

RESEARCH

Open Access



Generalized probabilistic satisfiability through integer programming

Glauber De Bona^{1*}, Fabio G. Cozman² and Marcelo Finger¹

Abstract

Background: This paper studies the generalized probabilistic satisfiability (GPSAT) problem, where the probabilistic satisfiability (PSAT) problem is extended by allowing Boolean combinations of probabilistic assertions and nested probabilistic formulas.

Methods: We introduce a normal form for this problem and show that both nesting of probabilities and multi-agent probabilities do not increase the expressivity of GPSAT. An algorithm to solve GPSAT instances in the normal form via mixed integer linear programming is proposed.

Results: The implementation of the algorithm is used to explore the complexity profile of GPSAT, and it shows evidence of phase-transition phenomena.

Conclusions: Even though GPSAT is considerably more expressive than PSAT, it can be handled using integer linear programming techniques.

Keywords: Probabilistic logic; Probabilistic satisfiability; Mixed integer linear programming

Background

Propositional logic and probability theory stand as major knowledge representation tools in many fields, and notably in artificial intelligence. Useful combinations of propositional logic and probability theory are already pursued by Boole [3, Chapter XVIII], who was concerned with problems where propositional formulas are associated with probability assertions. Loosely speaking, we have propositional sentences $\{\phi_i\}_{i=1}^q$, each containing a subset of atomic propositions $\{A_j\}_{j=1}^n$. We may associate one or more of these sentences with probabilities, writing for instance $P(\phi_i) = \alpha_i$. To establish semantics for these assessments, we consider a probability measure over the set of truth assignments. The *probabilistic satisfiability* (PSAT) problem is to determine whether it is possible to find a probability measure over truth assignments such that all assessments are satisfied [13]. PSAT problems have received attention in a variety of fields [4, 9, 14, 15, 17]; in artificial intelligence research, PSAT problems appear

as a foundation for probabilistic rules [21] and first-order probabilistic logic [16, 20, 22].

In this paper, we consider an extended version of PSAT problems. To illustrate the PSAT problem, consider the following situation:

Problem 1. Three friends, Alice, Bob, and Charlie, go to the pub everyday and they have two options: “Bar Phi” and “Bar Not Phi”. Each friend goes exactly to one bar per night. Talking about tonight, a fourth friend, David, says that Alice goes to “Bar Phi” with probability $6/7$, that Bob goes to “Bar Phi” with probability $5/7$; and that the probability of Charlie drinking at “Bar Phi” is $4/7$. Furthermore, David states that exactly two of them will be at “Bar Phi” tonight.

Question: Is David being consistent?

This problem can be captured by the PSAT formalism, as it will be shown later. The idea is to check the consistency of the probabilities assigned to each friend going to “Bar Phi” tonight together with the fact that exactly two of them being there with probability one. However flexible it may be, PSAT problem can only handle conjunction

*Correspondence: debona@ime.usp.br

¹ Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, São Paulo, Brazil

Full list of author information is available at the end of the article

of probability assessments. To understand our goal, note that the satisfiability of probabilistic formulas $P(\phi_i) = \alpha_i$, $1 \leq i \leq q$ obtains when there is a probability measure (over the truth assignments) that satisfies

$$P(\phi_1) = \alpha_1 \wedge P(\phi_2) = \alpha_2 \wedge \cdots \wedge P(\phi_q) = \alpha_q.$$

One obvious generalization is to deal with disjunction and negation of probabilistic assertions to construct more complex formulas, such as

$$\neg (P(\phi_1) = \alpha_1) \wedge (P(\phi_2) = \alpha_2) \vee (P(\phi_3) = \alpha_3).$$

We will see that the same semantics used for PSAT can be adopted in such an extended probabilistic satisfiability problem. As an example of the higher expressivity introduced by this generalization, consider a modification of Problem 1:

Problem 2. Three friends, Alice, Bob, and Charlie, go to the pub everyday and they have two options: “Bar Phi” and “Bar Not Phi”. Each friend goes exactly to one bar per night. Talking about tonight, a fourth friend, David, says that Alice goes to her favorite bar with probability $6/7$, that Bob goes to the bar he likes most with probability $5/7$; and that the probability of Charlie drinking at his favorite place is $4/7$. Furthermore, David states that exactly two of them will be at “Bar Phi” tonight. *Question:* Is David being consistent?

The key difference between Problems 1 and 2 is that in the former we have probabilities for each friend to go to “Bar Phi” and we are asked about the consistency of such assignments together with the constraint that exactly two of them are going there tonight; in the latter, the probabilities are assigned to *either* “Bar Phi” *or* “Bar Not Phi”, and we want to check whether David is stating consistent probabilities with which Alice, Bob, and Charlie are going to their favorite bars. Such difference might be clearer in the next sections, where these problems are formalized.

We can move to an even more expressive language by allowing probabilistic formulas to be nested; that is, by allowing a subformula of ϕ in $P(\phi) = \alpha$ to be $P(\psi) = \alpha'$. Nested probabilities may receive different interpretations, and ours can be exemplified with a problem:

Problem 3. Alice says that the probability of her going to “Bar Phi” tonight with probability $5/6$ is $1/2$; and, with probability $1/2$, she is going to “Bar Phi” tonight with probability $1/6$. *Question:* Can these probabilistic statements be consistent?

A possible scenario consistent with what Alice says could be the following: Alice initially wanted to go to the bar, but as she likes some randomness in her life, she decided to throw a fair die tonight. She will go drink unless she gets a 6. In the middle of the day, she is not in the mood. She makes her mind to go to the bar only if the die shows up a 6. But she suddenly realizes she was too conservative, and gives to a fair coin the responsibility of choosing her destiny: in case of heads, she is going out only with a 6 in the die; with the coin landing tails, she is going to the bar unless a 6 happens to be the case. Just for fun, she tossed the coin in the afternoon, but will only see the outcome in the evening—right before throwing the die. In the meantime, she is thinking about the probability of her going out tonight. With probability $1/2$, she thinks she is in a world where the coin landed heads, and the probability of her getting a 6 to drink is $1/6$. But with probability $1/2$, the actual case is with the coin showing tails, and she is going to the bar with probability $5/6$ —only a 6 can spoil her night.

In such a scenario, there is a probability measure over possible worlds (tails and heads) which have their own probability measure over other possible worlds (going to the bar or not). This is the case in the logic devised in ref. [10], which finds its application in stochastic systems, for instance. In this paper, however, we follow a different path, keeping only a single probability measure over all possible worlds. To give meaning to nested probabilities, we use the *lifting* assumption, that “one believes the probability of ϕ is α if, and only if, one believes that the probability of the probability of ϕ being α is 1” [23]. In Problem 3, one can argue that, if the probability of Alice going out is $5/6$ or $1/6$, with probability $1/2$ each, then the probability of her going out is actually $1/2 \times 5/6 + 1/2 \times 1/6 = 1/2$ —with probability one. As Alice states that a probability of a probability being $5/6$ is not one or zero, but $1/2$, she is being inconsistent under the lifting assumption. Clearly, this assumption is restrictive, in the sense that it claims an agent may not have uncertainty about his own probabilistic beliefs. However, maybe surprisingly, under this premise, we can prove that satisfiability of formulas with nested probabilities can be reduced to satisfiability without nesting.

The problem of deciding satisfiability of these formulas is what we define as *generalized probabilistic satisfiability* (GPSAT). The resulting language can be viewed as the closure of probabilistic formulas with respect to Boolean and probabilistic operators.

There are no algorithms currently available to determine satisfiability of GPSAT problems, as far as we know. In this paper, we present the first such algorithm. The most direct way to solve a PSAT problem is through its linear programming formulation [14], using column generation methods to handle the exponential number of

columns [19]. A recent alternative approach reduces PSAT into logical satisfiability [11]. Neither of these approaches is easily extended to deal with disjunction of constraints, that is essential to solve GPSAT.

In this paper, we present an approach to generalized probabilistic satisfiability, where the original problem is written as a mixed integer linear program of a size that is polynomial on the size of the original problem. This technique was firstly proposed to solve probabilistic satisfiability in ref. [6], and this paper builds on that work to deal with GPSAT problems.

The remaining of this section summarizes necessary fundamentals on SAT and PSAT. The generalized probabilistic satisfiability is defined in the section “The problem”, where a normal form is introduced. In the same section, we show how a multi-agent scenario is reduced to the single-agent case. An algorithm for GPSAT is described in the section “Methods”, building on a reduction from PSAT to mixed integer linear programming. Implementation and experiments, with a discussion of phase transitions, are presented in the section “Results”.

Consider the set \mathcal{L}_{LP} of well-formed formulas built with a set $A = \{A_1, A_2, A_3 \dots\}$ of atomic propositions using the usual Boolean connectives \neg , \wedge , \vee and \rightarrow . A truth assignment (or valuation) ω is a function $\omega : A \rightarrow \{0, 1\}$ that takes atomic propositions to truth values, which we denote by 0 and 1. Through the classical semantics of propositional logic¹ ω can have its domain extended to \mathcal{L}_{PL} , assigning a truth value to any propositional formula. If ϕ is true, write $\omega \models \phi$; if ϕ is false, we write $\omega \not\models \phi$. Given a finite a set of formulas $\{\phi_1, \dots, \phi_q\} \subset \mathcal{L}_{PL}$, the satisfiability (SAT) problem is to determine whether or not there exists a truth assignment to all variables such that all sentences evaluate to true [5, 12]. We call such a set a SAT instance.

If every sentence ϕ_i is a conjunction of clauses, then we have a SAT instance in *CNF*. A SAT instance in *CNF* is a k -SAT instance when each clause has k literals. The 2-SAT problem has a polynomial solution, while k -SAT is NP-complete for $k > 2$.

Suppose now we have q propositional sentences over n atomic propositions, say $\phi_1, \dots, \phi_q \in \mathcal{L}_{LP}$, associated with probabilities through *assessments* of the form $P(\phi_i) \bowtie \alpha_i$, where \bowtie is one of \geq , $=$ or \leq . For computability, let the probabilities α_i be rational numbers. The semantics of such an assessment is as follows. Take the set of 2^n truth assignments that can be generated for the n propositions. A probability measure P over this set satisfies the assessments if, for each assessment $P(\phi_i) \bowtie \alpha_i$,

$$\sum_{\omega \models \phi_i} P(\omega) \bowtie \alpha_i. \quad (1)$$

The *probabilistic satisfiability* (PSAT) problem is to determine whether a given set of probabilistic assessments (a

PSAT instance) can be satisfied in the sense that there is a probability measure over truth assignments such that all assessments are satisfied. PSAT is known to be in NP, as it has a small model, and since SAT is a subproblem where all assigned probabilities are 1, PSAT is NP-complete [13]. A few polynomial special cases of PSAT are known [1].

Example 4 (Friends’ Favorite Bar). Consider Problem 1, with friends $\{A, B, C\}$. Let ϕ_i represent that person i is at the “Bar Phi” tonight, $i \in \{A, B, C\}$. As exactly two friends are going to the “Bar Phi”, according to David, no two friends can be absent tonight, represented by $\neg(\neg\phi_i \wedge \neg\phi_j)$ with 100 % certainty, for $i \neq j$:

$$P(\phi_A \vee \phi_B) = P(\phi_A \vee \phi_C) = P(\phi_B \vee \phi_C) = 1.$$

Furthermore, at least one must be at “Bar Not Phi”:

$$P(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1.$$

David assigns the probability for each friend being at “Bar Phi”:

$$P(\phi_A) = 6/7, P(\phi_B) = 5/7, P(\phi_C) = 4/7,$$

and the question is if there exists a probability measure that simultaneously satisfies these seven probability assignments. It happens that there is no such a probability assignment and this PSAT instance is unsatisfiable. We can conclude that David is being inconsistent in his probabilistic assertions. \square

There are many proposed algorithms for PSAT. The most obvious one is to write down q constraints of the form (1), one for each sentence, associated with assignments $P(\phi_i) \bowtie \alpha_i$. Each constraint can be written as

$$\sum_{j=1}^{2^n} \omega_j(\phi_i) P(\omega_j) \bowtie \alpha_i, \quad (2)$$

while truth assignments ω_j are ordered from 1 to 2^n (say by the n -bit binary number obtained by writing 0 for false and 1 for true as assigned to A_1, \dots, A_n). Probabilistic satisfiability is then obtained when the resulting set of linear constraints has a solution. The challenge is that we have 2^n truth assignments, so the size of the linear constraints is exponential in the input.

The most efficient algorithms for PSAT apply linear programming techniques to this set of constraints. As a linear program, if there is a solution, there is one with no more than $q + 1$ truth assignments with positive probability [13]. The two-phase simplex method can be used with the addition of $q + 1$ artificial variables in order to find a feasible solution in the first phase. Starting with a basis with these $q + 1$ artificial variables, at each iteration, a new column (variable) enter the basis, keeping the solution feasible, until the basis has no artificial variables. If this point

is reached, a solution is found; otherwise, the linear program has no feasible solution and the PSAT instance is unsatisfiable. As the number of columns is exponential, column generation techniques are used. A good survey on this approach is [17]. Combining inference rules with linear programming techniques leads to the currently most efficient algorithms, as showed in ref. [18].

The problem

The language we contemplate is syntactically easy to specify: simply take Boolean operations over assessments, and allow a subformula in an assessment to be itself any well-formed formula of the language. That is, we allow sentences such as

$$(P(A_1 \vee \neg A_2) \geq 1/2) \vee \neg (P(A_3) \geq 1/4) \vee (P(A_1) = 1/5),$$

and even sentences where probabilities are nested as follows:

$$\neg (P(A_1) \leq 1/3) \vee P(A_2 \wedge (P(A_3) \geq 3/4)) \geq 1/2,$$

both of which contain a single “clause” of (possibly negated) assessments. Another example is the following, where we find two “clauses” of assessments:

$$((P(A_1) = 1/3) \vee \neg (P(A_2) \leq 3/5)) \wedge \\ ((P(\neg A_1) \geq 2/3) \vee \neg (P(A_1 \vee \neg A_3) = 3/7)).$$

To make sense of these sentences, we must establish an appropriate semantics. Note that in any PSAT instance, the truth value of an assessment such as $P(\phi_i) = \alpha_i$ is derived from a probability measure on the truth assignments $\omega_1, \dots, \omega_{2^n}$ (because the relation $\omega_j \models \phi_i$ is well defined). This is not the case when nested probabilities are allowed and ϕ_i may contain probabilistic subformulas. This next section shows how to define a semantics that characterizes when a probabilistic assessment is true in a possible world ω_j .

Syntax and semantics

We start with an infinite set of atomic (or primitive) propositions $A = \{A_1, A_2, A_3, \dots\}$.

Our language is then defined recursively as the smallest set \mathcal{L} such that:²

- If $\phi \in A$, then $\phi \in \mathcal{L}$;
- If $\phi \in \mathcal{L}$, then $\neg\phi \in \mathcal{L}$;
- If $\phi \in \mathcal{L}$ and $\theta \in \mathcal{L}$, then $(\phi \vee \theta) \in \mathcal{L}$;
- If $\phi \in \mathcal{L}$ and $\theta \in \mathcal{L}$, then $(\phi \wedge \theta) \in \mathcal{L}$;
- If $\phi \in \mathcal{L}$, then $(P(\phi) \bowtie \alpha) \in \mathcal{L}$, for $\bowtie \in \{\leq, \geq\}$ and $\alpha \in [0, 1] \cap \mathbb{Q}$.

Parentheses are omitted whenever possible. As usual, $P(\phi) = \alpha$, $P(\phi) < \alpha$ and $P(\phi) > \alpha$ are abbreviations for $(P(\phi) \leq \alpha) \wedge (P(\phi) \geq \alpha)$, $\neg(P(\phi) \geq \alpha)$ and $\neg(P(\phi) \leq \alpha)$, respectively.

To give truth values to formulas in \mathcal{L} , we define a structure $\mathcal{M} = (\Omega, P(\cdot))$, where Ω is a finite set of propositional truth assignments ω_i (valuations), and $P : \Omega \rightarrow [0, 1]$ is a probability measure over Ω . With this machinery in hand, we now define when a pair (\mathcal{M}, ω) satisfies a formula $\phi \in \mathcal{L}$, or $(\mathcal{M}, \omega) \models \phi$:

- $(\mathcal{M}, \omega) \models A_i$ if $\omega \models A_i$;
- $(\mathcal{M}, \omega) \models \neg\phi$ if $(\mathcal{M}, \omega) \not\models \phi$;
- $(\mathcal{M}, \omega) \models \phi_1 \wedge \phi_2$ if $(\mathcal{M}, \omega) \models \phi_1$ and $(\mathcal{M}, \omega) \models \phi_2$;
- $(\mathcal{M}, \omega) \models \phi_1 \vee \phi_2$ if $(\mathcal{M}, \omega) \models \phi_1$ or $(\mathcal{M}, \omega) \models \phi_2$;
- $(\mathcal{M}, \omega) \models P(\phi) \bowtie \alpha$ if $\sum\{P(\omega_i) \mid (\mathcal{M}, \omega_i) \models \phi\} \bowtie \alpha$, for $\bowtie \in \{\leq, \geq\}$ and $\alpha \in [0, 1] \cap \mathbb{Q}$;

This semantics leads propositional formulas and probabilistic formulas to have their truth value associated to possible worlds (\mathcal{M}, ω) . This means one can talk about the truth value of formulas such as

$$\phi_1 \wedge (P(\phi_2) \leq 0.5) \vee (P(\neg(P(\phi_3) \geq 0.7)) \leq 0.2)$$

in a given pair (\mathcal{M}, ω) . However, as the truth value of $P(\phi) \bowtie \alpha$ in (\mathcal{M}, ω) does not depend on ω , for a fixed \mathcal{M} , $P(\phi) \bowtie \alpha$ will be true either in all pairs (\mathcal{M}, ω) or in none of them – this captures the intuition that nested probabilities collapse to either 1 or 0. We write $\mathcal{M} \models \phi$ if $(\mathcal{M}, \omega) \models \phi$ for all $\omega \in \Omega$.

Given a formula $\phi \in \mathcal{L}$, we say that it is *satisfiable* if there is a pair (\mathcal{M}, ω) , where $\mathcal{M} = \{\Omega, P(\cdot)\}$ and $\omega \in \Omega$, such that $(\mathcal{M}, \omega) \models \phi$; otherwise, we say ϕ is *unsatisfiable*.

The *generalized probabilistic satisfiability* (GPSAT) is then the problem of deciding whether or not a given $\phi \in \mathcal{L}$ is satisfiable. In this way, any formula $\phi \in \mathcal{L}$ is a *GPSAT instance*.

Example 5 (Friends Telling the Truth). Consider Problem 2. Using the same formalization as in Example 4, at least two friends must be at “Bar Phi” tonight:

$$P(\phi_A \vee \phi_B) = 1 \quad (3)$$

$$P(\phi_A \vee \phi_C) = 1 \quad (4)$$

$$P(\phi_B \vee \phi_C) = 1. \quad (5)$$

Furthermore, at least one must be at “Bar Not Phi”:

$$P(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1. \quad (6)$$

We want to check if David is being consistent, but we do not know if the probabilities he is claiming correspond to “Bar Phi” or “Bar Not Phi”:

$$(P(\phi_A) = 6/7 \vee P(\neg\phi_A) = 6/7), \quad (7)$$

$$(P(\phi_B) = 5/7 \vee P(\neg\phi_B) = 5/7), \quad (8)$$

$$(P(\phi_C) = 4/7 \vee P(\neg\phi_C) = 4/7). \quad (9)$$

Now the question is if there exists a probability measure that simultaneously satisfies the first four probability

assignments together with at least one probability assignment per each clause above. More precisely, our GPSAT instance is a formula θ , whose satisfiability we want to check, built as the conjunction of the formulas in Expressions (3) to (9).

Consider a structure $\mathcal{M} = (\Omega, P(\cdot))$, in which $\Omega = \{\omega_1, \omega_2, \omega_3\}$ is a set of possible worlds and $P : \Omega \rightarrow [0, 1]$ is a probability measure over it. Let Ω be such that $\omega_1 \models \phi_A \wedge \phi_B \wedge \neg\phi_C$, $\omega_2 \models \phi_A \wedge \neg\phi_B \wedge \phi_C$ and $\omega_3 \models \neg\phi_A \wedge \phi_B \wedge \phi_C$. Make $P(\omega_1) = 4/7$, $P(\omega_2) = 2/7$ and $P(\omega_3) = 1/7$. Note that in each world $\omega \in \Omega$, exactly two of ϕ_A , ϕ_B , and ϕ_C are true, so

$$\mathcal{M} \models P(\phi_i \vee \phi_j) = 1,$$

for all pairs $i \neq j$, satisfying the conjuncts of Expressions (3) to (5). As $\omega \models (\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C)$ for all $\omega \in \Omega$,

$$\mathcal{M} \models P(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1.$$

Finally, we have $\sum\{P(\omega_i) \mid (\mathcal{M}, \omega_i) \models \phi_A\} = 6/7$, $\sum\{P(\omega_i) \mid (\mathcal{M}, \omega_i) \models \phi_B\} = 5/7$ and $\sum\{P(\omega_i) \mid (\mathcal{M}, \omega_i) \models \neg\phi_C\} = 4/7$, yielding:

$$\mathcal{M} \models P(\phi_A) = 6/7,$$

$$\mathcal{M} \models P(\phi_B) = 5/7,$$

$$\mathcal{M} \models P(\neg\phi_C) = 4/7,$$

satisfying the clauses in Expressions (7), (8), and (9), respectively. Hence, $\mathcal{M} \models \theta$ and θ is satisfiable.

We conclude that David is being consistent, with a possible scenario being with only Charlie preferring the “Bar Not Phi”. Indeed, every structure satisfying θ leads to this scenario. Any other assignment of favorite bars for Alice, Bob, and Charlie different from this one (only Charlie preferring “Bar Not Phi”) makes the probabilities inconsistent. Therefore, it can be inferred that this scenario is the case, given that David is consistent. \square

Note that this problem could be solved as a sequence of PSAT instances: we just guess the favorite bar of each friend and decide the corresponding PSAT instance. But to cover all possible assignments of favorite bars leads to an exponential quantity (on the number of friends) of PSAT instances, thus it is a less efficient solution. As we shall see, GPSAT is in *NP*, and there must exist a polynomial reduction from it to PSAT, but there seems to be no simple way of performing it.

The logic presented by Fagin, Halpern, and Megido in ref. [9] had already dealt with Boolean combinations of probabilistic assessments, but they kept the probabilities applied only to pure propositional formulas, hence avoiding nesting. Their semantics assigns truth values only to probability assessments through a whole structure \mathcal{M} , and not a pair (\mathcal{M}, ω) . So their semantics is closer to PSAT’s, even though they use probability “inside” the language. Another difference is that Fagin et al. allow more

general assessments of the form $a_1P(\alpha_1) + a_2P(\alpha_2) + \dots + a_mP(\alpha_m) \geq \alpha$ (this enabled them to axiomatize their logic). They show that the corresponding satisfiability problem is NP-complete, but do not propose algorithms to solve it. One of our goals here is to provide concrete algorithms.

In another work, Fagin and Halpern [10] investigated a more general logic to reason about knowledge and probabilities. Its probabilistic semantics is similar to ours because the probabilistic formulas have truth value in specific states (worlds), but they introduce a probability distribution for each set of indistinguishable possible worlds (states). Additionally, their logic has an epistemic modal relation, so that multi-agent and linear combinations of probabilities are also allowed. Again, they axiomatize their logic, show the complexity of the related decision procedure, but do not provide algorithms. Variations of logics adding probabilities to the propositional language can be found in ref. [7].

The logic we propose here can be seen as a particular case of the logic in ref. [10], where there is only one agent and all possible worlds are indistinguishable (all with identical probability distributions). It follows that GPSAT inherits an NP upper bound proved by Fagin and Halpern (Theorem 4.6 in ref. [10]); and as PSAT is a subproblem of GPSAT, the latter is also NP-complete.

GPSAT normal form

The algorithm for GPSAT to be proposed decides satisfiability only for formulas without nested probabilities and with all propositional formulas in a probability assessment (like in PSAT). That is, the algorithm cannot handle a formula such as $P(P(\phi) \geq 0.2) \leq 0.1$ or $\phi_1 \wedge P(\phi_2) \leq 0.5$. As we now show, any GPSAT instance can be reduced in polynomial time to a normal form that complies with these constraints and preserves satisfiability.

We say a formula $\phi \in \mathcal{L}$ is in *normal form* if it is the conjunction of two formulas $\Psi \wedge \Gamma$, where

- Ψ is the conjunction of 3-clauses over probabilistic assessments in the form $P(A_i) \geq \alpha$, in which each assessment is over a different atomic proposition A_i , and
- Γ is a probabilistic assignment $P(\gamma) \geq 1$, in which γ is a conjunction of propositional 3-clauses.

Note that probability values that are smaller than one can be assigned only to atomic propositions, and additionally, every atomic proposition occurrence is a subformula of no more than one probabilistic assessment in Ψ .

An example of a formula in normal form is:

$$((P(A_1) \geq 0.3) \vee \neg(P(A_2) \geq 0.1) \vee \neg(P(A_3) \geq 0.4)) \wedge (P((A_1 \vee \neg A_2 \vee A_3) \wedge (A_1 \vee A_2 \vee \neg A_3)) \geq 1)).$$

This normal form is based on the PSAT normal form introduced by Finger and De Bona [11], and, although it may seem quite restrictive, we can show that all formulas in \mathcal{L} can be brought to the normal form. Before we apply techniques from Finger and De Bona, a sequence of intermediate results is needed.

A formula in normal form has no nested probabilities nor propositions outside the scope of a probability assignment. A particular case of a result in ref. [7] (Theorem 1) can be used to eliminate these undesired subformulas. First, we introduce an auxiliary notation. We call a *probabilistic atom* any formula $P(\phi) \bowtie \alpha$ in \mathcal{L} such that $\phi \in \mathcal{L}_{PL}$ is purely propositional. Let $\mathcal{L}' \subset \mathcal{L}$ be the language formed by the Boolean closure of probabilistic atoms:

- If ϕ is a probabilistic atom, then $\phi \in \mathcal{L}'$;
- If $\phi \in \mathcal{L}'$, then $\neg\phi \in \mathcal{L}'$;
- If $\phi \in \mathcal{L}'$ and $\theta \in \mathcal{L}'$, then $(\phi \vee \theta) \in \mathcal{L}'$;
- If $\phi \in \mathcal{L}'$ and $\theta \in \mathcal{L}'$, then $(\phi \wedge \theta) \in \mathcal{L}'$.

Now we can state the result we need:

Lemma 6. For every $\phi \in \mathcal{L}$, there exists $\theta \in \mathcal{L}'$, such that ϕ is satisfiable if, and only if, θ is; furthermore, θ is computed in polynomial time.

Proof Sketch. Let I be the set of indexes of all atomic propositions A_i occurring in ϕ outside the scope of a probability assignment. To build ϕ_0 , for all $i \in I$, substitute $P(B_i) \geq 1$, where B_i is a fresh atomic proposition, for all occurrences of A_i out of a probability assessment; this is done in linear time in the size of ϕ .

Given ϕ_0 , construct ϕ'_0 by substituting a new atomic proposition C_i for a expression $P(\psi) \bowtie \alpha$ (\bowtie is \leq or \geq) inside the scope of probability assessment. Define $\phi_1 = \phi'_0 \wedge ((P(C_i) \geq 1) \vee P(C_i) \leq 0) \wedge (\neg(P(C_i) \geq 1) \vee (P(\psi) \bowtie \alpha)) \wedge ((P(C_i) \geq 1) \vee \neg(P(\psi) \bowtie \alpha))$. Note that ϕ_1 can be computed in polynomial time. Construct ϕ_{m+1} from ϕ_m until there is no more nesting of probabilities, obtaining $\theta \in \mathcal{L}'$. As the number of nested probabilities has a linear upper bound on the size of ϕ , the whole process of building θ is polynomial in time. To verify that θ is satisfiable iff ϕ is, see the complete proof (Theorem 1 in ref. [7]). \square

So, by using Lemma 6, we have a formula which is a Boolean combination of probability assignments over pure propositional formulas. If each probabilistic atom of a formula ϕ is replaced by a new atomic proposition B_i , then we have a formula θ from classical propositional logic. Using standard techniques, by adding new atoms, we can build a 3-SAT instance θ' which is (Boolean) satisfiable iff θ is. Replace the atomic propositions B_i by the corresponding probabilistic atoms and the new atomic propositions by probabilistic atoms of the form $P(C_i) \geq$

1, where C_i a fresh atomic proposition. Now we have a GPSAT instance ϕ' that is satisfiable only if ϕ is. For our normal form transformation procedure, we start from a formula like ϕ' , which is the conjunction of 3-clauses, each formed from probabilistic atoms or their negation using only \geq . Observe that a probability assessment in the form $P(\psi_i) \leq \alpha_i$ is equivalent to the assignment $P(\neg\psi_i) \geq 1 - \alpha_i$.

Theorem 7. For all formulas $\phi \in \mathcal{L}$, there is a formula $\theta \in \mathcal{L}$ in normal form that is satisfiable iff ϕ is; θ can be computed in polynomial time.

Proof. Given a formula ϕ with 3-clauses of probabilistic atoms, we construct Ψ from ϕ , and Γ from scratch. For each $P(\psi_i) \geq \alpha_i$ in ϕ , substitute a new atomic proposition B_i for ψ_i to construct Ψ . Let Γ be assignment $P(\gamma) \geq 1$, where γ is the conjunction of the clauses corresponding to $(\psi_i) \leftrightarrow B_i, (\neg\psi_i \vee B_i) \wedge (\psi_i \vee \neg B_i)$, for all B_i introduced. Using propositional techniques again, one can transform γ into a 3-SAT in polynomial time, possibly adding fresh atomic propositions. We have built $\theta = \Psi \wedge \Gamma$ in polynomial time, it remains to prove that θ is satisfiable iff ϕ is.

(\leftarrow) Suppose $(\mathcal{M}, \omega_{j*}) \models \phi$, with $\mathcal{M} = (\Omega, P(\cdot))$. We can extend \mathcal{M} to satisfy θ . For each $\omega_j \in \Omega$, make $\omega_j \models B_i$ iff $(\mathcal{M}, \omega_{j*}) \models \psi_i$ to form Ω' . Create a structure $\mathcal{M}' = (\Omega', P(\cdot))$. Observe that $(\mathcal{M}', \omega_{j*}) \models P(B_i) \geq \alpha_i$ iff $(\mathcal{M}', \omega_j) \models P(\psi_i) \geq \alpha_i$, for all i . It follows that $(\mathcal{M}', \omega_{j*}) \models \Psi$. As $\phi_i \leftrightarrow B_i$ holds in all worlds ω_j for all i , $(\mathcal{M}', \omega_{j*}) \models P(\gamma) \geq 1$. Thus $(\mathcal{M}', \omega_{j*}) \models \theta$.

(\rightarrow) Now suppose $(\mathcal{M}, \omega_{j*}) \models \theta$, with $\mathcal{M} = (\Omega, P(\cdot))$. For all $\omega_j \in \Omega$ with $P(\omega_j) > 0$, since $(\mathcal{M}, \omega_{j*}) \models P((\neg\psi_i \vee B_i) \wedge (\psi_i \vee \neg B_i)) \geq 1$, $(\mathcal{M}, \omega_j) \models B_i$ iff $(\mathcal{M}, \omega_j) \models \psi_i$, for all i . Hence $(\mathcal{M}, \omega_{j*}) \models P(B_i) \geq \alpha_i$ iff $(\mathcal{M}, \omega_{j*}) \models P(\psi_i) \geq \alpha_i$, for all i . Finally, $(\mathcal{M}, \omega_{j*}) \models \phi$. \square

Note that a GPSAT instance in normal form has no proposition outside the scope of a probabilistic assignment. This means that if $(\mathcal{M}, \omega_{j*}) \models \phi$, for some $\mathcal{M} = (\Omega, P(\cdot))$ and $\phi = \Psi \wedge \Gamma$ in normal form, then $(\mathcal{M}, \omega_{j*}) \models \phi$ for all $\omega_j \in \Omega$. We may write simply $\mathcal{M} \models \phi$ in such case.

The normal form allows us to see a GPSAT instance $\Psi \wedge P(\gamma) = 1$ as an interaction between a probability problem (represented by Ψ) and a SAT instance γ . Solutions to the instance can be seen as solutions to Ψ constrained by the SAT instance γ . This is formalized as follows.

Lemma 8. A normal form GPSAT instance $\phi = \Psi \wedge P(\gamma) = 1$, with q clauses in Ψ , is satisfiable iff there is a structure $\mathcal{M} = (\Omega, P(\cdot))$ such that $|\Omega| \leq q + 1$, $\mathcal{M} \models \Psi$ and, for all $\omega \in \Omega$, and $\omega \models \gamma$.

Proof. (\leftarrow) Suppose first there is a structure $\mathcal{M} = (\Omega, P(\cdot))$, such that $|\Omega| \leq q+1$, $\mathcal{M} \models \Psi$ and, for all $\omega \in \Omega$, and $\omega \models \gamma$. Trivially, $\mathcal{M} \models P(\gamma) = 1$ and $\mathcal{M} \models \phi$.

(\rightarrow) Now suppose that $\phi = \Psi \wedge P(\gamma) = 1$ is satisfiable. Then there is a pair $\mathcal{M} = (\Omega, P(\cdot))$ such that $(\Omega, P(\cdot)) \models \Psi \wedge P(\gamma) = 1$. Without loss of generality, we can suppose that $P(\omega) > 0$ for all $\omega \in \Omega$ —one could just rule out any $\omega \in \Omega$ with zero probability. If there is a $\omega \in \Omega$ such that $\omega \models \neg\gamma$, then $(\Omega, P(\cdot)) \models P(\neg\gamma) > 0$ and thus $(\Omega, P(\cdot)) \not\models P(\gamma) = 1$, a contradiction. So, $\omega \models \gamma$ for all $\omega \in \Omega$. As $(\Omega, P(\cdot)) \models \Psi$, $(\Omega, P(\cdot))$ satisfies at least one probabilistic atom per clause in Ψ . Let $P(A_{ij}) \bowtie_{i_j} \alpha_{ij}$ be a probabilistic assignment satisfied in the j th-clause, with $\bowtie_{i_j} \in \{\geq, <\}$, so $(\Omega, P(\cdot)) \models P(A_{ij}) \bowtie_{i_j} \alpha_{ij}$, for $1 \leq j \leq q$. Suppose $\mathcal{M} = (\Omega, P(\cdot))$ is such that $(\Omega, P(\cdot)) \models P(A_{ij}) = \beta_{ij}$, for $1 \leq j \leq q$. Using Caratheodory's Lemma [8], there is a $\mathcal{M}' = (\Omega', P'(\cdot))$ such that $\Omega' \subseteq \Omega$, $|\Omega'| \leq q+1$ and $(\Omega', P'(\cdot)) \models P(A_{ij}) = \beta_{ij}$ for all $1 \leq j \leq q$. Hence, $\mathcal{M}' \models P(A_{ij}) \bowtie_{i_j} \alpha_{ij}$ for $1 \leq j \leq q$ and $\mathcal{M}' \models \Psi$. Finally, as $\Omega' \subseteq \Omega$, $\omega \models \gamma$ for all $\omega \in \Omega'$ \square

A multi-agent setup

The language of our logic can be extended to deal with more than one probability measure. This may be useful in the case where different agents might assess different probability measures to the same set of events. For instance, we can modify Problem 1 to consider three probability measures, one for each friend.

Problem 9. Three friends, Alice, Bob, and Charlie, go to the pub everyday and they have two options: “Bar Phi” and “Bar Not Phi”. Each friend goes exactly to one bar per night. Talking about tonight, each friend has its own probability assessments: Alice says that she goes to “Bar Phi” with probability $6/7$, Bob tells he goes to “Bar Phi” with probability $5/7$; and Charlie states that the probability of him drinking at “Bar Phi” is $4/7$. Furthermore, the three friends agree that exactly two of them will be at “Bar Phi” tonight.

Question: Are Alice, Bob, and Charlie being consistent?

While in Problem 1, there is one probability measure (David's), in Problem 9, Alice, Bob, and Charlie have different opinions about the probabilities, and each one states his own belief. To formalize this situation, we extend the language of GPSAT to consider the case with N probability measures, so that probability assignments may constrain the values of one out of the N different probability measures. We call this language \mathcal{L}_N , and its syntax is identical to \mathcal{L} 's except from the construction of the probability assessment:

- If $\phi \in \mathcal{L}$, then $(P_i(\phi) \bowtie \alpha) \in \mathcal{L}$, for $\bowtie \in \{\leq, \geq\}$, $\alpha \in [0, 1] \cap \mathbb{Q}$ and $i \in \{1, 2, \dots, N\}$.

To give truth values to formulas in \mathcal{L}_N , we define a structure $\mathcal{M} = (\Omega, \Pi)$, where Ω is a finite set of propositional truth assignments ω_i (as before), and Π is a set of N probability measures $P_i : \Omega \rightarrow [0, 1]$ for $1 \leq i \leq N$. For a formula $\phi \in \mathcal{L}_N$, we change the definition of $(\mathcal{M}, \omega) \models \phi$ by modifying only the semantics of the probability assignment:

- $(\mathcal{M}, \omega) \models P_i(\phi) \bowtie \alpha$ if $\sum \{P_i(\omega_j) \mid (\mathcal{M}, \omega_j) \models \phi\} \bowtie \alpha$, for $\bowtie \in \{\leq, \geq\}$, $\alpha \in [0, 1] \cap \mathbb{Q}$ and $i \in \{1, 2, \dots, N\}$.

It is clear that, if $N = 1$, the former GPSAT logic is recovered. Using the language \mathcal{L}_3 , we can model Problem 9:

Example 10 (Friends' Favorite Bar Revisited). Let ϕ_i , with $i \in \{A, B, C\}$, be the same as Example 4. Let P_1 , P_2 , and P_3 be the probability measures that Alice, Bob, and Charlie, respectively, assign over the possible worlds. We then assign, for each friend, the probability of him being “Bar Phi” tonight, constraining his own probability measure:

$$P_1(\phi_A) = 6/7, P_2(\phi_B) = 5/7, P_3(\phi_C) = 4/7,$$

As they agree that two friends are going to the “Bar Phi”, the formulas $\neg(\neg\phi_i \wedge \neg\phi_j)$, for $i \neq j$, must have 100 % probability in all three probability measures:

$$P_i(\phi_A \vee \phi_B) = P_i(\phi_A \vee \phi_C) = P_i(\phi_B \vee \phi_C) = 1,$$

for $i \in \{1, 2, 3\}$. Furthermore, since they agree that at least one must be at “Bar Not Phi”:

$$P_i(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1,$$

for all $i \in \{1, 2, 3\}$.

The question is if there are three probability measures (P_1 , P_2 , and P_3) that satisfy these 15 probability assignments. We show that it is the case. Let Ω be such that $\omega_1 \models \phi_A \wedge \phi_B \wedge \neg\phi_C$, $\omega_2 \models \phi_A \wedge \neg\phi_B \wedge \phi_C$ and $\omega_3 \models \neg\phi_A \wedge \phi_B \wedge \phi_C$. Let $\Pi = \{P_1, P_2, P_3\}$; and $P_1(\omega_1) = 6/7$, $P_1(\omega_3) = 1/7$, $P_2(\omega_2) = 2/7$, $P_2(\omega_3) = 5/7$, $P_3(\omega_1) = 3/7$, $P_3(\omega_2) = 4/7$, and $P_1(\omega_2) = P_2(\omega_1) = P_3(\omega_3) = 0$. As in Example 4, within each $\omega \in \Omega$, the formulas stating that exactly two friends are going to “Bar Phi” tonight are all satisfied, so they have probability one for the three friends:

$$(\Omega, \Pi) \models P_i(\phi_j \vee \phi_k) = 1,$$

$$(\Omega, \Pi) \models P_i(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1,$$

for all pairs $j \neq k$ and $i \in \{1, 2, 3\}$. Additionally, it is the case that

$$(\Omega, \Pi) \models P_1(\phi_A) = 6/7 \wedge P_2(\phi_B) = 5/7 \wedge P_3(\phi_C) = 4/7.$$

Differently from Example 4, the three friends are each one consistent. However, they do not agree with each other's probabilities; for instance, $P_1(\phi_A) = 6/7 \neq 2/7 = P_2(\phi_A)$. \square

In practice, Example 10 can be seen as three separated GPSAT instances. As the three probability measures are independent from each other, the consistency check can take place within the probability assignments of the same agent. Alice's probability measure has to satisfy only $P(\phi_A) = 6/7$, $P(\neg\phi_A \vee \neg\phi_B \vee \neg\phi_C) = 1$ and $P(\phi_i \vee \phi_j) = 1$ for $i \neq j$. Analogously, Bob's probability measure must model $P(\phi_B) = 5/7$, and Charlie's, $P(\phi_C) = 4/7$, each together with the constraint that exactly 2 of the friends are going to "Bar Phi" tonight. Indeed, this correspondence between satisfiability of formulas in \mathcal{L}_N and GPSAT can be formalized as a polynomial reduction. To prove that, we depart from formulas $\phi \in \mathcal{L}_N$ without nesting, as it can be eliminated with a slight modification of Lemma 6:

Lemma 11. For every $\phi \in \mathcal{L}_N$, for a fixed $N \in \mathbb{N}$, there exists a $\theta \in \mathcal{L}_N$ in which all probabilistic formulas are not subformulas of another probabilistic formulas, such that ϕ is satisfiable if, and only if, θ is. Furthermore, θ is computed in polynomial time and $(\mathcal{M}, w) \models \theta$ implies $(\mathcal{M}, w) \models \phi$.

Proof. To prove by induction, we show how to decrease the number of nested probabilities, keeping the satisfiability and the connection between the models. Given a formula ϕ with nested probabilities, construct ϕ' by substituting a new atomic proposition B for a basic probabilistic formula $P_i(\psi) \bowtie \alpha$, that is subformula of another basic probabilistic formula $P_j(\psi') \bowtie \alpha'$, with $1 \leq i, j \leq N$ and $\bowtie, \bowtie' \in \{\leq, \geq\}$. Define $\phi'' = \phi' \wedge ((P_j(B) \geq 1) \vee (P_j(B) \leq 0)) \wedge (\neg(P_j(B) \geq 1) \vee (P_i(\psi) \bowtie \alpha)) \wedge ((P_j(B) \geq 1) \vee \neg(P_i(\psi) \bowtie \alpha))$. Clearly, this can be done in polynomial time. Now we need to prove that ϕ'' is satisfiable iff ϕ is.

(\leftarrow) Suppose $(\mathcal{M}, \omega^*) \models \phi$, with $\mathcal{M} = (\Omega, \Pi)$. We can change \mathcal{M} to satisfy ϕ'' . For each $\omega \in \Omega$, make $\omega \models B$ iff $(\mathcal{M}, \omega^*) \models P_i(\psi) \bowtie \alpha$ to form Ω' . Create a structure $\mathcal{M}' = (\Omega', \Pi)$. If $(\mathcal{M}, \omega^*) \models P_i(\psi) \bowtie \alpha$, then $(\mathcal{M}', \omega) \models B$ for all $\omega \in \Omega'$ and $(\mathcal{M}', \omega^*) \models P_j(B) \geq 1$. Else, $(\mathcal{M}', \omega_j) \models \neg B$ for all $\omega \in \Omega'$, and $(\mathcal{M}', \omega^*) \models P_j(B) \leq 1$. Anyway, $(\mathcal{M}', w^*) \models ((P_j(B) \geq 1) \vee (P_j(B) \leq 0)) \wedge (\neg(P_j(B) \geq 1) \vee (P_i(\psi) \bowtie \alpha)) \wedge ((P_j(B) \geq 1) \vee \neg(P_i(\psi) \bowtie \alpha))$. Furthermore, as $(\mathcal{M}', \omega) \models P_i(\psi) \bowtie \alpha$ iff $(\mathcal{M}', \omega) \models B$, for all $\omega \in \Omega'$, it is the case that $(\mathcal{M}', \omega^*) \models \phi'$. Hence, $(\mathcal{M}', \omega^*) \models \phi''$.

(\rightarrow) Suppose now $(\mathcal{M}, \omega^*) \models \phi''$. Note that the last two clauses in ϕ'' state that $P_j(B) \geq 1 \leftrightarrow P_i(\psi) \bowtie \alpha$. Therefore, $(\mathcal{M}, \omega^*) \models P_j(B) \geq 1$ iff $(\mathcal{M}, \omega^*) \models P_i(\psi) \bowtie \alpha$. But $(\mathcal{M}, \omega^*) \models P_j(B) \geq 1$ iff $(\mathcal{M}, \omega) \models P_j(B) \geq 1$ for

every $\omega \in \Omega$; and $(\mathcal{M}, \omega^*) \models P_i(\psi) \bowtie \alpha$ iff $(\mathcal{M}, \omega) \models P_i(\psi) \bowtie \alpha$ for every $\omega \in \Omega$. Hence, for every $\omega \in \Omega$, $(\mathcal{M}, \omega) \models P_j(B) \geq 1$ iff $(\mathcal{M}, \omega) \models P_i(\psi) \bowtie \alpha$. Then, due to the clause $((P_j(B) \geq 1) \vee P_j(B) \leq 0)$, for every $\omega \in \Omega$ with $\pi_j(\omega) > 0$, we have that $(\mathcal{M}, \omega) \models B$ iff $(\mathcal{M}, \omega) \models P_i(\psi) \bowtie \alpha$. Finally, as $(\mathcal{M}, \omega_*) \models \phi'$, $(\mathcal{M}, \omega^*) \models \phi$. \square

We now prove the desired reduction.

Theorem 12. Given a formula $\phi \in \mathcal{L}_N$, for a fixed $N \in \mathbb{N}$, there is a formula $\theta \in \mathcal{L}$ such that ϕ is satisfiable if, and only if, θ is; furthermore, θ is computed in polynomial time.

Proof. Using Lemma 11, we first eliminate nested probabilities from ϕ , keeping its (un)satisfiability, and this is done in polynomial time. Suppose, without loss of generality, that ϕ is built using n atomic propositions A_1, A_2, \dots, A_n . To construct θ from ϕ , for each $i \in \{1, 2, \dots, N\}$, transform any probabilistic atom $P_i(\varphi) \bowtie \alpha$ into $P(\varphi^i) \bowtie \alpha$, in which φ^i is constructed from φ by replacing each A_j by $A_{(i-1)*n+j}$. Note that $\theta \in \mathcal{L}$ has N disjoint sets of n atomic propositions, and each set is used to build formulas whose probability was constrained in ϕ for a different probability measure. Clearly, θ is computed in polynomial time on the length of ϕ . It remains to prove that θ is satisfiable if, and only if, ϕ is.

(\rightarrow) Suppose θ is satisfiable, so there is a structure $\mathcal{M} = (\Omega, P(\cdot))$ such that $(\mathcal{M}, \omega^*) \models \theta$ for some $\omega^* \in \Omega$. Let $\Omega' = \{\omega_1, \omega_2, \dots, \omega_{2^n}\}$ be a set with the all possible valuations on A_1, A_2, \dots, A_n —the atomic propositions in ϕ . For each $i \in \{1, 2, \dots, N\}$ and all $1 \leq j \leq 2^n$, make $P_i(\omega_j) = \sum\{P(\omega) \mid \omega \in \Omega \text{ s.t. } \omega(A_{(i-1)N+k}) = \omega_j(A_k) \text{ for all } 1 \leq k \leq n\}$. Make $\Pi = \{P_1, P_2, \dots, P_N\}$ and $\mathcal{M}' = (\Omega', \Pi)$. It follows that, for any probabilistic atom $P_i(\varphi) \bowtie \alpha$ in ϕ , $\mathcal{M}' \models P_i(\varphi) \bowtie \alpha$ iff $\mathcal{M} \models P(\varphi^i) \bowtie \alpha$, where $P(\varphi^i) \bowtie \alpha$ is the corresponding probabilistic atom in θ . Let $\omega^* \in \Omega'$ be such that $\omega^{*}(A_k) = \omega^*(A_k)$ for all $1 \leq k \leq n$. Note that every atomic proposition outside of the scope of a probability assignment in θ is satisfied in (\mathcal{M}, ω^*) iff it is satisfied in (\mathcal{M}', ω^*) . We can conclude that $(\mathcal{M}', \omega^*) \models \phi$.

(\leftarrow) Now suppose ϕ is satisfiable, so there is a structure $\mathcal{M} = (\Omega, \Pi)$, with $\Pi = \{P_1, P_2, \dots, P_N\}$, such that $(\mathcal{M}, \omega^*) \models \phi$ for some $\omega^* \in \Omega$. Without loss of generality, we assume $\Omega = \{\omega_1, \omega_2, \dots, \omega_{2^n}\}$, since if less valuations are needed to satisfy ϕ , others can be inserted with zero probability. Make $\Omega' = \{\omega_{j_1, j_2, \dots, j_N} \mid 1 \leq j_k \leq 2^n, 1 \leq k \leq N\}$ such that $\omega_{j_1, j_2, \dots, j_N}(A_{(i-1)n+k}) = \omega_{j_i}(A_k)$, for $\omega_{j_i} \in \Omega$. That is, each valuation $\omega_{j_1, j_2, \dots, j_N} \in \Omega'$ can be seen as the aggregation of partial valuations $\omega_{j_1}, \dots, \omega_{j_N}$ over the disjoint sets of atomic propositions, $A_1, \dots, A_n, A_{n+1}, \dots, A_{2n}$,

until $A_{(N-1)n+1}, \dots, A_{Nn}$. Each partial valuation ω_{j_k} is a “translation” of $\omega_{j_k} \in \Omega$ to the k th set of atomic propositions $A_{(k-1)n+1}, \dots, A_{(k-1)n+n}$. Let $P(\omega_{j_1, j_2, \dots, j_N})$ be equal to the product $P_1(\omega_{j_1}) \times P_2(\omega_{j_2}) \times \dots \times P_N(\omega_{j_N})$. Create a structure $\mathcal{M}' = (\Omega', P(\cdot))$. As $\sum_{j=1}^{2^n} P_i(\omega_j) = 1$ for all $1 \leq i \leq N$, it follows that $(\Omega, \Pi) \models P_k(\phi) \bowtie \alpha$ iff $(\Omega', P(\cdot)) \models P(\phi^k) \bowtie \alpha$, in which ϕ^k is constructed from ϕ by replacing A_j by $A_{(k-1)n+j}$ for all $1 \leq j \leq n$. Finally, suppose $\omega^* = \omega_q \in \Omega$. There is a $\omega'^* = \omega_{qj_2, \dots, j_N} \in \Omega'$ such that every atomic proposition outside of the scope of a probability assignment in ϕ is satisfied in (\mathcal{M}, ω^*) iff it is satisfied in $(\mathcal{M}', \omega'^*)$; which yields $(\mathcal{M}', \omega'^*) \models \theta$, finishing the proof. \square

This result enables us to focus on how to solve GPSAT, as any formula of \mathcal{L}_N , with probability assignments for different probability measures, can be transformed into a GPSAT instance in polynomial time.

Methods

Assume our GPSAT instance is in normal form with q clauses, each with three (possibly negated) assessments $\{P(A_j) \geq \alpha_j\}$ and a sentence γ in CNF with m clauses, each clause with three literals. So our problem is parameterized by the number of atomic propositions, n , the number of clauses with probability assessments, q , and the number of 3-clauses in γ , m . Such a parameterized normal form neatly separates the probabilistic and the propositional aspects of general probabilistic satisfiability.

Suppose, without loss of generality, the k th clause contains assignments over A_{3k-2} , A_{3k-1} , and A_{3k} . By Lemma 8, if a normal form GPSAT instance is satisfiable, there is a structure with no more than $q+1$ possible worlds that satisfy it. So we can search for only $q+1$ valuations, as long as they all satisfy γ . Our problem becomes: find the $(q+1)$ truth assignments ω_j satisfying γ and the following disjunction of restrictions, for $1 \leq k \leq q$

$$\bigvee_{i=3k-2}^{i=3k} \sum_{j=1}^{q+1} \omega_j(A_i)P(\omega_j) \bowtie_i \alpha_i, \tag{10}$$

where $\bowtie_i \in \{<, \geq\}$.

Hence, we have $3q(q+1)$ optimization variables (values of $\omega_j(A_i)$, denoted by $a_{i,j}$); all of them are binary with values 0 and 1. Furthermore, we have a probability measure over $q+1$ truth assignments, represented by the real-valued variables $p_1, \dots, p_{q+1} \in [0, 1]$, that must sum up to 1. Following the approach of [6], we find $\{a_{i,j}\}$ and $\{p_j\}$ by solving a mixed integer linear program.

Note that, for each k , (10) is a disjunction of constraints and may contain strict inequalities—both characteristics not common to standard integer linear programming formulations. Since probabilities are bounded, one can eliminate disjunctions of inequalities by adding

fresh integer variables, for instance. Strict inequalities like $\sum_{j=1}^{q+1} \omega_j(A_i)P(\omega_j) < \alpha_i$ can be replaced by a non-strict one $\sum_{j=1}^{q+1} \omega_j(A_i)P(\omega_j) \leq \alpha_i - \epsilon$ for a suitably chosen $\epsilon > 0$. We leave these tasks for the integer programming solver to handle. So, we do not formally reduce GPSAT to mixed integer linear programming, but use techniques of the latter (described for instance in ref. [24]) to solve the former.

The elements $a_{i,j}$ for a fixed j corresponds to a truth assignment that satisfies γ . We explore the well-known connection between SAT and integer programming to find such a truth assignment [5]. Start by generating a vector \mathbf{a}_j with n binary variables $\{a_{i,j}\}_{i=1}^n$, all with values 0 and 1. Now take one clause of γ ; suppose it is written as

$$\left(\bigvee_{l'=1}^{k'} A_{i_{l'}}\right) \vee \left(\bigvee_{l''=1}^{k''} \neg A_{i_{l''}}\right).$$

For this clause, generate the linear inequality:

$$\left(\sum_{l'=1}^{k'} a_{k,i_{l'}}\right) + \left(\sum_{l''=1}^{k''} (1 - a_{k,i_{l''}})\right) \geq 1. \tag{11}$$

Consider the m inequalities generated this way (one per clause in γ). A vector \mathbf{a}_j that satisfies these m inequalities yields a truth assignment ω_j for γ such that $\omega_j(A_i) = a_{i,j}$, assigning true to A_i when $a_{i,j}$ is one, and assigning false to A_i when $a_{i,j}$ is zero.

We generate the truth assignments by generating $(q+1)$ sets of variables \mathbf{a}_j and their related inequalities. These valuations correspond to the possible worlds $\omega_j \in \Omega$, with repetitions allowed³. Now we have to force the probability measure over these worlds to satisfy a probabilistic literal in each clause of Ψ . To do so, note that each probabilistic atom in Ψ that is not negated represents an inequality as follows:

$$\sum_{j=1}^{q+1} a_{i,j}p_j \geq \alpha_i, \tag{12}$$

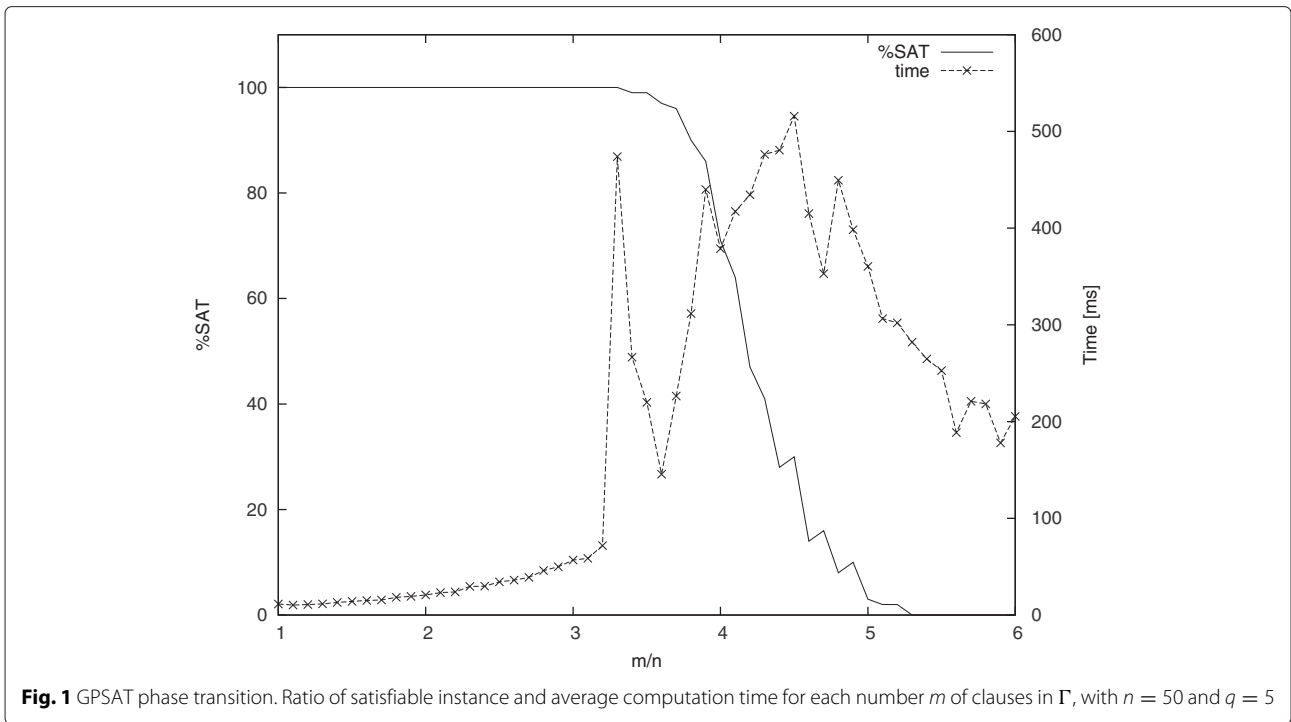
where p_j denotes $P(\omega_j)$. When the atom is negated, the constraint becomes:

$$\sum_{j=1}^{q+1} a_{i,j}p_j < \alpha_i, \tag{13}$$

The inequalities in (12) and (13) do not have to hold for all $3q$ probabilistic atoms. Only one per clause in Ψ must hold. So, we have the following disjunction of constraints for each clause $1 \leq k \leq q$:

$$\bigvee_{i=3k-2}^{3k} \sum_{j=1}^{q+1} a_{i,j}p_j \bowtie_i \alpha_i, \tag{14}$$

where \bowtie_i denotes $<$ if the probabilistic atom $P(A_i) \geq \alpha_i$ is negated; otherwise, it denotes \geq . The challenge is to

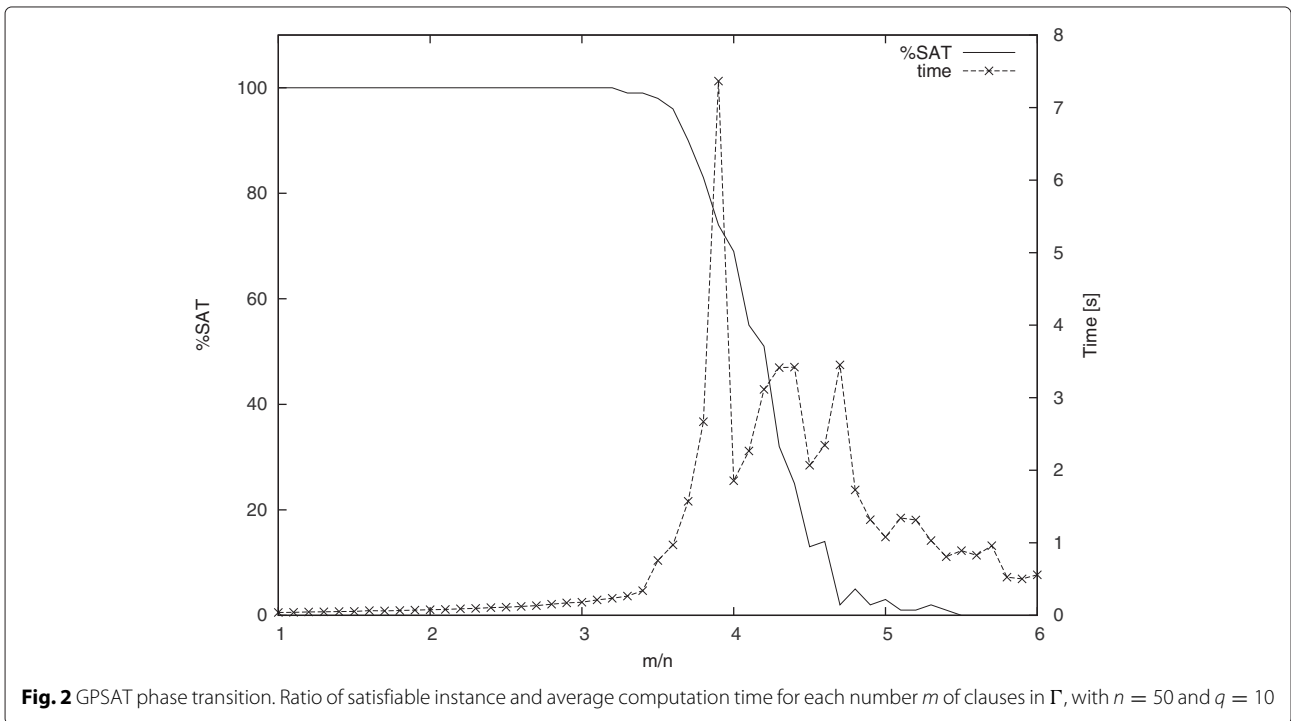


reduce the bilinear term $a_{i,j}p_j$ to linear constraints. We do that by introducing a new fresh variable $b_{i,j}$ and the constraints:

$$0 \leq b_{i,j} \leq a_{i,j} \quad \text{and} \quad a_{i,j} - 1 + p_j \leq b_{i,j} \leq p_j. \quad (15)$$

Note that if $a_{i,j} = 0$, then $b_{i,j} = 0$; and if $a_{i,j} = 1$, then $b_{i,j} = p_j$. The constraints from Expression (14) can be rewritten as:

$$\bigvee_{i=3k-2}^{3k} \sum_{j=1}^{q+1} b_{i,j} \triangleright_i \alpha_i, \quad (16)$$



for each clause $1 \leq k \leq q$ and with $\bowtie_i \in \{<, \geq\}$.

A last restriction ensures $P(\cdot)$ is a probability measure:

$$\sum_{j=1}^{q+1} p_j = 1. \quad (17)$$

The whole procedure is presented in Algorithm 1; it basically collects constraints from Expressions (11), (15), (16), and (17). Note that variables $a_{i,j} \in \{0, 1\}$ are restricted to be integers, while $b_{i,j}$ and p_j range within the real interval $[0, 1]$, leading to an instance of mixed integer linear programming. The algorithm produces a MILP instance that has a solution if and only if the original GPSAT instance is satisfiable.

It remains to prove that the mixed integer program generated has polynomial size on the length of the GPSAT instance and can be generated in polynomial time. First, we sum up the number of variables introduced at the line 1 of Algorithm 1:

- variables $a_{i,j}$: $q + 1$ valuations times n atomic propositions;

Algorithm 1 GPSAT solution based on mixed integer linear programming.

Input: Propositions $\{A_j\}_{j=1}^n$, Ψ with 3-clauses of (possibly negated) assignments over A_{3i-2} , A_{3i-1} and A_{3i} for $1 \leq i \leq q$, a sentence γ , from $\Gamma = P(\gamma) \geq 1$ in CNF with m 3-clauses.

Output: “Satisfiable” if $\Psi \wedge \Gamma$ is satisfiable; “Unsatisfiable” otherwise.

- 1: ▷ Variables $a_{i,j}$ are binary; variables $b_{i,j}$ and p_j are real-valued in $[0, 1]$.
 - 2: **for** $j \in \{1, \dots, q + 1\}$ and each clause $(\bigvee_{l=1}^{k'} A_{i_l'} \vee (\bigvee_{l''=1}^{k''} \neg A_{i_{l''}}))$ of γ **do**
 - 3: Generate linear constraint $(\sum_{l=1}^{k'} a_{i_l',j}) + (\sum_{l''=1}^{k''} (1 - a_{i_{l''},j})) \geq 1$.
 - 4: **end for**
 - 5: **for** each clause $\bigvee_{l=1}^{k'} P(A_{l'}) \geq \alpha_{l'} \vee \bigvee_{l''=1}^{k''} \neg P(A_{l''}) \geq \alpha_{l''}$ **do**
 - 6: Generate disjunction of linear constraints $(\bigvee_{l=1}^{k'} \sum_{j=1}^{q+1} b_{l',j} \geq \alpha_{l'}) \vee (\bigvee_{l''=1}^{k''} \sum_{j=1}^{q+1} b_{l'',j} < \alpha_{l''})$.
 - 7: **end for**
 - 8: **for** $i \in \{1, \dots, 3q\}$ **do**
 - 9: **for** $j \in \{1, \dots, q + 1\}$ **do**
 - 10: Generate linear constraints $0 \leq b_{i,j} \leq a_{i,j}$ and $a_{i,j} - 1 + p_j \leq b_{i,j} \leq p_j$.
 - 11: **end for**
 - 12: **end for**
 - 13: Generate linear constraint $\sum_{j=1}^{q+1} p_j = 1$.
 - 14: Submit constraints to an integer programming solver.
 - 15: **return** Satisfiable if linear constraints have a solution, Unsatisfiable otherwise.
-

- variables $b_{i,j}$: $q + 1$ valuations times $3q$ atomic propositions in Ψ ;
- variables p_j : one per valuation, $q + 1$.

Now we count the constraints created together with the time consumption of the loops in Algorithm 1 that generate them:

- loop at lines 2–4: $(q + 1)m$ inequalities with three variables each;
- loop at lines 5–7: q disjunctions of three inequalities with $q + 1$ variables each;
- loop at lines 8–12: $(q + 1)3q$ pairs of inequalities on three variables each;
- command at line 13: 1 equality with $q + 1$ variables.

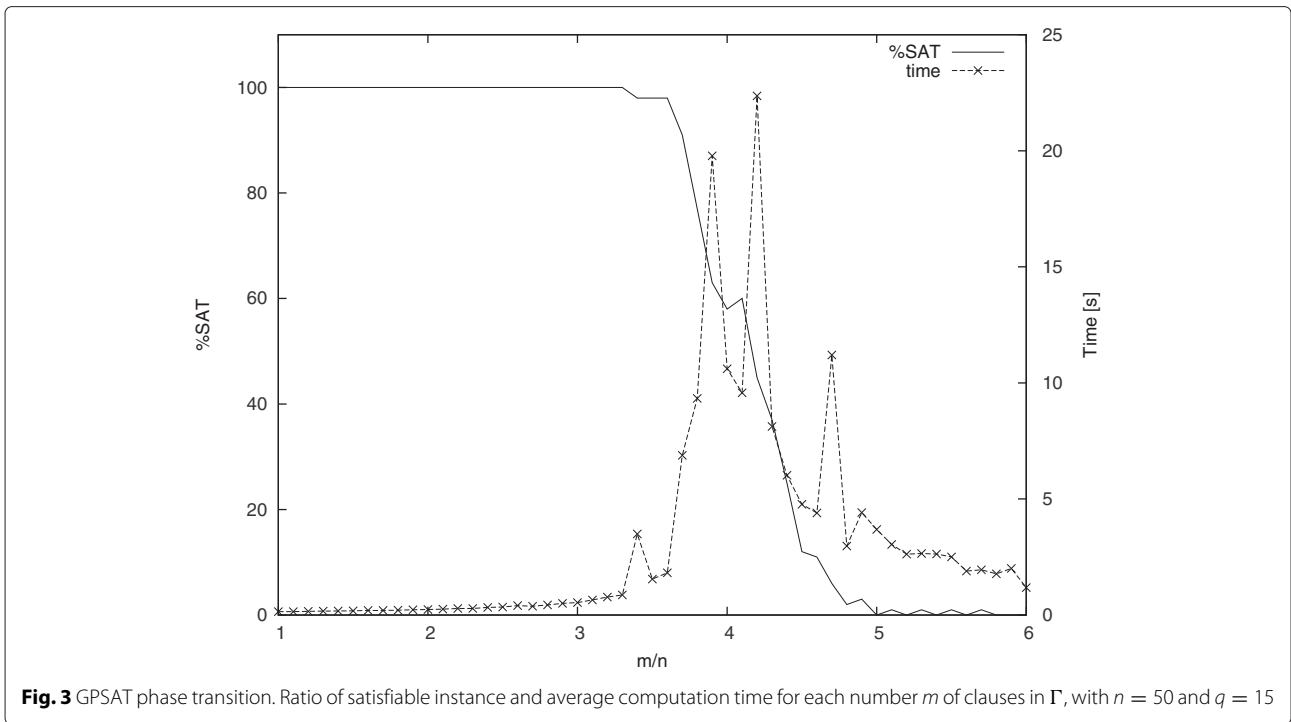
Summing up, Algorithm 1 generates $(m + 6q + 1)(q + 1) + 2q = O((m + q)q)$ inequalities on $(n + 3q + 1)(q + 1) = O(nq)$ variables. Therefore, the resulting mixed integer linear program has polynomial size and is built in polynomial time on the length of the corresponding GPSAT instance. Note that the total running time of Algorithm 1 depends largely on the command at line 14, that calls a solver for the MILP instance; as the decision version of mixed integer linear programming is itself an NP-complete problem, it might take exponential time.

Results and discussion

We have coded our GPSAT method using the Java language with calls to CPLEX version 12, extending the implementation from [6] to solve PSAT, and run experiments in a iMac computer with 4 GBytes of memory.

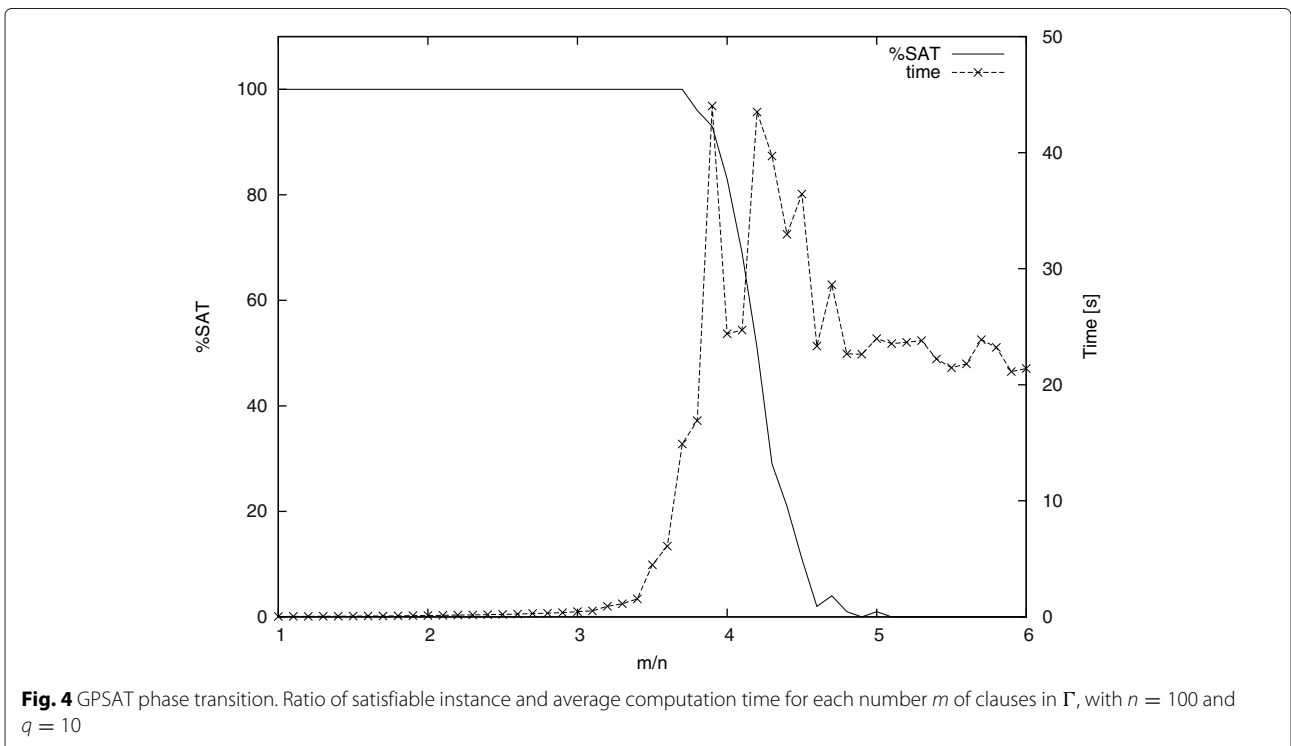
We were particularly interested in investigating whether phase transition phenomena can be identified in our solutions to GPSAT. Until the recent work of Finger and De Bona [11], there was little evidence of phase transition for PSAT in the literature. Baiotelli et al. [2] have shown empirical results for the 2CPA problem—that is equivalent to the PSAT in normal form with only two literals per clause—in which a typical phase transition shape is seen. However, 2CPA is not exactly a normal form of PSAT, even though it is NP-complete. As our normal form extends Finger and De Bona’s, and GPSAT is also NP-complete, we expect to find evidence of phase transition also for GPSAT. Consequently, we examine the behavior of GPSAT using instances $\Psi \wedge \Gamma$ in normal form for various values of m/n for fixed q and n , looking for hard instances instead of randomly trying out large instances that in the end may be easy.

Following the approach from Finger and De Bona [11], we fix the number of clauses in Ψ , q , and the total number of probabilistic literals in Ψ , $3q$. Each probability assessment $P(A_i) \bowtie_i \alpha_i$ in Ψ was generated in such a way that $\bowtie_i \in \{<, \geq\}$ and $\alpha_i \in [0, 1]$ were chosen under a



uniform distribution, discretizing $[0, 1]$ using two decimal digits. Clauses with three literals each in γ were randomly generated by selecting atomic propositions randomly out of the n propositions; each literal was negated or not with probability $1/2$. We varied the number m of

clauses in γ in order to explore the complexity profile of GPSAT for m/n . For each point m/n , 100 random GPSAT instances in normal form were generated, submitted to the implementation of Algorithm 1, and the average computation time was registered together with the percentage



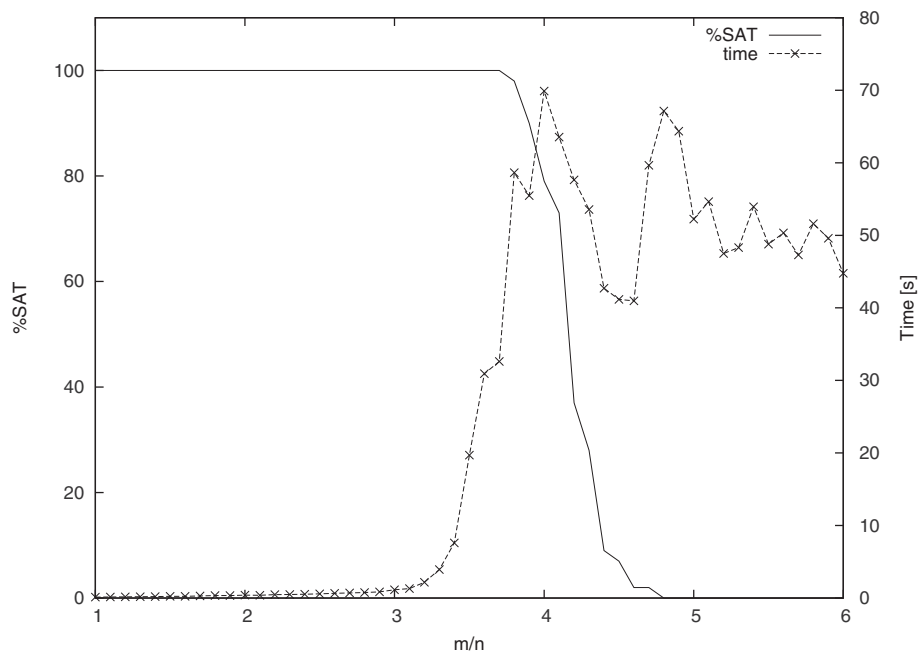


Fig. 5 GPSAT phase transition. Ratio of satisfiable instance and average computation time for each number m of clauses in Γ , with $n = 100$ and $q = 15$

of satisfiable formulas. Figures 1, 2, 3, 4, and 5 shows the results of such experiments, for different values of q and n .

An easy-hard-easy pattern, typical from phase transition phenomena, is seen in Figs. 1, 2, 3, 4, and 5. The hardest instances were found close to the transition between regions of satisfiable and unsatisfiable instances; furthermore, unsatisfiable instances were found harder to solve than satisfiable ones. If $q = 0$, the problem is Boolean satisfiability, and the point of approximately 50 % of satisfiable instances is around $m/n = 4.3$ [12]. Here we add some probabilistic constraints, so the phase transition point has moved leftwards, as expected.

Comparing to the results in ref. [6] solving GPSAT seems harder than PSAT. This is not surprising, since it is known that disjunctive linear programming is considerably harder than linear programming.

Conclusions

In this paper, we have introduced the generalized probabilistic satisfiability problem, a normal form for it, and an algorithm to solve it. GPSAT is considerably more expressive than PSAT, allowing negation and disjunction of probabilistic assessments and probabilistic nesting as well, but it is still NP-complete. Based on an integer linear programming solution to PSAT, we introduced an integer linear programming solution for GPSAT. Evidence for phase transition was found in our initial experiments,

but an exhaustive investigation is needed to confirm and understand this phenomenon in GPSAT and PSAT.

As future work, it would be interesting to enlarge our language to embed linear combinations of probabilistic assignments and conditional probabilities. Another path would be to generalize the coherence checking problem, as we did for PSAT, for de Finetti's framework for probability.

Endnotes

¹ $\omega(\neg\phi) = 1$ iff $\omega(\phi) = 0$; $\omega(\phi_1 \wedge \phi_2) = 1$ iff $\omega(\phi_1) = 1$ and $\omega(\phi_2) = 1$; $\omega(\phi_1 \vee \phi_2) = 1$ iff $\omega(\phi_1) = 1$ or $\omega(\phi_2) = 1$; $\omega(\phi_1 \rightarrow \phi_2) = 1$ iff $\omega(\phi_1) = 0$ or $\omega(\phi_2) = 1$.

² For simplicity, we left the connective \rightarrow outside the language, without loss of generality.

³ If a solution can be reached with less than $q + 1$ valuations with positive probability, then some \mathbf{a}_j can be repeated with zero probability.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

FGC suggested the idea, provided the basis for the implementation of the algorithm, and contributed to the text. GDB completed the implementation, ran the experiments, and contributed to the text; he is a PhD candidate under supervision of MF. All authors read and approved the final manuscript.

Acknowledgements

GDB is supported by CAPES. FGC is partially supported by CNPq. MF is partially supported by CNPq grant PQ 302553/2010-0.

Author details

¹Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, Sao Paulo, Brazil. ²Escola Politécnica, Universidade de São Paulo, Avenida Professor Luciano Gualberto 380, Sao Paulo, Brazil.

Received: 2 May 2014 Accepted: 10 June 2015

Published online: 15 July 2015

References

- Andersen KA, Pretolani D (2001) Easy cases of probabilistic satisfiability. *Ann Math Artif Intell* 33(1):69–91
- Baiocchi M, Capotorti A, Tiberi P, Tulipani S (2004) An empirical complexity for a 2CPA solver. 10th International Conference IPMU:1857–1864
- Boole G (1958) *An Investigation on the Laws of Thought*, Dover, New York
- Bruno G, Gilio A (1980) Applicazione del metodo del semplice al teorema fondamentale per le probabilità nella concezione soggettivistica. *Statistica* 40:337–344
- Chandru V, Hooker J (1999) *Optimization Methods for Logical Inference*, John Wiley & Sons Inc. ISBN: 978-0-471-57035-6
- Cozman FG, di Ianni LF (2013) Probabilistic satisfiability and coherence checking through integer programming. In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, The Netherlands. pp 145–156
- De Bona G, Cozman FG, Finger M (2014) Towards classifying propositional probabilistic logics. *J Appl Logic* 12(3):349–362
- Eckhoff J (1993) Helly, Radon, and Carathéodory type theorems. In: Grüber P, Wills J (eds). *Handbook of convex geometry*, North-Holland, Amsterdam, 1993. pp 389–448
- Fagin R, Halpern JY, Megiddo N (1990) A logic for reasoning about probabilities. *Inform Comput* 87:78–128
- Fagin R, Halpern JY (1994) Reasoning about knowledge and probability. *J ACM* 41(2):340–367
- Finger M, De Bona G (2011) Probabilistic satisfiability: Logic-based algorithms and phase transition. In: *IJCAI*. pp 528–533. doi:10.5591/978-1-57735-516-8/IJCAI11-096
- Gent IP, Walsh T (1994) The SAT phase transition. In: 11th European Conference on Artificial Intelligence. pp 105–109
- Georgakopoulos G, Kavvadias D, Papadimitriou CH (1988) Probabilistic satisfiability. *J Complexity* 4:1–11
- Hailperin T (1965) Best possible inequalities for the probability of a logical function of events. *72:343–359*
- Hailperin T (1976) *Boole's Logic and Probability: a Critical Exposition from the Standpoint of Contemporary Algebra, Logic, and Probability Theory*, North-Holland, Amsterdam
- Halpern JY (2003) *Reasoning about Uncertainty*. MIT Press, Cambridge, Massachusetts
- Hansen P, Jaumard B (1996) Probabilistic Satisfiability. Technical Report G-96-31, Les Cahiers du GERAD, École Polytechnique de Montréal
- Hansen P, Perron S (2008) Merging the local and global approaches to probabilistic satisfiability. *Int J Approximate Reasoning* 47(2):125–140
- Jaumard B, Hansen P, de Aragão MP (1991) Column generation methods for probabilistic logic. *ORSA J Comput* 3(2):135–148
- Lukasiewicz T (2008) Expressive probabilistic description logics. *Artif Intell* 172(6–7):852–883
- Ng R, Subrahmanian VS (1992) Probabilistic logic programming. *Inform Comput* 101(2):150–201
- Nilsson NJ (1986) Probabilistic logic. *Artif Intell* 28:71–87
- Uchii S (1973) Higher order probabilities and coherence. *Philos Sci* 40(3):373–381
- Williams HP (2009) *Logic and integer programming*. Springer. ISBN 978-0-387-92280-5

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com