Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 332 (2017) 39–56

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Generalized Probabilistic Satisfiability [1]

Carlos Caleiro[a] Filipe Casal[b] Andreia Mordido[a]

[a] *SQIG - Instituto de Telecomunicações*
Dep. Mathematics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

[b] *Centro de Matemática, Aplicações Fundamentais e Investigação Operacional (CMAF-CIO)*
Dep. Mathematics, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract

We analyze a generalized probabilistic satisfiability problem (GenPSAT) which consists in deciding the satisfiability of linear inequalities involving probabilities of classical propositional formulas. GenPSAT is proved to be NP-complete and we present a polynomial reduction to Mixed-Integer Programming. Capitalizing on this translation, we implement and test a solver for the GenPSAT problem. As previously observed for many other NP-complete problems, we are able to detect a phase transition behaviour for GenPSAT.

Keywords: Probabilistic Satisfiability, GenPSAT, Mixed-Integer Programming, Phase Transition

1 Introduction

For many years, the satisfiability problem for propositional logic (SAT) has been extensively studied both for theoretical purposes, such as complexity theory, and for practical purposes. In spite of its NP-completeness [9], modern tools for solving SAT are able to cope with very large problems in a very efficient manner, leading to applications in many different areas and industries [2].

Naturally, people started extending this problem to more expressive frameworks: for instance in Satisfiability Modulo Theories [10], instead of working in propositional logic, one can try to decide if a formula is valid in some specific first-order theory. One other direction is to extend propositional logic with probabilities. The probabilistic satisfiability problem (PSAT) was originally formulated by George

[1] Work done under the scope of R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER–PT2020) and partially supported by Fundação para a Ciência e a Tecnologia by way of grant UID/MAT/04561/2013 to Centro de Matemática, Aplicações Fundamentais e Investigação Operacional of Universidade de Lisboa (CMAF-CIO). AM was supported by FCT under the grant SFRH/BD/77648/2011 and by the Calouste Gulbenkian Foundation under *Programa de Estímulo à Investigação* 2011. FC acknowledges the support from the DP-PMI and FCT (Portugal) through scholarship SRFH/BD/52243/2013. CC acknowledges the support of EU FP7 Marie Curie PIRSES-GA-2012-318986 project GeTFun: Generalizing Truth-Functionality.

http://dx.doi.org/10.1016/j.entcs.2017.04.004
1571-0661/© 2017 The Author(s). Published by Elsevier B.V.

Boole [3] and later by Nilsson [18]. This problem consists in deciding the satisfiability of a set of assignments of probabilities to propositional formulas. There has been a great effort on the analysis of the probabilistic satisfiability problem and on the development of efficient tools for the automated treatment of this problem [11,14,6,1,12].

In this paper we study a Generalized Probabilistic Satisfiability problem (GenPSAT) extending the scope of PSAT by allowing linear combinations of probabilistic assignments of values to propositional formulas, and has applications in the analysis of the security of cryptographic protocols and on estimating the probability of existence of attacks [17]. Intuitively, GenPSAT consists in deciding the existence of a probability distribution satisfying a set of classical propositional formulas with probability 1, and a set of linear inequalities involving probabilities of propositional formulas. The GenPSAT problem was previously identified in the context of the satisfiability of the probabilistic logic in [13], where it was also shown to be NP-complete. Here, we explore the computational behaviour of this problem and present a polynomial reduction from GenPSAT to Mixed-Integer Programming, following the lines of [6,1].

Mixed-Integer Programming (MIP) [19] is a framework to find an optimal solution for a linear objective function subject to a set of linear constraints over real and integer variables. We will exploit the close relation between SAT and MIP [4] in order to reduce GenPSAT problems to suitable MIP problems.

As observed in many NP-complete problems [7], GenPSAT also presents a phase transition behaviour. By solving batches of parametrized random GenPSAT problems, we observe the existence of a threshold splitting a phase where almost every GenPSAT problem is satisfiable, and a phase where almost every GenPSAT problem is not satisfiable. During such transition, the problems become much harder to solve [7].

As the main contribution of this work, we develop the theoretical framework that allows the translation between GenPSAT and MIP problems, which then allows the implementation of a provably correct solver for GenPSAT. This translation is able to encode strict inequalities and disequalities into the MIP context. With the GenPSAT solver in hands, we are able to detect and study the phase transition behaviour.

The paper is outlined as follows: in Section 2 we briefly recall the PSAT problem; in Section 3 we carefully define the GenPSAT problem and establish some results on its complexity; Section 4 is dedicated to finding a polynomial reduction from GenPSAT to MIP and a prototype tool is provided for an automated analysis of the problem; in Section 5 we analyze the presence of phase transition; finally, in Section 6, we assess our contributions and discuss future work.

2 Preliminaries

Let us begin by fixing a set of propositional variables $\mathcal{P} = \{x_1, \dots, x_n\}$. A *literal* is either a propositional variable or its negation. A *clause* is a non-empty disjunction of one or more literals. A *propositional formula* is any Boolean combination of propositional variables.

A *propositional valuation* is a map $v : \mathcal{P} \to \{0,1\}$, which is extended to propositional formulas as usual. We say that a set of valuations $\mathcal{V}$ satisfies a propositional formula φ if, for each $v \in \mathcal{V}$, $v(\varphi) = 1$. This notion is extended to sets of propositional formulas as usual. Let $\mathcal{V}^* = \{v_1, \dots, v_{2^n}\}$ be the set of all valuations defined over variables of $\mathcal{P}$. We define a *probability distribution* π over $\mathcal{V}^*$ as a probability vector of size 2^n.

A *simple probabilistic formula* is an expression of the form $\mathsf{Pr}(c) \bowtie p$, where c is a clause, $p \in \mathbb{Q}$, $0 \le p \le 1$ and $\bowtie \in \{=, \le, \ge\}$. We say that a probability distribution π *satisfies* a formula $\mathsf{Pr}(c) \bowtie p$ if

$$\sum_{i=1}^{2^n} (v_i(c) \cdot \pi_i) \bowtie p \quad .$$

A probability distribution π satisfies a set of simple probabilistic formulas if it satisfies each one of them.

We now recall the PSAT problem [18,14,11].

Definition 2.1 [PSAT problem] Given a set of propositional variables $\mathcal{P}$ and a set of simple probabilistic formulas $\Sigma = \{\mathsf{Pr}(c_i) \bowtie p_i \mid 1 \le i \le k\}$, the Probabilistic Satisfiability problem (PSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies Σ.

The PSAT problem for $\{\mathsf{Pr}(c_i) \bowtie_i p_i \mid 1 \le i \le k\}$ can be formulated algebraically as the problem of finding a solution π for the system of inequalities

$$\begin{cases} V\pi \bowtie p \\ \sum \pi_i = 1 \\ \pi \ge 0 \end{cases} ,$$

where V is the $k \times 2^n$ matrix such that $V_{ij} = v_j(c_i)$, i.e., $V_{ij} = 1$ iff the j-th valuation satisfies the i-th clause, $p = [p_i]$ is the k vector of all p_i and $\bowtie = [\bowtie_i]$ is the k vector of all $\bowtie_i$.

The SAT problem can be modeled as a PSAT instance where the entries p_i of the probability vector are all identical to 1. The PSAT problem was shown to be NP-complete [14,13], even when the clauses consist of the disjunction of only two literals, 2-PSAT.

3 GenPSAT problem

We now extend the notion of simple probabilistic formula to handle linear inequalities involving probabilities of propositional formulas. A *probabilistic formula* is an expression of the form

$$\sum_{i=1}^{\ell}(a_i\mathsf{Pr}(c_i)) \bowtie p \ ,$$

where c_i are propositional clauses, $\bowtie \in \{\geq, <, \neq\}$, $\ell \in \mathbb{N}$ and $a_1, \ldots, a_\ell, p \in \mathbb{Q}$. Observe that formulas with the relational symbols $\leq, >, =$ can be obtained by abbreviation. In the case where $\ell = 1$ and $a_1 = 1$, we obtain a simple probabilistic formula. An *atomic probabilistic formula* is a probabilistic formula where each c_i is a propositional variable, i.e., $c_i \in \mathcal{P}$ for each i.

We say that a probability distribution π *satisfies* a formula $\sum_{i=1}^{\ell}(a_i\mathsf{Pr}(c_i)) \bowtie p$ if

$$\sum_{i=1}^{\ell}\left(a_i\left(\sum_{j=1}^{2^n} v_j(c_i)\cdot\pi_j\right)\right) \bowtie p \ .$$

A probability distribution π satisfies a set of probabilistic formulas if it satisfies each one of them.

An *instance* of GenPSAT is a pair (Γ, Σ) where Γ is a set of propositional clauses (also called hard constraints) and Σ is a set of probabilistic formulas (soft constraints). We say that a probability distribution π *satisfies* a GenPSAT instance (Γ, Σ) if it satisfies the set of probabilistic formulas

$$\Xi_{(\Gamma,\Sigma)} = \Sigma \cup \{\mathsf{Pr}(\gamma) = 1 \mid \gamma \in \Gamma\} \ . \tag{1}$$

Definition 3.1 [GenPSAT problem] Given a GenPSAT instance (Γ, Σ), the Generalized Probabilistic Satisfiability problem (GenPSAT) consists in determining whether there exists a probability distribution π over $\mathcal{V}^*$ that satisfies (Γ, Σ).

GenPSAT poses a convenient framework for specifying constraints involving different probabilistic formulas. For instance, one may want to impose that $2\mathsf{Pr}(A) \leq \mathsf{Pr}(B)$ for two propositional clauses A, B. Such requirements may be very useful in specifying properties of interesting systems but they cannot be easily expressed in the PSAT framework. We now showcase GenPSAT's expressiveness by encoding the Monty Hall problem [20].

Example 3.2 The Monty Hall problem is a puzzle where we are faced with the choice of picking one of three doors, knowing that a prize is behind one of them. After our initial choice, the game host opens one of the remaining doors provided that the prize is not behind it, and gives us the choice of switching or keeping the initial guess. The question is: which option is more advantageous?

To model this problem as a GenPSAT instance, let us define the following propositional variables: P_i holds if the prize is behind door i, X_i holds if our initial choice is door i, H_i holds if the host reveals door i after our initial choice, for $i \in \{1, 2, 3\}$. Since there are only one door with a prize, one initial choice, and one door revealed by the host, we impose the following restrictions:

$$\Gamma = \left\{ \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (P_i \wedge \neg P_j \wedge \neg P_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (X_i \wedge \neg X_j \wedge \neg X_k), \bigvee_{\substack{i,j,k\in\{1,2,3\}\\ i\neq j\neq k\neq i}} (H_i \wedge \neg H_j \wedge \neg H_k) \right\} .$$

Furthermore, the host cannot open neither the chosen door nor the door with the prize and so we include the followings constraints in Γ:

$$P_i \to \neg H_i \text{ and } X_i \to \neg H_i \quad \text{for each } i \in \{1,2,3\} .$$

We further assume that the prize has uniform probability of being behind each door and that the initial choice is independent of where the prize is:

$$\Sigma = \bigcup_{i,j\in\{1,2,3\}} \left\{ \mathsf{Pr}(P_i) = \frac{1}{3}, \quad \mathsf{Pr}(P_i \wedge X_j) = \frac{1}{3}\mathsf{Pr}(X_j) \right\}$$

Concerning the question of which is more advantageous, switching or keeping our initial choice, we encode *winning by switching* as

$$\mathsf{WbS}: \qquad \bigwedge_{i=1}^{3} (P_i \leftrightarrow (\neg X_i \wedge \neg H_i)) \ ,$$

and *winning by keeping* as

$$\mathsf{WbK}: \qquad \bigwedge_{i=1}^{3} (P_i \leftrightarrow X_i) \ .$$

We want to the decide whether it is always the case that $\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})$, which can be checked by testing the satisfiability of the GenPSAT instance

$$(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) < \mathsf{Pr}(\mathsf{WbK})\}) \ .$$

As expected, this instance is not satisfiable and the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \geq \mathsf{Pr}(\mathsf{WbK})\})$ is satisfiable, allowing us to conclude that it is always advantageous to switch our initial option.

We can take this analysis one step further, and show that the probability of winning by switching is $\frac{2}{3}$ by checking that the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) \neq \frac{2}{3}\})$ is unsatisfiable and that the instance $(\Gamma, \Sigma \cup \{\mathsf{Pr}(\mathsf{WbS}) = \frac{2}{3}\})$ is satisfiable. All these instances were checked using the tool we implemented, [8]. $\lozenge$

Notice that the PSAT problem for Σ can be modeled in GenPSAT by considering the instance $(\varnothing, \Sigma)$.

Given a GenPSAT instance (Γ, Σ), where Γ contains m clauses and Σ is composed of k probabilistic formulas, we follow the lines of Nilsson [18] for a linear algebraic formulation and consider a $(k+m)\times 2^n$ matrix $V = [V_{ij}]$, where for each $i \in \{1, \ldots, k+$

$m\}$ and $j \in \{1, \ldots, 2^n\}$ V_{ij} is defined from the j^{th} valuation v_j and from the i^{th} probabilistic formula $\sum_{u=1}^{\ell} a_u^i \mathsf{Pr}(c_u^i) \bowtie_i p_i$ of $\Xi_{(\Gamma,\Sigma)}$ as follows:

$$V_{ij} = \sum_{u=1}^{\ell} a_u^i \cdot v_j(c_u^i) \ .$$

Furthermore, define two vectors of size $k+m$, $p = [p_i]$ and $\bowtie = [\bowtie_i]$. GenPSAT is equivalent to the problem of deciding the existence of a solution π to the system

$$\begin{cases} V\pi \bowtie p \\ \sum \pi_i = 1 \\ \pi \geq 0 \end{cases} . \tag{2}$$

Given a set of probabilistic formulas $\Omega = \left\{ \sum_{u=1}^{\ell} a_u^i \cdot v_j(c_u^i) \bowtie_i p_i \mid 1 \leq i \leq k \right\}$ and a set of valuations $\mathcal{V} = \{v_1, \ldots, v_{k'}\}$, we define the $[\Omega, \mathcal{V}]$-*associated matrix* as the $(k+1) \times k'$ matrix $M_{[\Omega,\mathcal{V}]} = [M_{ij}]$ such that

$$M_{k+1,j} = 1 \text{ for each } 1 \leq j \leq k'$$

and

$$M_{ij} = \sum_{u=1}^{\ell} a_u^i \cdot v_j(c_u^i) \quad \text{for } 1 \leq i \leq k, \;\; 1 \leq j \leq k' \ .$$

Then, we can rewrite system (2) using the $[\Xi_{(\Gamma,\Sigma)}, \mathcal{V}^*]$-associated matrix V as

$$\begin{cases} V\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{3}$$

We now show that this problem is NP-complete. For this purpose, we first present the following lemma.

Lemma 3.3 ([13,5]) *If a system of ℓ linear inequalities with integer coefficients has a non-negative solution, then it has a non-negative solution with at most ℓ positive entries.*

Theorem 3.4 ([13]) GenPSAT *is* NP-*complete.*

Proof. We begin by showing that GenPSAT is in NP by providing a polynomial sized certificate. Notice that Lemma 3.3 can be extended to rational coefficients simply by normalizing with the greatest denominator. Applying this result to the system (3) we conclude that there is a $(k+m+1) \times (k+m+1)$ matrix W, composed of columns of V, whose system

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{4}$$

has a solution iff the original system (3) has a solution. Furthermore, the obtained solutions from (4) can be mapped to solutions of (3) by inserting zeros in the appropriate positions. Since the obtained solution from the latter system has $k+m+1$ elements, it constitutes the NP-certificate for the GenPSAT problem.

Furthermore, given that the PSAT problem can be modeled in GenPSAT, it follows that GenPSAT is NP-complete. □

We say that a GenPSAT instance (Γ, Σ) is in *normal form* if Γ is a set of propositional clauses with 3 literals, i.e., Γ can be seen as a 3CNF formula, and Σ is a set of atomic probabilistic formulas.

Lemma 3.5 *Given a* GenPSAT *instance* (Γ, Σ) *there exists an instance* (Γ', Σ') *in normal form such that* (Γ, Σ) *is satisfiable iff* (Γ', Σ') *is satisfiable. Moreover,* (Γ', Σ') *is obtained from* (Γ, Σ) *in polynomial time.*

Proof. Let (Γ, Σ) be the GenPSAT instance to be put in normal form. We obtain Σ' by transforming formulas in Σ into atomic probabilistic formulas. For this purpose, let $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(c_i) \bowtie p$ be a formula in Σ and consider the atomic probabilistic formula obtained by replacing (when needed) each clause c_i by a fresh variable y_i,

$$\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p \quad .$$

Furthermore, the y_i variable is added to $\mathcal{P}$ and the formula stating the equivalence between y_i and c_i, $(y_i \leftrightarrow c_i)$, is collected in a set Δ.

We are left with the transformation of the formula

$$\bigwedge_{\gamma \in \Gamma} \gamma \wedge \bigwedge_{(y \leftrightarrow c) \in \Delta} (y \leftrightarrow c)$$

into 3-CNF using Tseitin's transformation [21], which can increase linearly the size of the formula and add new variables to $\mathcal{P}$. The final Γ' is the set of conjuncts of the obtained 3-CNF formula. Since Tseitin's transformation preserves satisfiability of formulas, (Γ, Σ) is satisfiable iff (Γ', Σ') is satisfiable. □

4 Reducing GenPSAT to Mixed-Integer Programming

In this section we explore the close relation between satisfaction of propositional formulas and feasability of a set of linear constraints over binary variables (see [4]). With this, we present a reduction of GenPSAT to Mixed-Integer Programming (MIP), similarly to what was done for PSAT [6] and GPSAT [1]. A MIP problem consists in optimizing a linear objective function subject to a set of linear constraints over real and integer variables. MIP was shown to be NP-complete, see [19]. Observe that this translation to MIP also serves as a proof that GenPSAT is in NP.

4.1 Linear Algebraic Formulation for GenPSAT

Lemma 4.1 *A* GenPSAT *instance in normal form* (Γ, Σ), *with* $|\Sigma| = k$, *is satisfiable iff there exists a* $(k+1) \times k'$ *matrix* W *of rank* $k' \leq k+1$ *and a set of valuations* $\mathcal{V}_0$ *of size* k' *such that:*

(i) W *is the* $[\Sigma, \mathcal{V}_0]$*-associated matrix*

(ii) $\mathcal{V}_0$ *satisfies* Γ,

(iii) considering $p = [p_1, \ldots, p_k, 1]$ *and* $\bowtie = [\bowtie_1, \ldots, \bowtie_k, =]$*, the system*

$$\begin{cases} W\pi \bowtie p \\ \pi \geq 0 \end{cases} \tag{5}$$

is satisfiable.

Proof. Let (Γ, Σ) be a satisfiable GenPSAT instance in normal form, with $|\Sigma| = k$ and $|\Gamma| = m$. Then, denoting by V the $[\Xi_{(\Gamma,\Sigma)}, \mathcal{V}^*]$-associated matrix, the system

$$\begin{cases} V\pi \bowtie p \\ \pi \geq 0 \end{cases}$$

has a solution. And so, using Lemma 3.3, there is a $(k+m+1) \times \ell$ matrix V^*, where $\ell \leq k + m + 1$, and whose system has a positive solution π^*. Notice that the set of valuations underlying V^* certainly satisfies Γ, as $\pi^*_j > 0$ for each $1 \leq j \leq \ell$.

Let W^* be the matrix constructed from V^* by choosing the first k rows (corresponding to the probabilistic formulas in Σ) and the last row (requiring that the solution sums up to one) of V^*. Still, the corresponding system has a positive solution. Using Lemma 3.3 once more, we conclude that exists a $(k+1) \times k'$ matrix W, with $k' \leq k + 1$, whose system has a positive solution ρ^*. The solution π for (5) is obtained from ρ^* by inserting zeros in the appropriate positions.

Reciprocally, assume that there exists a $(k+1)\times k'$ matrix W of rank $k' \leq k+1$ satisfying (i), (ii), (iii), and let π denote the solution for (5). We are looking for a probability distribution π^* satisfying (Γ, Σ). For this purpose, let $\mathcal{V}_0 = \{v_{j_1}, \ldots, v_{j_{k'}}\} \subseteq \mathcal{V}$ denote the set of valuations underlying W according to condition (ii), and define $\pi^* = [\pi^*_i]$, where

$$\pi^*_i = \begin{cases} \pi_i & \text{if } i \in \{j_1, \ldots, j_{k'}\} \\ 0 & \text{otherwise} \end{cases} .$$

The verification that π^* satisfies the GenPSAT instance is now immediate:

- given $\gamma \in \Gamma$, we check that π^* verifies $\mathsf{Pr}(\gamma) = 1$ by observing that the last equality represented on W on (5) leads to $\sum_{s=1}^{k'} \pi_{j_s} = 1$ and so,

$$\sum_{j=1}^{2^n} v_j(\gamma) \cdot \pi^*_j = \sum_{\{j | v_j(\gamma)=1\}} \pi^*_j = \sum_{s=1}^{k'} \pi_{j_s} = 1 .$$

- given an atomic probabilistic formula $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ in Σ, we recall the definition of π^* and that π is a solution for (5) to conclude that

$$\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(y_i) \cdot \pi^*_j \right) = \sum_{i=1}^{\ell} a_i \left(\sum_{s=1}^{k'} v_{j_s}(y_i) \cdot \pi_{j_s} \right) = \sum_{s=1}^{k'} \left(\sum_{i=1}^{\ell} a_i \cdot v_{j_s}(y_i) \right) \pi_{j_s} \bowtie p ,$$

  i.e., π^* satisfies the formulas in Σ. □

4.2 *Translation to* MIP

Regarding Lemma 4.1, given a GenPSAT instance (Γ, Σ) in normal form, with $|\Sigma| = k$ and $|\Gamma| = m$, our goal is now to describe a procedure that encodes the problem of finding a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii), as a MIP problem. We dub this procedure GenToMIP.

Let us denote by $H = [h_{ij}]$ the (still unknown) matrix of size $n \times k'$ whose columns represent the valuations in $\mathcal{V}_0$ evaluated on each propositional variable of $\mathcal{P}$, i.e., $h_{ij} = v_j(x_i)$ for each $1 \le i \le n$ and $1 \le j \le k'$. Let $\alpha_1, \ldots, \alpha_n$ represent the probability of the propositional variables $x_1, \ldots, x_n$, respectively, and following the reasoning of [6,1] we model the non-linear constraint $\sum_{j=1}^{k'} h_{ij} \cdot \pi_j = \alpha_i$ as a linear inequality

$$\sum_{j=1}^{k'} b_{ij} = \alpha_i \quad , \tag{val1}$$

by introducing the extra variables b_{ij} which are subject to the appropriate constraints, namely forcing b_{ij} to be zero whenever $h_{ij} = 0$, and ensuring that $b_{ij} = \pi_j$ whenever $h_{ij} = 1$, i.e.,

$$0 \le b_{ij} \le h_{ij} \text{ and } h_{ij} - 1 + \pi_j \le b_{ij} \le \pi_j \quad . \tag{val2}$$

We ensure that π represents a probability distribution by imposing that

$$\sum_{j=1}^{k'} \pi_j = 1 \quad . \tag{sums1}$$

Still, as each valuation of $\mathcal{V}_0$ satisfies Γ, given a clause $\left(\bigvee_{r=1}^{w} x_{i_r}\right) \vee \left(\bigvee_{s=1}^{w'} \neg x_{i'_s}\right)$ of Γ, we generate a linear inequality for each valuation $1 \le j \le k'$,

$$\left(\sum_{r=1}^{w} h_{i_r,j}\right) + \left(\sum_{s=1}^{w'} (1 - h_{i'_s j})\right) \ge 1. \tag{gamma}$$

Notice that, if we have a total of m clauses in Γ, we generate $m \times k'$ such inequalities.

In order to verify the satisfiability of probabilistic formulas in the MIP framework, consider an atomic probabilistic formula $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \bowtie p$ in Σ. Since $\bowtie$ can either be the relational symbol $\ge$, $<$ or $\ne$, we can easily encode the first kind of inequalities as a MIP linear constraint, but should be careful when dealing with the remaining relational symbols.

For atomic probabilistic formulas of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \ge p$, we generate the linear inequality

$$\sum_{i=1}^{\ell} a_i \cdot \alpha_i \ge p \quad . \tag{prob$_\ge$}$$

In the case where $\bowtie$ is a strict inequality $<$, we use a specific variable introduced into the MIP problem, say ε, to fix the objective function as the maximization of ε,

$$\text{maximize } \varepsilon \tag{obj}$$

and further introduce the linear constraint

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + \varepsilon \le p \quad . \qquad (\mathsf{prob}_<)$$

For atomic probabilistic formulas φ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) \neq p$, i.e.

$$\sum_{i=1}^{\ell} a_i \mathsf{Pr}(y_i) - p \neq 0, \qquad (6)$$

we force the left hand side to be either strictly greater or strictly less than zero,

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p < 0 \qquad \text{or} \qquad \sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p > 0 \quad .$$

Even though these are linear constraints, the problem would explode if we treated the disjunction. In this sense, notice that, denoting by C a sufficiently large number, say $C = 1 + |p| + \sum_{i=1}^{\ell} |a_i|$, the inequality (6) holds if and only if there exists a fresh binary variable z_φ such that the following two strict inequalities hold simultaneously:

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p < C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + p < C - C \cdot z_\varphi \quad .$$

Then, we are left with two strict inequalities, thus reducing this analysis to a previous case, from which we obtain the constraints

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) - p + \varepsilon \;\le\; C \cdot z_\varphi \qquad \text{and} \qquad -\sum_{i=1}^{\ell}(a_i \cdot \alpha_i) + p + \varepsilon \;\le\; C - C \cdot z_\varphi \quad . \qquad (\mathsf{prob}_{\neq})$$

Denoting by $k_\geq$, $k_<$, $k_\neq$ the number of probabilistic formulas in Σ when $\bowtie$ coincides with $\geq$, $<$, $\neq$, respectively, so far we have introduced:

- n constraints (val1),
- $4 \times n \times k'$ constraints (val2),
- 1 constraint (sums1),
- $m \times k'$ constraints (gamma),
- $k_\geq$ constraints ($\mathsf{prob}_\geq$),
- $k_<$ constraints ($\mathsf{prob}_<$),
- $2 \times k_\neq$ constraints ($\mathsf{prob}_\neq$).

Hence, we have $\mathcal{O}(n + n \times k' + m \times k' + k)$ inequalities over $n \times k'$ binary variables h_{ij}, $n \times k'$ real variables b_{ij}, n real variables $0 \le \alpha_i \le 1$, $k_\neq$ binary variables z_φ, a real variable $\varepsilon \ge 0$ and k' real variables $\pi_j \ge 0$. Because of this, the GenToMIP translation is polynomial.

Proposition 4.2 *The* GenToMIP *procedure transforms a* GenPSAT *instance in normal form* (Γ, Σ) *into a* MIP *problem whose size is polynomial on the size of* (Γ, Σ).

We now need to show that the existence of a set of valuations $\mathcal{V}_0$ and a probability distribution π in the conditions (i),(ii),(iii) of Lemma 4.1 is equivalent to the feasibility of the MIP problem obtained through GenToMIP with an optimal value $\varepsilon > 0$ (when applicable).

This procedure is presented in Algorithm 1, which given a GenPSAT instance, translates it into a MIP problem and then solves the latter appropriately. For that, let us assume that we initialize an empty MIP problem and consider the following auxiliary procedures:

- add_const introduces a linear constraint into the MIP problem,
- set_obj defines the objective function (either as a maximization or as a minimization) when it was previously not defined,
- fresh declares a fresh binary variable into the MIP problem,
- mip_sat returns True or False depending on whether the problem is feasible (and achieves an optimal solution) or not,
- mip_objvalue returns the objective value, when an objective function was set.

Proposition 4.3 *A* GenPSAT *instance in normal form* (Γ, Σ) *is satisfiable iff Algorithm 1 returns* Sat.

Proof. Let (Γ, Σ) be a satisfiable GenPSAT instance in normal form, and also $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ and $\rho = [\rho_i]$ represent a set of valuations and a probability distribution given by Lemma 4.1 which satisfy conditions (i)-(iii). Then, consider the following values and afterwards let us check that they constitute an optimal solution for the MIP problem constructed at Algorithm 1: for each $1 \leq i \leq n$ and $1 \leq j \leq k'$, let

$$\begin{aligned} h^*_{ij} &= v_j(x_i), \\ b^*_{ij} &= h^*_{ij} \cdot \rho_j, \\ \pi^*_j &= \rho_j, \end{aligned}$$

$$\begin{aligned} \alpha^*_i &= \sum_{\{j | v_j(x_i)=1\}} \rho_j, \\ \varepsilon^* &= \mathsf{min}\ \Delta, \end{aligned}$$

Algorithm 1 GenPSAT solver based on MIP

```
 1: procedure GenPSAT(props $\{x_i\}_{i=1}^n$, form $\Gamma$, probform $\Sigma$)
 2:     declare: binary variables: $h_{ij}$ for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
 3:              $[0,1]$-variables: $\alpha_i$, $\pi_j$, $b_{ij}$ for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,k'\}$
 4:              real variable: $\varepsilon$
 5:     for $j = 1$ to $k'$ do
 6:         for each $(\bigvee_r x_r) \vee (\bigvee_s \neg x_s)$ in $\Gamma$ do
 7:             add_const($\sum_r h_{rj} + \sum_s (1 - h_{sj}) \geq 1$)                ▷ (gamma)
 8:     for $i = 1$ to $n$ do
 9:         add_const($\sum_j b_{ij} = \alpha_i$)                                      ▷ (val1)
10:         for $j = 1$ to $k'$ do
11:             add_const($0 \leq b_{ij} \leq h_{ij}$)                                 ▷ (val2)
12:             add_const($h_{ij} - 1 + \pi_j \leq b_{ij} \leq \pi_j$)                 ▷ (val2)
13:     $aux \leftarrow 0$
14:     for each $\sum a_i \cdot \mathsf{Pr}(x_i) \bowtie q$ in $\Sigma$ do
15:         switch($\bowtie$)
16:             case " $\geq$ " :
17:                 add_const($\sum a_i \cdot \alpha_i \geq q$)                        ▷ (prob$_\geq$)
18:             case " $<$ " :
19:                 $aux \leftarrow 1$
20:                 set_obj(max $\varepsilon$)                                         ▷ (obj)
21:                 add_const($\sum a_i \cdot \alpha_i + \varepsilon \leq q$)          ▷ (prob$_<$)
22:             case " $\neq$ " :
23:                 $aux \leftarrow 1$
24:                 $z \leftarrow$ fresh()                          ▷ $z$ is a fresh binary variable
25:                 $C \leftarrow 1 + |q| + \sum |a_i|$
26:                 set_obj(max $\varepsilon$)                                         ▷ (obj)
27:                 add_const($\sum a_i \cdot \alpha_i - C \cdot z - \varepsilon \geq q - C$)   ▷ (prob$_\neq$)
28:                 add_const($\sum a_i \cdot \alpha_i - C \cdot z + \varepsilon \leq q$)       ▷ (prob$_\neq$)
29:     add_const($\sum \pi_i = 1$)                                                    ▷ (sums1)
30:     if mip_sat() then
31:         if $(aux == 0)$ or $(aux == 1$ and mip_objvalue() $> 0)$ then
32:             return Sat
33:     return Unsat
```

$$\text{where}\quad \Delta = \{q - \textstyle\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid (\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) < q) \in \Sigma\} \;\cup$$
$$\cup\; \{C \cdot z_\varphi^* + q - \textstyle\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\} \;\cup$$
$$\cup\; \{C - C \cdot z_\varphi^* - q + \textstyle\sum_{i=1}^{\ell}(a_i \cdot \alpha_i^*) \mid \varphi \in \Sigma \text{ is of the form } \sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q\},$$

and, for each atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \neq q$,

$$z_\varphi^* = \begin{cases} 0, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha_i^* < q \\ 1, & \text{if } \sum_{i=1}^{\ell} a_i \cdot \alpha_i^* > q \end{cases}.$$

Now let us check that each linear constraint introduced into the MIP problem at Algorithm 1 is satisfied.

(gamma) $\{h^*_{ij}\}$ satisfy the constraints modeling Γ since each $v \in V_0$ satisfies Γ.

(val1) By definition of $\{b^*_{ij}\}$ and $\{h^*_{ij}\}$, we actually have

$$\sum_{j=1}^{k'} b^*_{ij} = \sum_{j=1}^{k'} h^*_{ij} \cdot \rho_j = \sum_{j=1}^{k'} v_j(x_i) \cdot \rho_j = \sum_{\{j|v_j(x_i)=1\}} \rho_j = \alpha^*_i \ .$$

(val2) Since $0 \le v_j(x_i) \le 1$ and $0 \le \rho_j \le 1$ we immediately have $0 \le b^*_{ij} \le h^*_{ij}$.

For the other inequality, recall that $h^*_{ij} = v_j(x_i)$ and that $\pi^*_j = \rho_j$ and note that:

- if $h^*_{ij} = 0$ then $b^*_{ij} = 0$ and, since $\pi^*_j \le 1$, it follows that $\pi^*_j - 1 \le b^*_{ij} \le \pi^*_j$, i.e.,

$$h^*_{ij} - 1 + \pi^*_j \le b^*_{ij} \le \pi^*_j$$

- if $h^*_{ij} = 1$ then $b^*_{ij} = \pi^*_j$ and so $\pi^*_j \le b^*_{ij} \le \pi^*_j$, i.e., $h^*_{ij} - 1 + \pi^*_j \le b^*_{ij} \le \pi^*_j$

(sums1) Since $\pi^*_j = \rho_j$, we immediately conclude that $\sum\limits_{j=1}^{k'} \pi^*_j = 1$.

To check that the probabilistic formulas are satisfiable, just note that, given a probabilistic formula $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \bowtie q) \in \Sigma$,

$$\sum_{i=1}^{\ell} a_i \cdot \alpha^*_i = \sum_{i=1}^{\ell} a_i \left(\sum_{\{j|v_j(x_i)=1\}} \rho_j \right) = \sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j \right).$$

(prob$_\ge$) Let $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \ge q) \in \Sigma$ and notice that, since ρ satisfies conditions (i), (ii), (iii), in particular it satisfies all the probabilistic formulas in Σ, and so $\sum_{i=1}^{\ell} a_i \left(\sum_{j=1}^{2^n} v_j(x_i) \cdot \rho_j\right) \ge q$, which implies that $\sum_{i=1}^{\ell} a_i \cdot \alpha^*_i \ge q$.

(prob$_<$) Now, let $(\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) < q) \in \Sigma$ and notice that, in a reasoning very similar to the previous one, we can conclude that $\sum_{i=1}^{\ell} a_i \cdot \alpha^*_i < q$, i.e.

$$q - \sum_{i=1}^{\ell} (a_i \cdot \alpha^*_i) > 0. \tag{7}$$

But we should also note that, since $\varepsilon^* = \min \Delta$, then $\varepsilon^* \le q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)$, and so we obtain

$$\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) + \varepsilon^* \le \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) = q \ .$$

(prob$_\ne$) Finally, let us consider an atomic probabilistic formula $\varphi \in \Sigma$ of the form $\sum_{i=1}^{\ell} a_i \mathsf{Pr}(x_i) \ne q$, and recall once more that since ρ satisfies each probabilistic formula of Σ, we have $\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) \ne q$, in other words, either $q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) > 0$ or $q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) < 0$. Recall the constant C defined as $C = 1 + |q| + \sum_{i=1}^{\ell} |a_i|$ and the definition of z^*_φ and notice that both

$$C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) > 0 \tag{8}$$

and

$$C - C \cdot z^*_\varphi - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) > 0 \tag{9}$$

are verified in either of the above cases. Also note that by definition of ε^*, $\varepsilon^* \leq C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)$ and $\varepsilon^* \leq C - C \cdot z^*_\varphi - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)$. Hence, we now analyze each of the previous cases:

- if $q > \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)$, then $z^*_\varphi = 0$ and it follows that
$$\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C \cdot z^*_\varphi - \varepsilon^* \geq \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C + C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) = q - C,$$
and further,
$$\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C \cdot z^*_\varphi + \varepsilon^* \leq \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) + C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) = q.$$
- if $q < \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)$, then $z^*_\varphi = 1$ and it follows that
$$\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C \cdot z^*_\varphi - \varepsilon^* \geq \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C - (C - C \cdot z^*_\varphi - q + \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i)) = q - C,$$
and further,
$$\sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C \cdot z^*_\varphi + \varepsilon^* \leq \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) - C + C \cdot z^*_\varphi + q - \sum_{i=1}^{\ell}(a_i \cdot \alpha^*_i) = q.$$

To finish the direct implication, notice that $\varepsilon^* > 0$ as a consequence of (7), (8) and (9), and it takes the maximum possible value since otherwise, let φ_Δ be the formula in Σ which has the minimum value in Δ. Then, if there was a solution with greater objective value it would violate the constraint ($\mathsf{prob}_{\bowtie}$) for φ_Δ.

Reciprocally, assume that Algorithm 1 returned Sat, and let us denote by h^*_{ij}, α^*_i, ε^* and π^*_j the (optimal) solution for the variables h_{ij}, α_i, ε and π_j, for each $1 \leq i \leq n$, $1 \leq j \leq k'$ respectively.

Consider the set of valuations $\mathcal{V}_0 = \{v_1, \ldots, v_{k'}\}$ where, for each propositional variable $x_i \in \mathcal{P}$, $v_j(x_i) = h^*_{ij}$. Due to constraints (gamma) it is immediate to conclude that each valuation satisfies Γ. Then, let the probability distribution π be defined over the set of valuations as the 2^n vector $\pi = [\rho_j]$ where $\rho_j = \pi^*_j$ for $1 \leq j \leq k'$ and $\rho_j = 0$ for $k' < j \leq 2^n$. Note that (sums1) implies that π is a probability vector. The third condition described in Lemma 4.1 is deduced by simple inspection of the linear constraints ($\mathsf{prob}_{\geq}$), ($\mathsf{prob}_{<}$), ($\mathsf{prob}_{\neq}$) and (sums1), by definition of the matrix associated to Σ over $\mathcal{V}_0$ and recalling that the optimal value ε^* is such that $\varepsilon^* > 0$. □

As a corollary of the previous propositions, we obtain the following result.

Theorem 4.4 *The* GenToMIP *algorithm is a correct translation of* GenPSAT *to a* MIP *problem of polynomial size.*

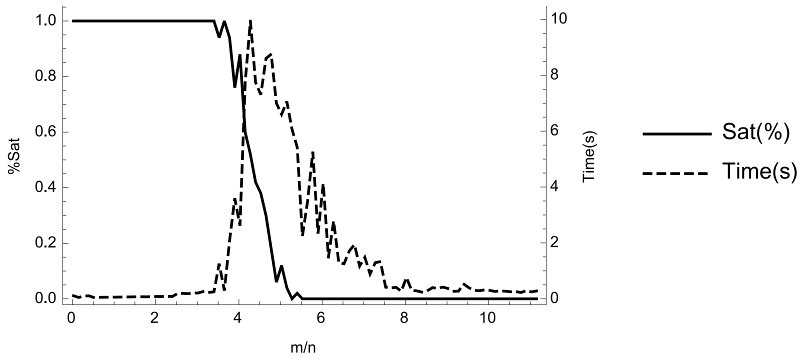
5 Phase Transition

Phase transition is a phenomenon that marks a hardness shift in the solution of instances of a problem. This behaviour was observed in many NP-complete problems [7], among which we highlight 3-SAT [16] and PSAT [11,12].

In this section, we study the GenPSAT phase transition, through an implementation of Algorithm 1 and tests comprised of batteries of random instances. For this, we measure the proportion of satisfiable instances as well as the average time the solver spent to solve them. The software was written in Java, and we used Gurobi [15], version 6.5.0, to solve the MIP problem. The machine used for the tests was a Mac Pro at 3,33 GHz 6-Core Intel Xeon with 6 GB of memory. Our implementation is available in [8].

It was noted that, in random 3-SAT instances [16] there is a clear stage where the instances are almost surely satisfiable and one where they are almost surely not satisfiable. This phenomenon is characterized by the existence of a threshold value for the ratio m/n, where m is the number of clauses, and n is the number of variables, for which: for smaller values of the ratio, the SAT instances are almost certainly satisfiable and easily solved, whereas instances with larger ratio values are almost certainly unsatisfiable and also easily solved. However, with values of the ratio very closed to this threshold, the instances are, on average, very hard to solve and there is no certainty on whether the problem is satisfiable or not. As we have already noted, any 3-SAT problem can be seen as a GenPSAT instance. We tested our GenPSAT solver with random instances of 3-SAT, and observed that a phase transition occurs when the ratio m/n is about 4.3, in accordance with [16], see Figure 1.

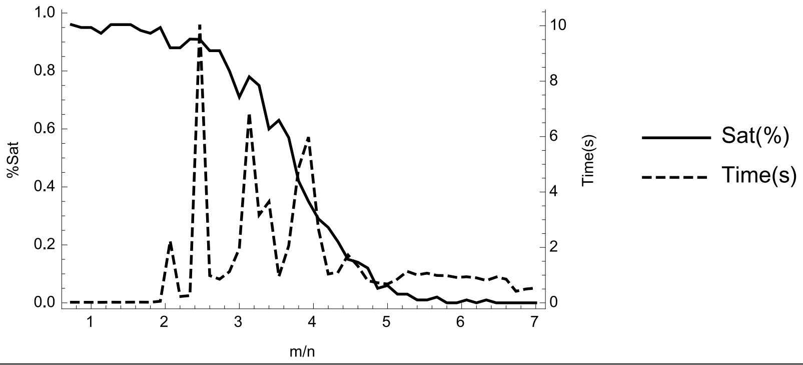
Fig. 1 Phase transition for SAT seen as a GenPSAT instance, with $n = 20$.



A deeper analysis of the probabilistic satisfiability problem PSAT [11,12] has shown the presence of a phase transition behaviour for PSAT for a ratio m/n, where m is the number of clauses and n is the number of variables. We tested random PSAT instances with the number of probabilistic formulas $k = 2$, $n = 15$ and m ranging from 1 to 105 in steps of 2. For each value of m, we generated 100 PSAT instances. The obtained results are presented in Figure 2.

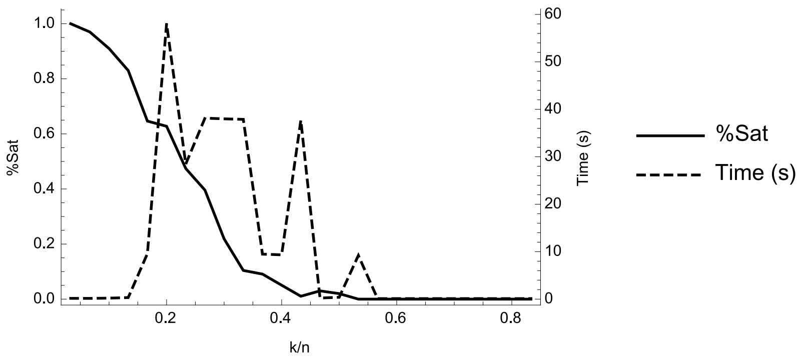
We highlight that the analysis of the existence of a phase transition with variation on k (instead of a variation on m) is essential for a deep understanding of the phase transition of the probabilistic satisfiability problem (instead of the phase transition of the satisfiability problem for propositional formulas in the presence of probabilistic formulas). For this purpose, we tested random PSAT instances with $n = 30$, $m = 40$ and k ranging from 1 to 25, and also observed a phase transition

Fig. 2 PSAT phase transition seen as a GenPSAT instance, with $n = 15$ and $k = 2$.



with respect to k/n based on 100 instances for each value of k, see Figure 3.

Fig. 3 PSAT phase transition seen as a GenPSAT instance, with $n = 30$ and $m = 40$.



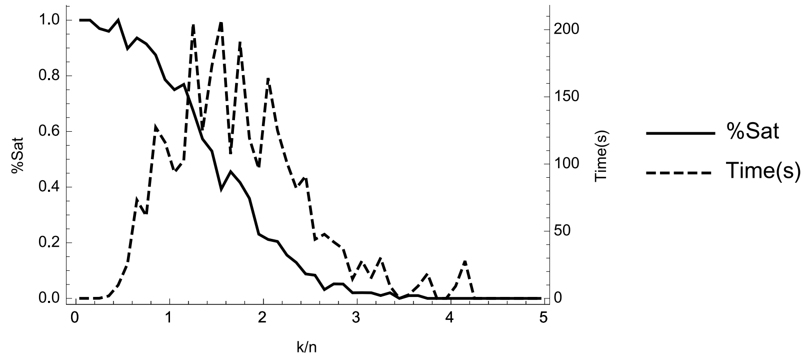
In [1], this phase transition analysis was performed on a generalization of the probabilistic satisfiability problem, GPSAT, which consists in Boolean combinations of simple probabilistic formulas.

In what concerns our generalized version of probabilistic satisfiability GenPSAT, notice that a randomly sampled probabilistic formula can easily be inconsistent by itself, e.g., when it implies one of the probabilities is greater than 1. Because of this, the sampling of the coefficients was performed in such a way that this case does not occur.

GenPSAT gives us a wider scope of ratios to study the phase transition behaviour. Due to its generalized nature, we have four dimensions to explore: the number of variables n, the number of clauses m, the number of probabilistic formulas k and the maximum size of the linear combination into the probabilistic formulas ℓ. We analyze the presence of phase transition for the ratios k/n and m/n and address the analysis of the phase transition for the variation of ℓ/n in future work.

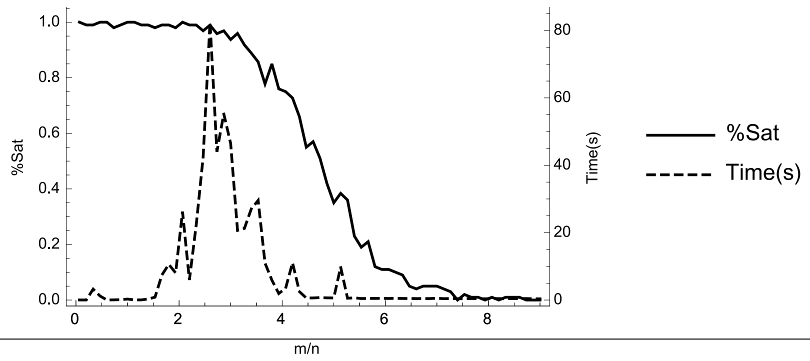
By performing random tests, we observe the presence of a phase transition for the ratio of k/n with a very short stage of satisfiable formulas. This is explained since a GenPSAT instance is more likely to be unsatisfiable. Figure 4 represents the phase transition for random GenPSAT instances with $n = 20$, $m = 10$ and k ranging from 1 to 100 in steps of 2. We generated 100 instances for each value of k.

Fig. 4 Phase transition for random GenPSAT instances, with $n = 20$ and $m = 10$.



On the other hand, when the parameters n and k are fixed, we are also able to detect a phase transition. Figure 5 represents the result of testing random GenPSAT instances with $n = 15$, $k = 2$ and m ranging from 1 to 105 in steps of 2. For each value of m we generated 100 GenPSAT instances.

Fig. 5 Phase transition for random GenPSAT instances, with $n = 15$ and $k = 2$.



6 Conclusion and future work

Throughout this work we explored a generalized version of probabilistic satisfiability, GenPSAT. Capitalizing on its NP-completeness, we presented a polynomial reduction from GenPSAT to MIP, which was proved to be correct. Since the translated MIP problem only suffers a quadratic growth, we were able to solve reasonably sized instances for different values of the parameters: number of variables, clauses and probabilistic formulas. Seeing that an instance can be parametrized by different combinations of these parameters, we are able to make a rich analysis of the phase transition, by analyzing the behaviour for different ratios. As future work, we leave open the study of the phase transition taking into account also the size of the linear combination in the probabilistic formulas, as well as a 4^{th}-dimensional analysis on the variation of the parameters.

We built a tool that implements this algorithm, which although being able to solve reasonably sized instances, can be greatly improved and optimized. In this sense, we are exploring the reductions of GenPSAT to SMT and SAT, which could significantly enhance the solver given the performance of the available tools for these problems. We also leave as future work the study of the relationship between GenPSAT and *weighted* MaxSAT.

Soon, we expect to develop applications of GenPSAT to model problems in several contexts, namely in the automated analysis of security protocols and estimation of probabilities for the existence of offline guessing attacks, [17]. For this purpose, we are currently developing an automated tool that uses the GenPSAT solver to reason about probabilistic formulas involving equations and domain restrictions.

References

[1] G.D. Bona, F. G. Cozman, and M. Finger. Generalized probabilistic satisfiability through integer programming. *Journal of the Brazilian Computer Society*, 21(1):1–14, 2015.

[2] A. Biere, M. Heule, and H. van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[3] G. Boole. *Investigation of The Laws of Thought On Which Are Founded the Mathematical Theories of Logic and Probabilities.* 1853. Also available from Dover, New York 1958, ISBN 0-486-60028-9.

[4] V. Chandru and J. Hooker. *Optimization methods for logical inference.* Wiley-Interscience series in discrete mathematics and optimization. John Wiley and sons, Inc, New York, 1999.

[5] V. Chvátal. *Linear programming.* Macmillan, 1983.

[6] F. G. Cozman and L. F. Ianni. *ECSQARU 2013. Proceedings*, chapter Probabilistic Satisfiability and Coherence Checking through Integer Programming, pages 145–156. Springer Berlin Heidelberg, 2013.

[7] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–340, 1991.

[8] F. Casal, A. Mordido, and C. Caleiro. Genpsat solver, 2016. Available online at https://github.com/fcasal/genpsat.git.

[9] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[10] L. De Moura and N. Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.

[11] M. Finger and G.D. Bona. Probabilistic satisfiability: Logic-based algorithms and phase transition. In *IJCAI*, pages 528–533. IJCAI/AAAI, 2011.

[12] M. Finger and G.D. Bona. Probabilistic satisfiability: algorithms with the presence and absence of a phase transition. *Annals of Mathematics and Artificial Intelligence*, 75(3):351–389, 2015.

[13] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87(1-2):78–128, 1990.

[14] G. Georgakopoulos, D. Kavvadias, and C. H Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4(1):1 – 11, 1988.

[15] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[16] I. P. Gent and T. Walsh. *The hardest random SAT problems.* Springer, 1994.

[17] A. Mordido and C. Caleiro. Probabilistic logic over equations and domain restrictions - full version. 2015. SQIG - Instituto de Telecomunicações and IST - U Lisboa, Portugal. Submitted for publication. Available online at http://sqig.math.ist.utl.pt/pub/CaleiroC/15-MC-probeq.pdf.

[18] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28(1):71–88, February 1986.

[19] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Books on Computer Science. Dover Publications, 1982.

[20] Jason Rosenhouse. *The Monty Hall problem: the remarkable story of Math's most contentious brain teaser.* Oxford University Press, 2009.

[21] G. S. Tseitin. On the complexity of derivations in the propositional calculus. *Studies in Mathematics and Mathematical Logic*, Part II:115–125, 1968.