FI MU

Faculty of Informatics
Masaryk University Brno

Human Problem Solving: Sudoku Case Study

by

Radek Pelánek

FI MU Report Series **FIMU-RS-2011-01**

 January 2011

Publications in the FI MU Report Series are in general accessible via WWW:

`http://www.fi.muni.cz/reports/`

Further information can be obtained by contacting:

**Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic**

Human Problem Solving: Sudoku Case Study*

Radek Pelánek
Faculty of Informatics
Masaryk University Brno, Czech Republic

January 31, 2011

Abstract

We discuss and evaluate metrics for difficulty rating of Sudoku puzzles. The correlation coefficient with human performance for our best metric is 0.95. The data on human performance were obtained from three web portals and they comprise thousands of hours of human solving over 2000 problems. We provide a simple computational model of human solving activity and evaluate it over collected data. Using the model we show that there are two sources of problem difficulty: complexity of individual steps (logic operations) and structure of dependency among steps. Beside providing a very good Sudoku-tuned metric, we also discuss a metric with few Sudoku-specific details, which still provides good results (correlation coefficient is 0.88). Hence we believe that the approach should be applicable to difficulty rating of other constraint satisfaction problems.

This technical report is a full version of a paper presented at the 24th Florida Artificial Intelligence Research Society Conference.

1 Introduction

The general theme of this work is human problem solving [19]. Particularly, we focus on the study of problem difficulty: What determines which problems are difficult for humans? Beside giving us insight into human cognition and thinking, the study of this issue has important applications in human-computer collaboration and training of problem solving skills, e.g., for developing intelligent tutoring systems [1, 2, 3].

*This work is supported by GA ČR grant no. P202/10/0334.

1.1 Difficulty of Problem Solving

We study problem difficulty of one particular problem – Sudoku puzzle. Our specific goal is the following: *Provide a difficulty rating metric for Sudoku puzzle, that achieves as high correlation with human performance (measured by time) as possible.* This goal has direct applications – such metrics are heavily used since Sudoku is currently very popular and even commercially important [12], and difficulty rating of puzzles is one of the key things which influence user's experience of puzzle solving.

Despite the straightforwardness of our goal and its direct applicability, there is no easily applicable theory that could be used to guide the development of difficulty rating metrics. Currently used Sudoku metrics are usually built in an ad-hoc manner, they are not properly evaluated and their merits are not clear. In general there has been only little research dealing with the issue of problem difficulty; results are available only for few specific puzzles, e.g., Tower of Hanoi (and its izomorphs) [10], Chinese rings [11], 15-puzzle [18], traveling salesman problem [4], and Sokoban puzzle [8].

The aim of this work goes beyond the specific study of Sudoku puzzle. We would like to raise the interest in the study of problem difficulty, for example by showing that extensive and robust data for study are easily available on the Internet. In this way we would like to contribute towards a theory of difficulty in human problem solving.

1.2 Sudoku and Constraint Satisfaction Problems

Sudoku is a well-known number placement puzzle: for a partially filled 9×9 grid, the goal is to place numbers 1 to 9 to each cell in such a way that in each row, column, and 3×3 sub-grid, each number occurs exactly once. Sudoku has been subject to many research studies, particularly with respect to its mathematical and algorithmic properties, e.g., enumerating possible Sudoku grid [5], NP-completeness of generalized version of Sudoku [21], use of constraint propagation [20, 15] or genetic algorithms [16] for solving the puzzle, or algorithms for generating puzzles [17, 6]. Recently, also psychological aspects of the puzzle has been studied [13].

We focus on the Sudoku puzzle for several reasons. The Sudoku puzzle has very simple rules, which makes it amenable to analysis. Thanks to its current popularity we can easily obtain large scale data on human solving activity. Sudoku is also a member of an important class of constraint satisfaction problems (CSP). The class of constraint satisfaction problems contains many other puzzles and also many real life problems

(e.g., timetabling, scheduling). Although we use data for the Sudoku puzzle, our goal is to make the analysis and difficulty metrics as general as possible, so that the results are potentially applicable to other CSPs.

1.3 Data from Internet

Due to the popularity of the Sudoku puzzle we have been able to obtain data capturing hundreds of thousands hours of human problem solving activity (approximately 2000 puzzles, hundreds of human solvers for each puzzle). This means that we have data several orders of magnitude more extensive than the usual data used in study of human problem solving – most previous research is based on data based on tens or hundreds of hours of human problem solving activity (usually about 20 people and 5 puzzles). Even though this way of data collection has its disadvantages (e.g., lack of direct control over participants), we show that thanks to the scale of the "experiment", the data are robust and applicable for research purposes.

1.4 Contributions

Difficulty rating of Sudoku puzzles is, of course, not a novel problem. The issue of Sudoku difficulty rating is widely discussed among Sudoku players and developers, but it has not been subject to serious scientific evaluation. Current rating algorithms are based mainly on personal experiences and ad-hoc tuning. There are several research papers which discuss methods for difficulty rating [20, 16, 7]; however, these works study the correlation of proposed metric with the difficulty rating provided by the puzzle source (usually a newspaper), not with the data on human performance. Such analysis is not very meaningful since the rating provided in puzzle sources is just another rating provided by a computer program (nearly all published puzzles are generated and rated by a computer). The only work that we are aware of and that uses data on real human performance is the brief report by Leone et al. [14].

The results of our study show that there are two main aspects of problem difficulty. The first is the complexity of individual steps (logic operations) involved in solving the problem – this is the usual approach used for rating Sudoku puzzles. We show that there is also a second aspect that has not yet been utilized for difficulty rating – the structure of dependency among individual steps, i.e., whether steps are independent (can be applied in parallel) or whether there are dependent (must be applied sequen-

tially). We provide a simple general model that captures both of these aspects. We show that even with rather simple rating metric with little Sudoku-specific details we can obtain correlation coefficient with human performance 0.88. By combination with previously proposed Sudoku-specific metrics we can obtain correlation coefficient 0.95.

2 Sudoku and Constraint Satisfaction Problems

The Sudoku puzzle is a special case of a more general type of problems called constraint satisfaction problems (CSP). In this section we describe both the general CSP and specific Sudoku problem. We also discuss basic techniques for solving these problems.

2.1 Constraint Satisfaction Problems

Constraint satisfaction problem is given by a set of variables $X = \{x_1, \ldots, x_n\}$, a set of domains of variables $\{D_1, \ldots, D_n\}$ (we consider only finite domains), and a set of constraints $\{C_1, \ldots, C_m\}$. Each constraint involves some subset of variables and specifies allowed combinations of variable values (usually given in a symbolic form, e.g., $x_1 \neq x_2$).

A solution of a constraint satisfaction problem is an assignment of values to all variables such that all constraints are satisfied. The class of CSPs contains many puzzles (e.g., eight queen problem, cryptarithmetic puzzle) as well as many important practical problems (map coloring problems, timetabling problems, transportation scheduling). The general CSP is NP-complete.

2.2 Sudoku Puzzle

Sudoku puzzle is a grid of 9×9 cells, which are divided into nine 3×3 sub-grids, partially filled with numbers 1 to 9. The solution of the puzzle is a complete assignment of numbers 1 to 9 to all cells in the grid such that each row, column and sub-grid contains exactly one occurrence of each number from the set $\{1, \ldots, 9\}$. Sudoku puzzle is well posed, if it admits exactly one solution. We study only well-posed puzzles.

Sudoku puzzle can be easily generalized for any grid size of $n^2 \times n^2$ and values from 1 to n^2 [20]. Moreover, there are many variants of Sudoku which use non-regular sub-grids (e.g, pentomino), or additional constraint (e.g., arithmetic comparison of val-

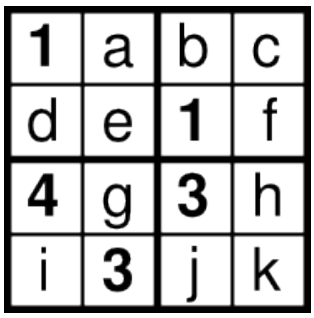

Figure 1: Example of a 4×4 Sudoku puzzle, empty cells are denoted as variables.

ues). In this work we use 4×4 Sudoku puzzle (Figure 1) as a small running example, otherwise we consider solely the classical 9×9 Sudoku puzzles.

Sudoku can be easily expressed as a constraint satisfaction problem [20]. Consider an example of a 4×4 Sudoku given in Figure 1; each cell corresponds to one variable, the domain of each variable is a set $\{1, 2, 3, 4\}$ and the constraints express non-equality of variables and constants in same row, column or sub-grid, e.g., for variable j we have the following constraints: $j \neq b, j \neq h, j \neq i, j \neq k, j \neq 3, j \neq 1$. The constraints can be expressed more compactly using "all different" constraint, which is an often used type of constraint even in many practical applications [20].

2.3 Backtracking

The brute-force approach to solving CSP is called backtracking. The backtracking search starts with an empty variable assignment and tries to find a solution by assigning values to variables one by one. Whenever it finds a violation of a constraint, it backtracks. In this way the search explores the tree of feasible partial assignments. Figure 2 shows the search tree for a sample 4×4 Sudoku.

The run time of a backtracking algorithm grows exponentially with the number of variables. Nevertheless, classical 9×9 Sudoku can be easily solved by computer using the backtracking search. For humans, however, this is not a favoured approach. As can be seen in Figure 2, even for a small 4×4 Sudoku, some branches in the search can be quite long. For humans, systematic search is laborious, error-prone, and definitely not entertaining.

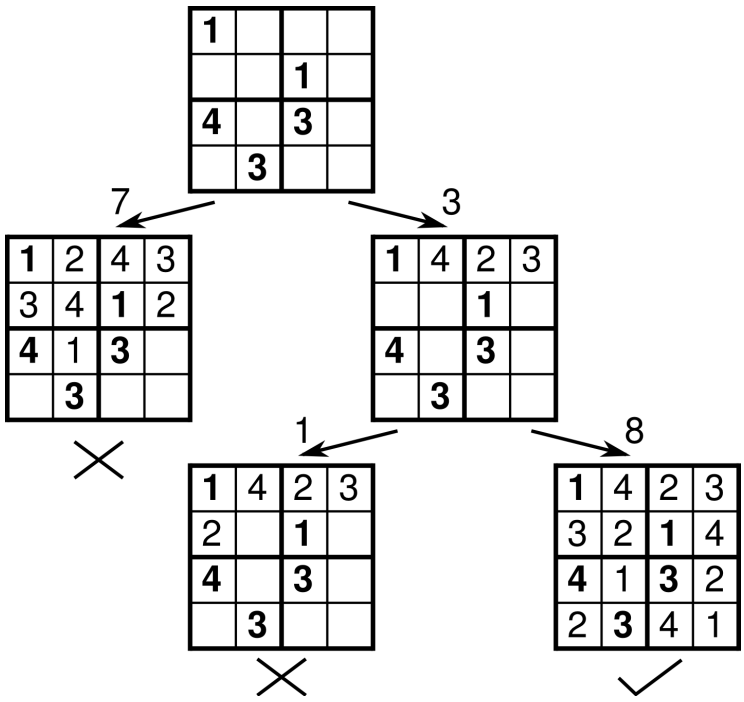

Figure 2: Comprised version of a search tree of a backtracking algorithm on a 4×4 Sudoku puzzle. Numbers over arrows indicate number of steps (filled numbers) without branching.

2.4 Constraint Propagation

Another approach to solving CSP is to find the value of (some) variables by reasoning about constraints. For each variable x_i we define a current candidate set – a set of such values that do not lead to direct violation of any constraint (see Figure 3). By reasoning about candidate sets and constraints we can often derive solution without any search.

For our example from Figure 1, it is easy to see that the value of variable i must be 2, because other constraints restrict the domain of the variable to a single value. Thus we can assign the value to the variable without any search. Using this partial assignment we can straighten constraints on other variables (and show, for example, that also h has to be 2).

Constraint propagation is not guaranteed to find a solution, but it may be more efficient than backtracking search and can also be combined combined with backtracking search to produce superior results. We are interested in constraint propagation particularly because this is the natural way how humans try to solve CSPs.

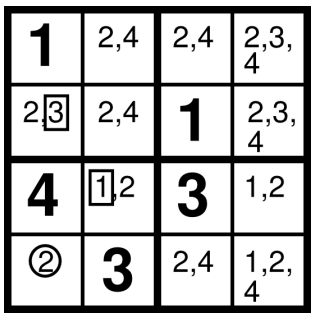

Figure 3: A sample 4×4 Sudoku puzzle with enumerated candidate sets. Circle marks naked single, rectangle marks hidden single.

Let us consider specifically the case of Sudoku puzzle. Human solving of Sudoku proceeds by sequence of steps in which (correct) values are filled into cells. Two basic techniques directly correspond to the rules of the puzzle (see also Figure 3).

Naked single technique (also called singleton, single value, forced value, exclusion principle): For a given cell there is only one value that can go into the cell, because all other values occur in row, column or sub-grid of the cell (any other number would lead to a direct violation of rules).

Hidden single technique (also called naked value, inclusion principle): For a given unit (row, column or sub-grid) there exists only one cell which can contain a given value (all other placements would lead to a direct violation of rules).

Sudoku problems solvable by iteration of these two techniques are further denoted as "simple Sudoku". Most of the publicly used puzzles which are ranked as easy or mild are simple Sudokus. There exists many advanced techniques, such as pointing, chaining, naked and hidden pairs (see, e.g., Sudoku Explainer [9]), but we do not elaborate on these techniques in order to keep Sudoku-specific details minimized.

3 Data on Human Sudoku Solving

In this section we describe the data on human problem solving activity that we use for evaluation of difficulty metrics.

3.1 Data Sources

For obtaining data on human problem solving we exploited the current popularity of on-line puzzle solving and particularly the popularity of Sudoku puzzle. The data were obtained from three Sudoku web portals. The individual data entries obtained from such a source are of worse quality than data from controlled laboratory experiments (e.g., it is probable that some solvers where distracted while solving a puzzle). However, in this way we can obtain significantly more data (by several orders of magnitude) than is feasible from any laboratory experiment. As we demonstrate, the data are very robust and thus can be used for evaluation.

The first dataset is from the web portal `fed-sudoku.eu` (all puzzles from the year 2008). We have in total 1089 puzzles, the mean number of solvers is 131 per puzzle. For each solution we have the total time taken to complete the puzzle. Each solution is identified by a user login, i.e., we can pair solutions by the same user. Most users solved many puzzles, i.e., the data reflect puzzle solving by experienced solvers. The server provides listings of results and hall of fame. Thus although there is no control over the users and no monetary incentives to perform well, users are well motivated.

The second dataset is from the web portal `sudoku.org.uk`. The data are from years 2006-2009; there was one puzzle per day. In this case we have only summary data provided by the server: total number of solvers (the mean is 1307 solvers per puzzle) and the mean time to solve the puzzle (no data on individual solvers). We have data about 1331 puzzles, but because of the significant improvement of human solvers during years 2006 and 2007 we have used for the evaluation only 731 puzzles (see the discussion bellow).

The third dataset is from the web portal `czech-sudoku.com`. This web portal was used in a different way from the other two. The portal provides not just the time to solve the puzzle, but also the data record of each play. More specifically, each move (filling a number) and time to make the move are stored. From this portal we analyzed these detailed records for about 60 users and 15 puzzles.

3.2 Analysis of Data

As a measure of problem difficulty for humans we use the mean solution time. Since our data do not come from a controlled experiment and mean is susceptible to outlier values, it is conceivable that this measure of difficulty is distorted. To get confidence

in this measure of problem difficulty, we analyzed the robustness of the detailed data from `fed-sudoku.eu` portal. In addition to mean we also computed median time, median time for active solvers (those who solved more than 900 puzzles), and mean of normalized times (normalized time is the ratio of solution time to mean solution time of the user). All these metrics are highly correlated (in all cases $r > 0.93$). Thus it seems that the particular way to choose the measure of human performance is not very important. In fact, the use of median, which is a more stable metric than mean, leads to slightly better results for all studied metrics. Nevertheless, for consistency we use as a difficulty measure the mean (for the `sudoku.org.uk` we do not have any other data). Moreover, the relative ordering of techniques is not dependent on this choice.

Another issue that has to be considered is the improvement of human problem solving capacities during time. Are solvers getting consistently better and thus distorting our "mean time" metric of puzzle difficulty? In both our datasets there is a correlation between time to solve the puzzle and the day since start of the "experiment". This correlation is presumably caused by improvement in users abilities to solve the puzzle. For `fed-sudoku.eu` the correlation is $r = -0.10$ and it is statistically significant; however, it is not statistically significant within first half of the year and the results for the first half of the year and the whole year are nearly identical. For the `sudoku.org.uk` dataset the correlation is more important and it does distort results. Over the whole set the correlation is $r = -0.30$, which is caused particularly by the improvement during the first two years. For the analysis we use only data after 600 days, for these data the correlation is not statistically significant.

Figure 4 shows histograms of mean time for the two datasets. Solution times for the `fed-sudoku.eu` are smaller than for `sudoku.org.eu` (mean solution time 8 minutes versus 23 minutes) and have smaller variance. We suppose that the main reason is that `fed-sudoku.eu` is used mainly by rather expert puzzle solvers, whereas `sudoku.org.eu` by general public, and `sudoku.org.eu` also seems to include more difficult puzzles (we can compare the difficulty of puzzles only indirectly via our metrics). This diversity between the two datasets is an advantage – despite the difference, our main results (Section 5) are the same over both datasets, and thus we can be quite confident that the results are not an artifact of a particular dataset.

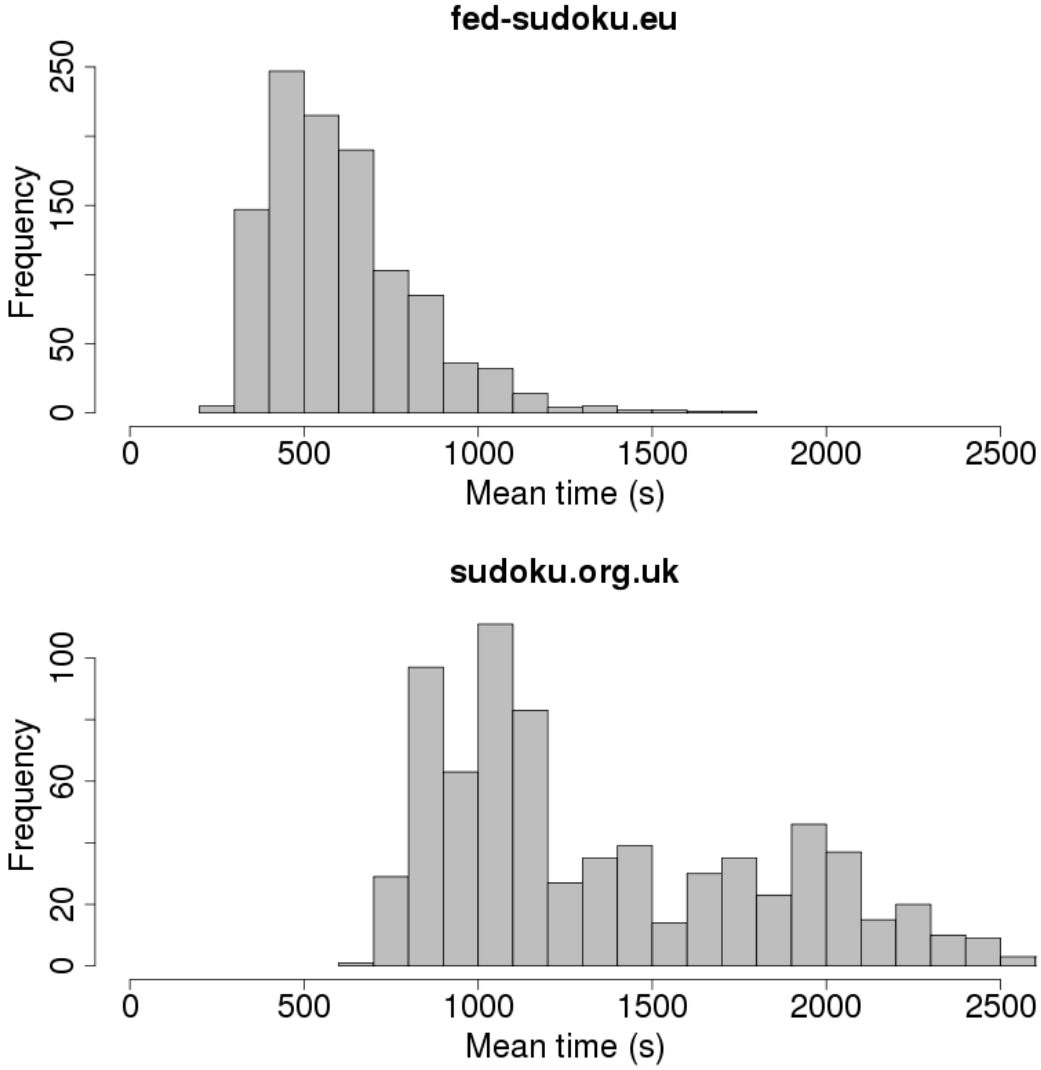

Figure 4: Histograms for the two datasets of mean time to solve the puzzle.

4 Computational Model of Human Solver

In this section we discuss a simple model of a human Sudoku solving. We also provide evaluation of the model using the data on human problem solving. Although we specify and evaluate the model only for Sudoku puzzle, the basic model is general, easily modifiable and applicable to other CSPs.

Our main motivation for developing the model is difficulty rating (Section 5). Nevertheless, the model could be useful in other applications as well, e.g., as a part of a tutoring system [3] or for detection of cheating in Internet Sudoku competitions (if the user fills repeatedly cells in wrong order, then it is probable that he did use computer solver to solve the puzzle).

4.1 General Model

We propose a simple model of human CSP solving, which is based on the following assumptions[1]. Humans are not good at performing systematic search, and there are not willing to do so. Humans rather try to solve CSPs by 'logic techniques', i.e., by constraint propagation. Moreover humans prefer 'simple' techniques over 'difficult' ones (we elaborate on difficulty of logic techniques bellow).

The model proceeds by repeatedly executing the following steps until the problem is solved (see Figure 5 for illustration):

1. Let L be the simplest logic technique which yields for the current state some result (variable assignment, restriction of a candidate set).
2. Let a by an action which can be performed by the technique L. If there are several possibilities how to apply L in the current state, select one of them randomly.
3. Apply a and obtain new current state.

Note that this model makes two simplifying assumptions: at first that the solver does not make any mistakes (i.e., no need to backtrack) and that the solver is always able to make progress using some logic technique, i.e., the solver does not need to perform search. These assumptions are reasonable for Sudoku puzzle and are supported by our data on human problem solving. For other CSPs it may be necessary to extend the model.

[1] We are not aware of any scientific research which could be used to support these assumptions, but there is ample support for them in popular books about puzzle solving.

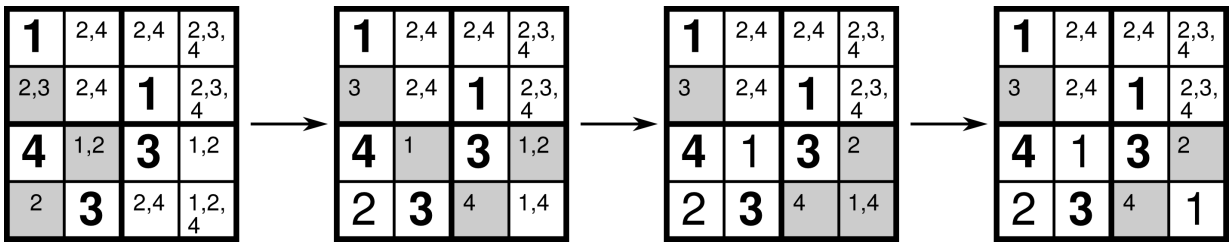

Figure 5: Example of a model run on a sample 4×4 Sudoku puzzle. Grey cells are cells for which the value can be directly determined using one of the simple techniques (naked single, hidden single). In each step one of these cells is selected randomly. Only first three steps of the model run are shown.

Table 1: Difficulty rating of logic techniques as used in the tool Sudoku Explainer [9] (only 8 simplest techniques are shown). The tool provides classification for more than 20 techniques. Some of the simple techniques can be even further characterized due to their relational complexity [13].

Technique	Rating	Technique	Rating
Hidden single	1.2	Naked Single	2.3
Direct Pointing	1.7	Direct Hidden Triple	2.5
Direct Claiming	1.9	Pointing	2.6
Direct Hidden Pair	2.0	Claiming	2.8

4.2 Logic Techniques and Their Difficulty Rating

To specify the stated abstract model, we have to provide list of logic techniques and their difficulty rating. The usual approach used by Sudoku tools is based on a list of logic techniques which are supposed to be simulations of techniques used by humans; each of these techniques is assigned difficulty rating. This rating is provided by the tool developer, usually based on personal experience and common knowledge. Table 1 gives an example of such a rating.

This approach has disadvantage that it contains lot of ad-hoc parameters and it is highly Sudoku-specific, i.e., it gives us limited insight into human problem solving and it is not portable to other problems (the success of the approach is based on significant experience with the problem).

We propose an alternative approach to classification of logic techniques. The approach is based on the assumption that many advanced logic techniques are in fact "short-cuts" for a search (what-if reasoning).

We therefore provide rating of difficulty of logic techniques with the use of search. This approach contains nearly no parameters and is not specific to Sudoku (i.e., it is applicable to any CSP). The only Sudoku specific issue is the selection and realization of "simple" techniques – in our case these are hidden single, naked single techniques; note that these techniques are basically derived from the rules of the problem. For most CSP problems it should be possible to derive basic simple techniques on a similar basis.

Let us suppose that we have a state in which the specified simple techniques do not yield any progress. For each unassigned variable (empty cell) we compute a "refutation score", this score expresses the difficulty of assigning the correct value to this variable in the given state by refuting all other possible candidates.

For each wrong candidate value v we denote ref_v the smallest number of simple steps which are necessary to demonstrate the inconsistency of the assignment. The "ideal refutation score" is obtained as a sum of values ref_v. If some of the values is not refutable by simple steps, we set the score to ∞.

The computation of ref_v can be done by breadth-first search over possible puzzle states, but it is computationally expensive and anyway the systematic search does not correspond to human behavior. Therefor we use randomized approach analogical to our main model – instead of computing the smallest number of steps necessary to refute a given value, we just use a randomized sequence of simple steps and count the number of steps needed to reach an inconsistency. The refutation score is thus a randomized variable.

The variable (cell) with the lowest score is deemed to be the easiest to fill and the refutation score is used as a difficulty rating of an (unknown) logic technique. For all our considered Sudoku puzzles there was always at least one cell with finite score; for more complex problems it may be necessary to further specify the model for the case that all refutation scores have value ∞.

4.3 Evaluation of the Model

Using the described notions we specify a "Simple Sudoku Solver" (SiSuS) model: the general model described in Section 4.1 with two hard-wired logic techniques (hidden single, naked single) of equal difficulty which uses refutation score when the basic techniques are not applicable.

We have evaluated the SiSuS model over detailed data records from `czech-sudoku.com`. To evaluate our model we compare the order in which the

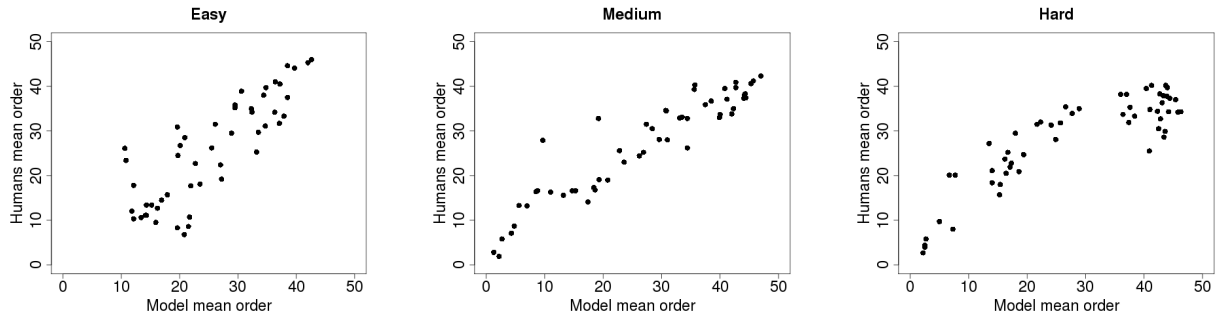

Figure 6: Comparison of cell filling ordering by humans and by model for three sample puzzles of different difficulty. Each dot corresponds to one cell, the positions denote mean order of filling. Correlation coefficients: 0.84 (easy), 0.94 (medium), 0.86 (hard).

cells are filled by humans and the model. For the evaluation we used 15 selected puzzles of wide range of difficulty (from very easy to very difficult). Each puzzle was solved by 10 to 60 solvers.

Based on the data records of human solvers we computed the mean order for each cell. Similarly we computed for each cell mean order over 30 randomized runs of our model. Figure 6 shows the relation between model and humans for three sample puzzles (the puzzles were manually selected to be representative of the results). In most cases the correlation coefficient is between 0.85 a 0.95. Best results are obtained for puzzles of intermediate difficulty. For very easy puzzles there are many ways in which cells can be filled and therefore it is hard to predict the exact order (in this cases the order also differs among individual solvers). Difficult puzzles cannot be solved by the basic techniques used by the model and hence the prediction is again bit worse. Nevertheless, given the simplicity of the SiSuS model, we consider the overall performance to be very good.

5 Difficulty Metrics

Based on the model of human solution progress (Section 4) we now provide several difficulty metrics and evaluate them on the data on human behaviour (Section 3). For all studied metrics we report the Pearson's correlation coefficient.

Note that difficulty rating is interwoven with modeling human solvers. Difficulty metrics are based on the data collected by simulating the model of human solver, but the model depends on rating of difficulty of techniques (see Figure 7). Models which

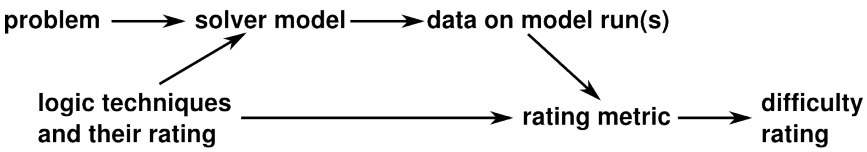

Figure 7: Relations between solver model and difficulty rating.

incorporate many logic technique depend on the intuition of the human designer (alternatively they could use some kind of bootstrapping).

5.1 Combining Rating of Logic Techniques

Given a model of a human solver, a straightforward approach to difficulty rating is to run the model, count how often each logic technique is used and produce the overall rating as a simple function of these statistics. This is the approach used by most Sudoku generators. For our evaluation, we use the following metrics:

Serate metric Default metric used by the Sudoku Explainer tool [9]; it is a maximal difficulty of a used logic technique.

Serate LM metric Linear model over techniques used by the Sudoku Explainer tool; this approach is inspired by [14]. We compute how many times each logic technique[2] was used over each problem. Using half of the problems as a training set we compute parameters for a linear model; the metric is evaluated on the remaining problems (a test set).

Fowler's metric Default metric used by G. Fowler's tool [6]; the metric is given by a (rather complicated) expression over number of occurrences of each logic techniques (with ad-hoc parameter values).

Refutation sum metric Mean sum of refutation scores (Section 4.2) over 30 randomized run of our SiSuS model.

5.2 Dependency Metric

So far we have focused on the difficulty involved in single steps. The overall organization of these steps was considered only in a simple way as a simple function of difficulty

[2]We take into account only techniques which were used in at least 0.5% of all technique applications. There are 13 such techniques, all other techniques were grouped together.

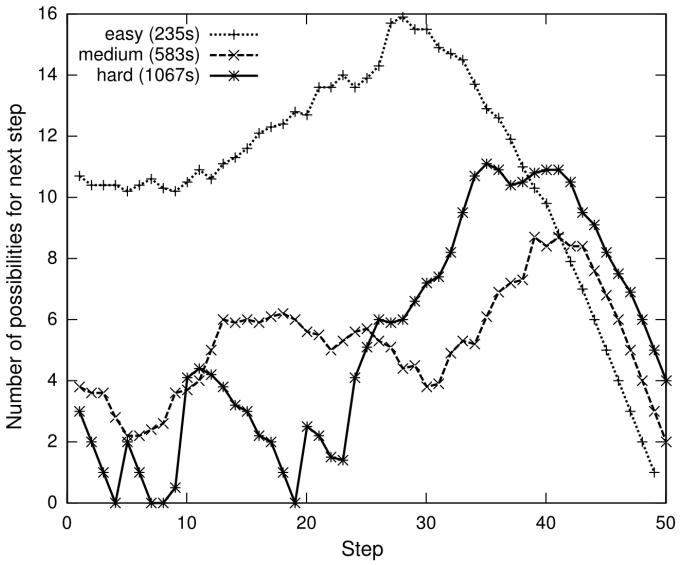

Figure 8: Dependency among steps captured by graph of number of possibilities for the next step. Results for three sample puzzles of different difficulty are shown (the difficulty is indicated by mean solution time of human solvers).

ratings of individual steps. Insufficiency of this approach can be seen particularly for simple Sudokus – these problems are solvable by the basic simple techniques (i.e., the above describe metrics return very similar numbers), but for humans there are still significant differences in difficulty (some problems are more than two times more difficult than others).

Some of this additional difficulty can be explained by the concept of 'dependency' among steps in the solution process (applications of logic techniques). An important aspect of human CSP solving is "the number of possibilities leading to a next step" in each step. For example in our small Sudoku example from Figure 5, there are 3 possibilities in the first step, 4 possibilities in the second and third steps, and so on. It is quite clear that for the classical 9×9 Sudoku it it makes a big difference if we can in the first step apply a logic technique at 10 different cells or only at just 2.

To apply this idea, we count in each step of the SiSuS model the number of possibilities to apply a simple technique. Since the model is randomized, we run several runs and compute for each step mean number of possibilities. Figure 8 shows illustrates a difference among several specific instances – it shows that for easy problem there are many possibilities for progress in each step whereas for hard problem there are only few of them.

To specify a difficulty metric, we need to convert the graphs in Figure 8 to a single number. We simply compute the mean over the first k steps (k is a parameter of the metric). But what is a good value of k? As illustrated by examples in the Figure 8, in the second half of the solution there are usually many possibilities for all problems; i.e., these steps probably do not contribute to the difficulty and therefore it is better to limit the parameter k, on the other hand too small k ignores potentially useful information. We have evaluated the preciseness of the metric with respect to the parameter k over our datasets (Table 2). The results show that a suitable value of k is slightly dependent on the dataset, but generally it is between 20 and 30 and results are not too much dependent on the precise choice of k (for the interval 20 to 30).

Table 2: Dependency metric – correlation coefficient with human performance for different values of parameter k.

k	5	10	15	20	25	30	35	40
`fed-sudoku.eu` all	0.42	0.57	0.65	**0.67**	0.64	0.58	0.51	0.47
`fed-sudoku.eu` simple	0.57	0.64	0.70	0.73	**0.74**	0.73	0.70	0.66
`sudoku.org.uk` all	0.31	0.54	0.62	0.70	0.74	**0.76**	0.76	0.73
`sudoku.org.uk` simple	0.62	0.71	0.76	0.79	**0.80**	0.80	0.78	0.75

5.3 Evaluation

Except for the metrics described above, we also evaluated combinations of metrics, more specifically linear models over several metrics. Parameters of linear models were determined over a training set (one half of the problems), results were evaluated over the other half of models (testing set). We evaluated two linear models. The first combined metric is based on data obtained only from our SiSuS model (linear combination of Refutation sum and Dependency metric; denoted "RD" in Table 3). The second combined metric is a based on four metrics (Serate, Fowler's, Refutation sum, Dependency; denoted "SFRD" in Table 3).

Results are given in Table 3. Figure 9 gives scatter plots for combined metric SFRD as an illustration of the distribution of the data points.

We get consistently better results for `sudoku.org.uk` than for `fed-sudoku.eu`. This is probably mainly due to the wider variability of difficulty in the `sudoku.org.uk` dataset (see the discussion of differences between these datasets in Section 3.2, particularly Fig-

Table 3: Correlation coefficients between metrics and human results. Refutation sum metric is not applicable to simple problems.

	`fed-sudoku`		`sudoku.org`	
metric	all	simple	all	simple
number of givens	0.25	0.22	0.27	0.34
Serate	0.70	0.55	0.86	0.28
Serate LM	0.78	0.60	0.86	0.66
Fowler's	0.68	0.53	0.87	0.64
Refutation sum	0.68	–	0.83	–
Dependency	0.67	0.73	0.69	0.78
Combined (RD)	0.74	–	0.88	–
Combined (SFRD)	0.84	0.75	0.95	0.83

ure 4). Beside the difference in absolute numbers, all other below discussed trends are the same over both datasets.

For the "Simple" subset of puzzles (solvable only by hidden single and naked single techniques), previously studied metrics (Serate, Fowler's) achieve rather poor results; on the other hand, the new Dependency metric works quite well.

The Refutation sum metric achieves only slightly worse results than classical metrics (Serate, Fowler's), despite the fact that it is much more general and simpler technique with only little Sudoku specific aspects (particularly it does not have ad hoc parameters).

Serate LM metric (linear model over data about the usage of 14 logic techniques) achieves similar results as basic Serate metric. Fowler's metric, which differs in details and parameter values but uses the same basic approach as Serate, also achieves similar results. It seems that given the basic approach, the selection of exact parameter values is not that much important. Nevertheless, by combining 4 different metrics, we can significantly improve the overall performance and achieve really good performance of the metric.

6 Conclusions and Future Work

Current popularity of puzzle solving via Internet enables us to easily collect extensive data on human problem solving. Although such data collection is not done under con-

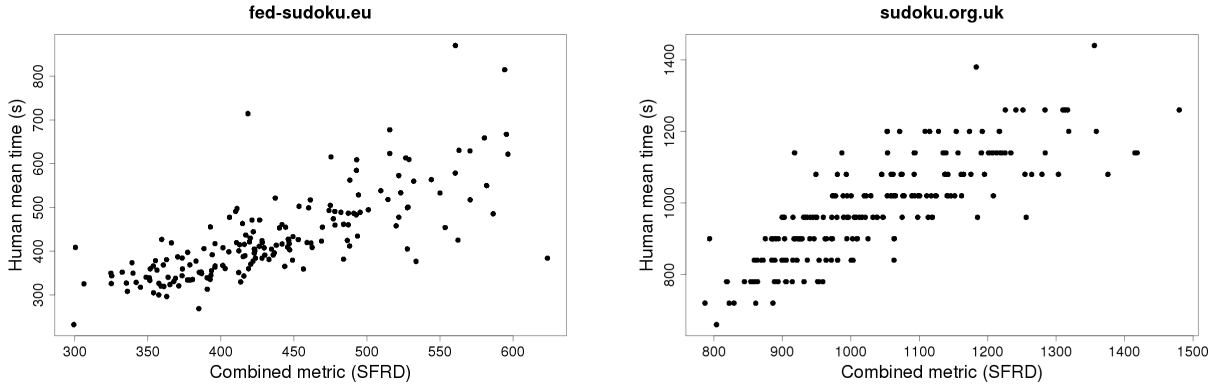

Figure 9: Scatter plots showing relation between prediction of difficulty by combined rating metric and real difficulty (measured as mean solving time). Graphs correspond to the last line in Table 3.

trolled laboratory conditions, our analysis shows that data from Internet may be robust and definitively useful. In our evaluation we used two very different datasets; although we did get different absolute results for each dataset, relative results (comparison among different techniques) was nearly the same – this supports our believe in robustness and usefulness of the data collected from Internet. In this work we use the data to study and evaluate difficulty ratings of a sample problem; but the approach could be used also for other problems and for studies of other cognitive issues (e.g., what kind of errors humans do).

In this work we study a Sudoku puzzle as an example of a constraint satisfaction problem. We provide a general model of human CSP solving. We show that by instantiating the model with only few and simple Sudoku-specific details, we can obtain quite reasonable difficulty rating metric (correlation coefficient up to 0.88). By combining several techniques which are specifically tuned for Sudoku we are able to obtain very good difficulty rating metric (correlation coefficient up to 0.95).

We identify two aspect which influence the problem difficulty: difficulty of individual logic steps during the solution and dependency among individual steps. Previously used techniques [6, 9] focused only on individual logic steps. The novel concept of dependency enabled us to significantly improve the performance of rating.

As the main direction for future work we consider the application of the dependency concept to difficulty rating of other CSPs. Another line for future research is to use similar methodology (Internet based data collection, computational modeling) to study different type of problems (e.g., transportation puzzles which lead to state space

traversal). By combining particular results for individual problems, we can hopefully proceed towards a general theory of problem difficulty.

Acknowledgement

The author thanks to webmasters of `fed-sudoku.eu` and `czech-sudoku.com` portals for providing the data and Jiří Šimša and Petr Jarušek for inspiring discussions.

References

[1] J.R. Anderson, C.F. Boyle, and B.J. Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.

[2] J. Beck, M. Stern, and B.P. Woolf. Using the student model to control problem difficulty. In *Proc. of the Seventh International Conference on User Modeling*, pages 277–288. Springer, 1997.

[3] A. Caine and R. Cohen. Tutoring an entire game with dynamic strategy graphs: The mixed-initiative sudoku tutor. *Journal of Computers*, 2(1):20–32, 2007.

[4] M. Dry, M.D. Lee, D. Vickers, and P. Hughes. Human performance on visually presented traveling salesperson problems with varying numbers of nodes. *Journal of Problem Solving*, 1(1):20–32, 2006.

[5] B. Felgenhauer and F. Jarvis. Enumerating possible Sudoku grids, 2005.

[6] G. Fowler. A 9x9 sudoku solver and generator, 2009. AT&T Labs Research.

[7] M. Henz and H.M. Truong. Sudoku Sat – A Tool for Analyzing Difficult Sudoku Puzzles. *Tools and Applications with Artificial Intelligence*, pages 25–35, 2009.

[8] P. Jarušek and R. Pelánek. Difficulty rating of sokoban puzzle. In *Proc. of the Fifth Starting AI Researchers' Symposium (STAIRS 2010)*. IOS Press, 2010.

[9] N. Juillerat. Sudoku Explainer, 2009.

[10] K. Kotovsky, J.R. Hayes, and H.A. Simon. Why are some problems hard? Evidence from tower of Hanoi. *Cognitive psychology*, 17(2):248–294, 1985.

[11] K. Kotovsky and H.A. Simon. What Makes Some Problems Really Hard: Explorations in the Problem Space of Difficulty. *Cognitive Psychology*, 22(2):143–83, 1990.

[12] P. R. La Monica. Much ado about sudoku. CNNMoney.com, 2005.

[13] N.Y.L. Lee, G.P. Goodwin, and P.N. Johnson-Laird. The psychological puzzle of Sudoku. *Thinking & Reasoning*, 14(4):342–364, 2008.

[14] A. Leone, D. Mills, and P. Vaswani. Sudoku: Bagging a difficulty metric and building up puzzles, 2008. Mathematical Contest in Modeling, University of Washington.

[15] I. Lynce and J. Ouaknine. Sudoku as a SAT problem. In *9th International Symposium on Artificial Intelligence and Mathematics*, 2006.

[16] T. Mantere and J. Koljonen. Solving, rating and generating Sudoku puzzles with GA. In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 1382–1389, 2007.

[17] B. O'Sullivan and J. Horan. Generating and Solving Logic Puzzles through Constraint Satisfaction. In *Proc. of the 22nd national conference on Artificial intelligence*, volume 2, pages 1974–1975. AAAI Press, 2007.

[18] Z. Pizlo and Z. Li. Solving combinatorial problems: The 15-puzzle. *Memory and Cognition*, 33(6):1069, 2005.

[19] H.A. Simon and A. Newell. *Human problem solving*. Prentice Hall, 1972.

[20] H. Simonis. Sudoku as a constraint problem. In *Proc. 4th Int. Workshop on Modelling and Reformulating Constraint Satisfaction Problems*, pages 13–27, 2005.

[21] T. Yato and T. Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 86(5):1052–1060, 2003.