A Multilevel Tabu Search with Backtracking for Exploring Weak Schur Numbers

Denis Robilliard, Cyril Fonlupt, Virginie Marion-Poty, and Amine Boumaza

{robilliard,fonlupt,poty,boumaza}@lisic.univ-littoral.fr
Univ Lille Nord de France
ULCO, LISIC, BP 719
62228 CALAIS Cedex, France

Abstract. In the field of Ramsey theory, the *weak Schur number* $WS(k)$ is the largest integer n for which their exists a partition into k subsets of the integers $[1, n]$ such that there is no $x < y < z$ all in the same subset with $x + y = z$. Although studied since 1941, only the weak Schur numbers $WS(1)$ through $WS(4)$ are precisely known, for $k \geq 5$ the $WS(k)$ are only bracketed within rather loose bounds. We tackle this problem with a tabu search scheme, enhanced by a multilevel and backtracking mechanism. While heuristic approaches cannot definitely settle the value of weak Schur numbers, they can improve the lower bounds by finding suitable partitions, which in turn can provide ideas on the structure of the problem. In particular we exhibit a suitable 6-partition of $[1, 574]$ obtained by tabu search, improving on the current best lower bound for $WS(6)$.

Keywords: tabu search, weak Schur numbers, optimization

1 Introduction

1.1 Mathematical description

Ramsey theory is a branch of mathematics that studies the existence of orderly sub-structures in large chaotic structures: *"complete disorder is impossible"*[1]. Many questions in Ramsey theory remain open, and they are typically phrased as *"how many elements of some structure must there be to guarantee that a particular property will hold?"*.

In this paper we focus on the following Ramsey theory problem: finding weak Schur numbers. A set P of integers is called *sum-free* if it contains no elements $x, y, z \in P$ such that $x + y = z$. A theorem of Schur [1] states that, given $k \geq 1$, there is a largest integer n for which the integer interval set $[1, n]$ (i.e. $\{1, 2, \ldots, n\}$) admits a partition into k sum-free sets. This largest integer n is called the k-th Schur number, denoted by $S(k)$.

Adding the assumption that x, y, z must be *pairwise distinct*, i.e. $x \neq y, y \neq z, x \neq z$, we define *weakly sum-free* sets, and a similar result was shown by

[1] Quote attributed to Theodore Motzkin.

Rado [2] : given $k \geq 1$, there is a largest integer n for which the interval set $[1, n]$ admits a partition into k weakly sum-free sets: this largest integer n is the weak Schur number denoted $WS(k)$.

For example, in the case $k = 2$, a weakly sum-free partition of the first 8 integers is provided by:

$$\{1, 2, 3, 4, 5, 6, 7, 8\} = \{1, 2, 4, 8\} \cup \{3, 5, 6, 7\}$$

It is straightforward to verify that increasing the interval to $[1, 9]$ yields a set that does not admit any weakly sum-free partition into 2 sets, thus we have $WS(2) = 8$.

Very few exact values are known for $WS(k)$ (or for $S(k)$):

- $WS(1) = 2$ and $WS(2) = 8$ are easily verified
- $WS(3) = 23$, $WS(4) = 66$ were shown by exhaustive computer search in [3].

The same study [3] shows the following lower bound: $WS(5) \geq 189$, while a much older note by Walker [4] claimed, without proof, that $WS(5) = 196$. Recently a paper by Eliahou, Marín, Revuelta and Sanz [5] provided a weakly sum-free partition of the set $[1, 196]$ in 5 sets, confirming Walker's claim that $WS(5)$ is at least as large as 196, and also gave a weakly sum-free partition of $[1, 572]$ in 6 sets, thus establishing that:

$$WS(5) \geq 196$$

$$WS(6) \geq 572$$

For the general case $k \geq 5$, the best known results are a lower bound from Abbot and Hanson [6] and an upper bound by Bornsztein [7]:

$$c89^{k/4} \leq S(k) \leq WS(k) \leq \lfloor k!ke \rfloor$$

with c a small positive constant.

1.2 Tabu Search

Local search methods are among the simplest iterative search methods. In local search, for a given solution i a neighborhood $N(i)$ is defined, and the next solution is searched among the solutions in $N(i)$.

Tabu search (TS) is based on local search principles. It was introduced by Glover [8]. Unlike the well-known hill-climber search, the current solution may deteriorate from one iteration to the next to get out of a local optimum. In order to avoid possible cycling in the vicinity of a local optimum, tabu search introduces the notion of tabu list, which prohibits moves towards solutions which were recently explored. The duration for a move (or attribute of a move) to remain tabu is called the tabu-tenure. However, the tabu list can be overridden if some conditions, called aspiration criterion, are met. For instance, a typical aspiration criterion is the discovery of a new best solution (better than any previously visited solution): the move towards such a solution is retained even if it is forbidden by the tabu list. TS was improved with many other schemes like intensification (to focus on some part of the search space) or diversification (to explore some new part of the search space). Details about tabu search and enhancements of tabu search can be found in [9, 10].

In the case of the weak Schur numbers, the search space size, including non feasible solutions, is $k^{WS(k)}$, where $WS(k)$ is at least polynomial in k. Indeed, even with small values of k a basic tabu search approach is unable to solve the problem in reasonable time limits, as shown in Section 3.

The basic idea of a multilevel approach is to create a sequence of successive coarser approximations of the problem to be solved in reverse order. The multilevel scheme was already successfully used for solving some combinatorial optimization problems [11]. In our case, we skip the explicit construction of the sequence, since it can be obtained naturally by considering each $WS(k)$ problem as a coarser version of step $WS(k+1)$, as explained later in Section 2.1: we start with $WS(2)$ as the coarsest level and we solve the sequence of problems of increasing difficulty obtained by adding successive partitions, i.e. increasing the value of k. Our method also differs from the original multilevel in that it does not create new temporary coarse variables that encapsulate several variables of the refined problem. Here a coarser version of the problem is coarse in the sense that it contains only a small part of the variables from the complete refined version. Nonetheless we think that it borrows enough inspiration from this scheme (increasing number of variables in several large steps, earlier choices remaining fixed and interfering with later ones) to still be qualified as multilevel, and we will keep this denomination through this paper.

Anyway the multilevel scheme proved useful but not yet sufficient, due to the fact that some intermediate stage may not possibly be refined to the next level. We added a backtracking mechanism to allow the search algorithm to rewind back and introduce more diversity into previous levels.

The rest of the paper is organized as follows. In Sect. 2, we detail our approach, with a description of the multilevel framework and the backtracking mechanism. In Sect. 3, we present the new results that were discovered using our enhanced tabu scheme.

2 Detailed Framework

2.1 Problem description and multilevel paradigm

As said in the Introduction, our aim is to find weakly sum-free partitions for the currently known lower bounds for $WS(5)$ and $WS(6)$, and possibly improve (increase) these lower bounds.

In [3], the authors pointed out that there exists 29931 different possible solutions to the $WS(4)$ problem (there are 29931 partitions of $[1, WS(4)]$ into four weakly sum-free subsets). Eliahou and Chappelon [12] found one interesting point: in all these 29931 solutions, the first 23 integers are always stored in only

3 of the 4 sets. Indeed, since $WS(3) = 23$, all the partitions that solve the $WS(4)$ problem are **extensions** of partitions that solve the $WS(3)$ problem. They also noticed that only 2 of the 3 possible weakly sum-free partitions of $[1, WS(3)]$ can be extended to form a weakly sum-free partition of $[1, WS(4)]$.

Eliahou *et al.* in [5] used the `march-pl` SAT solver [13], with a suitable translation of the problem in CNF form to study weak Schur numbers. As the search space increases exponentially with higher values of k for $WS(k)$, the authors could not explore in an exhaustive way all possible solutions. So they conjectured that the property that was exhibited from $WS(3)$ to $WS(4)$ could also hold true from $WS(4)$ to $WS(5)$. So when searching for $WS(5)$, Eliahou *et al.* first searched for solutions to $WS(4)$, then froze the 66 first integers into their respective sets (since $WS(4) = 66$), before searching for $WS(5)$. Conjecturing $WS(5) = 196$ leaves 130 remaining integers to assign, thus greatly reducing the size of the search space. Anyway this reduction was not yet enough to use the `march-pl` solver, and the authors of [5] had to bet on the position of part of the remaining 130 numbers in order to lower even further the computational cost. These hypotheses led them to discover a 5-partition of $[1, 196]$, proving $WS(5) \geq 196$ as part of G. W. Walker's claim. Using further placement hypotheses, they also obtained a 6-partition of $[1, 572]$, proving $WS(6) \geq 572$.

In this paper, we use the same multilevel approach for studying $WS(5)$ and $WS(6)$ by way of tabu search. When a weakly sum-free 4-partition is found for $[1, WS(4)]$, the solution is kept as an un-mutable part of the 5-partitions of $[1, x]$, with $x \geq 196$ the best known lower bound for $WS(5)$. When such a weakly sum-free 5-partition is found, it is again kept unchanged to form the basis of 6-partitions of $[1, y]$, with $y \geq 572$ the best current value for $WS(6)$.

Because some solutions at level k may be impossible to extend to give a level $(k + 1)$ solution, we enhance this technique with the following scheme: when a fixed number of trials fails to give a weakly sum-free $(k+1)$-partition, we assume that we may be trapped in a local optimum. Thus the algorithm is allowed to backtrack to the level k to find a new weakly sum-free k-partition. The details of the algorithm can be found in the next subsection.

2.2 Multilevel Tabu Search with Backtracking

The main components of our tabu search are given here, while the pseudo-code is available in Table 2.

Representation: a solution to the $WS(k)$ problem consists in a partition of the set $[1, n]$ into k sets (possibly non weakly sum-free).

Fitness function: the fitness function is simply the number of integers that are involved in a constraint violation. For example, if 2, 8 and 10 are in the same set, this implies the fitness is increased by 3 (as $2 + 8 = 10$). A perfect solution has a fitness $= 0$, i.e. whenever no constraint violations remain.
Formally, let c be a partition of $[1, n]$ into k sets (i.e. a solution), and let $P_i(c)$ be the i^{th} set of the k-partition of c:

$$f(c) = 3 * \sum_{i=1}^{k} |\{x \in P_i(c) \mid \exists y, z \in P_i(c), x < y < z, x + y = z\}|$$

Move operator: a move operator should transform a solution in such a way that it implies only a small change in the search space. In our case, we chose to move an integer from a partition to another. This is performed in two steps: removing the integer from its origin set, and adding it to its destination set (different from its origin set).

The move operator defines the notion of move gain, indicating how much the fitness of a solution is improved (or downgraded) according to the optimization objective when moving an integer to another set. Here a negative move gain (fitness change) characterizes a move that leads to a better solution (minimization problem). The basic idea of the tabu search (ignoring tabu list and implementation details) is to select the move leading to the best solution, so the search complexity is $O(WS(k) * k)$.

A cheap computation of the move gain, if available, allows a fast, incremental evaluation of the fitness of solutions, hence a fast exploration of the neighborhood in search for the best non tabu neighbor.

In order to accelerate the computation of the move gain, a vector of constraint count is maintained for each set of the partition. This vector associates a number of constraints to every integer in the interval to be partitioned. More precisely, for each pair of integers x and y in the set, we associate one constraint to $x + y$, $y - x$ and $x - y$ (provided these positions are in the interval $[1, n]$ and $y \neq (x - y)$, $x \neq (y - x)$ to respect the pairwise distinct numbers assumption). This constraint count is performed even if the integer is not currently in the set, as illustrated in Table 1: this is akin to *forward arc checking* in constraint propagation algorithms.

- The move gain is obtained by subtracting the constraint count associated to the former position of the integer from the constraint count of its new position.
- The update of the constraint vector can be easily performed:
  - When removing an integer y from a set s, for each other integer x in set s, the number of constraints is decremented by one for all the following indices : $x + y$, $y - x$ and $x - y$ (provided these positions are in the interval $[1, n]$ and $y \neq (x - y)$, $x \neq (y - x)$ to respect the pairwise distinct numbers assumption) — these changes are done in the constraint vector associated to set s;
  - When adding an integer y to a set, the reverse operation is done, i.e. incrementing the constraint count by one at the same positions;
  - The two previous operations have a computational complexity in $O(WS(k))$.

Neighborhood: as explained before, we are exploring partitions of $[1, WS(k)]$ by extending already found weakly sum-free solutions for the $WS(k - 1)$ level. So only the integers in $[WS(k - 1) + 1, n]$ are allowed to be moved

Table 1. Illustration of the constraint count vectors for a 2-partition of $[1, 8]$.

set 1	indices	1	2	3	4	5	6	7	8
	integers in subset	X	X		X		X		
	constraint count	0	1	2	1	2	1	1	1
set 2	indices	1	2	3	4	5	6	7	8
	integers in subset			X		X		X	X
	constraint count	1	2	1	1	1	0	0	1

Explanation (on set 2):

- integer 1 has 1 pending constraint since if we move 1 to set 2 a conflict will occur with 7 and 8;
- integer 2 has 2 pending constraints due to the presence of respectively $\{3, 5\}$ and $\{5, 7\}$ in set 2;
- integers 3, 5 and 8 have 1 actual constraint each, since they are involved in the same constraint violation (and only this one);
- integer 4 has 1 pending constraint due to the presence of the pair $\{3, 7\}$ in set 2;
- integers 6 and 7 have no pending (nor actual) constraint (this notably implies that we can insert 6 in set 2 without degrading fitness).

in a $WS(k)$ level solution. The neighborhood of a solution is then the set of all possible solutions that can be reached in one application of the move operator on the allowed subset of integers.

Formally, the neighborhood $N(c)$ of a k-partition c is the set of solutions c': $N(c) = \{c' \mid \exists\,!\,x \in [WS(k-1)+1, n], i, j \in [1, k], i \neq j, x \in P_i(c), x \in P_j(c')\}$

Tabu list and tabu tenure management: each time an integer i is moved from a partition to another partition, it is forbidden to move again the integer i in any partition for the next tt iterations. Each time we launch the tabu search on a given sub-level of the problem, we set dynamically this tabu tenure tt to a random number between 2 and half the number of integers allowed to move.

Aspiration criterion: the aspiration criterion overrides the tabu list when a solution is better than the currently best-known solution.

Perturbation strategy: a random neighbor can always be chosen with a small probability, or else the algorithm chooses the non-tabu neighbor with the best gain move (taking into account the aspiration criterion). As said above we keep the previous sub-level solution unchanged in order to focus on critical elements and narrow the search space. To complement this search strategy, a rewind back mechanism is applied when no improvement to the current solution is detected for a given number of restarts of the tabu (see Sect. 3 for details). In that case, we conjecture that the algorithm is working on a non-extensible solution and we settle back to the previous $(k-1)$ sub-level problem.

Table 2. Pseudo-code for the tabu search algorithm.

```
Step 1: Initialization
Initialize solution s to WS(2)
Set k = 3 {initial number of partitions}

Step 2: Extension of solution
Initialize iteration_counter, tabu_list, tabu_tenure
Initialize n to either WS(k) if known, or the expected value of WS(k)
for each integer i such that WS(k − 1) < i ≤ n
    randomly assign i to a subset in [1, k]
end for
f = fitness(s)        { evaluate the fitness of s }
Repeat
    { either a random move or a neighborhood exploration }
    if (random in [0, 1] > 0.9)
      perform a random move
    else        { apply neighborhood search to s }
      Initialize δ to ∞ {worst possible move gain}
      for each integer i such that WS(k − 1) < i ≤ n
          { P_i is the current subset holding i }
          for all possible subset P′ ≠ P_i
              evaluate δ the fitness improvement when moving i to P′
              if ((δ is the greater so far) &&
                 (move not in tabu_list || aspiration criterion is true))
                 record this move as best_move
              end if
          end for
      end for
      perform best_ move
    end if
    update fitness f        { quick computation using move gain }
until stop condition
{ either WS(k) is solved or maximum # iterations is reached }

Step 3: Extension or rewind back
if (s solution to WS(k))
    k = k + 1
    Go to Step 2
else
    if (maximum number of trial runs for WS(k) is not yet reached)
      runs = runs + 1
      Go to Step 2
    else
      { seemingly this solution is not extensible, rewind back }
      k = k - 1
      reset number of trial for WS(k)
      Go to Step 2
    end if
end if
```

3 Experimental results

3.1 Experimental setting

Experiments were conducted with four different types of tabu: the first one is a "classic" tabu that tried to solve directly the $WS(5)$ problem. The second scheme is the same tabu with a restart strategy. The third is a multilevel approach that first tried to solve $WS(2)$, then $WS(3)$ and so on, also using restarts. The fourth scheme is the multilevel with backtracking introduced in Sect. 2.2, referred to as "MLB tabu", starting from $WS(2)$.

In order to allow a fair comparison of the four approaches, all methods were allowed the same total of 10.000.000 fitness evaluations. For the classic tabu with restarts, the multilevel and the backtracking scheme, the maximum number of iterations for a tabu trial is set to 1.000.000 before a restart occurs. For the backtracking scheme we allow $Max\_Run = 5$ restarts before backtracking. We conducted 30 independent runs for the four approaches (each run with a maximum number of evaluations of 10.000.000).

Note that the comparison setting could be deemed slightly biased in favor of the two classic tabu versions, as they directly start with the $WS(5)$ problem unlike the other approaches which start with the $WS(2)$ problem. As a fitness evaluation for the $WS(5)$ problem is actually more time consuming than for the $WS(2)$ problem, the classic versions are in fact also allowed more computing time.

3.2 Results and Analysis

Experiments on $WS(5)$: Table 3 summarizes the results of our experiments on the $WS(5)$ problem: we call "successful run" a run that reaches the current best known bound, 196. For each scheme we report the results obtained within 30 independent runs.

Table 3. Experimental results for Schur number $WS(5)$

	Average fitness	# of successful runs
Classic tabu	17.2	0
Classic tabu with restarts	8.3	0
Multilevel tabu	6.15	1
MLB tabu	2.3	10

The running times are comparable, and even slightly in favor of the most successful methods. Indeed the first two methods work full time on $WS(5)$, i.e. on length 196 data structures, while both multilevel algorithms spend some time solving $WS(4)$, thus working on smaller data.

The multilevel backtracking mechanism outperforms the other methods considered here by a considerable margin. First, both classic versions were unable to find the current best known bound for the $WS(5)$ problem. Anyhow the restart version is better, certainly benefiting from a necessary diversification. The third algorithm, multilevel only, performs slightly better as it is able to find, once, the bound to the $WS(5)$ problem and gets an average error of 6.15 (2 triplet constraint violations remain on average). The multilevel coupled with backtracking outperforms its competitors, rediscovering the best known bound 10 times out of 30 runs, and getting the lowest average error. We recall that it is known from [5] that only 2 of the 3 partitions for $WS(3)$ can be extended to construct a successful partition for $WS(4)$. This was the motivation for introducing the backtracking mechanism, and from these results it also seems that not all $WS(4)$ partitions can be extended to $WS(5)$, explaining the difference in success ratio between the two last algorithms of Table 3.

As none of the algorithms were able to improve the 196 lower bound for $WS(5)$, this gives some confidence that it may be the exact bound.

Experiments on $WS(6)$: We also used the multilevel and backtracking version to work on the $WS(6)$ problem, obtaining several weakly sum-free 6-partitions of the set $[1, 572]$, the best known bound at the time. This algorithm was also able to find a weakly sum-free 6-partition of $[1, 574]$, setting a new lower bound to the 6^{th} weak Schur number problem:

$$WS(6) \geq 574$$

Table 4 presents this partition for the $WS(6)$ problem.

4 Conclusions

This work shows that local search optimization techniques such as tabu search can be used to tackle hard Ramsey theory problems, allowing to refine bounds and to obtain instances of solutions that would be difficult, if not impossible, to obtain by other methods, such as exhaustive constraint propagation.

While heuristic approaches cannot definitely settle the value of weak Schur numbers, the solutions we obtained can suggest ideas on the structure of the problem. This study also gives some confidence that 196 is a very probable tight bound for $WS(5)$, while it also sets a new lower bound for $WS(6) \geq 574$.

In our case, the tabu search had to be enhanced by multilevel search and backtracking in order to successfully tackle this problem. Hybridizing our tabu search with some constraint propagation mechanisms could probably enhance further the power of this heuristic.

Tuning the tabu parameters by using a tool based on racing algorithms [14], has been initiated, although the computing time cost is large. First results on $WS(3)$ enhance the importance of allowing a ratio of random moves in the search process, and plead for a slightly larger minimum tabu tenure.

Table 4. Sum free 6-partition of the 574 first integers.

$E_1 =$ [1, 2, 4, 8, 11, 22, 25, 31, 40, 50, 60, 63, 69, 84,
97, 135, 140, 145, 150, 155, 164, 169, 178, 183, 193, 199,
225, 258, 273, 330, 353, 356, 395, 400, 410, 415, 438, 447,
461, 504, 519, 533, 547, 556, 561, 568, 571]

$E_2 =$ [3, 5, 6, 7, 19, 21, 23, 35, 36, 51, 52, 53, 64, 65,
66, 80, 93, 109, 122, 124, 137, 138, 139, 151, 152, 153, 165,
180, 181, 182, 194, 195, 196, 210, 212, 226, 241, 251, 255,
298, 310, 314, 325, 340, 341, 354, 355, 369, 371, 384, 397,
398, 399, 411, 412, 413, 426, 440, 441, 442, 458, 472, 473,
482, 486, 498, 500, 502, 514, 515, 529, 530, 531, 542, 543,
558, 559, 560, 572, 573, 574]

$E_3 =$ [9, 10, 12, 13, 14, 15, 16, 17, 18, 20, 54, 55,
56, 57, 58, 59, 61, 62, 101, 103, 104, 107, 141, 142, 143,
144, 146, 147, 148, 149, 184, 185, 186, 187, 188, 189, 190,
191, 192, 227, 229, 230, 232, 233, 234, 235, 269, 270, 276,
317, 319, 320, 321, 322, 359, 360, 361, 363, 364, 365, 401,
402, 403, 404, 405, 406, 407, 408, 409, 443, 444, 445, 446,
448, 449, 450, 451, 476, 477, 478, 479, 483, 484, 520, 521,
522, 523, 524, 525, 526, 527, 528, 562, 563, 564, 565, 566,
567, 569, 570]

$E_4 =$ [24, 26, 27, 28, 29, 30, 32, 33, 34, 37, 38, 39,
41, 42, 43, 44, 45, 46, 47, 48, 49, 154, 156, 157, 158, 159,
160, 161, 162, 163, 166, 167, 168, 170, 171, 172, 173, 174,
175, 176, 177, 179, 284, 286, 288, 289, 292, 294, 295, 303,
304, 305, 309, 414, 416, 417, 418, 419, 420, 421, 422, 423,
424, 425, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436,
437, 439, 532, 534, 535, 536, 537, 538, 539, 540, 541, 544,
545, 546, 548, 549, 550, 551, 552, 553, 554, 555, 557]

$E_5 =$ [67, 68, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
81, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 94, 95, 96, 98,
99, 100, 102, 105, 106, 108, 110, 111, 112, 113, 114, 115,
116, 117, 118, 119, 120, 121, 123, 125, 126, 127, 128, 129,
130, 131, 132, 133, 134, 136, 271, 274, 275, 279, 280, 282,
283, 287, 291, 296, 299, 302, 307, 308, 312, 313, 315, 452,
453, 454, 455, 456, 457, 459, 460, 462, 463, 464, 465, 466,
467, 468, 469, 470, 471, 474, 475, 480, 481, 485, 487, 488,
489, 490, 491, 492, 493, 494, 495, 496, 497, 499, 501, 503,
505, 506, 507, 508, 509, 510, 511, 512, 513, 516, 517, 518]

$E_6 =$ [197, 198, 200, 201, 202, 203, 204, 205, 206, 207,
208, 209, 211, 213, 214, 215, 216, 217, 218, 219, 220, 221,
222, 223, 224, 228, 231, 236, 237, 238, 239, 240, 242, 243,
244, 245, 246, 247, 248, 249, 250, 252, 253, 254, 256, 257,
259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 272, 277,
278, 281, 285, 290, 293, 297, 300, 301, 306, 311, 316, 318,
323, 324, 326, 327, 328, 329, 331, 332, 333, 334, 335, 336,
337, 338, 339, 342, 343, 344, 345, 346, 347, 348, 349, 350,
351, 352, 357, 358, 362, 366, 367, 368, 370, 372, 373, 374,
375, 376, 377, 378, 379, 380, 381, 382, 383, 385, 386, 387,
388, 389, 390, 391, 392, 393, 394, 396]

Note added in proof. The lower bound on $WS(6)$ has been recently improved by our colleagues from [5] and is now given by $WS(6) \geq 575$. This result was obtained by using a constraint solver and information drawn from our proposal partition in Table 4.

References

1. Issai Schur. Über die kongruenz $x^m + y^m \equiv z^m \pmod{p}$. *Jahresbericht der Deutschen Mathematiker Vereinigung*, 25:114–117, 1916.
2. R. Rado. Some solved and unsolved problems in the theory of numbers. *Math. Gaz.*, 25:72–77, 1941.
3. Peter F. Blanchard, Frank Harary, and Rogerio Reis. Partitions into sum-free sets. In *Integers 6*, volume A7, 2006.
4. G.W. Walker. A problem in partitioning. *Amer. Math. Monthly*, 59:253, 1952.
5. S. Eliahou, J. M. Marín, M. P. Revuelta, and M. I. Sanz. Weak Schur numbers and the search for G. W. Walker's lost partitions. *Computer and Mathematics with Applications*, 63:175–182, 2012.
6. H.L. Abbott and D. Hanson. A problem of Schur and its generalizations. *Acta Arith.*, 20:175–187, 1972.
7. Pierre Bornsztein. On an extension of a theorem of Schur. *Acta Arith.*, 101:395–399, 2002.
8. Fred Glover. Tabu search - part I. *ORSA Journal of Computing*, 1(3):190–206, 1989.
9. Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.
10. J.P. Hamiez, J. K. Hao, and F. W. Glover. A study of tabu search for coloring random 3-colorable graphs around the phase transition. *Int. Journal of Applied Metaheuristic Computing*, 1(4):1–24, 2010.
11. C. Walshaw. Multilevel Refinement for Combinatorial Optimisation Problems. *Annals Oper. Res.*, 131:325–372, 2004.
12. Jonathan Chappelon and Shalom Eliahou, 2010. Personal communication.
13. M. Heule and H. V. Maaren. Improved version of march ks. http://www.st.ewi.tudelft.nl/sat/Sources/stable/march_pl. Last accessed on May 2010.
14. Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The *irace* package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.