Sudoku Players' Forums

Rating rules / Puzzles. Ordering the rules

Goto page Previous 1, 2, 3 ... , 25, **26**, 27 Next

newtopic postreply **Sudoku Players' Forums Forum Index -> Advanced solving techniques**

View previous topic :: View next topic

Author	Message

eleven
 	
Joined: 10 Feb 2008
Posts: 463
	Posted: Fri Jul 24, 2009 8:05 am Post subject: quote
	
	This modification would output the number of grids needed after the puzzle:
	Add a global variable cnt e.g. after the line
	int solve();
	double cnt = 0.0;
	Increment it at mark m0:
	m0:cnt+=1.0;for(i=1;i<=81;i++)A[i]=A0[i];part=0;if(argc<4)solve();
	Print it in the line with the puzzle after a blank
	for(i=1;i<=81;i++)printf("%c",L[A[i]]);printf(" %g\n", cnt);
	If you only want the grids for this puzzle, reset it then
	cnt = 0.0;
	
	@Red Ed
	What i really would like to hear is something like this:
	Take these 20 mio unbiased generated grids and i will not bother you with criticizing your results.
	
	Back to top profile pm

denis_berthier Joined: 19 Jun 2007 Posts: 790 Location: Paris, France	Posted: Fri Jul 24, 2009 9:42 am Post subject: quote edit **eleven wrote:** This modification would output the number of grids needed after the puzzle Thanks. I just did these modifications and I gave it a short run. My rough evaluation of the worth of a controlled-bias minimal puzzle (in my previous post) was largely underestimated: it is not worth 2000 to 3000 top-down minimals as I thought, but it is worth more than 100,000. It means that: - you need (in the mean) this quantity of complete grids to obtain one controlled-bias minimal puzzle; - for unbiased statistics, a collection of less than 10,000 controlled-bias minimals is worth a collection of 1,000,000,000 ordinary top-down minimals (the eternal trade quality vs quantity). Here's the beginning of the output of suexg-cb with seed 0:

....7..3.7.6.9...44.9.1....6.2..7.9.1.4..5...8...62.1.9.....2.8...1..7...3...8. 141709
..5....29.43.9.....7.....8.3763.6...95...4.....1.73.985...2.....4.2 300467
2....3.7..9.....837..69....3..75.....8....5..3...92.....84.....7653..46...27 22007
5.....12.....8..1...73..4.5..4.28...13.....6...1.95.....6....9..571..4.3..... 223580
...648.....6.1..3...92.....1....3..64..9..1...25.7.8..3...9.6...5....2....1.2..5. 45609
1.8.6..9..6.....74...628.....651.7...537.2.....4..8..5....9.....2.3..6 73537
.3....9.....65....61.3.45.....5..2.859...61...718.....96..2.....4.7..95.1...4. 445840
.3..6.58...12.....79..87.5...96..3....5.....14..87.....9681.....59.3... 47235
9...4658..6...2.4...81...9.....4...2.7.58.....51.49.....7.8.....1....3.6. 466614
..36.....4.937.5..75...3.....17...2.....1....95.43....68.....8..9.5..9..1.. 7452

(mean number of grids per puzzle: 177,605)

The sample of 10 puzzles is obviously too small for a precise estimation, but it gives an order of magnitude.

I wont' have much time before my summer holidays, but, if you're interested in such data, when I'm back I'll run it again for a better estimate (unless you do it in the meanwhile).

Back to top profile pm www

denis_berthier

Posted: Sat Sep 12, 2009 7:30 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

A few additional intermediary results about the controlled-bias generator

- as mentioned previously, it is very slow (it generates ~ 200 minimal puzzles per day on a 3 GHz CPU);
- in the mean, it consumes ~ 225,000 complete grids to generate a single minimal puzzle;
- unfortunately, as of today, we have no better generator to deal with unbiased statistics;

- a precise unbiased estimation of the number-of-clues distribution of minimal puzzles requires a large sample, especially if we want to estimate the tail of the distribution (... 19-clue, 20-clue ; 30-clue, 31-clue ... minimal puzzles);
- estimating precisely the lower part of the tail (... 19-clue, 20-clue) is not necessary because the correction factors lead to wiping away almost everything (this means that the real proportion of 17-, ... 20-clue minimal puzzles is very small, much smaller than top-down generators would suggest);
- but estimating the upper part (30-clue, 31-clue ...) is more useful, because the correction factors amplify everything.

Given the current sample of 50,000 puzzles obtained with this generator, and using the mean values for the SER and NRCZT obtained from larger collections of minimal puzzles for each fixed number of clues, I could compute mean values for the unbiased SER and NRCZT (see previous posts or my web pages).

Since then, I have noticed that, for fixed n, n-clue mean values computed with the controlled-bias generator do not coincide exactly with mean values computed with top-down or bottom-up generators. I had already noticed a small difference between the top-down and bottom-up generators, but it was unclear whether it was due to sampling errors.
For the unbiased generator, things seem to be clear (at least for n in the 22-29 range). The difference is not so large that it could change a lot about the SER or NRCZT mean values or standard deviations.
But it shows that:
- there are some forms of bias that can't be reduced to the number of clues;
- the controlled-bias generator (which, for each fixed n, is an unbiased generator of n-clue

puzzles) eliminates them.

Back to top profile pm www

Allan Barker

Posted: Sun Sep 13, 2009 6:05 am Post subject: quote

Joined: 21 Feb 2008
Posts: 337
Location: Bangkok

> **denis_berthier wrote:**
>
> If this discussion should continue, can we do it on the "Rating" thread, so as not to hijack eleven's thread?

At Denis's request I am post this here as a continuation of a discussion here

> **ronk wrote:**
>
> I think there was some agreement long ago that "length" would be based on the count of native strong inferences. What wasn't settled was the penalty, if any, for using networks instead of chains.

> **ronk wrote:**
>
> IIRC that agreement was for chains, but to remain consistent for naked/hidden quads, that would be four native strong inference sets.

> **denis_berthier wrote:**
>
> Yes, the agreement was about quads (naked, hidden and super-hidden) included in chains. That's the only choice we have if we want to make serious comparisons of different chain patterns.

> **denis_berthier wrote:**
>
> Of course, for a net, the lengths of all the branches should be added (with no repetition for the common parts). As to whether there should be an additional penalty for nets and how we could define it, I have no idea (we have no classification of the structure of nets that could help us in this matter)

> **ronk wrote:**
>
> In addition to counting the native strong inference sets in each branch, I think applying a "one count" penalty for each branch would be simple ... and fair too.

Counting native strong inference sets has been used in numerous ways and useful for a long time

I had suggested here that counting the absolute number of truths (same thing) can be applied to any kind of pattern/logic and doing so correlates well with other intrinsic properties of a puzzle.

While there is certainly good reason for adjusting such a scale for logic built into chains, hidden/naked complements, branching, etc., there is equally good reason to also retain an intrinsic scale based only on the number of truths. Such an intrinsic scale is equally applicable across all patterns and logic, and does not depend on any model or method.

> **denis_berthier wrote:**
>
> I also argued that, for consistency reasons, a quad in a chain should be counted as a substructure of length 4 because most quads can also be seen as nrczt-whips[4].

> But some very rare cases can only be seen as nrczt-braids[4]. Braids are a relatively mild kind of nets, but they are nets anyway (and you can see that I avoid using them). Whence the problem with your suggestion. Notice this problem doesn't exist within the AIC or ALS frameworks, but a different one may arise: the number of "restricted commons". Such problems are related to what I meant above by "structure" of a net.

A very good reason to have an intrinsic scale without additional rules attached. I would also wonder, if penalties were applied for branching, whould it be possible to decide what constitues branching? Along the same lines, and nrczt promotion would also count as a branch, right? My only purpose here is to stur the mud so as to say, "Look! how muddy this water is!"

Back to top profile pm www

denis_berthier

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

Posted: Sun Sep 13, 2009 8:12 am Post subject: quote edit

Allan Barker wrote:

> While there is certainly good reason for adjusting such a scale for logic built into chains, hidden/naked complements, branching, etc., there is equally good reason to also retain an intrinsic scale based only on the number of truths. Such an intrinsic scale is equally applicable across all patterns and logic, and does not depend on any model or method.

AFAIK, you've evoked an "intrinsic scale" and the "number of truths" several times, but you haven't yet defined any of them. (If I'm wrong, it's just that I'ven't been much present on this forum for several months and I may have missed it. In this case, please tell me where).
The whole question is again, what's the "number of truths" you assign to a quad? If your "number of truths" was the same for a quad and a single, I doubt it could have any interest for a measure of complexity of patterns.

denis_berthier wrote:

> if penalties were applied for branching, whould it be possible to decide what constitues branching?

That's why I evoked these problems in my post. I don't think we can consistently define a penalty for branching, unless we first define what kinds of net patterns and of branchings we allow.

denis_berthier wrote:

> Along the same lines, and nrczt promotion would also count as a branch, right?

No. (BTW, nrczt-promotion is Paul's vocabulary, not mine. And it is at the programming level, not at the conceptual.)
By *definition*, an nrczt- chain or whip is a (linear) sequence of left- and right- linking candidates, each nrc-linked to the previous one. There is no branching. (To justify the eliminations it allows, you have to look backwards at the previous rlc's, but that's not branching.)
To see the difference with a net, try to interpret any of your patterns as a sequence of candidates, each nrc-linked to the previous one!

Anyway, that's not the problem. It is obvious that the notion of branching can't be applied to your patterns, because branching would mean there is a partial order on your nets (instead of the total order we should have in a chain) and the notion of order is totally alien to your approach.

You have no possibility of making a difference between a net and a chain. This is true conceptually (as a result of considering sudoku as a set cover problem) and also for your solver: you have no possibility of asking the solver to find chains preferentially to nets. If a chain happens to appear, it isn't the result of the solver finding it as a chain, but it is because of chance and because you have interpreted it as a chain by superimposing an order on the set of candidates involved in the net. In this respect, I noted that some of your claims about the solver finding chains are very misleading.
BTW, these solving/interpreting phases aren't true only of chains but of any pattern. Your solver has three parts (correct me if I'm wrong):
- part one (the solver proper) is based on a classical general purpose algorithm (a set cover algorithm, i.e. a *non-pattern-based* algorithm), that can be very fast *because* it is not pattern-based,
- part two can output a graphical representation of each step of the solver, leaving the user free to interpret it as he wants,
- part three (the interpreter) can interpret some steps of the solver in terms of predefined patterns.
As I see it, the player is there as an assistant to the program to make human meaning out of it.

Consider now what I've shown previously in this thread:
- almost all the minimal puzzles can be solved with chains (specifically nrczt-whips);
- a natural definition of length can be given for these chains, consistent with the natural AICs or ALSs interpretation that can also be given to subparts of them in some cases;
- the NRCZT rating of nrczt-whips is an intrinsic measure of their complexity;
- it is strongly correlated with the SER, an independent (and not intrinsic) measure of complexity based on a completely different set of patterns;
- the number of partial chains (an intuitive measure of the difficulty of finding useful ones) increases exponentially with the nrczt rating.
All these properties speak in favour of this intrinsic rating.

If you succeed in defining a scale for patterns based on nets (which are needed only in very exceptional cases - fewer than 1 in a million) and if it is consistent with a few natural constraints (scale = 1 for singles, 2 for pairs, ...), I think that, when applied to rate puzzles in a SE max-complexity style, it will be strongly correlated with the NRCZT rating. We now have all the necessary background to test this.

I'm looking forward to seeing your definitions.

Back to top profile pm www

David P Bird

Posted: Sun Sep 13, 2009 8:19 am Post subject: quote

Joined: 17 Sep 2008
Posts: 129
Location: Middle England

Some observations from the touchline:

Leaving aside what solving methods are considered acceptable, the dominant factor for me is how a deduction would look if it were expressed in pure Boolean logic which should show how many different premises had to be considered simultaneously and for how many steps more than two truth states were actively under scrutiny (counting the logical ANDs). But in practice it's far more complicated than that!

Some time ago on the Eureka forum it was generally accepted that when we demonstrated that a candidate couldn't exist in a particular cell, that fact became a "Sudoku Truth" and could be used freely wherever and whenever we wanted. This seems so obvious that it's overlooked now, but it sets a precedent which allows puzzles to be solved piecemeal using remembered Sudoku Truths rather than in a single continuous logical chain.

What was NOT accepted at that time was that a proven fact such as (a)cell1 and (b)cell2 have

equivalent truth states could be accepted as a Sudoku Truth as well. Consequently although many follow-on deductions could be derived using an item of knowledge, every time it was applied, it racked up the cumulative difficulty score.

There are sound practical reasons behind this decision because manual solvers can mark the known true and false states for candidates using pencil marks but can't mark other facts. However it's irrational that we are allowed to remember that cell1 can't contain (a) but are obliged to forget equivalences and such. How well this applies to manual solvers who don't use pencil marks depends on their memories, and of course with computers it's trivial to save such information for re-use whenever convenient.

This philosophy washes over to net-based deductions including DB's chains. If we're not allowed to remember the outcomes of following the branches one at a time, we must show them as if they were being followed in parallel. Furthermore, in the midst of this is how to rank the use of pattern based theorems many of which can only be proved using the iteration of cases!

We therefore have a variety of ranking options open to us for dealing with these different aspects, when the most fundamental consideration is what is the assumed solving scenario? By changing the assumed scenario, we could probably completely reverse the rankings of virtually any two solutions paths we want. For individuals it's therefore a question of finding their own comfort zones, but for a ranking committee it's much more difficult.

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

Posted: Sun Sep 13, 2009 9:34 am Post subject: quote edit

> **David P Bird wrote:**
>
> it's irrational that we are allowed to remember that cell1 can't contain (a) but are obliged to forget equivalences and such.

Your point is about solving paradigms. The dominant solving paradigm is based on candidates and doesn't include equivalences of candidates or disjunctions of candidates. AFAIK, no one has succeeded in developing a paradigm based on such additional "basic truths". It is likely that this is related to the impossibility of writing them simply on the grid.

Such *modelling choices* are not only a matter of logic or "rationality", or even of ease of representation on the grid, but also of computational complexity.

> **David P Bird wrote:**
>
> with computers it's trivial to save such information for re-use whenever convenient.

But there is a far from trivial complexity problem with using such information.

> **David P Bird wrote:**
>
> This philosophy washes over to net-based deductions including DB's chains. If we're not allowed to remember the outcomes of following the branches one at a time, we must show them as if they were being followed in parallel.

If only you had read and understood the definition of nrczt-chains, you'd have seen that there are no branches in them. In these chains, the only thing we have to recall is the sequence of left- and right- linking candidates, i.e. the chain itself. Do you know of any pattern that you don't have to remember, if only to state it in the result?
If I follow your line of thought, considering ALS chains and all the links internal to each ALS and between adjacent ALSs, all of which are necessary for the deductions, you should also consider them as very complex nets. Idem for AICs containing ALSs.

> **David P Bird wrote:**

> Furthermore, in the midst of this is how to rank the use of pattern based theorems many of which can only be proved using the iteration of cases!

Do you have a problem with iterative proofs in maths? Then open a thread for this (or read the Eureka discussions you already had on this 2 years ago). Our problem here is not to rank theorems but patterns and puzzles. This is sufficiently complex that we don't have to add irrelevant topics to it, especially when these have already been discussed elsewhere.

David P Bird wrote:

> We therefore have a variety of ranking options open to us for dealing with these different aspects, when the most fundamental consideration is what is the assumed solving scenario?

Right. But if we don't accept the candidate based paradigm, there's currently not much we can do.

David P Bird wrote:

> By changing the assumed scenario, we could probably completely reverse the rankings of virtually any two solutions paths we want.

You can declare that a quad is simpler than a single or imagine any other exotic scenarios. If you want your scenario to be compatible with common sense, I'm not sure there are so many options. If you have one to propose, it'd be worth a thread of its own.

Back to top profile pm www

David P Bird

Joined: 17 Sep 2008
Posts: 129
Location: Middle England

Posted: Sun Sep 13, 2009 11:25 am Post subject: quote

Denis, as I see it you would like to know how your solver ranks against other methods for solving puzzles at different difficulty ratings. But this begs the question of how we rank the difficulty of different puzzles in the first place! That question opens up a completely different can of worms some of which I tried to cover in my résumé of the Eureka debate. The other worms involved are what methods and tools are the solver allowed to use, such as guessing, backtracking etc. or in terms of the old climbing analogy, are we free climbing, allowing a restricted set of climbing aids, or can we go ahead and order the scaffolding?

The items I raised effect how tough puzzles can be for a manual solver who relies on his wits more than a blanket coverage of all the options. I find that a difficult path which connects different areas of a grid once discovered will often be re-used several times to progressively achieve the necessary reductions. That's a lot easier than having to find several different difficult paths, but that's not reflected in the way existing rating systems work. So for me some quite highly rated puzzles are much easier than others with considerably lower ratings.

A manual solver will therefore be far better equipped for re-using a discovered equivalence than a computer that starts from scratch again for each deduction it makes. Yes it's awkward to program so as to take advantage of previous findings, but it's not impossible (spread sheets usually only cascade the changes from an edited cell without recalculating an entire workbook).

How you and I see nrctz chains working is a largely question of our respective definitions and is not really an issue. You simply allow yourself a climbing aid I don't.

You now come onto some points of interest regarding the use of patterns. Pattern theorems can often only be proved using methods that many refuse to accept in the course of a solution. I look on this as a weakness as it means that puzzles can't always be solved from first principles and the results of some theorems must be remembered. But the effect of this is that if we recognise a pattern, we simply apply the known results for it. How many native strong inference sets there are in a naked quad is therefore of no consequence to someone who has learnt what deductions follow. HOWEVER because of those SISs, checking by eye to

see of a naked or hidden quad exists is hard going! Much the same goes on with ALS - you're looking at the internal branching involved, and I'm looking at how easily they can be spotted by eye from which I think we reach much the same conclusions.

Given my way, we would score more difficulty points when a particular ALS is used for the first time, than for its subsequent re-use. So if we are using strong inferences as a measure, we keep an inventory of them, when one is used for the first time we look up its score, but thereafter simply score it as a single point. I could even take this further to cover for when we retrace previously used chain segments.

If the solving scenario that interests you is one involving a computer with no storage of previous intermediate results, then your approach using the internal complexity of the structures involved is sensible, but otherwise what I'm trying to point out is that we are comparing apples and oranges, and there is no single difficulty measure that can sensibly be applied.

Back to top profile pm

denis_berthier

Posted: Sun Sep 13, 2009 11:59 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

David,
There's not much in your post I can disagree with and maybe not much either I can completely agree with.

> **David P Bird wrote:**
>
> I see it you would like to know how your solver ranks against other methods for solving puzzles at different difficulty ratings.

The question is not about the solver, SudoRules. The nrczt-rating doesn't depend on any implementation. It is purely logical. SudoRules is just the tool I use to compute it in practice.
In a previous post, you mentioned ratings based on logical formulæ. Here, I don't consider any logical formula, but I consider ONLY logical formulæ.
But yes, I'd like to compare this nrczt-rating with other ratings. Indeed, I've already extensively done it with SER and to a lesser extent with suext, suex9, gsf's Q1 and Q2.
What I'd particularly like to do is to compare it with a rating based on the length of AICs including the full zoo of ALSs, AHSs and A-Fish. Unfortunately, it doesn't seem anyone has a solver able to compute such an AIC rating.
And, if Allan succeeds in defining his rating based on pure net structures (independent of any notion of order) I'd like to add it to the list for comparisons.

> **David P Bird wrote:**
>
> But this begs the question of how we rank the difficulty of different puzzles in the first place! That question opens up a completely different can of worms some of which I tried to cover in my résumé of the Eureka debate.

The above mentioned ratings are all based on different sets of rules. They are nevertheless pretty well correlated.
Again, the answer isn't going to fall from heaven. It is one more modelling problem. The only positive thing we can do if we are not pleased with the existing ratings is try to define new ones.

> **David P Bird wrote:**
>
> The other worms involved are what methods and tools are the solver allowed to use, such as guessing, backtracking etc.

Of course, such methods are not of any interest for this thread, because they are not related to any sensible notion of complexity. With backtracking or guessing, EasterMonster and c° are no harder than any other puzzle.

David P Bird wrote:

> If the solving scenario that interests you is one involving a computer with no storage of previous intermediate results, then your approach using the internal complexity of the structures involved is sensible, but otherwise what I'm trying to point out is that we are comparing apples and oranges, and there is no single difficulty measure that can sensibly be applied.

I'm not interested in programming details but in resolution rules and their intrinsic complexities, i.e. the complexities of the patterns in their condition part.
But, if this is what you're asking: no, SudoRules doesn't keep track of any "intermediate results" when building prgressively nrczt-chains or whips, it keeps track of exactly the same things a human solver must remember: the sequence of candidates.

Back to top

profile pm www

Steve K

Joined: 19 Jan 2007
Posts: 129
Location: Cincinnati Ohio

Posted: Sun Sep 13, 2009 1:14 pm Post subject:

quote

suppose one needs to consider a set T = [t1,t2, ... tn] of N truths to reach a deduction. Without considering the type of deduction, if we define exactly what we allow as members of T, N can be an objective measure of complexity.

Further measures of complexity are difficult, as they may presume something about the mechanics, or logic, of the deduction in question. For example, with an AIC which does not use any derived deductions within it, one might opine that the complexity would rise with something akin to 2**N, as one needs to search for another bi sis with which to link. However, this is imperfect. In Sudoku, ignoring uniqueness, one needs to search in 3 perpendicular directions (needs to consider 3 nand sets) from a sis to find the links. This would seem to indicate 3**N as a relative complexity measure. This is yet flawed, for many reasons.

Here is but one potential flaw. A human brain is a pattern recognition machine. It therefor seems to find sub-patterns which are incorporated into the logic. One does not search the internal mechanics of this sub-pattern once one has already incorporated it into the logical scheme. Many of these sub-patterns share a commonality: For example, the following sub-patterns: naked subset, hidden subset, Xwing, swordfish, xy ring each have as their basis a continuous loop/net. For each of these nets, at each node in each participating sis, there are only two directions: The sis direction, or sis container, and the "weak" link direction, or "nand" container. This suggests that there exists a common factor to these types of patterns, or sub-patterns. One might reasonably opine that such subpatterns would receive the rating akin to 2**N, where N is the number of native sis considered. However, incorporation of any adjustment to step ratings would require objective clarity. Objective clarity seems unlikely.

Here is a related potential flaw. Uniqueness derived inferences are a bit of a problem, as they, by their nature, are always comprised of a group of conditions. Some care needs be exercised to properly define T so that every uniqueness group can be objectively evaluated in every possible logical construct.

It would be nice if singles had a very low complexity rating, perhaps zero, perhaps 1. It would be nice if a formula would be simple. It would be nice if a formula seemed relevant enough to reflect something akin to the complexity of finding the deduction. Although no system will be perfect, I would suggest one of two alternatives:

A) ingore the possible (nand) complexity issue and simply consider the total number of "truths", or "Native sis" considered. This is very simple, and leaves open how to compare, for example, M1 chains of length N1 to M2 chains of length N2. That is a difficult thing to assess, but I think it needs assessing.
Some notes on A: One might expect that puzzle ratings which only consider the most complex

step that a solution uses stand some chance of showing a relationship with each other. One might suspect that a rating system which considers every step in the solution would not relate well to a rating system that considers only the most complex step. In fact, if such a relationship could be shown to exist, it would probably tell us something important.

Therefor, it is not completely unlikely that Denis' use of N as suggested above has a decent correalation with SE ratings. Both rate a puzzle considering only the most highly rated step used. I do not find that either rating tells the complete story, as some, for example, SE7.2 puzzles seem far more difficult than others. Generally this feeling of difficulty stems from the existence of more steps of "suffucient" complexity. Thus, a more complete rating system may need to address the following: How many deductions with complexity vector (N-1) are roughly equivalent to one deduction with complexity vector N.

With both SE ratings and Denis' ratings, only the most highly rated step within each puzzle is compared. We know therefor that there is a good relationship between the most highly rated SE step per puzzle and the most highly rated of Denis' steps per puzzle. We have no information regarding the complexity of any of the other steps per individual puzzle. We can guess that the use of N as a vector per step could maintain the consistency per individual step.

The possibility of N**3 seems to have some basis in the complexity of searches. However, this could imply that 3 steps of complexity (N-1) are roughly equivalent to one step of complexity N. This may be a matter of significant contention..... We may determine that the operator between steps of complexity N need not be addition.

B) Attempt to further refine the rating to include a downgrade for subparts that are less complex regarding Nand links. This rating system would be more complex and perhaps sometimes ambiguous. If potential ambiguity can be overcome, such a rating system might have advantages. Such advantages would probably be difficult to quantify. One could hardly argue that an advantage which is only subjective is objectively an advantage.

Summary: The proposal to use N, as described above, as a vector to rate an individual step seems reasonable. I am well aware that this is not a new proposal.
To expand that step rating system to an entire puzzle is more problematic. Some judgements would be required. Nevertheless, if we agree upon a good way to rate an individual step, we may be able to agree on a puzzle rating system that incorporates each step.
I would prefer a rating system that: a) looked at the complexity of each step and then. b) derived puzzle complexity ratings considering the relative complexity of each step. c) allowed some steps to achieve a lower rating based upon some objective criteria. I suspect that a) is possible and could be mathematically meaningful, b) may lose some mathematical meaning, c) is more likely to be political then mathematically meaningful. Thus, c) is a personal preference that is best totally ignored. a&b seem to have promise.

Denis has uncovered something which is meaningful:
A strong relationship between disparite rating systems. We know that both only consider the most highly rated step per puzzle. We can infer from the large sample sizes that there is some hope in extending this relationship if we also consider the other steps in the puzzles in each of the disparite rating systems.
I suggest that feelings and impressions are completely irrelevant to this endeavor. Whatever the vector may be that compares one step to another step, it should be determined by what is required to maintain the relative relationships that Denis has already shown.

Back to top profile pm email www YIM

Allan Barker Posted: Sun Sep 13, 2009 4:30 pm Post subject: quote

Joined: 21 Feb 2008
Posts: 337
Location: Bangkok

Denis Berthier wrote:

> AFAIK, you've evoked an "intrinsic scale" and the "number of truths" several times, but you haven't yet defined any of them. (If I'm wrong, it's just that I'ven't been much present on this forum for several months and I may have missed it. In this case, please tell me where). The whole question is again, what's the "number of truths" you assign to a quad? ....

Denis,

The truths I refer to are the original 324 constraints from the definition of Sudoku, the same as native sis or just "truth" without qualifiers. I have only pointed out one can count these truths for most any kind of deduction which, although simplistic, serves as a scale that is independent of methodology. I have not defined anything new that I know of. From this perspective a quad would have 4 truths.

Denis Berthier wrote:

> ..... BTW, these solving/interpreting phases aren't true only of chains but of any pattern. Your solver has three parts (correct me if I'm wrong): ....

Without right or wrong, I'd say your view is a bit upside down. I'm insterested in your impressions and will post something to clarify the picture, perhaps here or somewhere else.

Back to top profile pm www

denis_berthier

Posted: Sun Sep 13, 2009 4:42 pm Post subject: quote edit

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

It is true that considering only the hardest step may give a very partial idea of the complexity of a puzzle, especially if one forgets that such ratings (SER, NRCZT,...) have only statistical validity.
Considering instead the n hardest steps or the whole sequence of steps seems appealing. I remember seeing some nice graphs from Allan.
But it also suffers from severe problems, especially if one wants to use it in statistical studies:
- it depends on the resolution path; generally there are many resolution paths, some of which are harder than others;
- trying all the resolution paths is impossible in practice (computation times would be too long; it already took me several weeks to compute the NRCZT-ratings of the 1,000,000 puzzles in sudogen0; considering all the resolution paths, or even a small part of them, would multiply this by 1 or 2 orders of magnitude);
- if computed from a single resolution path, it is implementation dependent.

I have no miracle solution in my rucksack!

Back to top profile pm www

denis_berthier

Posted: Sun Sep 13, 2009 4:48 pm Post subject: quote edit

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

Allan Barker wrote:

> The truths I refer to are the original 324 constraints from the definition of Sudoku, the same as native sis or just "truth" without qualifiers. I have only pointed out one can count these truths for most any kind of deduction which, although simplistic, serves as a scale that is independent of methodology. I have not defined anything new that I know of. From this perspective a quad would have 4 truths.

OK. But then, can you define a rating of puzzles based on this (in the hardest step style) and use your solver to compute it?

Back to top profile pm www

denis_berthier

Posted: Wed Sep 16, 2009 10:54 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 790
Location: Paris, France

eleven wrote:

> One more puzzling result. I generated minimal puzzles bottom up without dropping any clues, i.e. i only took unique puzzles, if they were minmal from the adding clue phase.
> Maybe Mike can repeat it to exclude coding errors.
>
> **Code:**
>
> ```
> 20:       3
> 21:      46
> 22:     524
> 23:    2301
> 24:    3757
> 25:    2549
> 26:     710
> 27:     106
> 10000 puzzles, av. clue number 24.0804
> ```
>
> To my surprise the average clue number only is slightly higher and still clearly under the value from top down generation.

As I was involved in the controlled-bias generator, I had missed this.
Did you do this with a modified version of suexg1.4? Is it available?

Back to top profile pm www

eleven

Posted: Wed Sep 16, 2009 12:32 pm Post subject: quote

Joined: 10 Feb 2008
Posts: 463

denis_berthier wrote:

> As I was involved in the controlled-bias generator, I had missed this.
> Did you do this with a modified version of suexg1.4? Is it available?

I replaced the lines

Code:

```
//now we have a unique-solution sudoku. Now remove clues to make it
minimal
part++;if(solve()!=1)goto m0;
for(i=1;i<=81;i++){mr4:x=(MWC>>8)&127;if(x>=i)goto
mr4;x++;P[i]=P[x];P[x]=i;}
for(i1=1;i1<=81;i1++){s1=A[P[i1]];A[P[i1]]=0;if(solve()>1)A[P[i1]]=s1;}
```

by those:

Code:

```
//now we have a unique-solution sudoku
// only accept it, if it is mninimal, DONT remove clues to make
it minimal
for(i1=1;i1<=81;i1++)
{
   s1=A[i1];if (s1 < 1) continue; A[i1]=0;
   if(solve()>1)A[i1]=s1;
   else goto m0;
}
```

Back to top profile pm

Display posts from previous: All Posts | Oldest First | Go


Sudoku Players' Forums
Forum Index -> Advanced solving techniques

All times are GMT
Goto page Previous 1, 2, 3 ... , 25, **26**, 27 Next

Page 26 of 27

Stop watching this topic

Jump to: Advanced solving techniques | Go

You **can** post new topics in this forum
You **can** reply to topics in this forum
You **can** edit your posts in this forum
You **can** delete your posts in this forum
You **can** vote in polls in this forum

Powered by phpBB © 2001, 2005 phpBB Group