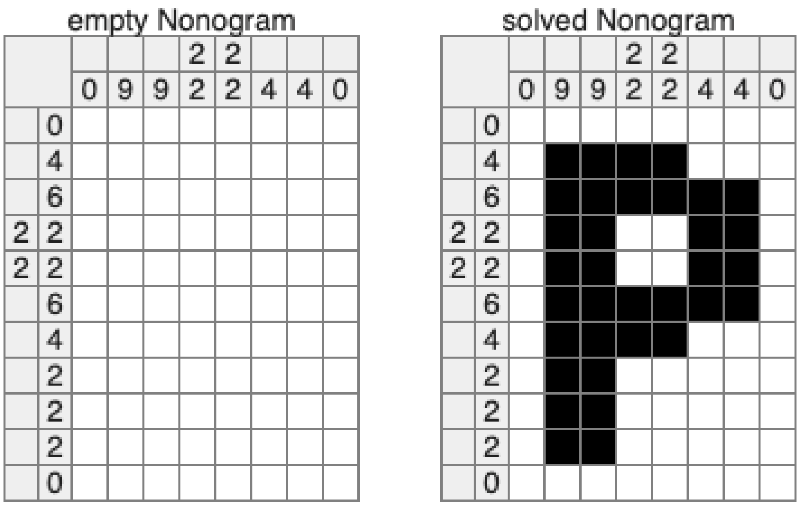
Kelby Sandick
Algorithms Extra Credit

NP Completeness in Nonograms

A particularly interesting real-world application of an NP Complete problem is in puzzles and games. One such type of logic puzzles which is included in this category are Nonograms. These are puzzles where each element of a grid is either filled in or kept blank based on clues given as numbers in each row and column. The image below (Figure 1) is an example of a nonogram puzzle in both its empty and solved states to provide clarity on how these puzzles are set up and solved.



Nonograms are particularly interesting from a computer science perspective because of the problems it poses, specifically in relation to time complexity. "The resulting (nonogram) puzzle poses a combinatorial problem that combines elements of logical reasoning with integer calculations. It can be approached using methods from combinatorial optimization, logical reasoning or both, which makes Nonograms highly suitable for educational use in Computer Science" (5).

The nonogram problem is particularly interesting because there are actually several different problems within the overarching puzzle problem. These can be simply stated as the problem of existence, the problem of solving, and the problem of uniqueness.

The problem of existence is simply, if given a nonogram, does there exist a solution. A solution here would be a grid where all squares are either filled or left unfilled, and every numerical requirement for the rows and columns. This is a problem which only has two options for the answer: yes, there does exist a solution, or no, there does not exist a solution. In order to understand the time complexity of finding whether a solution exists, we can define a nonogram as a graph, and redescribe the problem. The new definition of the problem of existence can be described as the following.

> "Given a certain AND/OR constraint graph G with the properties and restrictions as given in section 2.2, and an edge e. Is there a sequence of moves on G that will eventually reverse e, such that the final configuration is legal and each edge is reversed at most once? This problem is called the Bounded Nondeterministic Constraint Logic problem, or shortly the Bounded NCL problem" (1).

This Bounded NCL problem can be polynomially reduced to a satisfiability problem, or SAT problem. Here, specifically, the problem can be reduced into a 3-SAT problem, meaning that "each of the clauses may contain at most three literals" (1). We know that satisfiability problems are NP Complete (2). Thus, we can conclude that the Bounded NCL problem, and therefore the nonogram existence problems are NP as well.

Next, we can look at the nonogram solvability problem. Here, we can look at the Planar Bounded Nondeterministic Constraint Logic problem, or Planar Bounded NCL problem. This problem can be reduced to the nonogram solution problem (1). We know that the Planar Bounded NCL problem is NP-complete, thus the nonogram solution problem must also be NP complete.

> "We reduce the Nonogram solution problem to the Bounded NCL problem in the following way: we create a Nonogram in which we can perform pixel colourings that are equivalent to reversing edges in a planar constraint graph situations, and show that they correspond to eventually reversing an edge connected to an AND or an OR vertex" (1).

Lastly, we can look at the nonogram uniqueness problem. This problem is particularly similar to the Another Solution Problem, or ASP. From the ASP, we can then look at the Three Dimensional Matching Problem, or 3DM. The 3DM is an NP Complete problem (3). Without going into specifics of the 3DM, we know the following.

> "We show NP-completeness by reducing the 3DM problem to the Nonogram existence problem. We create a Nonogram situation in which we obtain the same number of results as in 3DM, such that we have a parsimonious reduction" (1).

Thus, we can conclude that all three sub-problems within the overarching Nonogram problem are NP Complete. After looking at all these problems, we can conclude easily that finding both the existence of a solution, the solution, and the number of solutions are all NP Complete. Thus, the problem must be solved heuristically.

> Note: Understanding nonograms and the time complexity of the nonogram problem is particularly helpful for understanding other puzzles and games. The ability to prove that the ASP is NP Complete not only provides more information about the nonogram problem, but also many other puzzle games which have NP time complexity. "An application to the field of combinatorial puzzles, we proved the ASP completeness of three popular puzzles: Slither Link, Number Place, and Fillomino. These results indicate that designing these sorts of puzzles, as well as solving, is essentially difficult. We hope that more ASP-completeness results will appear" (6). Puzzle games such as these

mentioned have a lot of mystery surrounding them in regards to whether they can be solved in polynomial time rather than being NP.

There are several different ways that people have tried to solve the nonogram problem. The two most prevalent solutions use either a genetic algorithm or a depth first search algorithm. The main issue with using genetic algorithms is that it is possible that the solution could get stuck in local optima. Using a depth first search algorithm doesn't cause this same issue, but the execution speed of this approach is extremely slow (4).

In a paper by K. Joost Batenburg and Walter A. Kosters titled, *ON THE DIFFICULTY OF NONOGRAMS*, several approaches were taken for solving nonograms. For a more detailed explanation of the techniques used to solve nonograms and collect the following data, please refer to their paper. In essence, Batenburg and Kosters employed a technique called FOURSOLVER to solve these nonograms. It included several steps, first employing The SETTLE approach was implemented by "fill(ing) in all unknown elements of a row or column that are uniquely defined by the combination of the line description and the set of known elements on that line" (5). Then, FOURSOLVER " considers four pixels at a time, that are in rectangular position and unknown so far" (5), and finishes solving the problem.

In the data shown below, 33,554,432 5×5 nonograms were tested. They were separated by difficulty based on its line descriptions. In this test, there were found to be difficulty levels from 1 to 17. The nonograms were also divided into simple and hard nonograms. Columns 2 and 3 show the number of nonograms which fall in each category. The next two categories are, respectively, the number of nonogram puzzles FOURSOLVER was able to make some progress on, and those where it could make no progress.

difficulty	simple Nonograms	(2,2)-hard Nonograms	some progress by FOURSOLVER	no progress by FOURSOLVER
1	7,776	0	0	0
2	3,409,924	5,752	72,052	124,189
3	9,367,027	56,158	299,778	576,293
4	6,514,096	156,236	701,152	1,846,395
5	3,232,762	50,796	233,590	1,621,838
6	1,350,200	31,844	145,070	1,277,062
7	633,400	9,654	42,232	590,436
8	267,632	5,396	18,660	378,354
9	115,178	1,208	5,004	153,364
10	47,584	580	2,644	93,320
11	19,972	180	888	39,904
12	7,132	120	404	24,512
13	2,864	16	0	6,892
14	728	4	0	4,664
15	216	0	0	744
16	16	0	0	496
17	4	0	0	24
18	0	0	0	16
19	0	0	0	0
total	24,976,511 74.44%	317,944 0.94%	1,521,474 4.53%	6,738,503 20.08%

The following is a similar set of data, but with nonograms of size 6x6 being tested instead of 5x5.

difficulty	simple Nonograms	(2,2)-hard Nonograms	some progress by FOURSOLVER	no progress by FOURSOLVER
1	531,441	0	0	0
2	1,892,287,503	168,790,528	806,262,697	110,274,370
3	12,508,732,323	97,411,401	381,662,356	237,190,334
4	14,306,915,587	242,390,921	988,352,708	1,927,512,716
5	9,524,244,629	201,850,110	892,360,826	3,271,038,528
6	5,110,680,772	155,867,778	694,418,286	3,243,954,226
7	2,645,426,982	84,706,478	382,676,366	2,125,778,718
8	1,295,645,826	51,037,482	221,491,162	1,550,293,616
9	666,279,114	24,030,088	102,398,912	844,903,646
10	336,135,394	13,572,368	56,564,246	560,868,894
11	173,613,082	5,915,660	23,220,212	277,760,640
12	85,805,862	2,979,724	11,498,196	169,399,812
13	42,890,084	1,206,236	4,131,016	77,735,516
14	19,823,016	583,232	1,930,888	44,335,244
15	9,241,252	211,856	601,564	18,482,368
16	3,992,532	86,032	232,984	9,667,988
17	1,750,180	29,776	71,404	3,742,560
18	683,456	11,536	25,600	1,796,320
19	275,508	3,612	6,604	628,632
20	100,160	1,156	1,716	270,200
21	36,692	352	432	85,932
22	12,204	88	128	36,172
23	3,760	36	48	11,136
24	1,176	12	12	4,248
25	332	4	12	820
26	64	0	0	256
27	0	0	0	56
28	0	0	0	16
29	0	0	0	0
total	48,625,108,931 70.76%	1,050,686,466 1.53%	4,567,908,375 6.65%	14,475,772,964 21.07%

From this data, Batenburg and Kosters were able to conclude that, "the difficulty of Nonograms follows a complex behaviour, ranging from simple Nonograms that can be found in puzzle books to highly difficult ones that can only be solved by an exhaustive search, handling the underlying NP-hard combinatorial problem" (5).

Based on this the conclusion of Batenburg and Kosters, we can now understand that both the difficulty of Nonograms and the existance one or more solutions to a given nonogram are both NP Complete.

Sources
(1) Complexity and solvability of Nonogram puzzles
http://fse.studenttheses.ub.rug.nl/15287/1/Master_Educatie_2017_RAOosterman.pdf
(2) Lecture 20- Satisfiability is NP Complete
http://www.cs.columbia.edu/~aho/cs3261/Lectures/L20-Satisfiability.html
(3) NP-completeness Results for Nonograms via Parsimonious Reductions
https://pdfs.semanticscholar.org/1bb2/3460c7f0462d95832bb876ec2ee0e5bc46cf.pdf
(4) An Efficient Algorithm for Solving Nonograms
https://www.researchgate.net/publication/220204911_An_efficient_algorithm_for_solving_nonograms
(5) ON THE DIFFICULTY OF NONOGRAMS
https://liacs.leidenuniv.nl/~kosterswa/nonodec2012.pdf
(6) COMPLEXITY AND COMPLETENESS OF FINDING ANOTHER SOLUTION AND ITS APPLICATION TO PUZZLES
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.6324&rep=rep1&type=pdf