# Chapter 1

# Introduction

## 1.1 Introduction

Since the introduction of Computer Science in our society, computer scientists have tried to model real-life problems and solve them using the power of the computer systems. Nevertheless, many problems are known to be *computationally intractable*, in the sense that, unless P=NP, the algorithms for solving them require an exponential number of steps in the length of the input for some of their instances. In this case, the challenge for computer scientists is to devise algorithms that solve as many instances of intractable problems as possible in a reasonable amount of time.

Among the intractable problems, the Boolean satisfiability problem (SAT) has been considered a central problem in Artificial Intelligence, Electronic Design Automation and Theoretical Computer Science, and nowadays it is commonly acknowledged that solving combinatorial *decision* problems via their reduction to SAT is one of the best performing problem solving approaches. SAT has shown to be competitive in a variety of domains, including hardware verification [eSSMS99, MMZ$^+$01, VB01, BK02, KSHK07], bioinformatics [LMS06b, LMS06a], planning [KS96, Kau06], and scheduling [BM00, ZLS04].

In this thesis we investigate different Max-SAT problems, which are optimization versions of SAT that, despite not being so well-studied as SAT, have seen an increasing activity in the community working on satisfiability problems [LM09]. Even an evaluation of Max-SAT solvers is organized since 2006 as a co-located event of the International Conference on Theory and Applications of Satisfiability Testing.

Our research program aims at converting Max-SAT formalisms into a competitive generic problem solving approach for solving combinatorial *optimization* problems, and in particular, converting them into competitive approaches for solving over-constrained problems with soft and hard constraints. To this end, in this thesis, we study Max-SAT formalisms that incorporate the notion of partiality, and design and implement solving techniques for such formalisms. Our empirical investigation provides evidence that the solvers that we have developed exhibit a good performance profile on a wide collection of benchmarks.

The structure of this chapter is as follows. In Section 1.2, we present basic definitions of SAT and Max-SAT. In Section 1.3, we explain the motivation of our work. In Section 1.4, we describe the objectives of our research. In Section 1.5, we describe the main contributions of the thesis. In Section 1.6, we enumerate the publications we have made during the course of the thesis. In Section 1.7, we present an overview of the remaining chapters.

## 1.2 SAT and Max-SAT problems

In propositional logic, a variable $x_i$ may take values 0 (for false) or 1 (for true). A *literal* $\ell_i$ is a variable $x_i$ or its negation $\neg x_i$. A *clause* is a disjunction of literals, and a CNF formula is a conjunction of clauses. A *weighted clause* is a pair $(C_i, w_i)$, where $C_i$ is a disjunction of literals and $w_i$, its weight, is a positive number, and a *weighted CNF formula* is a conjunction of weighted clauses. A (weighted) *CNF formula* is often represented as a set of clauses.

An *assignment* of truth values to the propositional variables satisfies a literal $x_i$ if $x_i$ takes the value 1 and satisfies a literal $\neg x_i$ if $x_i$ takes the value 0, satisfies a clause if it satisfies at least one literal of the clause, and satisfies a CNF formula if it satisfies all the clauses of the formula. A CNF formula is *satisfiable* if there exists an assignment that satisfies the formula; otherwise, it is *unsatisfiable*.

The *SAT problem* for a CNF formula $\phi$ is the problem of deciding whether there exists a satisfying assignment for $\phi$. The *Max-SAT problem* for a CNF formula $\phi$ is the problem of finding an assignment of values to propositional variables that maximizes the number of satisfied clauses. In this sequel we often use the term Max-SAT meaning Min-UNSAT. This is because, with respect to exact computations, finding an assignment that minimizes the number of unsatisfied clauses is equivalent to finding an assignment that maximizes the number of satisfied clauses.

Two SAT instances are *equivalent* if they are satisfied by the same set of assignments. In Max-SAT, two instances $\phi_1$ and $\phi_2$ are *equivalent* if $\phi_1$ and $\phi_2$ have the same number of unsatisfied clauses for every complete assignment of $\phi_1$ and $\phi_2$.

We will also consider three extensions of Max-SAT which are more well-suited for representing and solving over-constrained problems: weighted Max-SAT, Partial Max-SAT and weighted Partial Max-SAT.

The *weighted Max-SAT* problem for a weighted CNF formula $\phi$ is the problem of finding an assignment of values to propositional variables that maximizes the sum of weights of satisfied clauses (or equivalently, that minimizes the sum of weights of unsatisfied clauses).

A *Partial Max-SAT* instance is a CNF formula in which some clauses are *relaxable* or *soft* and the rest are *non-relaxable* or *hard*. The *Partial Max-SAT problem* for a Partial Max-SAT instance $\phi$ is the problem of finding an assignment that satisfies all the hard clauses and the maximum number of soft clauses.

The *weighted Partial Max-SAT* problem is the combination of weighted Max-SAT and Partial Max-SAT.

## 1.3 Motivation

We started our research on Max-SAT formalisms with hard and soft constraints in 2003. At that time, SAT was —as it is nowadays— a central topic in Artificial Intelligence, Electronic Design Automation and Theoretical Computer Science. There were publicly available complete solvers such as Chaff [MMZ+01], GRASP [MSS99], MiniSat [ES03], Posit [Fre95], Relsat [BS97], and Satz [LA97a, LA97b], as well as local search solvers such as GSAT and WalkSAT [SK93, SKC94, SLM92]. There was also enough empirical evidence about the merits of the generic problem solving approach which consists in modeling NP-complete decision problems as SAT instances, solving the resulting encodings with a state-of-the-art SAT solver, and mapping the solution back into the original problem.

Despite the remarkable activity on SAT, there was a reduced number of papers dealing with the design and implementation of exact Max-SAT solvers; solving NP-hard problems by reducing them to Max-SAT was not considered as a suitable alternative for solving optimization problems; and the activity on Max-SAT was basically concentrated on theoretical results. This is in contrast with what happened in the Constraint Programming community, where the Weighted Constraint Satisfaction Problem (Weighted CSP) was a problem attracting the interest of that community, which published a considerable amount of results about weighted CSP and consolidated a research line on soft constraints [MRS06].

The most remarkable implementations of exact Max-SAT solvers were the branch and bound solvers developed by Wallace and Freuder [WF96], and Borchers and Furman [BF99]. These solvers can be seen as an adaptation to Max-SAT of the Davis-Logemann-Loveland (DLL) procedure [DLL62], and were the starting point for developing some of the most successful modern Max-SAT solvers.

An approach for producing good performing Max-SAT solvers was based on adapting to Max-SAT technology that was proven to be successful in DLL-style SAT solvers such as optimized data structures, clever variable selection heuristics, clause learning, and non-chronological backtracking. Another approach was to improve the quality of the lower bounds in branch and bound Max-SAT solvers by incorporating powerful inference rules that preserve the number of unsatisfied clauses and that, in the best case, make explicit some contradictions; and by incorporating new ways of computing underestimations of the number of unsatisfied clauses that become unsatisfied if the partial assignment associated to a node of the search space is extended to a complete assignment.

The experience on SAT-based problem solving has shown that both the solver and the encoding are important for solving efficiently combinatorial problems. In contrast with other parallel investigations whose main focus were the solvers, *our initial motivation was to investigate Max-SAT formalisms that produce natural and compact encodings of combinatorial optimization problems, and equip them* *with robust solvers that exploit structural properties of the encodings.*

All our work is around the notion of *partiality* in Max-SAT. Partiality amounts to have clauses which are mandatory and clauses (or sets of clauses) which are relaxable. On the one hand, this notion captures the constraints of real problems in a more natural way, and produces more compact encodings. On the other hand, the distinction between mandatory and relaxable clauses has a significant impact on the solving techniques that can be applied in branch and bound solvers. In a sense, we could say that the notion of partiality allows to define formalisms between SAT and Max-SAT for effectively solving combinatorial optimization problems.

Finally, we would like to point out that our research has benefited a lot from the 2006 and 2007 editions of the Max-SAT Evaluation. They allowed to compare our solvers with the most representative state-of-the-art solvers, and make publicly available a good collection of benchmarks for testing our solvers.

## 1.4 Objectives

*The general objective of our research is to study Max-SAT formalisms that incorporate the notion of partiality, and design and implement solving techniques for such formalisms that exhibit a good performance profile on a wide collection of benchmarks. The final goal is to show that Max-SAT formalisms can become a competitive generic problem solving approach for solving over-constrained problems.*

The concrete objectives to achieve in the thesis can be summarized as follows:

- Define a new formalism for solving over-constrained problems, with hard and soft constraints, that deals with blocks of clauses rather than individual clauses in order to produce more compact and natural encodings. Equip such a formalism with exact solvers that incorporate good performing solving techniques, optimized data structures, and heuristics that exploit structural properties of the encodings.

- Define new inference rules and learning schemes for the Partial Max-SAT formalism, and design and implement exact Partial Max-SAT solvers that incorporate them. We plan to learn clauses from the conflicts produced when a hard clause is violated during the exploration of the search space (hard conflict), as well as when a soft clause is violated (soft conflict).

- Design and implement a preprocessor for Partial Max-SAT that applies solving techniques which, despite being too costly for being applied to each node of the search space, can produce gains if they are applied before starting the search. To assess the impact of the preprocessor, we plan to conduct an empirical evaluation using the most representative state-of-the-art Partial Max-SAT solvers and the instances of the last Max-SAT evaluation.

- Conduct an empirical evaluation of the techniques incorporated into the solvers we plan to develop in this thesis, and in particular of the techniques that take into account the distinction between hard and soft clauses. Identifying their strengths and weaknesses should allow us to gain new insights for developing more powerful solving techniques.
- Conduct an empirical comparison between the solvers of the thesis and the best performing state-of-the-art Partial Max-SAT solvers. Knowing the performance profile of other solvers can help improve the performance of our solvers. Moreover, we plan to make the solvers publicly available and actively participate in the Max-SAT evaluations.

## 1.5 Contributions

The main contributions of the thesis can be summarized as follows:

- We defined the Soft-SAT formalism, which allows to encode over-constrained problems in a natural and compact way. Soft-SAT encodes constraints as blocks of clauses without needing to introduce auxiliary variables for dealing with soft constraints. This has positive consequences in the solvers because we can define solving techniques which are local to each block, and apply inference rules, in which the premises are short clauses, earlier than in other formalisms that need to use auxiliary variables in order to ensure that there is exactly one clause for each violated constraint (cf. Example 4.5). Moreover, we developed Soft-SAT solvers with branching heuristics and underestimation techniques that take into account the structure of the domains in the original problem by exploiting information which is hidden in Boolean encodings.
- We extended, to Partial Max-SAT, existing solving techniques for SAT and Max-SAT. Such techniques include variable selection heuristics that take into account the size of the clause in which the variable appears, lower bound computation methods based on unit propagation, failed literal detection to improve the lower bound, and local search solvers to obtain a good initial upper bound.
- We defined new inference rules for Partial Max-SAT. These rules are proven to be sound, can be applied efficiently, and can be seen as unit resolution refinements.
- We incorporated the 1-UIP learning schema [MMZ$^+$01] to our solvers in order to analyze the conflicts detected in hard clauses. We learn a clause from each conflict and backtrack non-chronologically. To the best of our knowledge, it was the first time that learning was incorporated into a branch and bound Partial Max-SAT solver.
- We defined new soft learning techniques that are applied every time we reach a soft conflict.

- We designed and implemented two Soft-SAT solvers:
  - **Soft-SAT-S:** It was the first solver developed for the Soft-SAT formalism. It uses a static variable selection heuristic, extremely efficient lazy data structures, and an underestimation based on inconsistency counts.
  - **Soft-SAT-D:** It was the second solver developed for the Soft-SAT formalism. It uses a dynamic variable selection heuristic with n-ary branching for Soft-SAT encoded CSP instances, lazy data structures based on two-watched literals, and an underestimation based on inconsistency counts.
- We designed and implemented two Partial Max-SAT solvers:
  - **PMS:** It was the first branch and bound solver that we developed for the Partial Max-SAT formalism. It is an implementation from scratch, and the most important feature of this solver is the learning module for hard and soft constraints. PMS also incorporates advanced techniques for bounds computation and simple inference rules.
  - **W-MaxSatz:** It was the second solver that we developed for the Partial Max-SAT formalism. It is build on top of the Max-SAT solver MaxSatz, and the most important feature of this solver is the advanced techniques used for the lower bound computation, which include the computation of underestimation with unit propagation enhanced with failed literal detection and the application of sound inference rules. W-MaxSatz also incorporates a hard learning module.
- We designed and implemented several preprocessors for Partial Max-SAT instances. The most important are:
  - **Variable saturation:** This preprocessing saturates the formula w.r.t. a limited number of variables. It helps reduce the search space by removing variables from the initial formula.
  - **Learning and restarts:** This preprocess adds to the initial formula a set of redundant clauses from several search spaces.
- We conducted an empirical evaluation of the learning techniques and new inference rules developed in this thesis. We observed that hard learning is an important feature for Partial Max-SAT solvers, soft learning can improve the results in some sets of instances, and in combination with our inference rules, gives rise to the best performance profile.
- We conducted an empirical evaluation between our solvers and the best performing state-of-the-art Partial Max-SAT solvers. We observed that our Soft-SAT solvers are competitive against weighted CSP solvers and state-of-the-art Partial Max-SAT solvers. PMS has a good general performance,

and W-MaxSatz is competitive on several types of instances, specially on random instances.

- We conducted an empirical evaluation of the preprocessors developed in this thesis using both our solvers and the most representative state-of-the-art Partial Max-SAT solvers. We observed that our preprocessors can reduce the CPU time needed to solve several types of instances.

## 1.6 Publications

Some of the results presented in this thesis have already been published in journals and conference proceedings. The articles are chronologically listed and classified according to the main topics of the thesis, Soft-SAT and Partial Max-SAT:

**Soft-SAT**

– Josep Argelich and Felip Manyà. Solving Over-Constrained Problems with Max-SAT Algorithms. In *Workshop on Modelling and Solving Problems with Constraints, 16th European Conference on Artificial Intelligence, ECAI-2004, Valencia, Spain*, pages 116–124, Workshop Proceedings, 2004.

– Josep Argelich and Felip Manyà. An Exact Max-SAT Solver for Over-Constrained Problems. In *6th International Workshop on Preferences and Soft Constraints, 10th International Conference on Principles and Practice of Constraint Programming, CP-2004, Toronto, Canada*, pages 1–11, Workshop Proceedings, 2004.

– Josep Argelich. Solving Over-Constrained Problems with SAT. In *11th International Conference on Principles and Practice of Constraint Programming, CP-2005, Sitges, Spain*, page 838, Springer LNCS 3709, 2005.

– Josep Argelich and Felip Manyà. Solving Over-Constrained Problems with SAT Technology. In *8th International Conference on Theory and Applications of Satisfiability Testing, SAT-2005, St. Andrews, Scotland*, pages 1–15, Springer LNCS 3569, 2005.

– Josep Argelich and Felip Manyà. Exact Max-SAT Solvers for Over-Constrained Problems. *Journal of Heuristics*, 12(4-5):375-392, 2006.

**Partial Max-SAT**

– Josep Argelich and Felip Manyà. Learning Hard Constraints in Max-SAT. In *11th Annual ERCIM Workshop on Constraint Solving and Constraint Programming, CSCLP-2006, Caparica, Portugal*, pages 5–12, Workshop Proceedings, 2006.

– Josep Argelich and Felip Manyà. Partial Max-SAT Solvers with Clause Learning. In *10th International Conference on Theory and Applications of Satisfiability Testing, SAT-2007, Lisbon, Portugal*, pages 28–40, Springer LNCS 4501, 2007.

– Josep Argelich, Chu Min Li and Felip Manyà. An Improved Exact Solver for Partial Max-SAT. In *The International Conference on Nonconvex Programming: Local and Global Approaches, NCP-2007, Rouen, France*, pages 230–231, Conference Proceedings, 2007

**Preprocessing techniques**

– Josep Argelich, Chu Min Li and Felip Manyà. A Preprocessor for Max-SAT Solvers. In *11th International Conference on Theory and Applications of Satisfiability Testing, SAT-2008, Guangzhou, P. R. China*, pages 15–20, Springer LNCS 4996, 2008.

**Encodings**

– Josep Argelich, Alba Cabiscol, Inês Lynce and Felip Manyà. Encoding Max-CSP into Partial Max-SAT. In *38th International Symposium on Multiple-Valued Logic, ISMVL-2008, Dallas, Texas*, pages 106–111, IEEE CS Press, 2008.

– Josep Argelich, Alba Cabiscol, Inês Lynce and Felip Manyà. Modelling Max-CSP as Partial Max-SAT. In *11th International Conference on Theory and Applications of Satisfiability Testing, SAT-2008, Guangzhou, P. R. China*, pages 1–14, Springer LNCS 4996, 2008.

**Miscellaneous**

– Josep Argelich, Xavier Domingo, Felip Manyà and Jordi Planes. Towards Solving Many-Valued Max-SAT. In *36th International Symposium on Multiple-Valued Logic, ISMVL-2006, Singapore*, paper 26, IEEE CS Press, 2006.

– Josep Argelich, Chu Min Li, Felip Manyà and Jordi Planes. The First and Second Max-SAT Evaluations. *Journal on Satisfiability*, submitted for second review, 2008.

## 1.7 Overview

This section provides an overview of the thesis. We briefly describe the contents of each of the remaining chapters:

**Chapter 2: SAT algorithms.** We present an overview of the most relevant methods for solving SAT. First, we define some basic concepts in SAT. Second, we present the resolution method, which applies an inference rule that provides a refutation complete inference system. Third, we describe

DP, the first effective method for producing resolution refutations. Fourth, we present the DLL procedure, implemented in the majority of state-of-the-art complete SAT algorithms, and review the main solving techniques that have been incorporated into DLL in order to devise fast SAT solvers. Finally, we give the basis of the current state-of-the-art local search algorithms for SAT.

**Chapter 3: Max-SAT algorithms.** We introduce some background knowledge about Max-SAT, and review the solving techniques that have proved to be useful in terms of performance. First, we define some basic concepts in Max-SAT and Max-CSP. Second, we present the branch and bound schema, which is the most commonly used approach to exact Max-SAT solving. Third, we define a complete resolution rule for Max-SAT. Fourth, we review the main Max-SAT approximation algorithms. Fifth, we describe the solving techniques that have been defined for dealing with hard and soft constraints under the formalism of Partial Max-SAT. Finally, we present the 2006 and 2007 Max-SAT Evaluations.

**Chapter 4: The Soft-SAT formalism.** We present a new generic problem solving approach for over-constrained problems based on Max-SAT. We first define a Boolean clausal form formalism that deals with blocks of clauses instead of individual clauses, and that allows one to declare a block of hard clauses and several blocks of soft clauses. We call soft CNF formulas to this new kind of formulas. We then present two Max-SAT solvers that find a truth assignment that satisfies the hard block of clauses and maximizes the number of satisfied soft blocks. Our solvers are branch and bound algorithms equipped with original lazy data structures, powerful inference techniques, good quality lower bounds, and original variable selection heuristics. Finally, we report an experimental investigation on a representative sample of instances which provides experimental evidence that our approach is competitive with the state-of-the-art approaches developed in the CSP and SAT communities.

**Chapter 5: The Partial Max-SAT formalism.** We focus on Partial Max-SAT, which is a problem between SAT and Max-SAT which is well-suited for representing and solving over-constrained problems, and has become a standard in the recent years. First, we present an overview of the Partial Max-SAT problem. Second, we define novel techniques for Partial Max-SAT solving, and introduce the solving techniques that incorporate the modern Partial Max-SAT solvers. Third, we present some efficient and original preprocessing techniques for Partial Max-SAT. Next, we describe the two Partial Max-SAT solvers we have designed and implemented. Finally, we report on an experimental investigation that we conducted in order to assess the performance of our solvers and preprocessing techniques. The experimental results indicate that our solvers are among the best state-of-the-art Partial Max-SAT solvers.

**Chapter 6: Conclusions.** We briefly summarize the main contributions of the thesis, and point out some open problems and future research directions that we plan to tackle in the near future.