Satisfiability and computing van der Waerden numbers

Michael R. Dransfield[1], Victor W. Marek[2], and Mirosław Truszczyński[2]

[1] National Security Agency, Information Assurance Directorate, Ft. Meade, MD 20755

[2] Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

Abstract. In this paper we bring together the areas of combinatorics and propositional satisfiability. Many combinatorial theorems establish, often constructively, the existence of positive integer functions, without actually providing their closed algebraic form or tight lower and upper bounds. The area of Ramsey theory is especially rich in such results. Using the problem of computing van der Waerden numbers as an example, we show that these problems can be represented by parameterized propositional theories in such a way that decisions concerning their satisfiability determine the numbers (function) in question. We show that by using general-purpose complete and local-search techniques for testing propositional satisfiability, this approach becomes effective — competitive with specialized approaches. By following it, we were able to obtain several new results pertaining to the problem of computing van der Waerden numbers. We also note that due to their properties, especially their structural simplicity and computational hardness, propositional theories that arise in this research can be of use in development, testing and benchmarking of SAT solvers.

1 Introduction

In this paper we discuss how the areas of propositional satisfiability and combinatorics can help advance each other. On one hand, we show that recent dramatic improvements in the efficiency of SAT solvers and their extensions make it possible to obtain new results in combinatorics simply by encoding problems as propositional theories, and then computing their models (or deciding that none exist) using off-the-shelf general-purpose SAT solvers. On the other hand, we argue that combinatorics is a rich source of structured, parameterized families of hard propositional theories, and can provide useful sets of benchmarks for developing and testing new generations of SAT solvers.

In our paper we focus on the problem of computing van der Waerden numbers. The celebrated van der Waerden theorem [21] asserts that for every positive integers k and l there is a positive integer m such that every partition of $\{1, \ldots, m\}$ into k blocks (parts) has at least one block with an arithmetic progression of length l. The problem is to find the least such number m. This number is called the *van der Waerden number* $W(k,l)$. Exact values of $W(k,l)$ are known only for five pairs (k,l). For other combinations of k and l there are some general lower and upper bounds but they are very coarse and do not give any good idea about the actual value of $W(k,l)$. In the paper we show that SAT solvers such as POSIT [6], and SATO [22], as well as recently developed local-search solver *walkaspps* [13], designed to compute models for propositional theories extended by cardinality atoms [4], can improve lower bounds for van der Waerden numbers for several combinations of parameters k and l.

Theories that arise in these investigations are determined by the two parameters k and l. Therefore, they show a substantial degree of structure and similarity. Moreover, as k and l grow, these theories quickly become very hard. This hardness is only to some degree an effect of the growing size of the theories. For the most part, it is the result of the inherent difficulty of the combinatorial problem in question. All this suggests that theories resulting from hard combinatorial problems defined in terms of tuples of integers may serve as benchmark theories in experiments with SAT solvers.

There are other results similar in spirit to the van der Waerden theorem. The Schur theorem states that for every positive integer k there is an integer m such that every partition of $\{1,\ldots,m\}$ into k blocks contains a block that is not sum-free. Similarly, the Ramsey theorem (which gave name to this whole area in combinatorics) [17] concerns the existence of monochromatic cliques in edge-colored graphs, and the Hales-Jewett theorem [11] concerns the existence of monochromatic lines in colored cubes. Each of these results gives rise to a particular function defined on pairs or triples of integers and determining the values of these functions is a major challenge for combinatorialists. In all cases, only few exact values are known and lower and upper estimates are very far apart. Many of these results were obtained by means of specialized search algorithms highly depending on the combinatorial properties of the problem. Our paper shows that generic SAT solvers are maturing to the point where they are competitive and sometimes more effective than existing advanced specialized approaches.

2 van der Waerden numbers

In the paper we use the following terminology. By $\mathbb{Z}^+$ we denote the set of positive integers and, for $m \in \mathbb{Z}^+$, $[m]$ is the set $\{1,\ldots,m\}$. A *partition* of a set X is a collection $\mathcal{A}$ of nonempty and mutually disjoint subsets of X such that $\bigcup \mathcal{A} = X$. Elements of $\mathcal{A}$ are commonly called *blocks*.

Informally, the van der Waerden theorem [21] states that if a sufficiently long initial segment of positive integers is partitioned into a few blocks, then one of these blocks has to contain an arithmetic progression of a desired length. Formally, the theorem is usually stated as follows.

Theorem 1 (van der Waerden theorem). *For every $k,l \in \mathbb{Z}^+$, there is $m \in \mathbb{Z}^+$ such that for every partition $\{A_1,\ldots,A_k\}$ of $[m]$, there is i, $1 \le i \le k$, such that block A_i contains an arithmetic progression of length at least l.*

We define the *van der Waerden number* $W(k,l)$ to be the least number m for which the assertion of Theorem 1 holds. Theorem 1 states that van der Waerden numbers are well defined.

One can show that for every k and l, where $l \geq 2$, $W(k,l) > k$. In particular, it is easy to see that $W(k,2) = k+1$. From now on, we focus on the non-trivial case when $l \geq 3$.

Little is known about the numbers $W(k,l)$. In particular, no closed formula has been identified so far and only five exact values are known. They are shown in Table 1 [1, 10].

k \ l	3	4	5
2	9	35	178
3	27		
4	76		

Table 1. Known non-trivial values of van der Waerden numbers

Since we know few exact values for van der Waerden numbers, it is important to establish good estimates. One can show that the Hales-Jewett theorem entails the van der Waerden theorem, and some upper bounds for the numbers $W(k,l)$ can be derived from the Shelah's proof of the former [19]. Recently, Gowers [9] presented stronger upper bounds, which he derived from his proof of the Szemerédi theorem [20] on arithmetic progressions.

In our work, we focus on lower bounds. Several general results are known. For instance, Erdös and Rado [5] provided a non-constructive proof for the inequality

$$W(k,l) > (2(l-1)k^{l-1})^{1/2}.$$

For some special values of parameters k and l, Berlekamp obtained better bounds by using properties of finite fields [2]. These bounds are still rather weak. His strongest result concerns the case when $k = 2$ and $l-1$ is a prime number. Namely, he proved that when $l-1$ is a prime number,

$$W(2,l) > (l-1)2^{l-1}.$$

In particular, $W(2,6) > 160$ and $W(2,8) > 896$.

Our goal in this paper is to employ propositional satisfiability solvers to find lower bounds for several small van der Waerden numbers. The bounds we find significantly improve on the ones implied by the results of Erdös and Rado, and Berlekamp.

We proceed as follows. For each triple of positive integers $\langle k,l,m\rangle$, we define a propositional CNF theory $\mathrm{vdW}_{k,l,m}$ and then show that $\mathrm{vdW}_{k,l,m}$ is satisfiable if and only if $W(k,l) > m$. With such encodings, one can use SAT solvers (at least in principle) to determine the satisfiability of $\mathrm{vdW}_{k,l,m}$ and, consequently, find $W(k,l)$. Since $W(k,l) > k$, without loss of generality we can restrict our attention to $m > k$. We also show that more concise encodings are possible, leading ultimately to better bounds, if we use an extension of propositional logic by *cardinality atoms* and apply to them solvers capable of handling such atoms directly.

To describe $\mathrm{vdW}_{k,l,m}$ we will use a standard first-order language, without function symbols, but containing a predicate symbol *in_block* and constants $1,\ldots,m$. An intuitive reading of a ground atom $\mathit{in\_block}(i,b)$ is that an integer i is in block b.

We now define the theory $\mathrm{vdW}_{k,l,m}$ by including in it the following clauses:

vdW1: $\neg \mathit{in\_block}(i,b_1) \vee \neg \mathit{in\_block}(i,b_2)$, for every $i \in [m]$ and every $b_1, b_2 \in [k]$ such that $b_1 < b_2$,

vdW2: $\mathit{in\_block}(i,1) \vee \ldots \vee \mathit{in\_block}(i,k)$, for every $i \in [m]$,

vdW3: $\neg \mathit{in\_block}(i,b) \vee \neg \mathit{in\_block}(i+d,b) \vee \ldots \vee \neg \mathit{in\_block}(i+(l-1)d,b)$, for every $i,d \in [m]$ such that $i+(l-1)d \leq m$, and for every b such that $1 \leq b \leq k$.

As an aside, we note that we could design $\mathrm{vdW}_{k,l,m}$ strictly as a theory in propositional language using propositional atoms of the form $\mathit{in\_block}_{i,b}$ instead of ground atoms $\mathit{in\_block}(i,b)$. However, our approach opens a possibility to specify this theory as finite (and independent of data) collections of *propositional schemata*, that is, open clauses in the language of first-order logic without function symbols. Given a set of appropriate constants (to denote integers and blocks) such theory, after grounding, coincides with $\mathrm{vdW}_{k,l,m}$. In fact, we have defined an appropriate syntax that allows us to specify both data and schemata and implemented a grounding program *psgrnd* [4] that generates their equivalent ground (propositional) representation. This grounder accepts arithmetic expressions as well as simple regular expressions, and evaluates and eliminates them according to their standard interpretation. Such approach significantly simplifies the task of developing propositional theories that encode problems, as well as the use of SAT solvers [4].

Propositional interpretations of the theory $\mathrm{vdW}_{k,l,m}$ can be identified with subsets of the set of atoms $\{\mathit{in\_block}(i,b)\colon i \in [m],\ b \in [k]\}$. Namely, a set $M \subseteq \{\mathit{in\_block}(i,b)\colon i \in [m],\ b \in [k]\}$ determines an interpretation in which all atoms in M are true and all other atoms are false. In the paper we always assume that interpretations are represented as sets.

It is easy to see that clauses (vdW1) ensure that if M is a model of $\mathrm{vdW}_{k,l,m}$ (that is, is an interpretation satisfying all clauses of $\mathrm{vdW}_{k,l,m}$), then for every $i \in [m]$, M contains at most one atom of the form $\mathit{in\_block}(i,b)$. Clauses (vdW2) ensure that for every $i \in [m]$ there is at least one $b \in [k]$ such that $\mathit{in\_block}(i,b) \in M$. In other words, clauses (vdW1) and (vdW2) together ensure that if M is a model of $\mathrm{vdW}_{k,l,m}$, then M determines a partition of $[m]$ into k blocks.

The last group of constraints, clauses (vdW3), guarantee that elements from $[m]$ forming an arithmetic progression of length l do not all belong to the same block. All these observations imply the following result.

Proposition 1. *There is a one-to-one correspondence between models of the formula $vdW_{k,l,m}$ and partitions of $[m]$ into k blocks so that no block contains an arithmetic progression of length l. Specifically, an interpretation M is a model of $vdW_{k,l,m}$ if and only if $\{\{i \in [m] \colon in\_block(i, b) \in M\} \colon b \in [k]\}$ is a partition of $[m]$ into k blocks such that no block contains an arithmetic progression of length l.*

Proposition 1 has the following direct corollary.

Corollary 1. *For every positive integers k, l, and m, with $l \geq 2$ and $m > k$, $m < W(k, l)$ if and only if the formula $vdW_{k,l,m}$ is satisfiable.*

It is evident that if m has the property that $\mathrm{vdW}_{k,l,m}$ is unsatisfiable then for every $m' > m$, $\mathrm{vdW}_{k,l,m'}$ is also unsatisfiable. Thus, Corollary 1 suggests the following algorithm that, given k and l, computes the van der Waerden number $W(k, l)$: for consecutive integers $m = k+1, k+2, \ldots$ we test whether the theory $\mathrm{vdW}_{k,l,m}$ is satisfiable. If so, we continue. If not, we return m and terminate the algorithm. By the van der Waerden theorem, this algorithm terminates.

It is also clear that there are simple symmetries involved in the van der Waerden problem. If a set M of atoms of the form $in\_block(i, b)$ is a model of the theory $\mathrm{vdW}_{k,l,m}$, and π is a permutation of $[k]$, then the corresponding set of atoms $\{in\_block(i, \pi(b)) \colon in\_block(i, b) \in M\}$ is also a model of $\mathrm{vdW}_{k,l,m}$, and so is the set of atoms $\{in\_block(m + 1 - i, b) \colon in\_block(i, b) \in M\}$.

Following the approach outlined above, adding clauses to break these symmetries, and applying POSIT [6] and SATO [22] as a SAT solvers we were able to establish that $W(4, 3) = 76$ and compute a "library" of counterexamples (partitions with no block containing arithmetic progressions of a specified length) for $m = 75$. We were also able to find several lower bounds on van der Waerden numbers for larger values of k and m.

However, a major limitation of our first approach is that the size of theories $\mathrm{vdW}_{k,l,m}$ grows quickly and makes complete SAT solvers ineffective. Let us estimate the size of the theory $\mathrm{vdW}_{k,l,m}$. The total size of clauses (vdW1) (measured as the number of atom occurrences) is $\Theta(mk^2)$. The size of clauses (vdW2) is $\Theta(mk)$. Finally, the size of clauses (vdW3) is $\Theta(m^2)$ (indeed, there are $\Theta(m^2/l)$ arithmetic progressions of length l in $[m]$)[1]. Thus, the total size of the theory $\mathrm{vdW}_{k,l,m}$ is $\Theta(mk^2 + m^2)$.

To overcome this obstacle, we used a two-pronged approach. First, as a modeling language we used PS+ logic [4], which is an extension of propositional logic by cardinality atoms. Cardinality atoms support concise representations of constraints of the form "at least p and at most r elements in a set are true" and result in theories of smaller size. Second, we used a local-search algorithm, *walkaspps*, for finding models of theories in logic PS+ that we have designed and

[1] Goldstein [8] provided a precise formula. When $r = rm(m-1, l-1)$ and $q = q(m-1, l-1)$ then there are $q \cdot r + \binom{q-1}{2} \cdot (l-1)$ arithmetic progressions of length l in $[m]$.

implemented recently [13]. Using encodings as theories in logic PS+ and *walkaspps* as a solver, we were able to obtain substantially stronger lower bounds for van der Waerden numbers than those know to date.

We will now describe this alternative approach. For a detailed treatment of the PS+ logic we refer the reader to [4]. In this paper, we will only review most basic ideas underlying the logic PS+ (in its propositional form). By a *propositional cardinality atom* (*c-atom* for short), we mean any expression of the form $m\{p_1,\ldots,p_k\}n$ (one of m and n, but not both, may be missing), where m and n are non-negative integers and $p_1,\ldots,p_k$ are propositional atoms from At. The notion of a clause generalizes in an obvious way to the language with cardinality atoms. Namely, a *c-clause* is an expression of the form

$$C = \quad A_1 \vee \ldots \vee A_s \vee \neg B_1 \vee \ldots \vee \neg B_t, \tag{1}$$

where all A_i and B_i are (propositional) atoms or cardinality atoms.

Let $M \subseteq At$ be a set of atoms. We say that M *satisfies* a cardinality atom $m\{p_1,\ldots,p_k\}n$ if

$$m \leq |M \cap \{p_1,\ldots,p_k\}| \leq n.$$

If m is missing, we only require that $|M \cap \{p_1,\ldots,p_k\}| \leq n$. Similarly, when n is missing, we only require that $m \leq |M \cap \{p_1,\ldots,p_k\}|$. A set of atoms M *satisfies* a c-clause C of the form (1) if M satisfies at least one atom A_i or does not satisfy at least one atom B_j. W note that the expression $1\{p_1,\ldots,p_k\}1$ expresses the quantifier "There exists exactly one ..." - commonly used in mathematical statements.

It is now clear that all clauses (vdW1) and (vdW2) from $\mathrm{vdW}_{k,l,m}$ can be represented in a more concise way by the following collection of c-clauses:

vdW′1: $1\{in\_block(i,1),\ldots,in\_block(i,k)\}1$, for every $i \in [m]$.

Indeed, c-clauses (vdW′1) enforce that their models, for every $i \in [m]$ contain exactly one atom of the form $in\_block(i,b)$ — precisely the same effect as that of clauses (vdW1) and (vdW2). Let $\mathrm{vdW}'_{k,l,m}$ be a PS+ theory consisting of clauses (vdW′1) and (vdW3). It follows that Proposition 1 and Corollary 1 can be reformulated by replacing $\mathrm{vdW}_{k,l,m}$ with $\mathrm{vdW}'_{k,l,m}$ in their statements. Consequently, any algorithm for finding models of PS+ theories can be used to compute van der Waerden numbers (or, at least, some bounds for them) in the way we described above.

The adoption of cardinality atoms leads to a more concise representation of the problem. While, as we discussed above, the size of all clauses (vdW1) and (vdW2) is $\Theta(mk^2 + mk)$, the size of clauses (vdW′1) is $\Theta(mk)$.

In our experiments, for various lower bound results, we used the local-search algorithm *walkaspps* [13]. This algorithm is based on the same ideas as *walksat* [18]. A major difference is that due to the presence of c-atoms in c-clauses *walkaspps* uses different formulas to calculate the breakcount and proposes several other heuristics designed specifically to handle c-atoms.

3 Bootstrapping for *walkaspps*

Since *walkaspps* is an incomplete solver, it cannot guarantee that it can find a solution when there is one. The likelihood that a try terminates with the success depends on the proximity of the initial truth assignment (ITA, for short) used in the try to a satisfying truth assignment. That is true for *walkaspps* and, in fact, for most $WSAT$-like local-search SAT solvers. It is a non-trivial problem to generate "good" ITA's. In [14], we proposed and implemented a *bootstrapping* technique to address it. We call a theory T' a *relaxation* of a theory T if for every model M of T, $M \cap At(T')$ is a model of M'. The bootstrapping consists of using satisfying assignments for a relaxation T' of a theory T as ITAs in tries when using a local-search solver (*walkaspps*,in our case) to find satisfying assignments for T. The underlying intuition is that a relaxation of a theory is easier to solve than the theory itself and that solutions to T' are more likely to be close to solutions to T than random assignments.

For a van der Waerden number instance $\mathrm{vdW}_{k,l,m}$, the theory that corresponds to the instance $\mathrm{vdW}_{k,l,m'}$ with $m' < m$ is a relaxation of the original instance $\mathrm{vdW}_{k,l,m}$. Indeed, we are looking for counterexamples. If a counterexample (partition without the arithmetic progression of length k in any of its blocks) exists of a given m, then any restriction of such partition to $m' < m$ is a counterexample at m'.

In order to improve the lower bound on an van der Waerden instance $\mathrm{vdW}_{k,l,m}$, we construct a sequence of relaxations: vdW_{k,l,m_1}, ..., vdW_{k,l,m_k}, where $m_1 < \ldots < m_k = 3Dm$. Given a sequence of relaxation, the bootstrapping algorithm works as follows:

1. It starts at level 1 and uses *walkaspps* with randomly generated ITAs to solve the first theory in the sequence, vdW_{k,l,m_1}
2. Each time a solution S for a theory vdW_{k,l,m_i} is found, the algorithm moves to the next theory in the sequence, $\mathrm{vdW}_{k,l,m_{i+1}}$, and runs *walkaspps* on $\mathrm{vdW}_{k,l,m_{i+1}}$ with S as an ITA;
3. It restarts computation from level 1 if *walkaspps* fails to find any solutions at some level i;
4. It stops and outputs the solution if it succeeds at level k, which means it solves the target van der Waerden instance.

4 Results

Our goal is to establish lower bounds for small van der Waerden numbers by exploiting propositional satisfiability solvers. Here is a summary of our results.

1. Using complete SAT solvers POSIT and SATO and the encoding of the problem as $\mathrm{vdW}_{k,l,m}$, we found a "library" of all (up to obvious symmetries) counterexamples to the fact that $W(4,3) > 75$. There are 30 of them. We list two of them in the appendix. A complete list can be found at http://www.cs.uky.edu/ai/vdw/. Since there are 48 symmetries, of the

types discussed above, the full library of counterexamples consists of 1440 partitions.

2. We found that the formula $\mathrm{vdW}_{4,3,76}$ is unsatisfiable. Hence, we found that a “generic” SAT solver is capable of finding that $W(4,3) = 76$.
3. We established several new lower bounds for the numbers $W(k,l)$. They are presented in Table 4. Partitions demonstrating that $W(2,8) > 1295$, $W(3,5) > 650$, and $W(4,4) > 408$ are included in the appendix. Counterexample partitions for all other inequalities are available at http://www.cs.uky.edu/ai/vdw/. We note that our bounds for $W(2,6)$ and $W(2,8)$ are much stronger than those implied by the results of Berlekamp [2], which we stated earlier.

Table 2. Extended results on van der Waerden numbers

l / k	3	4	5	6	7	8
2	9	35	178	> 341	> 614	> 1316
3	27	> 193	> 671	> 2236		
4	76	> 416				
5	> 125	> 880				
6	> 194					

To provide some insight into the complexity of the satisfiability problems involved, in Table 4 we list the number of atoms and the number of clauses in the theories $\mathrm{vdW}'_{k,l,m}$. Specifically, the entry k, l in this table contains the number of atoms and the number of clauses in the theories $\mathrm{vdW}'_{k,l,m}$, where m is the value given in the entry k, l in Table 4.

Table 3. Numbers of atoms and clauses in theories $\mathrm{vdW}'_{k,l,m}$, used to establish the results presented in Table 4.

l / k	3	4	5	6	7	8
2	18, 41	70, 409	356, 7922	682, 23257	1208, 60804	2590, 239575
3	108, 534	579, 18529	1950, 158114	????		
4	304, 5700	1632, 110568				
5	625, 19345	????				
6	1080, 48240					

5 Discussion

Recent progress in the development of SAT solvers provides an important tool for researchers looking for both the existence and non-existence of various combinatorial objects. We have demonstrated that several classical questions related to van der Waerden numbers can be naturally cast as questions on the existence of satisfying valuations for some propositional CNF-formulas.

Computing combinatorial objects such as van der Waerden numbers is hard. They are structured but as we pointed out few values are known, and new results are hard to obtain. Thus, the computation of those numbers can serve as a benchmark ('can we find the configuration such that...') for complete and local-search methods, and as a challenge ('can we show that a configuration such that ...' does not exist) for complete SAT solvers. Moreover, with powerful SAT solvers it is likely that the bounds obtained by computation of counterexamples are "sharp" in the sense that when a configuration is not found then none exist. For instance it is likely that $W(5,3)$ is close to 126 (possibly, it is 126), because 125 was the last integer where we were able to find a counterexample despite significant computational effort. This claim is further supported by the fact that in all examples where exact values are known, our local-search algorithm was able to find counterexample partitions for the last possible value of m. The lower-bounds results of this sort may constitute an important clue for researchers looking for nonexistence arguments and, ultimately, for the closed form of van der Waerden numbers.

A major impetus for the recent progress of SAT solvers comes from applications in computer engineering. In fact, several leading SAT solvers such as zCHAFF [16] and *berkmin* [7] have been developed with the express goal of aiding engineers in correctly designing and implementing digital circuits. Yet, the fact that these solvers are able to deal with hard optimization problems in one area (hardware design and verification) carries the promise that they will be of use in another area — combinatorial optimization. Our results indicate that it is likely to be the case.

The current capabilities of SAT solvers has allowed us to handle large instances of these problems. Better heuristics and other techniques for pruning the search space will undoubtedly further expand the scope of applicability of generic SAT solvers to problems that, until recently, could only be solved using specialized software.

Acknowledgments

The authors thank Lengning Liu for developing software facilitating our experimental work. This research has been supported by the Center for Communication Research, La Jolla. During the research reported in this paper the second and third authors have been partially supported by an NSF grant IIS-0097278.

References

1. M.D. Beeler and P.E. O'Neil. Some new van der Waerden numbers, *Discrete Mathematics*, 28:135–146, 1979.
2. E. Berlekamp. A construction for partitions which avoid long arithmetic progressions. *Canadian Mathematical Bulletin* 11:409–414, 1968.
3. M. Davis and H. Putnam. A computing procedure for quantification theory, *Journal of the Association for Computing Machinery*, 7:201–215, 1960.
4. D. East and M. Truszczyński. Propositional satisfiability in answer-set programming. Proceedings of Joint German/Austrian Conference on Artificial Intelligence, KI'2001. Lecture Notes in Artificial Intelligence, Springer Verlag 2174, pages 138–153. (Full version available at http://xxx.lanl.gov/ps/cs.LO/0211033). 2001.
5. P. Erdös and R. Rado. Combinatorial theorems on classifications of subsets of a given set, *Proceedings of London Mathematical Society*, 2:417–439, 1952.
6. J.W. Freeman. *Improvements to propositional satisfiability search algorithms*, PhD thesis, Department of Computer Science, University of Pennsylvania, 1995.
7. E. Goldberg, Y. Novikov. BerkMin: a Fast and Robust SAT-Solver. DATE-2002, pages 142–149, 2002.
8. D. Goldstein. Personal communication, 2002.
9. T. Gowers. A new proof of Szemerédi theorem. *Geometric and Functional Analysis*, 11:465-588, 2001.
10. R.L. Graham, B.L. Rotschild, and J.H. Spencer. *Ramsey Theory*, Wiley, 1990.
11. A. Hales and R.I. Jewett. Regularity and positional games, *Transactions of American Mathematical Society*, 106:222–229, 1963.
12. R.E. Jeroslaw and J. Wang. solving propositional satisfiability problems, *Annals of Mathematics and Artificial Intelligence*, 1:167–187, 1990.
13. L. Liu and M. Truszczyński. Local-search techniques in propositional logic extended with cardinality atoms. In preparation.
14. L. Liu and M. Truszczyński. Local-search with bootstrapping. In *Proceedings of SAT 2004*, Vancouver, Canada, 2004.
15. J.P. Marques-Silva and K.A. Sakallah. GRASP: A new search algorithm for satisfiability, *IEEE Transactions on Computers*, 48:506–521, 1999.
16. M.W. Moskewicz, C.F. Magidan, Y. Zhao, L. Zhang, and S. Malik. Chaff: engineering an efficient SAT solver, in *SAT 2001*, 2001.
17. F.P. Ramsey. On a problem of formal logic, *Proceedings of London Mathematical Society*, 30:264–286, 1928.
18. B. Selman, H.A. Kautz, and B. Cohen. Noise Strategies for Improving Local Search. *Proceedings of AAAI'94*, pp. 337-343. MIT Press 1994.
19. S. Shelah. Primitive recursive bounds for van der Waerden numbers, *Journal of American Mathematical Society*, 1:683–697, 1988.
20. E. Szemerédi. On sets of integers containing no k elements in arithmetic progression, *Acta Arithmetica*, 27:199–243, 1975.
21. B.L. van der Waerden. Beweis einer Baudetschen Vermutung, *Nieuwe Archief voor Wiskunde*, 15:212–216, 1927.
22. H. Zhang. SATO: An efficient propositional prover, in *Proceedings of CADE-17*, pages 308–312, 1997. Springer Lecture Notes in Artificial Intelligence 1104.

Appendix

Using a complete SAT solver we computed the library of all partitions (up to isomorphism) of [75] showing that $75 < W(4, 3)$. Two of these 30 partitions are shown below:

Solution 1:
Block 1: 6 7 9 14 18 20 23 24 36 38 43 44 46 51 55 57 60 61 73 75
Block 2: 4 5 12 22 26 28 29 31 37 41 42 49 59 63 65 66 68 74
Block 3: 1 2 8 10 11 13 17 27 34 35 39 45 47 48 50 54 64 71 72
Block 4: 3 15 16 19 21 25 30 32 33 40 52 53 56 58 62 67 69 70

Solution 2:
Block 1: 6 7 9 14 18 20 23 24 36 38 43 44 46 51 55 57 60 61 73
Block 2: 4 5 12 22 26 28 29 31 37 41 42 49 59 63 65 66 68 74
Block 3: 1 2 8 10 11 13 17 27 34 35 39 45 47 48 50 54 64 71 72
Block 4: 3 15 16 19 21 25 30 32 33 40 52 53 56 58 62 67 69 70 75

These two and the remaining 28 partitions can be found at http://www.cs.uky.edu/ai/vdw/

Next, we exhibit a partition of [1316] into two blocks demonstrating that $W(2,8)$ > 1316.

Block 1:
1 2 3 8 10 12 13 14 15 16 18 26 28 29 30 32 34 38 39 40 42 43 44 45 50 52 54 55 57 59
60 66 69 70 71 72 74 75 76 79 81 83 87 88 91 95 100 101 103 104 106 107 108 109 110
112 114 117 118 121 122 126 127 128 131 133 134 136 138 143 144 145 146 147 149 150
151 152 157 161 162 163 165 167 168 169 172 175 176 183 186 189 193 194 195 198 199
201 205 207 208 209 211 212 214 215 217 218 219 220 222 227 233 234 235 236 239 240
242 244 245 247 248 249 250 252 254 255 260 266 267 269 270 273 274 275 282 283 284
285 286 287 290 291 296 297 300 301 303 305 307 308 309 310 312 313 314 315 317 318
319 321 322 324 328 331 333 334 338 339 343 344 349 350 351 352 353 354 355 357 360
361 369 371 373 374 376 378 379 385 387 388 389 390 391 392 396 397 399 400 401 402
403 404 406 413 416 417 419 424 428 430 432 433 435 436 437 438 440 446 447 452 454
455 456 458 460 461 468 470 475 476 480 486 488 489 491 492 493 497 498 500 504 508
509 511 512 513 514 515 517 518 519 521 525 527 528 529 530 533 534 536 538 539 540
542 546 547 548 550 551 552 554 555 556 557 560 561 563 568 569 571 572 573 577 578
582 585 592 594 595 596 599 600 601 602 603 604 605 607 608 609 610 611 615 616 617
618 620 621 622 623 626 627 628 633 635 637 640 641 642 644 646 651 652 653 658 659
660 661 664 669 670 673 679 682 683 685 686 688 689 690 692 693 694 695 697 702 703
706 707 709 711 712 713 714 717 718 719 723 725 727 731 733 734 737 741 744 745 747
748 749 750 751 753 754 755 759 761 762 766 770 771 775 776 777 778 781 783 784 785
786 787 788 789 792 793 795 796 797 798 800 801 802 806 808 812 813 814 815 820 821
822 824 826 827 832 834 841 843 846 847 848 849 851 852 856 860 863 864 867 869 870
871 878 879 881 882 888 890 891 894 895 896 900 901 903 904 906 909 911 913 915 916
917 919 922 924 926 927 928 930 931 933 934 936 937 940 941 944 945 946 947 948 949
955 956 960 964 965 967 968 969 970 972 973 974 975 976 978 979 984 985 986 987 989
990 991 992 993 994 996 997 998 1000 1003 1004 1007 1010 1015 1016 1017 1018 1019
1023 1025 1032 1034 1038 1040 1041 1042 1046 1051 1052 1056 1058 1059 1060 1061
1063 1064 1069 1071 1075 1076 1077 1079 1080 1081 1082 1083 1085 1089 1092 1093
1094 1095 1097 1098 1099 1100 1102 1103 1105 1108 1109 1111 1115 1119 1121 1122
1125 1130 1133 1134 1135 1136 1140 1143 1150 1151 1152 1153 1154 1155 1157 1160
1161 1162 1163 1166 1167 1168 1170 1171 1174 1177 1181 1182 1183 1184 1187 1188
1189 1190 1192 1194 1196 1200 1202 1205 1206 1209 1210 1216 1218 1219 1224 1225

1226 1227 1230 1233 1234 1235 1236 1239 1242 1243 1247 1249 1251 1252 1253 1255 1262 1264 1266 1267 1268 1269 1271 1273 1277 1279 1280 1281 1282 1285 1286 1290 1291 1293 1294 1295 1297 1298 1302 1304 1305 1308 1310 1312 1313 1316

Block 2:
4 5 6 7 9 11 17 19 20 21 22 23 24 25 27 31 33 35 36 37 41 46 47 48 49 51 53 56 58 61 62 63 64 65 67 68 73 77 78 80 82 84 85 86 89 90 92 93 94 96 97 98 99 102 105 111 113 115 116 119 120 123 124 125 129 130 132 135 137 139 140 141 142 148 153 154 155 156 158 159 160 164 166 170 171 173 174 177 178 179 180 181 182 184 185 187 188 190 191 192 196 197 200 202 203 204 206 210 213 216 221 223 224 225 226 228 229 230 231 232 237 238 241 243 246 251 253 256 257 258 259 261 262 263 264 265 268 271 272 276 277 278 279 280 281 288 289 292 293 294 295 298 299 302 304 306 311 316 320 323 325 326 327 329 330 332 335 336 337 340 341 342 345 346 347 348 356 358 359 362 363 364 365 366 367 368 370 372 375 377 380 381 382 383 384 386 393 394 395 398 405 407 408 409 410 411 412 414 415 418 420 421 422 423 425 426 427 429 431 434 439 441 442 443 444 445 448 449 450 451 453 457 459 462 463 464 465 466 467 469 471 472 473 474 477 478 479 481 482 483 484 485 487 490 494 495 496 499 501 502 503 505 506 507 510 516 520 522 523 524 526 531 532 535 537 541 543 544 545 549 553 558 559 562 564 565 566 567 570 574 575 576 579 580 581 583 584 586 587 588 589 590 591 593 597 598 606 612 613 614 619 624 625 629 630 631 632 634 636 638 639 643 645 647 648 649 650 654 655 656 657 662 663 665 666 667 668 671 672 674 675 676 677 678 680 681 684 687 691 696 698 699 700 701 704 705 708 710 715 716 720 721 722 724 726 728 729 730 732 735 736 738 739 740 742 743 746 752 756 757 758 760 763 764 765 767 768 769 772 773 774 779 780 782 790 791 794 799 803 804 805 807 809 810 811 816 817 818 819 823 825 828 829 830 831 833 835 836 837 838 839 840 842 844 845 850 853 854 855 857 858 859 861 862 865 866 868 872 873 874 875 876 877 880 883 884 885 886 887 889 892 893 897 898 899 902 905 907 908 910 912 914 918 920 921 923 925 929 932 935 938 939 942 943 950 951 952 953 954 957 958 959 961 962 963 966 971 977 980 981 982 983 988 995 999 1001 1002 1005 1006 1008 1009 1011 1012 1013 1014 1020 1021 1022 1024 1026 1027 1028 1029 1030 1031 1033 1035 1036 1037 1039 1043 1044 1045 1047 1048 1049 1050 1053 1054 1055 1057 1062 1065 1066 1067 1068 1070 1072 1073 1074 1078 1084 1086 1087 1088 1090 1091 1096 1101 1104 1106 1107 1110 1112 1113 1114 1116 1117 1118 1120 1123 1124 1126 1127 1128 1129 1131 1132 1137 1138 1139 1141 1142 1144 1145 1146 1147 1148 1149 1156 1158 1159 1164 1165 1169 1172 1173 1175 1176 1178 1179 1180 1185 1186 1191 1193 1195 1197 1198 1199 1201 1203 1204 1207 1208 1211 1212 1213 1214 1215 1217 1220 1221 1222 1223 1228 1229 1231 1232 1237 1238 1240 1241 1244 1245 1246 1248 1250 1254 1256 1257 1258 1259 1260 1261 1263 1265 1270 1272 1274 1275 1276 1278 1283 1284 1287 1288 1289 1292 1296 1299 1300 1301 1303 1306 1307 1309 1311 1314 1315

Next, we exhibit a partition of [671] into three blocks demonstrating that $W(3,5) > 671$.

Block 1:
2 3 7 12 13 14 18 20 26 27 33 34 36 37 44 50 51 54 57 58 59 62 73 75 77 81 82 83 84 86 87 93 104 111 114 117 118 119 120 127 128 130 133 134 137 139 144 148 156 161 162 164 167 168 169 175 177 178 182 187 192 199 201 202 204 206 207 209 210 211 217 219 225 229 231 232 234 236 238 242 252 253 257 262 263 274 279 281 283 287 289 291 292 293 296 299 301 302 303 304 307 308 311 312 313 314 316 318 319 324 327 329 331 333 336 338 342 356 359 361 366 368 370 371 372 377 382 388 393 400 402 403 406 407 412

413 417 421 425 427 428 436 438 449 450 451 452 458 461 467 468 469 476 478 479 482 486 487 488 489 492 494 497 499 503 504 508 511 514 515 517 518 526 527 528 541 542 543 546 547 553 556 557 558 559 561 562 566 572 580 584 588 591 592 597 599 600 603 604 612 614 619 623 624 626 627 629 630 633 637 638 639 641 643 646 651 653 657 659 660 661 662 664 668 671

Block 2:
4 5 11 15 17 21 24 25 29 30 32 42 47 52 55 60 61 66 67 68 72 74 79 85 90 94 95 96 97 99 102 103 106 107 108 110 112 113 115 125 129 132 135 136 141 142 143 147 150 152 154 155 158 171 172 173 174 176 184 189 190 193 194 197 200 213 216 221 222 223 226 227 228 230 233 237 245 250 251 254 259 260 261 264 267 271 272 273 275 276 277 280 282 288 297 298 300 310 315 317 322 325 326 332 335 337 339 341 344 347 348 351 352 360 362 364 365 367 369 374 375 381 383 386 394 395 397 405 409 410 411 415 420 422 426 429 432 434 435 437 439 440 442 445 446 447 457 459 462 463 464 466 470 472 473 474 475 477 480 481 490 496 502 510 512 516 519 524 525 529 531 535 536 537 539 550 555 560 563 564 567 568 570 571 577 578 582 585 586 587 590 598 601 605 606 607 611 613 617 618 620 621 622 625 628 631 632 636 640 642 645 647 650 654 655 656 667 670

Block 3:
1 6 8 9 10 16 19 22 23 28 31 35 38 39 40 41 43 45 46 48 49 53 56 63 64 65 69 70 71 76 78 80 88 89 91 92 98 100 101 105 109 116 121 122 123 124 126 131 138 140 145 146 149 151 153 157 159 160 163 165 166 170 179 180 181 183 185 186 188 191 195 196 198 203 205 208 212 214 215 218 220 224 235 239 240 241 243 244 246 247 248 249 255 256 258 265 266 268 269 270 278 284 285 286 290 294 295 305 306 309 320 321 323 328 330 334 340 343 345 346 349 350 353 354 355 357 358 363 373 376 378 379 380 384 385 387 389 390 391 392 396 398 399 401 404 408 414 416 418 419 423 424 430 431 433 441 443 444 448 453 454 455 456 460 465 471 483 484 485 491 493 495 498 500 501 505 506 507 509 513 520 521 522 523 530 532 533 534 538 540 544 545 548 549 551 552 554 565 569 573 574 575 576 579 581 583 589 593 594 595 596 602 608 609 610 615 616 634 635 644 648 649 652 658 663 665 666 669

Finally, we exhibit a partition of [416] into four blocks demonstrating that $W(4,4) > 416$.

Block 1:
2 7 11 16 17 21 24 29 30 32 39 41 42 50 51 57 64 67 68 69 76 78 80 88 91 93 96 110 122 124 130 132 133 134 137 142 148 155 157 159 160 164 165 166 169 172 176 181 182 183 185 194 195 202 204 209 212 213 219 243 246 247 248 253 254 255 257 260 264 270 272 276 277 278 280 281 286 289 293 303 304 309 310 312 313 317 322 330 336 341 345 347 350 359 361 375 381 383 384 385 394 398 399 400 403 404 406 410 411

Block 2:
3 4 8 13 14 20 28 31 35 40 44 45 52 59 61 71 79 82 83 85 89 92 97 98 100 101 106 109 117 120 127 128 135 140 141 144 146 147 152 154 156 163 168 177 179 189 193 203 208 216 217 222 224 233 235 236 244 249 251 256 258 267 268 273 274 275 279 282 284 287 294 295 297 298 300 301 305 307 324 326 331 333 338 339 340 348 349 353 356 360 362 365 368 369 370 376 386 387 396 402 408

Block 3:
6 15 18 19 22 23 43 46 47 49 54 55 56 60 62 63 65 66 73 75 77 81 84 87 102 104 107

111 112 113 115 116 125 126 129 136 138 143 158 162 178 180 187 190 191 192 197 201 206 207 210 211 218 223 225 226 228 229 237 238 241 242 245 250 252 261 263 265 266 269 271 291 306 308 311 315 318 319 321 327 343 344 352 354 355 357 358 363 374 377 378 379 382 388 389 390 392 395 405 407 409 412 414 415 416

Block 4:
1 5 9 10 12 25 26 27 33 34 36 37 38 48 53 58 70 72 74 86 90 94 95 99 103 105 108 114 118 119 121 123 131 139 145 149 150 151 153 161 167 170 171 173 174 175 184 186 188 196 198 199 200 205 214 215 220 221 227 230 231 232 234 239 240 259 262 283 285 288 290 292 296 299 302 314 316 320 323 325 328 329 332 334 335 337 342 346 351 364 366 367 371 372 373 380 391 393 397 401 413

Configurations showing the validity of other lower bounds listed in Table 4 are available at http://www.cs.uky.edu/ai/vdw/.