Sudoku Players' Forums

Rating rules / Puzzles. Ordering the rules

Goto page Previous 1, 2, 3 ... , 22, 23, 24 Next

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

View previous topic :: View next topic

Author | **Message**

eleven

Posted: Wed Jul 08, 2009 2:53 pm Post subject: quote

Joined: 10 Feb 2008
Posts: 325

denis_berthier wrote:

> But the version of SE I have (downloaded from gsf's page, with serate added) works fine and gives 7.1

Ah, thank you, it was fixed in the last version 1.2.1 (i had 1.2).

Changes wrote:

> Sudoku Explainer: Changes since version 1.2
> ==========
> Added an additional check to fix false positives in BUG detection

Back to top profile pm

denis_berthier

Posted: Wed Jul 08, 2009 2:55 pm Post subject: quote edit

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

m_b_metcalf wrote:

> **denis_berthier wrote:**
>
> > **Mike**:
> > - do you allow me to put your 64,000 bottom-up collection on my web pages? (with a reference to you, of course)
> > - is the description of your generator OK?
>
> Denis, yes and yes.

Done.
I don't think I'll be able to consider larger collections, because my computing resources are limited and holidays are approaching.
Also, I think we wouldn't learn much more from the existing generators.
If a generator of a completely new kind appeared, it would be interesting to add it to our list and resume these comparisons.

Allan,
One of the big surprises I forgot to mention is about your generator. As the generation phase is very different from all the other generators, I thought it would lead to different results. But it is incredibly close to sudogen0_1M.
What seems the most important factor is not how the complete grids are (randomly) generated but rather the fact that we build complete grids before we start deleting redundant clues.
I nevertheless continue computations for a little more puzzles.

Back to top profile pm www

ronk

Posted: Wed Jul 08, 2009 3:45 pm Post subject: quote

*** off-topic ***

Joined: 02 Nov 2005
Posts: 2390
Location: Southeastern USA

eleven wrote:

> if you generated suexg14-0_1M with seed 0, and let it rate by Sudoku Explainer, keep an eye on puzzle nr. 737294. SE failed to rate it due to a bug in its BUG algorithm.
> The correct rating would be 7.1.
>
> **Code:**
>
> ```
> ........5.23....8...6.1..2...8.....9..4.3.2..6.9..5..7..59.7.....7...2..3.
> +----------------+----------------+----------------+
> | 48   1    7   | 3    68   2    | 46   9    5    |
> | 5    2    3   | 7    9    46   | 46   8    1    |
> | 9    48   6   | 5    1    48   | 7    2    3    |
> +----------------+----------------+----------------+
> | 2    5    8   | 46   7    1    | 3    46   9    |
> | 1    7    4   | 68   3    9    | 2    5    68   |
> | 6    3    9   | 2    48   5    | 18   14   7    |
> +----------------+----------------+----------------+
> | 3   #468  5   | 9   #468  7    | 1-8  16   2    |
> | 48   68   2   | 1    5    3    | 9    7   #468  |
> | 7    9    1   | 48   2    68   | 5    3   #468  |
> +----------------+----------------+----------------+
> ```
>
> It says, that in one of the marked 3-value cells there must be 8, otherwise there would be a BUG. But this is wrong.
> **ronk** or someone else, can you explain, why the rule of thumb (2 more in the line/col/box) does not work here ?

After removing all the presumed extra candidates, every candidate must exist exactly twice (or not at all) in each row, column and box. For the above, if all four of the presumed extra 8s were removed, there would still be three 6s in each of r7, c9 and b9.

Back to top profile pm

eleven Posted: Wed Jul 08, 2009 4:33 pm Post subject: quote

Joined: 10 Feb 2008
Posts: 325

ronk wrote:

> *** off-topic ***
> After removing all the presumed extra candidates, every candidate must exist exactly twice (or not at all) in each row, column and box. For the above, if all four of the presumed extra 8s were removed, there would still be three 6s in each of r7, c9 and b9.

Got it, thanks. I have to be careful, when BUG cells are in the same unit.

Back to top profile pm

StrmCkr Posted: Wed Jul 08, 2009 6:53 pm Post subject: quote

Joined: 05 Sep 2006
Posts: 473
Location: Winterpeg

my generator: written in free {turbo}pascal..

uses the coded solving techniques to verify the grid is unique. and for deleting extra clues.. thats its downfall so far...

notes: setpm updates the grid. wpm writes it to screen.
initiate resets everything back to zero.

wpm also calculates the number of sovled cells,
and setpm calulates the #of pm's.

pm:[0..8,0..8,1..9] stores all the pencilmarks. {as x,y,N}
s:[0..8,0..8,0] stores the active candidates.

solve functions reduces pm states.

Code:

```
procedure generate;
var
```

```
xn,yn,n,ns,temp,xn2,yn2:integer;
gen: array[0..8,0..8,0..0] of integer;

begin

for xn:= 0 to 8 do
 for yn:= 0 to 8 do
  gen[xn,yn,0]:=0;

initiate;
setpm;
wpm;

while (( count < 81) ) do

begin
randomize;
temp:=0;

 repeat
   xn:= (random(9));
   yn:= (random(9));

   for n:= 1 to 9 do
    if pm[xn,yn,N]>0
     then temp:=1+temp;

 until (s[xn,yn,0]=0) and (temp>1);

 if s[xn,yn,0]=0
 then

 repeat
     n:=(random(9));

 until (pm[xn,yn,(N+1)]=(N+1));

 s[xn,yn,0]:=(N+1);

 gen[xn,yn,0]:=(N+1);
 setpm;

 solve;
 setpm;
 cal;

 if ((countpm=0) and (count <> 81))

  then
  begin
   initiate;

    for xn:= 0 to 8 do
     for yn:= 0 to 8 do
      gen[xn,yn,0]:=0;

    setpm;
    cal;
    end;

  for xn:= 0 to 8 do
   for yn:= 0 to 8 do
     if (nm[xn,yn,1]=0) and (s[xn,yn,0]=0)
      then
         begin
           initiate;

         for xn2:= 0 to 8 do
          for yn2:= 0 to 8 do
           gen[xn2,yn2,0]:=0;

           setpm;
           cal;
          end;

end;

n:=0;

repeat
```

```
initiate;

for xn:= 0 to 8 do
 for yn:= 0 to 8 do
   s[xn,yn,0]:=(gen[xn,yn,0]);

cal;
ns:=count;

 randomize;

  repeat
   xn:= (random(9));
   yn:= (random(9));

  until (s[xn,yn,0]>0);

  s[xn,yn,0]:=0;

  setpm;
   solve;
    setpm;
     cal;

  if count=81
  then

   begin
    gen[xn,yn,0]:=0;
    n:=0;
   end

  else
  n:=(1+N);

until (n=ns);

initiate;

for xn:= 0 to 8 do
 for yn:= 0 to 8 do
   s[xn,yn,0]:=(gen[xn,yn,0]);

setpm;
wpm;
end;
```

Last edited by StrmCkr on Wed Jul 08, 2009 7:06 pm; edited 3 times in total

Back to top profile pm

ronk

Posted: Wed Jul 08, 2009 6:53 pm Post subject: quote

Joined: 02 Nov 2005
Posts: 2390
Location: Southeastern USA

Allan Barker wrote:

> **denis_berthier wrote:**
>
> > What would be your length for a Naked Pair?
>
> Now I see what you mean. One answer is 2, because the definition of a naked pair is based on two cells that both contain a truth. The best answer is perhaps that the number of truths used in a logical deduction are what matter. In the above case, the NP has cleared the grid leaving behind a bi-local pair and only one is needed by the logic. Normally such an NP wouldn't be in the logic but if it were there for pratical reasons then the bi-local can be substituted. (Good example!).

Borrowing **Denis'** term, it's a limit case in "complementarity." The complement to an AHS(1) is an LS(2) in the above example. Examples where the complement to an AHS(1) is LS(3) or LS(4), etc. are easy to find. As before, the **N** of LS(**N**) is the number of cells in the set.

[edit: To add scope to *borrowing*, quoted the term *complementarity*]

Last edited by ronk on Wed Jul 08, 2009 8:02 pm; edited 1 time in total

Back to top profile pm

denis_berthier

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

Posted: Wed Jul 08, 2009 7:33 pm Post subject: quote edit

> **ronk wrote:**
>
> Borrowing **Denis'** term, it's a limit case in complementarity. The complement to an AHS(1) is an LS(2) in the above example. Examples where the complement to an AHS(1) is LS(3) or LS(4), etc. are easy to find. As before, the **N** of LS(**N**) is the number of cells in the set.

I doubt ever speaking of "a limit case in complementarity".
I've never spoken of complementarity for anything else than pure (i.e. not almost or almost-almost...) Naked or Hidden Subsets. For whips or braids (pure or with ALS), that's all we ever need.
I'm not even sure there is a consistent definition of complementarity for the almost-somethings.

But I'm almost certain one can give similar definitions for Allan's patterns, consistent with all the cases for which these patterns can be interpreted as whips(ALS) or braids(ALS). As I'm not very familiar with the technical details of his approach, I'll wait for his analysis of the question.
Anyway, the general rule that assigns a length to a pattern should be based on the general form of this pattern and not on "limit cases".

Back to top profile pm www

denis_berthier

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

Posted: Thu Jul 09, 2009 4:24 am Post subject: quote edit

I think there's one more thing we could learn.

The previous computations have shown that puzzles generated bottom-up tend to be easier than puzzles generated top-down.
This remains very mysterious.
The computations have also shown that bottom-up puzzles tend to have fewer clues; but, in spite of the small trend more clues -> harder, this is not enough to explain the mystery, because the above tendency is true for any fixed number of clues.

I imagined the following double generation process:

1) take the first part of a bottom-up generator and get a non complete grid G1 with a unique solution
2) with the second part of the generator obtain a minimal puzzle P1
3) let G2 be the complete (solution) grid obtained from G1
4) with exactly the same second part (this is very important) obtain a minimal puzzle P2

Question: is there any correlation between P1 and P2:
- number of clues
- SER
- NRCZT

Mike, do you think your bottom-up generator could be adapted to do this? (10,000 couples of puzzles would be enough for such computations.)

Back to top profile pm www

StrmCkr

Joined: 05 Sep 2006
Posts: 473
Location: Winterpeg

Posted: Thu Jul 09, 2009 6:25 am Post subject: quote

> **Quote:**
>
> This remains very mysterious.

the mystery is the logic set used to solve the puzzle.

in top down builds you remove clues from a fixed point until logic sets are in place to produces a valid grid, which can mean harder move sets to.

in bottom up builds generator has less chance of creating complex logic sets until it hits a state that is unique.

my generator code has been producing a few 7+ er rated puzzles with few clues and some with many,

because it uses logic to reduce the pm states via logic before it places the next randomly generated digit.

which gives it a increased chance of randomly creating a harder puzzle.

Back to top profile pm

denis_berthier

Posted: Thu Jul 09, 2009 6:35 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

StrmCkr wrote:

> **Quote:**
> This remains very mysterious.
>
> the mystery is the logic set used to solve the puzzle.

Not sure I understand this part of your post.
It seems clear to me that the only thing that counts for the solver part of any generator is its output:
- 0 (0 solution)
- 1 (1 solution)
- 2 (at least 2 solutions).
How this result is obtained doesn't matter.

Or am I missing anything? Are you suggesting that the solver part of some of the generators that have been discussed here don't produce the correct 0/1/2 value?

I understand though that the choice of the next clue may have an impact on the difficulty of the puzzles - but aren't you thus introducing a strong bias?

Back to top profile pm www

m_b_metcalf

Posted: Thu Jul 09, 2009 6:45 am Post subject: quote

Joined: 15 May 2006
Posts: 2191
Location: Berlin

StrmCkr wrote:

> **Quote:**
> This remains very mysterious.
>
> the mystery is the logic set used to solve the puzzle.

My generators use no logic, rather backtracking preceded by singles solving, solely for efficiency reasons (to prune the search tree).

Regards,

Mike Metcalf

Back to top profile pm

StrmCkr

Posted: Thu Jul 09, 2009 6:53 am Post subject: quote

Joined: 05 Sep 2006
Posts: 473
Location: Winterpeg

Quote:

> Or am I missing anything? Are you suggesting that the solver part of some of the generators that have been discussed here don't produce the correct 0/1/2 value?

no they solve the state counts correctly.

what i mean with a fixed point of space, there is already defined pattern logic in place or n many cover sets keeping it as unique or near unique depending on the starting fixed point.

reducing clues from that state allows more complex patterns to cover the cells.

if the pattern allows the puzzle to remain as a unique solution and no further clues can be reduced.

the grid will more likely have extremely complex logic validating the solution.

in an up build u have zero patterns in place for logic sets and must build the logic into a singular state.

building the logic upwards in a random fashion u have a reduced chance of having complex logic in place.

the total clue count would be a by produce of the logic sets in place ie the minimal needed given the sets used to create the unique state.

> **Quote:**
>
> My generators use no logic, rather backtracking preceded by singles solving, solely for efficiency reasons (to prune the search tree).

understood a recursive back track is far faster.

Ive been using logic sets to reduce the pm state, to only choose potions that are not covered yet.

the bias ism would be the limitation of implemented logic.

{my solver only has a limited amount ATM thus it cant fully optimize a gird.}

Last edited by StrmCkr on Thu Jul 09, 2009 7:02 am; edited 2 times in total

Back to top profile pm

denis_berthier

Posted: Thu Jul 09, 2009 6:59 am Post subject: quote edit

StrmCkr,

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

I understand this may be interesting when the purpose is to find hard puzzles. But when the purpose is statistical classification and rating, as in this thread, one can't accept such a deliberate bias.

Back to top profile pm www

StrmCkr

Posted: Thu Jul 09, 2009 7:04 am Post subject: quote

i think u missed my point,

Joined: 05 Sep 2006
Posts: 473
Location: Winterpeg

i am attempting to answering this question.

> **Quote:**
>
> The previous computations have shown that puzzles generated bottom-up tend to be easier than puzzles generated top-down.
> This remains very mysterious.

your listing statics on a group of random puzzles, top up or bottom up building.

and comparing rating and statical mean of clues on minimal puzzles between the different generators.

in a top down it has fixed logic in place already.
so it has an increased chance of generating more complex logic.

bottom up builds the logic sets randomly.

that is the difference in rating SE ratings.

the clue counts are a by products of the grid being a singular solutions,

which has no direct indication of the complexities of the logic inherent.

Back to top profile pm

denis_berthier

Posted: Thu Jul 09, 2009 7:17 am Post subject: quote edit

Joined: 19 Jun 2007
Posts: 673
Location: Paris, France

StrmCkr wrote:

> i think u missed my point.

That's true.
As any structure ("fixed logic") of a complete grid seems to be washed away by the deletion phase of the generator, I don't understand this explanation.
Can **anyone** explain StrmCkr's argument in different terms?

Back to top profile pm www

Display posts from previous: All Posts Oldest First Go

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

All times are GMT
Goto page Previous 1, 2, 3 ... , 22, **23**, 24 Next

Page 23 of 24

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