Solving a Large Sudoku by Co-evolving Numerals

Jeffrey Horn
Northern Michigan University
1401 Presque Isle Avenue, Marquette, MI 49855 USA
www: euclid.nmu.edu/~jeffhorn
jhorn@nmu.edu

CCS CONCEPTS

•Computing methodologies → Bio-inspired approaches; Genetic algorithms; •Theory of computation → Evolutionary algorithms; *Self-organization;* **•Mathematics of computing →** *Combinatorial optimization;*

KEYWORDS

genetic algorithm, evolutionary algorithm, evolutionary computation, coevolution, cooperative coevolution, niching, niches, speciation, species, sharing, fitness sharing, resource-defined fitness sharing, exact cover, Sudoku

ACM Reference format:
Jeffrey Horn. 2017. Solving a Large Sudoku by Co-evolving Numerals. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082050

1 INTRODUCTION

Recently we introduced an approach to solving Sudoku problems with co-evolution [4]: *Resource-defined Fitness Sharing for Sudoku* (RFSS). The idea is to find a set of non-conflicting numerals such that every cell in the puzzle is "covered" by a numeral. Each numeral is a species, and species that don't compete (i.e., don't conflict, according to the rules of Sudoku) are *cooperating*. Using a well-known co-evolution algorithm, fitness sharing, we were able to co-evolve numerals to solve a large number of example puzzles. The algorithm is deterministic; there is no discovery operator such as crossover or mutation. It consists solely of selection on *shared fitnesses*. It assumes no knowledge of the domain, no strategies or heuristics for Sudoku, but only the basic rules. The algorithm is general enough to work on any exact cover problem.

In [4] we only looked at the common 9x9 Sudoku puzzle. We have found few heuristic approaches to puzzles larger than 9x9, and we have found no evolutionary approaches to any Sudokus other than 9x9 (and 4x4). We are interested here in whether RFSS can scale to larger Sudoku sizes, such as 16x16, 25x25, 36x36, etc., as the general case of NxN Sudoku is in NP-complete.

Resource-defined Fitness Sharing (RFS) [1], is aimed at the general problem of covering a set of resources with multiple, cooperating (non-competing) species. RFS works by applying proportionate selection to the *shared fitnesses* [2] of species in the population. The shared fitness calculation implements simple co-evolution by reducing the fitness of species in proportion to the amount of overlap they have with other species. Specifically, let p_x be the proportion of the population P occupied by species x. Let S be the set of unique species (i.e., chromosomes) in P, so that $x \in S$. Let y vary over S so that $\sum_{y \in S} p_y = 1$. We can express the shared fitness of a species x as $f_{sh}(x) = (\sum_{y \in S} p_y * f_{x,y})^{-1}$, where $f_{x,y}$ is the pairwise overlap in "coverage" between species x and y. The denominator of $f_{sh}(x)$, known as the *niche count* for x, is the cumulative pairwise overlap between x and other species in S, thus simulating sharing of finite resources.

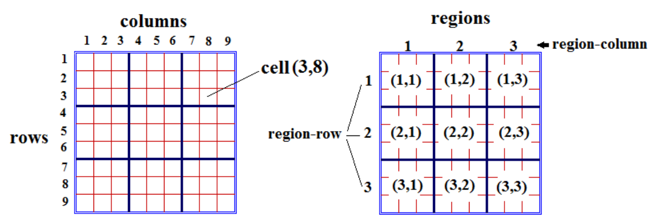

Figure 1: Sudoku terms illustrated by the 9x9 puzzle grid.

In a NxN Sudoku grid there are N rows and N columns of cells, each of which must contain exactly one of the numerals 1..N for a puzzle to be solved. The grid is further divided into N regions of $\sqrt{N}$ by $\sqrt{N}$ cells. Each region also must contain exactly one of each numeral 1..N in a solved puzzle. Here we use the 9x9 grid in Figure 1 to illustrate the terms.

In [4] we cast Sudoku as a resource covering problem in which a solution to the puzzle forms an exact cover of the resources, and vice versa. The constraints are implemented as resources. For example (switching to the 25x25 size grid of Figure 2 to illustrate) every row must have exactly one instance of each numeral 1–25. So there are twenty five row resources per row, e.g., placing a "5" in row 8 covers one resource, while placing a "6" in row 8 covers a different resource, and a "5" in row 7 is yet another resource. Thus there are $25 * 25 = 625$ "row resources." Similarly there are another 625 "column resources," and other 625 "region resources." (Note that Figure 2 shows 25 regions of 5x5 cells square.) Finally, there is a fourth category of resources corresponding to each of the 625 cells: there must be a numeral placed into each cell. Thus there are 625 "cell resources" to be covered. There are a total of $625 + 625 + 625 + 625 = 2500$ unique resources to be covered, but clues will eliminate many of these.

The species are placements of numerals into particular cells. Thus one unique species is "numeral 14 in row 3, column 8" (which

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082050

−	2	−	−	−	3	14	−	8	−	−	−	−	−	−	−	−	13	4	24	−	7	1	−	−
−	10	17	−	−	−	6	18	−	−	22	16	−	12	−	−	−	−	1	−	−	−	13	19	−
−	15	24	13	7	−	−	−	4	−	10	−	−	3	14	−	18	−	−	−	−	22	2	6	−
−	−	1	21	−	−	15	−	22	−	−	19	13	−	−	−	8	−	−	−	−	16	18	20	−
−	5	−	−	20	7	25	19	−	−	−	21	17	18	2	10	12	22	9	15	11	−	−	−	−
11	−	−	−	22	8	−	24	7	1	5	−	−	−	13	16	17	25	23	2	4	−	6	−	19
16	9	12	−	17	−	19	22	−	−	−	−	18	21	−	−	20	6	13	−	7	−	−	23	11
−	−	6	−	21	9	16	−	3	−	−	22	20	19	−	−	−	−	15	8	25	−	−	−	−
−	−	23	5	−	2	−	−	11	17	8	−	−	−	16	12	9	−	−	21	−	3	10	−	−
−	−	−	−	−	6	−	−	12	−	9	1	25	−	3	−	11	−	−	7	−	−	21	−	−
−	−	9	−	−	23	−	5	17	4	16	−	11	−	22	18	2	−	21	13	−	−	7	−	−
4	6	−	−	5	−	−	2	−	−	−	18	21	24	−	−	19	3	−	12	23	−	−	17	−
−	−	−	12	11	−	7	3	−	24	17	20	15	13	19	1	−	5	8	−	6	9	−	−	−
−	22	−	−	14	19	−	6	16	−	−	8	9	7	−	−	−	24	−	−	3	−	−	1	18
−	−	21	−	−	25	13	−	20	8	12	−	14	−	10	9	16	15	−	6	−	−	4	−	−
−	−	25	−	−	24	−	−	18	−	4	−	3	10	5	−	1	−	−	14	−	−	−	−	−
−	−	5	3	−	17	−	−	23	7	13	−	−	−	18	19	21	−	−	22	−	11	12	−	−
−	−	−	−	18	10	8	−	−	−	−	25	23	2	−	−	5	−	16	11	9	−	3	−	−
17	20	−	−	2	−	22	16	6	−	−	7	12	−	−	−	−	9	3	−	18	−	23	24	25
6	−	4	−	16	1	11	12	25	3	19	−	−	−	21	17	23	8	−	18	2	−	−	−	14
−	−	−	−	4	14	24	11	19	23	21	17	16	8	−	−	−	1	2	9	13	−	−	5	−
−	1	14	23	−	−	−	−	9	−	−	−	19	5	−	−	24	−	12	−	−	8	17	−	−
−	16	11	8	−	−	−	−	1	−	6	4	−	−	23	−	15	−	−	−	14	12	9	10	−
−	21	3	−	−	−	17	−	−	−	−	15	−	25	20	−	−	4	10	−	−	−	16	11	−
−	−	20	2	−	16	5	8	−	−	−	−	−	−	−	−	6	−	19	25	−	−	−	3	−

Figure 2: Sudoku puzzle 25x25.

might be encoded on a chromosome as $\langle 14, 3, 8 \rangle$). Such a species would cover four resources: a row resource for placing a "14" in row 3, a column resource for placing a "14" in column 8, a region resource for placing a "14" in region (1,2), and a cell resource for placing a numeral in cell (3,8). Since every species covers exactly four resources, we assign a credit of $\frac{1}{4}$ for covering one resource. (In a puzzle with no clues there are $25 * 25 * 25 = 15,625$ species, one for each combination of row, column, numeral.)

Next we need a computation for species overlap, that is, the size of the intersection of two sets of resources covered by two species x and y: $f_{x,y}$:

$$f_{x,y} = \begin{cases} 0 \\ +1/4 \text{ iff SameCell}(x, y) \\ +1/4 \text{ iff SameNumeral}(x, y) \wedge \text{SameRow}(x, y) \\ +1/4 \text{ iff SameNumeral}(x, y) \wedge \text{SameColumn}(x, y) \\ +1/4 \text{ iff SameNumeral}(x, y) \wedge \text{SameRegion}(x, y). \end{cases} \tag{1}$$

where "SameCell(x,y)" returns **true** if species x and y specify the same row and column and returns **false** otherwise, "SameNumeral(x,y)" returns **true** if species x and y specify the same numeral, etc. Note that $f_{x,y} \in [0..1]$, where $f_{x,x} = 1$ (a species completely overlaps with itself) and $f_{x,y} = 0$ (if species x and y have no overlap). For example if species $x = \langle 14, 3, 8 \rangle$ and species $y = \langle 22, 6, 1 \rangle$ then $f_{x,y} = 0$ (different numerals, different cells). As another example, $f_{\langle 18,11,22 \rangle, \langle 18,11,23 \rangle} = \frac{1}{2}$, since both species place the same numeral, 18, in the same row and the same region (third region-row down, fifth region-column from the left), but in different columns and different cells.

For the initial set of species S we start with all 15,625 possible numeral placements, eliminate all those that are identical to the k clues of the given Sudoku puzzle, and eliminate any species that conflict with the clues. This last step shrinks the population considerably (typically down to 1000-2000 species) as many species conflict with the clues.

2 RESULTS

Here we present a run of RFSS on a 25x25 Sudoku from the internet [3], shown in Figure 2. The puzzle has 301 clues, leaving $625 - 301 = 324$ unknowns (i.e., the dashes, or hyphens, in Figure 2). The clues eliminate 13,922 of the possible 15,625 species, leaving 1703 *competing species* for the evolution. Each species starts off with an initial population proportion of $\frac{1}{1703}$, indicated by the horizontal dotted line in Figure 3. This figure plots the evolution of the population over 750,000 generations (species proportions are plotted every 10000 generations) during which the proportions diverge. Although it takes over 700,000 generations, there is a clear partitioning of the species into two groups. The upper group numbers 324 species and does indeed form a solution to the puzzle.

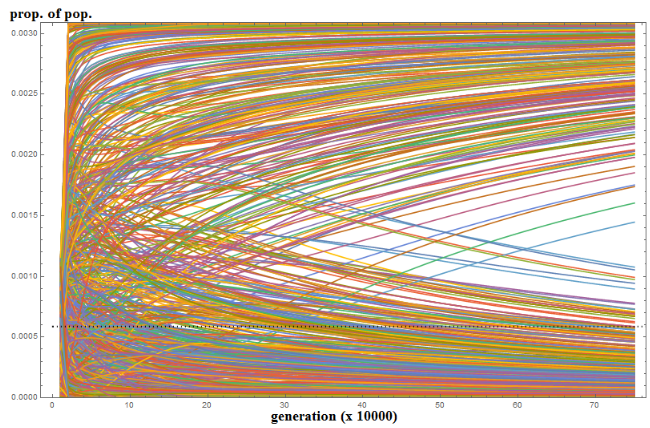

Figure 3: RFSS for 750,000 gen.s on the 25x25.

3 CONCLUSION

This one run of RFSS on a single 25x25 Sudoku puzzle shows that it can find a solution to a large Sudoku problem, but it is not clear how difficult this problem is. For the larger Sudokus there are few example puzzles to be found, let alone tools, techniques, or consensus on how to rate their difficulty. For many Sudoku enthusiasts it is the sheer size of the puzzles that provides the most obvious increase in difficulty. And that is precisely why the result shown here is intriguing to the author: the algorithm seems able to cope with interactions among >1000 species. But clearly more problems at different size levels need to be run.

References

[1] Horn, J.: Resource-based fitness sharing. In: J.J. Merelo Guervós, et. al. (ed.s) Parallel Problem Solving From Nature – PPSN VII, Lecture Notes in Computer Science, Vol. 2439, pp. 381–390. Granada, Spain, Springer-Verlag, Berlin, Germany. (2002).
[2] Goldberg, D. E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: JJ. Grefenstette (ed.) Proceedings of the Second International Conference on Genetic Algorithms – ICGA2, pp. 41–49. Lawrence Erlbaum Associates. (1987).
[3] http://www.puzzle-magazine.com/25x25-Sudoku-magazine.php
[4] Horn, J.: Co-evolution of Sudoku solutions. In: M. Adek (ed.) 20^{th} International Conference on Soft Computing – MENDEL 2014, pp. 117–122. (ISSN 1803-3814, ISBN 978-80-214-4984-8) Brno, Czech Republic. (2014).