Horn upper bounds and renaming*

Marina Langlois, Robert H. Sloan, and György Turán**

University of Illinois at Chicago
Chicago, IL 60607, USA
{mirodo1, sloan, gyt}@uic.edu

Abstract. We consider the problem of computing tractable approximations to CNF formulas, extending the approach of Selman and Kautz to compute the Horn-LUB to involve renaming of variables. Negative results are given for the quality of approximation in this extended version. On the other hand, experiments for random 3-CNF show that the new algorithms improve both running time and approximation quality. The output sizes and approximation errors exhibit a 'Horn bump' phenomenon: unimodal patterns are observed with maxima in some intermediate range of densities. We also present the results of experiments generating pseudo-random satisfying assignments for Horn formulas.

1 Introduction

A general formulation of the reasoning problem in propositional logic is to decide if a clause C is implied by a CNF expression φ. Here φ is often viewed as a fixed *knowledge base*, and it is assumed that a large number of *queries* C have to be answered for the same knowledge base. Therefore, it may be useful to preprocess φ into a more tractable form, resulting in a new knowledge base which may be only approximately equivalent to the original one. This approach, called *knowledge compilation*, goes back to the seminal work of Selman and Kautz [22] (see also [5, 7, 24]).

Selman and Kautz suggested considering *Horn formulas* approximating the initial knowledge base from above and below, and using these formulas to answer the queries. In particular, they gave an algorithm (outlined in Section 4) computing a *Horn least upper bound (Horn-LUB)* of φ which is equivalent to the conjunction of all its Horn prime implicates.[1] The set of truth assignments satisfying the Horn-LUB of φ has a natural combinatorial characterization which suggests that this notion may be of interest in itself. The intersection of two truth assignments is obtained by taking their componentwise conjunction. In

* This material is based upon work supported by the National Science Foundation under grant CCF-0431059. A preliminary version of this work appeared in [14].

** Also affiliated with Hungarian Academy of Sciences and University of Szeged, Research Group on Artificial Intelligence, Szeged, Hungary.

[1] We omit the definition of Horn *greatest lower bounds*, as those will not be discussed in this paper.

other words, the intersection is the greatest lower bound of the two truth assignments in the componentwise partial ordering of the hypercube. The set of truth assignments satisfying the Horn-LUB of φ, then, can be obtained as the *closure under intersections of the set of satisfying truth assignments of* φ.

Queries to Horn formulas can be answered efficiently, but the approach can have the following drawbacks: it may be inefficient as the resulting Horn upper bound may be large, and it may fail to answer certain queries (those implied by the lower bound, but not implied by the upper bound). Indeed, such theoretical negative results on the worst-case performance of the approach have been obtained in Selman and Kautz [22] and del Val [8]. Another interesting question is the performance of the algorithms on random examples. Initial experiments in this direction were performed by Kautz and Selman [12]. Boufkhad [4] gives the results of experiments for Horn lower bounds using an extension to renamable Horn formulas. The main test examples were random 3-CNF formulas with density around 4.2, which are well known to be hard for satisfiability algorithms.[2] In other related work, Van Maaren and van Norden [25] considered the connection between the efficiency of satisfiability algorithms and the size of a largest renamable sub-CNF for random 3-CNF.

In this paper we introduce new variants of the Horn-LUB algorithm and present theoretical and experimental results on their performance. The new variants involve the construction of a renaming of some variables, i.e., switching some variables and their complements. Thereby one hopes to bring the original formula closer to being a Horn formula, possibly resulting in a smaller Horn-LUB with better approximation quality. We use an algorithm of Boros [3] to find a large renamable subformula of the original knowledge base. Another possibility we consider is the use of resolvents of bounded size only. This is expected to speed up the algorithm and decrease the size of the Horn upper bound at the price of decreasing approximation quality. Thus it may be of interest to explore the trade-offs to find an optimal size bound. The combinatorial interpretation of the Horn-LUB mentioned above carries over to the case of renaming. Informally, a renaming corresponds to a reorientation of the hypercube, by choosing an arbitrary vector as the 'bottom' of the hypercube instead of the all 0's vector. In order to obtain the Horn-LUB after the renaming, intersections have to be taken with respect to this new orientation.

The theoretical results show similar worst-case behavior as in the original case. In particular, 3-CNF expressions are presented with only a polynomial number of truth assignments such that for every renaming of the variables, the Horn-LUB obtained after the renaming has superpolynomially many satisfying truth assignments. We also construct a polynomial size CNF expression such that for every renaming, only a superpolynomially small fraction of the prime implicates are Horn, and therefore most of the prime implicate queries are answered incorrectly by the renamed Horn-LUB.

[2] Kautz and Selman also considered a class of planning problems, and Boufkhad also considered 4-CNF formulas.

In the second part of the paper we present experiments indicating that, on the other hand, the new algorithms give improvements in both efficiency and approximation quality. We have compared Selman and Kautz's original algorithm and three variants. The best one appears to be the one that uses both renaming and bounded size resolvents. As the performance of the algorithms is evaluated by exhaustive testing over all truth assignments, we have chosen to run the experiments for 20 variables. In order to consider a larger number of variables, it would be necessary to be able to efficiently sample random satisfying truth assignments of a Horn formula. It appears to be an open question whether this is possible. We have run experiments with some natural candidates for such a sampling algorithm. The results of these experiments are also included in the paper.

In contrast to previous work, in the experiments we have considered 3-CNF formulas of *different densities*, in particular, for densities well below the critical range. Here we have observed an interesting phenomenon—the *Horn bump*: the performance of each algorithm is the worst in an intermediate range of densities. This phenomenon may be of interest for the study of the evolution of random 3-CNF formulas [18, 19, 21].

2 Preliminaries

A clause is a disjunction of literals; a clause is Horn (resp., definite, negative) if it contains at most one (resp., exactly one, no) unnegated literal. A CNF is a conjunction of clauses; it is a 3-CNF if each clause contains exactly 3 literals. A clause C is an *implicate* of a CNF expression φ if every *truth assignment* or *vector* in $\{0,1\}^n$ satisfying φ also satisfies C; it is a *prime* implicate if none of its sub-clauses is an implicate. An n-variable *random 3-CNF* formula of *density* α is obtained by selecting $\alpha \cdot n$ clauses of size 3, selecting each clause from the uniform distribution over all such clauses. A *Horn formula* or *Horn-CNF* is a conjunction of Horn clauses.

The set of satisfying truth assignments of a formula φ is denoted by $T(\varphi)$. The weight of a 0-1 vector is number of its 1 components. The *intersection* of vectors $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n) \in \{0,1\}^n$ is $(x_1 \wedge y_1, \ldots, x_n \wedge y_n)$. A Boolean function can be described by a Horn formula if and only if its set of satisfying truth assignments is closed under intersection [10, 17]. The *(ordinary) Horn closure* $\mathcal{H}(S)$ of any set S of truth assignments is the smallest intersection-closed set of truth assignments containing S.

A Horn least upper bound of φ (Horn-LUB(φ)) is any conjunction of Horn clauses logically equivalent to the conjunction of all Horn prime implicates of φ. The set of satisfying truth assignments of Horn-LUB(φ) is $\mathcal{H}(T(\varphi))$, the Horn closure of $T(\varphi)$. Selman and Kautz [22] give an algorithm for computing a Horn-LUB(φ).

Renaming a variable x in a CNF is the operation of simultaneously *switching* every occurrence of x to $\bar{x}$ and of $\bar{x}$ to x. A *renaming (function) with respect to vector* $d \in \{0,1\}^n$, denoted by $\mathcal{R}_d$, maps a CNF formula φ to $\mathcal{R}_d(\varphi)$, obtained by switching every pair of literals x_i and $\bar{x}_i$ such that $d_i = 1$. The following easily verified proposition shows that the operation on truth assignments corresponding to renaming w.r.t. vector d is taking the exclusive or with d, and one can use renaming to solve the reasoning problem formulated in the introduction.

Proposition 1. a) *A truth assignment a satisfies CNF φ iff the truth assignment $a \oplus d$ satisfies $\mathcal{R}_d(\varphi)$.*

b) *For any CNF φ, clause C, and vector d, we have $\varphi \models C$ if and only if $\mathcal{R}_d(\varphi) \models \mathcal{R}_d(C)$.*

A CNF φ is *Horn renamable* if $\mathcal{R}_d(\varphi)$ is a Horn formula for some vector d. It can be decided in polynomial time if a CNF is Horn renamable [1,16], but finding a largest Horn renamable sub-CNF of a given CNF is NP-hard [6]. Boros [3] gave an approximation algorithm for finding a large Horn renamable sub-CNF in an arbitrary CNF in linear time. Given a direction $d \in \{0,1\}^n$, the *d-Horn closure of a set S of truth assignments* is

$$\mathcal{H}_d(S) = \mathcal{H}(\{a \oplus d : a \in S\}).$$

With an abuse of notation, we refer to $\mathcal{H}_d(T(\varphi)) = \mathcal{H}(T(\mathcal{R}_d(\varphi)))$ as the *d-Horn closure* of φ.

3 Negative results for Horn upper bounds with renaming

In this section we present negative results for Horn closures and Horn-LUB with renaming, analogous to those for the ordinary Horn closure and Horn-LUB.

Theorem 1. *There are 3-CNF formulas φ with a polynomial number of satisfying truth assignments such that for every direction d, the size of the d-Horn closure of φ is superpolynomial.*

Proof. The construction uses the following lemma.

Lemma 1. *There is a set $S \subseteq \{0,1\}^m$ with $|S| = 2m$ such that for every direction $d \in \{0,1\}^m$ it holds that*

$$|\mathcal{H}_d(S)| \geq 2^{\lceil m/2 \rceil}.$$

Proof. Let S be the set of vectors of weight 1 and $(m-1)$ and let $d \in \{0,1\}^m$ be any direction. Then d has at least $\lceil m/2 \rceil$ 0's or $\lceil m/2 \rceil$ 1's. Assume w.l.o.g. that the first $\lceil m/2 \rceil$ components of d are 0 (resp., 1). Consider those vectors from S which have a single 0 (resp. 1) in one of the first $\lceil m/2 \rceil$ components. All possible intersections of these vectors (resp., the complements of these vectors) are contained in the d-Horn closure of S. Thus all possible vectors on the first $\lceil m/2 \rceil$ components occur in the d-Horn closure and the bound of the lemma follows. □

Now consider the 4-CNF φ formed by taking the conjunction of all possible clauses of size 4 containing two unnegated and two negated literals over m variables. The vectors satisfying this formula are those in the set S in the proof of Lemma 1 plus the all 0's and the all 1's vectors. Hence by Lemma 1, φ's Horn closure with respect to any direction has size at least $2^{\lceil m/2 \rceil}$.

In order to obtain a 3-CNF ψ, introduce a new variable z for each clause $(a \vee b \vee c \vee d)$ in φ, and replace the clause by five new clauses: $(a \vee b \vee \bar{z}), (c \vee d \vee z), (\bar{a} \vee \bar{b} \vee z), (\bar{a} \vee b \vee z)$ and $(a \vee \bar{b} \vee z)$. It follows by a standard argument (omitted for brevity) that ψ has the same number of satisfying truth assignments as φ, and every truth assignment of φ has a unique extension to a satisfying truth assignment of ψ. Hence the Horn closure of ψ in any direction has size at least $2^{\lceil m/2 \rceil}$. Thus ψ has $n = \Theta(m^4)$ variables, $\Theta(m)$ satisfying truth assignments and its Horn closure in every direction has size at least $2^{\lceil m/2 \rceil}$, so the theorem follows. □

It may be of interest to note that the bound of Lemma 1 is fairly tight.

Theorem 2. *For every polynomial p and every $\epsilon > 0$, for all sufficiently large m, for every set S of at most $p(m)$ binary vectors of length m, there exists a direction d such that the size of the d-Horn closure of S is at most $2^{\frac{m}{2}(1+\epsilon)}$.*

Proof. We show that a randomly chosen direction $d \in \{0,1\}^m$ has nonzero probability of having the desired property. For every vector $a \in S$, the probability that $a \oplus d$ has more than $\frac{m}{2}(1+\frac{\epsilon}{2})$ 1's is at most $e^{-\epsilon^2 m/8}$ using a Chernoff bound [13, Additive Form, page 190]. If m is sufficiently large then $p(m)e^{-\epsilon^2 m/8} < 1$. In this case there is a direction d such that $a \oplus d$ has at most $\frac{m}{2}(1+\frac{\epsilon}{2})$ 1's for every $a \in S$. Every vector in the d-Horn closure of S is below one of the vectors $a \oplus d$. Hence the size of the d-closure is at most $p(m)2^{\frac{m}{2}(1+\frac{\epsilon}{2})}$, which is less than $2^{\frac{m}{2}(1+\epsilon)}$ for all sufficiently large m. □

The following result shows the existence of CNF formulas for which the Horn-LUB in every direction d gives an incorrect answer to a large fraction of the prime implicate queries. The construction is based on a construction of Levin [15] of a DNF formula with a bounded number of terms, having the maximal number of prime implicants. In [23], we showed that all bounded term DNF with the maximal number of prime implicants can be obtained as a natural generalization of this example.

Theorem 3. *There are polynomial size CNF formulas φ such that for every direction d, the ratio of the number of non-Horn and Horn prime implicates of $\mathcal{R}_d(\varphi)$ is superpolynomial.*

Proof. To construct φ, we begin with a complete binary tree of the height k and put $n = 2^k$. The variables of φ are $x_1, \ldots, x_{n-1}$ and $y_1, \ldots, y_n$. Each internal node of the tree is labeled with a distinct x variable and the ith leaf is labeled with $\bar{y}_i$. The formula φ has n clauses, one for each leaf. The clause corresponding to a leaf is the disjunction of all the variables on the root-to-leaf path to the leaf, with each x variable being negated if and only if the path went left when leaving that node. Thus the depth-1 tree pictured in Figure 1 corresponds to $(\bar{x}_1 \vee \bar{y}_1) \wedge (x_1 \vee \bar{y}_2)$.

The formula φ has a distinct prime implicate for each of the $2^n - 1$ nonempty subsets of the leaves [15, 23]. The prime implicate corresponding to a particular subset S of leaves is the disjunction of x variables corresponding to any inner node such that *exactly one* of the two subtrees of the node contains a leaf in S, and the negated y variables corresponding to the leaves in S. An x variable in the prime implicate is negated iff its left subtree is the one containing leaves in S. Thus, for example, the formula corresponding to the tree in Figure 1 has three prime implicates, one for each nonempty subset of the two leaves. For $\{\bar{y}_1\}$ we have $(\bar{x}_1 \vee \bar{y}_1)$; for $\{\bar{y}_2\}$ we have $(x_1 \vee \bar{y}_2)$; for $\{\bar{y}_1, \bar{y}_2\}$ we have $(\bar{y}_1 \vee \bar{y}_2)$.

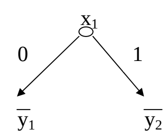

Fig. 1. Tree of depth 1

We give an upper bound for the number of Horn prime implicates under any renaming d. Notice that by symmetry, renaming internal nodes does not change the number of Horn or non-Horn prime implicates. At the leaves, making all the y's negated maximizes the number of the prime implicates that are Horn. Thus it is in fact sufficient to estimate the number of Horn prime implicates of the original formula.

Let H_k (resp., N_k, D_k) be the number of Horn (resp., negative, definite) prime implicates of the formula built from a binary tree of height k. Then $H_k = N_k + D_k$. The numbers H_k satisfy the following recurrence: $H_1 = 3$ and

$$H_k = H_{k-1} + N_{k-1} + H_{k-1} \cdot N_{k-1} + N_{k-1} \cdot D_{k-1} \ . \tag{1}$$

Here the first item is the number of Horn implicates of the left subtree. The second term is the number of negative Horn implicates of the right subtree: by adding the unnegated variable from the root, those correspond to definite Horn implicates. The third (resp, the fourth) term corresponds to prime implicates obtained from an arbitrary Horn (resp., a negative) prime implicate of the left subtree and a negative (resp., a definite) one form the right subtree. Note that two definite prime implicates from the two subtrees will form a non-Horn prime implicate. In order to use (1) to get an upper bound on H_k, we must bound N_k. Similarly to (1), one can derive the following recurrence: $N_1 = 2$ and

$$N_k = (N_{k-1})^2 + N_{k-1}.$$

It can be shown that $N_k < 2^{\frac{11}{16}n}$ and $H_k \leq 3^{k-4} \cdot 2^{14} \cdot 2^{\frac{11}{16}n}$. □

4 Computational results

Selman and Kautz's original Horn-LUB algorithm for a set of clauses (i.e., a CNF) proceeds by repeatedly performing resolution steps between two clauses, at least one of which must be non-Horn. Any clauses in the set that are subsumed by the new resolvent are removed, and the new resolvent is added to the set. This process continues until it becomes impossible to find two clauses to resolve such that at least one is non-Horn and their resolvent is not subsumed by some clause already in the set.

In addition to Selman and Kautz's original Horn-LUB algorithm, we considered three other algorithms to compute Horn approximations, which are modifications of the original algorithm:

Renamed-Horn-LUB finds a renaming of the variables using a heuristic algorithm of [3], and then applies the Horn-LUB algorithm. Notice that we need only linear time to find a renaming.

4-Horn-UB works as the Horn-LUB algorithm, but only performs resolution steps that produce clauses of size at most 4.

Renamed-4-Horn-UB is the combination of the first two algorithms: it first performs a renaming, and then does those resolution steps that produce clauses of size at most 4.

It turns out that Renamed-4-Horn-UB gives the best performance, so we give a snapshot of its running time and of the size of its output versus that of the original Horn-LUB in Table 1. (All running times reported in this paper were measured on a Dell laptop with a 2.40 GHz CPU and 256MB RAM.)

Table 1. Mean running time in CPU seconds and number of clauses in the output for Horn-LUB and Renamed-4-Horn-UB on random 3-CNF formulas on 20 variables as a function of density α, averaged over 50 runs.

	Original LUB		Renamed-4	
α	Time	Size	Time	Size
1	0.96	96.1	0.00	28.2
2	50.49	1044.7	0.16	236.2
3	126.81	889.8	1.49	704.8
4	224.56	409.3	0.91	452.7

As mentioned in the introduction, we chose to make most of our measurements on formulas on $n = 20$ variables. We restricted n to 20 for two main reasons. First, the running time of the original Horn-LUB algorithm increased by roughly a factor of 5–10 for every additional two variables. So, while by using significantly greater computational resources we could have computed the original Horn-LUB for formulas with 25 variables, it would have been completely infeasible to do so for, say, 75 variables. (Selman and Kautz reported empirical data only on such things as the *unit clauses within the Horn-LUB*, which can be computed much more quickly by using a SAT solver, not by the Horn-LUB algorithm itself.) Second, in several cases we needed to perform exhaustive testing over all 2^n vectors, and this testing becomes impractical for values much above $n = 20$. (More discussion of this issue is given at the end of this section.)

We observe that the running time of Renamed-4-Horn-UB is significantly smaller than for Horn-LUB, and the size of the output formulas is smaller for Renamed-4-Horn-UB for density $\alpha \leq 3$, and modestly larger for density 4. The output sizes for both algorithms are unimodal as a function of CNF density. More detailed data show that the maximum size occurs around density 2.5.

As all these algorithms produce a conjunction of some implicates of the original formula φ, their output is implied by φ; that is, each algorithm's output has a one-sided error. The *relative error* of such an algorithm A on an input formula φ is measured by

$$r_A(\varphi) = \frac{|T(A(\varphi))| - |T(\varphi)|}{|T(\varphi)|},$$

where $A(\varphi)$ denotes the formula output by A on φ.

Figure 2 presents computational results for the relative errors of the four algorithms for different densities on 20 variables. Statistical values on all the figures are median, max and min values; the values at the ends of the white bars are 25% and 75%. The error curves are again unimodal, with maxima around density 2.4. Experiments for fewer variables show similar values of the maxima.

Of the two heuristics taken alone, renaming improves the relative error more dramatically than limiting clauses to size 4; notice that the relative errors were sufficiently different that the two parts of Figure 2 for the renaming heuristic use a different scale. The overall conclusion is that Renamed-4-Horn-UB is the best algorithm for 20 variables.[3] It is significantly faster than either Horn-LUB or Renamed-Horn-LUB, and it is even somewhat faster than 4-Horn-UB. Its output size is significantly smaller than those of Horn-LUB or Renamed-Horn-LUB, but larger than that of 4-Horn-UB. On the other hand, its relative error is only slightly worse than that of Renamed-Horn-LUB, which has the smallest relative error. Replacing the limit 4 on clause size with 3, or even using all implicates of size at most 3, results in a large increase in the relative error for densities below the satisfiability threshold.

It is to be expected, and it is supported by some experimental evidence, that as the number of variables increases, the limit on the clause size required for producing reasonable relative error will also increase.

Another way to evaluate the algorithms is to consider the number of queries that are answered incorrectly by their output. Notice that by Proposition 1, if we have used renaming, we can simply query the renamed clause. We will use the prime implicates of the original formula φ as our test set of clauses. The original Horn-LUB algorithm gives the correct answer for any Horn clause query, and

[3] Preliminary experiments show Renamed-4-Horn-UB also performing relatively well for up to at least 40 variables, but for 40 variables, simply *measuring* performance is computationally expensive.

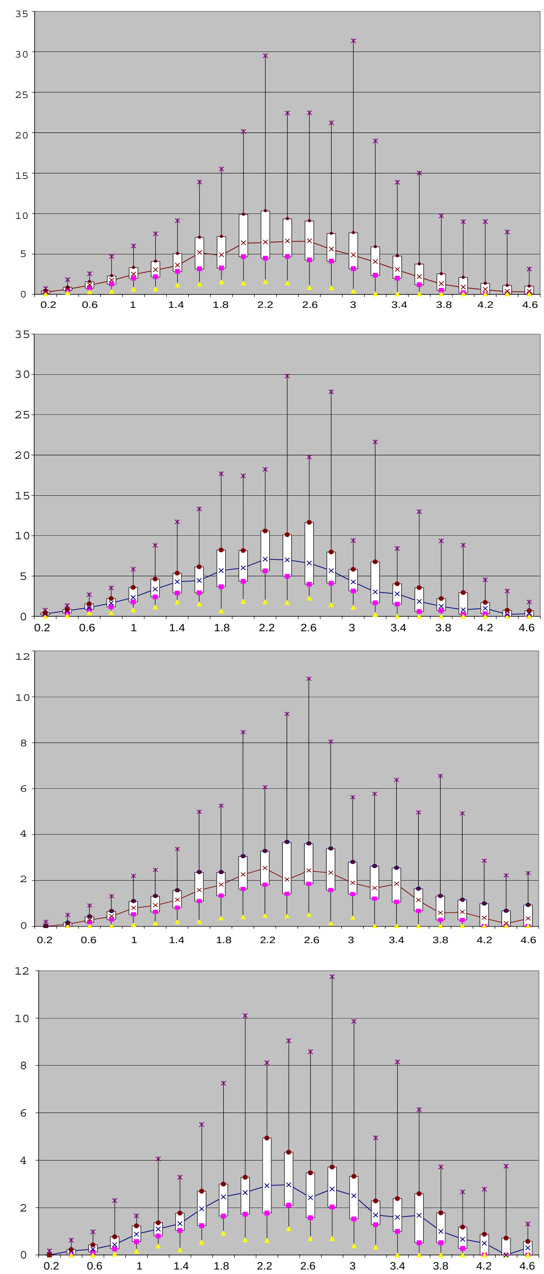

Fig. 2. Relative errors $r_A(\varphi)$ of the algorithms for random 3-CNF with 20 variables as function of density. Measured by exhaustive examination of all length 20 vectors. Averaged over 100 runs. From top to bottom: Horn-LUB, 4-Horn-UB, Renamed-Horn-LUB, Renamed-4-Horn-UB. *The scales for the relative error run from 0–35 for the first two algorithms, but from 0–12 for the Renamed variants.*

the wrong answer for any non-Horn prime implicate query. Thus the renaming heuristic will improve the query-answering accuracy of the LUB. Restricting the length of resolvents in the upper bound, on the other hand, will worsen the accuracy, as some Horn prime implicates may receive the wrong answer. We show the performance of the four algorithms in Figure 3; these are error ratios, and all are fairly high for densities significantly below the critical density of $\alpha \approx 4.2$. We again observe unimodal behavior, with the maximum around a density of 1.6, with some variation by algorithm. The best performance is indeed for Renamed-Horn-LUB, but Renamed-4-Horn-UB is only a little worse than Renamed-Horn-LUB.

For a larger number of variables it is not feasible to exhaustively measure the behavior of a Horn approximation on all truth assignments. In order to estimate the relative error, one could try to use random sampling by generating a random satisfying truth assignment of the Horn upper bounds. This raises the question whether a random satisfying truth assignment of a Horn formula can be generated (almost) uniformly in polynomial time. As far as we know, this is open. In related work, Roth [20] showed that it is NP-hard to approximate the number of satisfying truth assignments of a Horn formula within a multiplicative factor of $2^{n^{1-\epsilon}}$ (for any ϵ) in polynomial time, even if the clauses have size 2 and every variable occurs at most 3 times, and Jerrum et al. [11] established a connection between almost uniform generation and randomized approximate counting.

Table 2. Percentage error in measuring $r_A(\varphi)$ using "pseudo-random" sampling of vectors versus exhaustive. Generates 100,000 "pseudo-random" length 20 vectors; stops early if 50,000 *distinct* vectors obtained; uses only distinct vectors to measure error. Averaged over 10 runs.

α	0.2	0.6	1	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2
%	1.3	5.4	12	12.9	11.8	8.4	4.2	2.3	1.3	0.0	0.0

Table 3. Percentage error in measuring $r_A(\varphi)$ using "pseudo-random" sampling of vectors using weighted selection of variables versus exhaustive. Generates 100,000 "pseudo-random" length 20 vectors; stops early if 50,000 *distinct* vectors obtained; uses only distinct vectors to measure error. Averaged over 10 runs.

α	0.25	0.65	1.05	1.45	1.85	2.25	2.65	3.05	3.45	3.85	4.25
%	3.3	9.1	11.57	11.5	13.4	9.5	3.3	0.4	0.1	0.0	0.0

We have started to do some initial experiments with various heuristics for generating a satisfying truth assignment of a Horn formula. Table 2 compares the relative error of the Horn-LUB algorithm with the estimate of the relative error obtained by "pseudo-random" sampling. The algorithm is a naive one, randomly

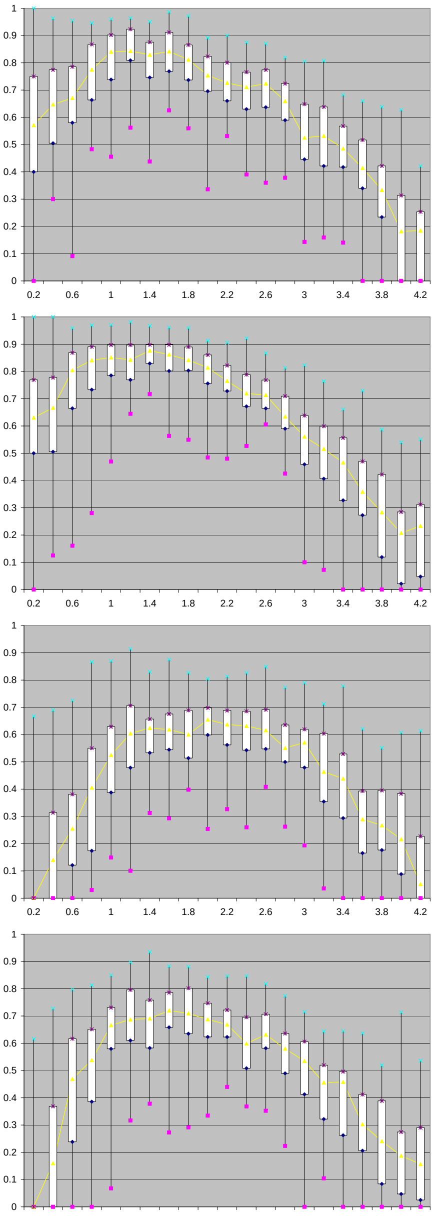

Fig. 3. Fraction of all prime implicate queries to a random 3-CNF formula on 20 variables that would receive the wrong answer from the particular type of Horn upper bound, as a function of density. Averaged over 50 runs. From top to bottom: Horn-LUB, 4-Horn-UB, Renamed-Horn-LUB, Renamed-4-Horn-UB.

selecting variables to be fixed (assigning the same probability to each variable), and deriving all assignments that are forced by the previous choices.

The second algorithm is similar to the first but using a different probability distribution over the variables: the probability assigned to each variable is proportional to the number of its occurrences in the formula. Table 3 also compares the relative error of the Horn-LUB algorithm with the estimate of the relative error obtained by the second algorithm.

Note that truly uniform random generation would require weighting the choices of the two values by the number of satisfying truth assignments corresponding to each value. As Tables 2 and 3 show, the error estimates obtained by "pseudo-random" sampling are rather close to the actual values.

5 Further remarks

We have given negative results on the approximation quality of Horn upper bounds using renaming, and we have presented experimental results for algorithms generating Horn upper bounds. Based on experiments with random 3-CNF for different densities, we have concluded that for 20 variables the algorithm Renamed-4-Horn-UB provides the best compromise in terms of running time, output size and relative error. Also, a Horn bump was observed for the different performance measures in an intermediate range of densities.

There are several directions for further work. An interesting theoretical problem is to construct CNF expressions having superpolynomially large Horn-LUB for every direction (an example for the ordinary Horn closure is given in [22]). The question of almost uniform random generation of a satisfying truth assignment of a Horn formula seems to be of interest from the point of view of extending the experiments to more variables, and also as a question in itself.

The interpretation of the Horn least upper bound as the intersection closure of the set of satisfying truth assignments for a Horn formula leads to the following general question: what is the expected size of the intersection closure of a random subset of $\{0,1\}^n$, given a probability distribution on the subsets? We are not aware of results of this kind. (Another closure problem, the dimension of the subspace spanned by a random set of vectors, has been studied in great detail.) A basic case to consider would be when m random vectors are generated, each component of which is set to 1 with probability p. The size of the Horn-LUB of a random 3-CNF with a given density is a special case of the general question, when the distribution is generated by picking a random formula.

A much studied problem related to the phase transition of random 3-CNF is the evolution of random 3-CNF, in analogy to the classic work of Erdős and Rényi [9] on the evolution of random graphs, and to the study of the evolution of random Boolean functions (see, e.g., Bollobás et al. [2]). In this direction, all of Mora et al. [19], Mézard and Zecchina [18], and San Miguel Aguirre and Vardi [21] show some interesting behavior at densities below the critical density (clustering of the solutions in the case of Mora et al., a particular behavior of a class of local search algorithms in the case of Mézard and Zecchina, and running times for various solvers in the case of San Miguel Aguirre and Vardi). It would be interesting to know if the Horn bump has any connections to these phenomena.

Acknowledgment We would like to thank Eli Ben-Sasson and Bart Selman for useful comments.

References

1. B. Aspvall. Recognizing disguised NR(1) instances of the satisfiability problem. *Journal of Algorithms*, 1:97–103, 1980.
2. B. Bollobás, Y. Kohayakawa, and T. Łuczak. On the evolution of random Boolean functions. In P. Frankl, Z. Füredi, G. Katona, and D. Miklós, editors, *Extremal Problems for Finite Sets* (Visegrád), volume 3 of *Bolyai Society Mathematical Studies*, pages 137–156, Budapest, 1994. János Bolyai Mathematical Society.
3. E. Boros. Maximum renamable Horn sub-CNFs. *Discrete Appl. Math.*, 96-97:29–40, 1999.
4. Y. Boufkhad. Algorithms for propositional KB approximation. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 280–285, 1998.
5. M. Cadoli and F. M. Donini. A survey on knowledge compilation. *AI Communications*, 10(3–4):137–150, 1997.
6. Y. Crama, O. Ekin, and P. L. Hammer. Variable and term removal from Boolean formulae. *Discrete Applied Mathematics*, 75(3):217–230, 1997.
7. A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
8. A. del Val. First order LUB approximations: characterization and algorithms. *Artificial Intelligence*, 162(1-2):7–48, 2005.
9. P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.
10. A. Horn. On sentences which are true on direct unions of algebras. *J. Symbolic Logic*, 16:14–21, 1951.
11. M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.
12. H. Kautz and B. Selman. An empirical evaluation of knowledge compilation by theory approximation. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 155–161, 1994.
13. M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, Massachusetts, 1994.
14. M. Langlois, R. H. Sloan, and G. Turán. Horn upper bounds of random 3-CNF: A computational study. In *Ninth Int. Symp. Artificial Intelligence and Mathematics*, 2006. Available on-line from URL `http://anytime.cs.umass.edu/aimath06/`.
15. A. A. Levin. Comparative complexity of disjunctive normal forms. *Metody Discret. Analiz.*, 36:23–38, 1981. In Russian.
16. H. R. Lewis. Renaming a set of clauses as a Horn set. *J. ACM*, 25:134–135, 1978.
17. J. C. C. McKinsey. The decision problem for some classes without quantifiers. *J. Symbolic Logic*, 8:61–76, 1943.

18. M. Mézard and R. Zecchina. The random K-satisfiability problem: from an analytic solution to an efficient algorithm. *Physical Review E*, 66:056126, 2002.
19. T. Mora, M. Mézard, and R. Zecchina. Pairs of SAT assignments and clustering in random Boolean formulae, 2005. Submitted to *Theoretical Computer Science*, available from URL `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cond-mat/050%6053`.
20. D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82:273–302, 1996.
21. A. San Miguel Aguirre and M. Y. Vardi. Random 3-SAT and BDDs: The plot thickens further. In *Proc. Seventh Int. Conf. Principles and Practice of Constraint Programming (CP)*, pages 121–136, 2001.
22. B. Selman and H. Kautz. Knowledge compilation and theory approximation. *J. ACM*, 43:193–224, 1996.
23. R. H. Sloan, B. Szörényi, and G. Turán. On k-term DNF with the maximal number of prime implicants. Submitted for publication. Preliminary version available as Electronic Colloquium on Computational Complexity (ECCC) Technical Report TR05-023, available on-line at `http://www.eccc.uni-trier.de/eccc/`.
24. K. Truemper. *Effective Logic Computation*. Wiley-Interscience, 1998.
25. H. van Maaren and L. van Norden. Hidden threshold phenomena for fixed-density SAT-formulae. In *Proc. Int. Conf. Theory and Applications of Satisfiability Testing (SAT 2003)*, volume 2219 of *Springer LNCS*, pages 135–149, 2004.