Report on the

# Workshop on Satisfiability

Certosa di Pontignano
Universita degli Studi di Siena
Siena, Italy

April 28, 1996 to May 3, 1996


**Organized by**

**John Franco**, University of Cincinnati, USA
**Giorgio Gallo**, Universita di Pisa, Italy
**Hans Kleine Büning**, Paderborn, Germany
**Ewald Speckenmeyer**, Universität zum Köln, Germany

**Local Arrangements by**

**Cosimo Spera**, Universita degli Studi di Siena, Italy


Prepared for the

## Office of Naval Research

by


**John Franco**[1]
Computer Science Division, ECECS
University of Cincinnati
Cincinnati, Ohio 45221-0030



[1]Funding for John Franco, one of the organizers, and John Schlipf, an invited participant, was provided by the office of Naval Research under grant number N00014-94-1-0382.

# Contents

# 1 Introduction

This workshop is the latest example of steadily increasing work and results in the area of propositional satisfiability over the last few years. The following recent events attest to this rising interest in the subject. There have been Satisfiability competitions in Paderborn and FAW (Ulm), Germany; Rutgers University (DIMACS), the United States; and Beijing, China. There have been two DIMACS workshops on satisfiability including one organized by Selman and Kautz and one organized this year by Gu, Du, and Pardalos. Last year, a Boolean symposium was organized by Golombic as part of EURO 14 in Jerusalem, and Boolean sessions within OR conferences such as EURO, INFORMS and the Symposium on Math Programming have become common. Books by Kleine Büning, Truemper, and Hooker either have appeared or are due to appear soon. A comprehensive survey on satisfiability algorithms, by Gu, Purdom, Franco, and Wah, should appear soon. Recently initiated major projects include a new book on satsfiability algorithms, an IEEE tutorial on satisfiability, and a special issue of the *Communications of the ACM* on satisfiability algorithms.

Although quite a large body of results on the subject had been obtained before the late 1980's, there are perhaps two main reasons for the sudden entry into the field of extremely good talent and the subsequent explosion of new results; one of these has to do with special applications and one with theoretical aspects of satisfiability.

On the applied side, a growing number of companies seem convinced that a good understanding of satisfiability algorithms will help their competitive standing. This may be due to the success of certain non-resolution-based algorithms, such as the local improvement algorithms of Gu, Selman and others, on a host of military and industrial applications. While not yet explainable analytically, the performance of such algorithms on satisfiable formulas has been reported to be surprisingly good even when such formulas have thousands of variables. Moreover, it seems that for some $\mathcal{NP}$-complete problems, such as the Steiner Tree problem, transforming to a satisfiability problem and solving with one of the new satisfiability algorithms can be more effective than solving the problem in the original domain. Finally, chip design diasters, such as the floating point failure of Pentium processors, point out the need for satisfiability-based tools to aid in the verification process. Indeed, Randy Bryant has recently shown that the floating point problem could have been discovered if tools already developed by him had been used by Intel.

On the theoretical side, two major questions concerning the lengths of resolution proofs were at least partially answered in the late 1980s and early 1990's. One series of papers authored by Tseitin, Galil, Haken, Urquhart, and Chvátal and Szemerédi reached a milestone with the classic 1988 result that all resolution proofs on nearly all randomly generated unsatisfiable formulas of constant-width are of exponential length if the ratio of clauses to variables is held constant. Simultaneously, another collection of papers by Chao, Franco, Chvátal, Reed, Frieze, and Suen showed that a Davis-Putnam variant is effective on nearly all random formulas of constant-width if the ratio of clauses to variables is not too high. Such results, although startling, raised more open questions such as the following. Why can't Davis-Putnam algorithms be proven effective for random formulas with close to a 50 percent chance of being satisfiable? Can it be shown that resolution proofs must be large for most random formulas even if the ratio of clauses to variables grows? Is there a 0-1 (threshold) law governing the probability that a random formula of constant-width is satisfiable? Do random formulas of constant-width usually belong to a special polynomially-solved subclass of satisfiability (such as q-SAT).

Recent attacks on these and other questions have yielded a steady stream of improvements to our understanding of resolution proof size and the threshold phenomenon. Recently, Beame and Pitassi developed new techniques to show that the result of Chvátal and Szemerédi can be extended to growing ratios of clauses to variables (up to a point). Several recent papers involving de la Vega, Motwani, Spirakis, Kamath, Dubois, and others introduced techniques that have reduced the known upper limit on the satisfiability threshold to somewhere near where it is suspected to be. Special polynomially solvable subclasses of satisfiability that are based on LP relaxations have recently been shown by Schlipf, Annexstein, Franco, and Swaminathan to be subsumed by Davis-Putnam algorithms and, in a probabilistic sense, these classes have been shown by Franco and VanGelder to be considerably smaller than the class of constant-width formulas solved in polynomial time by matching. Hierarchies of fixed-parameter-tractable classes of constant-width formulas have been defined by Gallo, Heusch, Speckenmeyer, Dalal and others, and surprising progress on the complexities of such classes has been made by Schlipf, Franco, Goldsmith, Swaminathan, and Speckenmeyer. Also of interest are the series of improvements by Monien, Speckenmeyer, Schiermeyer, and Kullmann to upper bounds on the complexity of Davis-Putnam style algorithms given constant-width formulas (now down to approximately $\Theta(1.5^n)$).

On the experimental side, some old techniques are beginning to be applied to satisfiability with promising results. Notable among these are the Lagrangian techniques of (independent and different) vanMaaren, Nobili, and Wah.

With this exciting backdrop of activity, the organizers chose to have a small workshop, lasting an entire week, of representatives of the main areas identified above. There were only four or five 45 minute talks a day with plenty of time to get into details. There was also plenty of time for informal one-one and group discussions afterwards. The idea was to have all the participants know each other much better after the workshop than before.

The body of this report explains what was learned, and contains the personal reactions of the author of this report. Appendices include open problems, the announcement, the program, and abstracts. Much of the information presented here can be found on the World Wide Web at the URL:

*http://www.ece.uc.edu/~franco/sat-workshop-aftermath.html*

# 2 Results

Results are presented by category, following threads of research activity. The *constant-width* model used below is a parameterized probability distribution on satisfiability formulas in Conjunctive Normal Form (CNF) described as follows: a random formula contains $m$ clauses independently chosen from the set of all $k$-literal clauses that can be composed from $n$ variables (no two literls are the same or complementary). The *random-width* model is described as follows: a random formula contains $m$ clauses independently constructed from a set of $n$ variables and their complements ($2n$ literals) where each literal exists in a given clause with probability $p$, independently of the rest. When we say Davis-Putnam style algorithms we mean algorithms incorporating splitting and possibly other rules for variable elimination found in [8]. Sections are ordered according to the personal taste of the author.

## 2.1 0-1 threshold for random constant-width formulas

Early work [4] on the probabilistic analysis of satisfiability algorithms with respect to the constant-width model revealed the probability that a random formula is satisfiable tends to 0 if the ratio $m/n > -1/\log_2(1-2^{-k})$. If $k = 3$ this is about $m/n > 5.19$. The result is simple to obtain and has been rederived many times. The idea is to find an expression for the expected number of satisfying truth assignments for a random formula and then find the conditions for which that expectation tends to 0. Since the expectation is an upper bound on the probability that a random formula is satisfiable, these conditions also apply to the probability that a random formula is satisfiable. Unfortunately, the variance of the number of satisfying truth assignments is too large for the above bound to be tight.

Later work [2, 3, 5] showed that Davis-Putnam like algorithms with no or limited backtracking can find a truth assignment satisfying a random formula with probability tending to 1 if

$$m/n < .46 \left(\frac{k-1}{k-2}\right)^{k-2} \left(\frac{2^k}{k+1}\right) - 1, \quad \text{for } 4 \leq k \leq 40;$$

$$m/n < .125 \left(\frac{k-1}{k-3}\right)^{k-3} \left(\frac{k-1}{k-2}\right) \left(\frac{2^k}{k}\right), \quad \text{for } 3 \leq k;$$

$$m/n < 3.003, \quad \text{for } k = 3.$$

These results plus the simple result cited above have left an irresistable gap of roughly $2^k/k < m/n < 2^k$ where it is not known even whether the probability that a random formula is satisfiable tends to 0 or 1, let alone whether any algorithms are efficient in a probabilistic sense.

Recent work by several groups has attacked the gap from above for $k = 3$. In the span of a little over a year, it has fallen from 5.19, to 5.01, to 4.75, and now to 4.64. The techniques used to derive the last result promise to settle the question from above, once and for all. The idea, shared by two groups, is to lexicographically order satisfying truth assignments and count the subset beginning with a particular kind of assignment. If the subset is empty, then there are no satisfying truth assignments for the given formula. Since the subset is very small and has a low variance, the expectation of its size is a tight bound for the probability that a satisfying assignment exists.

Representatives from both groups were invited to the workshop. One agreed to come but at the last minute could not obtain the funding to do so. Therefore, Goerdt presented these results at the workshop.

We remark that in 1991 Goerdt and Chvátal and Szemerédi showed that there is a 0-1 threshold for $k = 2$ at $m/n = 1$.

## 2.2 Lagrangian methods

Lagrangian methods have recently been applied to satisfiability. Two papers presented at the workshop are concerned with such relaxations. The general goal is to introduce quadratic cuts which, in one case (vanMaaren), leads to better approximations for the solution space of LP formulations of satisfiability and, in the other case

(Nobili), for the maximum satisfiability problem, leads to computations of approximations to LP optima that are faster than applying the simplex method.

The use of Lagrangian methods is outlined as follows. Let $\hat{A}^{\mathcal{F}}$ be the clause-variable incidence matrix associated with formula $\mathcal{F}$. That is, element $\hat{A}^{\mathcal{F}}_{ij}$ has value 1 if literal $x_j$ is in the $i$th clause, -1 if literal $\bar{x}_j$ is in the $i$th clause, and 0 if neither literal $x_j$ nor literal $\bar{x}_j$ are in the $i$th clause. Let $\nu(\hat{A})$ be the vector such that the $i$th component is the number of -1's in the $i$th row of $\hat{A}$. Let $\mathbf{e}_n$ be the $n$-vector of all 1's. Then the satisfiability problem may be formulated as

$$\hat{A}z \geq \mathbf{e}_n - \nu(\hat{A}), \qquad z \in \{0,1\}^n,$$

and the maximum satisfiability problem may be formulated as minimize $c^T x$ subject to

$$[\hat{A} \;\; I]x \geq \mathbf{e}_{m+n} - \nu([\hat{A} \;\; I]), \qquad x \in \{0,1\}^{m+n},$$

where $c$ is the vector of dimension $n+m$ whose first $n$ elements are 0 and whose remaining elements are 1; $x$ is the vector of dimension $n+m$ whose first $n$ elements are $z$, above, and whose remaining elements "measure" which clauses are satisfied; and $I$ is the appropriate identity matrix.

The relaxation for maximum satisfiability considered by Nobili is

$$\max_{\lambda}\{L(\lambda) = \min_{x}\{(c^T - \lambda^T[\hat{A} \;\; I])x + \lambda^T(\mathbf{e}_{n+m} - \nu([\hat{A} \;\; I]))\}\},$$

where $\lambda$ is an $m$ dimensional vector of positive reals. $L(\lambda)$ is an approximation for $c^T x$ that bounds it from below. A $\lambda$ can be built iteratively to get a good approximation: at iteration $t$, the aim is to find the $\lambda$ that maximizes $L$ for a row-submatrix of a modified $\hat{A}$ (see below) then add violated constraints and continue the process. In general, it is not easy to extend the $\lambda$ that is optimal at iteration $t$ to a $\lambda$ that is optimal at iteration $t+1$. In fact, the non-zero elements of the optimal $\lambda$ for iteration $t$ may be completely different from those of the optimal $\lambda$ for iteration $t+1$. However, Nobili has noticed some conditions under which it can be done easily (that is, a new $\lambda$ and additional constraints can be constructed by inspection instead of by solving a linear programming relaxation). Such tests may be added to speed up existing simplex-based solutions borrowed from the Set Covering problem. We note that the constraints added by Nobili aim toward the *logical completion* of $\hat{A}$. The logical completion includes $\hat{A}$ and all pairwise "resolvents" derived from rows of $\hat{A}$. The logical completion of $\hat{A}$ is usually much bigger than $\hat{A}$. Currently, not enough experiments have been performed to ascertain the effectiveness of these results.

The relaxation for satisfiability considered by vanMaaren is more complex. Let $\hat{A}$ be defined for CNF formula $\mathcal{F}$. Let $C_i$ be the $i$th row of $\hat{A}$. Let $P_i$ be the set of integers representing column numbers of $\hat{A}$ for which 1s exist in $C_i$. Let $N_i$ be the set of integers representing column numbers for which -1s exist in $C_i$. Define

$$w_i(x) = 1 - \sum_{k \in P_i} x_k - \sum_{j \in N_i} (1 - x_j), \quad \text{for } -\infty < s \leq 1.$$

Define

$$\mathcal{A}_\epsilon(s) = \frac{s + \sqrt{s^2 + \epsilon}}{1 + \sqrt{1 + \epsilon}},$$

where $\epsilon$ is a parameter depending only on clause length. Define

$$\Phi^{\mathcal{F}}(x) = (\sum_{i=1}^{m} (1 - \mathcal{A}_{\epsilon_i}(w_i(x)))^r)^{1/r}/m,$$

where $r < 0$ is a parameter and $\epsilon_i = \epsilon_j$ if the number of non-zero entries in $C_i$ and $C_j$, $i \neq j$, is the same. It has been shown that there exists a threshold $h_{\mathcal{F},\epsilon,r}$, depending only on countable properties of $\mathcal{F}$ and the $\epsilon$ and $r$ parameters, such that $\Phi^{\mathcal{F}}(t) \geq h_{\mathcal{F},\epsilon,r}$ if and only if $\mathcal{F}$ is satisfied at $t$. Thus, the inequality $\Phi^{\mathcal{F}}(x) \geq h_{\mathcal{F},\epsilon,r}$ defines a region separating the satisfying and non-satisfying assignments of $\mathcal{F}$. For the sake of computational tractability, vanMaaren replaces $\Phi$ with its second order Taylor expansion around the center $c$ of the unit cube ($x_i = 1/2$, $1 \leq i \leq n$). The aim is to find, for some parameter values, a *good* pair of thresholds $h_n$ and $h_s$ such that $\phi(\psi) < h_n$ defines a region containing no satisfying truth assignments and $\phi(\psi) > h_s$ defines a region containing only satisfying truth assignments, where $\phi(\psi) = \Phi(c + \psi) - \Phi(c)$ and $\Phi$ is the second order Taylor expansion of $\Phi^{\mathcal{F}}$.

The relaxation proposed by vanMaaren can be used to determine satisfiability for given formulas. In this workshop vanMaaren shows that his *geometric interpretation* can possibly be used to develop intuition about the properties of formulas that relate to "hardness." For example, 3-CNF formulas with the property that the number of occurrences of positive literals for each variable are about equal to the number of occurrence of negative literals yield regions that are nearly spherical and centered at the center of the unit cube: that is, they present little discriminating information. Such "sign balanced" formulas are thought to be very hard.

## 2.3 Probabilistic analysis of Davis-Putnam variants

Although most current research activity on the probabilistic analysis of satisfiability algorithms is related to the constant-width model, a string of results spanning about 10 years, mainly due to Purdom and colleagues, has revealed the average performance of satisfiability algorithms with respect to the random-width model.

Early work on this model showed that most random formulas are trivial. Thus, if $p > \ln(m)/n$, a random assignment satisfies a random instance with high probability and if $p < \ln(m)/(2n)$, random formulas have at least one null clause (and therefore are unsatisfiable) with high probability. In between, a Davis-Putnam variant finds satisfying truth assignments, when at least one exists, with high probability.

Surprizingly, finding a collection of algorithms that, when run simultaneously have polynomial average time behavior, proved very hard. Interest in such results is due to the fact that, in practice, one pathological formula (requiring an extraordinary amount of time) out of a large sample of formulas can be intolerable; a polynomial-time *average* result suggests this won't happen. Until recently, no such result was known for much of the parameter space of the random-width model. As reported in this workshop, Purdom has found a collection of algorithms that succeeds.

The collection consists of one previously studied algorithm and a new one. The new algorithm is another variant of the Davis-Putnam procedure. Fix any truth assignment $t$ to the variables of the formula, and recurse on the following as long as necessary: locate a clause $C = \{l_1, l_2, ..., l_x\}$ (where $l_i$, $1 \leq i \leq x$ are literals) that is falsified by $t$, split on $l_1$, under $l_1 = false$ split on $l_2$, under $l_1 = false$ and $l_2 = false$ split on $l_3$ and so on up to $l_x$. Purdom shows that the algorithm outlined above has polynomial average-time complexity when $p \geq \ln(m)/n$. Previously, it had been shown that a restricted form of resolution provides polynomial-size proofs, on the average, when $p \leq \ln(m)/n$.

## 2.4 Fixed-parameter-tractable hierarchies of SAT classes

Several hierarchies of successively "harder" satisfiability classes have been proposed. Recently, Heusch proposed such a hierarchy based on pure implication logic. Pure implication formulas are defined recursively as follows:

1. A variable is a pure implication formula.
2. If $\mathcal{F}_1$ and $\mathcal{F}_2$ are pure implication formulas then $(\mathcal{F}_1 \rightarrow \mathcal{F}_2)$ is a pure implication formula.

Eliminating parentheses on right to left associativity, a pure implication formula can be written $\mathcal{F}_1 \rightarrow \mathcal{F}_2 \rightarrow ... \rightarrow \mathcal{F}_p \rightarrow z$ where $z$ is a variable. We call the $z$ variable of a formula the *right-end* variable.

The satisfiability problem is trivial for a pure implication formula but the problem of falsifiability is $\mathcal{NP}$-complete even if all variables except the right-end variable occur at most twice in the formula. (note that this is obviously equivalent to a satisfiability problem on CNF formulas constructed in a particular way from a given pure implication formula). Let $k$ be the number of occurrences of the right-end variable. Heusch [6] defined a hierarchy of satisfiability classes based on pure implication logic, with parameter $k$, and showed that the $k$th level of the hierarchy can be solved in time $O(|\mathcal{F}|^k)$. Thus, level 1 is linear-time solvable, level 2 is quadratically solvable, and so on. A remarkable feature of the lower levels of the hierarchy is that they have extremely limited expressibility and yet these classes are incomparable with repect to other polynomial-time solvable classes such as 2-SAT, q-SAT, and SLUR. This feature might become a tool for investigating the hardness of formulas, both for satisfiability and other $\mathcal{NP}$-complete problems.

To use this feature effectively in this regard, we must better understand the complexities associated with each level of the hierarchy. In particular, there is nothing known that theoretically prevents the complexity of level $k$ formulas from being $O(2^k|\mathcal{F}|^c)$ where $c$ is a constant. The paper presented by John Schlipf finds that level $k$ of the hierarchy can be solved in $O(k^k|\mathcal{F}|^2)$ time. Thus, Heusch's hierarchy is the first satisfiability hierarchy known to us that is *fixed-parameter-tractable*. This result may impact known complexities of hierarchies of problems analysed in the growing fixed-parameter-tractable literature. For some of these, the best known complexities are of $O(n^k)$ where $n$ represents the length of the input. The result also provides a better understanding of where to look for "borderline-hard" formulas.

Heusch has looked at properties of pure implication formulas that lead to linear-time solutions. In his talk, he showed that if all subformulas of the type $(u \rightarrow (v \rightarrow w))$ are forbidden, then the falsifiability problem can be solved in linear time. Other linear-time subclasses were presented as well.

## 2.5 Upper bounds on the complexity of 3-satisfiability

An interesting thread of results concerns upper bounds on the complexity of certain Davis-Putnam style algorithms for solving 3-satisfiability formulas. Notable progress over the obvious $O(2^n)$ complexity of simple backtracking was first achieved by Monien and Speckenmeyer [7] using the concept of autarkness. A partial truth assignment $t$ for a given subset $L$ of complementary pairs of literals is *autark* in formula $\mathcal{F}$ if and only if for every clause $C \in \mathcal{F}$, if $C$ contains at least one literal in $L$, then $C$ is satisfied by $t$. The following two statements are obvious and lead to a $O(1.61^n)$ algorithm for 3-satisfiability. If a truth assignment $t$ is autark, then all clauses in $\mathcal{F}$ containing at least one literal assigned a value by $t$ are satisfied by $t$. If $t$ is not autark in $\mathcal{F}$ then, in some clause, all literals assigned values by $t$ are falsified.

This result has now been improved by Schiermeyer. The algorithm is too complicated to state here. It is based on repeatedly selecting a shortest clause and then adding a number of tests and modifications to $\mathcal{F}$ which either add or remove clauses from $\mathcal{F}$ and remove literals from clauses in $\mathcal{F}$. The reader is referred to the abstract for details.

One key notion seems to be *pure literal look ahead* which is based on the following idea. Let $C$ be a clause containing two literals $x_1$ and $x_2$ and suppose that for all clauses $C'$ other than $C$, if $x_2 \in C'$ then $x_1 \in C'$. Then assign $x_2$ the value *true* and the result is that $\bar{x}_1$ is a pure literal in $\mathcal{F}$. The above ideas have been used by Schiermeyer to reduce the upper bound to $O(1.57^n)$. His contribution to this workshop is an improved analysis that brings the upper bound on 3-satisfiability complexity to $O(1.49^n)$.

Kullmann has also made an interesting contribution to this area with the introduction of blocked clauses. A clause $C$ is *blocked* with respect to a formula $\mathcal{F}$ if it contains a literal $x$ such that every resolvent of $C$ with another clause on $\bar{x}$ is in some sense not interesting. Using blocked clauses, Kullman has obtained an $O(1.5045^n)$ upper bound for 3-satisfiability complexity.

Perhaps the greatest value of this line of work is 1) the intuition that is developed from the study of many different ideas for choosing a variable to split on and the notion of blocked clauses; 2) the generalizations devised by Kullmann.

## 2.6 Resolution proof length

Finding bounds on the lengths of resolution proofs has a long history involving classic results by Tseitin, Haken, Urquhart, Chvátal, Szemerédi, Goerdt and others. Generally, the results are negative: families of formulas are shown to require exponential length proofs.

Urquhart studies the effect of symmetry rules on resolution proof length. Let $\sigma_\pi(\mathcal{C})$ be the clauses obtained by applying a permutation $\pi$ on the variables of $\mathcal{C}$ to each of the clauses of $\mathcal{C}$. Suppose a clause $C$ has been derived from a set of clauses $\mathcal{C}$ and $\sigma_\pi(\mathcal{C}) = \mathcal{C}$. Then $\sigma_\pi(C)$ can be inferred as the next step in the derivation. This is the global symmetry rule. Suppose $C$ is a clause derived from $\mathcal{C}$ and for every $C' \in \mathcal{C}$ used in the derivation of $C$ $\sigma_\pi(C')$ is also in $\mathcal{C}$. Then $\sigma_\pi(C)$ can be inferred on the next step. This is the local symmetry rule. The symmetry rules can have a dramatic effect on resolution proof length in a variety of instances. For example, proofs on pigeon-hole formulas can be reduced from exponential to quadratic length. Moreover, these rules can be generalized by allowing complementation of literals as well as permutations. Urquhart shows that this extension of the symmetry rules is as powerful as extended resolution. That is, resolution with the global symmetry rule allowing complementation is p-equivalent to extended resolution.

Van Gelder proposes a new pruning method to reduce the length of proofs. This method is designed to prevent refutation attempts that can not possibly succeed. The method is based on autarkness (see Section 2.5). The reader is referred to the extended abstract for details.

Kleine Büning and Lettmann consider the question whether, for an arbitrary clause $C$ in formula $\mathcal{F}$, $\mathcal{F} \models C$ holds. This is often hard to determine. In their paper they investigate whether the hardness is due to structural properties of formulas or to the meaning of the formula: that is, whether, for a given formula $\mathcal{F}$, there exists an equivalent formula $\mathcal{F}$' of restricted length for which $\mathcal{F}' \models C$ and this can be determined quickly. The conclusion is that for all polynomials $p$ and $q$, there exists a formula $\mathcal{F}$ such that, for every equivalent formula $\mathcal{F}$' with length less than $q(|\mathcal{F}|)$, there is a clause $C$ for which the shortest resolution proof proving that $C$ is a consequence of $\mathcal{F}$' requires more than $p(|\mathcal{F}|)$ time.

## 2.7 Polynomial time solvable subclasses of satisfiability

Several polynomial-time solvable subclasses of satisfiability have been proposed. Among the more interesting are the Horn, extended Horn, balanced matrix, single lookahead unit resolution, and q-Horn formulas. An intriguing question is whether any of these classes comes close to what in some sense may be the largest easily definable subclass of satisfiability that is solved in polynomial time. For now we will call such a class of formulas *p-dominant*. The class of q-Horn formulas is thought to be p-dominant because of the following result due to Boros, Crama, Hammer, and Saks [1]. Let $\{v_1, v_2, ..., v_m\}$ be a set of Boolean variables, and $P_k$ and $N_k$, $P_k \cap N_k = \emptyset$ be subsets of $\{1, 2, ..., m\}$ such that the $k$th clause in a CNF formula is given by $\vee_{i \in P_k} v_i \vee_{i \in N_k} \bar{v}_i$.

Construct the following system of inequalities:

$$\sum_{i \in P_k} \alpha_i + \sum_{i \in N_k} (1 - \alpha_i) \leq Z, \quad (k = 1, 2, ..., n), \text{ and}$$

$$0 \leq \alpha_i \leq 1, \quad (i = 1, 2, ..., m).$$

where $Z \in R^+$. If all these constraints are satisfied with $Z \leq 1$ then the formula is q-Horn. On the other hand, the class of formulas such that the minimum $Z$ required to satisfy these constraints is greater than $1 + \epsilon$, for any $\epsilon > 0$, is NP-complete.

Franco and Van Gelder have investigated the possibility that one of the above classes is p-dominant from a probabilistic perspective. Under the constant-width model ($k$ literals per clause), there are several Davis-Putnam style algorithms that almost always find satisfying truth assignments in polynomial time when $m/n < c2^k/k$, $c$ a constant. We would hope, then, that a random formula generated under this constraint would be, say, q-Horn with probability tending to 1. In fact we find that this can only happen if $m/n < c/k^2$. Moreover, the same can be said of the other classes. The reason is that, for each class, there are forbidden structures, typically "cycles" among clauses, and, since the model is symmetric, when cycles begin to be generated in preponderance (that is, when $m/n > c/k^2$) so do forbidden ones.

Equally surprising is a random formula can be solved in polynomial time by the following *matching* algorithm if $m/n < c$. Create a bipartite graph with vertices on one side representing clauses and on the other variables. Put an edge between two vertices on opposite sides if the corresponding variable is in the corresponding clause regardless of polarity. Find a maximum matching for the graph. If the matching covers all clause-vertices then, for each clause-vertex, assign the variable corresponding to the vertex adjacent to it in the matching the value which satisfies the corresponding clause.

The above results show that previously studied classes are, in some probabilistic sense, not p-dominant because a seemingly wide range of random formulas that can be solved easily by even a simple matching algorithm. These results have not yet been written up.

Another interesting question concerning polynomial time solvability is, given a CNF formula $\mathcal{F}$, what is the largest subset of clauses of $\mathcal{F}$ that is renamable Horn (or q-SAT, etc.). Identifying such subsets may have an impact on the time required to solve $\mathcal{F}$. Boros presented an algorithm for finding a non-optimal renamable Horn set $\mathcal{C} \subseteq \mathcal{F}$ such that $|\mathcal{C}|$ is no less than about 60% of the size of the optimal subset. Boros formulates the problem as a pseudo-Boolean function in variables $s_1, ..., s_n$ where $s_i$ is 1 if the literals of the $i$th variable are switched (or renamed) and 0 otherwise. The maximum value of this function is the maximum size of a renamable Horn subset of $\mathcal{F}$. Moreover, the value of the function given a real valued vector $\mathbf{q}$ for the $s_i$ variables is always less than the value given a binary vector and that binary vector can be derived from $\mathbf{q}$ by a simple linear-time rounding procedure. Thus, the continuous solution to the psuedo-Boolean function is less than the optimum size of a renamable Horn subset of clauses. The problem can alternatively be formulated as an Integer Programming problem on $s_1, ..., s_n$ such that the optimum value of the continuous relaxation of this IP is always at least the optimum size of a renamable Horn subset of clauses. Hence, one can find the optimum solution to the LP relaxation, if this is small enough, use $\mathbf{q} = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$ and round to determine an approximate subset, otherwise, use as $\mathbf{q}$ the optimum solution to the IP relaxation and round.

## 2.8 Partially defined Boolean functions

Partially defined Boolean functions (pdBF) have applications in a number of areas of computer science and operations research. An interesting question is whether a given pdBF can be completed to a total Boolean function that is a polynomial-time subclass of satisfiability. It is known that a pdBF can be completed to a Horn formula, if possible, in polynomial time. Such a completion is characteristically unbalanced among the input vectors that satisfy and those that do not. Looking for more balanced completions, Eiter introduces special kinds of Horn formulas, namely *double Horn* and *bi-dual Horn* formulas. He then derives certains properties of these classes related to the completion problem. The reader is referred to the abstract for details.

## 2.9 Multispace search

Gu presented his ideas on the concept of Multispace algorithms. The following description is taken from our joint survey paper.

The goal of traditional optimization algorithms is to find an assignment of values to variables such that all constraints are satisfied and performance criteria are optimized. An optimization algorithm changes values and tries to find the "goal values." Traditional value search methods do not provide structural information to the search problem. It is difficult for them to handle difficult problems, e.g., local minimum points, in a hard search problem. In multispace search, any component related to the given search problem forms a new class of search space. For a given search problem, we define variable space, value space, constraint space, objective space, parameter space, and other search spaces. The totality of all the search spaces constitutes a *multispace*. During the search process, a multispace search algorithm not only changes values in the value space. It also scrambles across other spaces and dynamically *reconstructs* the problem structures that are related to the variables, constraints, objectives, parameters, and other components of the given search problem. Only at the *last moment* of the search, the "reconstructed" problem structure is replaced by the original problem structure, and thus the final value assignment represents the solution to the original search problem.

# 3 Personal Comments

The organizers believe the workshop was a success. The total number of attendees was 25, about 2/3 of whom came from Western Europe and the rest from North America. Most participants stayed for an entire week. Speaking for myself (and certainly similar statements are true for most attendees), I enjoyed the unhurried, lengthy, informal technical and personal conversations with several colleagues, especially vanMaaren, Urquhart, Goerdt, Gu, and Mitchell.

The workshop came only six weeks after a DIMACS workshop on satisfiability. The two complemented each other quite well: there was only a small overlap in results presented; the pace of each was different; and, since the composition of each was different, research thrusts were somewhat different.

In the previous section I have reported interesting research thrusts represented in the Siena workshop. Some of these seem very promising. I most enjoyed the Lagrangian papers of Nobili and vanMaaren. Dave Mitchell's talk, on a topic very similar to our own SLUR paper, was outstanding (I did not report on this above because he has not yet finished writing up his results). John Schlipf gave an excellent account of our work on fixed-parameter-tractable hierarchies for satisfiability. In fact, the spectrum of polynomial-time solvable subclass results, including the work of Boros, Heusch, Gallo, Mitchell, and others, has fascinated me before and during the workshop (particularly because of the close relationship of some of this work to Linear Programming). Work on resolution proof length continues to confirm its limitations from a complexity perspective. However, the work of Urquhart, Van Gelder, Kleine Büning, and others suggests what needs to be added to resolution to make it much faster in some applications. Finally, I want to comment that I enjoyed Eiter's talk on partially defined Boolean formulas, mainly because it was so well presented.

Finally, through my contact with Gu, I will almost certainly be migrating somewhat towards industrial applications of satisfiability and the study of algorithms that are most suitable in that domain.

# References

[1] E. Boros, Y. Crama, P. L. Hammer, and M. Saks. A complexity index for satisfiability problems. *SIAM Journal on Computing* **23**, pages 45–49, 1994.

[2] M. Chao and J. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristic for the $k$-satisfiability problem. *Information Sciences* **51**, pages 289–314, 1990.

[3] V. Chvatal and B. Reed. Mick gets some (the odds are on his side). *Proceedings of the 33rd annual Symposium on Foundations of Computer Science*, IEEE, Los Alamitos, California, 1992.

[4] J. Franco and M. Paull. Probabilistic analysis of the Davis-Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics* **5**, pages 77–87, 1983.

[5] A. M. Frieze, and S. Suen. Analysis of simple heuristics for random instances of 3-SAT. Carnegie-Mellon Tech Report. 1993

[6] P. Heusch. The Complexity of the Falsifiability Problem for Pure Implicational Formulas. *Proc. 20th Int. Symposium on Mathematical Foundations of Computer Science (MFCS'95)*, J. Wiedermann, P. Hajek (Eds.), Prague, Czech Republic. *Lecture Notes in Computer Science* (LNCS **969**), Springer-Verlag, Berlin, pages 221–226, 1995.

[7] B. Monien, and E. Speckenmeyer. Solving satisfiability is less than $O(2^n)$ steps. *Discrete Applied Mathematics* **10**, pages 287–295, 1985.

[8] M. Davis, H. Putnam, and D. Loveland. A machine program for theorem proving. *CACM* **5**, pages 394–397, 1962.

# **APPENDIX A** - Open Problems

# Workshop on Satisfiability

Certosa di Pontignano
Universita Degli Studi Di Siena
Siena, Italy April 28 - May 3, 1996

# Open Problems

## Hans Kleine Büning

**All Literal Horn**

INSTANCE: Satisfiable Horn formula $\mathcal{F}$.

QUESTION: Determine $\{l \text{ literal } | \mathcal{F} \models l\}$.

The set $\{l \text{ literal } | \mathcal{F} \models l\}$ obviously can be calculated in time $O(|\mathcal{F}| * n)$, where $|\mathcal{F}|$ is the length of the formula and $n$ is the number of variables.

The set of positive literals (variables) $\{x \text{ variable } | \mathcal{F} \models x\}$ can be determined in time $O(|\mathcal{F}|)$ using unit resolution. The problem arises with negative literals.

The unique–satisfiability problem for Horn formulas is a special case and known to be solvable in linear time.

Now the question is whether there exists an algorithm solving the Literal Horn problem in time less than $O(|\mathcal{F}| * n)$.

---

## Endre Boros

Given a CNF expression $\mathcal{C} = \{C_1, C_2, ..., C_m\}$, define $L(\mathcal{C}) = \{u |$ both $u$ and $\bar{u}$ occur in $\mathcal{C}\}$.

Prove or disprove the following conjecture: if $|C_i \cap \bar{C}_j| = 1$ for all $i \neq j$, then there exists a $u \in L(\mathcal{C})$ with exactly *one* occurrence in $\mathcal{C}$.

---

## Thomas Eiter

The following restriction (**IMUNSAT**) of the classical satisfiability problem is currently open:

**Intersecting Montone UNSAT**

INSTANCE: A set $\mathcal{C}$ of clauses such that each clause is either positive (i.e., consists entirely of positive literals) or negative (i.e., consists entirely of negative literals) and for each positive clause $C_1$ and negative clause $C_2$ of $\mathcal{C}$, there exists an atom $u$ such that $u$ is in $C_1$ and $\bar{u}$ is in $C_2$.

QUESTION: Is $\mathcal{C}$ unsatisfiable?

**Reference:** Eiter & Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *SIAM J. Computing*, **24**(6): 1278–1304, 1995.

This problem is the (un)satisfiability variant of a number of equivalent (under polynomial time transformation) problems in different areas of operations research and computer science. A number of people have considered essentially this problem in papers, including M. Fredman, T. Ibaraki, D.S. Johnson, L. Khachiyan, R. Khardon, E. Lawler, J. Lenstra, H. Mannila, Ch. Papadimitriou, K.-J. Räihä, A. Rinnooy Kan, M. Yannakakis, to mention some of them. The problem (in equivalent formulations) has been open for more than 15 years.

There is a yet more restricted version of **IMSAT**, which is as hard as the general case.

**Symmetric Intersecting Monotone UNSAT (SIMUNSAT)**

INSTANCE: Restriction of **IMSAT** to instances $\mathcal{C}$ where the negative clauses are precisely all clauses $C-$ such that $C- = \{\bar{u} : u \in C+\}$ for some positive clause $C+ \in \mathcal{C}$. (By this restriction, nonempty positive clauses of $\mathcal{C}$ are mutually intersecting.)

QUESTION: Is $\mathcal{C}$ unsatisfiable?

Although this problem statement seems easy, it appears to be difficult to come up with a polynomial time algorithm. What is known to date is that the problem is most likely not co$\mathcal{NP}$-complete, due to the recent quasi-polynomial time algorithm of Fredman and Khachiyan for the positive dualization problem (see below). This algorithm implies that **IMUNSAT** and **SIMUNSAT** can be solved in quasi-polynomial time, as well as all problems mentioned below.

There are a number of problems that are equivalent to **SIMUNSAT** (under polynomial time transformation). The problems are in different areas of logic, operations research, and computer science. We describe here some of them. (For a more detailed account, see the quoted paper of Eiter and Gottlob.)

**Problems Equivalent to IMUNSAT and SIMUNSAT**

A *hypergraph* $\mathcal{H}$ is a pair $(V, \mathcal{E})$ of a finite set $V$ and a family $\mathcal{E} = \{E_1, ..., E_m\}$ of finite subsets $E_i \subseteq V$ (called edges). A hypergraph is *simple*, if it has no pair of edges $E_i$, $E_j$ such that $E_i$ is properly contained in $E_j$. A simple hypergraph is also known as a *Sperner family*.

Say that a simple hypergraph $(V, \mathcal{E})$ is *saturated* if adding any further edge violates the property of being simple, i.e., $(V, \mathcal{E} \cup \{X\})$ where $X \subseteq V$, is not simple.

The following problem is equivalent to **SIMUNSAT**:

**Simple Hypergraph Saturation (SIMPLE-H-SAT)**

INSTANCE: A simple hypergraph $\mathcal{H} = (V, \mathcal{E})$ on vertices $V = \{v_1, ..., v_n\}$.

QUESTION: Is $\mathcal{H}$ saturated?

A variant of this problem is that, besides being simple, the hypergraph must be (and remain) in addition *intersecting*, i.e., each pair of edges $E_i$, $E_j$ has nonempty intersection. This problem, known as **Maximal Hypergraph Clique**, has been presented by D.S. Johnson as an open problem in a lecture "Open and Closed Problems in NP-Completeness" at the Symposium and Summer School "$\mathcal{NP}$-completeness: The First 20 Years", Erice, Sicily, 1991. (Actually, Johnson presented a spoiled version of the problem there (personal communication) which is clearly solvable in polynomial time).

A *transversal* (or *hitting set*) of a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is a subset $T \subseteq V$ such that $T$ meets each edge in at least one vertex, i.e., $|T \cap E_i| \geq 1$, for all $E_i$ in $\mathcal{E}$. The *transversal hypergraph* of $\mathcal{H}$ is the hypergraph $Tr(\mathcal{H}) = (V, \mathcal{F})$ such that $\mathcal{F}$ is the family of all minimal (w.r.t. inclusion) transversals of $\mathcal{H}$ (see C. Berge's book *Hypergraphs*, North-Holland 1989, for a detailed study of the transversal hypergraph).

The following problem is equivalent to **SIMUNSAT**:

**Transversal Hypergraph (TRANS-HYP)**

INSTANCE: Two hypergraphs $\mathcal{G} = (V, \mathcal{E}_\infty)$ and $\mathcal{H} = (V, \mathcal{E}_\in)$ on vertices $V = \{v_1, ..., v_n\}$.

QUESTION: Does $\mathcal{H} = Tr(\mathcal{G})$ hold?

This problem has immediate applications e.g. in Boolean logic and in the context of model based diagnosis.

Equivalent to this problem is the positive dualization problem:

**Positive Dualization (POS-DUAL)**

**INSTANCE:** Positive Boolean CNFs $\mathcal{E}$ and $\mathcal{F}$.

**QUESTION:** Does $\mathcal{F}$ represent the dual of the function represented by $\mathcal{E}$?

A variant of this problem is the one where $\mathcal{E}$ is a positive CNF and $\mathcal{F}$ is a positive DNF, and we ask whether $\mathcal{E}$ and $\mathcal{F}$ represent the same Boolean function. Fredman and Khachiyan, On the Complexity of Dualization of Monotone Disjunctive Normal Forms, Technical Report LCS-TR-225, Dept. of Computer Science, Rutgers University, 1994, have shown that **POS-DUAL** can be solved in quasi-polynomial time, i.e., in time $O(m^{O(\log(m))})$ where $m$ is the size of the input. See also Bioch and Ibaraki, Complexity of dualization and identification of positive boolean functions, Information and Computation, 123 (1995).

Next we consider problems equivalent to **SIMUNSAT** from database theory.

**FD-Relation Equivalence**

INSTANCE: A relation instance $\mathcal{R}$ (collection of tuples) and a set $F$ of functional dependencies in BCNF, both on a set of attributes $U$.

QUESTION: Is $\mathcal{R}$ an Armstrong relation for $F$, i.e., do on $\mathcal{R}$ hold exactly the dependencies represented by $F$?

A set $F$ of functional dependencies on a set of attributes $U$ is in *Boyce-Codd Normal Form* (BCNF), if in every nontrivial functional dependency $X \rightarrow Y$ logically implied by $F$ the left hand side $X$ is a minimal key, i.e., a wrt. inclusion minimal subset $X$ of attributes that determines all other attributes in $U$.

Also a variant of the key problem is equivalent to **SIMUNSAT**:

**Additional Key** (for relation instances)

INSTANCE: A relation instance $\mathcal{R}$ on attributes $U$, a set $K$ of minimal keys for $\mathcal{R}$.

QUESTION: Is there a minimal key for $\mathcal{R}$ (i.e., the set of functional dependencies $F$ that hold on $R$) which is not contained in $K$?

Notice that the key problem for relation *schemes* (instead of relation *instances*), where the input is a set $F$ of functional dependencies on $U$, is polynomial.

We nect describe a problem from the area of reliability and distributed computing which is equivalent to **SIMUNSAT**.

A coterie $\mathcal{C}$ is, using hypergraph terminology, an intersecting hypergraph.

A coterie $\mathcal{C}'$ *dominates* a coterie $\mathcal{C}$ if they are different coteries on the same set of vertices and for every edge $E \in \mathcal{C}$ there is an edge $F \in \mathcal{C}'$ such that $F \subseteq E$; a coterie is non-dominated (ND) if there is no coterie $\mathcal{C}'$ that dominates $\mathcal{C}$. Nondominated coteries have been introduced by Garcia-Molina & Barbara, How to assign votes in a distributed system, *JACM* 32 (1985); they have been studied extensively e.g. by Ibaraki and Kameda, A theory of coteries: Mutual exclusion in distributed systems, IEEE Trans. on Parallel and Distributed Systems, 4 (1993).

The following problem is equivalent to **SIMUNSAT**:

**Nondominated Coterie (ND-COTERIE)**

INSTANCE: A coterie $\mathcal{C}$.

QUESTION: Is $\mathcal{C}$ nondominated?

It is known that nondominatedness coincides with the property that $\mathcal{C}$ is a self-transversal hypergraph, i.e., a hypergraph $\mathcal{H}$ is identical to its transversal hypergraph. This yields the following problem equivalent to **SIMUNSAT**, which is a restriction of **TRANS-HYP**:

**SELF-TRANSVERSALITY (SELF-TRANS)**

INSTANCE: A hypergraph $\mathcal{H}$.

QUESTION: Is $\mathcal{H}$ self-transversal, i.e., does $\mathcal{H} = Tr(\mathcal{H})$ hold ?

There are other problems that are equivalent to **SIMUNSAT**, which have been recently identified in the area of propositional knowledge representation; see the papers by Khardon, Translating between Horn Representations and their Characteristic Models, *J. Artificial Intelligence Research* **3**, 1995, and D. Kavvadias, Ch. Papadimitriou and M. Sideri, On Horn Envelopes and Hypergraph Tranversals, ISAAC-93 Proceedings, LNCS 762.

A polynomial time algorithm for any of the problems above would be highly appreciated.

---

# John Franco

Does there exist an algorithm for verifying that a random 3-SAT formula is unsatisfiable and that has polynomial time complexity with probability tending to 1 when $m/n$

is any constant greater than 5? or even when $m/n$ grows but $\lim_{m,n\to\infty} m/n^{8/7} = 0$? or even when $\lim_{m,n\to\infty} m/n^{2-\epsilon} = c$ for some small $\epsilon > 0$ and constant $c$?

---

## Oliver Kullman

Let $\mathcal{F}$ be a CNF formula. Define $\mathcal{F}^*$ to be an *extension* for $\mathcal{F}$ if for all subsets $\mathcal{S}$ of clauses of $\mathcal{F}$ that are satisfiable, $\mathcal{S} \cup \mathcal{F}^*$ is also satisfiable. If $\mathcal{F}$ is not satisfiable, define $Comp_E(\mathcal{F})$ to be the minimum value of $Comp_{\text{RES}}(\mathcal{F} \cup \mathcal{F}^*)$ such that $\mathcal{F}^*$ is an extension for $\mathcal{F}$; define $Comp_{\text{RES}}(\mathcal{G})$ to be the minimum $n$ such that $\mathcal{G} \vdash^n_{\text{RES}} 1$.

Is $Comp_E(\mathcal{F})$ polynomially bounded?

Let $\mathcal{D} = \{\mathcal{F} \in CLS | \forall_{C_i, C_j \in \mathcal{F}, i \neq j} C_i \cap \bar{C}_j \neq \emptyset\}$.

Is $Comp_{\text{RES}}(\mathcal{F})_{\mathcal{F} \in \mathcal{D}}$ polynomially bounded?

Consider any feasible notion of redundant clauses with respect to sat-equivalence (like blocked clauses). Which is better: adding such clauses or deleting them?

---

## Ewald Speckenmeyer

Let $\mathcal{I}(n,k)^m$ be the set of all CNF formulas containing $n$ variables and $m$ clauses, where each clause has exactly $k$ literals. Let $f(n,m,k)$ be the minimum number of clauses which have to be removed from formulas in $\mathcal{I}(n,k)^m$ in order to make them satisfiable.

What is $f(n,m,k)$?

By experiment, $f(500, 5000, 3)$ is approximately 162.

---

## Alasdair Urquhart - Multilated Chessboards

Consider the following well known puzzle: given a chessboard with two diagonally opposite squares removed, can you cover it with dominoes each of which can cover two adjacent squares? The answer is of course negative because any domino must cover a black square and a white square, and diagonally opposite squares have the same colour.

This puzzle, generalized to a $2n$ by $2n$ board, can be formalized as an unsatisfiable set of clauses. We use variables $D_{xy}$ to symbolize: "Square $x$ and square $y$ are covered by a domino." For each square $x$, we have a positive disjunction saying that it must be covered by a domino that also covers an adjacent square, and a set of negative disjunctions saying that $x$ cannot be covered by more than one domino. This set of clauses has size quadratic in $n$. Problem: show that these sets of clauses require super-polynomial size resolution refutations.

**References:**

"Automation of Reasoning," edited by J. Siekmann and G. Wrightson, Vol. 2, pp. 157-8, Springer-Verlag 1983.

"Short proofs for tricky formulas," by B. Krishnamurthy, *Acta Informatica*, Vol. 22 (1985), pp. 253-275.

---

# Allen Van Gelder

Determine the complexity of Davis-Putnam style algorithms that are enhanced by 2-closure or similar processing of binary clauses. Where do subsumptions make an impact?

---

The following two people have submitted open problems but existing descriptions have not yet been updated: Paul Purdom, Cosimo Spera.

# **APPENDIX B** - Announcement

Ewald Speckenmeyer
Universität zu Köln
Institut für Informatik
49 221 470-5377/5378
FAX:49 221 470-5387
esp@informatik.uni-koeln.de

John Franco
University of Cincinnati
Computer Science
513-556-1817
FAX:513-556-3417
franco@franco.csm.cu.edu

Hans Kleine Büning
Universität-Gesamthochschule Paderborn
Fachbereich Informatik
49 5251 60-3360
FAX: 49 5251 60-3338
kbcsl@uni-paderborn.de

Giorgio Gallo
Universita di Pisa
Dipartimento Informatica
39 50 887-214
FAX: 39 50 887-226
gallo@di.unipi.it

# Workshop On The Satisfiability Problem

April 29 - May 3, 1996
Siena, Italy

The opening session begins on April 29th at 9 a.m. and the closing session ends on May 3rd at 1 p.m.

---

## Contact Address

Mrs. Neugebauer
Universität zu Köln
Institut für Informatik
Pohligstr. 1
D-50969 Köln
Germany

sat-workshop@informatik.uni-koeln.de

---

# Proceedings

Selected papers from the talks to be presented at the workshop will be published in a special issue of em Discrete Applied Mathematics (J. Franco, G. Gallo, H. Kleine Büning, E. Speckenmeyer, eds.) after a refereeing process following the high standards of the journal.

Postscript files, dvi files or hard copy will be accepted for contribution with the first two preferred.

Electronic versions should be emailed to **sat-workshop@informatik.uni-koeln.de**. Hard copy versions should be sent to Mrs. Neugebauer at the contact address above. Submissions to the special issue should be ready before

**April 17, 1996**

In order to speed up the refereeing process submission of your paper as early as possible is highly recommended!

---

## Abstracts

A Technical Report on the workshop will be distributed at Siena. All participants are expected to send extended abstracts or full papers of no more than 10 pages to E. Speckenmeyer at the contact address above before

**March 25, 1996**

Electronic version in dvi or postscript form are preferred and should be sent to **sat-workshop@informatik.uni-koeln.de**

---

## Accommodation/Cost

All the rooms in the conference center Certosa have been reserved for the time of the workshop:

There are two conference rooms with projectors.

20 single, 11 double rooms, and 4 double rooms with one queen bed.

**Price list** (Italian lire, Lit) in 1995:

| | single | double |
|---|---|---|
| full | 84.000 | 138.000 |
| half | 75.000 | 120.000 |
| acc. | 70.000 | 110.000 |

Prices may increase by 5%, where full accommodation includes breakfast, lunch and dinner; half accommodation includes breakfast, lunch or dinner; accommodation includes breakfast, only.

Current exchange rate: $1 US is approximately 1600 Lit., 1 DM is approximately 1114 Lit.

Reservations should be made via **spera@sivax.unisi.it** before March 1, 1996.

In case you can't attend the workshop for certain reasons, please send a short email to **sat-workshop@informatik.uni-koeln.de**.

---

# Address of Conference Center

Certosa di Pontignano
Universita' degli Studi di Siena
53010 Pontignano, Siena

tel:+39-577-356851 / fax:+39-577-356669

---

# Travel Information

We intend to organize a bus transfer from the Siena train station to the Certosa. In order to arrange the transport, please let us know, whether you plan to use the bus, and when you will arrive and leave. If desired we will help you to find out the best connection from your city of arrival in Italy to Siena.

### Getting to Siena

Closest airports are Pisa and Firenze.

**From Pisa** take the train from the airport to Empoli then change to get the connecting train to Siena (approximate time 1 h:45 min - including the waiting time in Empoli).

**From Firenze** take a bus or taxi from the airport to Firenze train station "Santa Maria Novella" and get a train to Siena there (the train takes 1 h:35 min).

### Train: Firenze - Siena (extract)

| Depart Firenze | Arrival Empoli | Depart Empoli | Arrival Siena |
|---|---|---|---|
| 9.20 h | 9.55 h | 9.58 h | 10.50 h |
| 11.20 h | 11.55 h | 11.59 h | 13.05 h |
| 14.13 h | 14.42 h | 14.43 h | 15.33 h |
| 16.05 h | 16.31 h | 16.36 h | 17.32 h |

Alternatively: Go from the airport to the bus station "Sita", which is located near the train station and from there take a bus "Rapida" to Siena San Domenico, which takes 1 h:15 min. (The second connection is recommanded).

**From Milano**

Take an InterCity train to Firenze and the connection to Siena (InterCity train requires extra charge).

**Train: Milano - Siena (extract)**

| | Depart Milano | Arrival Firenze | Depart Firenze | Arrival Siena |
|---|---|---|---|---|
| IC | 8.00 h | 10.51 h | 11.20 h | 13.05 h |
| IC | 10.00 h | 12.51 h | 13.25 h | 15.05 h |
| IC | 14.00 h | 16.51 h | 17.10 h | 18.33 h |
| IC | 15.00 h | 17.51 h | 18.20 h | 19.55 h |

There are also trains at 11.00 a.m. and 12.00 a.m. You can also take a bus.

**From Rome**

Take a train to Chiusi - Chianciano Terme and then the connection to Siena. The following contains an extract of the schedule:

**Train: Rome - Siena (extract)**

| | Depart Rom Termini | Arrival Chiusi | Depart Chiusi | Arrival Siena |
|---|---|---|---|---|
| IC | 8.30 h | 10.11 h | 10.40 h | 11.50 h |
| IC | 9.20 h | 10.34 h | 10.40 h | 11.50 h |
| IC | 13.50 h | 15.21 h | 15.26 h | 16.45 h |
| IC | 15.20 h | 16.43 h | 16.50 h | 18.08 h |
| | 17.50 h | 19.13 h | 19.18 h | 20.30 h |

**From Siena to Pontignano**

There is a regular Bus from Siena centre to Pontignano.

**Bus: Siena centre - Pontignano**

| | Depart Siena (piazza Gramsci) | Arrival Pontignano | Days |
|---|---|---|---|
| Bus | 06.50 h | 07.10 h | (Mon-Fri) |
| Bus | 08.00 h | 08.20 h | (Mon-Fri) |
| Bus | 09.00 h | 09.20 h | (Wed) |
| Bus | 13.10 h | 13.30 h | (Mon-Fri) |
| Bus | 13.45 h | 14.05 h | (Mon-Fri) |

**Bus: Siena centre - Pontignano (cont)**

| | Depart Siena (piazza Gramsci) | Arrival Pontignano | Days |
|---|---|---|---|
| Bus | 14.30 h | 14.50 h | (Sun) |
| Bus | 15.30 h | 15.50 h | (Mon-Fri) |
| Bus | 18.35 h | 18.55 h | (Mon-Fri) |
| Bus | 19.30 h | 19.50 h | (Sun) |
| Bus | 20.10 h | 20.30 h | (Mon-Fri) |

**By car from Milano**

Take the Autostrada A 1 to Firenze Certosa and then follow the signs to Siena.

**By car from Rome**

Take the Autostrada (north) to Firenze, leave the Autostrada in Valdichiana. Then follow the signs to Siena.

# Participants, SAT Workshop Siena 1996

| Name | Affilation | email-address |
|---|---|---|
| Boros, Endre | USA, New Brunswick | boros@rutcor.rutgers.edu |
| Dubois, Olivier | France, Paris | dubois@laforia.ibp.fr |
| Eiter, Thomas | Austria, A-1040 Vienna | eiter@dbai.tuwien.ac.at |
| Franco, John | USA, Cincinnati/OH | franco@franco.csm.uc.edu |
| Gallo, Giorgio | Italy, Pisa | gallo@di.unipi.it |
| Goerdt, Andreas | Germany, Chemnitz-Zwickau | goerdt@informatik.tu-chemnitz.de |
| Grandjean, Etienne | France, Caen | grandjean@univ-caen.fr |
| Gu, Jun | Hongkong, Kowloon | gu@cs.ust.hk |
| Hammer, Peter L. | USA, Piscataway | hammer@rutcor.rutgers.edu |
| Heusch, Peter | Germany, Köln | heusch@informatik.uni-koeln.de |
| Jaumard, Brigitte | Canada, Montreal | brigitt@crt.umontreal.ca |
| Kleine Büning, Hans | Germany, Paderborn | kbcsl@uni-paderborn.de |
| Lettmann, Theo | Germany, Paderborn | lettmann@uni-paderborn.de |
| van Maaren, H. | The Netherlands, Delft | H.vanMaaren@twi.tudelft.nl |
| Mitchell, David | Canada, Toronto | mitchell@cs.toronto.edu |
| Mundici, Daniele | Italy, Milano | mundici@imiucca.csi.unimi.it |
| Plaisted, David A. | USA, Chapel Hill | plaisted@cs.unc.edu |
| Pretolani, D. | Italy, Pisa | pretola@di.unipi.it |
| Purdom, Paul W. | USA, Bloomington | pwp@cs.indiana.edu |
| Sassano, Antonio | Italy, Roma | sassano@peano.dis.uniroma1.it |
| Schiermeyer, Ingo | Germany, Cottbus | schierme@math.tu-cottbus.de |
| Schlipf, John | USA, Cincinnati | schlipf@jschlipf.csm.uc.edu |
| Selman, B. | USA, Murray Hill | selman@research.att.com |
| Speckenmeyer, Ewald | Germany, Köln | esp@informatik.uni-koeln.de |
| Spera, Cosimo | Italy, Siena | spera@sivax.unisi.it |
| Urquhart, Alasdair | Canada, Ontario | urquhart@cs.toronto.edu |
| Van Gelder, Allen | USA, Santa Cruz | avg@cse.ucsc.edu |
| Wang, Jinchang | USA, Lenexa | wang@griffon.mwsc.edu |

---

**Organizers**

John Franco, Cincinnati, USA
Giorgio Gallo, Pisa, Italy
Hans Kleine Büning, Paderborn, Germany
Ewald Speckenmeyer, Köln, Germany
Cosimo Spera, Siena, Italy

---

# **APPENDIX C** - Program

# Workshop On The Satisfiability Problem[32]

## April 29 - May 3, 1996

## Siena, Italy

### Program

Monday, 29.04.96

| | |
|---|---|
| 09.45 - 10.00 | **Opening Address** |
| 10.00 - 10.45 | Thomas Eiter, Toshihide Ibaraki, Kazuhisa Makino<br>On Satisfiability of Partially Defined Double and Bidual Horn Functions |
| 10.45 - 11.15 | **Coffee Break** |
| 11.15 - 12.00 | Paul Purdom<br>Probe Order Backtracking |
| 12.00 - 12.45 | Hans Kleine Büning, Theodor Lettmenn<br>Resolution Remains Hard Under Equivalence |
| 12.45 - 14.30 | **Lunch** |
| 14.30 - 15.15 | Alasdair Urquhart<br>The Symmetry Rule in Propositional Logic |
| 20.00 | **Dinner** |

Tuesday, 30.04.96

| | |
|---|---|
| 9.15 - 10.00 | Peter Heusch, Marc-Andre Lemburg, Ewald Speckenmeyer<br>Complexity Results of Subclasses of the Pure Implicational Calculus |
| 10.00 - 10.45 | John Franco, Judy Goldsmith, John Schlipf,<br>Ewald Speckenmeyer, R. Swaminathan<br>An Algorithm for the class of Pure Implicational Formulas |
| 10.45 - 11.15 | **Coffee Break** |
| 11.15 - 12.00 | Jinchang Wang<br>Testing Propositional Satisfiability by Using Binary Trees |
| 12.00 - 12.45 | Ingo Schiermeyer<br>Pure Literal Look Ahead: An $O(1.479^n)$ 3-Satisfiability Algorithm |
| 12.45 - 14.30 | **Lunch** |
| 14.30 - 15.15 | Allen Van Gelder, Fumiaki Kamiya<br>Lemma and Cut Strategies for Two-Sided Propositional Resolution |
| 15.15 - 16.00 | Oliver Kullmann<br>Blocked Clauses; Their Use for SAT Decision, And An Analysis of Their Strength |
| 20.00 | **Dinner** |

Wednesday, 01.05.96

| | |
|---|---|
| 09.15 - 10.00 | Hans Kleine Büning, Theodor Lettmann<br>Closure Under Replacements Versus Run Time of the Davis-Putnam Algorithms and Distribution of Satisfiable Formulas |
| 10.00 - 10.45 | Hans Van Maaren<br>Discriminative Properties of the Smooth Convex Quadratic Approximation of a 3-SAT Problem |
| 10.45 - 11.15 | **Coffee Break** |
| 11.15 - 12.00 | Jun Gu<br>Multi-SAT Algorithm |
| 12.00 - 12.45 | Marco Protassi<br>MAX-SAT and the Class APX |

Thursday, 02.05.96

| | |
|---|---|
| 09.15 - 10.00 | Paolo Nobili, Antonio Sassano<br>Strengthening Lagrangian Bounds for the MAX-SAT Problem |
| 10.00 - 10.45 | Giorgio Gallo, C. Gentile, D. Pretolani<br>MAX Horn SAT and Directed Hypergraphs: Algorithmic Enhancements and Easy Cases |
| 10.45 - 11.15 | **Coffee Break** |
| 11.15 - 12.00 | Roberto Battiti, Marco Protassi<br>Reactive Search: A History-Based Heuristic for MAX-SAT |
| 12.00 - 12.45 | Endre Boros<br>On Maximum Renamable Horn Sub-CNFs |
| 12.45 - 14.30 | **Lunch** |
| 14.30 - 15.15 | Andreas Goerdt<br>Probability of Satisfiability of Random 2-SAT Instances with Quantification |
| 15.15 - 16.00 | David Mitchell<br>Toward an Adequate SAT Algorithm |

Friday, 03.05.96

| | |
|---|---|
| 10.00 - 11.00 | Open Problems I |
| 11.00 - 11.30 | **Coffee Break** |
| 11.30 - 12.30 | **Open Problems II** |