Preliminary Report on Input Cover Number as a Metric for Propositional Resolution Proofs

Allen Van Gelder

University of California, Santa Cruz CA 95060, USA,
WWW home page: `http://www.cse.ucsc.edu/~avg`

Abstract. Input Cover Number (denoted by κ) is introduced as a metric for difficulty of propositional resolution derivations. If $\mathcal{F} = \{C_i\}$ is the input CNF formula, then $\kappa_{\mathcal{F}}(D)$ is defined as the minimum number of clauses C_i needed to form a superset of (i.e., cover) clause D. Input Cover Number provides a refinement of the clause-width metric in the sense that it applies to families of formulas whose clause width grows with formula size, such as pigeon-hole formulas $\mathrm{PHP}(m, n)$ and $\mathrm{GT}(n)$. Although these two families have much different general-resolution complexities, it is known that both require $\Theta(n)$ clause width (after transforming to 3-CNF). It is shown here that κ is $\Theta(n)$ for pigeon-hole formulas and is $\Theta(1)$ for $\mathrm{GT}(n)$ formulas and variants of $\mathrm{GT}(n)$.

1 Introduction

Ben-Sasson and Wigderson showed that, if the minimum-length general resolution refutation for a CNF formula $\mathcal{F}$ has S steps, and if the minimum-length tree-like refutation of $\mathcal{F}$ has S_T steps, then there is a (possibly different) refutation of $\mathcal{F}$ using clauses of width at most:

$$w(\mathcal{F} \vdash \bot) \leq w(\mathcal{F}) + c\sqrt{n \ln S}; \tag{1}$$

$$w(\mathcal{F} \vdash \bot) \leq w(\mathcal{F}) + \lg S_T. \tag{2}$$

where $\mathcal{F}$ has n variables and $w(\mathcal{F} \vdash \bot)$ denotes resolution-refutation width. The $w(\mathcal{F})$ terms were omitted from their statement in the introduction, but appear in the theorems [3].

Our first results essentially eliminate the $w(\mathcal{F})$ terms in the Ben-Sasson and Wigderson theorems, and replace resolution width by $\kappa_{\mathcal{F}}(\pi)$, the *input cover number*, as defined below.

Our interest in input cover number stems from the indications that it separates polynomial families from super-polynomial families for a wide class of formulas that represent SAT encodings of *constraint satisfaction* problems.

Two prototypical and widely studied examples are the pigeon-hole family $\mathrm{PHP}(n+1, n)$ and the $\mathrm{GT}(n)$ family. Both families have a similar appearance: $\Theta(n)$ clause width, $\Theta(n^2)$ propositional variables, $\Theta(n^3)$ clauses, and $\Theta(n^3)$ overall formula length. However, the pigeon-hole family has minimum resolution length in $\Omega(2^n)$ [6, 3], whereas the $\mathrm{GT}(n)$ family has minimum resolution length in $O(n^3)$ [9, 4]. The clause-width metric does not distinguish between these two families: after the standard transformations into 3-CNF, giving $\mathrm{EPHP}(n+1,n)$ and $\mathrm{MGT}(n)$, they both have lower bounds for $w(\mathcal{F} \vdash \bot)$ in $\Omega(n)$ [3, 4]. The input distance metric [10] also does not distinguish them. We show that the input-cover-number metric distinguishes sharply between them: $\kappa(\mathrm{PHP}(n+1,n) \vdash \bot)$ is in $\Theta(n)$, whereas $\kappa(\mathrm{GT}(n) \vdash \bot)$ is in $\Theta(1)$.

Definition 1.1. (input cover number) All clauses mentioned are nontautologous sets of literals. Let D be a clause; let C be a clause of formula $\mathcal{F}$. The *input cover number* of D w.r.t. $\mathcal{F}$, denoted $\kappa_{\mathcal{F}}(D)$, is the minimum number of clauses $C_i \in \mathcal{F}$ such that $D \subseteq \bigcup_i C_i$, i.e., the cardinality of the minimum set cover.

For a resolution proof π $\kappa_{\mathcal{F}}(\pi)$ is the maximum over $D \in \pi$ of the *input cover numbers* of D w.r.t. $\mathcal{F}$.

When $\mathcal{F}$ is understood from the context, $\kappa(D)$ and $\kappa(\pi)$ are written. $\kappa(\mathcal{F} \vdash D)$ denotes the minimum of $\kappa_{\mathcal{F}}(\pi)$ over all π that are derivations of D from $\mathcal{F}$.

The theorems shown in the full paper[1] are that, if π is a resolution refutation of $\mathcal{F}$ and π uses all clauses of $\mathcal{F}$ and the length of π is S, then there is a refutation of $\mathcal{F}$ using clauses that have *input cover number* w.r.t. $\mathcal{F}$ that is at most:

$$\kappa(\mathcal{F} \vdash \bot) \leq c\sqrt{n \ln S}; \tag{3}$$

$$\kappa(\mathcal{F} \vdash \bot) \leq \lg S_T. \tag{4}$$

Proofs and additional details may be found in full paper.

Also, we show that the pigeon-hole family of formulas $\mathrm{PHP}(m,n)$ require refutations with input cover number $\Omega(n)$, although they contain clauses of width n. This result suggests that input cover number provides a refinement of the clause-width metric as a measure of resolution difficulty. That is, when a family of formulas with increasing clause-width, such as $\mathrm{PHP}(m,n)$, is transformed into a bounded-width family, such as $\mathrm{EPHP}(m,n)$, and the bounded-width family has large resolution width, this is not simply because they rederive the wide clauses of the original family, then proceed to refute the original family. Rather, it is the case that wide clauses substantially different from those in the original family must be derived.

Although the results are promising in some cases, the input-cover-number metric has an inherent fragility. Although $\kappa(\mathrm{GT}(n) \vdash \bot)$ is in $\Theta(1)$ for the natural encoding of $\mathrm{GT}(n)$, for 3-CNF variant, $\kappa(\mathrm{MGT}(n) \vdash \bot)$ is necessarily the same order of magnitude as the clause-width lower bound, $w(\mathrm{MGT}(n) \vdash \bot)$, i.e., in $\Omega(n)$.

Recall that Bonet and Galesi showed that $w(\mathrm{MGT}(n) \vdash \bot)$ is in $\Omega(n)$, yet $\mathrm{MGT}(n)$ has a refutation in $\Theta(n^3)$ [4]. Due to the fragility of κ mentioned in the previous paragraph, the following attractive conjecture must *fail*: If a family has κ in $\Omega(n)$, its refutation length must be super-polynomial in n. The full paper discusses fragilities of κ in greater length.

[1] See `http://www.cse.ucsc.edu/~avg/Papers/cover-number.{pdf,ps}`.

Table 1. Summary of notations.

$a, \ldots, z$	Literal; i.e., propositional variable or negated propositional variable.
$A, \ldots, Z$	Disjunctive clause, or set of literals, depending on context.
$\mathcal{A}, \ldots, \mathcal{H}$	CNF formula, or set of literals, depending on context.
π	Resolution derivation DAG.
$[p_1, \ldots, p_k]$	Clause consisting of literals $p_1, \ldots, p_k$.
$\bot$, $\top$	*empty clause*, *tautologous clause*.
$\alpha, \ldots, \delta$	Subclause, in the notation $[p, q, \alpha]$.
C^-	Read as "C, or some clause that subsumes C".
$\mathbf{res}(q, C, D)$	Resolvent of C and D, where q and $\neg q$ are the clashing literals (see Definition 2.1).
$C\vert\mathcal{A}$, $\mathcal{F}\vert\mathcal{A}$, $\pi\vert\mathcal{A}$	C (respectively $\mathcal{F}$, π) *restricted* by $\mathcal{A}$ (see Definition 2.3).

2 Preliminaries

Notations are summarized in Table 1. Although the general ideas of resolution and derivations are well known, there is no standard notation for many of the technical aspects, so it is necessary to specify our notation in detail.

Defining resolution as a total function removes the need to include the weakening rule in the proof system. Numerous proof complexity papers include the weakening rule as a crutch to handle "life after restrictions" [3, 4, 1]. However, according to Alasdair Urquhart, the weakening rule might add power to some resolution strategies, such as linear resolution. See Table 1 for the notation of the resolution operator, which satisfies these symmetries:

$$\mathbf{res}(q, C, D) = \mathbf{res}(q, D, C) = \mathbf{res}(\neg q, C, D) = \mathbf{res}(\neg q, D, C).$$

Definition 2.1. (resolution, tautologous) A clause is *tautologous* if it contains complementary literals. All tautologous clauses are considered to be indistinguishable and are denoted by $\top$.

Fix a total order on the clauses definable with the n propositional variables such that $\bot$ is smallest, $\top$ is largest, and wider clauses are "bigger" than narrower clauses. Other details of the total order are not important. The following table, in which α and β denote clauses that do not contain q or $\neg q$, extends resolution to a total function:

C	D	$\mathbf{res}(q, C, D)$
$[q, \alpha]$	$[\neg q, \beta]$	$[\alpha, \beta]$
$[\gamma]$	$\top$	$[\gamma]$
$[\alpha]$	$[\neg q, \beta]$	$[\alpha]$
$[\alpha]$	$[\beta]$	smaller of α, β

□

With this generalized definition of resolution, we have an algebra, and the set of clauses (including $\top$) is a lattice. Now resolution "commutes up to subsumption" with *restriction* (see Definition 2.3), so restriction can be applied to any resolution derivation to produce another derivation.

Definition 2.2. (derivation, refutation) A *derivation* (short for *propositional resolution derivation*) from formula $\mathcal{F}$ is a *rooted*, directed acyclic graph (DAG) in which each vertex is labeled with a clause and, unless it is a *leaf* ($C \in \mathcal{F}$), it is also labeled with a clashing literal and has two out-edges. □

Definition 2.3. (restricted formula, restricted derivation) Let $\mathcal{A}$ be a partial assignment for formula $\mathcal{F}$. Let π be a derivation from $\mathcal{F}$. Read "$|\mathcal{A}$" as "restricted by $\mathcal{A}$".

1. $C|\mathcal{A} = \top$, if C contains any literal that occurs in $\mathcal{A}$, otherwise $C|\mathcal{A} = C - \{\neg q \mid q \in \mathcal{A}\}$.
2. $\mathcal{F}|\mathcal{A}$ results by applying restriction to each clause in $\mathcal{F}$.
3. $\pi|\mathcal{A}$ is defined differently from most previous papers. It is the same DAG as π structurally, but the clauses labeling the vertices are changed as follows. If a leaf (input clause) of π contains C, then the corresponding leaf of $\pi|\mathcal{A}$ contains $C|\mathcal{A}$. Each derived clause of $\pi|\mathcal{A}$ uses resolution on the same clashing literal as the corresponding vertex of π. □

Lemma 2.4. Given formula $\mathcal{F}$, and a restriction literal p,

$$\mathbf{res}(q, D_1|p, D_2|p) \subseteq \mathbf{res}(q, D_1, D_2)|p.$$

Lemma 2.5. Given formula $\mathcal{F}$, and a restriction literal p, if π is a derivation of C from $\mathcal{F}$, then $\pi|p$ is a derivation of $(C|p)^-$ (a clause that subsumes $C|p$) from $\mathcal{F}|p$.

Lemma 2.6. Let C be a clause of $\mathcal{F}$ and let $\mathcal{A}$ be a partial assignment. If $C|\mathcal{A} \neq \top$ (i.e., $\mathcal{A}$ does not satisfy C), then $\kappa_{\mathcal{F}}(C|\mathcal{A}) = 1$.

Lemma 2.7. Let D be a clause of $\mathcal{F}$, let $\mathcal{A}$ be a partial assignment, and let $\mathcal{G} = \mathcal{F}|\mathcal{A}$. If $D|\mathcal{A} \neq \top$ (i.e., $\mathcal{A}$ does not satisfy D), then $\kappa_{\mathcal{F}}(D) \leq \kappa_{\mathcal{G}}(D|\mathcal{A}) + |\mathcal{A}|$.

3 Size vs. Input Cover Number Relationships

Ben-Sasson and Wigderson [3] derived size-width relationships that they describe as a "direct translation of [CEI96] to resolution derivations." Their informal statement, "if $\mathcal{F}$ has a *short* resolution refutation then it has a refutation with a small *width*," applies only when $\mathcal{F}$ has no wide clauses.

This section shows that by using input cover number rather than clause width, the restriction on the width of $\mathcal{F}$ can be removed. That is, the relationships are strengthened by removing the additive term, *width*($\mathcal{F}$).

The use of restriction for recursive construction of refutations with special properties originates with Anderson and Bledsoe [2], and has been used by numerous researchers subsequently [5, 3, 10]. We use it to construct resolution refutations of small input cover number.

Lemma 3.1. Let $\mathcal{G} = \mathcal{F}|p$. If derivation π_1 derives clause D from $\mathcal{G}$ with $\kappa_{\mathcal{G}}(\pi_1) = (d-1)$, then there is a derivation π_2 that derives $(D + \neg p)^-$ from $\mathcal{F}$ with $\kappa_{\mathcal{F}}(\pi_2) \leq d$.

Lemma 3.2. Let $\mathcal{G} = \mathcal{F}|p$ and $\mathcal{H} = \mathcal{F}|\neg p$. If derivation π_1 derives $\perp$ from $\mathcal{G}$ with $\kappa_{\mathcal{G}}(\pi_1) = d-1$, and derivation π_2 derives $\perp$ from $\mathcal{H}$ with $\kappa_{\mathcal{H}}(\pi_2) = d$, then there is a derivation π_3 that derives $\perp$ from $\mathcal{F}$ with $\kappa_{\mathcal{F}}(\pi_3) \leq d$.

Theorem 3.3. Let $\mathcal{F}$ be an unsatisfiable formula on $n \geq 1$ variables and let $d \geq 0$ be an integer. Let S_T be the size of the shortest tree-like refutation of $\mathcal{F}$. If $S_T \leq 2^d$, then $\mathcal{F}$ has a refutation π with $\kappa_{\mathcal{F}}(\pi) \leq d$.

Corollary 3.4. $S_T(\mathcal{F}) \geq 2^{\kappa(\mathcal{F} \vdash \perp)}$.

Theorem 3.5. Let $\mathcal{F}$ be an unsatisfiable formula on $n \geq 1$ variables and let $d \geq 0$ be an integer. Let $S(\mathcal{F})$ be the size of the shortest refutation of $\mathcal{F}$. If $S(\mathcal{F}) \leq e^{(d^2/8n)}$, then $\mathcal{F}$ has a refutation π_1 with $\kappa_{\mathcal{F}}(\pi_1) \leq d$.

Corollary 3.6. $S(\mathcal{F}) \geq e^{(\kappa(\mathcal{F} \vdash \perp)^2/8n)}$.

4 Pigeon-Hole Formulas

The well-known family of Pigeon-Hole formulas for m pigeons and n holes ($\mathrm{PHP}(m, n)$) is defined by these clauses:

$$\begin{aligned} C_i &= [x_{i,1}, \ldots, x_{i,n}] && \text{for } 1 \leq i \leq m \\ B_{ijk} &= [\neg x_{i,k}, \neg x_{j,k}] && \text{for } 1 \leq i \leq m,\ 1 \leq j \leq m,\ 1 \leq k \leq n. \end{aligned}$$

Theorem 4.1. Any refutation of $\mathrm{PHP}(m, n)$ with $m > n$ has input cover number at least $n/6$.

5 The GT(n) Family

The GT(n) family was conjectured to require exponential length refutations [7], but Stålmarck demonstrated the first polynomial solution, then Bonet and Galesi found another [9, 4]. Both of these solutions produce derived clauses of width about double that of the input and have input cover numbers of two. The full paper describes a refutation with input cover number 3, which also has no derived clause wider than an input clause. This new refutation is half as long as those previously published.

Definition 5.1. The clauses of GT(n) are named as follows for indexes indicated.

$$\begin{aligned} C_n(j) &\equiv [\langle 1, j\rangle, \ldots, \langle j-1, j\rangle, \langle j+1, j\rangle, \ldots, \langle n, j\rangle] && 1 \leq j \leq n \\ B(i,j) &\equiv [\neg\langle i, j\rangle, \neg\langle j, i\rangle] && 1 \leq i < j \leq n \\ A(i,j,k) &\equiv [\neg\langle i, j\rangle, \neg\langle j, k\rangle, \langle i, k\rangle] && 1 \leq i, j, k \leq n \text{ and } i, j, k \text{ distinct.} \end{aligned}$$

We recursively construct a refutation with input cover number 3, which also limits derived clause width to that of the input. The base case is GT(1), in which $C_1(1) = \perp$. For GT(n), where $n > 1$, the refutation begins by deriving $C_{n-1}(m)$ for $1 \leq m \leq n-1$. Then GT($n-1$) is refuted. The subderivation of $C_{n-1}(m)$ from $C_n(m)$, $C_n(n)$, $B(i,j)$ and $A(i,j,k)$ begins by resolving $C_n(n)$ with $B(m,n)$. This is the key difference from earlier published refutations, and introduces $\neg\langle n,m\rangle$ in place of $\langle m,n\rangle$. Then $\langle i,n\rangle$ are replaced one by one with $\langle i,m\rangle$ by resolving with $A(i,n,m)$. Finally, $\neg\langle n,m\rangle$ is removed by subsumption resolution with $C_n(m)$.

6 Conclusion

We proposed the *input cover number* metric (κ) as a refinement of clause width and input distance for studying the complexity of resolution. For families with wide clauses, the trade-off between resolution refutation size and κ is sharper than the trade-off between resolution refutation size and clause width.

The GT(n) family has exponential tree-like refutations [4], and can be modified so that regular refutations are also exponential [1]. The original and modified families have $\kappa = 3$. These results suggest (very tentatively) that κ might be the sharper metric for general resolution, while clause-width is sharper for tree-like resolution.

References

1. Alekhnovich, M., Johannsen, J., Pitassi, T., Urquhart, A.: An exponential separation between regular and unrestricted resolution. In: Proc. 34th ACM Symposium on Theory of Computing. (2002) 448–456
2. Anderson, R., Bledsoe, W.W.: A linear format for resolution with merging and a new technique for establishing completeness. Journal of the ACM **17** (1970) 525–534
3. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow — resolution made simple. JACM **48** (2001) 149–168
4. Bonet, M., Galesi, N.: Optimality of size-width tradeoffs for resolution. Computational Complexity **10** (2001) 261–276
5. Clegg, M., Edmonds, J., Impagliazzo, R.: Using the Groebner basis algorithm to find proofs of unsatisfiability. In: Proc. 28th ACM Symposium on Theory of Computing. (1996) 174–183
6. Haken, A.: The intractability of resolution. Theoretical Computer Science **39** (1985) 297–308
7. Krishnamurthy, B.: Short proofs for tricky formulas. Acta Informatica **22** (1985) 253–274
8. Letz, R., Mayr, K., Goller, C.: Controlled integration of the cut rule into connection tableau calculi. Journal of Automated Reasoning **13** (1994) 297–337
9. Stålmarck, G.: Short resolution proofs for a sequence of tricky formulas. Acta Informatica **33** (1996) 277–280
10. Van Gelder, A.: Lower bounds for propositional resolution proof length based on input distance. In: Eighth International Conference on Theory and Applications of Satisfiability Testing, St. Andrews, Scotland (2005)