

数独 と J

Jsoftware の R. Hui の Essay を超えた話

中野嘉弘 (札幌市83才) 西川利男 (柏市) JAPLA会長 山下紀幸 (横浜市81才)

FAX 専 011-588-3354
yoshihiro@river.ocn.ne.jp
nakano@mta.biglobe.ne.jp

T・F 047-163-0364
Toshio.Nishikawa@kiu.ne.jp

T・F 045-851-3721

Sudoku の解法、Mr. Hui のエッセイへの高踏的解釈論は後にして、Hui を超えた実用の話をする。Hui は classic (size 9x9, 正方 box) のみしか扱わない。これでは「数独」パズルのレベルの実状に合わぬ。

なお、Hui プログラムの最新版を、本稿の末尾の付録に示しておく。

は し が き

ロートルもナウイ事をやる。最近、欧米にも人気のパズル SuDoku 「数独」を J で対応する話題である。昔、1979 年当時、USA で "Number Place (ナンバープレ)" と呼ばれる logic puzzle があった。1986 年頃、日本で人気となり「数独」と称した。今年 (2005 年) に入ってから "SuDoku" の名で、国際的に大変な人気を得たものである。香港在住のニュージーランド人 Wayne Gould の紹介のお蔭様だとか。

さて、我らの関心は、一つは、数独のパズルの問題を作成する事、これは初級群論の応用問題として可能となった。第二は、問題を解く事である。西川は表計算ソフト EXCEL を用いて解く方法を示した。我らが苦闘している中に近着の VECTOR 誌に、名案を發表された。先ず、カナダの Jsoftware の Roger Hui が J 言語で、それに並んで Adrian Smith が APL での、パソコン解法の例を紹介したものだ。

あれこれ、好機だと思うので、取りあげる。昨年末の J のシンポジウム向けには、惜しくも間に合わなかったので、新春のお年玉作品 (エッセイ) としよう。

1. 三老間の数独の話題

山下は「数独について知見あれば教えて下さい」と毎日新聞記事 "DO YOU SUDOKU?" を FAX して来た (文献 1)。続いて中野は読売新聞記事 (例解付き) を FAX した (文献 2)。山下は、その例題を解いたが、大きな疑問「原問題自身は、どうやって作るのかな？」が残った (文献 3)。山下流の解き方の考えを、西川は表計算ソフト EXCEL 上で実現して、パソコンで、数独問題を解いて見せた (文献 4)。この話題を西川は読売新聞に投稿したが、採択の気配は今の処無い。 (文献 5) EXCEL 関係の話は西川の別稿に譲る。 (文献 4 a)

ニコリ社出版の小冊子「数独」は既に 2 2 巻以上出ているが、書き込み式であるので、図書館には馴染まない。中野の住む札幌市の書店には、「ナンバーレ・・・」と云う名前の週刊誌的な類似誌が CD 付録付きで、各種、売られている。 (文献 6)

実は、後日に見た資料には、数独の解答にパソコンを用いれば、ミリ秒程度程度の短時間で解けて仕舞うので、ルールとしては「禁じ手」となっているようだ。

ついでに云えば、数独の原問題は copyright (著作権) の対象である。(文献7、8)

そこで、用語やルール等を略述して置く。基本形 (classic) の場合

0) 数字 1 ~ 9 を空所に入れるパズルである。

1) 9 x 9 ケの grid (格子) の cell (マス目) と、その中に サイズ 3 x 3 の subgrids (box とか region とか block と呼ぶ) がある。

既に数字が入っている処がある (givens とか clues = 手がかり、糸口 と呼ぶ)。

2) 各行 row、各列 column、各 box の空所 (empty cells、 missing numbers) に、重複しないように (only once, singly) 数字を入れよ (fill)。

と云う簡単なルールで、「数独 single numbers」 の名前の起源である。

この数字とは、単に「目印、マーク」 (例えばトランプの) であって、数学演算の対象では無い。 アルファベット や イロハニホヘト等でも同じことだ。

その点、魔方陣などより簡単である。(発展形については、後述する。)

The Daily SuDoku (文献8) などの如く、連日の問題作成は キツイ作業であろう。実は当初 ('05.10月7日) の西川 F A X でも既に「我らの Sudoku ゲーム問題は、解答を出すことが主眼か、あるいは問題を作ることが主眼か? もしも、前者ならば、ある種のパズルの解答問題で、既に、我ら JAPLA 会友・鈴木義一郎教授の報告例がある。」とコメントして来た。(文献9)

中野は当初から、Sudoku 問題の作成の方を狙っていたが、実は、作成と解答とは表裏一体のものである。下手な問題を作れば、テスト担当の山下に「解が無い」等と云われる。これは極端として、解が単一で無く、複数解を持つなど、その時は大変な難問になる。(文献10、11、12)

もっとも、業者提供の問題には難易度の階級があって、質的には EASY, MEDIUM, HARD そして VERY_HARD とか、或いは サイズ的に KIDS (お子様向け)、CLASSIC (古典的) あるいは MONSTER (怪物) など、発展の段階がある。

最近 ('05. Nov.28) 到来の VECTOR 誌内に、カナダの Jsoftware、Roger Hui の論文を発見した。彼自身はこれを、エッセイだと称している。これは J504 版を用いている。この中の primitive I. は、それ以前の J の旧版には無い。これと並んで、Adrian Smith の APL による同巧の論文もあり、これには西川も一驚した。これは J 言語のものより3倍位、長大である。しかも、なにやら、特異な手法を秘めているらしい(文献13、-a、-b)。

その Hui と、多少のやりとり(文献14、15)の後、Hui 流の J の Sudoku 解法を試用の結果、先の中野の原問題は、4種の複数解を持つ事が判った。その後、更に数字の givens の個数を増やして、単一解になるように改める事も出来た。

この様に、問題作成と解法は、車の両輪の如きものである。

なお、Hui のプログラムは、クラシックな 9x9 grids (格子) の Sudoku 問題のみである。これを、より一般的に色々拡張する事も可能で、それが先ず、今回の我らの報告の目的である。つまり、実用性である。次節以降に、詳細の各論を述べる。APLでの議論(文献13-a)には今は、少ししか触れられない。

その間に、西川は新型の PC (Sharp Mebius AE-50J, Win XP、中野の姉妹機) を購入し、J504版をダウンロードして、Hui のプログラムとの対応を探り、並びに、旧 J3版 (JFW、JPC) や旧 J4 版でもラン出来るような「西川版」を作成した(文献20)。

山下は DOS-V の JPC 版で、簡略型の「山下版」を試作した(文献21、-a, b)。

この原稿作製の終わり頃、Vector 誌に Hui の論文と並んでいた Adrian Smith の

APL 関連の報告 (文献 13-a) を読んで、気が付いた事にも、急ぎ触れておく。
 ゆっくり読んだら、大変有効な話であろう。特に、実践的な「数独問題作製」の立場
 では勇気付けられる事があった。

2. 中野の Sudoku 問題作成法

別稿「J と 群論入門」(文献 16、17) を書いている折にふと気が付いた事である。
 詳しくは、その別稿に譲るが、原理は至って簡単である。

例えば 4 次の置換群の乗積表を考える。群の要素 p1 (単元), p2, p3, p4 の間で

•		p1	p2	p3	p4	
p1		p1	p2	p3	p4	
p2		p2	p3	p4	p1	
p3		p3	p4	p1	p2	
p4		p4	p1	p2	p3	とする。

添字の数字 1,2,3,4 に着目すれば、各行、各列に数字の重複は無い。
 また、2x2 小行列 (数独では BOX) が 4 ケあるが、その中では重複が見える。
 しかし、もし左端が p2 と p3 の横行を交換すれば、その後の BOX 内に重複は無くなる。
 これは、4 次の「数独問題」の解に該当している。どこかの数字をマスク
 すれば、その数字配列は、まさに 4 次の「数独問題」である。

より高次にして、9 次の置換群の乗積表からは、CLASSIC (9 次) の数独問題が
 作り得る。次は 16 次、25 次 等々、一般に $n=1, 2, 3, 4, 5, \dots$ について
 n の 2 乗を次数とするものが作られる。

実際に、例えば The Daily SuDoku の 毎日の問題中に見ることが出来る。
 $n=2$ は for Kids (お子様)、 $n=3$ は CLASSIC、それ以上は Monster に当たる。

実例 n99 を挙げる。数字が隠されたマスク位置は 0 で表してある。
【作成法】 9 次の置換群の乗積表を作る。BOX 内に同じ数値が入らぬように、
 今は、例えば 第 3 行と第 6 行を交換する。
 マスクの箇所は、とりあえず、西川が EXCEL で解いた問題と同じ升目の場所を
 借用した。

n 9 9	0	2	0	4	0	6	0	8	0
	4	0	6	0	0	0	1	0	3
	0	9	0	2	0	1	0	6	0
	2	0	1	0	6	0	8	0	7
	0	0	0	8	0	7	0	0	0
	7	0	9	0	2	0	4	0	6
	0	1	0	6	0	5	0	7	0
	6	0	5	0	0	0	3	0	2
	0	7	0	3	0	2	0	4	0

山下が「目の子」で解をトライしたが、始めの 3 手 (0 オリジンで、8 行 6 列に 6、
 1 行 7 列に 2 を、2 行 8 列に 4 を置数まで) で、その先に進めなくなった。
 そして、この問題の数値の設定・配置に問題がないか? と反問して来た (文献 11)。
 この問題 n99 は、Mr. Hui のエッセイを知る以前の作品であるが、何れにしても
 very_hard な問題であった (結果的に)。

実は、この上の問題 n99 の解は、後に Hui 流の解法で調べれば、下掲の如く、
 4 通りも可能である事が判ったので、山下が苦勞したのも当然であり、更に極言すれば
 1 つの解では不十分なのであった。(同じ困難が、次節 3. や後節 10. APL

内の例題でも起きている。)

若干の追加指定 (givens を増す) した後は、山下は、キレイに解けたと FAX して来た (文献 18)。複数解では無くなったのである。

【4通りの解】

図 n99

【追加指定】 上記の n99 に、0オリジンで

- 1) (0行 6列)に 7 を指定すれば、解は 4通りから 2通りに減る。
- 2) 更に (7行 3列)に 9 を指定すれば、解は先の2通りから1通りに減る。

図 n997

このような複数解の起きないように、必要な数字を与える事は、SuDoku の出題者の苦勞する処である。出題者は、解いて見てから、出題している筈であるが、複数解は旨く避けられるのかな？ 中野は甘かったか？ 山下例の如き「ハイ！ 解けました」では不十分かも？ また、西川の Excel による解法は、これらを考慮しない easy な問題用

なのであろう。

Jsoftwareの達人、Roger Huiの今回の解法(文献13、15)では、この複数解を処理する事が出来た(複数解の個数分の出力も出来る)。常に可能とは云わぬが!

3. 4 x 4 Grid の場合

Huiの解法(文献13)は classicな 9 x 9 Gridの場合のみを想定している。それ以外の場合には、プログラムに多少の書換えが必要である。その変更は、行や列の個数が平方数(4, 9, 16, 25..)なら、わざわざ例示するまでもなからう。

例題は copyrightの関係上、前節 2.の方法で作ったもの X4を用いる。これしきの問題の解に、PCは不要であらうが、論理の順序として採り挙げる。

X46 02 | 40 この「数独」問題には、givens(手かかり)は 6ケある。
30 | 00 givensの個数を尾に付けてある。
----- この程度ならば、3才の幼児でも解けよう。
00 | 04
03 | 10 次に、givensの数字の個数を1ケずつ減らして調べよう。

X45	X44	x43	x42	X41
0040	0000	0000	0000	0000
3000	3000	3000	3000	0000
0004	0004	0004	0004	0004
0310	0310	0010	0000	0000

最後に givenを僅か1ケだけにして見る。このX41は解けますか!?
実は解けるのです。中野がHuiの拡張法で試みたところでは、なんと72通りがありました。その最初の数列を示す。

x41の解

0) 2431	1) 2431	2) 2431	3) 4231	etc
1342	1342	3142	1342	
3124	3214	1324	3124	
4213	4123	4213	2413	

これで驚いてはいけない。X41から最後のgivenを取り去ったもの、つまりX40(全てが0指定)でも解ける。解の個数は288通りとなる。しかし、これでも、4x4の「数独」問題を、日替わりで提供すれば、1年足らずで種切れになる。これらの珍事件に、Huiは無言であるが、我ら三老は気が付いた。

これら超能力は、Huiのプログラムで言えば、その中の guess(推測)機能の結果である。パズル「数独」の常識的ルールのみで解けるのは、初級 easyの場合だけで、中級 medium、難級 hard、大難級 very_hardの問題は、それでは解けない。上級レベルで「数独」の解法を求めるなら、この guess機能ははずす訳には行かぬ。そこまでは考えない簡略版(関数 assignのみでやる)の例を、J言語のJPC版のレベルで、山下が示したものを後節 8.に示す。その基になったJの旧版の西川のプログラミング例は、7.節に示した。

逆に、関数 assignをはずして、関数 guessのみで求解する事も出来る。

【関数 guessのみで求解】 中野簡易法の変更した文の番号は NB.(n1) etc. 番号の数値は、末尾の11. 節【付録】huiプログラムの文番号の数値に対応。

j2 =:(|/. i.@#),{;~2#i.2 NB.(n1)
r4 =:4#i.44 NB.(n2)

```

c4  =: 16$|: i.4 4      NB. (n3)
b4  =: (j2|4 # i.4 4){ j2      NB. (n4)
I4  =: ~."1 r4.,. c4 ,. b4      NB. (n5)
R4  =: j2, (, |: ) i. 4 4      NB. (n6)
free4 =: 0&=>(1+i.4)"_ e. "1 I4&{      NB. (n8)

guessa4n =: 3 : 0      NB. (n19)
  if. -. 0 e. y. do. ,:y. return. end.      NB. (n20)
  b =. free4 y.      NB. (n21)
  i =. (i. <./)(+/"1 b){ 5,}. i. 5      NB. (n22)
  , (y. +"1 (1+I. i{b)*/i = i.16)      NB. (n23) 先頭の “ , “ に注意。
)      NB. (n24)

sdk4n =: 3 : 'guessa4n ^:(+/(0=y.)) y.'      NB. (26)

```

ここまで僅々、13行 である。

```

実行例      問題      x4 =: , 0 ". ] ; _2 (0 : 0)
              0 2 4 0
              3 0 0 0
              0 0 0 4
              0 3 1 0
              )

```

```

解          4 4 $ sdk4n x4
          1 2 4 3
          3 4 2 1
          2 1 3 4
          4 3 1 2

```

【注意】 本法は 9 x 9 grid の場合でも有効であるが、こんな事で、多くが解けるとは
 思うまい。 元来の hui の解法でも、解けない場合が、山ほどあるのだから。
 要は有効度で判断すべきである。 それでも簡便法の存在は、原の hui プログラム
 の流れを理解する事には役立っているだろう。
 これは、後節 8. の「山下の簡便例」でも云えることだ。

4. 文字データの Sudoku 問題

「数独」問題は昔、ナンプレ Number Place と呼ばれた。 しかし、この「数」は
 演算の対象ではなく、単なる「目印」であるから、一般に「文字データ」で置換して
 構わない。 具体的には Mr. Hui の Sudoku プログラム内の関数 see1 内の
 文字列、例えば 4 x 4 配列の時なら、'.1234' を '.ABCD' 等と置換すれば良い。
 トランプの4種のマークでも良い。
 ただし、当然ながら、利用 (place) べき文字の全体範囲は示して置かねばならぬ。
 なお、サイズが 9 x 9 以上の「数独」問題では、ほとんど必然的に、文字 A, B, C
 を利用する事になろう。

例 4 x 4 配列

図 4 x 4

5. 矩形 BOX の 問題

6 x 6 次の grids 内に、矩形の 2 x 3 次の box を 6 ケ設けた場合を考えよう。

「数独」の問題作りに「置換群」を持ち出したのは、実はペダグチック過ぎるかも知れぬ（こけ脅しかも？）。 本当は簡単な事だ。

数列 p1 として 1 2 3 4 5 6 を先ず採る。これを 3 つ シフトして
数列 p2 4 5 6 1 2 3 を作る。 数列 p1 を 1 つシフトして

数列 p3 2 3 4 5 6 1 を作る。これを 3 つ シフトして
数列 p4 5 6 1 2 3 4 を作る。 同様にして

数列 p5 3 4 5 6 1 2
数列 p6 6 1 2 3 4 5 を作る。

かくして作られた 6 x 6 数列は、各行、各列、各 box 内に重複数字が現れ無いので、「数独」の解の条件を満たしている。これに、適当なマスクを掛けて、空所を作れば、矩形 BOX の「数独」問題が出来た事になる。

例えば x6 として

```
0 0 0 4 5 0
0 0 6 0 0 3

2 0 0 5 0 0
0 0 1 0 0 4

3 0 0 6 0 0
0 1 2 0 0 0    の如し。
```

この手の問題は、数列 p1 の順列 6 ! 即ち 7 2 0 通りは、マスク掛け (open の場所指定) の問題を別にして可能である。

この矩形「数独」問題の解法や如何に？ Jsoftware の Hui 氏は解法を用意していない！ 我ら三老は、我らの答を開陳すべきであろうか、それとも読者の宿題に？ 一応、ヒントらしき解説を提供しよう。

工夫は 正方形でない BOX の処理だけである。

hui プログラム (1 1. 付録) の 3 行目 b = (h4) の処理が簡単では無くなる。しかし、高々 36 行 6 列分であるから、

b66 = , 0 ".] ; _ (0 : 0) の下段に、下記の bx623 を参考にして
手動で入力するのも簡単でよい。

bx623

```

0 1 2 6 7 8
3 4 5 9 10 11
12 13 14 18 19 20
15 16 17 21 22 23
24 25 26 30 31 32
27 28 29 33 34 35

```

今後の為に、より一般化したいならば、例えば

```

b0 =. (3 6 $ 0{ bx623)
b1 =. (3 6 $ 1{ bx623)
b2 =. (3 6 $ 2{ bx623)
b3 =. (3 6 $ 3{ bx623)
b4 =. (3 6 $ 4{ bx623)
b5 =. (3 6 $ 5{ bx623)

```

として

```

(12 6 $ ,(b0, b1)), (12 6 $ ,(b2, b3))(12 6 $ ,(b4, b5)) から
0 1 2 6 7 8
0 1 2 6 7 8
0 1 2 6 7 8
3 4 5 9 10 11
3 4 5 9 10 11
3 4 5 9 10 11
.....
.....
27 28 29 33 34 35      らて 3 6 行 6 列 が得られる。

```

この苦勞を厭うて、「2次方程式の根の公式」の類を期待する事は出来るかな？
hui プログラム自身、Vector 誌の 旧版から、本稿付録の新版に更新した際に、定義
文番号 (h1) から (h4)迄の定義記号を =: から =. に変更して、問題毎
に、ユーザー指定に直したのは、意図的である。
つまり、「数独」の問題の型式毎に、対応するプログラムも異なるのが必然とされる。
繰り返すが「根の公式」的なものは無い！

文番号 (h2) から (h3) までの 変数 r (行指定) , c (列指定) はサイズ変更
のみで良い。 文番号 (h1) の 変数 j は BOX のサイズ関係としては、正方形で
無い今の場合、上記の別途指定により、それとの関係は無くなった。
文番号 (h6) の 変数 R 及び (h7) の regions には関係する。

```

中野では j23 =: 6 6 $ ,(|/. i.@#), {;~2#i.3
R6 =: j23, (|:) i. 6 6
regions6 =: R6"_ {"_ 1 ]

```

その他は、それらに従って当然の変更がなされる。これで、計算は出来る。
しかし、表示の問題が残る。 難問は BOX の範囲を示す境界線に相当する
何かである。 正方形の型式の問題では、関数 see が、その役目を担当した。
矩形の場合には、これを、どうするか？ さらに工夫が必要かもしれぬ？

中野の解 (資料の都合で別な、Daily SuDoku for Kids: 29-Nov 2005 の問題で)

図 see23

6. 高次正方Box の 問題

Box が正方形となるのは、次数が平方数の場合である。
つまり 次数 N が n^2 , $n=2, 3, 4, 5, \dots$ の場合である。
次数 9 は Mr. Hui の original であり、次数 4 は本稿の前節の例で示した。
高次とは先ず、Mr. Hui が扱わなかった 16 次 の場合である。 さらに、25 次、
36 ……。 一つの汎用プログラムで処理するのは、難儀な事であろう。
ところで、16 次までなら、「数独」関係の業界が、すでに問題（モンスター級と
して）公表しているのだから、解くだけであり、これは Mr. Hui の処方借りれば、
殆ど百発百中の（勿論、例外はあるが！）可能である。
しかし、25 次の場合となると、幸か不幸か？、問題自身が見当たらない。
そこで、中野は、その問題作りから始めた。
そして、正解が得られるように（Hui プログラムでも解けない場合が大半）、手直しを
努力した。 問題作成と解答とは、正に、車の両輪の如き関係にある。
問題作成としては下手かも知れないが、ひょっとして、世界で初めての「25 次」の
「数独」問題になったかも？
作り方の原理は「郡論の利用」で、前節 2. や 3. で、屢々、説明済みであるので、
ここには出来上がった問題のみを示す。 読者諸賢の求解を期待する。 解答は末尾、
付録の Mr. Hui のプログラムの後【付録 2】に示しておく。

図 n25

7. 不定形の BOX では

インターネットの Sudoku サイト The Daily SUDOKU は高級である。類書が 9 x 9 正方形 BOX の問題しか扱わないのに対し、バラエテに富んでいる。年末のクリスマス頃の 10 日間には、特に、プレゼント的な変形出題が追加された。

例えば

Star Sudoku: Thu 22-Dec-2005 medium
Scary Angel Sudoku: Wed 21-Dec-2005 hard
Church window Sudoku: Tue 20-Dec-2005 hard
Angel: Fri 30-Dec-2005 hard
To replace the duplicate: Thu 29-Dec-2005 very hard
Christmas tree Sudoku: Fri 23-Dec-2005 very hard
Take down the tree: Sat 31-Dec-2005 very hard

まるで、教会のステンドグラスもどきの多様性である。

その中、Christmas tree 型の 1 例 にトライした件を報告する。

これは givens 21 ケ、 free 60 ケ の極々の難問である。

問題 Christmas tree Sudoku: Fri23-DEc-2005

(c) Daily Sudoku Ltd 2005. All rights reserved.

のみを下掲させて頂く。 サイズは 81、 名前は xc90 として置く。

☒ xmas

クリスマス・ツリー 型の不定形 BOX (grid) であるから、Mr. Hui 流とは、真っ向から別な処理が必要になる。(最近、西川も不定形 BOX の存在を知り、Hui エッセイの件も西川個人としては、根本から考え直す必要があるか?と述べている。)

```
旧・正方形      j=(|/. i.@#), {;~3#i.3
                  b=: (j{9#i.9)j
新・不定形      b=:81 9 $ b0
                  b0=: , 0 ". ] ; _2 (0 : 0)
                  0 1 2 3 9 10 11 12 18
                  0 1 2 3 9 10 11 12 18
                  0 1 2 3 9 10 11 12 18
                  0 1 2 3 9 10 11 12 18
                  4 13 21 22 23 30 31 32 40
                  5 6 7 8 14 15 16 17 26
                  5 6 7 8 14 15 16 17 26
                  5 6 7 8 14 15 16 17 26
                  5 6 7 8 14 15 16 17 26
                  0 1 2 3 9 10 11 12 18
                  .....
                  .....
                  .....
                  54 63 64 65 66 72 73 74 75
                  54 63 64 65 66 72 73 74 75
                  54 63 64 65 66 72 73 74 75
                  54 63 64 65 66 72 73 74 75
                  55 56 57 58 59 60 61 67 76
                  62 68 69 70 71 77 78 79 80
                  62 68 69 70 71 77 78 79 80
                  62 68 69 70 71 77 78 79 80
                  62 68 69 70 71 77 78 79 80
```

) べて 81行

Hui 流の演算 sudoku xc90 は残念ながら、|index error free|全く動かぬ。そこで中野は、特に僅少の工夫をし、時間計測も含めた関数 sudoknt を作った。

```
結果は      sudoknt xc90
            0 3 0 0 2 0 0 9 0   左の9行は問題の原データ
            0 0 1 0 0 0 2 0 0
            0 0 0 7 0 3 0 0 0
            0 7 0 4 0 9 0 2 0
            0 6 2 0 0 0 8 3 0
            0 0 0 1 0 5 0 0 0
            0 0 0 0 0 0 0 0 0
            0 0 0 8 0 4 0 0 0
            3 0 0 0 0 0 0 0 5
```

空行の下9行が結果

```

638521697   しかし、万全では無い
741980253           0が残る
215753968
873469521
562917834
926135378
183372459     33が重複
657894012     0が残る
394206185     0が残る

```

計算所要時間は 0.02 sec と僅少であった。

この関数 `sudoknt` は、`Hui` 関数の内、`assign` のみを用いる簡単なものである。もっともらしく動いているが、さすがに、`very hard` の問題、簡単には行かぬ。比較して見る。 `sudoknt` から、時間と原データ出力を省略したものを、`sudoknt0` として、さらに 関数 `assign` や 末尾の 1 1. 節【付録 1】内 `Hui` の一手関数 `Phrase (P1) f:=+(ac>.ar)@free` と比較して見る。

```
(99$ f^:6 xc90)=99$ assign ^:4 xc90 = sudoknt0 xc90
```

```

6回反復           4回反復           1回

```

結果は、互いに完全に同じである (99\$ 1)。

しかし、問題の提供者のが正解では

```

834621597 791568243 248713956
573489621 162957834 926135478
485372169 657894312 319246785

```

と、大きく違う。

その理由は、: 関数 `assign` のみを用い、`guess` 無しで、処理した為であろう。同好のトライを、後節 9. 山下簡略法も行っているが、やはり無理な注文らしい。では、`guess` 関数の威力は如何か？

```
99$ ,guess ^:50 xc90 と 50回反復すれば、結果は
```

```
830521497 741658203 208703916 .....
```

と、正解に接近する。問題の原データの `free` 箇所は 60 であるから、60 回まで反復したいが、それは `|length error|` を発して不可能であった。

実は、それどころで無く、次の反復 51 回以上の結果は

```
738621594 .....
```

`Hui` 流の 推測関数 `guess` が駄目ならと、全く別法な「中野法」で処理したが、その話は次回に譲る。この節の話題だけでも、資料膨大、書き切れぬ。

「コロンブスの卵」の譬もあるが、諸賢の工夫を待ちたい。問題、甘くはない！

例えば、問題のデータで、80ヶ所を埋め、空所を1ヶ所のみとして、その後を

「`Hui` の関数 `sudoku` で完成せよ」としても、その最後の一手が出来ないのだ。

目視で直解の筈なのに！ 実戦は、先人のプログラムの理解だけでは出来ぬ。

8. 西川の 旧版プログラム

新春に、新しいノート型パソコンを新調した西川は、Jの最新版 J504 をダウンロードし、`Mr. Hui Sudoku` プログラムをトライした。ついでに、Jの旧版しか用意せぬ老友の為に、御親切にも、`Hui` のそれ(稿末、1 1. 付録)を、若干修正した改訂プログラムを FAX して来た(文献 20)。大きな改訂は J の `primitive I` を用い無い方法である。変更箇所は 2 箇所、文番号 NB.(10a と 13)で示した。その他は、稿末の【付録】 1 1. と全く同じであるが、読者の利便のため、折角である

から、そのまま掲げる。

NB. Hui's Sudoku Solving Program

NB. Modified for J4, J3 (JFW, JPC) by T. Nishikawa, 2006/1/2

j =: (|/. i.@#),{~3# i.3

r =: 9 # i.9 9

c =: 81 \$ |: i.9 9

b =: (, j{9 # i.9) { j

I =: ~."1 r ,. c ,. b

R =: j, (, |:) i.9 9

regions =: R" _ {" _ 1]

free =: 0&=> (1+i.9)" _ e."1 I&{

ok =: (27 9\$1)" _ -: "2 (0&= +, ~:"1) @ regions

ac =: +/. * &(1+i.9) * : = +/"1

Ip =: # i.@# **NB. I. (indices) is defined as Ip** **NB. (ns 10a)**

ar =: 3 : 0

m =. 1=+/"2 R{y.

j =. Ip +. /"1 m

NB. (ns 13)

k =. 1 i."1~ j{m

i =. ,(k|" _ 1 |: "2 (j|R){ .) #"1 j{R

(1+k) i;81\$0

)

assign =: (+ (ac >. ar)@free): _ "1

guessa =: 3 : 0

if. -. 0 e. y. **do.** ,y. **return.** **end.**

b =. free y.

i =. (i. <./)(+/"1 b){10 }.i.10

y. +/"1 (1+ Ip i{b)*/i.81

)

guess =: ; @: (<@guessa"1)

sudoku =: guess @: (ok # |) @: assign ^: _ @ ,

see0 =: ({&'123456789')@(9 9&\$) @ , **NB. modified by TN**

see1 =: (3 3 ;: 3 3)&(<; 3) @ see0 **NB. modified by TN**

NB. see1 =: (;~9\$1 0 0)&(<;.1)@({'123456789')@(9 9&\$)@ ,

see =: <@see1"1`see1@.(:=#@ \$)

diff =: * 0&=@: @:(0&.)

f =: + (ac >. ar)@free

この関数 **f** は、解を一手づつ見る場合用に **hui** が用意してくれたものである。

データ例

x0 =: | ; _ 2 (0 : 0)

2 0 0 6 7 0 0 0 0

0 0 6 0 0 0 2 0 1

4 0 0 0 0 0 8 0 0

ym =: | ; _ 2 (0 : 0)

0 5 0 7 0 1 0 4 0

7 0 3 0 0 0 1 0 3

0 8 0 4 0 6 0 9 0

```

500 009 300          904 060 803
030 000 050          000 807 000
002 800 007          108 050 609
001 000 004          010 603 080
708 000 600          506 000 701
000 053 008          030 509 020
)
x =: ,0". x0          ym9 =: ,0". ym

```

see ym9, sudoku ym9

他に ニコリ社刊 sudoku Vol.21 (文献 6) の Sudoku21, p.10 No.1 Easy、Sudoku21, p.110 No.99 Hard などをトライをした。

9. 山下流 簡略例

新春の西川 F A X (文献 20) に励まされて、山下は、簡単化のトライを行った (文献 21)。 hui (や 西川旧版) プログラムを、文番号 (h1) から (h18) の assign 関数までで打ち切り、そのあと一挙に関数 sdky で纏めようと云うもの。簡単に云えば、推測用の関数 guess を使わぬ簡単化である。

NB. Modified for DOS-V J (JPC) by K. Yamashita 2006/1/8

NB. assign =: (+ (ac >. ar) @free) ^: _"1 NB. (h18)

sdky =: 3 : '<9 \$ y.),<9 9 \$ assign y.' NB. (y 18a)

文は 18 行である。

実行例 (前節、西川と同じデータ ym)

```
ym =: | ; _2 (0 : 0)
```

NB. data array

```
)
```

```
ym9 =: , 0". ym
```

(普通は ym9 =: , 0". | ; _2 (0 : 0) と一挙にデータ定義する。)

そして sdky ym9 で結果は、西川と同じ。 (以上)

この簡単化は Mr. Hui の論理の入り口を眺めるには判り易い。しかし、これで解けるのは、9 x 9 型の「数独」で、かつ ラッキー な問題だけであるまいか? Mr. Hui の狙いはもっと、深いところにあると思われる。それは、guess (推測) 機能である。assign 関数を用いずに、guess 関数だけでも、解の得られる事を中野は示してある (先の 第2節 参照)。要は問題がラッキーであれば出来る。目視でも良い。元来、パズル作者は、P C の使用を禁じている。解の成功度を高める為には、関数 assign と guess の両方があった方がベターである。そしてその時は、もはや簡略版ではない。

次節 10. で触れるが、「数独と A P L との関連」で Adrian Smith の例題が A P L では簡単でも J (の Hui 流の解法) にとって、大変な難問らしい事を述べる。

10. A P L と の 関 連

Vector 誌の「数独」記事には、Mr. Hui の J 言語関係の他に、A P L 関係が掲載されている。話題を絞るため、今まで触れないで来たが、先日、仲間の西川会長から、要望があったので、関係した若干のやりとりを紹介する。(文献 22)

表題は Sudoku with Dyalog A P L、内容は Adrian Smith 氏が、2つの研究の紹介をしたもの。

1) John Clark 氏の backtracking solver (解法) と puzzle-generator (作成法)

2) Ellis Morgan 氏 の recursive puzzle solver (解法)

例題がある。形式は classic (9 x 9、正方 BOX)、clark と命名しよう。

```
0 3 4 0 0 5 0 0 0
0 7 5 0 2 0 1 0 0
0 0 0 0 0 0 8 7 5
```

```
3 4 0 0 0 0 0 9
0 0 0 2 6 0 3 4 0
0 6 0 3 4 0 0 0 0
```

```
0 0 3 5 8 0 0 9 0
0 0 0 0 0 0 0 0 3
0 0 0 0 0 0 0 0 0
```

難易度は medium である。given は 25ヶ所、free (open) は 56ヶ所。山下の統計によれば、日本のニコリ社の SUDOKU (Vol 21) 誌 (文献6) では given が ~ 32 easy、~ 27 medium、~ 21 hard であるから、上の given 25 は、難易度 medium と hard の中間位である。very hard では無いから、勝手知ったる「J 言語で解け」は妥当なところである。

ところが、山下 FAX は「あえなく討死！ 追加 fill 出来たのは 56ヶ所中の 5ヶ所のみ。」つまり、「解けぬ」と伝えて来た。使用した山下の簡易版プログラム (前節8.) は、先の SUDOKU 誌の 99 題中 74題 (easy + medium) を解いた実績があると云うものである。何故駄目か？

London Times のレベルが高いのか？ どうも、free (open) の箇所の分布に関係するかも？ 日本の例では、free の箇所が、中心点に関して、殆ど全て、点対称に見えるが、この例 clark では、そうでは無い！

John Clark や Ellis Morgan などの別な手法が必要な理由であろうか？ と、山下は結んであった。

これを、我が読者諸賢にバトンタッチする前に、中野の計算結果を紹介する。classic 型であるから、手法は元来の Hui 流である (勿論、老友の簡易法ではない)。ただし、解を見て「この APL での問題の作成には、中野の作成問題 n99 や n2525 と同様、群論を利用したな！」と強く感じた。それなりの特徴が私には見えるのだ。

中野の計算結果 (問題中で free 箇所を、もう 1ヶ所増やすと、最早解けぬ。複数解が一挙の増大するのだ。群論利用の問題はきわどい事が多い！)

解けぬ例 APL9次
open 56, fill 25
途中打ちきり

解けた例 APL9次
open 41, fill 40
完了 0.01 sec

図 apl

11. 【付録1】 Hui の 原プログラム等 (新版)

中野が Mr. Hui とやりとりの際、Vector誌の記事の改訂版として、知らされたものである (最終更新日 2005-12-17、更新者 Roger Hui、文献15)。
 我らが本稿を書き始めたのは、その前の '05-12-1 であった。
 新版でも、原の Vector誌の Hui の text ('05 autumn) と本質的には、同じことだと Hui 自身は云っている。

文番号 (NB. 以下) は、参考の為に、中野が付けたもの。

```

j =. (|/. i.@#) ,{:~3#i.3      NB. (h1)
r =. 9#i.9 9                  NB. (h2)
c =. 81$|:i.9 9              NB. (h3)
b =. (j{9#i.9) {j            NB. (h4)
                               NB. ここまでは =. を使用
I =. ~."1 r,c,b              NB. (h5)
R =. j,(|:)i.9 9             NB. (h6)

regions =. R"_{"_1]         NB. (h7)
free =. 0&=>(1+i.9)"_e."1 I&{  NB. (h8)
ok =. (27 9$1)"_-"2 (0&=+.~:"1)@ regions  NB. (h9)

ac =. +/. *&(1+i.9) * 1: = +/"1  NB. (h10)

ar =. 3 : 0                  NB. (h11)
m=. 1=+/"2 R{y.             NB. (h12)
j=. I. +/"1 m                NB. (h13)
k=. 1 i."1~j{m              NB. (h14)
i=. ,(k{"_1 |:"2 (j{R){y.) # "1 j{R  NB. (h15)
(1+k) i}81$0                NB. (h16)
)                             NB. (h17)

```



```

assign =: (+ (ac >. ar)@free)^:_"1 NB. (h18)

guessa =: 3 : 0 NB. (h19)
  if. -. 0 e. y. do. ,:y. return. end. NB. (h20)
  b=. free y. NB. (h21)
  i=. (i.<./) (+/"1 b){10,}.i.10 NB. (h22)
  y. +"1 (1+I.i{b})*i=i.81 NB. (h23)
) NB. (h24)

guess =: ; @: (<@guessa"1) NB. (h25)

sudoku =: guess @: (ok # |) @: assign ^:_" @, NB. (h26)

see1 =: (;~9$1 0 0)&(<:,1) @ ({{&'123456789'}) @ (9 9&$) @, NB. (h27)
see =: <@see1"1`see1@.(1:=#@)$ NB. (h28)
diff =: * 0&=@}:@:(0&,) NB. (h29)

```

その他の追加資料として、御親切にも one step ずつ推進用などの phrases が 9 ケほど用意されている。「数独」の実践家には有効であろう。(以上)

有効 phrases

```

(P1) f=: + (ac >. ar)@free NB. one step of assign
(P2) see t=: f^:a: x NB. forced moves leading from grid x
(P3) see diff t NB. differences from one grid to the next
(P4) see assign x NB. same as the last grid above
(P5) see g=: guess (ok#) assign x NB. guesses after exhausting forced move
(P6) see t0=: f^:a: 0{g NB. forced moves leading from guess 0
(P7) see diff t0 NB. differences from one grid to the next
(P8) see t1=: f^:a: 1{g NB. forced moves leading from guess 1
(P9) see diff t1 NB. differences from one grid to the next;
NB. note the obviously invalid assignments

```

1 2. 【付録 2】 2 5 次の数独問題 (中野作品) の解

☒ gij251

13. むすび

Sudoku (数独) の解法に、PC利用は元来は禁手であるが、Mr. Hui の自称 Essay のお蔭で、面白い問題に発展しつつある。Hui はサイズが 9×9 で BOX も正方形の classic な場合しか考えていないが、我ら日本の三老は、それを一般化した。

APLでの試みもあり、今後の発展が期待される面白い分野である。Hui 流が時に「立ち往生、解けぬ」となるのは、複数解を許す問題の場合である。この複数解対策が肝心なところだ。PCでなく、目視・腕力で作成した古典的な問題では、この困難は途中で排除される確率が高いだろう。心すべきだ。しかし、中野のサイズ 25 次の問題作成は、あるいは世界で最大かも？ Hi!

文 献

- 1) 山下 FAX ('05.10.7. 15:10) 毎日新聞 平成17年10月 7日 (金)
- 2) 中野 FAX (10.15) 読売新聞 平成17年10月15日 「数独」解答例付き
- 3) 山下 FAX (10.16) 読売新聞の例題を解いた。問題は どうやって作る？
- 4) 西川 FAX (10.17.pm 6.19) 読売問題を解いた。山下の考えを EXCEL で実現。
 - a) 西川利男：「J のオブジェクト指向プログラミング (OOP) その2 J のスプレッドシート (Grid) と数独パズルへの適用」JAPLA シンポジウム 2005.12.10
- 5) 西川 FAX (10.25) 上記の EXCEL法 を読売新聞に投稿した。
- 6) 山下 FAX ('05.10.19) 関係書「ペンシルブック数独 21」と「激辛数独 1」
 - 山下製 (SUDOKU) 第1号と、その解をお知らせ。
 - a) ニコリ「ペンシルパズル本 数独 22」H17.12.19、山下氏より恵贈。
 - b) ビデオ出版「ナンプレ広場」隔月刊、サムソン1月号増刊、'06.1.10 発行
- 7) 例えば：<http://www.puzzle.jp/letsplay/sudokurule-e.html>

- a) <http://en.wikipedia.org/wiki/Sudoku> 解説 pp.10
- b) <http://www.dailysudoku.co.uk/sudoku/faq.shtml> FAQの分 pp.5
- 8) <http://www.dailysudoku.co.uk/sudoku/> 解説コーナー FAQ 等もあり
- 9) 西川 FAX ('05.10.7) 補足：ハノイ塔論文 APL 原稿の画像、Sudoku ゲーム
- 1 0) 中野 FAX ('05.11.23) Sudoku の問題を作った、初級群論の例題として。
本稿では、旧稿の数値に 1 を加えものを採用。
- 1 1) 山下 FAX (11.23.12:33) 中野 Sudoku 問題は解けない。途中で stop !
- 1 2) 中野 FAX (11.23.16:45) ソリー！ 小生の出題に記入漏れ。0 オリジンで
(4行、1列)の箇所を 6 を省いてあったので、問題は困難化された。
- 1 3) Roger Hui: "A Suduko Solver in J" VECTOR Vol.21 No.4 Autumn 2005
pp. 49-52
【註】表題の Suduko は、そのまま。出版所のミスである(文献 1 4、1 5)。
- a) Adrian Amith: "Sudoku with Dyalog APL from John Clark & Ellis Morgan"
ibid. pp.53-61
- b) 西川 FAX ('05.12.2.9:41) Vector誌拝見、驚き。私のテーマと衝突？
- 1 4) Hui 宛の中野メール：貴方の日本語にのみコメント、Sudoku が正しい、その他。1 5
Hui からの返メール：誤りは既に指摘されたが、時すでに遅し。訂正版あり。
- a) <http://www.jsoftware.com/jwiki/Essays/Sudoku/>
この Web サイトは JAPLA 会友の志村氏が、次期の研究報告メディアとして、考
えて居るものらしいが？
- 1 6) 中野嘉弘：「群論演習と J (その 0) 副題：志村論文を応援して」 '05. 12. 10
J A P L A シンポジウム
- 1 7) 中野嘉弘：「群論演習と J (その 1) 副題：ハノイ群を目指して」 予定稿
- 1 8) 山下 FAX ('05.12.2.17:36) 9次 SuDoku n99 (中野 '05.11.23 出題)
キレイに解けた。本稿 2 節 (p.3) 中野の問題
- 1 9) 中野 FAX (2005.12.24: Sat) 「数独クリスマスツリー」を送ります。
- 1 0) 西川 FAX ('06.1.6, pp.5) 新年挨拶、暮れに新パソコン購入 (シャープメビウス
AE-50J, Windows XP)。J 5 0 4 ダウンロード、H u i 数独プログラム
をラン、修正版として、J 3 (J F W), J 4 対応する「西川版」を作成。
- 2 1) 山下 FAX ('06.1.8, 10.56) J P C 版で「数独・簡略版 山下流」出来た！？
山下 FAX ('06.1.8, 13.41) 「山下流 数独・簡略版」全 1 8 行です。
- a) 山下 FAX ('06.1.8, 15.41) 「激辛数独 Vol.1～ Vol.2」級の難問では
簡略版では難儀かも？ やはり、g u e s s 機能は省けぬか？？
- b) 西川 FAX ('06.1.15,pm 3:20) 旧 DOS 機内の J P C を探し出して、エラー
無しで動く事を見た。
- 2 2) 西川 FAX ('06.1.17,am 10:55) 「群論」の勉強、A P L、L a b s
- a) 中野 FAX ('06.1.17,pm:5) 文献 1 3 - a の A P L 論文 pp.8 送信
山下先生へ：John Clark の APL 関係の Sudoku 例題 9x9 難易度 medium,
from London Times を解け
- b) 山下 FAX ('06.1.18,13:41) APL の例題にはあえなく討ち死！ 解けず！
(# 20:00) London Times の問題は、日本のニコリ社より難しい！
1 つの解に対し、マスクは 1 つしか無いのか？？
※ マスク掛け (free または open 位置) の問題は、別途、研究対象になり得る
であろう。