Sudoku Players' Forums

THE REAL DISTRIBUTION OF MINIMAL PUZZLES

Goto page Previous 1, 2, 3 ... 8, 9, 10, 11, 12 Next

newtopic postreply **Sudoku Players' Forums Forum Index -> General/puzzle**

View previous topic :: View next topic

Author	Message
Red Ed Joined: 06 Jun 2005 Posts: 611	Posted: Thu Jul 16, 2009 1:53 pm Post subject: quote **denis_berthier wrote:** So congratulations to eleven for the first application of these formulæ to the unbiased estimation of the mean number of clues. I don't suppose that I get any congratulations for the first application of my algorithm which generalises and preceded your formulae. :) Back to top profile pm
eleven Joined: 10 Feb 2008 Posts: 364	Posted: Thu Jul 16, 2009 2:16 pm Post subject: quote Red Ed, your 77 clue example. Do you mean, it shows, that puzzles with (exactly) n clues you get with that (classical) top down algorithm can be biased (some would be found more often than the others like in Coloins sample) ? If so, i did not get it. Back to top profile pm
Red Ed Joined: 06 Jun 2005 Posts: 611	Posted: Thu Jul 16, 2009 2:27 pm Post subject: quote Yep, that's what I was showing. I had posted evidence of the same phenomenon in the "Top-down toy box" thread, too. The claim is fine for the modified generator, but not for the original one. Since Denis has turned his attentions to the modified generator now, perhaps(?) he too now rejects the claim for the original top-down generator. Is there a part of my post that you'd like me to explain, or shall we just forget about the original generator? Back to top profile pm
Red Ed Joined: 06 Jun 2005 Posts: 611	Posted: Thu Jul 16, 2009 2:41 pm Post subject: quote There's potentially something more important to worry about, and that's bias in the solution grid source upon which these unbiased estimation procedures rely. I ran my algorithm with s=29,c=24 on two sources: (1) my unbiased solution grid generator; (2) the complete solution grids from sudogen0_1M. I had to stop

the processes early to write this post, so here are just some interim results:

Unbiased solution grid generator

Code:

```
Number of solution grids: 444992
Number of 29-clue subgrids containing a 24-clue
minimal puzzle: 1708

Total number of 24-clue minimal puzzles in those 1708
29-clue subgrids:
  +----+---------+------------+------------+
  | Cl |   Count | E(nr/grid) | E(std dev) |
  +----+---------+------------+------------+
  | 24 |   23371 |  1.02e+014 |  3.67e+012 |
  +----+---------+------------+------------+
```

Solution grids from sudogen0_1M

Code:

```
Number of solution grids: 416795
Number of 29-clue subgrids containing a 24-clue
minimal puzzle: 1811

Total number of 24-clue minimal puzzles in those 1811
29-clue subgrids:
  +----+---------+------------+------------+
  | Cl |   Count | E(nr/grid) | E(std dev) |
  +----+---------+------------+------------+
  | 24 |   24771 |  1.15e+014 |  3.91e+012 |
  +----+---------+------------+------------+
```

Informally, these appear to have diverged: that is, the sudogen results are significantly higher than the unbiased ones.

I'll rerun the test overnight on the whole of sudogen0_1M, but in my opinion this preliminary result should be quite worrying for users of suexg or whatever it was that generated sudogen0_1M.

Back to top profile pm

Red Ed

Joined: 06 Jun 2005
Posts: 611

Posted: Thu Jul 16, 2009 10:17 pm Post subject: quote

While I was sleeping the computer finished the sudogen0_1M collection (twice), getting estimates of 1.22e14 (std dev=2.73e12) and 1.25e14 (std dev=2.74e12). That's even worse than the apparent bias reported above.

I'd like to repeat the test with exactly the same solution grid generator that you (eleven) are using, for several seeds. Which one is it?

Back to top profile pm

eleven

Joined: 10 Feb 2008
Posts: 364

Posted: Thu Jul 16, 2009 11:43 pm Post subject: quote

Red Ed wrote:

> I had posted evidence of the same phenomenon in the "Top-down toy box" thread, too.

Ah thanks, i have missed that there *is* a concrete example proving it. I did not manage to construct one.

Red Ed wrote:

> I'd like to repeat the test with exactly the same solution grid generator that you (eleven) are using, for several seeds. Which one is it?

I am using the same suexg version (sudo_gen.c), with which sudogen0_1M was calculated.

I was aware, that dukuso's algorithm would not generate unbiased grids, but i am very surprised, that this gives such a strong bias.
Alternatively the program can read the grids from a file. Maybe this is a better option than to replace the generator in the source.

I also dont know, if dukuso's simple RNG can be a problem.

But when we only can generate unbiased n clue puzzles using the slow modified algorithm, these are just theoretical questions for me. it would take me a year to get 350 29-clue puzzles for a random 10000 puzzle collection.

Back to top profile pm

eleven

Posted: Fri Jul 17, 2009 12:43 am Post subject: quote

Joined: 10 Feb 2008
Posts: 364

Now these are the results of the run overnight (as we now know, biased anyway).

Code:

```
59508900 tries, 246 puzzles, (241906/puzzle)
         estimate      %
22:   1  6.25e+11    0.013
23:   6  9.62e+12    0.216
24:  29  1.12e+14    2.425
25:  83  7.33e+14   15.762
26:  73  1.39e+15   29.860
27:  47  1.82e+15   39.162
28:   6  4.48e+14    9.641
29:   1  1.37e+14    2.946
average 26.4599
```

Back to top profile pm

Red Ed

Posted: Fri Jul 17, 2009 1:57 am Post subject: quote

Joined: 06 Jun 2005
Posts: 611

Thanks. So let's look at your 24-clue estimate, as that's the #clues that I was exploring above. 29 puzzles in 59508900 attempts gives E(nr/grid) = 1.12e14 as you said and E(std dev) = about 18% of that figure. So the main problem with your experiment appears not to be the solution grid generator, but rather large standard deviation: you'd have to run it for something like 20x as long to get down to the std dev that I reported. That's a problem inherent in Denis' sampling algorithm, quite independent of any bias there may or may not be in suexg.

59 million tries is impressive, though. I'm using quite a slow solver. I'll try replacing it with the one in suexg to see if I can get better performance.

Back to top profile pm

denis_berthier

Posted: Fri Jul 17, 2009 2:03 am Post subject: quote

Joined: 19 Jun 2007
Posts: 746
Location: Paris, France

Red Ed wrote:

> While I was sleeping the computer finished the sudogen0_1M collection (twice), getting estimates of 1.22e14 (std dev=2.73e12) and 1.25e14 (std dev=2.74e12). That's even worse than the apparent bias reported above.

I wonder how 2 computations with the same data can give (even slightly)

different results (1.22e14 and 1.25e14)?
Anyway, we already know that top-down generaotrs are biased wrt to the number of clues. I don't understand what you want to prove here.

When you compare this to your supposedly unbiased 1.02e+014, that makes a 20% difference. Not so worrying, considering that we don't know that much of your "unbiased" generator.
Can we use this generator?
Can we have large collections of grids generated by it?

Could you make the same tests on Allan's collection of puzzles (http://www.sudokuone.com/xsudo1/rab1mran.zip), generated with his own top-down generator whose first phase (complete grids) is based on completely different principles?
Why, if suexg was so bad in the complete grid generation phase, would Allan's generator give results in every respect very close to those of sudogen0_1M?

Back to top profile pm www

eleven

Joined: 10 Feb 2008
Posts: 364

Posted: Fri Jul 17, 2009 2:41 am Post subject: quote

Thanks for the analysis, Red Ed.

After all i am happy to find good reasons to stop the puzzle generations on my computer

Back to top profile pm

Red Ed

Joined: 06 Jun 2005
Posts: 611

Posted: Fri Jul 17, 2009 2:57 am Post subject: quote

denis_berthier wrote:

> I wonder how 2 computations with the same data can give (even slightly) different results (1.22e14 and 1.25e14)?

There are two sources of randomness normally: the solution grids and the paths (or in my case subgrids). Only the former is fixed in this experiment of course.

Quote:

> Anyway, we already know that top-down generaotrs are biased wrt to the number of clues. I don't understand what you want to prove here.

A top-down ("classical" or "modified") puzzle generator consists of solution grid generation and clue deletion. It seemed that you were assuming that bias problems lay only in clue deletion, so that "modified" top-down generation would be unbiased. My results suggest that small, but significant, problems lie in the (suexg) solution grid generator too.

Quote:

> When you compare this to your supposedly unbiased 1.02e+014, that makes a 20% difference. Not so worrying, considering that we don't know that much of your "unbiased" generator.

Whether or not my solution grid generator is "unbiased" (it is, by the way), you should at least appreciate that the style of solution grid generator has an affect on the estimates that come out of the clue deletion process.

Quote:

> Can we use this generator?

> Can we have large collections of grids generated by it?

Not yet.

Quote:

> Could you make the same tests on Allan's collection of puzzles (http://www.sudokuone.com/xsudo1/rab1mran.zip), generated with his own top-down generator whose first phase (complete grids) is based on completely different principles?

I'll need to find the time to generate the complete solution grids from his puzzles but, yes, good idea, I'll do that some time.

Quote:

> Why, if suexg was so bad in the complete grid generation phase, would Allan's generator give results in every respect very close to those of sudogen0_1M?

"... in every respect very close ..." might be overstating it: I don't recall any very formal analysis being done to compare the two. I'm only claiming a few percent difference in the statistics we're trying to estimate. If that's close enough for you then maybe you shouldn't worry after all.

Back to top profile pm

denis_berthier

Posted: Fri Jul 17, 2009 3:12 am Post subject: quote

Joined: 19 Jun 2007
Posts: 746
Location: Paris, France

Red Ed wrote:

> **denis_berthier wrote:**
>
> > When you compare this to your supposedly unbiased 1.02e+014, that makes a 20% difference. Not so worrying, considering that we don't know that much of your "unbiased" generator.
>
> Whether or not my solution grid generator is "unbiased" (it is, by the way), you should at least appreciate that the style of solution grid generator has an affect on the estimates that come out of the clue deletion process.

I do. But this supposed bias is not yet proven (cf question about Allan's generator).

Red Ed wrote:

> **denis_berthier wrote:**
>
> > Can we use this generator?
> > Can we have large collections of grids generated by it?
>
> Not yet.

Then we have no chance of knowing for sure if it's biased or not wrt puzzle creation.

Red Ed wrote:

> **denis_berthier wrote:**
>
> > Could you make the same tests on Allan's collection of puzzles (http://www.sudokuone.com/xsudo1/rab1mran.zip), generated with his own top-down generator whose first phase (complete grids) is based on completely different principles?

> I'll need to find the time to generate the complete solution grids from his puzzles but, yes, good idea, I'll do that some time.

Don't waste your time doing this (I mean the complete grids). This computation is currently running, because I needed it for other purposes. As soon as it is finished, I'll put it on my web pages.

> **Red Ed wrote:**
>
> > **denis_berthier wrote:**
> >
> > > Why, if suexg was so bad in the complete grid generation phase, would Allan's generator give results in every respect very close to those of sudogen0_1M?
>
> "... in every respect very close ..." might be overstating it: I don't recall any very formal analysis being done to compare the two. I'm only claiming a few percent difference in the statistics we're trying to estimate. If that's close enough for you then maybe you shouldn't worry after all.

I've given detailed comparisons for 4 generators (including suexg-x.x and Allan's) on my web pages:
http://www.carva.org/denis.berthier/HLS/Classification/index.html

Back to top profile pm www

Red Ed

Joined: 06 Jun 2005
Posts: 611

Posted: Fri Jul 17, 2009 3:49 am Post subject: quote

> **denis_berthier wrote:**
>
> > this supposed bias is not yet proven

Indeed not *proven*; but the current statistical evidence is quite strong, and I plan on gathering further evidence soon.

> **Quote:**
>
> > we have no chance of knowing for sure if [Red Ed's solution grid generator]'s biased or not wrt puzzle creation.

You only need to know that it samples in an unbiased manner from the space of all solution grids. And for that, I refer you to the "unbiased grid generation" thread for the time being. If you have questions, ask them there please.

> **Quote:**
>
> > Don't waste your time doing this (I mean the complete grids). This computation is currently running, because I needed it for other purposes. As soon as it is finished, I'll put it on my web pages.

Excellent; thanks.

> **Denis wrote:**
>
> > **Red Ed wrote:**
> >
> > > **denis_berthier wrote:**
> > >
> > > > Why, if suexg was so bad in the complete grid generation phase, would Allan's generator give results in every respect very close to those of sudogen0_1M?
> >
> > "... in every respect very close ..." might be overstating it: I don't recall any very formal analysis being done to compare the two. I'm only claiming a few percent difference in the

> statistics we're trying to estimate. If that's close enough for you then maybe you shouldn't worry after all.

I've given detailed comparisons for 4 generators (including suexg-x.x and Allan's) on my web pages:
http://www.carva.org/denis.berthier/HLS/Classification/index.html

Ah yes, I'd overlooked that. Well then, my answer to your question is: "I don't know".

Back to top profile pm

denis_berthier

Posted: Fri Jul 17, 2009 3:50 am Post subject: quote

Joined: 19 Jun 2007
Posts: 746
Location: Paris, France

eleven wrote:

> But when we only can generate unbiased n clue puzzles using the slow modified algorithm, these are just theoretical questions for me. it would take me a year to get 350 29-clue puzzles for a random 10000 puzzle collection.

You're right that these are just theoretical questions. If your main purpose is to generate minimal puzzles and/or to select interesting ones, you don't have to bother with bias.
Nevertheless the modified algorithm is interesting (and thank you again for implementing these changes), because it allows a theoretical analysis of bias, which the original suexg didn't.

eleven wrote:

> After all i am happy to find good reasons to stop the puzzle generations on my computer

I'm continuing.
As my main interest is not the number of clues itself but the complexity of puzzles and as the correlation between the two is small, I need only a rough estimate of the distribution of the number of clues.

Back to top profile pm www

denis_berthier

Posted: Fri Jul 17, 2009 4:11 am Post subject: quote

Joined: 19 Jun 2007
Posts: 746
Location: Paris, France

GENERATING MINIMAL PUZZLES WITH CONTROLLED BIAS

1) A controlled-bias top-down generator

A standard top-down generator works as follows:

Code:

```
1) generate a random complete grid
2) loop:
     let P be the current puzzle
     2a) choose one clue randomly from P and delete
it, you get a puzzle P2
     2b) if P2 is minimal, return P2
     2c) if P2 has several solutions, GOTO 2a
     2d) otherwise, set P=P2
end loop
```

Clause 2c makes any analysis very difficlut. Moroever, it also causes the generator to go deeper, i.e. towards puzzles with fewer clues. It thus introduces a strong bias.
Consider therefore the following, modified top-down generator of minimal puzzles.

Code:

```
1) generate a random complete grid
2) loop:
     let P be the current puzzle
     2a) choose one clue randomly from P and delete
it, you get a puzzle P2
     2b) if P2 is minimal, return P2
     2c) if P2 has several solutions, GOTO 1
     2d) otherwise, set P=P2
end loop
```

The only difference is in clause 2c: if we find a multi-solution puzzle, instead of backtracking to the previous state, we merely discard the current complete grid and restart the search with another one.
Notice that, contrary to the standard top-down algorithm which produces one minimal puzzle per complete grid, the modified algorithm will generally use several complete grids before it outputs a minimal puzzle. (The question is, how many? Experimentations show that many complete grids are necessary before a minimal puzzle is reached. But this is a question of efficiency of the generator, not a conceptual problem.)
Once this algorithm is defined, it can be implemented by a simple modification of the top-down sueg-x.x (the version of suexg used to build the sudogen0_1M collection), call it suexg-cb. Thanks to eleven for implementing this modification. Of course, the modified generator is much slower than the original one. The purpose here is not speed, but controlled bias.

This top-down generator works similarly to the random uniform search defined in section 2 below and according to the same transition probabilities. And, as we stop it when B is reached, it outputs non indexed puzzles according to the probability Pr on B defined below.

Let us now build our formal model of this generator.

2) A forest of paths from complete grids to puzzles

Consider first the following layered structure (a forest of trees with branches pointing downwards), the nodes being indexed (single or multi- solution) puzzles:
- floor 81 : the N different complete solution grids, each indexed by the empty sequence; notice that all the puzzles at floor 81 have 81 clues;
- floor 80: each indexed puzzle P at floor 81 sprouts 81 branches pointing to floor 80, one for each clue C in P; the other end of this C branch will be the indexed puzzle obtained from P by removing clue C and indexed by the 1-element sequence (C); notice that all the puzzles at floor 80 have 80 clues;
- recursive step: given floor n+1 (all indexed puzzles of which have n+1 clues and are indexed by sequences of length 81-(n+1)), build floor n as follows: each indexed puzzle P at floor n+1 sprouts n+1 branches leading to an indexed

puzzle at floor n: for each clue C in P, the indexed puzzle Q obtained from P by removing clue C has its index defined as the (81-n)-element sequence obtained by appending C to the end of the index of P; notice that all the indexed puzzles at floor n have n clues and index length equal to 1 + (81-(n+1)) = 81-n.

The index of an indexed puzzle P has a clear intuitive meaning: it is the sequence of clue deletions starting from the complete grid of P and leading downwards to P.
Given an indexed puzzle P, there is an underlying (non-indexed) puzzle: the ordinary puzzle obtained by forgetting the index of P.

It is easy to see that, at floor n, each indexed puzzle has an underlying (non indexed) puzzle identical to that of (81 - n)! indexed puzzles at the same floor (including itself).
This is equivalent to saying that, at any floor n < 81, any non-indexed puzzle P can be reached by exactly (81 - n)! different paths from the top (all of which start necessarily from the complete grid of P). These paths are the (81 - n)! different ways of deleting its missing 81-n clues.

Let N be the number of complete grids. At each floor n, there are N * 81! / (81-n)! / n! non-indexed puzzles.
At each floor n, there is therefore a uniform probability P(n) = 1/N * 1/81! * (81-n)! * n! that a non-indexed puzzle Q at floor n is reached by a random (uniform) search from the top.

What is important here is the ratio (valid globally if we start from all the complete grids, but also valid if we start from a single complete grid): **P(n+1) / P(n) = (n + 1) / (81 - n)**.

Call B the set of (non-indexed) minimal puzzles. On B, all the puzzles are minimal. Above B, all the puzzles have redundant clues.
On the set B of minimal puzzles there is a probabily Pr naturally induced by the different Pn's (and renormalised to sum up to 1) and it is the probability that a minimal puzzle is reached by our controlled-bias generator.
On B, by construction of Pr, the above relation **Pr(n+1) / Pr(n) = (n + 1) / (81 - n)** remains valid between any two minimal puzzles, with respectively n+1 and n clues.

[Edit 07/20/09. Added the following paragraph]
For n < 41, this relation means that a minimal puzzle with n clues is less likely to be reached from the top than a minimal puzzles with n+1 clues. More precisely, we have:
Pr(40) = Pr(41),
Pr(39) = 42/40 * Pr(40),
Pr(38) = 43/39 * Pr(39).

A non-biased search would give the same probability to all the minimal puzzles. The above relation shows that the uniform search of the controlled bias generator is biased towards puzzles with fewer clues but that this bias is well known.

As we know precisely the bias with respect to uniformity, we can correct it easily by applying correction factors cf(n) to the probabilities on B. Only the relative values of the cf(n) is important: they satisfy **cf(n+1) / cf(n) = (81 - n) / (n + 1)**.

Mathematically, after normalisation, cf is just the relative density of the uniform distribution on B with respect to the probability distribution Pr.

[Notice that a classical top-down generator is still more biased towards puzzles with fewer clues because, instead of discarding the current path when it meets a multi-solution puzzle, it backtracks and tries to go deeper.]

3) Computing unbiased means and standard deviations using a controlled-bias top-down generator

How can we, in practice, compute statistics of minimal puzzles based on a (large) sample produced by a controlled-bias top-down generator?
If we consider any random variable X defined (at least) on minimal puzzles, let:
- on(n) be the observed number of puzzles with n clues in the sample,
- E(X, n) be the observed mean value of X for puzzles with n clues in the same sample,
- sd(X, n) be the observed mean value of X for puzzles with n clues in the same sample.

The raw (biased) mean of X is classically estimated as sum[E(X, n) * on(n)] / sum[on(n)] (theorem on the additivity of the mean values).
The unbiased mean of X must be estimated as: **unbiased-mean(X) = sum[E(X, n) * on(n) * cf(n)] / sum[on(n) * cf(n)]**.

Similarly, the raw (biased) standard deviation of X is classically estimated as sqrt{sum[sd(X, n)^2 * on(n)] / sum[on(n)]} (theorem on the additivity of the variances - beware, not the standard deviations!).
And the unbiased standard deviation of X must be estimated as: **unbiased-sd(X) = sqrt{sum[sd(X, n)^2 * on(n) * cf(n)] / sum[on(n) * cf(n)]}**.

These formulæ show that the cf(n) sequence needs be defined only modulo a multiplicative factor.

It is convenient to choose cf(24) =1. This gives the following sequence of correction factors (in the range 19-31, which includes all the puzzles of the sample):

cf-sequence[19...31] = 0.006568 0.02036 0.05915 0.1613 0.41379 1 2.28 4.9108 10.003 19.29 35.258 61.11 100.54

It may be shocking to consider that 30-clue puzzles must be given a weight 61 times greater than the 24-clue puzzles, but that's how it is.
A consequence of all this is that unbiased statistics on the mean number of clues of minimal puzzles must rely on extremely large samples with sufficiently many 29-clue and 30-clue puzzles.

4) Applications (figures to be updated when a larger sample is available)

Let's use the above definitions on a small (4,000) collection of puzzles generated

by the controlled-bias top-down generator suexg-cb.
This is just an illustration of the method. The sample is still too small to draw strong conclusions.
All the figures in this section should be taken with caution.

4.1) The unbiased mean number of clues of minimal puzzles = 26.54

raw-average = 25.65 unbiased-average = 26.54

The unbiased value for the mean number of clues is 0.9 clues more than the raw mean number.

We can have more precise mean values and standard deviations for the SER and NRCZT of minimal puzzles with a given number of clues, just by using the already known results from larger samples, e.g. sudogen0_1M. Let's do this in the sequel.

4.2) The unbiased mean SER of minimal puzzles = 4.47

raw-average = 4.15 unbiased-average = 4.47
raw-standard-deviation = 2.493 unbiased-standard-deviation = 2.53

This unbiased value for the mean SER is only 0.32 above the raw mean value.
The standrad deviation is unchanged.

4.3) The unbiased mean NRCZT-rating of minimal puzzles

raw-average = 2.138 unbiased-average = 2.303
raw-standard-deviation = 1.34 unbiased-standard-deviation = 1.379

This unbiased value for the mean NRCZT is only 0.16 above the raw mean value.
The standard deviation is almost unchanged.

Form the above results, we can conclude that the huge difference between the raw and the unbiased mean number of clues leads to moderate differences between the raw and the unbiased SER rartings.
This can be understood on the basis of the results in section 2 relative to:
- the very weak correlation between the number of clues and the SER or NRCZT ratings,
- the small trend for increasing SER or NRCZT with increasing number of clues.

4.4) The real distribution of clues of minimal puzzles

The following estimation is merely the product of the observed distribution and the correction factors, namely on(n) * cf(n) (normalised, of course, by sum(on(n) * cf(n)).

Code:

#clues	raw-occurrences (4,000sample)	unbiased-occurrences (normalised to 1,000,000 puzzles)
19	0	0.0
20	0	0.0
21	0	0.0
22	3	23.47
23	69	1,384.90
24	481	23,330.94
25	1223	135,253.48
26	1354	322,519.01
27	679	329,462.02
28	179	167,503.67
29	12	20,533.51 (*)
30	0	0.0 (*)
31	0	0.0 (*)

* values based on few data are not reliable.

The vast majority of puzzles produced by the controlled-bias top-down algorithm is still in the range [23 - 28] clues, but the real distribution is notably different from the raw distribution. For n ≤ 26, it has fewer occurrences, for n > 26 it has more occurrences.

This table shows that precise knowledge of few-clue puzzles (< 25) is not necessary. But precise knowledge of puzzles with more than 28 clues is capital. Unfortunately, they remain very hard to find.

Last edited by denis_berthier on Mon Jul 20, 2009 12:31 am; edited 2 times in total

Back to top profile pm www

Display posts from previous: All Posts Oldest First Go

newtopic postreply **Sudoku Players' Forums Forum Index -> General/puzzle**

All times are GMT - 8 Hours

Goto page Previous 1, 2, 3 ... 8, 9, 10, 11, 12 Next

Page 9 of 12

Jump to: General/puzzle Go

You **cannot** post new topics in this forum
You **cannot** reply to topics in this forum
You **cannot** edit your posts in this forum
You **cannot** delete your posts in this forum
You **cannot** vote in polls in this forum