Sudoku Players' Forums

THE REAL DISTRIBUTION OF MINIMAL PUZZLES

Goto page Previous 1, 2, 3 ... , 20, 21, 22 Next

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

View previous topic :: View next topic

Author	Message
David P Bird Joined: 16 Sep 2008 Posts: 150 Location: Middle England **Back to top**	Posted: Thu Sep 24, 2009 10:25 am Post subject: quote **Red Ed,** That's not what I would prefer - you've upset me - I've told you - now let's behave like adults. profile pm
Red Ed Joined: 06 Jun 2005 Posts: 728 **Back to top**	Posted: Thu Sep 24, 2009 10:36 am Post subject: quote *<rolls eyes>* Fine. profile pm
Red Ed Joined: 06 Jun 2005 Posts: 728	Posted: Thu Sep 24, 2009 10:56 am Post subject: quote Consider this: **Code:** 123|...|... 456|...|... 789|...|... ---+---+--- ...|123|... ...|456|... ...|789|... ---+---+--- ...|...|x.. ...|...|.y. ...|...|..z If x,y,z = 1,2,3 then there are 133213192 solutions. If x,y,z = 1,5,9 then there are 124117956 solutions.

Therefore selecting x,y,z at random introduces bias.

Whether or not the bias is relevant to a particular experiment depends upon the experiment, of course.

Back to top profile pm

gsf

Posted: Thu Sep 24, 2009 11:24 am Post subject: quote

Joined: 21 Sep 2005
Posts: 3814
Location: NJ USA

I had forgotten that I added an option for dukosu's randomized minimization to my generator
here are the counts by #clues for 1M minimal puzzles, each batch generated in < 1hr
the first batch uses my bottom up generator
(generate a complete grid C, start with an empty grid G, and add clues from C to G until G has one solution)

Code:

```
# sudoku -n1000000 -qFN -gp -m1 -f%v
     82  20
   3464  21
  44715  22
 202218  23
 357974  24
 274236  25
  98095  26
  17420  27
   1698  28
     92  29
      6  30
```

and the second batch uses dukosu's random minimization

Code:

```
# sudoku -n1000000 -qFN -gp -mr1 -f%v
     57  20
   2693  21
  37263  22
 184541  23
 351893  24
 289903  25
 110541  26
  20934  27
   2052  28
    120  29
      3  30
```

Back to top profile pm www

David P Bird

Posted: Thu Sep 24, 2009 11:40 am Post subject: quote

Joined: 16 Sep 2008
Posts: 150
Location: Middle England

Red Ed wrote:

> Therefore selecting x,y,z at random introduces bias.

Not only are there the conventional one-only constraints, but there are additional braiding constraints which have to be satisfied as the grid is filled - this contributes towards the extent of the back-tracking that has to go on in conventional generators, because no simple theorem can cover every braid pattern case that can arise. This two box and three diagonal cells start is as far as I could go and stay clear of these

extra constraints. After that I don't think they can be avoided.

Now suppose you fill in any arrangement of 18 cells and haven't invalidated the puzzle by creating a one-only or braiding conflict, you would be left with much the same problem about different numbers of continuations depending on the next three you chose to set. According to the test you've applied, both methods would have a bias imposed by the particular choice of start cells wouldn't they? I can't see how this interim measurement will produce a different distribution of the puzzles in the end by the two methods.

Back to top profile pm

Red Ed

Posted: Thu Sep 24, 2009 11:56 am Post subject: quote

Joined: 06 Jun 2005
Posts: 728

Suppose you have an algorithm that:

1. based on some fixed pattern of cell positions ...
2. ... randomly fills them then ...
3. ... picks a random completion then ...
4. ... does a "controlled-bias" walk down to an m-clue minimal puzzle.

Suppose in step 2 that there are N equally likely ways of filling the base. For the algorithm to output a particular minimal puzzle, it must:

1. n/a
2. Pick the right base (prob 1/N)
3. Pick the right completion (prob = 1/#completions)
4. Pick the right m-clue subset of 81 clues (prob = 1/choose(81,m)).

Since step 3 depends on #completions then, unless that's constant (which it's usually not), the algorithm is biased.

Unbiased algorithms work by weighting the choice of base by the number of completions.

Back to top profile pm

David P Bird

Posted: Thu Sep 24, 2009 1:28 pm Post subject: quote

Joined: 16 Sep 2008
Posts: 150
Location: Middle England

The proof must be in the pudding here I think.

> **Red Ed wrote:**
>
> Since step 3 depends on #completions then, unless that's constant (which it's usually not), the algorithm is biased.

I have an intuitive feel of the thinking behind this statement and can see how for medical trials etc. this sort of treatment has to be used in analysing the robustness of the experimental design.

However at every step the number of possible completions comes down and it would be possible to graph them if we wanted. Whatever generation method we use no two graphs would be the same after the first few interactions between candidates occurred so whatever we do will suffer from localised bias. Now we can normalise every chance so that they all have an equal probability of success if you want, and

report the distribution accordingly. But the more trials you conduct the less this approach is needed as I see it, as the effect of the localised individual biases observed will average out.

My assertion is that every possible starting combination is equally likely in the flying start system I've described, and between them they make no individual or group of puzzles any more or less likely than any other.

AFTER the flying start then the chances of a candidate/cell selection creating a contradiction will come into play however, and we are at the mercy of the same factors that control your completely random grid generator.

Now all this would be meaningful if we had some measure to form the basis of the distribution we're watching to use as a yardstick, but so far we haven't done that have we? One possibility would be to check the percentage of puzzles with different numbers band containing rope patterns which I believe you can calculate for the whole population.

Back to top profile pm

Red Ed

Posted: Thu Sep 24, 2009 1:48 pm Post subject: quote

Joined: 06 Jun 2005
Posts: 728

> **David P Bird wrote:**
>
> > **Red Ed wrote:**
> >
> > > Since step 3 depends on #completions then, unless that's constant (which it's usually not), the algorithm is biased.
> >
> > I have an intuitive feel of the thinking behind this statement

There should be no intuition needed: it's a straight calculation. Think of any m-clue minimal puzzle. Its pick-me probability is 1/N * 1/#completions * 1/choose(81,m) as described above, where #completions refers to the unique base (result of step 2) corresponding to the puzzle. The algorithm is biased if this probability is non-constant (by definiton).

As I said previously, unbiased algorithms work by weighting their random choices. Any practical algorithm that doesn't do that with carefully chosen (and justified) weights is virtually certain to be biased. It's the same every time. Every generator where someone's said "localised individual biases observed will average out" (or words to that effect) has been biased. Sorry!

Back to top profile pm

David P Bird

Posted: Thu Sep 24, 2009 3:02 pm Post subject: quote

Joined: 16 Sep 2008
Posts: 150
Location: Middle England

> **Red Ed**
> Let's get this straight, are you saying that for every grid you're generating that you are calculating the number of completions possible after every step and weighting the probabilities or are you using average weights?

If you're not counting completions, you're stowing thrones in glass houses. As soon as you have the first few candidate interactions you're on a path to a particular group of end puzzles. Now say a particularly restrictive combination arises early on in the

random process. The number of possible completions drops enormously, and the chance that the next selection will provide a contradiction and produce an aborted run rises very considerably. You can't even include an unsuccessful run in the distribution because you don't know what a typical end result would have been. The number of trials you would need to justify the use of average weights would be enormous because the range of possible completion counts can vary considerably. Are variability and trial size covered in your equations?

The basis of the flying start approach is a) every way of filling in those 21 cells is equally likely and b) there is no possibility of a contradiction between them. I accept that once these are set, there will be different number of possible completions but that's what happens with a hit and miss method of filling in the first 21 cells, and however you deal with that can be used for the randomised continuation process.

I don't see how you can throw out one approach because you suppose the odds don't average out, but then assert that they do for another with a shrug of the shoulders and a smiley "Sorry!"

So what about that trial with an agreed yardstick? The distribution of 2-digit unavoidable sets would be another possibility, but I don't know that we can be sure of what the true distribution should be.

Back to top profile pm

Red Ed

Posted: Thu Sep 24, 2009 10:23 pm Post subject: quote

Joined: 06 Jun 2005
Posts: 728

> **Quote:**
> Let's get this straight, are you saying that for every grid you're generating that you are calculating the number of completions possible after every step and weighting the probabilities or are you using average weights?

The former - here - with some pretabulation.

I couldn't make out what you were after re "yardsticks".

Back to top profile pm

denis_berthier

Posted: Fri Sep 25, 2009 12:03 am Post subject: quote

Joined: 19 Jun 2007
Posts: 829
Location: Paris, France

As it seems we are drifting away from the main topic of this thread, time may have come to assess where we are.
Regarding the previous discussion, I don't have much to say about the generation of unbiased complete grids.
Admittedly, the topic can be understood in a broad sense. Estimating the real (i.e. unbiased) distribution of minimal puzzles may a priori be considered as depending on first having some unbiased source of complete grids.
But it relies on such a source to a different extent, depending on one's specific goals.
This is where it may be useful, for practical purposes, to recall where my approach and Red Ed's differed (that is to say, before he finally adopted my controlled-bias approach).

Red Ed was looking for the absolute numbers of minimal puzzles with n clues. As a result, (he thought) he needed an unbiased source of complete grids and he had to deal with astronomical numbers (see here:

http://www.sudoku.com/boards/viewtopic.php?t=4771&postdays=0&postorder=asc&start=57, where he concluded that the mean number of clues is 26.4 from a sample in which only **0.49 %** of the puzzles were above this mean).
Notice that, contrary to what he assumed, starting from an unbiased source of complete grids is not a guarantee that the minimal puzzles thus obtained are unbiased. As I've said several times and as was reminded in some posts above, the equation "1 complete grid => 1 minimal puzzle" is in itself a source of bias. Notice that this also invalidates Red Ed's method of testing puzzles bias through the detection of some bias in their solution grids.

I was looking only for the number-of-clues distribution, i.e. for relative proportions. I could therefore introduce the notions of a controlled-bias generator (associated with a simple formula for the P(n+1)/P(n) and with well defined and non-astronomical correction factors). This new generator was easily programmed by Eleven as a modification of top-down suexg. This is a slow generator, but at least it outputs minimal puzzles (contrary to what Red Ed was doing previously), **21.5 %** of which have a number of clues above the unbiased mean value. Moreover, these puzzles can be used for other purposes, such as complexity computations, because they are unbiased for each fixed number of clues.

One can imagine that the impact of the complete grid generator used by the controlled-bias generator is important here also. And it is certainly the case that some forms of systematic bias in the complete grids couldn't be eliminated by the controlled-bias algorithm (imagine we fix the values of the first block).
But I've previously given several intuitive reasons why the controlled-bias algorithm can eliminate some bias in the complete grids (even large). This intuition has been confirmed (ironically, by Red Ed, while he was trying to prove the contrary):

Red Ed wrote:

> **denis_berthier wrote:**
>
> > You said A is your "unbiased generator" and B is suexg's. So suexg would produce complete grids with 20% more minimals than normal, right?
>
> Yep, the bias is real.

Red Ed wrote:

> **denis_berthier wrote:**
>
> > even a strong (20%) bias in the mean number of minimals of a stream of complete grids used as input to a controlled-bias generator of minimal puzzles has only a very small impact on the distribution of clues of the output.
>
> Certainly appears to be true for the *suexg* input.

Now, what can we head for? I imagined a small non exhaustive list:

1) Speed improvement of the controlled-bias generator :
- deletion part: it seems from the above discussion that, after the easy (and innocuous) simplification of deleting the first 46 clues with no testing, nothing both easy and efficient can be done;

- generation part: can we accept some systematic bias in the complete grids, such as fixing the values of the first block ? This is equivalent to adopting a (partial) canonical representation (renaming the clues). As such it is harmless for the distribution of clues problem; but accepting it is a matter of taste. If the global gain is below 25%, I personally wouldn't accept it.

2) Check more extensively that the distribution-of-clues results don't depend on the generator of complete grids.
- this has been partially done with 3 existing generators of complete grids: suexg's, Red Ed's and Mike's (on a small sample). It would be interesting to do it also with Allan's generator, which is based on very different principles.
- modify the existing generators: several ideas have been given above. If we are unable to prove their (un-)biasedness, there is still the possibility of implementing them, generating puzzles with them and testing the compatibility of the distributions.

Back to top profile pm www

eleven

Joined: 10 Feb 2008
Posts: 485

Posted: Fri Sep 25, 2009 12:08 am Post subject: quote

Red Ed wrote:

> I'd be happy to test the output if you post the code.

```
// Test version with 21 cells prefilled according to David B. Bird
// Program to generate puzzles with controlled bias, see
// http://www.sudoku.com/boards/viewtopic.php?t=14615
// Modified version of the top down generator by dukuso (sterten@aol.com),
// which is public domain

#include <stdio.h>
#include <stdlib.h>
#define MWC
((zr=36969*(zr&65535)+(zr>>16))^(wr=18000*(wr&65535)+(wr>>16)))
unsigned zr=362436069, wr=521288629;
int Rows[325],Cols[730],Row[325][10],Col[730][5],Ur[730],Uc[325],V[325],W[325];
int P[88],A[88],A0[88],C[88],I[88],Two[888];
int b,w,f,s1,m0,c1,c2,r1,l,i1,m1,m2,a,p,i,j,k,r,c,d,n=729,m=324,x,y,s;
int mi1,mi2,q7,part,nt,nodes,seed,solutions,min,samples,sam1,clues;
char L[11]=".123456789";
FILE *file;
int solve();
double cnt = 0.0;
int nClues;

int prefill()
{
int i,k,o[10];
for(i=1;i<=9;i++){do k=(MWC>>9)&15;while (k>=i);k++;o[i]=o[k];o[k]=i;}
for (i=0;i<9;i++) A[i%3+9*(i/3)+1]=o[i+1];
for(i=1;i<=9;i++){do k=(MWC>>9)&15;while(k>=i);k++;o[i]=o[k];o[k]=i;}
for (i=0;i<9;i++) A[i%3+31+9*(i/3)]=o[i+1];
for(i=1;i<=9;i++){do k=(MWC>>9)&15;while(k>=i);k++;o[i]=o[k];o[k]=i;}
for (i=0;i<3;i++) A[61+10*i]=o[i+1];
}

int main(int argc,char*argv[]){
if(argc<3){printf("\nusage:<program name> random-seed max-puzzles [file with
grids] \n\n");
return(1);}
```

```
sscanf(argv[1],"%i",&seed);zr^=seed;wr+=seed;
sscanf(argv[2],"%i",&samples);
if(argc>3)if((file=fopen(argv[3],"rb"))==NULL)
{printf("\ncan't find file %s\n",argv[3]);return(1);}

for(i=0;i<888;i++){j=1;while(j<=i)j+=j;Two[i]=j-1;}
for(i=1;i<=81;i++)A0[i]=0;

r=0;for(x=1;x<=9;x++)for(y=1;y<=9;y++)for(s=1;s<=9;s++){
r++;Cols[r]=4;k=3;
Col[r][1]=x*9-9+y;
Col[r][2]=(k*((x-1)/k)+(y-1)/3)*9+s+81*1;
Col[r][3]=x*9-9+s+81*2;
Col[r][4]=y*9-9+s+81*3;}
for(c=1;c<=m;c++)Rows[c]=0;
for(r=1;r<=n;r++)for(c=1;c<=Cols[r];c++){
a=Col[r][c];Rows[a]++;Row[a][Rows[a]]=r;}


m6:if(argc>3)
for(i=1;i<=81;i++){
m6a:A0[i]=fgetc(file)-48;if(feof(file))return(8);
if(A0[i]==-2)A0[i]=0;if(A0[i]<0 || A0[i]>9)goto m6a;}

sam1=0;
m0s:sam1++;if(sam1>samples){if(argc>3)goto m6;return(0);}

// dont call prefill() to get the original version
m0:cnt+=1.0;for(i=1;i<=81;i++)A[i]=A0[i];part=0;if(argc<4){prefill();solve();}
// m0:cnt+=1.0;for(i=1;i<=81;i++)A[i]=A0[i];part=0;if(argc<4)solve();
nClues=81;

part++;

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]];if(s1){A[P[i1]]=0;
if(--nClues==34){if(solve()>1)goto m0;}if(nClues<
34&&solve()>1){A[P[i1]]=s1;break;}}}


i=++i1;for(i1=i;i1<=81;i1++){s1=A[P[i1]];if(s1){A[P[i1]]=0;if(solve()<2)goto m0;
A[P[i1]]=s1;}}

for(i=1;i<=81;i++)printf("%c",L[A[i]]);printf(" %g\n", cnt);
fflush(stdout);
goto m0s;}


int solve(){
for(i=0;i<=n;i++)Ur[i]=0;for(i=0;i<=m;i++)Uc[i]=0;
clues=0;for(i=1;i<=81;i++)
if(A[i]){clues++;r=i*9-9+A[i];
for(j=1;j<=Cols[r];j++){d=Col[r][j];if(Uc[d])return -1;Uc[d]++;
for(k=1;k<=Rows[d];k++){Ur[Row[d][k]]++;}}}
for(c=1;c<=m;c++){V[c]=0;for(r=1;r<=Rows[c];r++)if(Ur[Row[c][r]]==0)V[c]++;}
i=clues;m0=0;m1=0;solutions=0;if(i==81)return 1;
m2:i++;I[i]=0;min=n+1;if(i>81 || m0)goto m4;
if(m1){C[i]=m1;goto m3;}
w=0;for(c=1;c<=m;c++)if(!Uc[c]) { if(V[c]<2){C[i]=c;goto m3;}
if(V[c]<=min){w++;W[w]=c;};
if(V[c]<min){w=1;W[w]=c;min=V[c];} }
```

```
mr:c2=MWC&Two[w];if(c2>=w)goto mr;C[i]=W[c2+1];
m3:c=C[i];I[i]++;if(I[i]>Rows[c])goto m4;
r=Row[c][I[i]];if(Ur[r])goto m3;m0=0;m1=0;
if(part==0){j=9;k=81;x=(r-1)/k+1;y=((r-1)%k)/j+1;s=(r-1)%j+1;A[x*9-
9+y]=s;P[x*9-9+y]=i;}
for(j=1;j<=Cols[r];j++){c1=Col[r][j];Uc[c1]++;}
for(j=1;j<=Cols[r];j++){c1=Col[r][j];
for(k=1;k<=Rows[c1];k++){r1=Row[c1][k];Ur[r1]++;if(Ur[r1]==1)
for(l=1;l<=Cols[r1];l++){c2=Col[r1][l];V[c2]--;
if(Uc[c2]+V[c2]<1)m0=c2;if(Uc[c2]==0 && V[c2]<2)m1=c2;}}}
if(i==81){solutions++;if(solutions>1)return 2;if(part==0)return 1;}
goto m2;
m4:i--;c=C[i];r=Row[c][I[i]];if(i==clues)goto m9;
for(j=1;j<=Cols[r];j++){c1=Col[r][j];Uc[c1]--;
for(k=1;k<=Rows[c1];k++){r1=Row[c1][k];Ur[r1]--;
if(Ur[r1]==0)for(l=1;l<=Cols[r1];l++){c2=Col[r1][l];V[c2]++;}}}
if(i>clues)goto m3;
m9:return solutions;}
```

Back to top profile pm

David P Bird

Posted: Fri Sep 25, 2009 12:18 am Post subject: quote

Joined: 16 Sep 2008
Posts: 150
Location: Middle England

Red Ed, It will take me a while to do the homework you've set me regarding how to count completions, but the yardstick suggestion is simple:

If we know what the distribution of some characterisation measure should be for an entire population, we can see if the sample we're using is biased or not by checking if the sample distribution is in accordance with what it should be - a nice straightforward approach.

If you want to see some examples of some real bias, check how two competing clinical trials use statistics to favour their cases. You'll then understand my reservations about the use and misuse of statistical tests.

While I'm doing my homework, perhaps you could turn your mind around about using flying start approaches - not what's wrong with them, but the possibilities of using one to take advantage of the time savings that could be had. It doesn't have to be the one I've suggested.

Back to top profile pm

eleven

Posted: Fri Sep 25, 2009 1:04 am Post subject: quote

Joined: 10 Feb 2008
Posts: 485

Prefilling only 2 diagonal boxes randomly before the rest of the grid generation - done by dukuso with his implementation of a dancing links (DLX) solver - would save almost the same time, the grid generation part would be about 1.7 times faster (1.8 times faster with the 3 additional clues).

I am sensibilised enough for bias to know that also fixing one box would introduce bias (because grids with box automorphism would be preferred). But without the 3 (or 2) clues in the 3rd box it looks less suspicious 

Back to top profile pm

Red Ed

Posted: Fri Sep 25, 2009 9:41 am Post subject: quote

Joined: 06 Jun 2005
Posts: 728

Denis, why the attempt at points scoring? Can't we just focus on taking the technology forward?

Well now I'll have to respond.

> **denis_berthier wrote:**
>
> recall where my approach and Red Ed's differed (that is to say, before he finally adopted my controlled-bias approach).

No, I have not adopted your approach. First, I don't aim to generate puzzles. Second, the technique I use for estimating the number-of-clues distribution was -- and remains (if I ever run it again) -- the c-clue-subsets-of-s-clue-subgrids method.

> **Quote:**
>
> (he thought) he needed an unbiased source of complete grids and he had to deal with astronomical numbers (see here: http://www.sudoku.com/boards/viewtopic.php?t=4771&postdays=0&postorder=asc&start=57, where he concluded that the mean number of clues is 26.4 from a sample in which only **0.49 %** of the puzzles were above this mean).

Read more carefully: I said that I thought my method (the estimator, not the solution grid source) was unbiased. btw, why are you scared of "astronomical" numbers?

> **Quote:**
>
> Notice that, contrary to what he assumed, starting from an unbiased source of complete grids is not a guarantee that the minimal puzzles thus obtained are unbiased.

I've never assumed that, as you'll discover if you attempt to provide any evidence.

> **Quote:**
>
> this also invalidates Red Ed's method of testing puzzles bias through the detection of some bias in their solution grids.

I'm not testing puzzle bias, I'm only testing solution grid bias. What on earth makes you think my bias tester applies to puzzles?

> **Quote:**
>
> This new generator was easily programmed by Eleven as a modification of top-down suexg. This is a slow generator, but at least it outputs minimal puzzles (contrary to what Red Ed was doing previously)

I was generating minimal puzzles, too, albeit for a different purpose. What on earth makes you think I wasn't?

> **Quote:**
>
> these puzzles can be used for other purposes, such as complexity computations, because they are unbiased for each fixed number of clues.

As can mine. Your puzzles have the additional nice property that they are *uncorrelated*, but you have - rightly - not used that property in the computation of complexity rating means. My puzzles can be used for that purpose too. Which converges quicker for complexity stats ...? - I honestly don't know, as I've not even started to investigate that angle.

Quote:

> I've previously given several intuitive reasons why the controlled-bias algorithm can eliminate some bias in the complete grids (even large). This intuition has been confirmed (ironically, by Red Ed, while he was trying to prove the contrary)

Why can't you grasp this? As I keep explaining, the only thing I was trying to prove was that the bias was real. I agree that the size (small) of the bias was a surprise to me, but in no sense does this mean that I failed to prove what I set out to prove.

Quote:

> Now, what can we head for? I imagined a small non exhaustive list:

Alrighty, now we're talking! I'm happy to see that you managed to speed up the deletion part as much as you did: I don't think we can do much better than that. As for distribution sensitivity to the grids source - I'll be an interested observer, (now!) expecting the relative insensitivity to be demonstrated again. I won't take part by burning CPU, but I'll certainly ponder a little more why the results should be as they are.

Back to top profile pm

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 ... , 20, 21, 22 Next

Page 21 of 22

Jump to: General/puzzle Go