Chapter 10
Algorithms for Maximum Satisfiability Using Unsatisfiable Cores
Joao Marques-Sila and Jordi Planes

Abstract Many decision and optimization problems in electronic design automation (EDA) can be solved with Boolean satisfiability (SAT). These include binate covering problem (BCP), pseudo-Boolean optimization (PBO), quantified Boolean formulas (QBF), multi-valued SAT, and, more recently, maximum satisfiability (MaxSAT). The first generation of MaxSAT algorithms are known to be fairly inefficient in industrial settings, in part because the most effective SAT techniques cannot be easily extended to MaxSAT. This chapter proposes a novel algorithm for MaxSAT that improves existing state-of-the-art solvers by orders of magnitude on industrial benchmarks. The new algorithm exploits modern SAT solvers, being based on the identification of unsatisfiable subformulas. Moreover, the new algorithm provides additional insights between unsatisfiable subformulas and the maximum satisfiability problem.

10.1 Introduction

Boolean satisfiability (SAT) is used for solving an ever increasing number of decision and optimization problems in electronic design automation (EDA). These include model checking, equivalence checking, design debugging, logic synthesis, and technology mapping [8, 19, 34, 36]. Besides SAT, a number of well-known extensions of SAT also find application in EDA, including pseudo-Boolean optimization (PBO) (e.g., [27]), quantified Boolean formulas (QBF) (e.g., [13]), multi-valued SAT [26], and, more recently, maximum satisfiability (MaxSAT) [33].

MaxSAT is a well-known problem in computer science, consisting of finding the largest number of satisfied clauses in unsatisfiable instances of SAT. Algorithms for MaxSAT are in general not effective for large industrial problem instances, in part...
because the most effective SAT techniques cannot be applied directly to MaxSAT [9] (e.g., unit propagation).

Motivated by the recent and promising application of MaxSAT in EDA (e.g., [33]) this chapter proposes a novel algorithm for MaxSAT, msu4, that performs particularly well for large industrial instances. Instead of the usual algorithms for MaxSAT, the proposed algorithm exploits existing SAT solver technology and the ability of SAT solvers for finding unsatisfiable subformulas. Despite building on the work of others, on the relationship between maximally satisfiable and minimally unsatisfiable subformulas [6, 16, 20, 21, 24], the approach outlined in this chapter is new, in that unsatisfiable subformulas are used for guiding the search for the solution to the MaxSAT problem. The msu4 algorithm builds on recent algorithms for the identification of unsatisfiable subformulas, which find other significant applications in EDA [32, 37]. The msu4 algorithm also builds on recent work on solving PBO with SAT [15], namely on techniques for encoding cardinality constraints as Boolean circuits obtained from BDDs. The msu4 algorithm differs from the one in [16] in the way unsatisfiable subformulas are manipulated and in the overall organization of the algorithm.

Experimental results, obtained on representative EDA industrial instances, indicate that in most cases the new msu4 algorithm is orders of magnitude more efficient than the best existing MaxSAT algorithms. The msu4 also opens a new line of research that tightly integrates SAT, unsatisfiable subformulas, and MaxSAT.

The chapter is organized as follows. The next section provides a brief overview of MaxSAT and existing algorithms. Section 10.3 describes the msu4 algorithm and proves the correctness of the proposed approach. Section 10.4 provides experimental results, comparing msu4 with alternative MaxSAT algorithms. The chapter concludes in Section 10.6.

10.2 Background

This section provides definitions and background knowledge for the MaxSAT problem. Due to space constraints, familiarity with SAT and related topics is assumed and the reader is directed to the bibliography [10].

10.2.1 The MaxSAT Problem

The maximum satisfiability (MaxSAT) problem can be stated as follows. Given an instance of SAT represented in CNF, compute an assignment that maximizes the number of satisfied clauses. During the last decade there has been a growing interest on studying MaxSAT, motivated by an increasing number of practical applications, including scheduling, routing, bioinformatics, and EDA [33].

Despite the clear relationship with the SAT problem, most modern SAT techniques cannot be applied directly to the MaxSAT problem. As a result, most