Solving MaxSAT and #SAT on Structured CNF Formulas

Sigve Hortemo Sæther, Jan Arne Telle, and Martin Vatshelle

Department of Informatics, University of Bergen, Norway

Abstract. In this paper we propose a structural parameter of CNF formulas and use it to identify instances of weighted MaxSAT and #SAT that can be solved in polynomial time. Given a CNF formula we say that a set of clauses is projection satisfiable if there is some complete assignment satisfying these clauses only. Let the ps-value of the formula be the number of projection satisfiable sets of clauses. Applying the notion of branch decompositions to CNF formulas and using ps-value as cut function, we define the ps-width of a formula. For a formula given with a decomposition of polynomial ps-width we show dynamic programming algorithms solving weighted MaxSAT and #SAT in polynomial time. Combining with results of ‘Belmonte and Vatshelle, Graph classes with structured neighborhoods and algorithmic applications, THEOR. COMPUT. SCI. 511: 54-65 (2013)’ we get polynomial-time algorithms solving weighted MaxSAT and #SAT for some classes of structured CNF formulas. For example, we get $O(m^2(m+n)s)$ algorithms for formulas $F$ of $m$ clauses and $n$ variables and total size $s$, if $F$ has a linear ordering of the variables and clauses such that for any variable $x$ occurring in clause $C$, if $x$ appears before $C$ then any variable between them also occurs in $C$, and if $C$ appears before $x$ then $x$ occurs also in any clause between them. Note that the class of incidence graphs of such formulas do not have bounded clique-width.

1 Introduction

Given a CNF formula, propositional model counting (#SAT) is the problem of computing the number of satisfying assignments, and maximum satisfiability (MaxSAT) is the problem of determining the maximum number of clauses that can be satisfied by some assignment. Both problems are significantly harder than simply deciding if a satisfying assignment exists. #SAT is #P-hard [11] even when restricted to Horn 2-CNF formulas, and to monotone 2-CNF formulas [22]. MaxSAT is NP-hard even when restricted to Horn 2-CNF formulas [15], and to 2-CNF formulas where each variable appears at most 3 times [20]. Both problems become tractable under certain structural restrictions obtained by bounding width parameters of graphs associated with formulas, see for example [9, 10, 23, 25]. For earlier work on width decompositions in this setting see e.g. [31]. The work we present here is inspired by the recent results of Paulusma
et al. [18] and Slivovsky and Szeider [24] showing that \#SAT is solvable in polynomial time when the incidence graph $I(F)$ of the input formula $F$ has bounded modular treewidth, and more strongly, bounded symmetric clique-width.

We extend these results in several ways. We give algorithms for both \#SAT and MAXSAT, and also weighted MAXSAT, finding the maximum weight of satisfiable clauses, given a set of weighted clauses. We introduce the parameter ps-width, and express the runtime of our algorithms as a function of ps-width.

**Theorem 3.** Given a formula $F$ over $n$ variables and $m$ clauses and of total size $s$, and a decomposition of $F$ of ps-width $k$, we solve \#SAT, and weighted MAXSAT in time $O(k^3 s (m + n))$.

Thus, given a decomposition having a ps-width $k$ that is polynomially-bounded in the number of variables $n$ and clauses $m$ of the formula, we get polynomial-time algorithms. These are dynamic programming algorithms similar to the one given for \#SAT in [24], but we believe that the ps-width parameter is a better measure of the inherent runtime bottleneck of \#SAT and MAXSAT when using this type of dynamic programming. The essential combinatorial result enabling this improvement is Lemma 5 of this paper. The algorithm of [24] solves \#SAT in time $(n + m)^{O(w)}$ for $w$ being the symmetric clique-width of the decomposition, and is thus a polynomial-time algorithm if given a decomposition with constantly bounded $w$. The result of Theorem 3 encompasses this, since we show via the concept of MIM-width [26], that any formula with constantly bounded symmetric clique-width also has polynomially bounded ps-width.

We show that a relatively rich class of formulas, including classes of unbounded clique-width, have polynomially bounded ps-width. This is shown using the concept of MIM-width of graphs, introduced in the thesis of Vatshelle [26]. See Figure 1. In particular, this holds for classes of formulas having incidence graphs that can be represented as intersection graphs of certain objects, like interval graphs [2]. We prove this also for bigraph bipartizations of these graphs, which are obtained by imposing a bipartition on the vertex set and keeping only edges between the partition classes. Some such bigraph bipartizations have been studied previously, in particular the interval bigraphs. The interval bigraphs contain all bipartite permutation graphs, and these latter graphs have been shown to have unbounded clique-width [4].

By combining an alternative definition of interval bigraphs [13] with a fast recognition algorithm [17,19] we arrive at the following. Say that a CNF formula $F$ has an interval ordering if there exists a linear ordering of variables and clauses such that for any variable $x$ occurring in clause $C$, if $x$ appears before $C$ then any variable between them also occurs in $C$, and if $C$ appears before $x$ then $x$ occurs also in any clause between them.

**Theorem 10.** Given a CNF formula $F$ over $n$ variables and $m$ clauses and of total size $s$, we can in time $O((m + n)s)$ decide if $F$ has an interval ordering (yes iff $I(F)$ is an interval bigraph), and if yes we solve \#SAT and weighted MAXSAT with a runtime of $O(m^2 (m + n)s)$.