PARTITIONS INTO SUM-FREE SETS

Peter F. Blanchard
Miami University, Hamilton, OH 45011, USA
blanchpf@muohio.edu

Frank Harary[1]
New Mexico State University, Las Cruces, NM 88003, USA
fnh@crl.nmsu.edu

Rogério Reis
Universidade do Porto, DCC-FC & LIACC, Porto, Portugal
rvr@ncc.up.pt

Received: 10/23/04, Revised: 7/11/05, Accepted: 2/4/06, Published: 3/9/06

Abstract

We define a sum as a set $\{x, y, z\}$ of distinct natural numbers such that $x + y = z$, and let $N_m = \{1, 2, \ldots, m\}$. We introduce a new sequence $h(n)$ defined as the smallest s such that there is no partition of N_s into n sum-free parts. We determine $h(n)$ for $n = 3, 4$ after easily noting that $h(1) = 3$ and $h(2) = 9$. We find that $h(3) = 24$ and $h(4) = 67$ using a computer program.

1. Introduction

In this paper we investigate a variation on a celebrated theorem of Issai Schur. In what follows, $N = \{1, 2, 3, \ldots\}$ refers to the set of positive integers, and $N_m = \{1, 2, 3, \ldots, m\}$ denotes an initial segment of length m. A *finite coloring* of a set A is simply a partition of A into finitely many disjoint subsets: $A = A_1 \cup A_2 \cup \cdots \cup A_m$. One thinks of the elements of A_i as having color i. Alternately, such a partition defines a function $\pi : A \to N_n$. The color classes then are the inverse images $A_i = \pi^{-1}(i)$. We will also refer to this as an m-coloring of A. Any set $B \subseteq A_i$ is referred to as *monochromatic*, as all of its elements are of the same color.

Ramsey Theory on the natural numbers has a long and varied history. Nine pages into Graham, Rothschild, and Spencer's [3] standard reference on the subject, one finds the

[1]Frank passed away January 4, 2005. The date 1-4-'05 is a triple of the type studied in this paper.

theorems of Van der Waerden, Schur, and Rado listed among the 'Super Six'. Each concerns colorings of N and asserts the existence of some type of monochromatic set in any finite coloring of N_n if n is sufficiently large. Briefly, Van der Waerden's theorem asserts the existence of monochromatic arithmetic progressions, Schur's theorem asserts the existence of monochromatic solutions to $x + y - z = 0$, and Rado's theorem extends Schur's theorem to systems of linear equations.

Theorem (Schur's Theorem). *For any $m > 0$, there is a finite natural number $s = s(m)$ so that for any finite coloring $\pi : N_s \to N_m$ there exist x, y so that $\{x, y, x + y\}$ is monochromatic.*

The standard derivation of Schur's Theorem from Ramsey's Theorem is as follows: Given an m-coloring $\pi : [n] \to [m]$, construct an m-coloring $\pi' : K_{n+1} \to [m]$ of a complete graph on the vertices $[1, n + 1]$ so that each edge ij is colored $\pi(j - i)$, i.e. $\pi'(ij) = \pi(j - i)$. If n is large enough, Ramsey's theorem asserts the existence of vertices a, b, c, such that $\pi'(ab) = \pi'(bc) = \pi'(ac)$. Now $(b - a) + (c - b) = (c - a)$ gives the required monochromatic solution $x + y = z$ where $x = b - a, y = c - b$, and $z = c - a$. In this case, nothing prevents $x = y$. Indeed, in the case of 2 colors, $s(2) = 5$, and it is not hard to see 3/4 of all 2-colorings of $[1, 5]$ have monochromatic solutions of the form $x + x = z$.

In the following theorem, we explore here the case in which $x \neq y$.

Theorem 1. *For any $m > 0$, there is a finite natural number $h = h(m)$ so that for any finite coloring $\pi : N_h \to N_m$ there exist $x \neq y$ so that $\{x, y, x + y\}$ is monochromatic. Moreover, $h(1) = 3$, $h(2) = 9$, $h(3) = 24$, and $h(4) = 67$.*

In Section 3 we discuss the three smallest values in the context of combinatorial games. Sections 2 and 3 for computing $h(m)$. Sections 6 and 7 concern values of $h(m)$ for $m \geq 4$, while Section 8 contains further discussion relating values $h(m)$ to combinatorial games. The appendix contains C-language computer code implementing the algorithm discussed earlier.

2. Proof of Theorem 1

Proof. Ramsey's theorem [3] guarantees the existence of a number $t = R(l; m)$ so that any m-coloring of (the edges of) a complete graph K_t must have a monochromatic collection of edges forming a complete graph on l vertices. In particular, given, $m > 0$, let $t = R(4; m)$. We claim that $h(m) \leq t - 1$. For any m-coloring $\pi : [t - 1] \to [m]$, define an m-coloring $\pi' : K_t \to [m]$ by $\pi'(ij) = \pi(|j - i|)$. Here ij represents the edge of K_q connecting vertices i and j. π' is well-defined as $1 \leq |j-i| \leq t-1$. By Ramsey's theorem, there is a set $\{w, x, y, z\}$ of vertices forming a monochromatic K_4. Since the edges are monochromatic in the coloring π', it follows that the differences $x - w, y - x, y - w, z - y, z - x, z - w$ are monochromatic for π. If $w - x = y - x = z - y = a$ then $y - w = 2a$ and $z - w = (z - y) + (y - w)$ yields the monochromatic triple $\{a, 2a, 3a\}$. Otherwise, either $x - w \neq y - x$ yields the monochromatic triple $\{x - w, y - x, y - w\}$ or $z - y \neq y - x$, yields $\{y - x, z - y, z - x\}$. $\square$

3. The Three Smallest h-numbers

Consider a two person game, with players A and B, that begins with a small number of empty columns in which A places number 1 in any column, then B places number 2 in either column, etc. The column in which a number is placed must not be the sum of two numbers above. The last player who can move is the winner W.

Here is an example of a game with 3 columns, in which player A places 1 in column C_1 and wins because eventually player B cannot place number 12 in any column. Note that $3'$ means that 3 is forced to be placed in a different column, and $8^\star$ means 8 is forced in a unique column.

C₁	**C₂**	**C₃**
1	$3'$	4
2	5	6
7	9	$8^\star$
10		11

This game is completed since number 12 cannot be placed, so $W = A$.

We define a sum as a set $\{x, y, z\}$ of distinct natural numbers such that $x + y = z$. In this context we think of $h(n)$ as the smallest m such that there is no partition of N_m into n sum-free parts.

Trivially,

$$h(1) = 3 \tag{1}$$

with the only solution:

C₁
$1^\star$
$2^\star$

.

In the same way it is easy to verify that

$$h(2) = 9 \tag{2}$$

and in this case also there is only one solution:

$\mathbf{C_1}$	$\mathbf{C_2}$
1	$3^\star$
2	$5^\star$
4	$6^\star$
$8^\star$	7

.

This shows that $h(2) > 8$. Only a few cases must be considered to see that $h(2) = 9$. Let $\pi : N_9 \to N_2$ be a placement in the 2 columns.

If 1 and 2 are in two different columns we are quickly done. If $\pi(1) = 1$, $\pi(2) = 2$ and $\pi(i) = 1$ for any $i \geq 4$, then $\pi(i \pm 1) = 2$ leads to the sum $\{2, i-1, i+1\}$ in the same column. Avoiding $\pi(i) = 1$ leads to $\pi(4) = \pi(5) = \pi(6) = 2$ which fails, because then the sum $\{2, 4, 6\}$ would be in in the same column.

Thus let $\pi(1) = \pi(2) = 1$ so $\pi(3)$ must be 2. If $\pi(4) = 1$, then to avoid sums $\{1, 4, 5\}$ and $\{2, 4, 6\}$, $\pi(5) = \pi(6) = 2$ are forced, and then because 3, 5 and 6 are in column 2, we would need $\pi(8) = \pi(9) = 1$, but then the sum $\{1, 8, 9\}$ would be in column 1.

Continuing to assume $\pi(1) = \pi(2) = 1$ and $\pi(3) = 2$, if $\pi(4) = 2$, then to avoid the sum $\{3, 4, 7\}$ in the second column, $\pi(7) = 1$ is forced, which in turn forces $\pi(6) = \pi(8) = 2$, but now we would get either $\{3, 6, 9\}$ or $\{2, 7, 9\}$ in the same column.

It was proven similarly that for $c = 3$ columns the maximum possible length of the game is 23, so

$$h(3) = 24. \tag{3}$$

All three examples of such games are

$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
1	$3'$	$9^\star$
2	$5'$	$10^\star$
4	6	$12^\star$
$8'$	7	$13^\star$
11	$19^\star$	14
16	$21'$	15
$22^\star$	$23^\star$	$17'$
		18
		20

$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
1	$3'$	$9^\star$
2	$5'$	$10^\star$
4	6	$12^\star$
$8'$	7	$13^\star$
11	$19^\star$	14
17	$21^\star$	15
$22^\star$	$23^\star$	16
		$18'$
		20

$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$
1	$3'$	$9^\star$
2	$5'$	$10^\star$
4	6	$12^\star$
$8'$	7	$13^\star$
11	$19^\star$	14
$22^\star$	$21'$	15
	$23^\star$	16
		17
		18
		20

These examples can be simply represented by their column sequences:

[11212221331333313323212]
[11212221331333331323212]
[11212221331333333323212]

4. Brute Force Approach

A brute force approach to the problem is very easy to state: a program that tries every possible different choice of placement of each integer, with a depth-first search for example, that covers all possible game configurations. The longest one is the answer to our question. To be sure that the answer we are getting from the computer, is right, and not only a bunch of different game configurations, we only need to verify that the program does not skip any possibility. That is the purpose of the following two sections.

It is easy to see that the whole game configuration can be represented by the sequence of columns chosen by the players to place the numbers, as those numbers are determined by the natural order of N. With this observation the longest games for the problem $c = 3$ can be completely described by the strings above.

As usual denote the Kleene closure [5] of A by $A^\star$; game descriptions can be seen as members of $N_n^\star$, that is, the set of sequences of elements of N_n. To simplify the representation, and because we will need computationally fast implementations of these data structures, let us assume we have an upper bound for the maximum length of the game for a given n. Let us represent that limit by `L`. Then, we concatenate a string of zeros to a string description of a game, so that the result always has length `L`. Now we can see a game description as a member of

$$(\{0\} \cup N_n)^{\mathtt{L}}.$$

This can be easily and efficiently represented in a computer data structure.

Here brute force means to generate all the possible elements of

$$(\{0\} \cup N_n)^{\mathtt{L}} \cap (N_n)^\star \{0\}^\star$$

that stand for legal configurations of the game.

The problem is that the cost of computing whether an integer is placeable in a given column is too high. A direct evaluation from the data of the game configuration, for the mth placement will have to compute in the worst case $(m-1)(m-2)$ additions and comparisons, for a configuration with all previous placements in the same column. We can estimate an average of $n(\frac{m}{n})^2$, supposing an even distribution of placements by the different columns. This is clearly too much for a computation that, in the case $n = 4$, is going to be repeated a number of times that we only know[2] to be bounded by 4^{67}.

A solution for this problem is to use another data structure, that can store the "forbidden values" for each column, and that can be calculated in an incremental way. For each column we can store the values that can be obtained by adding two of the members of the column. We call that data structure `Base` and use it so that `Base[i][j]` is 0 means that the number `j` can be "legally" placed in column `i+1`.

[2]Assuming that the solution already known for the problem is indeed the longest.

Testing whether a move is legal is now much more efficient as its order is $O(m)$. But the backtracking of a move will still take too much time. We are already storing in `Base[col][i]` not simply a 1 when the integer `i` is "forbidden" in that `col`, but the number of different ways it is possible to write it as a sum of two elements of the same column. In this way, backtracking will be easier when one of the contributing parcels of `i` is to be removed from column `col` as we only need subtract 1 from this value. We keep, in another data array, the list of integers that each `i` contributes to "outcast," then backtracking can be done with minimal computational effort, that is, $O(m)$.

5. Trimming the Tree

We can significantly improve the computational speed by pruning the search tree whenever we know that all new solutions will be permutations of the ones already found.

Without loss of generality we can place the number 1 in column C_1 and save time by reducing the tree to $1/n$ (in this case $1/4$). But this method can be generalized, resulting in a much more effective pruning.

Let us consider again the domain

$$(\{0\} \cup N_n)^{\mathtt{L}} \cap N_n^{\star}\{0\}^{\star}.$$

Our depth-first descent corresponds to the lexicographic enumeration. We denote by $[a/b]s$ the application of the substitution of a for each b in the string s. Then we observe that for any string

$$s = a_1 a_2 \cdots a_{m-1} a_m a_{m+1} \cdots a_k$$

let

$$s' = [(max\{a_1, \ldots, a_{m-1}\} + 1)/a_m, a_m/(max\{a_1, \ldots, a_{m-1}\} + 1)]s$$

if $a_m > 1 + max\{a_1, \ldots, a_{m-1}\}$, then

$$s' < s.$$

The placement of number 1 in the first column, can be seen as a special case of this last observation.

The domain where we need to perform the brute force search is restricted to

$$(\{0\} \cup N_n)^{\mathtt{L}} \cap \left(\left(\bigcup_{i=1}^{c} \bigotimes_{j=1}^{i} \{j\} N_j^{\star} \right) \{0\}^{\star} \right),$$

where $\otimes$ stands for concatenation. An exact evaluation of this effort and the density of these languages was presented by Moreira and Reis [6].

We can trim the search tree even more. Notice that, when placing an integer in a column, and marking all the consequent forbidden numbers in that column, if it becomes impossible to place an integer in the range where we still expect solutions, then we know that we are on the wrong track.

6. The Main Computation: $h(4)$

Running the program for $c = 4$ took 120,231s (not much more than 34h) in an *Intel Pentium 4* with a clock rate of 3.0GHz and listed the 29,931 maximum game configurations with length 66.

We will not list the complete set of configurations but can present the first of the long list of the 29,931 solutions:

[112122213313333133232124144444144
4221441441444444122233313313 31222]

Thus we have confirmed that indeed

$$h(4) = 67. \tag{4}$$

7. Higher Numbers

Appending to the above solution the string of values

$$5^{68}1^{1}2^{3}3^{9}1^{1}2^{3}4^{24}1^{1}2^{3}3^{9}1^{1}$$

where exponents indicate repetition, it is not hard to verify that one gets a 5-column solution of length 189, hence $h(5) \geq 190$.

For any value n, the number of admissible triples $\{x, y, x+y\}$ for $h(n)$ is $\lfloor \frac{n^2-2n+1}{4} \rfloor$. Each of these is also admissible for $s(n)$, as are the additional $\lfloor \frac{n}{2} \rfloor$ triples of the form $\{x, x, 2x\}$ for $s(n)$. The ratio of sizes of sets of these two types of admissible triples clearly approaches 1 as n increases, hence we make the

Conjecture.

$$\lim_{n\to\infty} \frac{h(n)}{s(n)} = \lim_{n\to\infty} \frac{s(n)}{h(n)}.$$

Known values support this conjecture:

	1	2	3	4	5
$s(n)$	2	5	14	45	$161 \leq s(5) \leq 322$
$h(n)$	3	9	24	67	≥ 190
$h(n)/s(n)$	1.5	1.8	1.71	1.49	

The last entry in the first row is from Exoo [2].

In the book [3], it is stated that the utterly trivial value of $s(1)$ is 2 and obviously $s(2) = 5$. Then $s(3) = 14$ was easily obtained by hand. However $s(4) = 45$ required a computer program! With our program the value of $s(4)$ can be verified in only 16 seconds[3]!!

8. On the Sum-free Games

In an *achievement game*, the last player who can move wins; in the corresponding *avoidance game*, he/she loses. We found in [4] that in the 2-column h-game, B wins achievement and A wins avoidance! On a variation of this game by Curtin and Harary [1], the first move by A selects any number in N_8 and places it in either of two columns. Then B picks one of the remaining seven numbers and puts it in one of the columns so that both columns remain sum-free. As before, the last player who can move wins. We proved that A can win regardless of his first move but this is demonstrated most easily when his first move is 7.

In the definition of the Schur numbers $s(1), s(2), s(3), s(4), ...$, the exclusion of a sum $x + y = z$ permits $x = y$. Thus $s(1) = 2$ since after writing 1 in the one column, 2 cannot be added. Similarly $s(2) = 5$. Therefore a nontrivial 2-player Schur game must have at least three columns. We plan to investigate this and other similar games later.

Appendix A: The Program Code

```
#include <stdio.h>
#define MAX(X,Y) ((X>Y)?X:Y)
#define MIN(X,Y) ((X>Y)?Y:X)

#define COLS 4
#define L 68 // upper bound to exptd sols
#define LL 64 // lower bound intrstng sols

#define excludedp(N) (Base[0][N]&Base[1][N]&Base[2][N]&Base[3][N]&Base[4][N])

#define colp(X) (!Base[X][wmax+1])
```

[3] Using the same *Intel Pentium 4* at 3.0GHz.

```
short int Base[COLS][L+1], wsol[L];
int nums[L+1][L+1], wmax = 0, max=0;

void updatesol(){
  int i, max=0;
  static lprint = 0;

  max = wmax;
   if(max > LL && max >= lprint){
    for(i=1;i<=max;i++) printf("%d",wsol[i]+1);
    printf(" %d{}n",wmax+1); lprint = max;
   }
 fflush(stdout);
}

int add_col(int col){
 int i, j=1, sum, ex=0;
 wsol[++wmax] = col;
 if (wmax >= max) updatesol();

 for(i=1;i< MIN(wmax,L-wmax) ;i++){
  sum = wmax + i ;
  if ((wsol[i]==col)){
   nums[wmax][j++] = sum;
   Base[col][sum]++;
   if((sum < MAX(LL,max)) && excludedp(sum))
    ex++;
  }
 }
 return(ex);
}

void del_col(void){
 int i=1, col;

 col = wsol[wmax];
 while (nums[wmax][i]){
  Base[col][nums[wmax][i]]--;
  nums[wmax][i++] = 0;
 }
 wmax--;
}

void brute_force(int b){
 int i;

 for(i=0;i<= MIN(COLS-1,b+1);i++)
  if(colp(i)){
   if(!add_col(i))
    brute_force(MAX(i,b));
   del_col();
```

```
  }
}

int main(int argc, char **argv){
 int i, j;
 char *p;

 for(i=0;i<COLS;i++)
  for(j=0;j<=L;j++)
   Base[i][j] = 0;
 max=LL;

 brute_force(-1);
}
```

Acknowledgments

Desh Ranjan and Gopal Gupta earlier constructed a 4-column solution of length 66 and conjectured correctly that this is the best possible.

We thank Nelma Moreira and Joo Pedro Pedroso for discussions about the preliminary versions of the program and paper.

Support for this project was provided in part by the Ohio Supercomputer Center through a Cluster Ohio Grant (Rev2, 2003) to the Miami University Center for the Advancement of Computational Research.

References

[1] E. Curtin and F. Harary. Another sum-free game. *J. Recrational Math.*, 2004. (to appear).

[2] Geoffrey Exoo. A lower bound for Schur numbers and multicolor Ramsey numbers of K_3. *Electronic J. Combinatorics*, 1(1), 1994.

[3] R. L. Graham, B. L. Rothschild, and J. H. Spencer. *Ramsey Theory.* Wiley, New York, 1980.

[4] F. Harary. Sum-free games. In T. Rodgers and T. Wolfe, editors, *Puzzlers' Tribute: A feast for the mind*, pages 395–398. A.K.Peters, Natick, MA, 2002.

[5] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, 1956.

[6] Nelma Moreira and Rogério Reis. On the density of languages representing finite set partitions. *Journal of Integer Sequences*, 8(05.2.8), 2005.