments satisfying φ equals the number of solution to the Battleships puzzle above. In fact we will show that every satisfying truth assignment of φ corresponds to exactly one solution of φ's reduction, and vice versa.

Assume φ has a satisfying truth assignment $t : \{x_1, \ldots, x_m\} \rightarrow \{$*true*,*false*$\}$. Then we will solve the reduced puzzle using t as follows. For every variable x_j occurring in clause C_i we put the X_{ij}-ship

in the *north-east* slot, if x_j occurs positively in C_i and $t(x_j) =$ *true* (see Figure 4a);

in the *south-west* slot, if x_j occurs positively in C_i and $t(x_j) =$ *false* (see Figure 4b);

in the *south-east* slot, if x_j occurs negatively in C_i and $t(x_j) =$ *true* (see Figure 4c);

in the *north-west* slot, if x_j occurs negatively in C_i and $t(x_j) =$ *false* (see Figure 4d).

It is immediate that t is uniquely encoded by the position of the X_{ij}-ships.

Whether the truth assignment of x_j contributes to the truth of its clause C_i depends on whether x_j occurs negatively or positively in C_i. For instance, if x_j occurs negatively in C_i, the gadget from Figure 3b is on region X_{ij}. Furthermore assume that x_j is assigned *true* — so it does not contribute to the truth of C_i —, then the X_{ij}-ship must be put on the right open slot. Consequently it is put on the lower row. From Figure 4 it becomes clear that an X_{ij}-ship is put in the lower open slot iff it does not contribute to C_i being true. Since t is a satisfying assignment, at least one of C_i's literals is true. In the Battleships puzzle this corresponds to at least oneF X_{ij}-ship being put on the upper row.

The gadget on Y_i (see Figure 3d) and its tally can be satisfied for any arrangement of X_{ij}-ships where the number of ships in the upper (lower) row is at least (most) one (two). As to the number of X-ships on the upper row we make a case distinction.

Assume the upper row contains *three* X-ships, then the corresponding row tally is already satisfied. The remainder of gadget Y_i can *only* be solved by arranging the Y_i-ships as in Figure 5a.

Assume the upper row contains *two* X-ships, then Figures 5b and 5c depict the possible solutions. We rule out the solution depicted in 5b by demanding that the first and third column of the Y-gadgets contain $\sum_{1 \leq i \leq m} 4i + 1$ ship segments. If one would choose the pattern from Figure 5b, one would never be able to solve the puzzle due to this constraint.

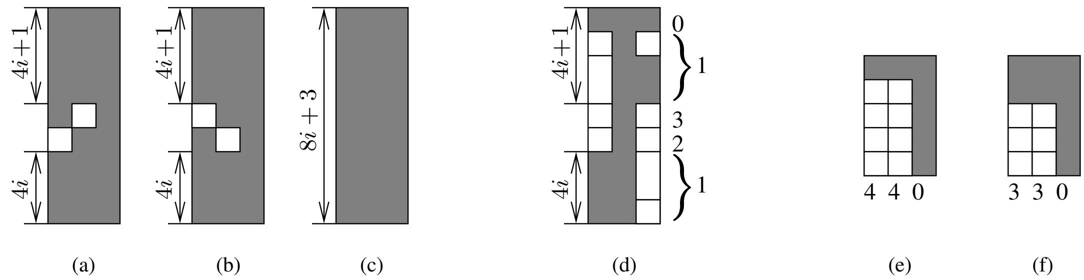

Figure 3: Gadgets a, b, and c are used to fill regions marked with an X. Gadget d is put on regions marked with a Y, and gadgets e and f are put on regions marked with a Z.

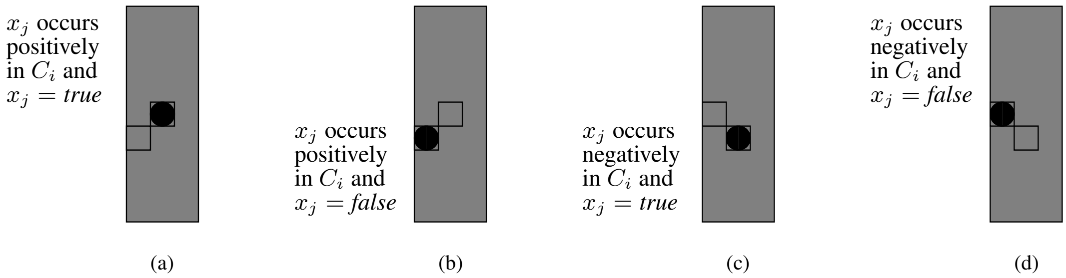

Figure 4: The four possible configurations of an X_{ij}-ship of on an X-gadget, reflecting a variable that is assigned *true* (a and c) or *false* (b and d), while occurring positively in its clause (a and b) or negatively (c and d).

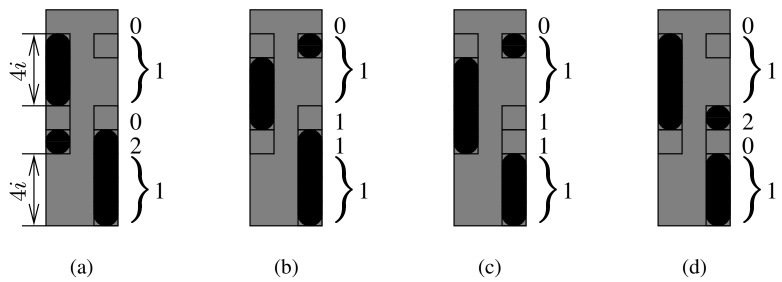

Figure 5: The gadget on Y_i from Figure 3d must be solvable iff the fifth row associated with C_i contains at least one ship segment. Thus, it must be solvable in case this fifth column contains three segments (a); two segments (b and c); and one segment (d).

> Assume the upper row contains *one* X-ship, then the unique solution to the puzzle from Figure 5d must be respected.

From the definition of t it follows that every variable x_j is assigned only one truth value. In Figure 4 we see that if $t(x_j)$ equals *true* (*false*), then it is put on the right (left) column. Now assume that $t(x_j) =$ *true*, then the right column tally of 3 (or 4, in case x_j occurs four times in ϕ) is satisfied; the left column tally of 3 (4) is satisfied by putting the Z_j-ship on the left open slot of the gadget in Figure 3f (3e). The argument runs analogously in case $t(x_j) =$ *false*. Again the position of the Z_j-ship is uniquely determined by the arrangement of the X-ships.

Conversely, assume the Battleships puzzle of φ is solvable and that φ has m different variables. We can show by an easy inductive argument that in every solution the Y_i-ships of length $4i$ and $4i + 1$ must be put on the gadget covering region Y_i. The base case would concern the Y_m-ships of length $4m$ and $4m + 1$. These ships evidently fit the Y_m-gadget, but do not fit any other gadget due to the factor 4.

Furthermore, it is not hard to see that if a puzzle is solvable, every Z_j-ship must be put on the Z_j-gadget. As a consequence, every X-gadget contains exactly one X-ship.

The positioning of every Z_j-ship on the Z_j-gadget enforces that the truth assignment of every variable x_j is consistent throughout the clauses in φ; whereas every solved Y_i-gadget guarantees that clause C_i is true. Hence, the arrangement of X-ships encodes a unique satisfying truth assignment.

This proves the existence of a parsimonious reduction from 3-SAT to BATTLESHIPS. □

4. PROMISE VARIANT

One can reasonably object against the formalization of Battleships into BATTLESHIPS that this is not the problem we actually solve at our leisure. The magazines offering the puzzles tacitly promise that there is a *unique* solution to the puzzle printed. Intuition predicts that finding a unique solution is easier than finding out whether there is a solution at all. However, it follows as a corollary to our parsimonious reduction from 3-SAT to BATTLESHIPS that this promise variant is NP-hard as well, under randomized reductions.

We define the promise variant, PROMISE BATTLESHIPS as follows: Given a Battleships puzzle $\langle I, C, R, F\rangle$ and the promise that this puzzle has a unique solution; produce the solution.

Valiant and Vazirani (**Undefined reference**) proved that any NP-complete problem to which SAT is parsimoniously reducible has an NP-hard promise variant, under randomized reductions. Since satisfiability is parsimoniously reducible to 3-SAT; and 3-SAT is parsimoniously reducible to BATTLESHIPS, via $3, \{3, 4\}$-SAT, it follows immediately that PROMISE BATTLESHIPS is NP-hard. Hence, it is NP-complete under randomized reductions.

Furthermore, it also follows from **Undefined reference**, that the existence of a randomized polynomial-time procedure for deciding BATTLESHIPS implies that $\mathrm{NP} = \mathrm{RP}$, which is generally considered unlikely.

5. HUMANLY SOLVABLE BATTLESHIPS PUZZLES WITH UNIQUE SOLUTIONS

It might come as a surprise that also PROMISE BATTLESHIPS is NP-complete under randomized reductions, since according to our daily experience in solving a Battleships puzzle in a magazine every step is a necessary one — justified by a rule that is a logical consequence of conditions C1 to C4. For instance, to get from the left-most puzzle in Figure 1 to the puzzle in the middle and from there to the solution, one only needs the following rules (and their column analogues).

> *If* the largest ship is not put in the grid yet and of all columns and rows only row i can host it, *then* every cell (i, j) that would contain a ship segment under any allowed positioning of the largest ship, must contain a ship segment.
>
> *If* the number of ship segments in row i equals $R(i)$ and the content of cell (i, j) is unknown, *then* cell (i, j) must be filled with water.
>
> *If* the number of ship segments in row i plus the number of cells (i, j) with unknown value equals $R(i)$, *then* all undecided cells (i, j) must contain a ship segment.

The fact that the antecedents of these rules can be checked to hold in linear time, makes them locally applicable. As it appears, humans use rules that are effectively computable and that follow logically from conditions C1 to C4. We call such rules *humanly applicable*. It is exactly this property of human puzzle-solving that (automated) Battleships puzzle-composers do exploit (**Undefined reference**): One starts out with a correctly solved Battleships puzzle and iteratively removes information by inversely applying a predefined set of humanly applicable rules. The following rule, for instance, describes the inverse application of one of the above rules.

If the number of ship segments in row i equals $R(i)$ and cell (i, j) contains water, *then* change (i, j)'s water-filling into $?$.

The result of removing information by inversely applying a set of rules is a puzzle that itself is of course solvable by the same set of rules, used straightforwardly. Furthermore, note that this procedure guarantees a puzzle instance with a unique solution. This is basically due to the fact that we start out with a solution. Every time we retract information from it, we do so in such a way that it can be restored using rules that are logical consequences of conditions C1 to C4.

In fact, the difficulty indications that are provided by *Puzzelsport* (**Undefined reference**) are based on a classification of the rules used to create a puzzle. If hard rules are used to create the puzzle, hard rules will have to be used to solve it. It then is obvious that the class of Battleships puzzles that are solvable by a set of rules is a subset of the class of puzzles that are solvable by a bigger set of rules — provided that the former is not logically equivalent to the latter. Asking whether there is any effectively searchable set of humanly applicable rules that solve any instance of PROMISE BATTLESHIPS is to ask whether P equals NP, under randomized reductions.

6. SUMMARY OF RESULTS

In this contribution we proved that Battleships puzzles in general cannot be solved effectively. However, Battleships puzzles are often accompanied by their solution, for players to check their solution or to get a hint on how to arrive at the solution. We showed that the fact that such puzzles are humanly solvable cannot be due to the fact that they have unique solutions. Finally, we provided hints as to how one can generate Battleships puzzles, that are both uniquely and humanly solvable.

Although our formal results are restricted to Battleships we are convinced that our ideas and methods apply to many other one-player puzzles encountered in magazines and on the Internet.

7. ACKNOWLEDGEMENTS

I gratefully acknowledge Peter van Emde Boas, Jaap van den Herik, and the anonymous referees for their comments and remarks to improve this article. I thank Joke Hellemons, Aline Honingh, Sieuwert van Otterloo, and Yoav Seginer for helping me preparing the document.