Satisfiability and computing van der Waerden numbers*

Michael R. Dransfield
National Security Agency
Information Assurance Directorate
Ft. Meade, MD 20755

Lengning Liu
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046

Victor W. Marek
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046

Mirosław Truszczyński
Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046

Submitted: Oct 23, 2002; Accepted: Jun 10, 2004; Published: Jun 16, 2004
MR Subject Classifications: 05D10

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

*This is an expanded and updated version of the conference paper [4].

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 [22] 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 [7], and SATO [23], as well as recently developed local-search solver *wsat(cc)* [14], designed to compute models for propositional theories extended by cardinality atoms [5], 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) [18] concerns the existence of monochromatic cliques in edge-colored graphs, and the Hales-Jewett theorem [12] 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 (although, we know of no such computational approaches in the context of the van der Waerden numbers). Our paper shows that generic SAT solvers are maturing to the point where they are capable of generating new results for hard combinatorial problems. That combined together with the ease with which they can be used in experimentation (almost no code development time) makes them a useful tool in combinatorial research.

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

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

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 *sequence* $\mathcal{A} = \langle A_1, \ldots, A_k \rangle$ of mutually disjoint subsets of X such that $\bigcup \mathcal{A} = X$. Elements of $\mathcal{A}$ are commonly called *blocks*. We note that we deviate here from the standard definition of a partition of a set X as a collection of nonempty and mutually disjoint subsets covering all elements of X. From the perspective of van der Waerden numbers, both definitions are equivalent. However, the definition we use here is better aligned with propositional theories we develop later in the paper to model the problem of computing van der Waerden numbers.

Informally, the van der Waerden theorem [22] 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 2.1 (van der Waerden theorem) *For every $k, l \in \mathbb{Z}^+$, there exists $m \in \mathbb{Z}^+$ such that for every partition $\langle A_1, \ldots, A_k \rangle$ of $[m]$, there is i, $1 \leq i \leq k$, 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 2.1 holds. Theorem 2.1 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, 11].

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 [20]. Recently, Gowers [10] presented stronger upper bounds, which he derived from his proof of the Szemerédi theorem [21] on arithmetic progressions.

In our work, we focus on lower bounds. Several general results are known. For instance, Erdös and Rado [6] 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 $1,\dots,m$. 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 $i+(l-1)d \le m$, and for every b such that $1\le b\le k$.

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* [5] that generates their equivalent ground (propositional) representation. This grounder accepts arithmetic expressions as well as simple relational expressions (equalities and comparisons), 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 [5].

Propositional interpretations of the theory $\mathrm{vdW}_{k,l,m}$ can be identified with subsets of the set of atoms $\{in\_block(i,b)\colon i \in [m],\ b \in [k]\}$. Namely, a set $M \subseteq \{in\_block(i,b)\colon i \in [m],\ b \in [k]\}$ 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 $in\_block(i,b)$. Clauses (vdW2) ensure that for every $i \in [m]$ there is at least one $b \in [k]$ such that $in\_block(i,b) \in M$. 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 2.2 *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 $\langle\{i \in [m]\colon in\_block(i,b) \in M\}\colon b \in [k]\rangle$ is a partition of $[m]$ into k blocks such that no block contains an arithmetic progression of length l.*

Proposition 2.2 has the following direct corollary.

Corollary 2.3 *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 2.3 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 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 [7] and SATO [23] as 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 [5], 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, *wsat(cc)*, for finding models of theories in logic PS+ that we have designed and implemented recently [14]. Using encodings as theories in logic PS+ and *wsat(cc)* as a solver, we were able to obtain substantially stronger lower bounds for van der Waerden numbers than those known to date.

We will now describe this alternative approach. For a detailed treatment of the PS+ logic we refer the reader to [5]. In this paper, we will only review most basic ideas underlying the logic PS+ (in its propositional form). Let At be a set of propositional atoms. 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 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. We 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: $1\{\mathit{in\_block}(i, 1), \ldots, \mathit{in\_block}(i, k)\}1$, 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 $\mathit{in\_block}(i, b)$ — precisely the same effect as that of clauses (vdW1)

[1] Goldstein [9] provided a precise formula. When $r = (m-1) - (l-1)\lfloor \frac{m-1}{l-1} \rfloor$ and $q = \lfloor \frac{m-1}{l-1} \rfloor$ then there are $q \cdot r + \binom{q-1}{2} \cdot (l-1)$ arithmetic progressions of length l in $[m]$.

and (vdW2). Let $vdW'_{k,l,m}$ be a PS+ theory consisting of clauses (vdW$'$1) and (vdW3). It follows that Proposition 2.2 and Corollary 2.3 can be reformulated by replacing $vdW_{k,l,m}$ with $vdW'_{k,l,m}$ 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 $\Theta(mk)$.

3 Computing models of theories $\mathbf{vdW}'_{k,l,m}$

As we noted earlier, to compute models of theories $vdW_{k,l,m}$ (no c-atoms) we used complete solvers POSIT and SATO. They were only practical for our experiments with the theory $vdW_{4,3,m}$. For the most part, we were working with theories $vdW'_{k,l,m}$ and to compute their models we used the local-search algorithm *wsat(cc)* [14], extended with *bootstrapping* [15]. *Wsat(cc)* is based on the same ideas as *wsat* [19]. The search consists of t *tries*, each starting in a complete truth assignment, called an initial truth assignment (ITA, for short), and proceeding in a sequence of f local improvement steps, called *flips*. A major difference is that due to the presence of c-atoms in c-clauses, *wsat(cc)* uses different formulas to calculate the breakcount and proposes several other heuristics designed specifically to handle c-atoms.

Wsat(cc) is an incomplete solver and it does not 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 an ITA used in the try to a satisfying truth assignment. It is a non-trivial problem to generate "good" ITA's. In [15], 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 T'. Given a theory T and its relaxation T', The bootstrapping consists of using satisfying assignments for T' as ITAs in tries when searching for 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.

To search for models of theories $vdW'_{k,l,m}$, we used *wsat(cc)* combined with bootstrapping. Our approach exploited the fact that if $m' < m$, then the theory $vdW'_{k,l,m'}$ is a relaxation of the theory $vdW'_{k,l,m}$. Indeed, for every partition of $[m]$ into k blocks so that none of the blocks contains an arithmetic progression of length l, the restriction of this partition to $[m']$ is a partition of $[m']$ into k blocks, none of which contains an arithmetic progression of length l. That observation, expressed in terms of models of theories $vdW'_{k,l,m}$ and $vdW'_{k,l,m'}$ directly implies the claim.

In its implementation, bootstrapping uses a sequence of relaxations. To compute models of $vdW_{k,l.m}$, we construct a sequence of relaxations: $vdW_{k,l,m_1}, \ldots, vdW_{k,l,m_k}$, where $m_1 < \ldots < m_k = m$. Given that sequence, the algorithm proceeds as follows:

1. it starts at level 1 and uses *wsat(cc)* with randomly generated ITAs to find models

for the first theory in the sequence, vdW_{k,l,m_1};

2. each time the algorithm finds a model S for a theory vdW_{k,l,m_i}, it moves to the next theory in the sequence, $\mathrm{vdW}_{k,l,m_{i+1}}$, and runs *wsat(cc)* on $\mathrm{vdW}_{k,l,m_{i+1}}$ with the truth assignment given by S as an ITA (we randomly extend S to a complete assignment for the language of the theory $\mathrm{vdW}_{k,l,m_{i+1}}$);

3. if at any level i, *wsat(cc)* fails to find models, the algorithm restarts computation from level 1;

4. if the algorithm finds a model at level k (for the theory $\mathrm{vdW}'_{k,l,m}$), the algorithm stops and outputs the model. Moreover, m is a lower bound for the van der Waerden number $W(k,l)$.

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 counterexamples to the fact that $W(4,3) = 75$. Up to obvious symmetries — permutations of blocks and the "reflection" symmetry $i \mapsto m+1-i$ — 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/`. By inspecting all partitions in the library, one can see that applying the "reflection" symmetry never leads to the same result as applying a "block-permutation" symmetry. It is also easy to see that the "reflection" symmetry commutes with every "block-permutation" symmetry. It follows from these two observations, that the cardinality of the orbit of each of the library partitions is 48. Consequently, the full list of counterexample partitions consists of 1440 elements.

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 2. Partitions demonstrating that $W(2,8) > 1322$, $W(3,5) > 676$, and $W(4,4) > 416$ are included in the appendix. All up-to-date results on the lower bounds on van der Waerden numbers are available at `http://www.cs.uky.edu/ai/vdw/` (we are continually running our local-search solver and update the bounds as we improve on them). To the best of our knowledge there have been no published results on lower bounds for the unknown van der Waerden numbers other than those that follow from the formula of Erdös and Rado [6] and (restricted to only some combinations of k and l) a stronger formula implied by the result of Berlekamp [2]. Our lower bounds are first results obtained through computer calculations and they significantly improve on the values implied by the two formulas mentioned above.

Table 2: Extended results on van der Waerden numbers

k \ l	3	4	5	6	7	8
2	9	35	178	> 341	> 614	> 1322
3	27	> 193	> 676	> 2236		
4	76	> 416				
5	> 125	> 880				
6	> 194					

Table 3: Numbers of atoms and clauses in theories $\mathrm{vdW}'_{k,l,m}$, used to establish the results presented in Table 2.

k \ l	3	4	5	6	7	8
2	18 41	70 409	356 7922	682 23257	1228 23834	2644 249670
3	108 534	579 18529	2028 171028	6708 1498792		
4	304 5700	1664 114956				
5	625 19345	4400 644015				
6	1164 56066					

To provide some insight into the complexity of the satisfiability problems involved, in Table 3 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 2.

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 [17] and *berkmin* [8] 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

This research has been supported by the Center for Communication Research, La Jolla and by the NSF grants IIS-0097278 and IIS-0325063.

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] M.R. Dransfield, V.M. Marek and M. Truszczyński. Satisfiability and computing van der Waerden numbers, in *Theory and Applications of Satisfiability Testing, 6th In-*

ternational Conference, SAT-2003, Lecture Notes in Computer Science, 2919, pages 1–13, Springer, 2001.

[5] D. East and M. Truszczyński. Predicate-calculus based logics for modeling and solving search problems. *ACM Transactions on Computational Logic*, To appear, available at `http://www.acm.org/tocl/accepted.html`, 2005.

[6] P. Erdös and R. Rado. Combinatorial theorems on classifications of subsets of a given set, *Proceedings of London Mathematical Society*, 2:417–439, 1952.

[7] J.W. Freeman. *Improvements to propositional satisfiability search algorithms*, PhD thesis, Department of Computer Science, University of Pennsylvania, 1995.

[8] E. Goldberg, Y. Novikov. BerkMin: a Fast and Robust SAT-Solver. DATE-2002, pages 142–149, 2002.

[9] D. Goldstein. Personal communication, 2002.

[10] T. Gowers. A new proof of Szemerédi theorem. *Geometric and Functional Analysis*, 11:465-588, 2001.

[11] R.L. Graham, B.L. Rotschild, and J.H. Spencer. *Ramsey Theory*, Wiley, 1990.

[12] A. Hales and R.I. Jewett. Regularity and positional games, *Transactions of American Mathematical Society*, 106:222–229, 1963.

[13] R.E. Jeroslaw and J. Wang. solving propositional satisfiability problems, *Annals of Mathematics and Artificial Intelligence*, 1:167–187, 1990.

[14] L. Liu and M. Truszczyński. Local-search techniques in propositional logic extended with cardinality atoms, *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming, CP-2003*, Lecture Notes in Computer Science, vol. 2833, 495–509, Springer, 2003.

[15] L. Liu and M. Truszczyński. Local-search with bootstrapping. In *Proceedings of SAT 2004*, Vancouver, Canada, 2004.

[16] J.P. Marques-Silva and K.A. Sakallah. GRASP: A new search algorithm for satisfiability, *IEEE Transactions on Computers*, 48:506–521, 1999.

[17] M.W. Moskewicz, C.F. Magidan, Y. Zhao, L. Zhang, and S. Malik. Chaff: engineering an efficient SAT solver, in *SAT 2001*, 2001.

[18] F.P. Ramsey. On a problem of formal logic, *Proceedings of London Mathematical Society*, 30:264–286, 1928.

[19] B. Selman, H.A. Kautz, and B. Cohen. Noise Strategies for Improving Local Search. *Proceedings of AAAI'94*, pp. 337-343. MIT Press 1994.

[20] S. Shelah. Primitive recursive bounds for van der Waerden numbers, *Journal of American Mathematical Society*, 1:683–697, 1988.

[21] E. Szemerédi. On sets of integers containing no k elements in arithmetic progression, *Acta Arithmetica*, 27:199–243, 1975.

[22] B.L. van der Waerden. Beweis einer Baudetschen Vermutung, *Nieuwe Archief voor Wiskunde*, 15:212–216, 1927.

[23] 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 [1322] into two blocks demonstrating that $W(2,8) > 1322$.

Block 1:
1 2 3 4 5 6 7 9 11 12 16 19 20 21 22 25 27 29 31 32 37 40 41 42 43 44 46 52 53 54 56 58 63 65
66 67 69 70 72 73 79 81 82 83 84 85 87 88 89 90 92 94 96 97 98 100 103 104 106 109 110 114
116 117 118 120 127 129 131 133 136 138 142 143 146 147 149 150 154 155 159 161 163 167 168
172 173 175 176 178 185 186 188 189 190 191 193 194 195 196 197 201 202 206 209 211 213 216
218 222 224 225 227 228 229 234 236 237 238 239 240 241 242 248 252 253 257 259 260 261 264
265 266 268 269 270 272 273 275 276 277 278 279 280 282 283 285 291 292 293 294 297 298 299
300 302 306 312 313 314 316 318 321 322 324 326 327 330 332 334 338 339 340 343 345 347 352
355 356 357 358 359 360 362 364 365 367 369 371 372 375 380 382 383 384 386 387 388 389 391
395 396 398 405 409 410 411 415 419 423 424 425 426 427 429 430 433 435 436 440 444 445 447
448 452 453 454 460 466 467 468 469 472 475 477 479 481 482 483 484 485 487 490 491 492 494
496 500 501 502 503 504 505 507 509 510 511 513 514 519 521 522 523 ! 527 529 530 531 533

534 536 537 538 539 540 541 542 544 547 549 550 552 553 554 555 556 560 565 566 568 570 571 572 574 575 576 578 585 586 588 595 601 602 608 612 614 617 620 623 625 627 629 632 633 634 635 636 637 638 641 642 644 647 652 654 656 659 660 661 662 667 669 670 671 673 674 675 676 677 680 682 685 686 687 689 691 693 695 697 698 700 702 703 705 708 709 710 711 712 715 716 718 720 721 722 723 724 725 732 735 736 738 739 743 747 751 753 754 755 758 762 763 764 767 769 771 772 773 774 775 776 777 780 781 782 783 786 788 789 791 796 798 799 801 803 811 812 813 814 817 818 819 820 823 825 827 828 830 831 832 833 834 835 836 838 844 846 847 854 858 861 862 864 867 868 869 870 871 874 876 879 884 885 888 889 890 891 892 893 894 896 897 898 900 901 905 906 907 910 911 912 913 914 915 916 918 923 925 926 927 928 930 931 932 934 936 938 939 941 942 943 948 953 955 959 960 961 969 970 971 972 973 977 980 981 983 985 986 991 994 995 996 997 999 1000 1001 1002 1004 1006 100! 8 1009 1012 1013 1015 1016 1017 1018 1019 1021 1022 1026 1027 ! 1029 103 0 1033 1038 1039 1041 1046 1047 1048 1050 1051 1053 1054 1056 1057 1061 1062 1063 1065 1066 1068 1069 1071 1073 1074 1077 1078 1082 1084 1085 1086 1087 1090 1092 1093 1096 1098 1102 1103 1109 1112 1115 1117 1118 1122 1123 1125 1129 1130 1131 1133 1135 1137 1139 1140 1142 1143 1144 1145 1147 1148 1149 1153 1154 1156 1157 1163 1166 1169 1171 1172 1173 1175 1176 1180 1184 1186 1187 1188 1194 1198 1199 1203 1204 1205 1206 1208 1210 1211 1212 1213 1216 1217 1220 1221 1224 1227 1229 1230 1235 1236 1238 1240 1241 1243 1247 1248 1249 1250 1251 1255 1256 1257 1258 1259 1262 1264 1267 1268 1270 1273 1275 1276 1278 1280 1285 1286 1287 1288 1290 1291 1295 1296 1298 1299 1301 1302 1304 1306 1309 1311 1315 1320 1321

Block 2:
8 10 13 14 15 17 18 23 24 26 28 30 33 34 35 36 38 39 45 47 48 49 50 51 55 57 59 60 61 62 64 68 71 74 75 76 77 78 80 86 91 93 95 99 101 102 105 107 108 111 112 113 115 119 121 122 123 124 125 126 128 130 132 134 135 137 139 140 141 144 145 148 151 152 153 156 157 158 160 162 164 165 166 169 170 171 174 177 179 180 181 182 183 184 187 192 198 199 200 203 204 205 207 208 210 212 214 215 217 219 220 221 223 226 230 231 232 233 235 243 244 245 246 247 249 250 251 254 255 256 258 262 263 267 271 274 281 284 286 287 288 289 290 295 296 301 303 304 305 307 308 309 310 311 315 317 319 320 323 325 328 329 331 333 335 336 337 341 342 344 346 348 349 350 351 353 354 361 363 366 368 370 373 374 376 377 378 379 381 385 390 392 393 394 397 399 400 401 402 403 404 406 407 408 412 413 414 416 417 418 420 421 422 428 431 432 434 437 438 439 441 442 443 446 449 450 451 455 456 457 458 459 461 462 463 464 465 470 471 473 474 476 478 480 486 488 489 493 495 497 498 499 506 508 512 515 516 517 51! 8 520 524 525 526 528 532 535 543 545 546 548 551 557 558 559 561 562 563 564 567 569 573 577 579 580 581 582 583 584 587 589 590 591 592 593 594 596 597 598 599 600 603 604 605 606 607 609 610 611 613 615 616 618 619 621 622 624 626 628 630 631 639 640 643 645 646 648 649 650 651 653 655 657 658 663 664 665 666 668 672 678 679 681 683 684 688 690 692 694 696 699 701 704 706 707 713 714 717 719 726 727 728 729 730 731 733 734 737 740 741 742 744 745 746 748 749 750 752 756 757 759 760 761 765 766 768 770 778 779 784 785 787 790 792 793 794 795 797 800 802 804 805 806 807 808 809 810 815 816 821 822 824 826 829 837 839 840 841 842 843 845 848 849 850 851 852 853 855 856 857 859 860 863 865 866 872 873 875 877 878 880 881 882 883 886 887 895 899 902 903 904 908 909 917 919 920 921 922 924 929 933 935 937 940 944 945 946 947 949 950 951 952 954 956 957 958 962 963 964 965 966 967 968 974 975 976 978 979 982 984 987 988 989 990 992 993 998 1003 1005 1007 1010 1011 1014 1020 102! 3 1024 1025 1028 1031 1032 1034 1035 1036 1037 1040 1042 1043 ! 1044 104 5 1049 1052 1055 1058 1059 1060 1064 1067 1070 1072 1075 1076 1079 1080 1081 1083 1088 1089 1091 1094 1095 1097 1099 1100 1101 1104 1105 1106 1107

1108 1110 1111 1113 1114 1116 1119 1120 1121 1124 1126 1127 1128 1132 1134 1136 1138 1141 1146 1150 1151 1152 1155 1158 1159 1160 1161 1162 1164 1165 1167 1168 1170 1174 1177 1178 1179 1181 1182 1183 1185 1189 1190 1191 1192 1193 1195 1196 1197 1200 1201 1202 1207 1209 1214 1215 1218 1219 1222 1223 1225 1226 1228 1231 1232 1233 1234 1237 1239 1242 1244 1245 1246 1252 1253 1254 1260 1261 1263 1265 1266 1269 1271 1272 1274 1277 1279 1281 1282 1283 1284 1289 1292 1293 1294 1297 1300 1303 1305 1307 1308 1310 1312 1313 1314 1316 1317 1318 1319 1322

Next, we exhibit a partition of [676] into three blocks demonstrating that $W(3,5) > 676$.

Block 1:
2 5 6 7 8 10 11 15 25 30 31 32 33 39 41 43 47 49 56 58 62 63 65 67 71 73 75 76 77 87 88 93 95 106 108 109 110 112 118 120 122 125 126 128 129 130 132 133 136 137 138 145 147 150 153 155 157 159 166 167 172 173 174 176 178 179 182 183 184 186 187 188 191 197 198 202 205 208 210 211 220 231 233 251 252 266 268 273 276 277 278 281 282 286 288 289 291 292 293 297 301 302 307 308 310 311 313 315 316 317 318 320 322 323 327 330 331 332 336 340 341 342 345 348 351 353 357 359 360 365 369 372 376 377 386 405 411 414 417 419 422 423 425 426 432 434 435 442 443 444 446 447 449 451 454 455 457 458 460 461 466 477 480 484 485 486 489 490 492 500 501 505 507 508 511 513 515 517 520 521 522 524 530 532 536 541 552 562 563 565 566 567 568 570 571 572 577 591 592 598 601 610 616 617 618 622 627 630 632 634 635 636 640 651 653 656 657 660 661 662 666 667 672 676

Block 2:
1 3 4 9 12 13 16 21 22 26 29 37 38 44 46 48 50 51 54 55 59 61 64 66 69 79 80 82 84 85 86 91 92 96 97 98 100 105 107 111 113 116 119 123 131 134 135 141 144 146 149 151 158 161 164 168 170 180 181 190 193 194 203 206 207 213 215 216 218 219 221 223 226 227 228 229 234 235 236 238 239 241 243 248 250 256 259 260 261 264 270 271 274 275 279 284 285 296 300 304 306 312 319 324 325 326 328 334 335 338 339 346 349 355 358 366 367 368 371 373 374 378 379 380 384 387 389 391 392 396 398 400 401 406 409 413 416 421 428 430 431 433 436 437 438 440 441 445 450 453 456 459 463 465 467 468 470 471 473 479 481 483 491 495 497 499 503 504 510 514 528 531 534 535 540 543 544 545 546 549 550 556 558 559 560 561 564 569 574 575 576 580 581 583 584 585 586 588 595 599 605 606 608 609 611 613 620 621 625 626 629 638 639 641 643 646 648 649 650 659 663 664 665 668 670 671 674

Block 3:
14 17 18 19 20 23 24 27 28 34 35 36 40 42 45 52 53 57 60 68 70 72 74 78 81 83 89 90 94 99 101 102 103 104 114 115 117 121 124 127 139 140 142 143 148 152 154 156 160 162 163 165 169 171 175 177 185 189 192 195 196 199 200 201 204 209 212 214 217 222 224 225 230 232 237 240 242 244 245 246 247 249 253 254 255 257 258 262 263 265 267 269 272 280 283 287 290 294 295 298 299 303 305 309 314 321 329 333 337 343 344 347 350 352 354 356 361 362 363 364 370 375 381 382 383 385 388 390 393 394 395 397 399 402 403 404 407 408 410 412 415 418 420 424 427 429 439 448 452 462 464 469 472 474 475 476 478 482 487 488 493 494 496 498 502 506 509 512 516 518 519 523 525 526 527 529 533 537 538 539 542 547 548 551 553 554 555 557 573 578 579 582 587 589 590 593 594 596 597 600 602 603 604 607 612 614 615 619 623 624 628 631 633 637 642 644 645 647 652 654 655 658 669 673 675

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 2 are available at `http://www.cs.uky.edu/ai/vdw/`.