Operations Research Letters 38 (2010) 582–584

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

How ∗not∗ to solve a Sudoku

Adriana F. Gabor [a], Gerhard J. Woeginger [b,∗]

[a] Department of Econometrics, Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

[b] Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 10 February 2010
Accepted 30 August 2010
Available online 9 September 2010

Keywords:
Computational complexity
Matching problem
Assignment problem

ABSTRACT

We prove the NP-hardness of a consistency checking problem that arises in certain elimination strategies for solving Sudoku-type problems.

1. Introduction

A Sudoku is a combinatorial number-placement puzzle where the objective is to fill a 9 × 9 grid with digits so that (i) each column, (ii) each row, and (iii) each of the nine 3 × 3 sub-grids that compose the grid contains all the digits from 1 to 9. The puzzle provides a partially completed grid, and the goal is to extend this to the full grid. Sudoku puzzles and their variations have become very popular in recent years. We refer the reader for instance to [1,2] for mathematical results and observations on mathematical solution approaches to Sudokus. Provan [4] considers the following mathematical model of Sudoku-type problems.

- A finite ground set $\mathcal{S}$ models the grid squares.
- A finite color set $\mathcal{C}$ models the digits from 1 to 9.
- A family $\mathcal{B}$ of subsets of $\mathcal{S}$ models the rows, columns, and 3 × 3 sub-grids. These subsets are called *blocks*, and they all have the same cardinality as $\mathcal{C}$.
- A function $C : \mathcal{S} \to 2^{\mathcal{C}}$ models the initial situation of the Sudoku.

The function C specifies for every element $s \in \mathcal{S}$ the set $C(s)$ of colors that can feasibly be assigned to s. If $C(s)$ contains a single element, then the contents of the grid square s is fixed right from the beginning. If $C(s) = \mathcal{C}$, then the grid square s is initially empty, and any color in $\mathcal{C}$ may be assigned to it. The goal is to find a coloring $\gamma : \mathcal{S} \to \mathcal{C}$ of the elements such that $\gamma(s) \in C(s)$ for all $s \in \mathcal{S}$, and such that in every block $B \in \mathcal{B}$ every color is assigned to exactly one element.

A common approach to Sudoku puzzles are *elimination strategies* that step by step eliminate colors from the feasible color sets $C(s)$, until each set has been cut down to a single element. Provan [4] investigates a particular family of elimination strategies which he calls *one-block strategies*. Every single step of a one-block strategy can eliminate a single color from a single set $C(s)$ by drawing conclusions based on a single block. Hence one-block strategies are centered around the following algorithmic problem.

Problem: One-Block Elimination
Input: A block $B \in \mathcal{B}$; color sets $C(s)$ for all $s \in B$; a fixed element $s_0 \in B$ and a fixed color $c_0 \in C(s_0)$.
Question: Does there exist a coloring $\gamma : B \to \mathcal{C}$ such that every color in $\mathcal{C}$ is assigned to exactly one element in B, such that $\gamma(s) \in C(s)$ for all $s \in B$, and such that $\gamma(s_0) = c_0$?

Whenever a one-block strategy detects a NO instance of One-Block Elimination, it makes progress by eliminating the color c_0 from set $C(s_0)$. According to Provan [4] this simple strategy solves about 90% of all Sudoku puzzles; for the remaining 10% it reduces the search space considerably.

It is easy to see that the problem One-Block Elimination is a bipartite perfect matching problem, and hence is well behaved and solvable in polynomial time. Provan [4] derives the following beautiful characterization result from this: A Sudoku can be fully solved by a one-block strategy if and only if a certain underlying linear equation system has a non-negative solution over the real numbers. Provan poses the open question whether there also is a good description of *two-block strategies* that are centered around the following problem.

Problem: Two-Block Elimination
Input: Two blocks $B_1, B_2 \in \mathcal{B}$; color sets $C(s)$ for all $s \in B_1 \cup B_2$; a fixed element $s_0 \in B_1 \cup B_2$ and a fixed color $c_0 \in C(s_0)$.

∗ Corresponding author.
E-mail address: gwoegi@win.tue.nl (G.J. Woeginger).

0167-6377/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2010.09.001

QUESTION: Does there exist a coloring $\gamma : B_1 \cup B_2 \to \mathcal{C}$ such that every color in $\mathcal{C}$ is assigned to exactly one element in B_1 and to exactly one element in B_2, such that $\gamma(s) \in C(s)$ for all $s \in B$, and such that $\gamma(s_0) = c_0$?

In this technical note we will prove that TWO-BLOCK ELIMINATION, in fact, is an intractable problem.

Theorem 1. *Problem* TWO-BLOCK ELIMINATION *is NP-complete.*

This hints at a negative answer to Provan's open question: Whenever NP-hardness rears its ugly head, there is little hope for beautiful characterizations and fast algorithms.

2. The hardness proof

We prove the NP-hardness of the problem TWO-BLOCK ELIMINATION by a reduction from the following variant of the satisfiability problem (see [3]).

PROBLEM: SATISFIABILITY
INPUT: A set $X = \{x_1, \dots, x_n\}$ of n logical variables; a set $K = \{k_1, \dots, k_m\}$ of m clauses over X where every variable occurs twice as negated and once as unnegated literal.
QUESTION: Is there a truth assignment for X that satisfies all clauses in K?

Note that [3] only says that the satisfiability problem remains NP-complete if each variable (negated or unnegated) occurs at most three times. It is straightforward to add some clauses and to negate some variables without changing the nature of the problem.

Now consider an arbitrary instance of SATISFIABILITY. Without loss of generality we assume that every clause contains at least two literals; this implies $m \le 3n/2$. We construct a corresponding instance of TWO-BLOCK ELIMINATION that consists of the following colors, elements, and color sets:

- For every variable x_i we introduce a corresponding color $c(i)$.
- If the unnegated literal x_i occurs in clause k_j then we create a corresponding color $c(x_i, k_j)$, and if the negated literal $\overline{x_i}$ occurs in clause k_j then we create the corresponding color $c(\overline{x_i}, k_j)$.
- Finally, there is a special color c_0.

Since every variable occurs (negated or unnegated) in exactly three clauses, the overall number of colors in $\mathcal{C}$ equals $4n + 1$.

- For every clause k_j we introduce a corresponding clause element $s(k_j)$ that belongs to both blocks B_1 and B_2. The color set of $s(k_j)$ contains all colors $c(\ell, k_j)$ for which ℓ is a literal in clause k_j.
- If the unnegated literal x_i occurs in clause k_j then we create a corresponding literal element $s(x_i, k_j)$ that belongs to both blocks B_1 and B_2. The color set of element $s(x_i, k_j)$ consists of color $c(i)$ and color $c(x_i, k_j)$.
- If the negated literal $\overline{x_i}$ occurs in clause k_j then we create a corresponding literal element $s(\overline{x_i}, k_j)$. Every negated literal occurs in two clauses, and one of its corresponding literal elements belongs to $B_1 - B_2$ and the other one belongs to $B_2 - B_1$. The color set of element $s(\overline{x_i}, k_j)$ consists of the colors $c(i)$ and $c(\overline{x_i}, k_j)$.
- Finally, there is a special element s_0 with color set $C(s_0) = \{c_0\}$, and there are $4n - 2m$ dummy elements with color set $\mathcal{C}$. The special element belongs to $B_1 \cap B_2$, and the dummy elements are equally divided between $B_1 - B_2$ and $B_2 - B_1$.

Note that both blocks contain $4n + 1 = |\mathcal{C}|$ elements, and that their intersection $B_1 \cap B_2$ contains $n + m + 1$ elements.

Lemma 2. *If the* SATISFIABILITY *instance has answer YES, then the constructed instance of* TWO-BLOCK ELIMINATION *has answer YES.*

Proof. Consider a satisfying truth assignment t. If $t(x_i)$ is TRUE then color the (unique) corresponding unnegated literal element $s(x_i, k_j)$ by color $c(i)$, and color both the corresponding negated literal elements $s(\overline{x_i}, k_j)$ by their corresponding color $c(\overline{x_i}, k_j)$. Similarly, if $t(x_i)$ is FALSE then color the (unique) corresponding unnegated literal element $s(x_i, k_j)$ by its corresponding color $c(x_i, k_j)$, and color both the corresponding negated literal elements $s(\overline{x_i}, k_j)$ by color $c(i)$. Note that for a true variable x_i this leaves the color $c(x_i, k_j)$ free, and that for a false variable x_i this leaves the two colors $c(\overline{x_i}, k_j)$ (with two distinct clauses k_j) free.

Since every clause contains at least one true literal, these free colors can be used to feasibly color all clause elements. The special element s_0 is colored by c_0. Finally, the $2n - m$ dummy elements in each block B_r $(r = 1, 2)$ are colored by the $2n - m$ remaining colors that have not been used in this block so far. □

Lemma 3. *If the* TWO-BLOCK ELIMINATION *instance has answer YES, then the* SATISFIABILITY *instance has answer YES.*

Proof. Consider a feasible coloring. Suppose for the sake of contradiction that there is a variable x_i and two clauses k_a and k_b such that the clause elements $s(k_a)$ and $s(k_b)$ are colored by colors $c(x_i, k_a)$ and $c(\overline{x_i}, k_b)$, respectively. Then the color $c(x_i, k_a)$ cannot be used for the element $s(x_i, k_a)$, and hence this element must be colored by $c(i)$. Similarly, the color $c(\overline{x_i}, k_b)$ cannot be used for element $s(\overline{x_i}, k_b)$ and also that element must be colored by the same color $c(i)$. Since $s(x_i, k_a)$ lies in both blocks, color $c(i)$ is used twice in the same block. That is the desired contradiction.

This observation suggests the following truth assignment: Whenever a clause element $s(k_j)$ is colored by $c(x_i, k_j)$ with $x_i \in k_j$, then variable x_i is set to TRUE. Whenever a clause element $s(k_j)$ is colored by $c(\overline{x_i}, k_j)$ with $\overline{x_i} \in k_j$, then variable x_i is set to FALSE. All remaining variables are set to TRUE. This yields a valid truth assignment that by definition satisfies all clauses. □

This completes the proof of Theorem 1, and TWO-BLOCK ELIMINATION indeed is a hard problem.

3. A more realistic variant of two-block elimination

There is something extremely dissatisfying about the NP-hardness construction in the preceding section: Every Sudoku puzzle in the real world has a solution. Whenever we invoke an algorithm for TWO-BLOCK ELIMINATION while solving a Sudoku, we want to know whether by assigning color c_0 to element s_0 we jump from a solvable situation into an unsolvable situation. But the instance constructed above does not reflect this behavior: Color c_0 is the only feasible color for element s_0, and the solvability of the instance does not depend at all on the way we color s_0. In the rest of this paper, we will show how to slightly modify our construction so that the coloring decision for s_0 becomes crucial.

PROBLEM: REAL-WORLD TWO-BLOCK ELIMINATION
INPUT: An instance of the standard version of TWO-BLOCK ELIMINATION plus a second color $c'_0 \in C(s_0)$ with $c'_0 \ne c_0$. This instance has a feasible coloring $\gamma' : B_1 \cup B_2 \to \mathcal{C}$ with $\gamma'(s_0) = c'_0$.
QUESTION: Does there also exist another feasible coloring $\gamma : B_1 \cup B_2 \to \mathcal{C}$ with $\gamma(s_0) = c_0$?

Our NP-hardness proof for REAL-WORLD TWO-BLOCK ELIMINATION closely follows the arguments in the preceding section. Hence we will only sketch how to modify these arguments.

We construct a gadget that relies on a directed out-tree $T = (Y, A)$. The vertex set Y consists of $3m$ vertices u_i, v_i, w_i with $1 \le i \le m$ (where m is the number of clauses in K). For $1 \le i \le m$, the tree contains the arcs $u_i \to v_i$ and $u_i \to w_i$, and for $1 \le i \le m-1$, the tree furthermore contains the arc $v_i \rightarrow u_{i+1}$. This out-tree is rooted at u_1, and all vertices $y \neq u_1$ have a unique father. We define for every vertex $y \in Y$ a corresponding element $s(y)$ and a corresponding color $c(y)$. Elements $s(u_i)$ are put into $B_1 \cap B_2$, elements $s(v_i)$ into $B_1 - B_2$, and elements $s(w_i)$ into $B_2 - B_1$. The element $s(u_1)$ corresponding to the root has color set $C(s(u_1)) = \{c_0, c(u_1)\}$. For a non-root vertex y with father z, we define $C(s(y)) = \{c(y), c(z)\}$.

We add the new elements and colors from this out-tree gadget to the old construction. Furthermore we create a brand new color c_0'. The color set for the special element s_0 is updated to $C(s_0) = \{c_0, c_0'\}$. The color set of the jth clause element $s(k_j)$ receives $c(w_j)$ as additional new color. In order to account for the $3m + 1$ new colors we also add $m + 1$ new dummy elements to each of $B_1 - B_2$ and each of $B_2 - B_1$ (that all can be colored by any possible color). Then both blocks have $3m + 4n + 2$ elements, and there are $3m + 4n + 2$ colors available. Here are two useful observations:

(i) Assume that element s_0 is colored by c_0. Then element $s(u_1)$ must be colored by color $c(u_1)$, and an easy inductive argument shows that all other elements $s(y)$ in the gadget must be colored by the color $c(y)$ that corresponds to the same vertex y. In this case all colors $c(y)$ with $y \in Y$ are absorbed by the gadget elements. In particular, the colors $c(w_j)$ cannot be used to color the clause elements, and the arguments in Lemmas 2 and 3 hold as before.

(ii) Assume that element s_0 is colored by c_0'. Then element $s(u_1)$ may be colored by c_0, and for every non-root vertex y with father z we may color $s(y)$ by $c(z)$. This yields a feasible coloring for the gadget that leaves the m colors $c(w_i)$ with $1 \leq i \leq m$ free for the m clause elements. It is straightforward to extend this coloring to a feasible coloring of the entire instance.

Summarizing, the new instance possesses the feasible coloring described in (ii). Furthermore by (i) it is NP-hard to decide whether assigning color c_0 to element s_0 makes the situation to jump from solvable to unsolvable. This completes our argument.

Theorem 4. *Problem* Real-world Two-Block Elimination *is NP-complete.*

Acknowledgements

This research has been supported by the Netherlands Organisation for Scientific Research (NWO), grant 639.033.403, by DIAMANT (an NWO mathematics cluster), and by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society).

References

[1] D. Berthier, The Hidden Logic of Sudoku, Lulu.com Publishers, Morrisville, 2007.
[2] A.E. Brouwer, Sudoku puzzles and how to solve them, Nieuw Archief voor Wiskunde (Vijfde Serie) 7 (2006) 258–263.
[3] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[4] J.S. Provan, Sudoku: strategy versus structure, American Mathematical Monthly 116 (2009) 702–707.