THE REAL DISTRIBUTION OF MINIMAL PUZZLES

Author	Message
denis_berthier Joined: 19 Jun 2007 Posts: 822 Location: Paris, France	Posted: Sun Sep 20, 2009 10:54 pm Post subject: quote **Red Ed wrote:** the path-probing part now runs 4.9 times quicker. Owing to the overhead of generating the solution grid in the first place remaining unchanged, that translates to a 3.5 times speed-up per puzzle generated. When you start suexg-cb with a given fixed seed, do you get the same sequence of puzzles before and after your optimisations?
Back to top	profile pm www
m_b_metcalf Joined: 15 May 2006 Posts: 2357 Location: Berlin	Posted: Mon Sep 21, 2009 1:29 am Post subject: quote **denis_berthier wrote:** **m_b_metcalf wrote:** Please remind of, or point me to, the algorithm for a 'controlled-bias' generator. Either my web page (the most up to date place) http://carva.org/denis.berthier/HLS/Classification, or here : http://www.sudoku.com/boards/viewtopic.php?t=14615&start=134 Thanks. Before I look into this, can you tell me what the approximate yield is (minimal puzzles/million grids, say)? I'm not willing, yet, to commit to a huge run. Thanks, Mike Metcalf
Back to top	profile pm
David P Bird Joined: 16 Sep 2008 Posts: 139 Location: Middle England	Posted: Mon Sep 21, 2009 1:59 am Post subject: quote **Denis** Thank you for your explanation. Scheme II identifies clues which become required in the final puzzle grid because all their equivalents have been eliminated. This would allow these clues to be protected if we wanted. In the later stages, when we are quite close to a success, the solver will have been run, and will have identified the clues still required and the ones which aren't, when we've located a valid minimum set. If the current run is leading to a puzzle with a high number of required clues, the probability that a required cell will be removed is proportionately much higher, and the result will be mostly be lost. I believe this means the frequency of 17 clue puzzles will always be exaggerated in comparison to those with 30 clues. Integral to your analysis is the distribution of possible clue sets for any particular starting grid. The final clue counts depend on how much or how little information provided by each clue is duplicated. Some of this is dependent on the particular clues combinations eventually provided, but the interactions between the 9 digit patterns that exist in the starting grid govern how this can be achieved.

It might be better to start with a sample of different starting grids and then from each of these to explore the range of clue sets that they could provide using repeated runs. I would imagine that in this design it would be safer to preserve required cells and eliminate those that are not required (which would produce a result for every trial) because each puzzle would contribute equally to the final distribution we would build.

Even if this isn't the case and it is judged that this approach would favour clue sets with high counts it might still be worth doing to help assess our other results.

Back to top profile pm

denis_berthier Posted: Mon Sep 21, 2009 3:10 am Post subject: quote

Joined: 19 Jun 2007
Posts: 822
Location: Paris, France

m_b_metcalf wrote:

> Before I look into this, can you tell me what the approximate yield is (minimal puzzles/million grids, say)?

About 1 minimal puzzle for 225000 complete grids. On a standard 3 Ghz processor, this gives about 200 puzzles per day.
I've spent much computing power on this because it is currently the only way of getting unbiased statistics.

Back to top profile pm www

denis_berthier Posted: Mon Sep 21, 2009 4:51 am Post subject: quote

Joined: 19 Jun 2007
Posts: 822
Location: Paris, France

Here is a very simple optimisation of suexg-cb: randomly delete the first 46 clues without doing any test.
We now know (from the results with the "standard" version of suexg-cb) that the probability of finding an n-clue puzzle with suexg-cb, n> 34, is very small. Discarding them a priori can't change significantly the distribution of clues.

Back to top profile pm www

m_b_metcalf Posted: Mon Sep 21, 2009 5:37 am Post subject: quote

Joined: 15 May 2006
Posts: 2357
Location: Berlin

denis_berthier wrote:

> **m_b_metcalf wrote:**
>
> > Before I look into this, can you tell me what the approximate yield is (minimal puzzles/million grids, say)?
>
> About 1 minimal puzzle for 225000 complete grids. On a standard 3 Ghz processor, this gives about 200 puzzles per day.
> I've spent much computing power on this because it is currently the only way of getting unbiased statistics.

This doesn't fit with my generator. You say elsewhere that a standard top-down generator works thus:

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
```

However, mine works thus:

Code:

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

Do I misunderstand something? In particular your step 2b?

Regards,

Mike Metcalf

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 822
Location: Paris, France

Posted: Mon Sep 21, 2009 7:42 am Post subject: quote

Mike,

I think the 2 descriptions are identical (yours is cleaner)

"return P2" in my step 2b means "output P2 and exit"

Back to top profile pm www

m_b_metcalf

Joined: 15 May 2006
Posts: 2357
Location: Berlin

Posted: Mon Sep 21, 2009 7:49 am Post subject: quote

> **denis_berthier wrote:**
> I think the 2 descriptions are identical (yours is cleaner)
>
> "return P2" in my step 2b means "output P2 and exit"

Denis,
It's the "if P2 is minimal" I'm worried about. This requires a pass over all the clues in P2 until one is found that is redundant, if any. That would be expensive.

Maybe you would prefer to modify *my* description to tell me what you mean by controlled bias.

Thanks,

Mike

Back to top profile pm

gsf

Joined: 21 Sep 2005
Posts: 3807
Location: NJ USA

Posted: Mon Sep 21, 2009 8:07 am Post subject: quote

[quote="m_b_metcalf"]

Code:

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

Mike, two questions:

you have "2a) choose one clue randomly" inside a "loop over all cells" -- how does the loop affect the choice?

how do you break out of the loop to get to "3)"?

Back to top profile pm www

Red Ed

Joined: 06 Jun 2005
Posts: 717

Posted: Mon Sep 21, 2009 8:15 am Post subject: quote

> **denis_berthier wrote:**
> > **Red Ed wrote:**
> > the path-probing part now runs 4.9 times quicker. Owing to the overhead of generating the solution grid in the first place remaining unchanged, that translates to a 3.5 times speed-up per puzzle generated.
>
> When you start suexg-cb with a given fixed seed, do you get the same sequence of puzzles before

> and after your optimisations?

If I modify suexg-cb.c to keep RNG state for solution grid generation separate from that for subgrid solving, then yes. (And if not then obviously no because the subgrid-solving strategies differ between exhaustive top-down probing, which is what suexg-cb.c does, and optimal probing.)

This suggestion -

Quote:

> Here is a very simple optimisation of suexg-cb: randomly delete the first 46 clues without doing any test.

- will work of course, ~~but it won't be as quick as optimal path probing.~~
EDIT: oops! - it *will* be as quick - in fact probably quicker - because it combines steps 3 & 4. Well spotted.

Last edited by Red Ed on Mon Sep 21, 2009 10:11 am; edited 1 time in total

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 822
Location: Paris, France

Posted: Mon Sep 21, 2009 8:23 am Post subject: quote

m_b_metcalf wrote:

> Maybe you would prefer to modify *my* description to tell me what you mean by controlled bias.

Let's write a compromise between the 2.

This is the classical top-down procedure for generating ONE minimal puzzle (it has to be iterated for as many puzzles as desired):

1) generate a random complete grid P
2a) choose one clue randomly from **P** and delete it, you get a puzzle P2
2b) if P2 has several solutions, (restore that clue and) GOTO 2a
2c) if P2 is minimal, printout P2 and exit the whole procedure
2d) otherwise (P2 has a single solution but is not minimal), set P=P2 and GOTO 2a

This is the controlled-bias procedure for generating ONE or ZERO minimal puzzle (it has to be iterated for as many puzzles as desired):

1) generate a random complete grid P
2a) choose one clue randomly from **P** and delete it, you get a puzzle P2
2b) if P2 has several solutions, exit the whole procedure (no output)
2c) if P2 is minimal, printout P2 and exit the whole procedure
2d) otherwise (P2 has a single solution but is not minimal), set P=P2 and GOTO 2a

Back to top profile pm www

eleven

Joined: 10 Feb 2008
Posts: 478

Posted: Mon Sep 21, 2009 8:52 am Post subject: quote

denis_berthier wrote:

> Here is a very simple optimisation of suexg-cb: randomly delete the first 46 clues without doing any test.

You can try this:

after

Code:

```
double cnt = 0.0;
```

add the line

Code:

```
int nClues;
```

after

Code:

```
  m0:cnt+=1.0;for(i=1;i<=81;i++)A[i]=A0[i];part=0;if(argc<4)solve();
```

add

Code:

```
nClues=81;
```

change

Code:

```
for(i1=1;i1<=81;i1++){s1=A[P[i1]];if(s1){A[P[i1]]=0;if(solve()>1){A[P[i1]]=s1;break;}}}
```

to

Code:

```
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;}}}
```

Mike:
For the cb version you must restart (goto 1), if you get a multisolution puzzle and the puzzle with the last deleted clue reinserted is not minimal.
[Added to be clearer:]

Code:

```
1) Generate a random complete grid
2) Delete clues, until you have a multisolution puzzle
3) Reinsert the last clue
4) If the puzzle is minimal, report it
5) Goto 1.
```

Last edited by eleven on Mon Sep 21, 2009 9:44 am; edited 1 time in total

Back to top profile pm

m_b_metcalf Posted: Mon Sep 21, 2009 9:42 am Post subject: quote

Joined: 15 May 2006
Posts: 2357
Location: Berlin

gsf wrote:

> **m_b_metcalf wrote:**
>
> > **Code:**
> >
> > ```
> > 1) generate a random complete grid
> > 2) loop over all cells:
> >      let P be the current puzzle
> >      2a) choose one clue randomly from P and delete it, you
> > get a puzzle P2
> >      2b) if P2 has several solutions, restore the clue and
> > GOTO 2a
> >      2c) otherwise, set P=P2
> >    end loop
> > 3) return P (which is minimal).
> > ```
>
> Mike, two questions:
>
> you have "2a) choose one clue randomly" inside a "loop over all cells" -- how does the loop affect the choice?
>
> how do you break out of the loop to get to "3)"?

Glenn,

I copied the term loop from Denis' description. It means looping over a rank-1 array containing the numbers 1 to 81 in random order. These are used to index the cells. The loop then just terminates. Probably better to say 'cycle' rather than 'GOTO 2a'.

Regards,

Mike

Back to top profile pm

JPF

Joined: 06 Dec 2005
Posts: 2861
Location: Paris, France

Posted: Mon Sep 21, 2009 10:08 am Post subject: quote

Here are, for some sets of minimal puzzles, the ratio (%) : singles / total number of puzzles

Code:

```
               Gordon    suexg14-0_1M    Metcalf#5   rabrnd1m


Sample         47793       1000000         64410     1000000

 Clues

    17          44.7

    18

    19                       50.0

    20                       46.6         47.4        50.0

    21                       48.3         45.7        44.3

    22                       48.4         48.3        45.7

    23                       48.1         48.0        45.1

    24                       46.4         45.9        43.7

    25                       43.1         41.8        40.6

    26                       37.9         39.2        36.3

    27                       31.6         34.7        30.8

    28                       22.5         19.0        23.5

    29                        8.3                     20.5

    30                       50.0                     12.5
```

JPF

Back to top profile pm

Red Ed

Joined: 06 Jun 2005
Posts: 717

Posted: Mon Sep 21, 2009 10:15 am Post subject: quote

My edit will have been lost as posts shot past recently - but I realise now that Denis' suggestion to drop the first 46 (say) clues will be quicker than optimal path probing. OPP *is* still optimal for step 3 as stated, but the fact I missed was that step 3 as stated achieves more (with more work) than necessary. Duh.

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 ... , 17, 18, 19 Next

Page 18 of 19

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