On Approximate Horn Minimization

Amitava Bhattacharya
School of Mathematics
Tata Institute of Fundamental Research
Colaba, Mumbai 400 005
INDIA
Email: amitava@math.tifr.res.in

Bhaskar DasGupta
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7053
Email: dasgupta@cs.uic.edu

Dhruv Mubayi
Department of Mathematics, Statistics & Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7045
Email: mubayi@math.uic.edu

György Turán
Department of Mathematics, Statistics & Computer Science
University of Illinois at Chicago
Chicago, IL 60607-7045
and
Research Group on Artificial Intelligence
University of Szeged, Hungary
Email: gyt@uic.edu

(Extended Abstract)

Abstract

The minimization problem for Horn formulas is to find a Horn formula equivalent to a given formula, using a minimal number of clauses. This problem is important for applications such as the automated development of large knowledge bases. A $2^{\log^{1-\epsilon}(n)}$-inapproximability result is proven, which is the first inapproximability result for the problem. We also consider several other versions of Horn minimization. An $n \log\log n/(\log n)^{1/4}$-approximation algorithm is given for definite Horn formulas for the Steiner-minimization version, where one can introduce new variables in a restricted manner. This is the first algorithm with a $o(n)$ performance guarantee. The algorithm is based on a new result in algorithmic extremal graph theory, on partitioning bipartite graphs into complete bipartite graphs, which may be of independent interest. Inapproximability results and approximation algorithms are also given for restricted versions where only clauses present in the original formula may be used.

1 Introduction

The CNF minimization problem is to find a shortest CNF expression equivalent to a given expression. This problem has been studied, in different versions[1], for many decades in switching theory, computer science and engineering, and it is still the topic of active research, both in complexity theory and circuit design (see, *e.g.*, the survey [56]). It was shown only relatively recently to be Σ_p^2-complete by Umans [54, 55], who also showed a $O(n^{1-\varepsilon})$-inapproximability result for the problem.

Horn minimization is the special case of CNF minimization for Horn formulas. Horn formulas are conjunctions of *Horn clauses*, *i.e.*, of disjunctions containing *at most one* unnegated variable, which can also be written as implications. For instance, $\bar{a} \vee \bar{b} \vee c$ is a Horn clause which can also be written as $a, b \to c$.

Horn formulas are an expressive and tractable fragment of propositional logic, and therefore provide a basic framework for knowledge representation and reasoning [47]. Horn formulas provide, for example, a natural framework for representing systems of rules for expert systems (see, *e.g.*, [14, 17, 26, 27, 45, 50]).

An interesting potential new application area for Horn formulas, involving minimization, is the automated, interactive development of large-scale knowledge bases of commonsense knowledge (see, *e.g.*, [48] for the description of such a project). This application has algorithmic aspects involving knowledge representation, reasoning, learning and knowledge update. A model incorporating these aspects, called *Knowledge Base Learning (KnowBLe)* is formulated in [40] (see also [39, 41]) for related work). Efficient algorithms for approximate Horn minimization would be very useful in these applications, and thus it is of interest to understand the complexity of this problem.

Satisfiability of Horn formulas can be decided in linear time [19] and the equivalence of Horn formulas can be decided in polynomial time (see, *e.g.* [36] for a detailed discussion of algorithmic results on Horn formulas). Thus Horn minimization is expected to be easier than CNF minimization. Horn minimization was shown to be NP-complete by Hammer and Kogan [31] if the number of literals is to be minimized, and by Boros and Čepek [9] if the number of clauses is to be minimized (in contrast to the result by Umans showing Σ_p^2-completeness for general CNF-minimization).

On the positive side, Hammer and Kogan [32] gave a polynomial algorithm for minimizing quasi-acyclic Horn formulas, which include both acyclic and 2-Horn formulas. It was also shown in [31] that there is an efficient $(n-1)$-approximation algorithm for general Horn minimization, where n is the *number of different variables* in the formula (not the number of variable occurrences). As noted in [23], such an algorithm is also provided by the Horn formula learning algorithm of [4].

1.1 Our Contribution and Algorithmic Techniques

We now describe the contribution of this paper together with a succint description of algorithmic and lower-bound techniques used. The main results appear as Theorem 1, Theorem 4, Theorem 17, Theorem 19 and Theorem 20.

In Theorem 4 we prove a super-polylogarithmic $2^{\log^{1-\epsilon}(n)}$-inapproximability result for Horn minimization assuming NP$\not\subseteq$DTIME($n^{\mathrm{polylog(n)}}$) This appears to be the *first inapproximability result* for this problem. The proof is a reduction from a version of parallel repetition of two-prover one-round system (the MINREP problem [37]). The reduction is further complicated by the fact that one needs to rule out new clauses that are not in the original formula.

We next consider several other versions of the Horn minimization problem. It is important to point out that different versions may be relevant, depending on the application, and thus it is of interest to explore their approximability properties. This aspect of the Horn minimization problem does not seem to be considered before.

[1] For instance, the function represented by the formula may be given by its truth table (see, *e.g.*, the recent $O(n^{1-\varepsilon})$-inapproximability result of [33]).

We consider the possibility of *adding new variables* in order to compress the formula. For example, the formula

$$\varphi = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} (x_i \to y_j) \tag{1}$$

having n^2 clauses can be compressed to the $2n$ clause formula

$$\psi = \bigwedge_{i=1}^{n} (x_i \to z) \wedge \bigwedge_{j=1}^{n} (z \to y_j), \tag{2}$$

where z is a new variable. Note that φ and ψ are clearly *not* equivalent, *e.g.*, φ does not depend on z, while ψ does. On the other hand, φ and ψ *are* equivalent in the sense that they both imply the same set of clauses over the original variables $x_1, \ldots, x_n, y_1, \ldots, y_n$. Thus, in terms of the knowledge base application, the new variable z can be thought of as being *internal* to the knowledge base and invisible to the user. Such extensions have been considered by Flögel *et al.* [22]. They showed that deciding the equivalence of extended Horn formulas is co-NP-complete. This is bad news, as it suggests that the extended version is too expressive and therefore intractable[2].

On the other hand, notice that in the example above the new variable is added in a rather restricted manner. Formula (1) can be thought of as a complete directed bipartite graph with parts $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$. Formula (2), then, represents the operation of adding a new node z in the middle, with edges from the x_i's to z and from z to the y_j's. The two graphs have the same reachability relations, as far as the original nodes are concerned. Using the similarity to Steiner problems where new points may be added [57], we refer to this operation as a *Steiner extension* (a formal definition appears in Section 5). As we observe, in contrast to general extensions, the equivalence of Steiner extensions *can* be decided efficiently (Corollary 14). Thus this type of extension could be considered as a tractable alternative in the applications mentioned.

The *Steiner minimization* problem for Horn formulas is to find an equivalent Steiner-extended Horn formula, for a given Horn formula, with a minimal number of terms. In Theorem 17 we prove that there is an efficient $n \log\log n/(\log n)^{1/4}$-approximation algorithm for this problem, where n is the number of variables in the original formula. This is the *first* approximation algorithms for Horn minimization with a $o(n)$ approximation guarantee.

The algorithm makes use of an algorithmic result on the *partition of bipartite graphs into complete bipartite graphs*, which may be of interest on its own. It was shown by Chung, Erdős and Spencer [15] and Bublitz [11] that the edges of every n-vertex graph can be partitioned into complete bipartite graphs such that the sum of the number of vertices in these bipartite graphs[3] is $O(n^2/\log n)$, and this is best possible up to order of magnitude. Tuza [53] gave an analogous result for bipartite graphs. These results are based on a counting argument due to Kővári, Sós and Turán [38], which shows that sufficiently dense graphs contain large complete bipartite subgraphs, and thus are non-constructive. Kirchner [34] considered the problem of finding an algorithmic version, and gave an efficient algorithm to find complete balanced bipartite graphs of size $\Omega(\sqrt{\log n})$ in dense graphs. In a previous paper [44] we improved this to the optimal $\Omega(\log n)$, and as a corollary, showed that partitions proved to *exist* in [11, 15] can also be *found efficiently*[4].

[2]Introducing new variables has also been considered earlier, for example, for resolution proofs, going back to the seminal paper of Tseitin [52]. While there are many exponential lower bounds for resolution proofs (see, *e.g.*, [16]), complexity theoretic results suggest that proving such results for extended resolution is much harder [18].

[3]Note that the complexity of a partition is not measured by the number of graphs in the partition, but by a different measure of complexity, which comes from circuit complexity theory [49].

[4]Extremal combinatorics [8] provides results on the existence of substructures. The results of [34] and [44] can be viewed as *algorithmic extremal combinatorics* in the sense that they also give efficient algorithms to actually find such substructures. Previous results in this direction are given, for example, in [3]. Note that the results of [3] apply to dense graphs and find substructures of constant size, while in our case we also have to handle sparser graphs as well and to find substructures of nonconstant size.

In this paper we give an algorithmic version for the bipartite case. It turns out that Tuza's result [53] is not quite correct, as it fails for very unbalanced bipartite graphs, such as a star. We show in Theorem 1 that the edges of every bipartite graph with sides a and b, where $a \geq b$, can be partitioned into complete bipartite graphs such that the sum of the number of vertices of these graphs is $O((ab/\log a) + a \log b)$, and we give an efficient algorithm to find such a partition [5].

We also consider *restricted* versions of the Horn minimization problem, where one is restricted to use clauses from the original formula. Such a restriction may be justified in applications where the particular rules, provided by an expert, are supposed to be meaningful and thus cannot be replaced. The goal is to eliminate redundant rules. One may want to optimize by *minimizing the number of rules left*, or by *maximizing the number of rules removed*. The two versions may differ in their approximability (cf. the maximum independent set and the minimum vertex cover problems for graphs). As (1) suggests, Horn formulas with clauses of the form $x \to y$ correspond to directed graphs. For such formulas, optimization corresponds to *transitive reduction* problems for directed graphs[6]. Thus approximation algorithms for these directed graph problems may be applied for Horn formulas. We give several examples of this connection. We also prove $2^{\log^{1-\epsilon}(n)}$-inapproximability for the restricted case, which holds *even if the input formula has clauses of size at most* 3. The proof idea is similar to that in Theorem 4; however, additional care is necessary to ensure that no clause in the input formula has more than three literals.

The rest of the paper is organized as follows. In Section 2 we give background material for Horn formulas. Bipartite graph decompositions are discussed in Section 3. Section 4 contains the inapproximability result for Horn minimization. Horn minimization with new variables is discussed in Section 5, and restricted Horn minimization in Section 6. *All proofs omitted due to lack of space appear in the appendix.*

2 Preliminaries

A *literal* is a variable or the negation of a variable. We use n to denote the number of variables. A *clause* is a disjunction of literals. A *Horn clause* is a clause with at most one unnegated variable. A *definite* (or *pure*) Horn clause has exactly one unnegated variable, called its *head*; the other variables form its *body*. A *negative* clause consists of only negated variables. The *size* of a clause is the number of its literals. A clause of size 1 (resp., 2) is called a *unit* (resp., *binary*) clause.

A *Horn formula* is a conjunction (or a set) of Horn clauses. A *definite* (or *pure*) Horn formula is a conjunction of definite (or pure) Horn clauses. *In this paper we restrict our attention to definite Horn formulas*[7]. The *size* of a formula is the number of its clauses. A *k-Horn formula* is a Horn formula with clauses of size at most k.

A clause C is an *implicate* of a formula φ if every truth assignment satisfying φ also satisfies C. An implicate is a *prime* implicate if none of its proper subclauses is an implicate.

Clauses C_1 and C_2 *conflict* in variable x if C_1 contains x and C_2 contains $\bar{x}$ or vice versa. Let C_1 and C_2 be clauses conflicting in x but not conflicting in any other variable. The *resolvent* of C_1 and C_2 is the clause containing all literals occurring in either C_1 or C_2, except x and $\bar{x}$. It is a basic result that a clause C is a prime implicate of a CNF φ iff it can be derived from φ by a sequence of resolution steps. Deciding whether a clause C is an implicate of a CNF expression φ is co-NP-complete.

Deciding whether a *definite Horn* clause C is an implicate of a *definite Horn* formula φ can be decided by a simple and well-known marking procedure often called *forward chaining*. The procedure begins by marking the variables in the body of C. If every variable in the body of a clause in φ is marked then its head is marked as well. This is repeated as long as new variables get marked. Then

---

[5] The $a \log b$ term is an additional term needed to cover the case when b is much smaller than a.

[6] From this perspective, Horn minimization problems may be viewed as the hypergraph versions of transitive reduction problems for directed graphs [5].

[7] The extension of the algorithms to the general case will be discussed in the final version of the paper. Note that structural results on Horn formulas show that negative clauses are often easier to handle than definite ones [29].

it holds that C is an implicate of φ iff its head gets marked.

3 Partitioning bipartite graphs

Let $G = G(A, B, E)$ be a bipartite graph with parts A, B of sizes a, b, respectively, and edge set E of size m. We assume w.l.o.g. that $a \geq b$. A bipartite graph is *balanced* if $|A| = |B|$. The complete balanced bipartite graph with parts of size q is denoted by $K_{q,q}$.

We consider complete bipartite graphs $G_i = (A_i, B_i, E_i)$ for $i = 1, \ldots, t$ such that $A_i \subseteq A, B_i \subseteq B$, and $(E_1, \ldots, E_t)$ is a partition of E. The complexity of such a decomposition is $\sum_{i=1}^{t}(|A_i| + |B_i|)$. The problem is to find in polynomial time a decomposition of small complexity. The trivial decomposition into single edges has complexity $2m \leq 2ab$. We are interested in upper bounds depending on a and b, independent of m.

Theorem 1 *For every bipartite graph G one can find a decomposition of complexity $O\left(\frac{ab}{\ln a} + a \ln b\right)$ in polynomial time.*

Proof. We give a polynomial time procedure to find a large balanced complete bipartite subgraph of a bipartite graph. The algorithm to find a decomposition, referred to below as $DECOMP$, calls this procedure repeatedly, and deletes edges contained in the new subgraph. When the number of edges drops below $6a$, each remaining edge is included as a singleton. Let $3 \leq b \leq a$, $6a \leq m \leq ab$ and $f(a, b, m) = \left\lfloor \frac{\ln a}{\ln(2eab/m)} \right\rfloor$; note that the case $b \leq 2$ is trivial. The procedure for finding a large balanced complete bipartite subgraph is slightly different when the underlying graph is dense as opposed to sparse.

Lemma 2 *Suppose that $m \geq af(a, b, m)$. Then there is a polynomial time algorithm that finds a $K_{q,q}$ in G with $q = f(a, b, m)$.*

Lemma 3 *Suppose that $m < af(a, b, m)$. Then there is a polynomial time algorithm that finds a $K_{q,q}$ in G with $q = \lfloor m/a \rfloor$.*

Now we can complete the proof of the theorem using an argument similar to [53]. A bipartite subgraph $K_{q,q}$ adds $2q$ to the complexity of the decomposition: distribute this charge among the q^2 edges of $K_{q,q}$ by assigning a weight $2/q$ to each edge. Note that the density condition distinguishing between the two procedures has m on both sides, but a simple calculation shows that applications of Lemma 2 precede those of Lemma 3.

If Lemma 2 is applied and the current number of edges is between $ab/(s + 1)$ and ab/s for some s, then each edge gets weight $\Theta(\ln s/ \ln a)$. There are $\Theta(ab/s^2)$ edges in this range. Hence, up to order of magnitude, the total weight is at most $\displaystyle\sum_{s=1}^{\infty} \frac{ab}{s^2} \cdot \frac{\ln s}{\ln a} = O\left(\frac{ab}{\ln a}\right)$.

If Lemma 3 is applied and the current number of edges is between $ab/(s + 1)$ and ab/s for some s, then each edge gets weight $\Theta(s/b)$. Again, there are $\Theta(ab/s^2)$ edges in this range. By our assumption on the number of edges, it also holds that $s \leq b$. Hence, up to order of magnitude, the total weight is at most $\displaystyle\sum_{s=1}^{b} \frac{ab}{s^2} \cdot \frac{s}{b} = O(a \ln b)$. ❑

Remark 1 *If G is a star then $b = 1$ and the optimal decomposition has complexity $a + 1$, hence the upper bound $ab/ \ln a$ claimed in [53] does not hold, and an additional term (or some other modification) is needed. It is open whether the term $a \ln b$ can be improved.*

4 Inapproximability

Theorem 4 *For any fixed $0 < \epsilon < 1$, unless NP$\subseteq$DTIME$(n^{polylog(n)})$, the Horn minimization problem is $2^{\log^{1-\epsilon} n}$-inapproximable.*

Proof. We reduce a version of the parallel repetition of two-prover one-round system, described as the MINREP problem in [37]. Given a bipartite graph $G = (A, B, E)$, a partition of A into $|A|/\alpha$ equal-size subsets $A_1, A_2, \ldots, A_\alpha$ and a partition of B into $|B|/\beta$ equal-size subsets $B_1, B_2, \ldots, B_\beta$, one can define a natural bipartite super-graph H in the following manner. H has a super-vertex for every A_i and a super-vertex for every B_j. There exists a super-edge between the super-vertex A_i and the super-vertex B_j if and only if there exists $u \in A_i$ and $v \in B_j$ such that $\{u, v\}$ is an edge of G.

A pair of nodes u and v *witnesses* a super-edge $\{A_i, B_j\}$ provided $u \in A_i$, $v \in B_j$ and the edge $\{u, v\}$ exists in G. A set of nodes S of G witnesses a super-edge if and only if there exists at least one pair of nodes in S that witnesses the super-edge. The goal of MINREP is to find $A' \subseteq A$ and $B' \subseteq B$ such that $A' \cup B'$ witnesses every super-edge of H and $|A'| + |B'|$ is as small as possible. The *size* of an optimal solution is denoted by $OPT(G)$. Let $s = |A| + |B|$. The following result is a direct consequence of results in [37, 46], providing a $2^{\log^{1-\varepsilon} n}$-inapproximability result for MINREP under the complexity-theoretic assumption of NP$\not\subseteq$DTIME$(n^{\text{polylog(n)}})$.

Theorem 5 [37] *Let L be any language in NP and $0 < \varepsilon < 1$ be any constant. Then, there exists a reduction running in quasi-polynomial time, that, given an instance x of L, produces an instance y of size s of* MINREP *such that:*

- *if $x \in L$ then y has a solution of size $\alpha + \beta$;*
- *if $x \notin L$ then every solution for y has size at least $(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$.*

Consider an instance of MINREP . We use the following notation:

- s is the number of nodes in G,
- m is the number of edges in G,
- $\alpha + \beta$ is the number of super-nodes in H,
- p is the number of super-edges in H,
- t is a sufficiently large positive integer to be fixed later.

We produce a definite Horn formula φ corresponding to the MINREP instance. For simplicity and with an abuse of notation, some variables in φ are denoted as the corresponding objects (vertices and super-edges) in the MINREP instance. The formula φ contains the following variables:

Amplification variables: There are t such variables $x_1, \ldots, x_t$. Let $\mathcal{V}_{\text{amplification}}$ denote the set of amplification variables.

Node variables: There is a variable u for every vertex $u \in A \cup B$. Let $\mathcal{V}_{\text{node}}$ denote the set of node variables.

Super-edge variables: There is a variable e for every super-edge e in H. Let $\mathcal{V}_{\text{super-edge}}$ denote the set of super-edge variables.

The clauses of φ belong to the following groups:

Amplification clauses: There is a clause $x_i \to u$ for every $i \in \{1, \ldots, t\}$ and for every $u \in A \cup B$. There are st such clauses. Let $\mathcal{T}_{\text{amplification}}$ be the set of these clauses.

Witness clauses: There is a clause $u, v \to e$ for every super-edge e of H and for every pair of nodes $u \in A$ and $v \in B$ witnessing e. There are m such clauses. Let $\mathcal{T}_{\text{witness}}$ be the set of these clauses.

Feedback clauses: Let the super-edges of H be enumerated as $e_1, e_2, \ldots, e_p$. There is a clause $e_1, \ldots, e_p \to u$ for every $u \in A \cup B$. There are s such clauses. Let $\mathcal{T}_{\text{feedback}}$ be the set of these clauses.

Let $\mathcal{V} = \mathcal{V}_{\text{amplification}} \cup \mathcal{V}_{\text{node}} \cup \mathcal{V}_{\text{super-edge}}$ denote the set of all variables. The total number of variables is $s + p + t$ and the total number of clauses is $st + m + s$.

As φ is definite, all its prime implicates are definite. Also, as φ consists of non-unit definite clauses, all its prime implicates are non-unit (the all-zero vector satisfies φ and falsifies all unnegated variables). For a further analysis of the prime implicates of φ, we make use of the forward chaining procedure described in Section 2 in the following three lemmas.

Lemma 6 *Let x be an amplification variable. Then the prime implicates containing x are clauses of the form $x \to v$, where v is a node or super-edge variable.*

Proof. The variable x cannot be the head of an implicate as the truth assignment setting x to 0 and all other variables to 1 satisfies φ and falsifies such an implicate. Clauses of the form $x \to v$, where v is a node or super-edge variable are implicates as starting from x we can mark first all node variables and then all super-edge variables. If an implicant contains another variable in its body besides x than that variable can be deleted and we still get an implicant. ❑

Lemma 7 *Let U be a set of node variables such that U is not a solution to* Minrep *, and U' be the set of super-edge variables witnessed by U. Then every implicate with body contained in $U \cup U'$ has head in U'.*

Lemma 8 *Let x be an amplification variable and let ψ be a prime and irredundant Horn formula equivalent to φ. Then ψ has at least $OPT(G)/3$ clauses containing x.*

Proof. Consider the clauses in ψ containing x, and let the set of their heads be $U_1 \cup U_2$, where $U_1 \subseteq \mathcal{V}_{\text{node}}$ and $U_2 \subseteq \mathcal{V}_{\text{super-edge}}$. For every $e \in U_2$, pick node variables u, v such that $u, v \to e$ is a witness clause, and replace $x \to e$ by the prime implicates $x \to u, x \to v$ and $u, v \to e$ in ψ. The new formula ψ' is still equivalent to φ. Every clause in ψ' containing x has a node variable head.

Assume that $U = \{u : \ u \in \mathcal{V}_{\text{node}}, \ (x \to u) \in \psi\}$ is not a solution to Minrep and consider a node variable $v \notin U$. As $x \to v$ is a prime implicate of ψ', the forward chaining procedure over ψ', starting from x, has to mark v. The procedure, first, can mark all node variables in U. As U is not a solution to Minrep , Lemma 7 implies that after that we can mark all super-edges variables witnessed by U and nothing else. Hence v cannot be marked. This implies that U must be a solution to Minrep , and so $|U| \geq OPT(G)$. But $|U| \leq |U_1| + 3|U_2| \leq 3(|U_1| + |U_2|)$, and so the lemma follows. ❑

Lemma 9 (Gap preserving reduction lemma)

(a) *If $OPT(G) = \alpha + \beta$, then $OPT(\varphi) \leq t \cdot (\alpha + \beta) + m + s$.*
(b) *If $OPT(G) \geq (\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ then, $OPT(\varphi) \geq t(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}/3$.*

Proof. To prove part **(a)**, suppose that there is a solution U of Minrep containing α nodes of A and β nodes of B (and, thus of size $\alpha + \beta$). Then the formula ψ consisting of all witness and feedback clauses, plus all amplification clauses with heads in U, is equivalent to φ. This formula has the required size. In order to show that it is equivalent to φ, we only need to show that implies all amplification clauses $x \to u$ with $u \notin U$. Using ψ, starting from x, we can first mark U. Then, as U is a solution to Minrep , we can mark every super-edge variable using witness clauses, and, finally, we can mark u using the feedback clause with head u.

To prove part **(b)**, suppose that $\text{OPT}(G) \geq (\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ and consider any Horn formula ψ equivalent to φ. Lemma 23 implies that every amplification variables must participate in at least $OPT(G)/3$ clauses and since there are t such variables ψ contains at least $t(\alpha + \beta) \cdot (2^{\log^{1-\varepsilon} s}/3)$ amplification clauses. Thus, $\text{OPT}(\varphi) \geq t(\alpha + \beta) \cdot (2^{\log^{1-\varepsilon} s}/3)$. ❑

We are now ready to finish the proof of Theorem 4. Set $t = s^3$ and remember that $m < s^2$. Let L be any language in NP and x be an input to L.

- If $x \in L$, then $\mathrm{OPT}(G) = \alpha+\beta$ which implies $\mathrm{OPT}(M) \leq s^3\cdot(\alpha+\beta)+m+s \leq s^3\cdot(\alpha+\beta)\cdot(1+o(1))$.
- If $x \notin L$, then $\mathrm{OPT}(G) \geq (\alpha+\beta)\cdot 2^{\log^{1-\varepsilon} s}$ which implies $\mathrm{OPT}(M) \geq s^3(\alpha+\beta)\cdot(2^{\log^{1-\varepsilon} s}/3)$.

This gives an inapproximability bound of $2^{\log^{1-\varepsilon} s}$ for any $0 < \varepsilon < 1$. Since the number of variables n is given by $n = s^3 + s + m < 3s^3$, this gives us an inapproximability bound of $2^{\log^{1-\varepsilon} n}$. ❑

5 Formulas with new variables

In this section we consider versions of the Horn minimization problem where one can introduce new variables in order to compress the formula. First we consider the general version where there is no restriction on the way the new variables are introduced, which turns out to be too general for the intended knowledge base applications. Then we introduce a restricted version, which is both tractable and admits a $o(n)$-approximate minimization algorithm.

5.1 General extensions

The following generalized notion of equivalence was introduced in [22].

Definition 10 *(Generalized equivalence) Let X be a set of variables. Formulas φ and ψ are X-equivalent if for every clause C over X it holds that $\varphi \models C$ iff $\psi \models C$.*

Consider the set of variables $X = \{x_1, \ldots, x_n, y_1, \ldots, y_n, u\}$ and the 2^n-clause Horn formula

$$\varphi = \bigwedge (v_1, \ldots, v_n \to u) \tag{3}$$

where $v_i \in \{x_i, y_i\}$ for $i = 1, \ldots, n$, and the conjunction includes all possible such selections. As no resolutions can be performed, it follows that all the prime implicates of φ are the clauses themselves. Let now $\{z_1, \ldots, z_n\}$ be new variables. Then the $(2n+1)$-clause Horn formula

$$\psi = (z_1, \ldots, z_n \to u) \wedge \bigwedge_{i=1}^{n} (x_i \to z) \wedge (y_i \to z) \tag{4}$$

is X-equivalent to φ. Thus the introduction of new variables can lead to an exponential compression in size.

It was shown by [22] that generalized equivalence is too general for applications, as deciding it for Horn formulas is hard. We give a somewhat simplified proof. The proof shows that generalized equivalence is already hard if new variables are introduced in a rather restricted manner. This gives a motivation to consider even more stringent restrictions on the introduction of new variables.

Theorem 11 [22] *Generalized equivalence of definite Horn formulas is co-NP-complete.*

5.2 Steiner extension

Motivated by Theorem 11, we introduce a restricted form of introducing new variables.

Definition 12 *(Steiner extension) Let φ be a Horn formula and X be a subset of its variables. Then φ is a* Steiner extension *over X if every variable* not *in X occurs in φ either as a head, or as a single body variable in a binary definite clause having its head in X.*

The corresponding notion of equivalence is the following.

Definition 13 *(Steiner equivalence) Let X be a set of variables. Horn formulas φ and ψ are* Steiner *X-equivalent if*

- *φ and ψ are X-equivalent,*
- *both φ and ψ are Steiner extensions over X.*

Note that the Horn formulas in (1) and (2) above are both Steiner extensions over $X = \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ (the first formula vacuously so), and they are Steiner X-equivalent. On the other hand, (4) is not a Steiner extension over $\{x_1, \ldots, x_n, y_1, \ldots, y_n, u\}$, and neither (5) nor (6) are Steiner extensions over $X = \{x_1, \ldots, x_n, u\}$, as additional variables occur in the body of a non-binary clause. In contrast to Theorem 11, Steiner equivalence can be decided efficiently. For this, we first note the following.

Proposition 1 *There is a polynomial algorithm which, given a Steiner extension φ over X, computes a Steiner X-equivalent Horn formula $\psi(X)$ such that $size(\psi) = O(size(\varphi^2))$.*

Corollary 14 *Steiner equivalence of Horn formulas is in P.*

The optimization problem for Steiner equivalence is the following.

Definition 15 *(Steiner minimization of Horn formulas) Given a Horn formula φ over variables X, find a minimal size Horn formula that is Steiner X-equivalent to φ.*

We are going to consider different notions of optimality as summarized in the following definition.

Definition 16 *(Notions of optimality) For a Horn formula φ over the set of variables X, let*

- *$OPT(\varphi)$ denote the minimal number of clauses in any Horn formula equivalent to φ,*
- *$OPT_S(\varphi)$ the minimal number of terms in any Horn formula Steiner X-equivalent to φ,*
- *$OPT - BODY(\varphi)$ the minimal number of distinct bodies in any Horn formula equivalent to φ,*
- *$OPT - BODY_S(\varphi)$ the minimal number of distinct bodies containing only variables from X in any Horn formula Steiner X-equivalent to φ.*

There are various relationships between these notions such as the one stated below.

Proposition 2 $OPT - BODY_S(\varphi) = OPT - BODY(\varphi) \leq OPT_S(\varphi) \leq OPT(\varphi) \leq \frac{1}{4}OPT_S^2(\varphi)$

The main result of this section is that Steiner minimization of Horn formulas has an efficient approximation algorithm with performance guarantee $o(n)$. Note that n refers to the number of variables in the original formula (the formula produced by the algorithm can have more variables).

Theorem 17 *There is an efficient $O(n \log\log n/(\log n)^{1/4})$-approximation algorithm for Steiner minimization of definite Horn formulas.*

Proof. The algorithm uses previous algorithms listing prime implicates of Horn formulas and for body minimization. It also uses a procedure for the exact minimization of Horn formulas having a short equivalent formula and the bipartite graph decomposition algorithm *DECOMP* of Section 3.

The *prime implicate listing problem* for Horn formulas is to produce a list of all prime implicates of a Horn formula. As, for example, (4) shows, the number of prime implicates can be exponential in the size of the original formula, a possible criterion of efficiency is *total polynomial time*, *i.e.*, time polynomial in the combined size of the input and the output. Boros, Crama and Hammer [10] give an algorithm which lists all prime implicates of a Horn formula, in time polynomial in the size of the formula and the number of prime implicates.

Consider the following special case of the Horn minimization problem. Note that this problem is about standard Horn minimization, without introducing additional variables.

Problem 1 *($\sqrt{\log n}$-Horn minimization) Given a Horn formula φ over n variables, find an equivalent minimal size Horn formula of size at most $\sqrt{\log n}$ if such a formula exists, or output* 'none'.

Lemma 18 *The $\sqrt{\log n}$-Horn minimization problem can be solved in polynomial time.*

Proof. The algorithm, referred to below as $\sqrt{\log n} - HORN - MIN$, works as follows. Given a Horn formula φ, it starts listing its prime implicates, using the algorithm of [10]. It is shown in [13] that a k-term CNF can have at most $2^k - 1$ prime implicates. If more than $2^{\sqrt{\log n}}$ prime implicates are produced, or the running time allowed for $2^{\sqrt{\log n}}$ prime implicates is exceeded, the procedure is terminated and *'none'* is output. Otherwise we know that φ has at most $2^{\sqrt{\log n}}$ prime implicates. Then all sets of at most $\sqrt{\log n}$ prime implicates are checked for equivalence with φ, ordered according to increasing size, and the first one equivalent to φ is output. If neither of these is equivalent then *'none'* is output. This algorithm clearly works, and as

$$\binom{2^{\sqrt{\log n}}}{\sqrt{\log n}}$$

is polynomial in n, its running time is polynomial. ❑

A further ingredient of our algorithm is an efficient procedure for body minimization. The *body minimization* problem for Horn formulas asks for an equivalent Horn formula with the minimal number of distinct bodies. While Horn minimization is hard, there *are* efficient algorithms for body minimization. Interestingly, such algorithms were found in several different contexts, such as functional dependencies for databases [42], directed hypergraphs [5], lattices [12] and computational learning theory[4]. In what follows one can use any of those algorithms.

Given a Horn formula χ over a set of variables X, we now describe a construction of a Steiner extension ψ of χ. Let $Bodies(\chi)$ denote the set of bodies in χ, and $Heads(\chi)$ denote the set of heads in χ. Form a bipartite graph $G(\chi)$ with parts $Bodies(\chi)$ and $Heads(\chi)$, adding an edge between a body and a head if the corresponding Horn clause occurs in χ.

Let $G_1, \ldots, G_t$ be a decomposition of $G(\chi)$ into complete bipartite graphs[8]. Let the bipartite graphs in the decomposition have parts $A_i \subseteq Bodies(\chi)$ and $B_i \subseteq Heads(\chi)$ for $i = 1, \ldots, t$. Introduce new variables $y_1, \ldots, y_t$, and let ψ consist of the clauses $b \to y_i$ and $y_i \to h$ for every $b \in A_i$ and $h \in B_i$, $i = 1, \ldots, t$.

Input: a Horn formula φ
find an equivalent Horn formula χ with the minimal number of bodies
$k = |Bodies(\chi)|, \ \ell = |Heads(\chi)|$
if ($k \leq \ell$ and $k \leq \sqrt{\log n}$) **then**
  if $\sqrt{\log n}$-HORN-MIN(χ) returns $\varphi \neq$ '*none*' then **return** ψ
else
  return ψ representing $DECOMP(G(\chi))$

Given a Horn formula φ over X, let $ALG(\varphi)$ be the number of clauses in the formula produced by the algorithm.

If $\sqrt{\log n}$-HORN-MIN(χ) is successful, then $OPT(\varphi) \leq \sqrt{\log n}$ and the formula ψ has size $OPT(\varphi) \leq (1/4)OPT_S(\varphi)^2$ by Proposition 2. Thus $\dfrac{ALG(\varphi)}{OPT_S(\varphi)} \leq \dfrac{1}{4} \cdot OPT_S(\varphi) = O(\sqrt{\log n})$.

Otherwise it holds that $OPT(\varphi) \geq \sqrt{\log n}$, and so by Proposition 2 $OPT_S(\varphi) \geq 2(\log n)^{1/4}$. Proposition 2 also implies that $OPT_S(\varphi) \geq k$. We distinguish two cases.

If $k \geq \ell$ then

$$\frac{ALG(\varphi)}{OPT_S(\varphi)} \leq c \cdot \frac{\frac{k\ell}{\ln k} + k \ln \ell}{k} = c \cdot \frac{\ell}{\ln k} + \ln \ell = O\left(\frac{\ell}{\ln \ell}\right) = O\left(\frac{n}{\ln n}\right)$$

[8]The bipartite graphs need not be balanced. Also, for this application it would be sufficient to consider coverings instead of partitions. The result of Section 3 is formulated for balanced partitions in order to give a stronger positive result.

.

If $k < \ell$ then

$$\begin{aligned}
\frac{ALG(\varphi)}{OPT_S(\varphi)} &\leq c \cdot \left(\frac{\frac{k\ell}{\ln \ell} + \ell \ln k}{OPT_S(\varphi)} \right) \leq c \cdot \left(\frac{\ell}{\ln \ell} + \frac{\ell \ln OPT_S(\varphi)}{OPT_S(\varphi)} \right) \\
&\leq c' \cdot \left(\frac{n}{\ln n} + n \cdot \frac{\ln \ln n}{(\ln n)^{1/4}} \right) = O\left(\frac{n \ln \ln n}{(\ln n)^{1/4}} \right).
\end{aligned}$$

❑

6 Restricted Horn minimization

A special case of the Horn optimization problem is when when every clause of the optimized formula must also be a clause of the new formula. In this section we discuss approximability issues for this special case.

6.1 Inapproximability

An inapproximabilitity result similar to Theorem 4 holds in the case of restricted Horn minimization as well. It is stronger than Theorem 4 in that it applies *even if we assume that the formula to be minimized is* 3*-Horn.*

Theorem 19 *For any fixed* $0 < \epsilon < 1$*, unless NP*$\subseteq$*DTIME*$(n^{polylog(n)})$*, the restricted Horn minimization problem is* $2^{\log^{1-\epsilon} n}$*-inapproximable, even for 3-Horn formulas.*

Our proof of the above theorem has much in common with the proof of Theorem 4. In the gadget construction in the proof of Theorem 4 the only clauses that were not 3-Horn clauses were the s feedback clauses. We modify the construction to replace the feedback clauses by a new set of $s+p-1$ feedback clauses that are 3-Horn. Additional arguments are necessary to show that the construction is correct for the restricted case; details appear in Section H of the appendix.

6.2 Approximation algorithms for definite 2-Horn formulas

As noted in the introduction, by the "Horn maximization" problem, we refer to the problem in which our goal is to maximize the difference in size between the original and the optimized formula.

Theorem 20
(a)*Both the Horn minimization and Horn maximization problems are MAX-SNP-hard for definite* 2*-Horn formulas without unit clauses.*
(b)*Restricted Horn minimization for definite* 2*-Horn formulas admits a* 1.5*-approximation if no unit clauses are part of input and admits a* 2*-approximation otherwise.*
(c)*Restricted Horn maximization for definite* 2*-Horn formulas admits a* 2*-approximation.*

Remark 2 *For* **(a)***, the best inapproximability constants can be be obtained by using a randomized construction of a special class of Boolean satisfiability instances by Berman* et al.[7] *giving an inapproximability constant of* $1 + 1/896$ *for the minimization version and* $1 + 1/539$ *for the maximization version.*

References

[1] A. Aho, M. R. Garey and J. D. Ullman. *The transitive reduction of a directed graph*, SIAM Journal of Computing, 1 (2), 131-137, 1972.

[2] R. Albert, B. DasGupta, R. Dondi and E. Sontag. *Inferring (Biological) Signal Transduction Networks via Transitive Reductions of Directed Graphs*, Algorithmica, 51 (2), 129-159, June 2008.

[3] N. Alon, R. A. Duke, H. Lefmann, V. Rödl, R. Yuster: The algorithmic aspects of the regularity lemma, *J. of Algorithms* **16** (1994), 80-109.

[4] D. Angluin, M. Frazier, L. Pitt: Learning conjunctions of Horn clauses, *Machine Learning* **9** (1992), 147-164.

[5] G. Ausiello, A. D'atri and D. Saccà. *Minimal Representations of Directed Hypergraphs*, SIAM J. Computing, 15 (2), 418-431, 1986.

[6] P. Berman, B. DasGupta and M. Karpinski. *Approximating Transitive Reduction Problems for Directed Networks*, to appear in Algorithms and Data Structures Symposium (WADS), 21-23 August, 2009.

[7] P. Berman, M. Karpinski and A. D. Scott. *Approximation Hardness of Short Symmetric Instances of MAX-3SAT*, Electronic Colloquium on Computational Complexity, Report TR03-049, 2003.

[8] B. Bollobás: *Extremal Graph Theory.* Academic Press, 1978.

[9] E. Boros and E. Cepek̆. *On the Complexity of Horn Minimization*, Rutcor Research Report 1-94, 1994.

[10] E. Boros, Y. Crama, P. L. Hammer: Polynomial-time inference of all valid implications for Horn and related formulae, *Ann. Math. Artif. Intell.* **1** (1990), 21-32.

[11] S. Bublitz: Decomposition of graphs and monotone formula size of homogeneous fuctions, *Acta Informatica* **23** (1986), 689-696.

[12] N. Caspard, B. Monjardet: The lattice of closure systems, closure operators and implications on a finite set: a survey, *Discrete Appl. Math.* **127** (2003), 241-269.

[13] A. K. Chandra, G. Markowsky: On the number of prime implicants, *Discrete Math.* **24** (1978), 7-11.

[14] E Charles and O Dubois. *MELODIA: Logical methods for checking knowledge bases In Validation Verification and Test of Knowledge based Systems*, M. Ayel and J. P. Laurent (eds), Wiley & Sons, 95-105, 1991.

[15] F. R. K. Chung, P. Erdős, J. Spencer: On the decomposition of graphs into complete bipartite graphs, in: *Studies in Pure Mathematics, To the Memory of Paul Turán*, 95-101. Akadémiai Kiadó, 1983.

[16] P. Clote, E. Kranakis: *Boolean Functions and Models of Computation.* Springer, 2003.

[17] R. M. Colomb and C. Y. C. Chung. *Very fast decision table execution of propositional expert systems*, Proceedings of the 8th National Conference on Artificial Intelligence (AAAI), 237-270, 1990.

[18] S. A. Cook: Feasibly constructive proofs and the propositional calculus, *7. STOC* (1975), 83-97.

[19] W. F. Dowling and J. H. Gallier. *Linear time algorithms for testing the satisfiability of propositional Horn formulae*, Journal of Logic Programming, 3, 267-284, 1984.

[20] C.Delobel and R.G.Casey. *Decomposition of a Data Base and the Theory of Boolean Switching Functions*, IBM J. Res. Develop., 17, 374-386, 1973.

[21] U. Feige, S. Kogan: Hardness of approximation of the balanced complete bipartite subgraph problem, *Tech. Rep. MCS04-04, Dept. of Comp. Sci. and Appl. Math., The Weizmann Inst. of Science*, 2004.

[22] A. Flögel, H. Kleine Büning, T. Lettmann: On the restricted equivalence subclasses of propositional logic, *ITA* **27** (1993), 327-340.

[23] M. D. Frazier: *Matters Horn and other features in the computational learning theory landscape: the notion of membership.* Ph. D. thesis, Univ. of Illinois at Urbana-Champaign, 1994.

[24] M. D. Frazier, L. Pitt: Learning from entailment: an application to propositional Horn sentences, *Proc. ICML'93* (1993), 120-127.

[25] G. N. Frederickson and J. JàJà. *Approximation algorithms for several graph augmentation problems*, SIAM Journal of Computing, 10 (2), 270-283, 1981.

[26] A. Ginsberg. *Knowledge-base reduction: A new approach to checking knowledge bases for inconsistency redundancy*, Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-88), 585-589, 1988.

[27] F. Glover and H. J. Greenberg. *Logical testing for rule-base management*, Annals of Operations Research, 12, 199-215, 1988.

[28] P. Hammer and A. Kogan. *Optimal Compression of Propositional Horn Knowledge Bases: Complexity and Approximation*, Rutcor Research Report 2-93, January 1993.

[29] P. L. Hammer and A. Kogan. *Horn functions and their DNFs*, Information Processing Letters, 44, 23-29, 1992.

[30] P.L. Hammer and A. Kogan. *Horn Function Minimization and Knowledge Compression in Production Rule Bases*, RUTCOR Research Report RRR 892 , Rutgers University, New Brunswick, NJ, March 1992.

[31] P. L. Hammer and A. Kogan. *Optimal Compression of Propositional Horn Knowledge Bases: Complexity and Approximation*, Artificial Intelligence, 64, 131-145, 1993.

[32] P. L. Hammer, A. Kogan: Quasi-acyclic Horn knowledge bases: optimal compression, *IEEE Trans. on Knowledge and Data Engineering* **7** (1995), 751-762.

[33] S. Khot and R. Saket. *Hardness of Minimizing and Learning DNF Expressions*, 49th Annual IEEE Symposium on Foundations of Computer Science, 231-240, 2008.

[34] S. Kirchner: Lower bounds for Steiner tree algorithms and the construction of bicliques in dense graphs. Ph.D. Dissertation, Humboldt-Universität zu Berlin, 2008. (In German.)

[35] S. Khuller, B. Raghavachari and N. Young. *Approximating the minimum equivalent digraph*, SIAM Journal of Computing, 24 (4), 859-872, 1995.

[36] H. Kleine Büning, T. Lettmann: *Propositional Logic: Deduction and Algorithms.* Cambridge Univ. Press, 1999.

[37] G. Kortsarz, R. Krauthgamer and J. R. Lee. *Hardness of Approximating Vertex-Connectivity Network Design Problems*, SIAM J. of Computing, 33 (3), 704-720, 2004.

[38] T. Kővári, V. T. Sós, P. Turán: On a problem of K. Zarankiewicz, *Colloq. Math.* **3** (1954), 50-57.

[39] M. Langlois, R. H. Sloan, Gy. Turán: Horn Upper Bounds and Renaming, *SAT* (2007), 80-93.

[40] M. Langlois, R. H. Sloan, B. Szörényi, Gy. Turán: Horn Complements: Towards Horn-to-Horn Belief Revision, *AAAI* (2008), 466-471.

[41] M. Langlois, D. Mubayi, R. H. Sloan, Gy. Turán: Combinatorial problems for Horn formulas, in: *Graph Theory, Computational Intelligence and Thought*, Springer LNCS 5420 (2009), to appear.

[42] D. Maier: *The Theory of Relational Databases.* Comp. Sci. Press, 1983.

[43] D.M. Moyles and G.L. Thompson, *Finding a minimum equivalent of a digraph*, JACM, 16 (3), 455-460, 1969.

[44] D. Mubayi and Gy. Turán. *Finding bipartite subgraphs efficiently*, 2009, submitted for publication.

[45] T. A. Nguyen, W. A. Perkins, T. J. Laffey and D. Pecora. *Knowledge Base Verification*, AI Magazine, 8(2), 69-75, Summer 1987.

[46] R. Raz. *A parallel repetition theorem*, SIAM J. of Computing, 27 (3), 763-803, 1998.

[47] S. Russell, P. Norvig: *Artificial Intelligence: A Modern Approach*, 2nd Ed. Prentice Hall, 2002.

[48] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, W. L. Zhu: Open Mind Common Sense: Knowledge Acquisition from the General Public. CoopIS/DOA/ODBASE 2002: 1223-1237

[49] T. Tarján: Complexity of lattice-configurations, *Studia Sci. Math. Hung.* **10** (1975), 203-211.

[50] J. Tepandi. *Comparison of expert system verification criteria: redundancy*, in Validation, Verification and Test of Knowledge-based Systems, M. Ayel and J. P. Laurent (eds), Wiley & Sons, 49-62, 1991.

[51] P. Tison: Generalization of consensus theory and application to the minimization of Boolean functions, *IEEE Trans. on Electr. Comp.* **16** (1967), 446-456.

[52] G. S. Tseitin: On the complexity of derivation in propositional calculus, Seminars in Mathematics, V. A. Steklov Mathematical Institut, Leningrad, Vol. 8 (1968).

[53] Zs. Tuza: Covering of graphs by complete bipartite subgraphs; complexity of 0-1 matrices, *Combinatorica* **4** (1984), 111-116.

[54] C. Umans: Hardness of Approximating Σ_2^p Minimization Problems. *FOCS* (1999), 465-474.

[55] C. Umans: The Minimum Equivalent DNF Problem and Shortest Implicants. *J. Comput. Syst. Sci.* **63** (2001), 597-611.

[56] C. Umans, T. Villa, A. L. Sangiovanni-Vincentelli: Complexity of two-level logic minimization, *IEEE Trans. on CAD of Integrated Circuits and Systems* **25** (2006), 1230-46.

[57] V. V. Vazirani: *Approximation Algorithms.* Springer, 2001.

[58] A. Vetta. *Approximating the minimum strongly connected subgraph via a matching lower bound*, 12th ACM-SIAM Symposium on Discrete Algorithms, 417-426, 2001.

APPENDIX

(containing all proofs omitted from the main body of the paper)

A Proof of Lemma 2

Let $r = \lceil qab/m \rceil$. Since $m \geq 6a \geq 6b > 2eb$ we have $q \geq 1$ and therefore $r \geq 1$. Also, $q = f(a, b, m)$ implies that $r \leq b$. So we have $1 \leq r \leq b$. The algorithm works as follows:

- find R, the set of r highest degree vertices in B and let $H = H(R, A)$ be the bipartite graph between R and A,
- check all q-element subsets Q of R: if vertices in Q have at least q common neighbors in A then output $K_{q,q}$ formed by Q and q such common neighbors and stop.

We need to show that the procedure finds a $K_{q,q}$ and it runs in polynomial time. The first claim follows using the argument of [38]. The bipartite graph H has at least mr/b edges. The number of q-tuples of vertices of R included in the neighborhoods of vertices in A is, using convexity,

$$\sum_{v \in A} \binom{d_H(v)}{q} \geq a \binom{mr/ab}{q} > a \left(\frac{mr}{abq}\right)^q \geq \left(\frac{2re}{q}\right)^q > q\binom{r}{q},$$

where the second inequality is because $r \geq qab/m$ and the third inequality is because $q = f(a, b, m)$. Thus some q-tuple of vertices in R is included in the neighborhood of at least q vertices in A, and so $K_{q,q} \subset H$.

To show that this is a polynomial time algorithm, it suffices to show that $\binom{r}{q}$ is polynomial. Now $\binom{r}{q} < (re/q)^q = e^q(r/q)^q$. Since $q < \ln a$, we have that $e^q < a$. Also

$$(r/q)^q = e^{q \ln(r/q)} < e^{q \ln(2ab/m)} = e^{f(a,b,m) \ln(2ab/m)} < e^{\ln a} = a,$$

where the last inequality follows from the definition of $f(a, b, m)$.

B Proof of Lemma 3

In this case the algorithm is even simpler than that in the proof of Lemma 2, as the same search is performed on G itself. Since $b \geq 3$,

$$q \leq \min\{m/a, f(a, b, m)\}$$

and a similar calculation as in the proof of Lemma 2 gives

$$\sum_{v \in A} \binom{d_G(v)}{q} \geq a \binom{m/a}{q} > a \left(\frac{m}{aq}\right)^q \geq \left(\frac{2be}{q}\right)^q > q\binom{b}{q}.$$

Therefore $K_{q,q}$ is a subgraph of G. To show that the resulting algorithm is polynomial time, we need to show that $\binom{b}{q} < e^q(b/q)^q$ is polynomial. Since $q \leq m/a < f(a, b, m) < \ln a$ we have that $e^q < a$. Also, $q \geq m/2a$ so

$$(b/q)^q = e^{q \ln(b/q)} \leq e^{(m/a) \ln(b/q)} < e^{f(a,b,m) \ln(2ab/m)} \leq a^{\ln(2ab/m)/\ln(2eab/m)} < a.$$

C Proof of Lemma 7

Consider truth assignment setting to 1 every variable in $U \cup U'$, and setting every other variable to 0. This truth assignment satisfies φ but falsifies every clause with body contained in $U \cup U'$ and head outside this set.

D Proof of Theorem 11

The problem is clearly in co-NP. For its co-NP-completeness, a reduction is given from the problem of 2-coloring 3-uniform hypergraphs, proved to be NP-complete by Lovász CITE: given $H = (S, E)$, where S is a set and E is a set of 3-element subsets of S, can the elements of S be assigned the colors *red* and *blue* such that no triple in E is monochromatic?

Let $S = \{a_1, \ldots, a_n\},\ E = \{e_1, \ldots, e_m\}$ and $e_i = \{a_{i,1}, a_{i,2}, a_{i,3}\}$.

Consider the set of variables $X = \{x_1, \ldots, x_n, u\}$, and let $\{y_1, \ldots, y_m, z_1, \ldots, z_m\}$ be additional variables. Let

$$\varphi = \left(\bigwedge_{i=1}^{m} \bigwedge_{j=1}^{3} (x_{i,j} \to y_i) \right) \wedge (y_1, \ldots, y_m \to u) \tag{5}$$

and

$$\psi = \left(\bigwedge_{i=1}^{m} (x_{i,1}, x_{i,2}, x_{i,3} \to z_i) \right) \wedge \left(\bigwedge_{i=1}^{m} \bigwedge_{j=1}^{m} (x_{i,j} \to y_i) \right) \wedge \left(\bigwedge_{i=1}^{m} (z_i, y_1, \ldots, y_m \to u) \right). \tag{6}$$

We claim that H is 3-colorable iff φ and ψ are not X-equivalent.

For indices $\mathbf{j} = (j_1, \ldots, j_m)$, where $1 \leq j_i \leq 3$, let $C_{\mathbf{j}}$ the clause

$$C_{\mathbf{j}} = (x_{1,j_1}, \ldots, x_{m,j_m} \to u).$$

Considering all possible resolutions in φ it follows that the prime implicates of φ containing only variables from X are precisely the clauses of the form $C_{\mathbf{j}}$.

Assume that H is 2-colorable, and let a_{i,j_i} be a red element of e_i, for $i = 1, \ldots, m$. Consider the truth assignment which makes variables x_{i,j_i} and y_i $(i = 1, \ldots, m)$ true and all other variables false. This truth assignment satisfies ψ but falsifies $C_{\mathbf{j}}$ for $\mathbf{j} = (j_1, \ldots, j_m)$. Thus $C_{\mathbf{j}}$ is not an implicate of ψ.

Assume now that H is not 2-colorable, and consider a clause $C_{\mathbf{j}}$ for $\mathbf{j} = (j_1, \ldots, j_m)$. We claim that $C_{\mathbf{j}}$ is an implicate of ψ. Consider the coloring of S where elements a_{i,j_i} are red. Then, as H is not 2-colorable, it must be the case that some edge e_i contains three red elements. The clause $C_{\mathbf{j}}$ is obtained by resolving the clauses $(x_{i,1}, x_{i,2}, x_{i,3} \to z_i), (x_{i,j_i} \to y_i)$ with $(z_i, y_1, \ldots, y_m \to u)$. It also has to be shown that ψ has no other prime implicates besides the clauses $C_{\mathbf{j}}$. This follows by noting that the only way to 'resolve away' the y and z variables is to use some clause $(x_{i,1}, x_{i,2}, x_{i,3} \to z_i)$, some clauses $(x_{i,j_i} \to y_i)$ for every i and the clause $(z_i, y_1, \ldots, y_m \to u)$. All clauses obtained this way are subsumed by some clause of the form $C_{\mathbf{j}}$.

E Proof of Proposition 1

All prime implicates of a CNF can be generated by the variable depletion procedure or Tison's algorithm [51]. This procedure works by processing the variables in some order. For each variable, all resolutions using that variable are performed, and clauses subsumed by other clauses are removed.

Apply the variable depletion procedure to φ starting with variables not in X, and let ψ consist of clauses over X after all variables not in X have been processed. Then ψ is X-equivalent to φ.

The definition of Steiner equivalence implies that the depletions of different variables are independent from each other. The same result is obtained if the depletions are performed in any order, or even in parallel. Also, no new bodies are formed: every body occurring in the depleted formula occurs in the Steiner extension as well.

If we deplete t variables, and the i'th variables occurs c_i times as a head and d_i times as a body, then the depletion of that variables creates $c_i \cdot d_i$ clauses. This implies the size bound given.

F Proof of Corollary 14

The Steiner X-equivalence of φ_1 and φ_2 can be decided by using Proposition 1 to produce formulas $\psi_1(X)$ and $\psi_2(X)$ and checking their equivalence.

G Proof of Proposition 2

The first and last relationships follow from the proof of Proposition 1 and the other follow directly from the definitions.

H Proof of Theorem 19

We use a modification of the gadgets in the reduction in Theorem 4 in order to get a 3-Horn formula. In addition to the variables used there, we also use the following additional variables:

Feedback variables: There are $p-1$ such variables $f_1, \ldots, f_{p-1}$. Let $\mathcal{V}_{\text{feedback}}$ denote the set of feedback variables.

The clauses of φ are the same as in Theorem 4, except s feedback clauses are replaced by a set of $s+p-1$ clauses where the $p-1$ of these clauses form an "inverted" complete binary tree using the feedback variables. Without loss of generality assume p is a power of 2. Then, the new clauses are as follows:

Feedback clauses: Define variables $f_1, \ldots, f_{2p-1}$ with $f_p = e_1, f_{p+1} = e_2, \ldots, f_{2p-1} = e_p$. (Thus $f_p, \ldots, f_{2p-1}$ are only used for indexing, and are not new variables.) Feedback clauses are clauses $f_1 \to u$ for every $u \in A \cup B$ and clauses $f_{2i} \wedge f_{2i+1} \to f_i$ for $1 \le i \le p-1$. There are $s+p$ such clauses and $p-1$ new variables are used. Let $\mathcal{T}_{\text{feedback}}$ be the set of these clauses.

Let $\mathcal{V} = \mathcal{V}_{\text{amplification}} \cup \mathcal{V}_{\text{node}} \cup \mathcal{V}_{\text{super-edge}} \cup \mathcal{V}_{\text{feedback}}$ denote the set of all variables. The total number of variables is $s+2p+t-1$ and the total number of clauses is $st+m+s+p-1$.

The rest of the proof of the theorem has much common with the proof of Theorem 4. We provide all details for the sake of completeness and for the convenience of the reviewers. We reuse the definitions and notations of Theorem 4 whenever necessary.

As φ is definite, all its prime implicates are definite. Also, as φ consists of non-unit definite clauses, all its prime implicates are non-unit (the all-zero vector satisfies φ and falsifies all unnegated variables). For a further analysis of the prime implicates of φ, we make use of the forward chaining procedure described in Section 2.

Lemma 21 *Let x be an amplification variable. Then the prime implicates containing x are clauses of the form $x \to v$, where v is a node or super-edge variable.*

Proof. The variable x cannot be the head of an implicate as the truth assignment setting x to 0 and all other variables to 1 satisfies φ and falsifies such an implicate. Clauses of the form $x \to v$, where v is a node or super-edge variable are implicates as starting from x we can mark first all node variables and then all super-edge variables. If an implicant contains another variable in its body besides x than that variable can be deleted and we still get an implicant. ❑

Lemma 22 *Let U be a set of node variables such that U is not a solution to* Minrep *, and U' be the set of super-edge variables witnessed by U. Then every implicate with body contained in $U \cup U'$ has head in U'.*

Proof. Consider truth assignment setting to 1 every variable in $U \cup U'$, and setting every other variable to 0. This truth assignment satisfies φ but falsifies every clause with body contained in $U \cup U'$ and head outside this set. ❑

Lemma 23 *Let x be an amplification variable and let ψ be a prime and irredundant Horn formula equivalent to φ. Then ψ has at least $OPT(G)$ clauses containing x.*

Proof. Consider the clauses in ψ containing x, and let the set of their heads be U, where $U \subseteq \mathcal{V}_{\text{node}}$. Assume that U is not a solution to MINREP and consider a node variable $v \notin U$. As $x \to v$ is a prime implicate of ψ, the forward chaining procedure over ψ, starting from x, has to mark v. The procedure, first, can mark all node variables in U. As U is not a solution to MINREP , Lemma 7 implies that after that we can mark all super-edges variables witnessed by U and nothing else. Hence v cannot be marked. ❑

Lemma 24 (Gap preserving reduction lemma) **(a)** *If $OPT(G) = \alpha+\beta$, then $OPT(\varphi) \leq t \cdot (\alpha+\beta) + m + s + 2p - 1$.*

(b) *If $OPT(G) \geq (\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ then, $OPT(\varphi) \geq t(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$.*

Proof.
(a) Suppose that there is a solution U of MINREP containing α nodes of A and β nodes of B (and, thus of size $\alpha+\beta$). Then the formula ψ consisting of all witness and feedback clauses, plus all amplification clauses with heads in U, is equivalent to φ. This formula has the required size. In order to show that it is equivalent to φ, we only need to show that implies all amplification clauses $x \to u$ with $u \notin U$. Using ψ, starting from x, we can first mark U. Then, as U is a solution to MINREP , we can mark every super-edge variable using witness clauses, and, finally, we can mark u using a feedback clause with head u.

(b) Suppose that $\text{OPT}(G) \geq (\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ and consider any Horn formula ψ equivalent to φ. Lemma 23 implies that ψ contains at least $t(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ amplification clauses. Thus, $\text{OPT}(\varphi) \geq t(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$. ❑

We are now ready to finish the proof of Theorem 19. Set $t = s^3$ and let L be a ny language in NP and x be an input to L. Then the following holds.

- If $x \in L$, then $\text{OPT}(G) = \alpha + \beta$ which implies $\text{OPT}(\varphi) \leq s^3 \cdot (\alpha + \beta) + m + s + 2p - 1 \leq s^3 \cdot (\alpha + \beta) \cdot (1 + o(1))$.
- If $x \notin L$, then $\text{OPT}(G) \geq (\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$ which implies $\text{OPT}(\varphi) \geq s^3(\alpha + \beta) \cdot 2^{\log^{1-\varepsilon} s}$.

This gives an inapproximability bound of $2^{\log^{1-\varepsilon} s}$ for any $0 < \varepsilon < 1$. Since the number of variables n is given by $n = s^3 + s + m + p - 1 < 4s^3$, this gives us an inapproximability bound of $2^{\log^{1-\varepsilon} n}$.

I Proof of Theorem 20

We first need the *weighted minimum equivalent digraph* problem which is defined as follows. We are given a weighted digraph $G = (V, E, w)$ with an edge weight $w(e) \geq 0$ for every edge $e \in E$. For the digraph G, we use the notation $u \overset{E}{\to} v$ to indicate that E containing a path from u to v and the *transitive closure* of E is the relation $u \overset{E}{\to} v$ over all distinct pairs of nodes of V. Then, the digraph (V, A) is an equivalent digraph for $G = (V, E)$ if **(a)** $A \subseteq E$ and **(b)** transitive closures of A and E are the same. We consider two optimization versions, denoted by W-MIN-ED, resp., W-MAX-ED (and by MIN-ED, resp., MAX-ED if the graph is *unweighted*, *i.e.*, $w(e) = 1$ for every edge e):

- minimize $\sum_{e \in A} W(e)$, and
- maximize $\sum_{e \in E} w(e) - \sum_{e \in A} w(e)$.

The following results are known:

- Both MIN-ED and MAX-ED are MAX-SNP-hard [35].
- MIN-ED and MAX-ED admit 1.5-approximation and 2-approximation, respectively [6, 58].
- W-MIN-ED admits a 2-approximation [25][9].
- W-MAX-ED admits a 2-approximation *if all the weights are zero or one* [6][10].

For an unweighted digraph, if we skip condition **(a)** we obtain the *transitive reduction problem* which was solved in polynomial time by Aho *et al.* [1].

We now prove the following results. By the "Horn maximization" problem, we refer to the problem in which our goal is to maximize the difference in size between the original and the optimized formula.

I.1 Proof of Theorem 20 (a)

Consider an instance $G = (V, E)$ of MIN-ED or MAX-ED. We reduce G to an instance of our problem in a straightforward manner as follows. There is a variable $\mathbf{u}$ for every node u of G. A directed edge (u, v) generates a 2-Horn clause $\mathbf{u} \to \mathbf{v}$. Let φ be the resulting Horn formula.

A solution of MIN-ED or MAX-ED can be identified in an obvious manner with a 2-Horn formula φ' that is equivalent to φ. Every clause $\mathbf{u} \to \mathbf{v}$ in φ' corresponds to an edge $(u, v) \in A$ in the solution of MIN-ED or MAX-ED for G. Using the forward chaining procedure as outlined in Section 2, a clause $\mathbf{u} \to \mathbf{v}$ can be deleted if and only if there is a sequence of clauses $\mathbf{u} \to \mathbf{v}, \mathbf{v} \to \mathbf{v}_1, \mathbf{v}_1 \to \mathbf{v}_2, \ldots, \mathbf{v}_{t-1} \to \mathbf{v}_t, \mathbf{v}_t \to \mathbf{v}$ in M', which is true if and only if the removed edge (u, v) from G still allows a path $(u, v_1), (v_1, v_2), \ldots, (v_{t-1}, v_t), (v_t, v)$ in the solution of MIN-ED or MAX-ED for G. Thus, the solution graph $G = (V, A)$ for MIN-ED or MAX-ED has k edges if and only if the reduced formula φ' has k clauses and the reduction is approximation-preserving.

I.2 Proof of Theorem 20 (b) and (c)

If no unit clauses are part of input, we use the equivalence as outlined in the proof of **(a)**, namely a clause $u \to v$ is identified with a directed edge (u, v) of the graph. This together with the known approximations for MIN-ED and MAX-ED provide the desired 1.5-approximation for restricted minimization for pure 1-Horn formula and 2-approximation for restricted maximization for pure 2-Horn formula.

Otherwise, suppose that our given pure 2-Horn formula φ contains one or more unit clauses. It is easy to see that φ contains at least one satisfying assignment, namely the one in which we set every variable to $\mathbf{1}$.

Consider a given 2-Horn formula φ with n variables and m clauses. We generate an instance $G = (V, W, w)$ of W-MIN-ED or W-MAX-ED with $n + 1$ nodes and $m + n$ edges in the following manner.

- For every variable u of φ, we have a vertex $\mathbf{u}$. We also have an additional vertex $\mathbf{s}$.
- For every clause $u \to v$ in φ_1 (a subformula of φ), we have an edge $(\mathbf{u}, \mathbf{v})$ of weight 1.
- For every clause u in φ_2 (a subformula of φ), we have an edge $(\mathbf{s}, \mathbf{u})$ of weight 1.

[9] Actually, for our purposes, we will only need the special case of the weighted version when all weights are zero or one. Unfortunately, the 2-approximation is still the best known approximation ratio for this special case.

[10] Berman *et al.* call this the MAX-TR$_1$ problem.

- For every variable u, we have an edge $(\mathbf{u}, \mathbf{s})$ of weight *zero.*

An edge generated by a clause is said to be "associated" with that clause and vice versa.

Since all weights of G are 0 or 1, both W-MIN-ED and W-MAX-ED admits a 2-approximation on G. Note that any solution for W-MIN-ED or W-MAX-ED can be trivially augmented, without changing the total weight, such that it includes *all* edges with zero weight, so we will assume that any solution contains *all* edges of zero weights. In other words, all removed edges are of weight 1. Let $\text{OPT}(\varphi)$, $\text{OPT}_{\text{W-Min-ED}}(G)$ and $\text{OPT}_{\text{W-Max-ED}}$ denote the number of clauses in an optimal solution φ^{opt} of φ, the number of edges $|A|$ in an optimal solution (V, A) of W-MIN-ED and the number of removed edges $|E \setminus A|$ in an optimal solution (V, A) of W-MAX-ED, respectively. Let $\text{OPT}(\varphi) = m_1^{\text{opt}} + m_2^{\text{opt}}$ where m_i^{opt} is the number of clauses from the group φ_i.

We associate a 1-1 mapping between a solution φ' of the minimization or maximization problem for the pure 2-Horn formula φ and W-MIN-ED/W-MAX-ED on G in the obvious manner: a clause is in φ' if and only if its associated edge is in (V, A).

Lemma 25 *Consider any valid solution* (V, A) *of* W-MIN-ED *or* W-MAX-ED *of* G*. Then the following statements hold:*

(a) *If there is a path from* $\mathbf{u}$ *to* $\mathbf{v}$ *in* (V, A)*, then there is also a path from* $\mathbf{u}$ *to* $\mathbf{v}$ *in* (V, A) *in which, except the first edge, no other edge is of the form* $(\mathbf{x}, \mathbf{s})$ *for some* $\mathbf{x}$*.*

(b) *If there is a path from* $\mathbf{s}$ *to* $\mathbf{v}$ *in* (V, A)*, then there is also a path from* $\mathbf{s}$ *to* $\mathbf{v}$ *in* (V, A) *in which, except the first edge, no other edge is of the form* $(\mathbf{s}, \mathbf{x})$ *for some* $\mathbf{x}$*.*

Proof. Remember that A contains all edges of weight zero.
(a) Suppose not. Then, we can "short-cut" the path from $\mathbf{u}$ to $\mathbf{x}$ to $\mathbf{s}$ by using the zero-weight edge $(\mathbf{u}, \mathbf{s}) \in A$.
(b) Suppose not. Then, we can remove the "prefix" of the path from $\mathbf{s}$ to $\mathbf{s}$. ❑

Lemma 26 *Consider a associated pairs of solutions* φ' *and* (V, A)*. Then,* φ' *is a valid solution for* φ *if and only if* (V, A) *is a valid solution for* G*.*

Proof. Suppose that (V, A) is a valid solution for G. We need to show that φ' is indeed equivalent to φ, *i.e.*, every clause C in φ' but not in φ is redundant for φ'. There are two cases to consider.

First, suppose that C is a clause of the form $u \to v$ from φ_1. Thus, $(\mathbf{u}, \mathbf{v}) \notin A$. But, there must be a path $\mathcal{P}$ from $\mathbf{u}$ to $\mathbf{v}$ in (V, A). By Lemma 25, there is such a path $\mathcal{P}$ such that except the first edge, no other edge in $\mathcal{P}$ is of the form $(\mathbf{x}, \mathbf{s})$ for some $\mathbf{x}$. Thus, there are only two cases to consider, depending on whether the first edge of $\mathcal{P}$ is $(\mathbf{u}, \mathbf{s})$ or not.

If the first edge of $\mathcal{P}$ is not $(\mathbf{u}, \mathbf{s})$, then $\mathcal{P}$ is of the form $(\mathbf{u}, \mathbf{x}_1), (\mathbf{x}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_t, \mathbf{v})$ for some nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t \in V \setminus \{\mathbf{s}\}$. Then, φ' contains the associated clauses $u \to x_1, x_1 \to x_2, x_t \to v$. Obviously, these clauses imply the clause $u \to v$.

If the first edge of $\mathcal{P}$ is $(\mathbf{u}, \mathbf{s})$, then $\mathcal{P}$ is of the form $(\mathbf{u}, \mathbf{s}), (\mathbf{s}, \mathbf{x}_1), (\mathbf{x}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_t, \mathbf{v})$ for some nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t \in V \setminus \{\mathbf{s}\}$. Then, φ' contains the associated clauses $x_1, x_1 \to x_2, x_t \to v$. Consider the forward chaining procedure of Section 2. Because the left part of the clause x_1 is empty, x_1 will be added to S. Then, we progressively add $x_2, x_3, \ldots, x_t$ and finally v to S. Thus, the clause $u \to v$ is redundant.

Now, we prove the other direction of implication. Suppose φ' is a valid solution for φ. We need to show that (V, A) is a valid solution for G. Let C be a clause in φ but not in φ'. Thus, C is a redundant clause for M'. There are two cases to consider depending on the nature of the clause C.

Suppose that C is a clause of the form $u \to v$. the forward chaining procedure of Section 2 implies that C is redundant if and only if one of the following two conditions hold:

- φ' contains the clauses $u \to x_1, x_1 \to x_2, x_t \to v$ for some $x_1, x_2, \ldots, x_t$. Then, (V, A) contains the path $(\mathbf{u}, \mathbf{x}_1), (\mathbf{x}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_t, \mathbf{v})$ and indeed $(\mathbf{u}, \mathbf{v})$ is redundant.

- φ contains the clauses $x_1, x_1 \to x_2, x_t \to v$ for some $x_1, x_2, \ldots, x_t$. Then, (V, A) contains the path $(\mathbf{u}, \mathbf{s}), (\mathbf{s}, \mathbf{x}_1), (\mathbf{x}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_t, \mathbf{v})$ and indeed $(\mathbf{u}, \mathbf{v})$ is redundant.

Otherwise, suppose that C is an unit clause, say u. Then, φ' contains the clauses $x_1, x_1 \to x_2, x_t \to u$ for some $x_1, x_2, \ldots, x_t$. Obviously, x_1 is not u. Then, (V, A) contains the path $(\mathbf{s}, \mathbf{x}_1), (\mathbf{x}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_t, \mathbf{u})$ from s to u. ❑

Note that for an associated pair of solutions φ' and (V, A), $|\varphi'| = \sum_{e \in A} w(e)$. Thus, a 2-approximation for W-Min-ED implies a 2-approximation for restricted minimization of pure 2-Horn formula via Lemma 26. Moreover, $|\varphi| = \sum_{e \in E} w(e)$, and thus a 2-approximation for W-Max-ED implies a 2-approximation for restricted minimization of pure 2-Horn formula via Lemma 26.