An Abstract Procedure to Compute Weak Schur Number Lower Bounds

Bruno Bouzy
bruno.bouzy@parisdescartes.fr
Université Paris Descartes

Séminaire Jeux
Université Paris Dauphine
3 février 2015

Outline

- Weak Schur Number Definition
- Related work
- Observations on existing solutions
- An abstract recursive procedure
- An abstract simulation for Monte-Carlo
- Experiments and new results
- Discussion
- Conclusion

Definitions (1/2)

- (K, R) Schur problem :
  - Finding K partitions of the interval [1, N] such that :
  - In any partition, no number is the sum of R other numbers in that partition.
- « Weak » Schur problem :
  - The R numbers are distincts
- True Schur problem
  - otherwise

Definitions (2/2)

- R=2
- WS(K) :
  - Greatest value of N such that the weak schur problem (K) is solved.

Example

- WS(1) = 2

1	2			

- WS(2) = 8

1	2	7		
3	4	5	6	

:-(

1	2	4	8	
3	5	6	7	

:-)

Related work (1/2)

- Lower bounds and upped bounds
  - Bornzstein 2002,
  - Abbott & Hanson 1972 :

$(44/89)89^{K/4}$ <= S(K) <= WS(K) <= [k!ke]

- True Schur :
  - Schur 1916 : S(K+1) >= 3 S(K) + 1
  - Baumert 1961 : S(4) = 44
  - Exoo 1994 : S(5) >= 160
  - Fredericksen 2000 : S(6) >= 536, S(7) >= 1680

Related work (2/2)

- Blanchard & al 2006 : WS(4) = 66
  - 29,931 length-66 solutions
- Walker 1952 : WS(5) = 196 (claim, not proved)
- Robilliard & al 2011 :
  - WS(5)>=196 and WS(6) >= 574
- Bras & al 2012 : WS(6) >= 581
- Eliahou & al 2013 : WS(6) >= 582

Known values

Known values

- True Schur

K	1	2	3	4	5	6	7
S(K)	=1	=4	=13	=44	>=160	>=536	>=1680

- Weak Schur

K	1	2	3	4	5	6
WS(K)	=2	=8	=23	=66	>=196	>=582

The 3 solutions for K=3

- A

1	2	4	8	11	22				
3	5	6	7	19	21	23			
9	10	12	13	14	15	**16**	**17**	18	20

- B

1	2	4	8	11	**17**	22			
3	5	6	7	19	21	23			
9	10	12	13	14	15	**16**	18	20	

- C

1	2	4	8	11	**16**	22			
3	5	6	7	19	21	23			
9	10	12	13	14	15	**17**	18	20	

Observations (1/4)

- DFS : time to find the solutions
  - K=3 : instantaneous
  - K=4 : long (hours...)

- ~~Optimal solutions for K+1 columns contain optimal solutions for K columns~~
  - True for K=1, 2, 3
  - False for K=4, 5

Observations (2/4)

- « Good » solutions for K+1 columns contain « good » solutions for K columns
  - Good = near optimal
  - Good = long (the longer the better)
  - True for any K>=1

Observations (3/4)

- Almost consecutive numbers in the last column are common
- Consecutive numbers is a simple method

9	10	11	12	13	14	15	16	17	18

19, 20, …, 35 are forbidden

start a column with N and fill it until 2N

- « Holes » in the middle do not help

9	10	11	12	13		15	16	17	18

19, 20, …, 35 are still forbidden

Observations (4/4)

- Holes at the beginning of a column may help

9		11	12	13	14	15	16	17	18	19	
9	10		12	13	14	15	16	17	18		20
9			12	13	14	15	16	17	18	19	20

- Holes :
  - h-0 : absence of hole
  - h-N : a hole of size 1 at First(K)+N
  - H-M-N : combination of h-N and h-M

Abstraction

- Group = « almost consecutive » numbers in a column

9			12	13	14	15	16	17	18	19	20

- Action = putting a group in a column
- Contrary :
  - Putting numbers one by one in a column
- Monte-Carlo background:
  - a simulation can be abstract (our work) or not (related work).

The core recursive procedure

```
int Damier::coreRecursiveWeakSchur(int K)
{
  if (K<=0) return 0;
  coreRecursiveWeakSchur(K-1);
  fillColConsecutiveNumbers(K);
  coreRecursiveWeakSchur(K-1);
  return smallestOut – 1 ;
}
```

1	2	7	
3	4	5	6

1	2	7		17		21		
3	4	5	6	18	19	20		
8	9	10	11	12	13	14	15	16

K	1	2	3	4	5	6	7	8
wslb	2	7	21	61	180	536	1593	4733

The abstract procedure

```
int Damier::abstractRecursiveWeakSchur(int K)
{
    if (K<=3) { DFS(K) ; return smallestOut-1;  }

    abstractRecursiveWeakSchur(K-1);

    Imax = 0 ; bestB = nil ;
    For Hole {
        B2 = this;                          B2.fillColWithHole(K, Hole) ;
        B2.fillColConsecutiveNum(K); B2.fillEndCol(K) ;

        B2.abstractRecursiveWeakSchur(K-1);

        If (Imax<B2.smallestOut-1) { Imax = B2.smallestOut-1 ; BestB = B2 ; }
    }
    this = bestB ; return Imax ;
}
```

Criteria

- C_1: length
- C_2: ability to be filled later on
  - $C_2 = \sum_{n=1}^{\infty} A(n)\, 2^{-n}$
  - $A(n) = 1$ if smallestOut+n is allowed
  - $A(n) = 0$ otherwise

Evaluation with randomness

- Criterion C_3: random
  - C_3 = (rand()%100)/100
- Evaluation E
  - $E = C_1 + \lambda C_2 + (1-\lambda) C_3$

The abstract simulation for MCS

```
int Damier::abstractSimulation(int K)
{
   if (K<=3) { DFS(K) ; return smallestOut-1;  }

   for (stageN=0 ; stageN<=2^(K-2) -1 ; stageN++) {
      Kloc = 3+stageN2col(stageN) ;

      if (Kloc<=3) DFS(Kloc) ;
      else {
         Hole = holeRandomChoice() ;
         fillColWithHole(Kloc, Hole) ;
         fillColConsecutiveNum(Kloc);
         fillEndCol(Kloc) ;
      }
   }
   return smallestOut - 1;
}
```

New values for K=7, 8

K	1	2	3	4	5	6	7	8
WS(K)	=2	=8	=23	=66	>=196	>=582	>=1736	>=5105

- K=7
  - Abstract MCS at level 3, and RWS
  - Few hours
- K=8
  - Abstract MCS at level 2, and RWS
  - One day

K=4, 5

1 2 4 8 11 16 22 25 32 44 53 58 63
3 5-7 19 21 23 38 39 50-52 64-**66**
9 10 12-15 17 18 20 54-57 59-62
24 26-31 33-37 40-43 45-49

2 4 8 11 16 22 25 31 45 50 60 63 69 106 135 140 150 155 178 183 **196**
3 5-7 19 21 23 35 51-53 64-66 77-79 137-139 151-153 180-182 193-195
9 10 12-15 17 18 20 54-59 61 62 99-105 141-149 184-192
24 26-30 32-34 36-44 46-49 98 154 156-177 179
67 68 70-76 80-97 107-134 136

K=6 (Eliahou 2013)

1-2 4 8 11 22 25 40 50 63 68 73 82 87 92 97 116 121 133 139 149 154 159 177 182 187 192 197 252 304 342 370 394 407 412 417 435 440 445 450 455 464 469 474 479 488 493 502 507 521 526 536 541 554 564 569 **582**
3 5-7 19 21 23 37 51-53 64-66 79-81 93-95 109-111 122-124 136-138 150-152 167-168 179-181 193-195 368 395-397 408-410 424-425 437-439 451-453 465-467 480-482 495-497 512 523-525 537-539 551-553 566-568 579-581
9-10 12-18 20 54-62 103-108 140-148 183-186 188-191 398-406 441-444 446-449 486-487 490 492 494 527-535 570-578
24 26-36 38-39 41-49 98-102 153 155-158 160-166 169-176 178 292 411 413-416 418-423 426-434 436 540 542-550 555-563 565
67 69-72 74-78 83-86 88-91 96 112-115 117-120 125-132 134-135 454 456-463 468 470-473 475-478 483-485 489 491 498-501 503-506 508-511 513-520 522
196 198-251 253-291 293-303 305-341 343-367 369 371-393

K=6 (Bouzy 2014)

1-2 4 8 11 16 22 25 53 63 68 136 149 154 177 182 192 197 394 407 412 435 440 450 455 521 526 531 536 541 564 569 **582**
3 5-7 19 21 23 50-52 64-66 137-139 150-152 179-181 193-195 395-397 408-410 437-439 451-453 523-525 537-539 566-568 579-581
9-10 12-15 17-18 20 54-62 140-148 183-191 398-406 441-449 527-530 532-535 570-578
24 26-49 153 155-176 178 411 413-434 436 540 542-563 565
67 69-135 454 456-520 522
196 198-393

K=7 (Bouzy 2014)

1-2 4 8 11 16 22 25 50 63 68 136 149 154 177 182 187 192 197 394 397 407 412 435 440 450 455 521 526 531 536 541 564 569 582 1170 1175 1180 1185 1208 1213 1218 1223 1228 1294 1299 1309 1314 1337 1342 1355 1555 1565 1570 1593 1598 1603 1608 1613 1679 1684 1694 1699 1722 1727
3 5-7 19 21 23 51-53 64-66 137-139 150-152 179-181 193-195 395-396 408-410 437-439 451-453 523-525 537-539 566-568 579-581 1167-1169 1181-1183 1210-1212 1224-1226 1296-1298 1310-1312 1339-1341 1352-1354 1552-1554 1566-1568 1595-1597 1609-1611 1681-1683 1695-1697 1724-1726
9-10 12-15 17-18 20 54-62 140-148 183-186 188-191 398-406 441-449 527-530 532-535 570-578 1171-1174 1176-1179 1214-1217 1219-1222 1300-1308 1343-1351 1556-1564 1599-1602 1604-1607 1685-1693 1728-**1736**
24 26-49 153 155-176 178 411 413-434 436 540 542-563 565 1184 1186-1207 1209 1313 1315-1336 1338 1569 1571-1592 1594 1698 1700-1721 1723
67 69-135 454 456-520 522 1227 1229-1293 1295 1612 1614-1678 1680
196 198-393 1356-1551
583-1166

K=8 (Bouzy 2014)

1-2 4 8 11 22 52 55 62 67 137 147 152 178 181 188 193 389 399 405 430 435 440 445 510 515 525 531 556 559 566 571 1145 1155 1161 1186 1189 1196 1201 1266 1271 1281 1286 1309 1315 1322 1327 1518 1523 1533 1561 1564 1567 1574 1579 1644 1649 1659 1687 1690 1700 1705 3410 3413 3423 3451 3454 3457 3464 3469 3534 3539 3549 3555 3580 3585 3590 3595 3786 3791 3801 3806 3832 3835 3842 3847 3912 3917 3927 3932 3955 3968 3973 4542 4547 4552 4557 4562 4588 4591 4598 4603 4668 4673 4683 4711 4724 4729 4920 4925 4935 4966 4976 4981 5046 5051 5061 5066 5089 5102
3 5-7 19 21 23 49-51 63-65 134-136 148-150 175-177 189-191 386-388 400-402 427-429 441-443 512-514 526-528 553-555 567-569 1142-1144 1156-1158 1183-1185 1197-1199 1268-1270 1282-1284 1310-1312 1323-1325 1520-1522 1534-1536 1562-1563 1575-1577 1646-1648 1660-1662 1688-1689 1701-1703 3411-3412 3424-3426 3452-3453 3465-3467 3536-3538 3550-3552 3577-3579 3591-3593 3788-3790 3802-3804 3829-3831 3843-3845 3914-3916 3928-3930 3956-3958 3969-3971 4544-4546 4558-4560 4585-4587 4599-4601 4670-4672 4684-4686 4712-4714 4725-4727 4922-4924 4936-4938 4963-4965 4977-4979 5048-5050 5062-5064 5090-5092 5103-**5105**
9-10 12-18 20 53-54 56-61 138-146 179-180 182-187 390-398 431-434 436-439 516-524 557-558 560-565 1146-1154 1187-1188 1190-1195 1272-1280 1313-1314 1316-1321 1524-1532 1565-1566 1568-1573 1650-1658 1691-1699 3414-3422 3455-3456 3458-3463 3540-3548 3581-3584 3586-3589 3792-3800 3833-3834 3836-3841 3918-3926 3959-3967 4548-4551 4553-4556 4589-4590 4592-4597 4674-4682 4715-4723 4926-4934 4967-4975 5052-5060 5093-5101
24-48 151 153-174 403-404 406-426 529-530 532-552 1159-1160 1162-1182 1285 1287-1308 1537-1560 1663-1686 3427-3450 3553-3554 3556-3576 3805 3807-3828 3931 3933-3954 4561 4563-4584 4687-4710 4939-4962 5065 5067-5088
66 68-133 444 446-509 511 1200 1202-1265 1267 1578 1580-1643 1645 3468 3470-3533 3535 3846 3848-3911 3913 4602 4604-4667 4669 4980 4982-5045 5047
192 194-385 1326 1328-1517 1519 3594 3596-3785 3787 4728 4730-4919 4921
570 572-1141 3972 3974-4541 4543
1704 1706-3409

Discussion

- Our solutions are more compact
- Abstraction with holes is important
- Is 196 optimal for K=5 ? 582 for K=6 ?
- How optimal are our solutions for K=7, 8 ?
- Is the method complete ?
- Is the method adapted for True Schur ? No

Thank you for your attention!