Available online at www.sciencedirect.com
ScienceDirect

AKCE International Journal of Graphs and Combinatorics 13 (2016) 272–282

AKCE International Journal of Graphs and Combinatorics
www.elsevier.com/locate/akcej

Reducing the generalised Sudoku problem to the Hamiltonian cycle problem

Michael Haythorpe

School of Computer Science, Engineering and Mathematics, Flinders University, Bedford Park, 5042, Australia

Received 1 April 2016; received in revised form 16 September 2016; accepted 4 October 2016
Available online 27 October 2016

Abstract

The generalised Sudoku problem with N symbols is known to be NP-complete, and hence is equivalent to any other NP-complete problem, even for the standard restricted version where N is a perfect square. In particular, generalised Sudoku is equivalent to the, classical, Hamiltonian cycle problem. A constructive algorithm is given that reduces generalised Sudoku to the Hamiltonian cycle problem, where the resultant instance of Hamiltonian cycle problem is sparse, and has $O(N^3)$ vertices. The Hamiltonian cycle problem instance so constructed is a directed graph, and so a (known) conversion to undirected Hamiltonian cycle problem is also provided so that it can be submitted to the best heuristics. A simple algorithm for obtaining the valid Sudoku solution from the Hamiltonian cycle is provided. Techniques to reduce the size of the resultant graph are also discussed.

Keywords: Sudoku; NP-complete; Reduction; Hamiltonian cycle problem

1. Introduction

The *generalised Sudoku problem* is an NP-complete problem which, effectively, requests a Latin square that satisfies some additional constraints. In addition to the standard requirement that each row and column of the Latin square contains each symbol precisely once, Sudoku also demands *block constraints*. If there are N symbols, the Latin square is of size $N \times N$. If N is a perfect square, then the Latin square can be divided into N regions of size $\sqrt{N} \times \sqrt{N}$, called *blocks*. Then the block constraints demand that each of these blocks also contain each of the symbols precisely once. Typically, the symbols in a Sudoku puzzle are simply taken as the natural numbers 1 to N. In addition, Sudoku puzzles typically have fixed values in some of the cells, which dramatically limits the number of valid solutions. If the fixed values are such that only a unique solution remains, the Sudoku puzzle is said to be *well-formed*.

The standard version where $N = 9$ has, in recent years, become a common form of puzzle found in newspapers and magazines the world over. Although variants of the problem have existed for over a century, Sudoku in its current format is a fairly recent problem, first published in 1979 under the name Number Place. The name Sudoku only came

Peer review under responsibility of Kalasalingam University.
E-mail address: michael.haythorpe@flinders.edu.au.

http://dx.doi.org/10.1016/j.akcej.2016.10.001

into existence in the 1980s. In 2003, the generalised Sudoku problem was shown to be ASP-complete [1], which in turn implies that it is NP-complete. Hence, it is theoretically as difficult as any problems in the set $\mathcal{NP}$ of decision problems for which a positive solution can be certified in polynomial time. Note that although there are more general versions variants of Sudoku (such as rectangular versions), the square variant described above where N is a perfect square suffices for NP-completeness. Hence, for the remainder of this manuscript, it will be assumed that we are restricted to considering the square variant.

Since being shown to be NP-complete, Sudoku has subsequently been converted to various NP-complete problems, most notably constraint satisfaction [2], boolean satisfiability [3] and integer programming [4]. Another famous NP-complete problem is the *Hamiltonian cycle problem* (HCP), which is defined as follows. For a simple graph (that is, one containing no self-loops or multi-edges) containing vertex set V and edge set $E : V \to V$, determine whether any simple cycles containing all vertices in V exist in the graph. Such cycles are called *Hamiltonian cycles*, and a graph containing at least one Hamiltonian cycle is called *Hamiltonian*. Although HCP is defined for directed graphs, in practice most heuristics that actually solve HCP are written for undirected graphs.

Since both Sudoku and HCP are NP-complete, it should be possible to reduce Sudoku to HCP. In this manuscript, a constructive algorithm that constitutes such a reduction is given. The resultant instance of HCP is a sparse graph of order $O(N^3)$. If many values are fixed, it is likely that the resultant graph can be made smaller by clever graph reduction heuristics; to this end, we apply a basic graph reduction heuristic to two example Sudoku instances to investigate the improvement offered.

It should be noted that reductions of NP-complete problems to HCP is an interesting but still largely unexplored field of research. Being one of the classical NP-complete problems (indeed, one of the initial 21 NP-complete problems described by Karp [5]), HCP is widely studied and several very efficient algorithms for solving HCP exist. HCP is also an attractive target problem in many cases because the resultant size of the instance is relatively small by comparison to other potential target problems. Indeed, the study of which NP-complete problems provide the best target frameworks for reductions is an ongoing field of research. It seems that certain NP-complete problems work better than others as target problems for reductions, in the sense that more problems can be efficiently reduced to them. For example, it is known that boolean satisfiability can be reduced to HCP in such a way that the size of the resultant HCP instance is a linear function of the size of the original instance. However, no such linearly-growing reduction is known for HCP to boolean satisfiability. Hence, any problem which can be reduced to boolean satisfiability with linear growth can also be reduced to HCP with linear growth, but the converse is not true. For more on this topic, as well as examples of other reductions to HCP, the interested reader is referred to [6–10].

In addition to direct HCP methods such as those by Chalaturnyk [11] or Baniasadi et al. [12], HCP can also be solved relatively efficiently (in the average case) even by algorithms designed for other NP-complete problems such as those for travelling salesman problem (see Applegate et al. [13] or Helsgaun [14]) and boolean satisfiability (see Johnson [15]). It appears that truly difficult instances of HCP are quite rare and typically need to be constructed specifically to be difficult (e.g. see Haythorpe [16,17]), and for the vast majority of cases, the problem can be solved efficiently by the existing heuristics. In addition, we also need to consider the case where no solution exists. For most NP-complete problems, such cases are given very little attention, as in general such a result cannot be confirmed in polynomial time; however, for HCP, some preliminary results exist and the subject is a topic of ongoing research (e.g. see Newcombe [18] and Filar et al. [19]).

2. Conversion to HCP

At its core, a Sudoku problem with N symbols (which we will consider to be the natural numbers from 1 to N) has three sets of constraints to be simultaneously satisfied.

1. Each of the N blocks must contain each number from 1 to N precisely once.
2. Each of the N rows must contain each number from 1 to N precisely once.
3. Each of the N columns must contain each number from 1 to N precisely once.

The variables of the problem are the N^2 cells, which can each be assigned any of the N possible values, although some of the cells may have fixed values depending on the instance.

In order to cast an instance of Sudoku as an instance of Hamiltonian cycle problem, we need to first encode every possible variable choice as a subgraph. The idea will be that traversing the various subgraphs in certain ways will